The Dissertation Committee for Md. Shamsuzzoha Bayzid
certifies that this is the approved version of the following dissertation:

Estimating Species Trees from Gene Trees Despite Gene Tree Incongruence under Realistic Model Conditions

Committee:

Joydeep Ghosh, Supervisor

Tandy Warnow, Co-Supervisor

Vijaya Ramachandran

Greg Plaxton

Pradeep Ravikumar

Estimating Species Trees from Gene Trees Despite Gene Tree Incongruence under Realistic Model Conditions

by

Md. Shamsuzzoha Bayzid, M.S.E.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2016

Dedicated to my parents.

Acknowledgments

It has really been a long journey towards achieving the PhD. This journey would not have been a successful one without the help and support of many people throughout my life who I would like to thank.

First of all, I would like to declare that all the praises belong to the almighty Allah who endowed me with the capability to carry out this work successfully. I deeply express my sincere gratitude to the endless kindness of Allah.

I would like to express my deep gratitude to Professor Tandy Warnow. She has always been there to guide me ever since I joined her lab in 2010. I would like to thank her for introducing me to the fascinating and prospective field of computational phylogenetics. Having the privilege of working with her has been a great learning experience for me where I have learned to appreciate the value and importance of experimental works. As I mainly worked on theoretical computer science during my undergraduate and master's theses, it was never easy for me to cope up with the extensive experimental studies I had to conduct in her lab. I would like to thank her for providing me with such excellent training. I owe her a lot of gratitude for her patience in reviewing many of my inferior drafts, encouragement for working hard, and invaluable suggestions and guidance. Thank you Professor Warnow for your wholehearted supervision throughout the progress of this work, without which this thesis would not have been materialized.

I would like to thank Professor Joydeep Ghosh who I have been working with since Professor Warnow left UT Austin. That transition period was very difficult for me, and words really cannot describe how much grateful I am to Professor Ghosh for extending his unwavering support. His vast knowledge in data mining and machine learning helped me to extend the horizon of my knowledge. I would like to express my sincere gratitude to Professor Ghosh for accepting me as an advisee at a difficult moment and for supporting and guiding me to a successful path.

I owe special thanks to my committee members: Professor Vijaya Ramachandran, Professor Greg Plaxton and Professor Pradeep Ravikumar for their invaluable comments, thoughtful questions and ideas on how to expand the scope of my work. It has been a great honor and privilege for me to have them on my doctoral committee.

I also like to thank Professor Md. Saidur Rahman who supervised my undergraduate and master's theses at Bangladesh University of Engineering and Technology (BUET). All the skills and expertise I have in theoretical computer science, especially in graph algorithms and proof techniques, are thanks to Professor Rahman. Professor Rahman is more than just an academic supervisor to me. I am indebted to him for his moral support, inspiration, and everything that he has done to see myself as a PhD student at a prestigious university in the USA. Having the privilege of being his advisee, I can easily relate to the sentiment why people refer academic advisors as academic parents. He has always been and will always be a true inspiration for me. I would also like to express my deepest gratitude to all of my teachers at BUET, especially Professor Mohammad Kaykobad, Professor Masud Hasan, Professor Ashikur Rahman, Professor Sohel Rahman and Professor Mahmuda Naznin for their all-out support and encouragement.

I have had the great pleasure of being a member of the Phylo lab and working with wonderful colleagues, including Kevin, Jimmy, Andrei, Nathan, Nam, Shell, Siavash, Tyler and Théo. I must acknowledge the contributions of Jimmy and Andrei in the first year of my PhD. Jimmy helped me a lot to get familiar with different tools and softwares, and also helped me to set up the experimental pipelines. It was a pleasure for me to work with Andrei on some interesting theoretical problems on phylogenomics. He has been a very good friend to have around and discuss research and various interesting topics like politics, religion etc. I am grateful to Siavash for his help throughout my PhD. I also wish to thank Jeorge Álvarez who visited our lab in Fall 2013, and we had useful discussions on how to make existing phylogenomic methods scalable. I owe special thanks to everyone in Dr. Ghosh's lab for their welcoming manner when I joined the lab.

I am grateful to the Fulbright science and technology PhD program for their generous support during the first three years of my PhD. Especially, I acknowledge Sarah (the senior program officer) for her help and support. I would like to thank the staffs at the CS department and the international student and scholar services, especially Darcy McGillicuddy, Katherine Utz, Lydia Griffith and Tiffany Buckley for all that they have done for me. I also wish to thank Laurie Alvarez for being so nice and wonderful.

I do not know how to express my sincere gratitude to the Nueces mosque and everyone associated with it. I have been really fortunate to have a mosque around to go and pray. Being able to pray in the mosque was a source of my mental happiness that helped me to alleviate the stress and uncertainty of the PhD life.

I would like to thank my parents from the bottom of my heart. Everything I have achieved in my life is thanks to my parents. They are the reason I am here today. My mother is the best teacher I have ever had who teaches me, by her example, to be honest, polite, modest and kind, and always encourages me to be a good human being in general. I really don't know how to express my gratefulness to my parents, all I can say is – thank you Abba and Amma! I am indebted to my brothers and sisters for supporting me spiritually throughout my PhD and my life in general. I also owe special thanks to my parents-in-law for appreciating my works and inspiring me to do something better.

Finally, I must acknowledge my wife Rezwana for her unconditional love, relentless support, encouragement and patience throughout my PhD life. She was always there, in good times and in bad times, to boost my confidence and pave a smooth way for this long journey. Although Rezwana may not possibly know how much of a help she has been, I would like to express my utmost gratitude to her for all the favors she has done to me. Our daughter, the little princess Zunairah, was born just a couple of months before my dissertation defense, and she has been a constant source of joy and happiness ever since. Thank you Zunairah, you definitely are a part of my journey towards achieving the PhD.

Estimating Species Trees from Gene Trees Despite Gene Tree Incongruence under Realistic Model Conditions

Publication No. ________________

Md. Shamsuzzoha Bayzid, Ph.D.
The University of Texas at Austin, 2016

Supervisors: Joydeep Ghosh
Tandy Warnow

Species tree estimation is frequently based on phylogenomic approaches that use multiple genes from throughout the genome. With the rapid growth rate of newly sequenced genomes, species tree inference from multiple genes has become one of the basic and popular tasks in comparative and evolutionary biology. However, combining data on multiple genes is not a trivial task since genes evolve through biological processes that include *deep coalescence* (also known as *incomplete lineage sorting* (ILS)), *duplication and loss*, *horizontal gene transfer* etc., so that the individual gene histories can differ from each other. In this dissertation, we focus on making advances on phylogenomic analyses with particular attention to the *gene tree discordance.* In addition to gene tree discordance, we consider other challenging conditions that frequently arise in genome scale data. One of these major challenges is *incomplete gene trees*, meaning that not all gene trees have individuals from all the species. We performed an extensive simulation study under the *multi-species coalescent* (MSC) model that shows that existing methods have poor accuracy when gene trees are incomplete. We formalized the *optimal completion problem*, which seeks to add the missing taxa (species) into the gene trees with respect to a species tree such that the distance (in terms of ILS) between the gene tree and the species tree is minimized. We developed an algorithm for solving this problem. We formalized optimization problems in the context of species tree estimation from a set of incomplete gene trees under the multi-species coalescent model, and proposed algorithms for solving these problems. We formulated different mathematical models for "gene loss" based on different reasons for incompleteness. Next, we addressed the *Minimize Gene Duplication* (MGD) problem, that seeks to find a species tree from a set of gene trees so as to minimize the total number of duplications needed to explain the evolutionary history. We proposed exact and heuristic algorithms to solve this NP-hard problem. Next, we showed in a comprehensive experimental study that existing methods are susceptible to poorly estimated gene trees in the presence of ILS. We proposed a new technique called "binning" that dramatically improves the performance of species tree estimation methods when gene trees are poorly estimated. We developed a novel technique called "naive binning" and subsequently proposed an improved version called "weighted statistical binning" to address the problem of gene tree estimation error. Finally, we addressed the computational challenges to reconstruct highly accurate species tree from large scale genomic data. We developed divide-and-conquer based meta-methods that can make existing methods scalable to very large datasets (in terms of the number of species). Overall, this dissertation contributes to understanding the limitations of the existing methods under realistic model conditions, developing new approaches to handle the challenging issues that frequently arise in phylogenomics, and improving and scaling the existing methods to larger datasets.

Table of Contents

Acknowledgments **v**

Abstract **x**

List of Tables **xix**

List of Figures **xxiv**

Chapter 1. Introduction **1**

1.1 Our contributions 3

1.2 Organization of the dissertation 4

Chapter 2. Preliminaries **9**

2.1 Phylogenies 9

2.2 Gene tree and species tree 14

2.2.1 Gene tree reconciliation 21

2.2.2 Duplication, loss and deep coalescent events: mathematical formulation 24

2.2.2.1 Duplication and loss 24

2.2.2.2 Extra lineages 26

2.2.3 Gene tree parsimony 26

2.2.4 Statistical consistency 27

2.3 Phylogenomic analysis pipeline 28

2.3.1 Existing methods 29

2.4 Evaluation of species tree estimation methods 30

2.4.1 Evaluation on simulated datasets 30

2.4.2 Error metrics 31

2.4.3 Evaluation on real biological datasets 34

2.5 New data structures 34

Chapter 3. Inferring Optimal Species Trees under Gene Duplication and Loss 36

3.1 Introduction 37

3.2 Basics 41

3.2.1 Prior terminology and theory 41

3.3 New theorems 43

3.4 Algorithms for MGD_r on rooted binary gene trees 46

3.4.1 Graph-theoretic characterization of optimal solution to MGD_r 46

3.4.2 Dynamic programming algorithm for MGD_r 48

3.5 Algorithms for $MGDL_r$ 52

3.5.1 Graph-theoretic characterization 52

3.5.2 Dynamic programming approach for $MGDL_r$ 56

3.6 Conclusion 57

Chapter 4. Inferring Optimal Species Trees under Gene Duplication and Loss: Beyond Rooted Gene Trees 59

4.1 Introduction 60

4.2 Prior terminology and theory 61

4.3 MGD_{ru} and $MGDL_{ru}$ 61

4.4 MGD and MGDL on unrooted, binary, and incomplete gene trees 63

4.4.1 Optimal rooting 67

4.4.2 Solving MGD_{ru} 74

4.4.3 Extension to $MGDL_{ru}$ 78

4.4.4 Solving $MGDL_{ru}$ 80

4.5 Conclusion 82

Chapter 5. Gene Tree Parsimony for Incomplete Gene Trees under Gene Duplication and Loss 83

5.1 Introduction 84

5.2 Basics 86

5.2.1 Notation and terminology 86

5.2.2 Reconciliation of gene trees and species trees 88

5.2.3 The standard formula for computing losses 89

5.2.4 Incompleteness due to sampling strategies 90
5.2.5 Incompleteness due to gene birth and death 94
5.3 How to calculate losses 96
5.4 Algorithms to find species trees 101
5.4.1 Basic material 103
5.4.2 Deriving $L_{bd}(gt, ST)$ and $L^*_{bd}(gt, ST)$ 105
5.4.3 Assigning weights to subtree-bipartitions 110
5.4.4 Dynamic programming algorithm 114
5.4.5 Extensions 116
5.5 Conclusion 116

Chapter 6. Gene Tree Parsimony for Incomplete Gene Trees under ILS 118

6.1 Introduction 119
6.2 Theoretical results for MDC 121
6.2.1 MDC for complete gene trees 123
6.2.2 Extension to incomplete gene trees 125
6.3 Establishing the relationship between MDC_{inc} and $MBMC_{inc}$ 128
6.4 Materials and methods 132
6.4.1 Overview 132
6.4.2 Datasets 135
6.5 Results 135
6.5.1 Missing branch rates 135
6.5.1.1 Results on 11-taxon datasets 135
6.5.1.2 Results on 17-taxon datasets 136
6.5.1.3 Results on 100-taxon datasets 139
6.5.1.4 Overall results 139
6.5.2 Computational Issues 141
6.6 Discussion 142
6.7 Conclusion 145

Chapter 7. Naive Binning **147**
7.1 Introduction . . . 148
7.2 Materials and methods . . . 151
7.3 Results . . . 156
7.4 Discussion . . . 172
7.5 Additional discussion . . . 174
7.5.1 Previous studies comparing concatenation to coalescent-based estimation of species trees . . . 174
7.5.2 Limitations on binning . . . 181
7.6 Conclusion . . . 182

Chapter 8. Weighted Statistical Binning **184**
8.1 Introduction and background . . . 186
8.2 Weighted statistical binning . . . 188
8.3 Experimental study . . . 194
8.3.1 Datasets . . . 194
8.3.2 Methods . . . 198
8.3.3 Measurements . . . 199
8.4 Results and discussion . . . 201
8.4.1 Biologically-based simulated datasets . . . 201
8.4.2 Summary of observations . . . 225
8.5 Conclusion . . . 229

Chapter 9. Disk Covering Methods Improve Phylogenomic Analyses **235**
9.1 Introduction . . . 236
9.2 Methods . . . 238
9.3 Experiments . . . 243
9.3.1 Mammalian simulated datasets . . . 244
9.3.2 Biological datasets . . . 245
9.4 Results . . . 246
9.4.1 Running time on simulated datasets . . . 246
9.4.2 Impact of boosting on topological accuracy for simulated datasets . . . 249

9.4.3 Results on biological datasets 255
9.4.4 Pseudo-likelihood scores and quartet support values . . 259
9.4.5 Robustness to the starting trees 262
9.4.6 Statistical consistency 266
9.5 Discussion 266
9.6 Conclusion 269

Chapter 10. Conclusions 270

Appendices 277

Appendix A. Supplementary Materials for Naive Binning 278
A.1 Methods 278
A.1.1 Overview 278
A.1.2 Datasets 280
A.1.3 Methods 283
A.1.3.1 Gene tree estimation 283
A.1.3.2 Species tree estimation 284
A.1.4 Running time 292
A.2 Additional results 296
A.2.1 Experiment 1: Evaluating fast species tree estimation methods on 100 replicate datasets 296
A.2.2 Experiment 2: Evaluating species tree estimation methods on 20 replicate datasets 297
A.2.3 Experiment 3: Evaluating gene tree estimation error . . 298
A.2.4 Experiment 4: Evaluating summary methods on gene trees estimated by *BEAST 301
A.2.5 Experiment 5: Evaluating the impact of naive binning on fast methods - 100 replicate datasets 303
A.2.6 Experiment 5: Evaluating the impact of naive binning on all methods - 20 replicate datasets 310

Appendix B. Supplementary Materials for Weighted Statistical Binning 317

B.1 Evaluation 317

B.1.1 Triplet gene tree distribution error 318

B.1.2 Species tree estimation error and branch support 319

B.1.3 Simulated datasets 320

B.1.4 Biological datasets 323

B.2 Simulation protocols for 15- and 10-taxon datasets 326

B.3 Supplementary tables 330

B.4 Supplementary figures 332

Index 344

Bibliography 347

List of Tables

3.1 **Demonstration of the weights of the nodes in the compatibility graph illustrated in Fig. 3.1**. 50

6.1 **Acronyms used in this chapter** 122

6.2 **Software used in this study** 122

8.1 **Topological discordance between true gene trees and true species tree**. For each collection of simulated datasets (defined by the type of simulation and the ILS level), we show the average topological distance between true gene trees and the species tree. 198

8.2 **Statistical significance test results for choice of binning method on MP-EST.** We performed ANOVA to test the significance of the choice of methods (unbinned, weighted binned, unweighted binned, WSB-50: weighted statistical binning using 50% bootstrap support threshold and WSB-75: weighted binning using 75% bootstrap support threshold). For weighted vs. unweighted, we compared 50% bootstrap support threshold for avian, 75% for mammalian, and both 50% and 75% for 15- and 10-taxon datasets. All p-values are corrected for multiple hypothesis testing using the FDR correction ($n = 16$). "n.a." stands for "not available". 204

8.3 **Statistical significance test results for choice of binning method on ASTRAL.** We performed ANOVA to test the significance of the choice of methods (unbinned, weighted binned, unweighted binned, WSB-50: weighted statistical binning using 50% bootstrap support threshold and WSB-75: weighted binning using 75% bootstrap support threshold). For weighted vs. unweighted, we compared 50% bootstrap support threshold for avian, 75% for mammalian, and both 50% and 75% for 15- and 10-taxon datasets. All p-values are corrected for multiple hypothesis testing using the FDR correction ($n = 14$). "n.a." stands for "not available". 206

8.4 **Statistical significance test results for interaction effects (binning and simulation parameter) on MP-EST.** We performed ANOVA to test the significance of whether there is an interaction between the choice of the method (unbinned, weighted binned, unweighted binned, WSB-50: weighted statistical binning using 50% bootstrap support threshold and WSB-75: weighted statistical binning using 75% bootstrap support threshold) and the variable changed in each dataset. For weighted vs. unweighted, we compared 50% bootstrap support threshold for avian, 75% for mammalian, and both 50% and 75% for 15- and 10-taxon datasets. All *p*-values are corrected for multiple hypothesis testing using the FDR correction ($n = 21$). “n.a.” stands for “not available”. 207

8.5 **Statistical significance test results for interaction effects (binning and simulation parameter) on ASTRAL.** We performed ANOVA to test the significance of whether there is an interaction between the choice of the method (unbinned, weighted binned, unweighted binned, WSB-50: weighted statistical binning using 50% bootstrap support threshold and WSB-75: weighted statistical binning using 75% bootstrap support threshold) and the variable changed in each dataset. For weighted vs. unweighted, we compared 50% bootstrap support threshold for avian, 75% for mammalian, and both 50% and 75% for 15- and 10-taxon datasets. All *p*-values are corrected for multiple hypothesis testing using the FDR correction ($n = 17$). “n.a.” stands for “not available”. 208

9.1 ***p*-values measured by Wilcoxon signed-rank test for the simulated mammalian datasets**. We evaluate the statistical significance of differences in species tree topology using Wilcoxon signed-rank test with $\alpha = 0.05$. We show the *p*-values indicating whether the differences between two methods are statistically significant. We compare concatenation (CA) and MP-EST (unboosted) with SSG and DACTAL-boosted MP-EST. . 254

9.2 **Average log likelihood values (over 20 replicates) for different species trees. We estimated the log likelihood values using MP-EST**. We show the likelihood values for the initial tree estimated by MP-EST and the true species tree (which is also estimated by MP-EST from the biological datasets). For SSG and DACTAL-based boosting, we find the best tree across the five iterations with respect to the log likelihood value estimated by MP-EST with branch length optimization. The best likelihood values are shown in bold. 261

9.3 **Average quartet supports of different species trees. We show the average (over 20 replicates) number of satisfied quartets (in the input gene trees) by different species trees for various model conditions**. For SSG and DACTAL-based boosting, we find the best tree across the five iterations with respect to the number of satisfied quartets. The best quartet support values are shown in bold. 261

A.1 **Average missing branch rates (over 20 replicates) of gene trees estimated by different methods on 11-taxon strongILS datasets**. *BEAST could not be run on 100-gene datasets. Experiment 3. 298

A.2 **Average missing branch rates (over 20 replicates) of gene trees estimated by different methods on 11-taxon weakILS datasets**. Experiment 3. 299

A.3 **Average missing branch rates over 20 replicates of gene trees estimated by different methods on 17-taxon datasets**. Experiment 3. 299

A.4 **Average missing branch rates for methods (unbinned and binned) on 11-taxon strongILS 25, 50 and 100-gene cases**. Results are shown for 100 replicates. Each bin contains 5 genes. BUCKy (unbinned) was not run on 100 replicates. Experiment 5. 305

A.5 **Evaluating the statistical significance of using binning on fast methods, when analyzing 100 replicate 11-taxon strongILS datasets**. We show *p*-values for the statistical significance of a difference between binned and unbinned analyses. Each bin has 5 genes. Experiment 5. 307

A.6 **Average FN rates for methods (unbinned and binned) on 11-taxon weakILS 25-gene case; n = 100**. Each bin contains 5 genes. We did not run *BEAST or BUCKy on 100 replicates. Experiment 5. 307

A.7 **Evaluating the impact of binning on fast methods on 100 replicate 11-taxon weakILS datasets with 25 genes**. We show *p*-values for the statistical significance of a difference between binned and unbinned analyses. Each bin has 5 genes. Experiment 5. 308

A.8 **Average FN rates for methods (unbinned and binned) on 17-taxon 32-gene case; n = 100**. Each bin contains 4 genes. We did not run unbinned BUCKy on 100 replicates. Experiment 5. 308

A.9 **Evaluating the impact of binning for fast methods (binned vs. unbinned) on 100 replicates of 17-taxon 32-gene dataset**. We show p-values for the statistical significance of binned versus unbinned analyses. Each bin has 4 genes. Experiment 5. 309

A.10 **Average FN rates for methods (unbinned and binned) on 11-taxon strongILS 25, 50 and 100-gene cases; n = 20**. We do not show results for unbinned *BEAST on 100 genes, because it was not run to convergence. Each bin contains 5 genes. Experiment 5. 311

A.11 **Evaluating the impact of binning on all methods, applied to 20 replicates of the 11-taxon strongILS datasets**. We show p-values. We were not able to run *BEAST (unbinned) on 100-gene datasets. Experiment 5. 314

A.12 **Average FN rates for methods (unbinned and binned) on 17-taxon 32-gene case; n = 20**. Each bin contains 4 genes. Experiment 5. 314

A.13 **Evaluating the impact of binning on species tree estimation methods on 20 replicates of the 11-taxon weakILS datasets with 25 genes**. We show p-values for methods (binned vs. unbinned methods). Each bin has 5 genes. Experiment 5. 315

A.14 **Average FN rates for methods (unbinned and binned) on 11-taxon weakILS 25-gene case; n = 20**. Each bin contains 5 genes. Experiment 5. 315

A.15 ***p*-values for methods (binned vs. unbinned) on 20 replicates of 17-taxon 32-gene dataset**. Each bin has 4 genes. Experiment 5. 316

B.1 **Gene tree estimation error, with and without binning for simulated datasets**. We show the average gene tree estimation error for the simulated datasets analyzed in this study. Results are shown for fixed number of genes (1000 for avian and 200 for mammalian, 100 for 15-taxon and 100 for 10-taxon). We fixed the level of ILS to 1X for avian, mammalian and 15-taxon datasets; and varied the level of ILS for 10-taxon datasets with 100bp sequence length. Gene tree error is mean topological distance, measured using the missing branch rate between the true gene tree and all 200 bootstrap replicates of each estimated gene tree. For the supergene trees, each bootstrap replicate of each supergene tree is compared separately against each true gene tree for the genes put in that bin. "n.a." stands for "not available". 330

B.2 **Average bootstrap support**. We show the average bootstrap support values of the estimated gene trees for the simulated datasets. Results are shown for fixed number of genes (1000 for avian and 200 for mammalian, 100 for 15-taxon and 100 for 10-taxon datasets). We fixed the level of ILS to 1X for avian, mammalian and 15-taxon datasets; and varied the level of ILS for 10-taxon datasets with 100bp sequence length. 331

List of Figures

2.1 **Phylogenetic tree**. A phylogenetic tree relating four species: humans, chimpanzees, gorillas and orangutans. 10

2.2 **Unrooted and rooted trees**. (a) An unrooted tree, and (b) the rooted tree resulting from rooting the tree shown in (a) on the edge e. 11

2.3 (a) A binary tree representing the evolutionary history of 6 species: $\{A, B, C, D, E, F\}$, and (b) a non-binary tree on the same set of taxa, where node u is a polytomy. 12

2.4 **Refinement and contraction.** T is a refinement of T'; alternatively, T' is a contraction of T. Here T' can be obtained from T by contracting the edge incident on u and v. 13

2.5 **Gene tree-species tree incongruence**. A species tree (given in blue) and a gene tree (given in red) on the same set $\{A, B, C, D\}$ of taxa with different topologies. 15

2.6 **Reconciliation under gene duplication and loss model**. We show a reconciliation of the discordant gene tree-species tree pair shown in Fig. 2.5 using gene duplication and loss. We embed the reconciled tree inside the species boundaries. Duplicated gene copes (dashed and solid red lines) evolve independently and can go extinct. This reconciliation requires one duplication and three gene losses. 16

2.7 **Reconciliation under incomplete lineage sorting**. We show a reconciliation of the discordant gene tree-species tree pair shown in Fig. 2.5 using incomplete lineage sorting. We embed the reconciled tree inside the species boundaries. Going back in time, the gene copies within species B and C first meet at their corresponding speciation point (i.e, the most recent common ancestor of species B and C), but fail to coalesce at the speciation point. Both of these copies go further back in time, and hence we have two gene lineages (dashed and solid black lines) on deeper ancestral branch. Therefore we have one *extra lineage* on the ancestral branch. The gene from C first coalesces with the gene from species D, and subsequently with the gene from B. 18

2.8 **Effect of effective population size and branch length on incomplete lineage sorting under the multi-species coalescent model**. (a) Long branch (larger number of generations) and small population size increase the chance that two gene lineages will coalesce with each other; (b) Larger population size and smaller branch increase the possibility that gene lineages will fail to coalesce before reaching the deeper speciation event. 19

2.9 **Reconciliation under horizontal gene transfer**. We show a reconciliation of the discordant gene tree-species tree pair shown in Fig. 2.5 using horizontal gene transfer. Here, the gene lineage from species D moves horizontally across species boundaries and enters into the species boundary of C. This “foreign” gene lineage is maintained and spread into the receiving species population. If the receiving lineage (C) goes extinct or is not sampled, then there will be discordance between the species tree and the gene tree. 20

2.10 **Illustration of the $\mathcal{M}$ mapping of the nodes in a gene tree gt with respect to a species tree ST.** 22

2.11 **Optimal and non-optimal reconciliations under the gene duplication and loss model**. (a) A rooted, binary gene tree gt, (b) an optimal reconciliation of gt with a rooted, binary species tree that yields 1 duplication and 3 losses, and (c) a non-optimal reconciliation of gt using 1 duplication and 4 losses. 23

2.12 **Optimal and non-optimal reconciliations under the deep coalescence model**. (a) A rooted, binary gene tree gt, (b) an optimal reconciliation of gt with a rooted, binary species tree that yields 1 extra lineage, and (c) a non-optimal reconciliation of gt using 2 extra lineages. 23

2.13 **Illustration of a simulation protocol for evaluating species tree estimation techniques.** We start with a model species tree. Next, a collection of true gene trees are evolved down this model species tree, and gene sequences are evolved down the collection of true gene trees. Next, we estimate gene trees from the gene sequence alignments. Finally, a species tree is estimated from the estimated gene trees, and compared to the true species tree. 32

2.14 **Missing branch rate.** The branch separating $\{u, v, w\}$ and $\{x, y\}$ is not reconstructed in the estimated tree. 32

3.1 **Illustration of the constrained version of our algorithm for MGD**. (a) Three gene trees gt_1, gt_2, and gt_3, (b) the compatibility graph $CG(\mathcal{G})$. Cliques of size three correspond to a species tree on $\{a, b, c, d\}$. The maximum weight clique is indicated by dark lines. 49

4.1 **Illustration of different rooted versions of an unrooted tree around a particular internal node.** (a) An unrooted gene tree gt, (b)–(d) three possible rooted versions of gt on the three edges incident on u. 67

4.2 (a) An unrooted gene tree containing the root subtree-bipartition of ST ($\mathcal{SBP}_{ST}(root(ST)) = X|Y$), (b) an internal node inside the subtree induced by Y. 70

4.3 (a) An unrooted gene tree gt and (b) a rooted tree ST. Here $X = X_1 \cup X_2 \cup \{u\}$. 74

4.4 (a) An unrooted gene tree gt, (b) a rooted binary species tree ST. 77

5.1 (a) A gene tree gt, (b) a species tree ST. Here, $\mathcal{M}(r(gt)) \neq r(ST)$, and $UMMC(gt, ST) = \{\{e\}, \{f, g\}\}$. $L_{bd}(gt, ST)$ is the number of losses required to reconcile gt with ST according to the Eqn. 5.2, and we get $L^*_{bd}(gt, ST)$ by adding $|UMMC(gt, ST)|$ to $L_{bd}(gt, ST)$. 99

5.2 (a) Gene tree topology tp_1, and (b) gene tree topology tp_2. . . 101

6.1 **Average missing branch rates of methods on twenty (20) 11-taxon 10-gene datasets on true alignments (TA)**. Gene trees are estimated using FastTree-2 (FT), and in some cases the branches with support less than 75% are contracted (FT-75). From top to bottom, the number of missing taxa is 2, 3, and 5. 137

6.2 **Average missing branch rates of methods on twenty (20) 17-taxon 8-gene datasets on true alignments (TA)**. Gene trees are estimated using FastTree-2 (FT), and in some cases the branches with support less than 75% are contracted (FT-75). From top to bottom, the number of missing taxa is 1, 5, and 8. 138

6.3 **Average missing branch rates of methods on ten (10) 100-taxon 25-gene datasets on true alignments (TA)**. Gene trees are estimated using FastTree-2 (FT), and in some cases the branches with support less than 75% are contracted (FT-75). From top to bottom, the number of missing taxa is 10, 30, and 50. 140

7.1 **Results for "fast" methods on 100 replicate 11-taxon weakILS models.** CA-ML uses just the input alignments, and the other methods use gene trees estimated using RAxML. We show average FN rates with standard error bars. 158

7.2 **Results for "fast" methods on 100 replicate 11-taxon strongILS models.** We show results (average FN rate with standard error bars) for up to 100 genes. 159

7.3 **Results for "fast" methods on 100 replicate 17-taxon models.** We show average FN rates with standard error bars. 159

7.4 **Results for *BEAST, BUCKy, MP-EST, and CA-ML on 20 replicate 11-taxon weakILS datasets.** We show average FN rates with standard error bars. CA-ML and *BEAST return the true tree on the 25-gene case, and all methods shown return correct trees on the 50-gene case. Therefore, no results are shown for datasets with 100 genes. 160

7.5 **Results for *BEAST, BUCKy, MP-EST, and CA-ML on 20 replicate 11-taxon strongILS datasets.** We show average FN rates with standard error bars. 161

7.6 **Results for *BEAST, BUCKy, MP-EST, and CA-ML on 20 replicate 17-taxon datasets.** We show average FN rates with standard error bars. 161

7.7 **Gene tree estimation error rates on 17-taxon datasets.** Average and standard error bars (over 20 replicates) of *BEAST, RAxML, and FastTree-2. 164

7.8 **Results of binning experiments of the fast methods on 100 replicates of the 11-taxon 25-gene strongILS datasets.** We show average FN rate with standard error bars. Each bin contains 5 genes. We omit BUCKy on unbinned genes and *BEAST (binned or unbinned) because these are too slow to run on all 100 replicates within our time limits. CA-ML is not impacted by binning because it uses an unpartitioned analysis. 166

7.9 **Results of the binning experiment for the fast methods on 100 replicates of the 11-taxon 50-gene strongILS datasets.** We show average FN rates with standard error bars. Each bin contains 5 genes. We omit BUCKy on unbinned genes and *BEAST (whether binned or unbinned) because these are too slow to run on all 100 replicates within our time limits. CA-ML is not impacted by binning, because it uses an unpartitioned analysis. 167

7.10 **Results of the binning experiment for the fast methods on 100 replicates of the 11-taxon 100-gene strongILS datasets.** We show average FN rates with standard error bars. Each bin contains 5 genes. We omit BUCKy on unbinned genes and *BEAST (whether binned or unbinned) because these are too slow to run on all 100 replicates within our time limits. CA-ML is not impacted by binning because it uses an unpartitioned analysis. 167

7.11 **Results of the binning experiment for all methods on 20 replicate 11-taxon 50-gene strongILS datasets.** CA-ML is not impacted by binning since it is an unpartitioned analysis. Each bin contains 5 genes. Average and standard error bars shown. 170

7.12 **Results of the binning experiment for all methods (except *BEAST) on 20 replicate 11-taxon 100-gene strongILS datasets.** Each bin contains 5 genes. Average and standard error bars shown. We omit *BEAST on unbinned genes because it could not run to convergence on this dataset within the time limit; however, we show results for *BEAST on the binned datasets. CA-ML returns the true tree on these data. 171

8.1 **Pipeline for unbinned analyses, unweighted statistical binning, and weighted statistical binning.** The input to the pipeline is a set of sequences for different loci across different species. In the traditional pipeline, a multiple sequence alignment and gene tree is computed for each locus, and then these are given to the preferred coalescent-based summary method, and a species tree is returned. In the statistical binning pipeline, the estimated gene trees are used to compute an incompatibility graph, where each vertex represents a gene, and an edge between two genes indicates that the differences between the trees for these genes is considered significant (based on the bootstrap support of the conflicting edges between the trees). The vertices of the graph are then assigned colors, based on a heuristic for balanced minimum vertex coloring, so that no edge connects two vertices of the same color. The vertices with a given color are put into a bin, and the sequence alignments for the genes in a bin are combined into a supergene alignment. A (supergene) tree is then computed for each supergene alignment using a fully partitioned analysis. In the unweighted binning approach (presented in [90]), these supergene trees are then given to the preferred summary method, and a species tree is returned. In the weighted binning approach presented here, each supergene tree is repeated as many times as the number of genes in its bin, and this larger set is then given to the preferred summary method. 193

8.2 **Divergence of estimated gene tree (triplet) distributions from true gene tree distributions for MP-EST analyses of simulated avian datasets.** In (a), we vary the gene sequence length (250bp genes have the highest error, and 1500bp has the lowest error) and explore 1000 genes under default ILS levels, and in (b) we vary the amount of ILS and fix the number of genes to 1000 and sequence length to 500bp. . 203

8.3 **Species tree estimation error (FN) for MP-EST and ASTRAL with MLBS on avian simulated datasets**. (a) MP-EST on 1000 genes with varying gene sequence length (controlling gene tree error) and with 1X ILS. (b) ASTRAL on the exact same conditions, (c) MP-EST on varying numbers of genes with fixed default level of ILS (1X level) and 500bp sequence length, and (d) MP-EST on varying levels of ILS and 1000 genes of length 500bp. We show results for 20 replicates everywhere, except for 2000 genes that are based on 10 replicates. Binning was performed using 50% bootstrap support threshold. 205

8.4 **Effect of binning on the branch lengths (in coalescent units) estimated by MP-EST using MLBS on the avian and mammalian simulated datasets**. Results are shown for (a) 1000 avian genes of 1X ILS level with varying gene sequence length, (b) 1000 avian genes of 500bp and with varying levels of ILS, and (c) varying number of mammalian genes and varying sequence length (250bp, 500bp, and 1000bp) with 1X ILS level. Results are shown for 20 replicates. We used 50% and 75% bootstrap support threshold for binning on avian and mammalian datasets, respectively. 209

8.5 **Cumulative distribution of the bootstrap support values (obtained using MLBS) of true positive (TP) and false positive (FP) edges estimated by binned and unbinned MP-EST on avian datasets**. In (a) we fix the number of genes to 1000, use default ILS levels, and vary sequence length to control gene tree estimation error, and in (b) we study 1000 genes with 500bp sequence length, and vary ILS levels. We order the branches in the estimated species tree by their quality, so that the true positives with high support come first, followed by lower support true positives, then by false positives with low support, and finally by false positives with high support. The false positive branches with support above 75% are the most troublesome, and the highly supported false positives are indicated by the grey area. When the curve for a method lies above the curve for another method, then the first method has better bootstrap support. We used 50% bootstrap support threshold for binning. 212

8.6 **Species tree estimation error for MP-EST and ASTRAL using MLBS on mammalian simulated datasets**. We show average FN rate over 20 replicates. (a) Results for MP-EST. We varied the number of genes (50, 100, 200, 400 and 800) and sequence length (250bp (43% BS), 500bp (63% BS) and 1000bp (79% BS)) with default amount of ILS (1X level). (b) ASTRAL on varying numbers of genes with fixed 1X ILS level and 500bp sequence length. We used 50% and 75% bootstrap support threshold for binning on avian and mammalian datasets, respectively, and estimated the supergene trees using RAxML with unpartitioned analyses. 215

8.7 **Species tree estimation error for MP-EST and ASTRAL using MLBS on avian and mammalian simulated datasets with two support thresholds (B)**. We show average FN rate for unbinned, and weighted and unweighted binned analyses with both $B = 50\%$ and $B = 75\%$. Results are shown for (a) the avian dataset with 10 replicates of 1000 genes of length 500bp and 1X ILS level, and (b) the mammalian dataset with 20 replicates of 400 mixed genes (200 genes with 500bp and 200 genes with 1000bp) with 1X ILS level. 217

8.8 **Species tree estimation error for MP-EST and ASTRAL with MLBS on 15-taxon simulated datasets**. We show average FN rate over 10 replicates. We varied the number of genes (100 and 1000) and sequence length (100bp and 1000bp). We used 50% and 75% bootstrap support thresholds for binning, and estimated the supergene trees using RAxML with fully partitioned analyses. 219

8.9 **Species tree estimation error for MP-EST and ASTRAL with MLBS on 10-taxon simulated datasets**. We show average FN rate over 20 replicates. We varied the amount of ILS and fixed the number of genes to 200 and gene sequence length to 100bp. We used 50% and 75% bootstrap support thresholds for binning, and estimated the supergene trees using RAxML with fully partitioned analyses. 222

8.10 **Trees computed on the avian biological dataset using MP-EST on MLBS gene trees.** We show results with weighted and unweighted binning (left), and unbinned analyses (right). We used 50% bootstrap support threshold for binning. Supergene trees were estimated using fully partitioned analyses. MP-EST with weighted and unweighted binning returned the same tree. The branches on the binned MP-EST tree are labeled with two support values side by side: the first is for unweighted binning and the second is for weighted binning; branches without designation have 100% support. Branches in red indicate contradictions to known subgroups. 224

9.1 **Running time of MP-EST for varying number of taxa.** We show the running time of MP-EST on the simulated mammalian datasets for varying number of taxa on the model condition with moderate level of ILS, 200 genes and 500bp sequence length. The inset subfigure shows results in seconds for 8 to 15 taxa, and the larger figure also shows results in minutes on datasets with up to 37 taxa. 247

9.2 **Average FN rates of boosted MP-EST after two and five iterations.** We show the average FN rates of the best trees, with respect to the quartet support, after two and five iterations of SSG and DACTAL-based boosting on the simulated mammalian datasets with varying sequence length (200 genes, moderate amount of ILS). 248

9.3 **Average FN rates of boosted MP-EST after two and five iterations.** We show the average FN rates of the best trees, with respect to the quartet support, after 2 and 5 iterations of SSG and DACTAL-based boosting on the simulated mammalian datasets with varying amount of ILS (200 genes, 500bp). 248

9.4 **Average FN rates of boosted MP-EST after two and five iterations.** We show the average FN rates of the best trees, with respect to the quartet support, after 2 and 5 iterations of SSG- and DACTAL-based boosting on the simulated mammalian datasets with varying numbers of gene trees (moderate amount of ILS, 500bp). 249

9.5 **Impact of how the final tree is selected (using quartet support or pseudo-likelihood) in boosted versions of MP-EST.** We show average FN rates of MP-EST (with and without boosting) on the simulated mammalian datasets with varying amount of ILS, using two different ways of selecting the final tree: quartet support (q) or pseudo-likelihood (l). We fixed the number of genes to 200 and sequence length to 500bp, while varied the amount of ILS. 2X model condition contains the lowest amount of ILS while 0.2X refers to the model conditions with the highest amount of ILS. We show the results for SSG and DACTAL-based decomposition with maximum subset size 15. 250

9.6 **Impact of how the final tree is selected (using quartet support or pseudo-likelihood) in boosted versions of MP-EST.** We show average FN rates of MP-EST (with and without boosting) on the simulated mammalian datasets with varying numbers of gene trees, using two different ways of selecting the final tree: quartet support (q) or pseudo-likelihood (l). We fixed the amount of ILS to moderate level (1X) and sequence length to 500bp, and varied the number of genes from 100 to 800. We show the results for SSG- and DACTAL-based decompositions with maximum subset size 15. 251

9.7 **Impact of how the final tree is selected (using quartet support or pseudo-likelihood) in boosted versions of MP-EST.** We show average FN rates of MP-EST (with and without boosting) on the simulated mammalian datasets with varying numbers of gene trees, using two different ways of selecting the final tree: quartet support (q) or pseudo-likelihood (l). We fixed the amount of ILS to moderate level (1X) and number of genes to 200, and varied the sequence lengths from 250bp to 1000bp. We show the results for SSG- and DACTAL-based decompositions with maximum subset size 15. 252

9.8 **Average FN rates of MP-EST (with and without boosting) for different levels of ILS.** Average FN rates of MP-EST (with and without boosting) over 20 replicates on the simulated mammalian datasets with varying amount of ILS. We also show the FN rate of concatenation. We fixed the number of genes to 200 and sequence length to 500bp, while varied the amount of ILS. 2X model condition contains the lowest amount of ILS while 0.2X refers to the model conditions with the highest amount of ILS. We show the results for short subtree graph (SSG) and DACTAL-based decompositions with maximum subset size 15. We show the FN rate of the best tree with respect to quartet support (as denoted by q in the figure legend) across five iterations. 253

9.9 **Average FN rates of MP-EST (with and without boosting) for different number of gene trees.** Average FN rates of MP-EST (with and without boosting) over 20 replicates on the simulated mammalian datasets with varying number of gene trees. We also show the FN rate of concatenation. We varied the number of genes from 100 to 800, while set the amount of ILS to 1X level and the sequence length to 500bp. We show the results for short subtree graph (SSG) and DACTAL-based decompositions with maximum subset size 15. We show the FN rate of the best tree with respect to quartet support (as denoted by q in the figure legend) across five iterations. 256

9.10 **Average FN rates of MP-EST (with and without boosting) for different sequence lengths.** Average FN rates of MP-EST (with and without boosting) over 20 replicates on the simulated mammalian datasets with different amounts of gene tree estimation error by varying the sequence lengths. We also show the FN rate of concatenation. We varied the sequence lengths from 250bp to 1000bp with 200 genes and moderate amount of ILS (1X). We show the results for short subtree graph (SSG) and DACTAL-based decompositions with maximum subset size 15. We show the FN rate of the best tree with respect to quartet support (as denoted by q in the figure legend) across five iterations. 257

9.11 **Analyses of the amniota dataset using MP-EST (with and without boosting).** We show the trees estimated by MP-EST (right) and SSG and DACTAL-boosted MP-EST (left) using the MP-EST and MRP-estimated starting tree on the nucleotide amniota dataset from [17]. The sister relationship of crocodiles and birds is considered reliable, and is recovered in the SSG-boosted MP-EST tree. However, the MP-EST analysis of this dataset places crocodiles as sister to turtles (indicated by the red edge), and is not considered reliable. 258

9.12 **Analyses of the mammalian dataset using MP-EST (with and without boosting).** MP-EST with SSG- and DACTAL-based boosting using both MP-EST and MRP-estimated starting tree produced the same tree as un-boosted MP-EST. . . . 259

9.13 **Impact of different starting trees on DACTAL-based boosting with MP-EST.** We show the average FN rates of the best trees, with respect to the quartet support, after five iterations of DACTAL-based boosting using MP-EST and using the starting trees estimated by MRP and MP-EST on the simulated mammalian datasets with varying sequence length (200 genes, moderate amount of ILS). We ran MP-EST on the subsets produced by DACTAL-based decomposition with maximum subset size 15 using different starting trees. MP-EST(MRP,dactal,15,q) refers to the results obtained by using the MRP-estimated starting tree, while MP-EST(MP-EST, dactal, 15, q) refers to the results obtained by using the starting tree estimated by MP-EST. We also show the FN rates of concatenation and the starting trees estimated by MP-EST and MRP. 263

9.14 **Impact of different starting trees on DACTAL-based boosting with MP-EST.** We show the average FN rates of the best trees, with respect to the quartet support, after 5 iterations of DACTAL-based boosting using MP-EST and using the starting trees estimated by MRP and MP-EST on the simulated mammalian datasets with varying amount of ILS (200 genes and 500bp). We ran MP-EST on the subsets produced by DACTAL-based decomposition with maximum subset size 15 using different starting trees. MP-EST(MRP,dactal,15,q) refers to the results obtained by using the MRP-estimated starting tree, while MP-EST(MP-EST,dactal,15,q) refers to the results obtained by using the starting tree estimated by MP-EST. We also show the FN rates of concatenation and the starting trees estimated by MP-EST and MRP. 264

9.15 **Impact of different starting trees on DACTAL-based boosting with MP-EST.** We show the average FN rates of the best trees, with respect to the quartet support, after 5 iterations of DACTAL-based boosting using MP-EST and using the starting trees estimated by MRP and MP-EST on the simulated mammalian datasets with varying numbers of genes (500bp, moderate amount of ILS). We ran MP-EST on the subsets produced by DACTAL-based decomposition with maximum subset size 15 using different starting trees. MP-EST(MRP,dactal,15,q) refers to the results obtained by using the MRP-estimated starting tree, while MP-EST(MP-EST, dactal, 15, q) refers to the results obtained by using the starting tree estimated by MP-EST. We also show the FN rates of concatenation and the starting trees estimated by MP-EST and MRP. 265

A.1 **Gene tree estimation error rates on 11-taxon strongILS datasets.** Average and standard error bars (over 20 replicates) of *BEAST, RAxML, and FastTree-2. Experiment 3. 299

A.2 **Gene tree estimation error rates on 11-taxon weakILS datasets.** Average and standard error bars (over 20 replicates) of *BEAST, RAxML, and FastTree-2. Experiment 3. 300

A.3 **Results for summary methods on gene trees estimated using *BEAST on 11-taxon weakILS model conditions with up to 50 genes**. Results are shown for 20 replicates. Experiment 4. 301

A.4 **Results for summary methods with input gene tree distributions estimated using *BEAST on 11-taxon weakILS model conditions with up to 50 genes**. Results are shown for 20 replicates. Every method returns the true tree on the 25- and 50-gene datasets. Experiment 4. 302

A.5 **Results for methods with input gene tree distributions estimated using *BEAST on 17-taxon model conditions**. Results are shown for 20 replicates. Experiment 4. 302

A.6 **Results of binning experiment on 17-taxon datasets with 32 genes.** We show the performance (average and standard error bars) of methods other than BUCKy on unbinned genes and *BEAST. Each bin contains 4 genes; n=100 for all datapoints. Experiment 5. 306

A.7 **Results of the binning experiment on 11-taxon 25-gene weakILS datasets.** Each bin contains 5 genes. Average and standard error bars shown; n=100 for all datapoints. CA-ML returns the true tree on these data Experiment 5. 306

A.8 **Results of the binning experiment evaluating all methods on 20 replicates of the 11-taxon 25-gene weakILS datasets.** Results are shown (average and standard error bars) for bins with 5 genes each. CA-ML, *BEAST (binned and unbinned), BUCKy-con (binned), BUCKy-pop (binned), and Phylonet-MDC (binned and unbinned) all return the true tree on these data. 312

A.9 **Results of binning experiment of 17-taxon datasets with 32 genes.** Average and standard error bars shown for all methods. Each bin has 4 genes; n=20 for all datapoints. No changes are statistically significant ($p = 0.053$ for MRP, $p = 0.082$ for GC, and $p > 0.2$ for all other methods). Experiment 5. 313

A.10 **Results of the binning experiment on 11-taxon 25-gene strongILS datasets.** Each bin contains 5 genes. Average and standard error bars shown; n=20 for all datapoints. Experiment 5. 313

B.1 **Simulation parameters and commands for 15-taxon dataset**. Gene trees were simulated using MCcoal, with control files given here (a) and the command provided (b). The control files define the species tree, which is in the caterpillar form. Running MCcoal simulated 10,000 gene trees, which we divided into 10 replicates of 1000 genes or 100 genes. For each true gene tree, we then simulated alignments using bppseqgen [34], using the command given in (c). Here, the file "opts" is the same file we used in [90] and defines parameters given in (d). 328

B.2 **Simulation parameters and commands for 10-taxon dataset**. 329

B.3 **Effect of binning on the branch lengths (in coalescent units) estimated by MP-EST using MLBS on the avian simulated datasets with varying numbers of gene trees**. We show the species tree branch length error (the ratio of estimated branch length to true branch length for branches of the true tree that appear in the estimated tree; 1 indicates correct estimation). We varied the number of genes from 50 to 2000, and fixed the sequence length to 500bp with default amount of ILS (1X level). We used 50% bootstrap support threshold for binning. Supergene trees were estimated using unpartitioned analyses. 332

B.4 **Cumulative distribution of the bootstrap support values (obtained using MLBS) of true positive (TP) and false positive (FP) edges estimated by MP-EST on avian datasets**. We varied the numbers of genes, and fixed the sequence length to 500bp (UCE-like) with default amount of ILS (1X level). We used 50% bootstrap support threshold for binning. Supergene trees were estimated using unpartitioned analyses. To produce the graph, we order the branches in the estimated species tree by their quality, so that the true positives with high support come first, followed by lower support true positives, then by false positives with low support, and finally by false positives with high support. The false positive branches with support above 75% are the most troublesome, and that fraction are indicated in the grey area. When the curve for a method lies above the curve for another method, then the first method has better bootstrap support. 333

B.5 **Divergence of estimated gene trees triplet distributions from true gene tree distributions for simulated mammalian datasets**. (a) Varying gene sequence alignments lengths with 200 number of genes and default levels of ILS (1X); (b) varying ILS levels with fixed 200 genes and sequence length fixed to 500bp (63% BS). We used 75% bootstrap support threshold for binning. Supergene trees were estimated using unpartitioned analyses. True triplet frequencies are estimated based on true gene trees for each of the $\binom{n}{3}$ possible triplets, where n is the number of species. Similarly, triplet frequencies are calculated from estimated gene/supergene trees. For each of these $\binom{n}{3}$ triplets, we calculate the Jensen-Shannon divergence of the estimated triplet distribution from the true gene tree triplet distribution. We show the empirical cumulative distribution of these divergences. The empirical cumulative distribution shows that for a given divergence level, what percentage of the triplets are diverged from true triplet distribution at or below that level. Results are shown for 10 replicates. 334

B.6 **Species tree estimation error for MP-EST with MLBS on mammalian simulated datasets with varying amounts of ILS**. We show average FN rate over 20 replicates. We varied the amount of ILS, and fixed the number of genes to 200 and sequence length to 500bp (63% BS). We used 75% bootstrap support threshold for binning. Supergene trees were estimated using unpartitioned analyses. 335

B.7 **Effect of binning on the branch lengths (in coalescent unit) estimated by MP-EST using MLBS on the mammalian simulated datasets with varying amounts of ILS**. We show the species tree branch length error (the ratio of estimated branch length to true branch length for branches of the true tree that appear in the estimated tree; 1 indicates correct estimation). We varied the amount of ILS, and fixed the number of genes to 200 and sequence length to 500bp (63% BS). We used 75% bootstrap support threshold for binning. Supergene trees were estimated using unpartitioned analyses. 336

B.8 **Cumulative distribution of the bootstrap support values (obtained using MLBS) of true positive (TP) and false positive (FP) edges estimated by MP-EST on mammalian datasets**. We varied the numbers of genes, and gene sequence alignments length with default amount of ILS. We used 75% bootstrap support threshold for binning. Supergene trees were estimated using unpartitioned analyses. 337

B.9 **Cumulative distribution of the bootstrap support values (obtained using MLBS) of true positive (TP) and false positive (FP) edges estimated by MP-EST on mammalian datasets with varying amounts of ILS**. We varied the amount of ILS, and fixed the number of genes to 200 and sequence length to 500bp. We used 75% bootstrap support threshold for binning. Supergene trees were estimated using unpartitioned analyses. To produce the graph, we order the branches in the estimated species tree by their quality, so that the true positives with high support come first, followed by lower support true positives, then by false positives with low support, and finally by false positives with high support. When the curve for a method lies above the curve for another method, then the first method has better bootstrap support. 338

B.10 **Species trees estimated by unbinned ASTRAL using MLBS on avian biological datasets.** Branches without designation have 100% support. We used 50% bootstrap support threshold for binning. Supergene trees were estimated using fully partitioned analyses. 339

B.11 **Species trees estimated by binned (with and without weighting) ASTRAL using MLBS on avian biological datasets.** (a) Unweighted binned ASTRAL, and (b) weighted binned ASTRAL. Branches without designation have 100% support. We used 50% bootstrap support threshold for binning. Supergene trees were estimated using fully partitioned analyses. Binned ASTRAL with weighting and binned ASTRAL without weighting differ only in the placement of *Opisthocomus hoazin*. However, the branches supporting different placements of *Opisthocomus hoazin* have low support values (47% for unweighted binning and 55% for weighted binning). 340

B.12 **Species trees estimated by unbinned MP-EST using MLBS for mammalian biological datasets.** Branches without designation have 100% support. We used 75% bootstrap support threshold for binning. We estimated the supergene trees using fully partitioned analyses. 341

B.13 **Species trees estimated by binned (with and without weighting) MP-EST using MLBS for mammalian biological datasets.** Binned and unbinned ASTRAL returned identical topology. The branches on this tree are labeled with two support values side by side: the first one corresponds to unweighted binning and the next one corresponds to weighted binning. Branches without designation have 100% support. We used 75% bootstrap support threshold for binning. Supergene trees were estimated using fully partitioned analyses. 342

B.14 **Species trees estimated by unbinned and binned (with and without weighting) ASTRAL using MLBS for mammalian biological datasets.** Binned and unbinned ASTRAL returned identical topology. The branches on this tree are labeled with three support values side by side: the first one corresponds to unbinned ASTRAL, the next one corresponds to unweighted binning, and the last one is for weighted binning. Branches without designation have 100% support. We used 75% bootstrap support threshold for binning. Supergene trees were estimated using fully partitioned analyses. 343

Chapter 1

Introduction

Phylogenetic trees (evolutionary trees) provide insights into basic biology, including how life evolved, the mechanisms of evolution and how it modifies function and structure, orthology detection, disease evolution etc. [11, 13, 51, 75]. Evidence from morphological and gene sequence data suggests that all organisms on earth are genetically related, and the relationships of living things can be represented by a vast evolutionary tree – the "Tree of Life". Constructing the *Tree of Life* is one of the most ambitious goals and grand challenges of modern science [49]. Central to assembling this tree of life is the ability to efficiently analyze the vast amount of genomic data available these days due to the tremendous advancement in sequencing techniques, and computer hardware and software.

A species tree represents the evolutionary history of a group of organisms, while a gene tree shows the evolutionary pathways of a particular gene within a group of organisms. Estimations of species trees are typically based on multiple genes, in some cases from throughout the whole genome. Interestingly, different genes evolve in different ways, meaning that they do not necessarily have identical evolutionary histories. This is called *gene tree in-*

congruence/discordance, and can arise from incomplete lineage sorting, gene duplication and loss, horizontal gene transfer, hybridization etc. [84]. This disparity among the gene trees makes species tree construction complicated.

Species tree estimation from multiple genes is often performed using *concatenation* (also called "combined analysis"): alignments are estimated for each gene and concatenated into a supermatrix, which is then used to estimate the species tree. When gene trees have identical topologies, concatenation can give very accurate results; however, this approach can return incorrect trees with high confidence when gene trees differ from the species tree (and hence from each other) [24, 36, 52, 73, 74, 80]. Therefore, phylogenomic analyses where the species tree is constructed by summarizing a set of gene trees, estimated from individual gene sequence alignments, are becoming more popular [1, 15, 52, 78, 80, 93, 102, 141]. We call these methods *summary methods*.

Summarizing gene trees to get a single and coherent species tree by considering the reasons for discordance is not an easy task. Many summary methods have been developed over the last decade. Some of them have the nice theoretical guarantee that they are proven to reconstruct the true species tree with arbitrarily high probability, given a sufficiently large number of true gene trees [52, 73, 80, 93]. Unfortunately, however, we do not know the true gene histories and the number of genes is limited. Thus techniques that have nice statistical guarantee might perform poorly on biological datasets.

Apart from gene tree discordance, there are many other significant challenges in phylogenomic analyses. Incomplete gene trees and poorly estimated gene trees are two such major challenges. Incomplete gene trees, where the gene trees might not contain any individual for some species, are very common in biological datasets. Incompleteness can arise from various reasons, including poor *taxon sampling* (the gene may be available in the species' genome, but it was not sampled), *gene extinction* etc. Another major challenge in phylogenomic analyses arises from the fact that gene trees are very often poorly estimated. Individual gene sequence alignments can be short. These short sequences result in poorly estimated gene trees. We showed that species tree estimation methods are susceptible to gene tree error [8], and hence poorly estimated gene trees are one of the most challenging and important problems in phylogenomics.

Finally, large-scale genomic datasets present computational challenges in species tree construction. Highly accurate methods are typically computationally intensive and cannot be run on large datasets (in terms of the number of species and genes) [8]. Therefore, the ever increasing abundance of molecular data not only opens the opportunity to resolve challenging questions regarding evolution, but also creates the need for highly scalable methods that can analyze many genes across many species.

1.1 Our contributions

We have addressed several major challenges in estimating species trees from gene trees in the presence of gene tree discordance. The contributions described in this dissertation include:

- A mathematical model for estimating species trees by minimizing gene duplication and loss (MGD and MGDL) which enables us to design dynamic programming (DP) algorithms to solve these problems exactly. We also proposed a constrained version for which we developed an efficient DP-based polynomial time algorithm.

- An investigation on how reconciliation and species tree estimation are affected by different reasons for incompleteness (missing taxa). We propose different mathematical formulations of gene loss based on the different reasons for incompleteness.

- A mathematical model for *gene tree parsimony* under the multi-species coalescent model, and for the general case where the gene trees and alignments can be incomplete.

- A novel technique called "binning" to address the problem of gene tree estimation error.

- A meta-method based on a divide-and-conquer technique to make existing highly accurate species tree estimation techniques scalable to large-scale genomic data.

1.2 Organization of the dissertation

The rest of the dissertation is organized as follows.

Chapter 2 provides necessary background material for the problem of estimating species tree from gene trees in the presence of gene tree discordance.

In Chapter 3, we describe our work on estimating species trees under gene duplication and loss model when gene trees are rooted and fully resolved. Local search heuristics for two NP-hard optimization problems, minimize gene duplications (MGD) and minimize gene duplications and losses (MGDL), are popular techniques for estimating species trees in the presence of gene duplication and loss [15, 150]. We present a novel alternative approach (rather than local search heuristics) to solving MGD and MGDL from rooted gene trees. We show that MGD can be formulated as a max-weight clique problem, and the MGDL problem can be formulated as a min-weight clique problem on an associated graph called the "compatibility graph". We propose efficient dynamic programming algorithms to find these optimal cliques that run in polynomial time in the number of vertices of the graph. We also show that the constrained version of the problem, where the "subtree-bipartitions" (a concept we introduce) for the species tree are required to be drawn from a set $\mathcal{X}$, can be computed in time that is polynomial in $|\mathcal{X}|$. With sufficiently large numbers of gene trees without any missing taxa, it is likely that the subtree-bipartitions in the true species tree are present in at least one input gene tree, as it was observed in [141] that clusters (set of leaves present in a subtree) induced by the true species tree are found in the clusters induced by the gene trees. In this case, the globally optimal species tree can be obtained by finding the exact solution for the constrained version, where $\mathcal{X}$ is the set of subtree-bipartitions in the gene trees.

In Chapter 4, we extend our algorithms for MGD and $MGDL$ (described in Chapter 3) so that they can handle unrooted gene trees. In phylogenetic analyses of biological data sets, estimated gene trees are often unrooted, as rooting requires the assumption of "strict molecular clock", or finding a suitable "outgroup". Therefore, in the absence of a molecular clock or appropriate outgroup (or for missing data from an outgroup), rooting trees can be very difficult [9, 56]. We formulate the MGD and MGDL problem for unrooted gene trees, and provide exact algorithms and heuristics for inferring species trees for these cases.

In Chapter 5, we provide different formulations of gene tree parsimony (GTP) for incomplete gene trees based on two different reasons for incompleteness: 1) taxon sampling and 2) true biological loss. We show that the "standard" calculation for losses in GTP can be incorrect when incompleteness is due to true biological loss. We present exact and heuristic algorithms to solve GTP when incompleteness results from true biological loss.

In Chapter 6, we consider the problem of estimating species trees from estimated gene trees when the true gene trees can differ from the true species trees due to incomplete lineage sorting (ILS), and for the general case where the gene trees and alignments can be incomplete. We show how to complete an incomplete gene tree (i.e., adding the missing taxa back into the gene tree) in an optimal way under the multi-species coalescent model. We formalize optimization problems and present new theoretical results in the context of species tree estimation from incomplete gene trees in the presence of gene tree discordance due to ILS. We also report on an extensive simulation study that the existing methods perform poorly when gene trees are incomplete.

In Chapter 7, we address the challenge of constructing species trees from poorly estimated gene trees. First, we perform an extensive experimental study to evaluate the performance of a wide range of species tree estimation methods (e.g., combined analyses, *BEAST [52], BUCKy [1], MP-EST [80], greedy consensus etc.) in the presence of ILS. We show that summary methods have poor accuracy when the individual gene sequence alignments have low phylogenetic signal (i.e., short sequence alignments). We present a novel approach called "naive binning" to overcome the vulnerability of the species tree methods to poorly estimated gene trees.

In Chapter 8, we present a binning technique called "weighted statistical binning", which is an improvement over statistical binning [90]. The observation that summary methods are affected by poorly estimated gene trees (presented in Chapter 7) motivated the development of naive binning [8] and subsequently statistical binning [90]. Although statistical binning is shown to have good empirical performance [90], the phylogenomic pipeline with statistical binning is not statistically consistent [5]. In this chapter we present an improved version of the binning technique called "weighted statistical binning" that enables highly accurate genome-scale species tree estimation, and is also statistically consistent under the multi-species coalescent (MSC) model.

In Chapter 9, we address the challenge of handling large-scale genomic data in phylogenomic analyses. We present divide-and-conquer based techniques that improve the scalability of MP-EST so that it can run efficiently on large datasets. This technique also improves the accuracy of species trees estimated by MP-EST, as our study shows on a collection of simulated and biological datasets.

Finally we conclude in Chapter 10 with discussion of our work and future research directions.

Chapter 2

Preliminaries

In this chapter, we introduce the basic definitions and concepts that we will use throughout this dissertation. We first introduce phylogeny and discuss different properties of a phylogenetic tree. We then discuss the concepts of gene tree and species tree, gene tree discordance, and species tree reconstruction from the evolutionary histories of the genes present in the whole genome. These are the cornerstones of phylogenomic analysis – the main focus of this dissertation. Next, we discuss the traditional pipelines for phylogenomic analysis, and discuss the measures of accuracy to evaluate species tree reconstruction methods. Terminologies that are not included in this section will be introduced as they are needed.

2.1 Phylogenies

A phylogeny is a representation of the evolutionary relationships of a set of entities (species, genes, languages, etc.). Phylogenetic entities are commonly known as *taxa*. The simplest and the most useful representation of such evolutionary history is a "tree", which we call *phylogenetic tree*. A *tree* T is a connected acyclic graph with a set of vertices V and a set of edges E. A leaf in a phylogenetic tree represents a *taxon* that typically exists in the present day. The internal nodes represent the hypothetical ancestral taxa from which the descendant taxa evolved. The internal nodes typically represent extinct species that existed in the past, but do not exist anymore. An edge $e = (u, v) \in E$ represents an evolutionary relationship between the two taxa at the vertices u and v. We denote the set of vertices of a tree T by $V(T)$, the set of internal nodes by $V_{int}(T)$, the set of edges by $E(T)$, and the set of taxa that appear at the leaves by $L(T)$.

Figure 2.1 shows an example of a phylogenetic tree that illustrates the evolutionary history of humans, chimpanzees, gorillas and orangutans. It shows that humans are more closely related to chimpanzees (they share a common ancestor) than they are to gorillas and orangutans.

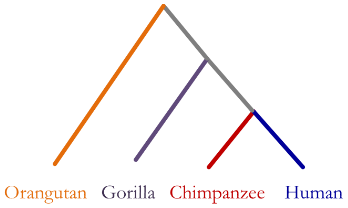


Figure 2.1: **Phylogenetic tree**. A phylogenetic tree relating four species: humans, chimpanzees, gorillas and orangutans.

The length of the edges (branches) in an evolutionary tree is known as *branch length*. Branch length is a non-negative real number that represent various quantities measured on a branch. Most often, a branch length represent the amount of evolutionary change or the amount of time between two nodes. When trees are not provided with branch lengths, we generally refer to them as *topologies*.

A phylogenetic tree $T = (V, E)$ can be rooted by designating a single vertex $r \in V$ as the *root* of the tree. Although true evolutionary histories are often best represented by a rooted tree, locating the root of the tree is usually hard to achieve. Accurately rooting a phylogenetic tree is a complex problem requiring specific knowledge of the set of taxa being studied or the assumption of a "molecular clock". Phylogenetic trees are often being rooted using the technique of using an *outgroup*, which is a taxon known to have branched off before all other taxa under consideration. Alternatively, one can root a tree based on the estimated time between speciation only if the molecular data used to reconstruct a phylogeny are assumed to have evolved at a constant rate over time, an assumption that is often violated in real datasets. Figure 2.2 shows examples of rooted and unrooted trees. We denote the root of a tree T by $root(T)$.

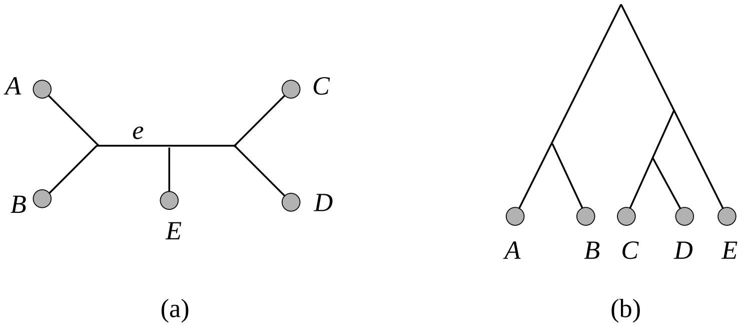


Figure 2.2: **Unrooted and rooted trees**. (a) An unrooted tree, and (b) the rooted tree resulting from rooting the tree shown in (a) on the edge e.

Phylogenetic trees can be binary or non-binary. A tree is called *binary* (also known as *fully-resolved*) if all internal nodes have degree at most three. Otherwise, the tree is *non-binary*, and has at least one node with degree greater than three, also known as a *polytomy*. Figure 2.3 shows examples of binary and non-binary trees.

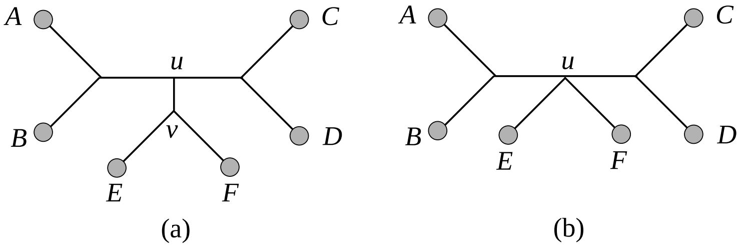


Figure 2.3: (a) A binary tree representing the evolutionary history of 6 species: $\{A, B, C, D, E, F\}$, and (b) a non-binary tree on the same set of taxa, where node u is a polytomy.

Each vertex in a rooted tree defines a group of taxa that are more closely related to each other than they are to any other taxon in the tree; such a group is called a *clade*. Formally, a clade in a phylogenetic tree T is a rooted subtree of T, which can be identified by a node v in T rooting the clade (represented by $clade_T(v)$). The set of leaves of a clade $clade_T(v)$ is called a *cluster*. We denote the cluster at v by $c_T(v)$; however, when the tree T is understood, we may also write $c(v)$. The set of all clades in a tree T is denoted by $C(T)$.

The analogous relationships in unrooted trees are *bipartitions* of the taxon set, and are defined by edges rather than vertices. Every edge e in a phylogenetic tree T defines a bipartition π_e. Deleting the edge e from T creates two subtrees T_1 and T_2, resulting into a bipartition of leaves $L(T_1)|L(T_2)$. The bipartitions corresponding to the edges incident on the leaves are called *trivial* bipartitions since they do not provide any information about the topology of the tree; whereas bipartitions corresponding to internal edges are called *non-trivial* bipartitions.

Given two trees T and T' on the same leaf set, we call T a *refinement* of T' if T' can be obtained from T by contracting some edges in T. Therefore, T refines T' if and only if $C(T') \subseteq C(T)$. Alternatively, T' is called the *contraction* of T. Figure 2.4 demonstrates the concept of refinement and contraction.

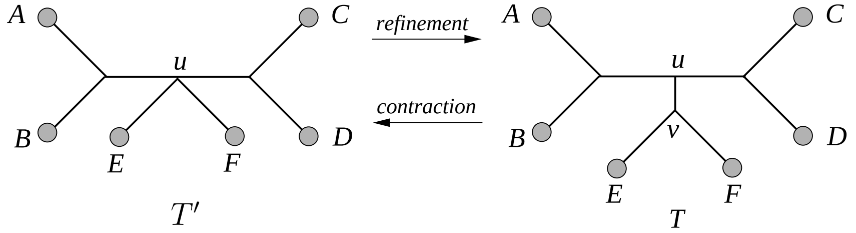


Figure 2.4: **Refinement and contraction.** T is a refinement of T'; alternatively, T' is a contraction of T. Here T' can be obtained from T by contracting the edge incident on u and v.

We now define the *phylogeny problem* as follows.

Problem	**Phylogeny**
INPUT	A set S of n taxa.
OUTPUT	A phylogenetic tree T with n leaves bijectively leaf-labeled by the taxa in S.

2.2 Gene tree and species tree

Most often, the goal of a phylogenetic reconstruction is to infer an evolutionary tree depicting the history of speciation events that lead to a currently extant set of taxa. A *species tree* can be defined as the pattern of branching of species lineages via the process of speciation. A *gene tree* represents the evolution of a particular "gene" (we interpret gene as a particular part of the whole genome) within a group of species. When species are split by speciation, the gene copies within species are also split into separate lineages of descent. Thus, gene trees are contained within species trees [84]. However, due to various biological processes, different genes (i.e., different parts of the whole genome) may have discordant evolutionary histories. Figure 2.5 shows an example of discordance between a species tree and a gene tree. Here, species C and species B are "sister" species in the species history, whereas C is closer to D than B in the gene history.

We now briefly describe various biological reasons for gene tree discordance.

Gene duplication and loss Gene duplication is the process of generating multiple gene lineages in coexisting in a species lineage [106]. A gene duplication event causes a second "locus", and these duplicated loci evolve independent of each other – resulting in incongruence between gene tree and the containing species tree [43]. Moreover, some of the gene lineages could go extinct if it decayed into a "pseudo-gene", or if it evolved a new function and

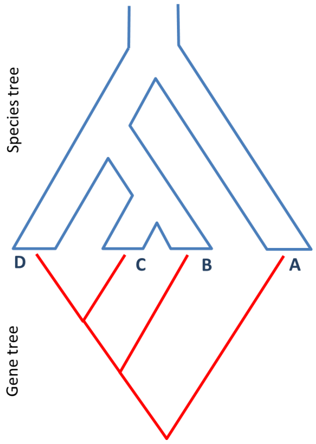


Figure 2.5: **Gene tree-species tree incongruence**. A species tree (given in blue) and a gene tree (given in red) on the same set $\{A, B, C, D\}$ of taxa with different topologies.

diverged [84]. This phenomenon is knows as *gene loss* (also known as *gene extinction*) which may result in gene tree discordance. Figure 2.6 shows how gene duplication and loss can cause gene tree discordance. Alternatively, this figure shows how to explain the discordance between a gene tree and a species tree using gene duplication and loss events. Such embedding of a gene tree inside a species tree is called "reconciliation" (see Section 2.2.1). Two gene copies are called *orthologous* if their most recent common ancestor is a speciation event, whereas they are called *paralogous* if the most recent common ancestor is traced back to a duplication event.

Incomplete lineage sorting Incomplete lineage sorting (ILS), also known as deep coalescence, is best understood under the coalescent model [27, 29, 54, 98, 99, 121, 139, 140]. The *coalescent model* describes the evolutionary process

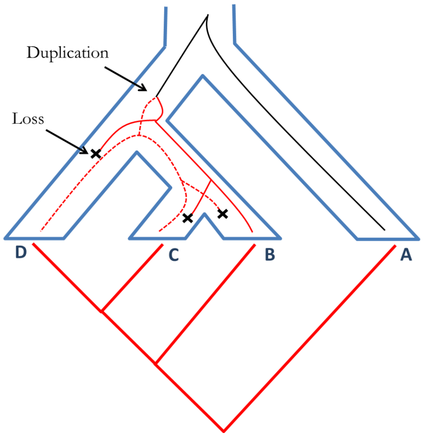


Figure 2.6: **Reconciliation under gene duplication and loss model**. We show a reconciliation of the discordant gene tree-species tree pair shown in Fig. 2.5 using gene duplication and loss. We embed the reconciled tree inside the species boundaries. Duplicated gene copes (dashed and solid red lines) evolve independently and can go extinct. This reconciliation requires one duplication and three gene losses.

as if it operates backwards in time, and connects gene lineages to a common ancestor through a process of "coalescence" of lineage pairs. This model treats a species as a population of individuals, having a pair of alleles for each gene. The coalescent process traces the present day variants of a gene (known as *alleles*) back in time across successive generations by following the ancestral alleles in the previous generation from which this given alleles evolved. Eventually we reach a point where two alleles coalesce (i.e., they find a common ancestor). The *multi-species coalescent* (MSC) model is the extension of this general coalescent framework where multiple randomly mating populations corresponding to multiple species are present. Thus, multi-species coalescent represents a gene tree inside a species tree.

Incomplete lineage sorting or deep coalescence refers to the case in which two lineages fail to coalesce at their speciation point. Under the coalescent model, deep coalescence can be a source of discordance, because the common ancestry of gene copies at a single locus can extend deeper than speciation events. Figure 2.7 shows an example of discordance due to ILS. Going back in time, the gene copies within species B and C first meet at their corresponding speciation point (i.e, the most recent common ancestor of species B and C), but fail to coalesce at the speciation point. Both of these copies go further back in time, and hence we have two gene lineages (dashed and solid black lines in Fig. 2.7) on deeper ancestral branch. Then the gene from C first coalesces with the gene from species D, and subsequently with the gene from B. Note that this deep coalescence results in one *extra lineage* (two gene lineages instead of one on a branch of the tree) for this particular example. In Fig. 2.8, we show the impact of population size and branch length on incomplete lineage sorting. As we have already mentioned, the coalescent process traces the ancestry of alleles (shown by small filled circles in Fig. 2.8) back in time (upward in a branch) across successive generations. Since the mating process is random, all the alleles in generation t are equally likely to be the ancestor of an allele in generation $t + 1$. Therefore, deep coalescence is more likely to happen when the population size is large and the branch length (in terms of the number of generations) is short [84, 110].

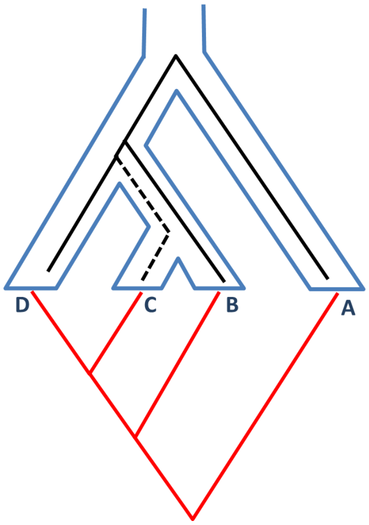


Figure 2.7: **Reconciliation under incomplete lineage sorting**. We show a reconciliation of the discordant gene tree-species tree pair shown in Fig. 2.5 using incomplete lineage sorting. We embed the reconciled tree inside the species boundaries. Going back in time, the gene copies within species B and C first meet at their corresponding speciation point (i.e, the most recent common ancestor of species B and C), but fail to coalesce at the speciation point. Both of these copies go further back in time, and hence we have two gene lineages (dashed and solid black lines) on deeper ancestral branch. Therefore we have one *extra lineage* on the ancestral branch. The gene from C first coalesces with the gene from species D, and subsequently with the gene from B.

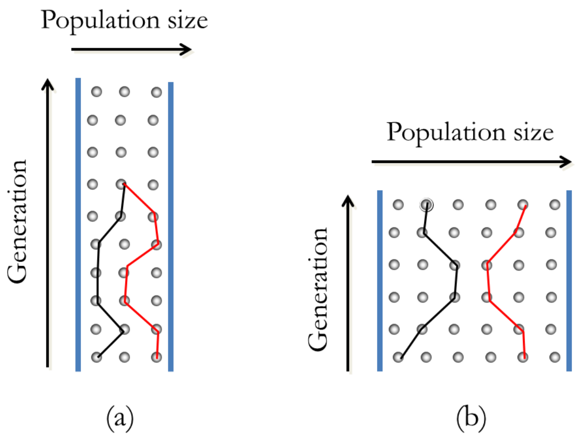


Figure 2.8: **Effect of effective population size and branch length on incomplete lineage sorting under the multi-species coalescent model**. (a) Long branch (larger number of generations) and small population size increase the chance that two gene lineages will coalesce with each other; (b) Larger population size and smaller branch increase the possibility that gene lineages will fail to coalesce before reaching the deeper speciation event.

Horizontal gene transfer In many organisms (bacteria for an example), a significant level of genetic exchange occurs between lineages, and lineages can combine to produce new independent lineages. *Horizontal gene transfer* (HGT), also known as *lateral gene transfer*, is the process that causes the genes to be transferred across species. These exchanges and combinations result in discordance between gene trees and species trees, and the accurate representation of evolutionary history requires a phylogenetic network instead of a tree. Figure 2.9 shows how HGT transforms a tree into a network, which results in gene tree-species tree discordance.

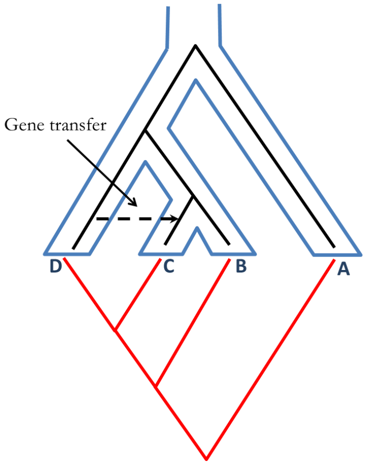


Figure 2.9: **Reconciliation under horizontal gene transfer**. We show a reconciliation of the discordant gene tree-species tree pair shown in Fig. 2.5 using horizontal gene transfer. Here, the gene lineage from species D moves horizontally across species boundaries and enters into the species boundary of C. This "foreign" gene lineage is maintained and spread into the receiving species population. If the receiving lineage (C) goes extinct or is not sampled, then there will be discordance between the species tree and the gene tree.

2.2.1 Gene tree reconciliation

With the abundance of molecular data available, species tree reconstruction from genes sampled from throughout the whole genome has drawn significant attention from systematists. However, species tree reconstruction from a set of gene trees, in the presence of different biological processes causing gene tree discordance, is not an easy task. One of the most important and difficult challenges in reconstructing the *Tree of Life* is to reliably address gene tree-species tree incongruence in the presence of such confounding evolutionary events. Central to addressing this challenge is to develop mathematical models to explain (or reconcile) gene tree-species tree incongruence assuming specific reasons for discordance. For example, how can we explain the difference between a gene tree and a species tree assuming that the discordance is due to gene duplication and loss? This requires us to embed/map the gene tree inside the species tree using a number of gene duplication and loss events. This concept of reconciling gene trees inside a species tree dates from Goodman *et al.*'s [43] attempt to find the most parsimonious reconciliation of a gene tree within a species tree under duplication and loss events. Later on this concept of reconciliation was explored quite extensively [44, 47, 95, 103–105, 158].

Fundamental to this reconciliation problem is to find an *optimal embedding* (i.e., most parsimonious embedding in terms of the number of confounding evolutionary events – duplication/loss, ILS etc.) of the gene tree inside a species tree. We now describe how to embed a gene tree inside a species tree.

Given a gene tree gt and a species tree ST, where $L(gt) \subseteq L(ST)$, we define $\mathcal{M} : V(gt) \rightarrow V(ST)$ by $\mathcal{M}(v) = MRCA_{ST}(c_{gt}(v)))$. In other words, $\mathcal{M}$ associates each node u of gt to the MRCA (most recent common ancestor) in ST of the cluster below u. Let gt and ST be rooted binary gene and species trees, respectively, on the same set $\mathcal{X}$ of taxa. Then the *optimal embedding* (also known as *optimal reconciliation*) of gt to ST under each of the three criteria (duplication, duplication-loss, or deep coalescence) is obtained using the $\mathcal{M}$ mapping [84, 141, 158].

Figure 2.10 demonstrates the $\mathcal{M}$ mapping between the nodes in a gene tree to the nodes in a species tree. Figures 2.11 and 2.12 show examples of the optimal and non-optimal reconciliation of a rooted, binary gene tree gt with a rooted, binary species tree ST under the duplication-loss and multi-species coalescent models, respectively.

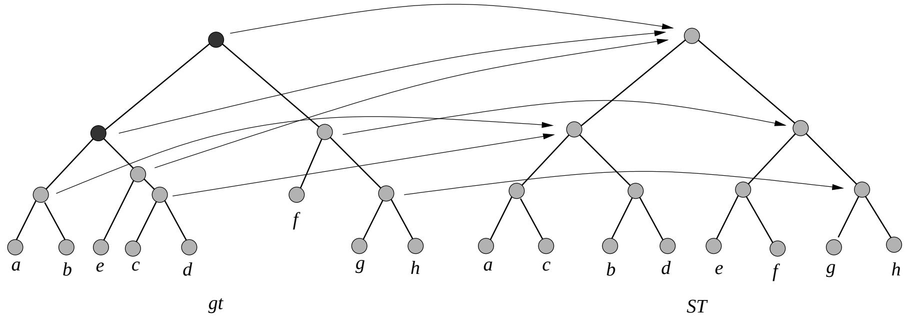


Figure 2.10: **Illustration of the $\mathcal{M}$ mapping of the nodes in a gene tree gt with respect to a species tree ST.**

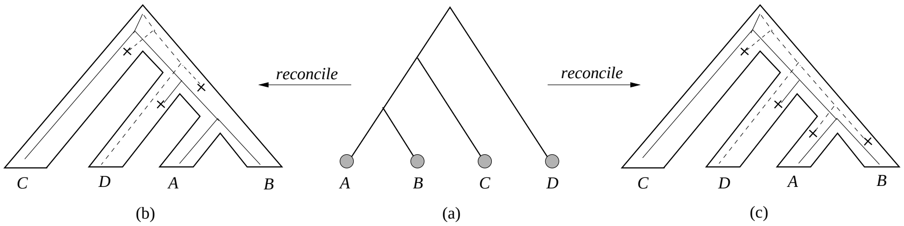


Figure 2.11: **Optimal and non-optimal reconciliations under the gene duplication and loss model**. (a) A rooted, binary gene tree *gt*, (b) an optimal reconciliation of *gt* with a rooted, binary species tree that yields 1 duplication and 3 losses, and (c) a non-optimal reconciliation of *gt* using 1 duplication and 4 losses.

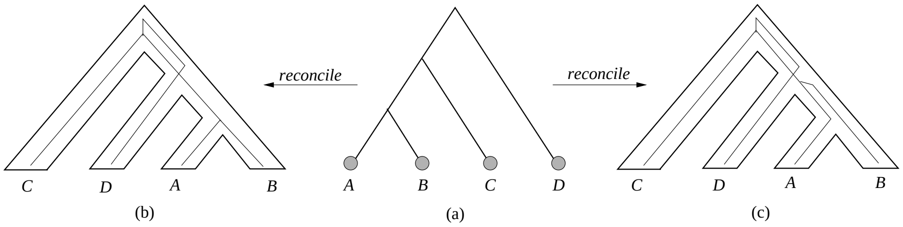


Figure 2.12: **Optimal and non-optimal reconciliations under the deep coalescence model**. (a) A rooted, binary gene tree *gt*, (b) an optimal reconciliation of *gt* with a rooted, binary species tree that yields 1 extra lineage, and (c) a non-optimal reconciliation of *gt* using 2 extra lineages.

2.2.2 Duplication, loss and deep coalescent events: mathematical formulation

Because of the relevance to this dissertation, we now provide mathematical formulations for gene duplication and loss, and extra lineage (resulting from deep coalescence).

2.2.2.1 Duplication and loss

For a rooted gene tree gt and a rooted species tree ST, where $L(gt) \subseteq L(ST)$, an internal node v in gt is called a *duplication node* if $\mathcal{M}(v) = \mathcal{M}(v')$, for some child v' of v, and otherwise v is a *speciation node* [44, 47, 82, 158]. We denote by $Dup(gt, ST)$ the number of duplications needed to reconcile gt with ST. For a set $\mathcal{G}$ of rooted, binary gene trees, the notation $Dup(\mathcal{G}, ST)$ extends in the obvious way as follows.

$$Dup(\mathcal{G}, ST) = \sum_{gt \in \mathcal{G}} Dup(gt, ST)$$

.

We now describe how to compute the number of loss events associated with an optimal reconciliation of a gene tree within a species tree. For two nodes x and y in T, $x < y$ if y is on the path between x and the root of T. Let $\mathcal{R}_{ST}(L(gt))$ be the *restriction* of ST to gt, that is, $L(\mathcal{R}_{ST}(L(gt))) = L(ST) \cap L(gt)$. In other words, $\mathcal{R}_{ST}(L(gt))$ is the subtree of ST induced by $L(gt)$. The *homeomorphic subtree* $ST(gt)$ of ST induced by $L(gt)$ is a tree obtained from $\mathcal{R}_{ST}(L(gt))$ by suppressing all nodes of $\mathcal{R}_{ST}(L(gt))$ with in-degree and out-degree 1. Let $\mathcal{M}$ be the MRCA mapping from gt to $ST(gt)$. Then the number of loss events required to explain the discordance between gt and ST, denoted by $loss(gt, ST)$, can be calculated as follows [44, 47, 82, 158].

Using notation from [44, 158], we let $loss(gt, ST)$ be defined by

$$loss(gt, ST) = \sum_{v \in V(gt)} loss_v, \tag{2.1}$$

where $loss_v$ is the number of losses associated to the internal node v. In turn, $loss_v$ is defined by the two children a and b of v, as follows:

$$loss_v = \begin{cases} d(\mathcal{M}(a), \mathcal{M}(v)) + 1 & \text{if } \mathcal{M}(a) < \mathcal{M}(v) \ \& \\ & \mathcal{M}(v) = \mathcal{M}(b), \\ d(\mathcal{M}(a), \mathcal{M}(v)) + d(\mathcal{M}(b), \mathcal{M}(v)) & \text{if } \mathcal{M}(a) < \mathcal{M}(v) \ \& \\ & \mathcal{M}(b) < \mathcal{M}(v), \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

Here $d(s, s')$ is the number of nodes on the path in $ST(gt)$ from s to s' excluding s and s'.

For a set $\mathcal{G}$ of rooted, binary gene trees, the number of losses is given by $loss(\mathcal{G}, ST) = \sum_{gt \in \mathcal{G}} loss(gt, ST)$. The number of duplications and losses, denoted by $Duploss(\mathcal{G}, ST)$, is the sum of the number of duplication and losses, i.e., $Duploss(\mathcal{G}, ST) = Dup(\mathcal{G}, ST) + loss(\mathcal{G}, ST)$.

2.2.2.2 Extra lineages

We denote the number of *extra lineages* on an edge $e \in E(ST)$ by $XL(gt, e)$, and note that this is one less than the number of lineages on e in an *optimal reconciliation* of gt within ST under deep coalescence. We denote by $XL(gt, ST)$ the total number of extra lineages within an optimal reconciliation of gt and ST. Thus, $XL(gt, ST) = \sum_{e \in E(ST)} XL(gt, e)$.

For any cluster A in gt and a cluster B in ST, we say that A is B-maximal if (1) $A \subseteq B$, and (2) for any cluster A' in gt, if $A \subseteq A'$, then $A' \nsubseteq B$. We define $k_B(gt)$ to be the number of B-maximal clusters within gt, and in a rooted tree T with cluster G, the unique edge e that separates G from the rest of the leaves in T is called the *parent edge* of the cluster G. Then $k_B(gt)$ is equal to the number of lineages on the parent edge e of B in an optimal reconciliation of gt within ST [141, 155].

2.2.3 Gene tree parsimony

We are now ready to define *gene tree parsimony* (GTP) which was first introduced by W. P. Maddison in 1997 [84]. Let $\mathcal{C}(gt, ST)$ be the cost (i.e., the number of duplication and loss events, the number of extra lineages etc.) associated with reconciling gt within ST. Then we define gene tree parsimony as follows.

Problem **Gene Tree Parsimony**

INPUT A set $\mathcal{G} = \{gt_1, gt_2, \ldots, gt_k\}$ of gene trees, and a reason for discordance (duplication-loss or ILS etc.).

OUTPUT A species tree ST that minimizes $\sum_{gt\in\mathcal{G}} \mathcal{C}(gt, ST)$ assuming the presence of the given reason for discordance.

2.2.4 Statistical consistency

A species tree reconstruction method is called *statistically consistent* under a particular model of evolution if the probability of returning the true species tree converges to one as the amount of data increases (we usually assume both the number of sites and the number of loci increase). Let $\mathcal{G} = \{g_1, g_2, \ldots, g_k\}$ be a set of genes, and let s_i be the number of sites in g_i $(1 \leq i \leq k)$. A species tree estimation method is said to be statistically consistent if the estimated species tree, which is considered as a random variable, converges in probability to the true species tree as $k \to \infty$, $\underset{1\leq i\leq k}{s_i} \to \infty$. Statistically consistent methods are typically preferred over the methods that are not statistically consistent [24, 33, 36, 73, 74, 80]. Many statistically consistent methods have been developed in the last decade to estimate species tree from a set of gene trees in the presence of gene tree discordance. *BEAST [52], BUCKy-pop [73], MP-EST [80] and ASTRAL [93, 94] are among the leading statistically consistent methods under the multi-species coalescent model.

2.3 Phylogenomic analysis pipeline

Two of the most popular approaches for estimating species trees from a collection of gene trees are *concatenation* (also known as “combined analysis”) and *summary* methods.

Concatenation Concatenation (also known as combined analyses) is the most basic and simple pipeline for phylogenomic analysis where alignments are estimated for each gene and concatenated into a supermatrix, which is then used to estimate the species tree. Concatenation does not consider gene tree discordance as it combines all the gene alignments into a supermatrix. Implicit in this analyses is the assumption that all the genes have the same evolutionary history. However, a partitioned analysis allows the branch lengths and other model parameters except the tree topology to be estimated separately for each gene. But an unpartitioned analysis estimates a single set of model parameters, including tree topology, branch lengths etc. Recently it has been proved that concatenation using an unpartitioned maximum likelihood analysis can be statistically inconsistent under the multi-species coalescent model [26, 119]. Empirical studies also suggest that it can return incorrect trees with high confidence [24, 36, 52, 73, 74, 80]. The statistical consistency of concatenation using partitioned analyses is unknown.

Summary methods Summary methods refer to a broad class of phylogenomic methods that construct a species tree by summarizing a collection of gene trees. Gene tree parsimony methods such as estimating species trees by minimizing deep coalescence (MDC) and minimizing duplication and loss (MGDL) are examples of summary methods. Unlike concatenation, summary methods are not necessarily agnostic about the reason for discordance and can be statistically consistent. Therefore, summary methods are becoming more popular and gaining much attention from systematists [1, 15, 52, 78, 80, 93, 102, 141].

2.3.1 Existing methods

Here we provide a list of the leading methods for estimating species trees under the multi-species coalescent and gene duplication and loss models.

- ILS-based methods: Many summary methods have been developed to estimate species trees from gene trees by considering ILS as the reason for discordance. *BEAST [52], BUCKy-pop [73], GLASS [96], STEM [68], STAR [81], NJst [79], MP-EST [80] and ASTRAL [93, 94] are well known statistically consistent methods. There are some other methods (greedy consensus, minimize deep coalescence (MDC) [84], matrix representation with parsimony (MRP) [3], matrix representation with likelihood (MRL) [102] etc.), which do not have the statistical guarantee, but perform well in practice. Phylonet [143] is a software package, which has the functionality for estimating species trees by solving the MDC problem using the algorithm described in [141, 155] (we call this Phylonet-MDC).

- Gene duplication and loss based methods: iGTP [15] and duptree [150] are the leading methods for constructing species trees by minimizing gene duplications and losses.

2.4 Evaluation of species tree estimation methods

We use extensive experimental studies to evaluate various species tree estimation methods. Throughout this dissertation, we use both simulated and real biological datasets for evaluation.

2.4.1 Evaluation on simulated datasets

A typical simulation protocol for evaluating species tree estimation techniques is modelled as follows. Figure 2.13 illustrates different steps in this simulation protocol.

- **Step 1:** A simulation study begins with a *model species tree* (also known as *true species tree*). A model species tree can be generated (typically using a birth-death process), or a biologically-based species tree (a tree estimated on real biological datasets) from existing literature can be chosen as a model tree.

- **Step 2:** A set of gene trees are simulated down the model species tree under a particular model (e.g., gene duplication and loss, ILS etc.). These are known as *true gene trees.*

- **Step 3:** A set of gene sequences are simulated by evolving nucleotide sequence down the true gene trees under a particular sequence evolution model.

- **Step 4:** Gene trees are estimated from the gene sequence alignments. These are called *estimated gene trees.*

- **Step 5:** Finally, we estimate a species tree from the set of estimated gene trees using the method of our consideration, and compare this estimated species tree to the model species tree using an appropriate error metric described below.

2.4.2 Error metrics

In simulation studies, since the ground truth (which we call the model tree or true tree) is known, we compare the species trees estimated by the methods of consideration with the true tree. There are various standard ways of measuring estimation error. We now describe the error metrics that are widely used to quantify the reconstruction error.

False negative (FN) rate The false negative (FN) rate (also known as missing branch rate) is the proportion of the edges present in the true tree but not present in the estimated tree. Figure 2.14 shows an example of a true tree and an estimated tree where one true branch is not reconstructed in the estimated tree.

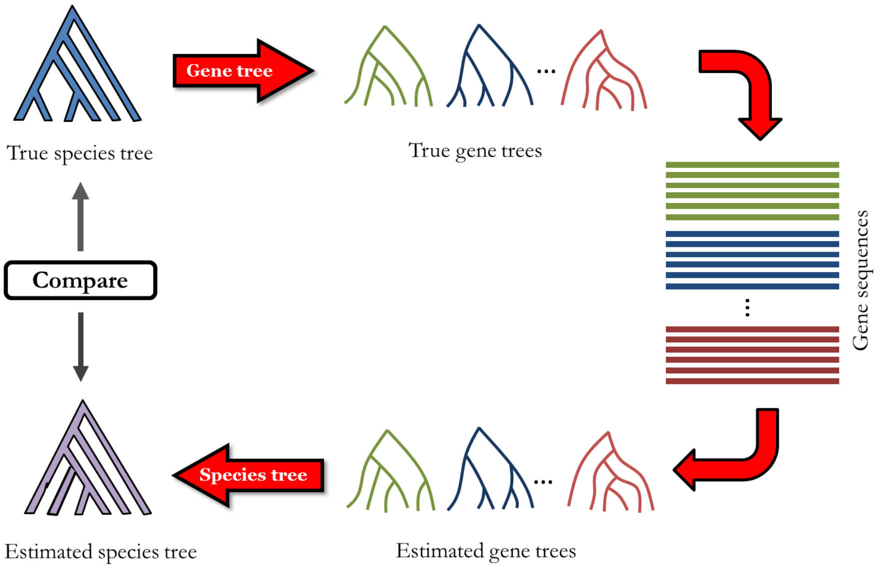


Figure 2.13: **Illustration of a simulation protocol for evaluating species tree estimation techniques.** We start with a model species tree. Next, a collection of true gene trees are evolved down this model species tree, and gene sequences are evolved down the collection of true gene trees. Next, we estimate gene trees from the gene sequence alignments. Finally, a species tree is estimated from the estimated gene trees, and compared to the true species tree.

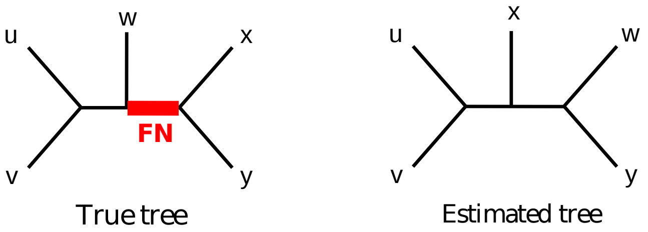


Figure 2.14: **Missing branch rate.** The branch separating $\{u, v, w\}$ and $\{x, y\}$ is not reconstructed in the estimated tree.

False positive (FP) rate The FP rate is the proportion of the edges present in the estimated tree but not in the true tree. Note that the FN rate is identical to the FP rate for binary trees. However, for non-binary trees, the FN rate and the FP rate are not necessarily identical, and the FP rate is not a good measure of accuracy in this case. For example, let T_r be a true binary species tree and T_e be an estimated tree, which is a star (a tree with one internal node). In this case, the FP rate is zero even though the estimated tree failed to reconstruct the internal edges.

Robinson-Foulds (RF) rate The Robinson-Foulds (RF) rate is the ratio of the total number of false positive and false negative edges to the total number of internal edges in the two trees. When true and estimated trees are binary, the RF rate is simply the average of the FN rate and the FP rate, and in this case the FN rate, the FP rate and the RF rate are all equal. The RF rate is the most commonly used error metric. However, because of the same reason as described for the FP rate, this metric is not appropriate when the trees are not binary.

All the trees reported in this dissertation are binary and hence the FN rate and the FP rate are identical, and both are also equal to the RF rate. Therefore, throughout this dissertation, we refer to the FN rate as the measure of topological error.

2.4.3 Evaluation on real biological datasets

In real biological datasets, the ground truth is not known; and hence we cannot use the error metrics described above. In this case, we have to rely on the existing literature and biological beliefs/evidence regarding the evolutionary history of the species of our consideration. For example, humans are believed to be more closely related to chimpanzees than they are to gorillas or orangutans; and therefore we expect a method to reconstruct this relationship (clade) using genome-scale data from humans, chimpanzees, gorillas and orangutans.

2.5 New data structures

To address the problem of estimating species trees from a collection of gene trees by minimizing gene duplication and loss, we have introduced *new* data structures that enable us to develop efficient dynamic programming (DP) based algorithms. We now describe these new data structures.

Subtree-bipartitions: Let T be a rooted binary tree and u an internal node in T. The *subtree-bipartition* of u, denoted by $\mathcal{SBP}_T(u)$, is the unordered pair $(c_T(l)|c_T(r))$, where l and r are the two children of u. Note that subtree-bipartitions are not defined for leaf nodes. The set of subtree-bipartitions of a tree T is denoted by $\mathcal{SBP}_T = \{\mathcal{SBP}_T(u) : u \in V_{int}(T)\}$.

Domination, containment, disjointness, and compatibility: Let $BP_i = (P_{i_1}|P_{i_2})$ and $BP_j = (P_{j_1}|P_{j_2})$ be two subtree-bipartitions. We say that

BP_i is *dominated* by BP_j (and conversely that BP_j *dominates* BP_i) if either of the following two conditions holds: (1) $P_{i_1} \subseteq P_{j_1}$ and $P_{i_2} \subseteq P_{j_2}$, or (2) $P_{i_1} \subseteq P_{j_2}$ and $P_{i_2} \subseteq P_{j_1}$. We say that BP_i *contains* BP_j if $P_{j_1} \cup P_{j_2} \subseteq P_{i_1}$ or $P_{j_1} \cup P_{j_2} \subseteq P_{i_2}$, and that BP_i and BP_j are *disjoint* if $[P_{i_1} \cup P_{i_2}] \cap [P_{j_1} \cup P_{j_2}] = \emptyset$. We say that two subtree bipartitions are *compatible* if one contains the other, or they are disjoint.

The compatibility graph $CG(\mathcal{G})$: Let $\mathcal{G}$ be a set of rooted binary gene trees on the set $\mathcal{X}$ of n taxa. The *compatibility graph* $CG(\mathcal{G})$ has one vertex for each possible subtree-bipartition defined on $\mathcal{X}$, and there is an edge between two vertices if and only if the associated subtree-bipartitions are compatible.

Note that if two subtree-bipartitions are compatible, then their associated clusters (produced by unioning the two parts of the bipartition) are also either disjoint or one contains the other.

Chapter 3

Inferring Optimal Species Trees under Gene Duplication and Loss

Species tree estimation from multiple markers is complicated by the fact that gene trees can differ from each other (and from the true species tree) due to several biological processes, one of which is gene duplication and loss. Local search heuristics for two NP-hard optimization problems – minimize gene duplications (MGD) and minimize gene duplications and losses (MGDL) – are popular techniques for estimating species trees in the presence of gene duplication and loss. In this chapter, we present an alternative approach to solving MGD and MGDL from rooted gene trees. First, we characterize each tree in terms of its subtree-bipartitions (a concept we introduce). Then we show that the MGD species tree is defined by a maximum weight clique in a vertex-weighted graph that can be computed from the subtree-bipartitions

Much of the material in this chapter is taken without alteration from the following paper.

- M. S. Bayzid, S. Mirarab, and T. Warnow. Inferring optimal species trees under gene duplication and loss. In *Proceedings of the of Pacific Symposium on Biocomputing (PSB)*, volume 18, pages 250–261, 2013

MSB designed the study, developed the clique-based formulations and dynamic programming algorithms for MGD and MGDL problems, and proved all the theoretical results. MSB and TW wrote the paper with comments from SM. SM implemented the algorithms as a software tool.

of the input gene trees, and the MGDL species tree is defined by a minimum weight clique in a similarly constructed graph. We also show that these optimal cliques can be found in polynomial time in the number of vertices of the graph using a dynamic programming algorithm, because of the special structure of the graphs. Finally, we show that a constrained version of these problems, where the subtree-bipartitions of the species tree are drawn from the subtree-bipartitions of the input gene trees, can be solved in time that is polynomial in the number of gene trees and taxa.

3.1 Introduction

The estimation of species trees typically proceeds by concatenating multiple sequence alignments together for many genes and then estimating a tree on the resultant super-matrix. These combined analyses require that all sequences be orthologous (hence each taxon should appear in each gene sequence alignment at most once), and assume that the true trees for the different genes are topologically identical. These two conditions can easily fail to hold when gene duplication and loss occurs, even when valiant efforts are made to estimate orthology. Thus, the estimation of species trees from gene trees that can differ due to gene duplication and loss [37, 43, 84, 107, 157], especially when these gene trees contain more than a single copy of each taxon, requires more care.

Two of the most popular approaches for species tree estimation in the presence of gene duplication and loss are methods, such as iGTP [15] and

DupTree [150], that employ local search techniques to "solve" the NP-hard optimization problems MGD (Minimize Gene Duplication) and MGDL (Minimize Gene Duplication and Loss). For example, analyses based upon MGD and MGDL have been used in estimating species trees for snakes [127], vertebrates [108, 109], *Drosophia* [20], and plants [125]. These local search strategies are effective for relatively small numbers of taxa, but their utility for very large numbers of taxa has not been explored.

In addition to local search techniques, exact solutions [14, 31] and fixed-parameter tractable algorithms [50, 132] have been proposed for addressing MGD and MGDL. Doyon and Chauve [31] have described an exact solution using branch-and-bound and a technique to constrain the branch-and-bound procedure using prior knowledge. A constrained version of this approach was applied to a dataset of 29 taxa; this analysis produced the same solution as iGTP, but took orders of magnitude more computational time. Chang *et al.* [14] also proposed an exact solution for MGD based on an ILP formulation, and were able to apply their approach to datasets of up to 14 taxa.

In this chapter we present a new approach for MGD and MGDL that does not use local search techniques or branch-and-bound techniques, but instead uses dynamic programming to produce an optimal solution within a user-specified subspace of the set of candidate species trees. Thus, by letting that subspace be all possible species trees we obtain a globally optimal solution for MGD or MGDL, while constraining the set allows us to obtain good (even if not globally optimal) solutions in polynomial time. Our dynamic programming approach is similar to the WIDTH K MGD and MGDL techniques introduced by Hallet and Lagergren [50]. Hallet and Lagergren propose creating the subspace of candidate species trees using a parameter they introduce called WIDTH, and show that with a bounded WIDTH, the dynamic programming approach can find the optimum species tree in the subspace defined by the given WIDTH. While our dynamic programming approach is similar to that of Hallet and Lagergren, our clique-based formulation of the problem is new, and some of our theoretical results are not explicitly stated in [50]. In addition, we have implemented our version of the dynamic programming algorithm in a publicly available software tool.

The algorithmic technique we present is also related to the approach used in Than and Nakhleh [141] (see also Yu, Warnow, and Nakhleh [155]) for the MDC (Minimize Deep Coalescence) problem [84], an optimization problem for species tree estimation in the presence of incomplete lineage sorting. In these papers, the optimal solution for MDC is characterized graph-theoretically, as follows. First, every binary rooted tree on n taxa can be represented by its set of clusters, where a cluster is the set of taxa that appear below a node in the tree. Furthermore, two clusters are said to be *compatible* if and only if they can co-exist in a tree (equivalently, two clusters are compatible if and only if they are pairwise disjoint or one contains the other). To solve MDC, each possible cluster is represented by a node in a graph, and edges exist between pairs of nodes whose clusters are compatible. It is known that whenever a set of clusters is given that are all pairwise compatible, then a rooted tree exists with precisely that set of clusters. Thus, a set of $n-1$ pairwise compatible clusters, where n is the number of species, defines a binary rooted species tree for that set of clusters.

Than and Nakhleh [141] showed that it is possible to weight the nodes in the graph so that the total weight of any $(n-1)$-clique is the MDC score for the species tree defined by that clique, so that solving the MDC problem is equivalent to finding a minimum weight $n-1$ clique.

This problem formulation seems to be particularly expensive, since MaxClique is NP-hard and the graph has an exponential number of vertices, but Than and Nakhleh also showed that finding the minimum weight clique of size $n-1$ can be obtained in time that is polynomial in the number of nodes in the graph, using dynamic programming (DP). They also presented a "heuristic" version that only uses clusters that appear in the input gene trees, and so runs in polynomial time. This heuristic version produces highly accurate species trees [61, 141, 155], suggesting that restricting the search space to clusters in the input trees is an effective strategy for MDC.

The approach we present here for optimizing MGD or MGDL builds on these ideas. We also build a graph, but the nodes of our graph correspond to subtree-bipartitions, a generalization of clusters that we defined in Chapter 2 (Sec. 2.5). We show how to define weights on vertices in the graph so that the optimal solution to MGD is obtained by finding a minimum weight clique of size $n-1$, and we show how to find that clique using dynamic programming. This technique directly allows us to solve the constrained MGD problem, in which we constrain the species tree solution to have its subtree-bipartitions from a user-provided set; as with MDC, a DP algorithm solves this in polynomial time. We then show how to extend this to the MGDL problem, using the same graph but with different weights on the edges.

3.2 Basics

3.2.1 Prior terminology and theory

We begin by defining the MGD, MGDL, and MDC problems. The input to each problem is the same: a set $\mathcal{G} = \{t_1, t_2, \ldots, t_k\}$ of rooted binary gene trees, with leaves drawn from the set $\mathcal{X}$ of n taxa, and we allow the gene trees to have multiple copies of the taxa, and even to miss some taxa. The output of each problem is a species tree T on $\mathcal{X}$ minimizing $\sum_i d(t_i, T)$, where $d(t_i, T)$ is defined differently for each problem.

The original definitions for these problems assumed that the gene tree t_i had at least one copy of each taxon, and so these definitions need to be modified in order to handle incomplete gene trees, which have no copies of some taxon.

Handling incomplete gene trees: Most of the literature has handled the case of incomplete gene trees t_i as follows. Let T' be the tree obtained by restricting T to the leaf set of t_i and then suppressing all non-root nodes of degree two (i.e., T' is the homeomorphic subtree of T defined on the leafset of t_i). Then, T' is used instead of T when computing the MDC, MGD, or MGDL score. We call this the *restriction*-based approach, and hence define the restriction-based optimization problems MGD_r, $MGDL_r$, and MDC_r.

Another approach is as follows. Given incomplete gene tree t_i and taxon set $\mathcal{X}$, we say that gene tree t'_i is a *completion* of t_i if t'_i is formed by adding one copy of each missing taxon into t_i. Given t_i, the completion t'_i is sought that gives the best score with respect to the species tree T. We call this the *completion*-based approach, and thus define the completion-based optimization problems $MGD_c, MGDL_c$, and MDC_c.

These two approaches are identical when gene trees are complete, but produce different optimization problems and have different theory for incomplete gene trees. For example, Bayzid and Warnow [7] showed that the MDC_r and MDC_c scores can be different, and that Phylonet-MDC produces MDC_c scores but that iGTP produces MGD_r scores. Here we address the problem of solving MGD_r and $MGDL_r$.

Optimal embeddings for $MGD_r, MDGL_r$, and MDC_r.

The optimal embedding for each of the three criteria we discuss (MDC_r, MGD_r, and $MGDL_r$) is obtained using $\mathcal{M}$-mapping (described in Chapter 2), even when the gene tree gt is incomplete (lacks some taxon) or contains more than one copy of some taxon [84, 141, 157, 158]. Therefore, since the same reconciliation of a gene tree into a species tree optimizes all three criteria, we may refer to an optimal reconciliation without specifying the criterion. Also, for any given mapping, the calculation of the three scores can be performed in polynomial time. Therefore, given a set of rooted gene trees and a rooted species tree, we can calculate the $MGD_r, MGDL_r$, and MDC_r scores of the species tree in polynomial time.

Given a rooted, binary gene tree gt and a rooted, binary species tree ST such that $L(gt) \subseteq L(ST)$, the number of duplications ($Dup(gt, ST)$), and the number of duplications and losses ($Duploss(gt, ST)$) needed to reconcile gt with ST under the $\mathcal{M}$ mapping can be calculated using the restriction-based analyses described in Chapter 2 (see Sec. 2.2.2.1).

3.3 New theorems

All results here are for rooted binary gene trees and species trees. We assume that the species tree has exactly one copy of each taxon in $\mathcal{X}$, but that the gene trees can have any number (including zero) of each taxon in $\mathcal{X}$. The total number of taxa in $\mathcal{X}$ is n.

Observation 3.3.1. *A set $\mathcal{C}$ of $n-1$ subtree bipartitions is compatible (meaning all pairs of clusters are compatible) if and only if there exists a binary rooted tree whose set of subtree bipartitions is exactly $\mathcal{C}$.*

Proof. Follows from the definition of subtree bipartition compatibility, and the fact that a set of $n-1$ compatible clusters on n taxa defines a binary tree with that set of clusters. □

We use the fact that $(n-1)$-cliques in the compatibility graph define rooted binary trees to develop solutions for the MGD_r and $MGDL_r$ problems.

To do this, we define weights on nodes in the compatibility graph to characterize the solutions to these problems as $(n-1)$-cliques with maximum weight (for MGD_r) or minimum weight (for $MGDL_r$). As was done by Than and Nakhleh [141] for the MDC_c problem, we will present a dynamic programming algorithm that finds an optimal $(n-1)$-clique in time that is polynomial in the number of nodes in the compatibility graph.

Lemma 3.3.2. *Let gt be a rooted binary gene tree, ST a rooted binary species tree, and u an internal node of gt. Suppose the subtree-bipartition for u is dominated by the subtree-bipartition of v in ST. Then $\mathcal{M}(u) = v$.*

Proof. Since $\mathcal{SBP}_{gt}(u)$ is dominated by $\mathcal{SBP}_{ST}(v)$, it follows that $c_{gt}(u) \subseteq c_{ST}(v)$. Let $w = \mathcal{M}(u)$. Hence, $c_{ST}(v) \cap c_{ST}(w) \neq \emptyset$, and so v and w are comparable (that is, either they are identical or one lies above the other in ST). Suppose by way of contradiction that $v \neq w$. Since $c_{gt}(u) \subseteq c_{ST}(v)$, it follows that v must lie above w. But then $c_{ST}(w)$ is a subset of the cluster of one of v's children, and so disjoint from the cluster for the other child. Hence, $\mathcal{SBP}_{gt}(u)$ is not dominated by $\mathcal{SBP}_{ST}(v)$, contradicting the initial assumption. □

The following corollary is then obvious:

Corollary 3.3.3. *Let gt be a rooted binary gene tree and ST a rooted binary species tree. Then every subtree-bipartition of gt is dominated by at most one subtree-bipartition in ST.*

Theorem 3.3.4. *Let ST be a rooted, binary species tree, gt be a rooted binary gene tree, and u an internal node in gt. Then the subtree-bipartition of u in gt is dominated by a subtree-bipartition in ST if and only if u is a speciation node.*

Proof. Suppose u is a node in gt such that its subtree-bipartition is dominated by a subtree bipartition in ST. Let l and r be the two children of u in gt. Then $\mathcal{SBP}_{gt}(u) = (c(l)|c(r))$. Let v be a node in ST such that $\mathcal{SBP}_{gt}(u)$ is dominated by $\mathcal{SBP}_{ST}(v)$. Let l' and r' be the children of v. Then, without loss of generality, $c(l) \subseteq c(l')$ and $c(r) \subseteq c(r')$. Therefore, under the MRCA mapping, l and r will be mapped to a node in the subtree rooted at l' and r', respectively. Moreover, by Lemma 3.3.2 $\mathcal{M}(u) = v$. Therefore, $\mathcal{M}(l) \neq \mathcal{M}(u)$, and $\mathcal{M}(r) \neq \mathcal{M}(u)$. Hence u is not a duplication node.

Next, assume that $\mathcal{SBP}_{gt}(u)$ is not dominated by any subtree-bipartition of ST, and let $\mathcal{SBP}_{ST}(\mathcal{M}(u)) = (p_1|p_2)$. Then at least one of the following holds (1) $c(l) \not\subset p_1$ and $c(l) \not\subset p_2$ or (2) $c(r) \not\subset p_1$ and $c(r) \not\subset p_2$. Without loss of generality, suppose (1) holds. Then l cannot map to a node strictly below v. However, it is also equally obvious that l cannot map to a node strictly above v, since $\mathcal{M}(u) = v$ and l is a child of u. Hence, it must be that $\mathcal{M}(l) = u$. But in this case, u is a duplication node. □

Comment: As a result, for a given species tree ST and gene tree gt, the vertices of gt partition into duplication nodes, speciation nodes, and leaves, and this partition can be computed using the domination relation.

We now define some functions:

- *dominated*$(bp, ST) \in \{0,1\}$, with *dominated*$(bp, ST) = 1$ if bp is dominated by a subtree-bipartition in $\mathcal{SBP}_{ST}$, and 0 otherwise.
- *dom*$(bp, bp') = 1$ if bp is dominated by bp' and 0 otherwise.

Corollary 3.3.5. *Let gt be a rooted binary gene tree and ST a rooted binary species tree. Then*

$$Dup(gt, ST) = |V_{int}(gt)| - \sum_{u \in V_{int}(gt)} dominated(\mathcal{SBP}_{gt}(u), ST).$$

Proof. Follows directly from Theorem 3.3.4. □

3.4 Algorithms for MGD_r on rooted binary gene trees

3.4.1 Graph-theoretic characterization of optimal solution to MGD_r

Let $\mathcal{G} = \{gt_1, gt_2, \ldots, gt_k\}$ be a set of rooted, binary gene trees on the set $\mathcal{X}$ of n taxa, and let n_i be the number of leaves in tree gt_i. Note that n_i does not refer to $|L(gt_i)|$, since $L(gt_i)$ is the set of taxa in $\mathcal{X}$ that appear at least once in gt_i, whereas n_i is the total number of leaves in gt_i. Since gt_i can have multiple copies of a taxon, n_i can be larger than $|L(gt_i)|$.

We construct the *compatibility graph* $CG(\mathcal{G})$ with one vertex for each possible subtree-bipartition defined on $\mathcal{X}$, as described in the previous section. We set the weight of each node v, denoted by $W_{dom}(v)$, to be the total number of subtree-bipartitions of $\mathcal{G}$ that are dominated by v. That is,

$$W_{dom}(v) = \sum_{gt \in \mathcal{G}} |\{bp : bp \in \mathcal{SBP}_{gt} \text{ and } dom(bp, v) = 1\}|.$$

We then find a clique $\mathcal{C}$ of size $n-1$ so as to maximize the weight $W_{dom}(\mathcal{C})$ of the clique $\mathcal{C}$, where $W_{dom}(\mathcal{C}) = \sum_{v \in \mathcal{C}} W_{dom}(v)$.

Theorem 3.4.1. *Let $\mathcal{G} = \{gt_1, gt_2, \ldots, gt_k\}$ be a set of binary, rooted gene trees on the n taxa in $\mathcal{X}$. Let $\mathcal{C}$ be an $(n-1)$-clique in $CG(\mathcal{G})$ maximizing $W_{dom}(\mathcal{C})$, and let ST be the species tree defined by the clique (so that $\mathcal{SBP}_{ST}$ corresponds to $\mathcal{C}$). Then ST is a binary species tree that optimizes MGD_r with respect to $\mathcal{G}$.*

Proof. Recall that any $(n-1)$-clique in the compatibility graph defines a rooted binary tree on $\mathcal{X}$. Let $\mathcal{C}$ be a clique of size $n-1$ and ST be the tree defined by $\mathcal{C}$. By Corollary 3.3.3, every subtree-bipartition in gt_i can be dominated by at most one node in $\mathcal{C}$. Therefore, each node of gt_i contributes either 1 (if the node is dominated) or 0 (if the node is not dominated) to the weight of $\mathcal{C}$. Let w_i be the amount contributed by gt_i to the weight of $\mathcal{C}$. Thus, w_i is the number of speciation nodes in gt_i with respect to the species tree corresponding to ST. By Corollary 3.3.3, $\sum_{v \in \mathcal{C}} W_{dom}(v)$ is equal to the total number of speciation nodes. Then

$$\sum_{v \in \mathcal{C}} W_{dom}(v) = \sum_{i=1}^{k} w_i = W_{dom}(\mathcal{C}).$$

Furthermore, by Corollary 3.3.5 and because a rooted binary tree with n_i leaves has $n_i - 1$ internal nodes, $Dup(gt_i, ST) = n_i - 1 - w_i$. Then,

$$Dup(\mathcal{G}, T) \;=\; \sum_{i=1}^{k} Dup(gt_i, ST) = \sum_{i=1}^{k} [n_i - 1 - w_i] = N - k - W_{dom}(\mathcal{C}),$$

where $\sum_{i=1}^{k} n_i = N$. Therefore, the clique with maximum weight defines a tree ST that minimizes $Dup(\mathcal{G}, ST)$.

□

3.4.2 Dynamic programming algorithm for MGD_r

The graph-theoretic characterization of the optimal solution for MGD_r given in the previous section suggests an algorithm for finding the optimal solution, in which a max weight clique is sought in an exponentially large graph. However, we will show that this optimal solution can be found in time that is polynomial in the number of vertices in the graph, using dynamic programming. In addition, we will show that a constrained version of the MGD_r problem, in which the allowed subtree-bipartitions are given as input, can also be solved using the same basic dynamic programming algorithm. Finally, when the set of allowed subtree-bipartitions comes from the input set of gene trees, the result is an algorithm that runs in polynomial time.

In the constrained version, instead of constructing a compatibility graph with one node for each subtree bipartition, the compatibility graph will only have nodes for the (at most) $N - k$ subtree bipartitions in the input gene trees (where $N = \sum_{i=1}^{k} n_i$). A clique of size $n - 1$ with the maximum weight will define an optimal solution to the constrained version of MGD_r where the species tree is only permitted to have subtree bipartitions from the input gene trees.

Figure 3.1 illustrates how the constrained version works. Here, the number of duplications associated with the species tree that corresponds to the maximum weight clique is $3 * (4 - 1) - (2 + 2 + 3) = 2$. It is easy to verify that this solution minimizes the possible total, even if all subtree bipartitions had been considered.

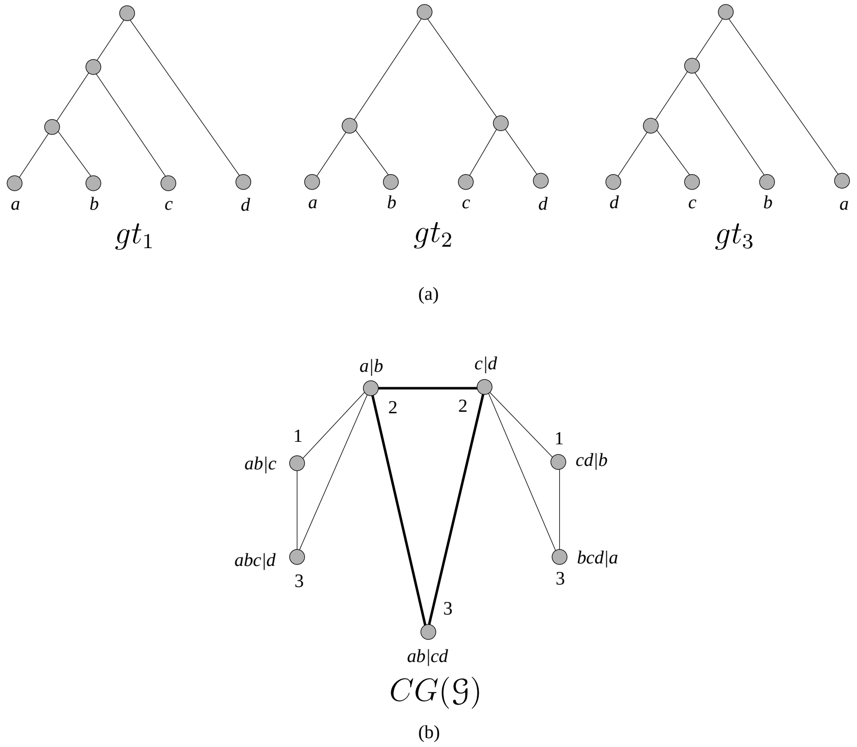


Figure 3.1: **Illustration of the constrained version of our algorithm for MGD**. (a) Three gene trees gt_1, gt_2, and gt_3, (b) the compatibility graph $CG(\mathcal{G})$. Cliques of size three correspond to a species tree on $\{a, b, c, d\}$. The maximum weight clique is indicated by dark lines.

Nodes in	gt_1			gt_2			gt_3		
$CG(\mathcal{G})$	$a\|b$	$ab\|c$	$abc\|d$	$a\|b$	$c\|d$	$ab\|cd$	$c\|d$	$cd\|b$	$bcd\|a$
$(a\|b)$	1	0	0	1	0	0	0	0	0
$(c\|d)$	0	0	0	0	1	0	1	0	0
$(ab\|c)$	0	1	0	0	0	0	0	0	0
$(cd\|b)$	0	0	0	0	0	0	0	1	0
$(ab\|cd)$	0	1	0	0	0	1	0	1	0
$(abc\|d)$	0	0	1	0	1	0	1	0	0
$(bcd\|a)$	1	0	0	1	0	0	0	0	1

Table 3.1: **Demonstration of the weights of the nodes in the compatibility graph illustrated in Fig. 3.1**.

Let $\mathcal{SBP}$ be any set of subtree-bipartitions. We will define the constrained MGD_r problem by limiting the solution space to those rooted, binary trees, all of whose subtree-bipartitions are in the set $\mathcal{SBP}$. Thus, by setting $\mathcal{SBP}$ to be the set of all possible subtree-bipartitions we obtain the globally optimal solution, but setting $\mathcal{SBP}$ to be a proper subset of the set of all subtree-bipartitions is also possible.

By Theorem 3.4.1, the binary species tree with a maximum total weight (as defined by summing up the weights of its subtree bipartitions) has a minimum number of duplications, because the duplication nodes are exactly those nodes whose subtree-bipartitions are not dominated by any subtree-bipartition in the species tree.

We now show how to calculate that optimal binary species tree directly, using dynamic programming. The DP algorithm computes a rooted, binary tree T_A for every cluster A of at least two elements that appears in some gene tree, such that T_A maximizes the sum, over all gene trees t, of the number of subtree-bipartitions in t that are dominated by some subtree-bipartition in T_A. We denote this total number by $value(A)$.

We preprocess the data as follows. First, we compute the cluster $c(x)$ (where $c(x) = p \cup q$ and $x = (p|q)$) for every subtree-bipartition $x \in \mathcal{SBP}$, and order them based on size. We also calculate $\mathcal{SBP}_{\mathcal{G}} = \bigcup_{i=1}^{k} \mathcal{SBP}_{gt_i}$, i.e. the set of all subtree bipartitions in all gene trees, and we set $count(x)$ for $x \in \mathcal{SBP}_{\mathcal{G}}$ to be the number of times x appears in any of the gene trees. Recall that for a subtree bipartition x, we define $W_{dom}(x)$ to be the number of subtree bipartitions of the gene trees that are dominated by x. We define a partial order for elements of $\mathcal{SBP}$ and $\mathcal{SBP}_{\mathcal{G}}$ based upon subtree-bipartition size. For every ordered pair $< x, y >$ such that $x \in \mathcal{SBP}_{\mathcal{G}}$ and $y \in \mathcal{SBP}$, we determine whether x is dominated by y; if y dominates x then $W_{dom}(y)$ is incremented by $count(x)$. At the end of this step, $W_{dom}(y)$ is calculated correctly for every $y \in \mathcal{SBP}$. All the preprocessing can be computed in $O(n|\mathcal{SBP}|^2)$.

We compute $value(A)$ in order, from the smallest cluster to the largest cluster $\mathcal{X}$. We set $value(A)$ as follows. For any cluster A with two taxa, we set $value(A) = W_{dom}(a_1|a_2)$, where $A = \{a_1, a_2\}$. For a cluster A with more than two taxa, we set $value(A)$ as follows:

$$\begin{gathered} value(A) = \max\{value(A_1) + value(A - A_1) + W_{dom}(A_1|A - A_1) : \\ (A_1|A - A_1) \in \mathcal{SBP}\} \end{gathered}$$

If there is no $(A_1|A - A_1) \in \mathcal{SBP}$, we set its $value(A)$ to $-\infty$, signifying that A cannot be further resolved. At the end of the algorithm, if $\mathcal{SBP}$ includes at least one clique of size $n-1$, we have computed $value(\mathcal{X})$ as well as sufficient information to construct the species tree having the minimum number of duplications. If subtree bipartitions in $\mathcal{SBP}$ are not sufficient for building a fully resolved tree on $\mathcal{X}$, then $value(\mathcal{X})$ will be $-\infty$, and our algorithm returns FAIL.

Note that for a specific cluster A, $value(A)$ can be computed in $O(|\mathcal{SBP}|)$ time, since at worst we need to look at every subtree-bipartition in $\mathcal{SBP}$. In other words, we have proven the following:

Theorem 3.4.2. *Let $\mathcal{G}$ be a set of rooted binary gene trees, $\mathcal{SBP}$ a set of subtree-bipartitions. Then the DP algorithm finds the species tree ST minimizing the total number of duplications subject to the constraint that $\mathcal{SBP}_{ST} \subseteq \mathcal{SBP}$ in $O(n|\mathcal{SBP}|^2)$ time. Therefore, if $\mathcal{SBP}$ is all possible subtree-bipartitions, we have an exact but exponential time algorithm. However, if $\mathcal{SBP}$ contains only those subtree-bipartitions from the input gene trees, then the DP algorithm finds the optimal constrained species tree in $O(d^2n^3k^2)$ (since the number of subtree-bipartitions $|\mathcal{SBP}|$ in $\mathcal{G}$ is $O(dkn)$), where n is the number of species, k is the number of gene trees, and d the maximum number of times that any taxon appears in any gene tree.*

3.5 Algorithms for $MGDL_r$

3.5.1 Graph-theoretic characterization

We begin with some additional theorems.

Theorem 3.5.1. *(From Than and Nakhleh [141] and Yu, Warnow, and Nakhleh [155]) Let gt be a rooted binary gene tree and ST a species tree on the same set of taxa. Let B be a cluster in ST and let e be the parent edge of B in ST. Then $k_B(gt)$ is equal to the number of lineages on e in an optimal reconciliation of gt within ST with respect to MDC_c. Therefore, $MDC_c(gt, ST) = \sum_B (k_B(gt) - 1)$, where B ranges over the clusters of ST.*

Theorem 3.5.2. *Let gt be a rooted binary gene tree and ST a species tree on the same set of leaves. Then $MDC_r(gt, ST) = \sum_B (k_B(gt) - 1)$, where B ranges over the clusters of $ST(gt)$.*

Proof. By definition, $MDC_r(gt, ST) = MDC_c(gt, ST(gt))$. However, gt and $ST(gt)$ have the same set of taxa. Therefore, by Theorem 3.5.1,

$$MDC_c(gt, ST(gt)) = \sum_B (k_B(gt) - 1),$$

as B ranges over the clusters of $ST(gt)$. □

Theorem 3.5.3. *(From Zhang [158]) Let gt be a rooted binary gene tree and ST a rooted binary species tree. Then, under the restriction-based analysis, $Duploss(gt, ST) = MDC_r(gt, ST) + 3 * Dup(gt, ST) + |V(gt)| - |V(\mathcal{R}_{ST}(L(gt)))|$.*

Let v be a vertex associated with the subtree-bipartition $(p|q)$, and let $B = p \cup q$ be the cluster associated with v. Define

$$W_{xl}(v, gt) = \begin{cases} 0 & \text{if } p \cap L(gt) = \emptyset \text{ or } q \cap L(gt) = \emptyset, \\ k_B(gt) - 1 & \text{otherwise.} \end{cases} \tag{3.1}$$

We then set $W_{xl}(v) = \sum_{i=1}^{k} W_{xl}(v, gt_i)$. Then, for any species tree ST and set $\mathcal{G}$ of gene trees,

$$MDC_r(\mathcal{G}, ST) = \sum_{i=1}^{k} MDC_r(gt_i, ST) = \sum_{v \in \mathcal{C}} W_{xl}(v) \tag{3.2}$$

where $\mathcal{C}$ is the clique in $CG(\mathcal{G})$ that corresponds to ST.

Theorem 3.5.4. *Let $\mathcal{G} = \{gt_1, gt_2, \ldots, gt_k\}$ be a set of binary rooted gene trees on set $\mathcal{X}$ of n species, and let $CG(\mathcal{G})$ be the compatibility graph with vertex weights defined by $W_{MGDL}(v) = W_{xl}(v) - 3W_{dom}(v)$. The set of bipartitions in an $(n-1)$-clique of minimum weight in $CG(\mathcal{G})$ defines a binary species tree ST that optimizes $MGDL_r$.*

Proof. Let $\mathcal{C}$ be a clique of size $n-1$ and ST be the rooted binary tree defined by the subtree-bipartitions represented by the nodes in $\mathcal{C}$. Let $\mathcal{SBP}_{dom}(gt, ST)$ be the set of subtree-bipartitions in gt that are dominated by a subtree-bipartition in ST, i.e.,

$$\mathcal{SBP}_{dom}(gt, ST) = \{bp : bp \in \mathcal{SBP}_{gt} \text{ and } \mathit{dominated}(bp, ST) = 1\}.$$

Note that $|\mathcal{SBP}_{dom}(gt, ST)|$ is the number of speciation nodes in gt with respect to ST. Therefore, the total number of speciation nodes in $\mathcal{G}$ is

$$\sum_{i=1}^{k} |\mathcal{SBP}_{dom}(gt_i, ST)| = \sum_{v \in V_{int}(ST)} W_{dom}(v).$$

Let $N = \sum_{i=1}^{k} n_i$. Then,

$$\begin{aligned}
Duploss(\mathcal{G}, ST) &= \sum_{i=1}^{k} Duploss(gt_i, ST) \\
&= \sum_{i=1}^{k} [MDC_r(gt_i, ST) + 3 * Dup(gt_i, ST) \\
&\quad -(|V(gt_i)| - |V(\mathcal{R}_{ST}(L(gt_i)))|)] \text{ (by Theorem 3.5.3)} \\
&= \sum_{i=1}^{k} [MDC_r(gt_i, ST) + 3 * Dup(gt_i, ST)] \\
&\quad -\sum_{i=1}^{k} (|V(gt_i)| - |V(\mathcal{R}_{ST}(L(gt_i)))|) \\
&= \sum_{i=1}^{k} [MDC_r(gt_i, ST) + 3 * ((n_i - 1) - |\mathcal{SBP}_{dom}(gt_i, ST)|)] \\
&\quad -\sum_{i=1}^{k} (|V(gt_i)| - |V(\mathcal{R}_{ST}(L(gt_i)))|) \text{ (by Corollary 3.3.5)} \\
&= \sum_{v \in \mathcal{C}} W_{xl}(v) + \sum_{i=1}^{k} 3(n_i - 1) - 3 \sum_{v \in \mathcal{C}} W_{dom}(v) - \sum_{i=1}^{k} (2n_i - 1) \\
&\quad + \sum_{i=1}^{k} |V(\mathcal{R}_{ST}(L(gt_i)))| \text{ (since } |V(gt_i)| = 2n_i - 1) \\
&= \sum_{v \in \mathcal{C}} (W_{xl}(v) - 3W_{dom}(v)) + 3 \sum_{i=1}^{k} n_i \; - 3k - 2 \sum_{i=1}^{k} n_i + k \\
&\quad + \sum_{i=1}^{k} |V(\mathcal{R}_{ST}(L(gt_i)))| \\
&= \sum_{v \in \mathcal{C}} W_{MGDL}(v) + \sum_{i=1}^{k} n_i - 2k + \sum_{i=1}^{k} |V(\mathcal{R}_{ST}(L(gt_i)))| \\
&= W_{MGDL}(\mathcal{C}) + N - 2k + \sum_{i=1}^{k} |V(\mathcal{R}_{ST}(L(gt_i)))|
\end{aligned}$$

Note that $|V(\mathcal{R}_{ST}(L(gt_i)))|$ does not depend on ST. Therefore, the clique $\mathcal{C}$

with minimum weight defines a tree ST that minimizes $Duploss(\mathcal{G}, ST)$.

□

3.5.2 Dynamic programming approach for $MGDL_r$

We now show how to use dynamic programming to find the optimal solution for $MGDL_r$ without having to explicitly search for the optimal clique. As we did for MGD_r, we generalize the problem to allow the user to provide a set $\mathcal{SBP}$ of subtree-bipartitions, and the solution space is restricted to those rooted, binary trees, all of whose subtree-bipartitions are in the set $\mathcal{SBP}$.

We compute $value(A)$ for all clusters A with at least two species as follows. If $|A| = 2$, we set $value(A) = W(a_1|a_2)$, where $A = \{a_1, a_2\}$. For set A with more than two taxa, we set $value(A)$ as follows:

$$value(A) = \min\{value(A_1) + value(A - A_1) + W_{xl}(A_1|A - A_1) - \\ 3W_{dom}(A_1|A - A_1) : (A_1|A - A_1) \in \mathcal{SBP}\}.$$

The optimal number of duplications and losses is given by $value(\mathcal{X}) + N - 2k + \sum_{i=1}^{k} |V(\mathcal{R}_{ST}(L(gt_i))|$, where $N = \sum_{i=1}^{k} n_i$, and n_i is the number of leaves in gene tree gt_i. By backtracking, we can find the optimal set of compatible clusters and hence can construct the optimal tree. We now have the following theorem:

Theorem 3.5.5. *Let $\mathcal{G}$ be a set of k rooted binary gene trees on the set $\mathcal{X}$ of n taxa. Let $\mathcal{SBP}$ be an arbitrary set of subtree bipartitions on $\mathcal{X}$. Then the DP algorithm finds the species tree ST optimizing the restriction-based DUPLOSS* *problem, subject to the constraint that $\mathcal{SBP}_{ST} \subseteq \mathcal{SBP}$, in $O(n|\mathcal{SBP}|^2)$ time. Therefore, for the case where $\mathcal{SBP}$ is the set of subtree-bipartitions from the k gene trees, the algorithm uses $O(d^2n^3k^2)$ time, where d is the maximum number of times any taxon appears in any gene tree.*

3.6 Conclusion

We have presented graph-theoretic characterizations of exact solutions to the MGD_r and $MGDL_r$ problems, and presented dynamic programming algorithms for these problems. Furthermore, these results enable the user to provide a set of subtree-bipartitions to define a constrained search for the species tree, and thus make it possible to find optimal solutions subject to these constraints in polynomial time. For the particular case where the set of subtree-bipartitions is all subtree-bipartitions from the input gene trees, these methods could be quite fast. Furthermore, for large enough numbers of taxa and gene trees, these algorithms may present an advantage compared to methods that are based upon local search techniques in which candidate species trees are visited and scored with respect to the desired criterion, and then the tree is topologically modified (for example by a TBR move) and the new tree scored, etc. However, the relative advantages of this approach compared to local search techniques still remains to be explored.

Certain theoretical questions are not addressed in this study. In particular, we have not addressed the case where the gene trees are unrooted. We have also not addressed optimizing MGD_c and $MGDL_c$ for sets of incomplete gene trees.

We have implemented our dynamic programming algorithm in a publicly available software tool (`http://www.cs.utexas.edu/users/phylo/software/dynadup/`).

Chapter 4

Inferring Optimal Species Trees under Gene Duplication and Loss: Beyond Rooted Gene Trees

In this chapter we extend our algorithms for MGD_r and $MGDL_r$ (Bayzid *et al.* [6], discussed in Chapter 3) so that they can handle unrooted gene trees when gene trees can have at most one copy per species. We extend the concept of subtree-bipartition and domination (introduced in Chapter 3) to unrooted gene trees, and show how to find the set of subtree-bipartitions for an unrooted gene tree gt. We first show how to root an unrooted gene tree gt with respect to a species tree ST so that the resulting rooted version gt^* of gt minimizes the duplication and duplication-loss costs over all possible rooted versions of gt. We develop linear time algorithm to find this optimal rooting. Finally, we modify the weight assignment for each vertex in the compatibility graph (as defined in Chapter 3) so that our DP-based algorithm can solve the MGD and MGDL problems for unrooted gene trees given the extended definitions of *subtree-bipartition* and *domination.*

The results presented in this chapter are not published yet. Md. Shamsuzzoha Bayzid designed the study, developed the algorithms, proved all the theoretical results, and wrote the chapter with comments from Tandy Warnow.

4.1 Introduction

One of the basic approaches to understanding differences between true gene trees is duplication events [37, 43, 84, 107, 157]. Gene duplication is a potential tool for constructing phylogenetic trees for snakes [127], vertebrates [108, 109], *Drosophia* [20], and plants [125]. In this context, a natural computational problem is the *Minimize Gene Duplications* (MGD) problem, which seeks a species tree minimizing the total number of duplications needed to explain the observed gene trees. Related to MGD is the *Minimize Gene Duplication and Loss* (MGDL) problem, which considers both duplications and losses in scoring a tree.

We introduced exact algorithms for inferring species trees under the MGD and MGDL criteria from a collection of rooted, binary gene trees [6] (described in Chapter 3). Nevertheless, in phylogenetic analyses of biological data sets, estimated gene trees may be unrooted. This is mostly because because DNA mutation models are time reversible, and this makes the root of the tree non-identifiable [25]. Unrooted gene trees are often converted to rooted trees using an outgroup so that the root is necessarily between the outgroup and the rest of the taxa in the tree, or introducing additional assumption of the presence of molecular clock. However, finding an appropriate outgroup is very difficult and pre-specified outgroup may result in biased placement of the root [46, 112]. Therefore, in the absence of a molecular clock or appropriate outgroup, rooting trees can be difficult [9, 56], and so it is desirable to develop methods for estimating species trees from unrooted gene trees. In this way, the

MGD (and MGDL) problem becomes one in which the input is a set of gene trees that may not be rooted, and the objective is a rooted, binary species tree that optimizes the MGD (and MGDL) criterion. We provide exact algorithms and heuristics for inferring species trees for these cases.

4.2 Prior terminology and theory

4.3 MGD_{ru} and $MGDL_{ru}$

We formulate the minimize gene duplication (MGD) and minimize gene duplication and loss (MGDL) problems for unrooted gene trees when the gene trees can have at most one copy per species, and show how to solve them with the dynamic programming based approach presented in [6] for rooted gene trees. The input to each problem is the same: a set $\mathcal{G} = \{gt_1, gt_2, \ldots, gt_k\}$ of rooted binary gene trees, with leaves drawn from the set $\mathcal{X}$ of n taxa, and we only allow the gene trees to have exactly one copy of the taxa, or to miss some taxa. MGD for unrooted case, which we call MGD_{ru}, can be defined as follows. The input to MGD_{ru} is a set $\mathcal{G} = \{gt_1, gt_2, \ldots, gt_k\}$ of unrooted and binary gene trees such that $L(gt_i) \subseteq \mathcal{X}$, where $i \in 1, 2, \ldots, k$, and at most one copy per species is present in each of the gene trees. The output is a rooted and binary species tree ST on $\mathcal{X}$ and set $\mathcal{G}^* = \{gt_1^*, gt_2^*, \ldots, gt_k^*\}$, where gt_i^* is a rooted version of gt_i, such that $Dup(\mathcal{G}^*, ST)$ is minimized.

$MGDL_{ru}$ is defined in a similar way, where the objective function to minimize is $Duploss(\mathcal{G}^*, ST)$. In Chapter 3, we proposed dynamic programming (DP) based exact solutions for MGD_r and $MGDL_r$, when gene trees are rooted, such that the DP algorithm finds the species tree ST minimizing the total number of duplications (for MGD) or total number of duplication and losses (for MGDL) subject to the constraint that $\mathcal{SBP}_{ST} \subseteq \mathcal{SBP}$ in $O(n|\mathcal{SBP}|^2)$ time, where $\mathcal{SBP}$ is a set of subtree-bipartitions. For rooted gene trees, our result was as follows.

Theorem 4.3.1. *(From Bayzid et al. [6], described in Chapter 3) Let $\mathcal{G}$ be a set of rooted binary gene trees, $\mathcal{SBP}$ a set of subtree-bipartitions. Then the DP algorithm finds the species tree ST minimizing the total number of duplications (for MGD_r) or total number of duplications and losses (for $MGDL_r$) subject to the constraint that $\mathcal{SBP}_{ST} \subseteq \mathcal{SBP}$ in $O(n|\mathcal{SBP}|^2)$ time. Therefore, if $\mathcal{SBP}$ is all possible subtree-bipartitions, we have an exact but exponential time algorithm. However, if $\mathcal{SBP}$ contains only those subtree-bipartitions from the input gene trees, then the DP algorithm finds the optimal constrained species tree in $O(d^2n^3k^2)$ (since the number of subtree-bipartitions $|\mathcal{SBP}|$ in $\mathcal{G}$ is bounded by $O(dkn)$), where n is the number of species, k is the number of gene trees, and d the maximum number of times that any taxon appears in any gene tree.*

In this chapter, we extend this result so that our algorithms can handle unrooted gene trees. Therefore, we have the following result when gene trees are unrooted and can have at most one copy of a particular gene per species.

Theorem 4.3.2. *Let $\mathcal{G}$ be a set of unrooted binary gene trees with single copy per species, $\mathcal{SBP}$ a set of subtree-bipartitions. Then the DP algorithm finds the species tree ST minimizing the total number of duplications (or duplications and losses when solving $MGDL_{ru}$), with respect to the optimal rooting* *of each of the gene trees in $\mathcal{G}$, subject to the constraint that $\mathcal{SBP}_{ST} \subseteq \mathcal{SBP}$ in $O(n|\mathcal{SBP}|^2)$ time. Therefore, if $\mathcal{SBP}$ is all possible subtree-bipartitions, we have an exact but exponential time algorithm. However, if $\mathcal{SBP}$ contains only those subtree-bipartitions from the input gene trees, then the DP algorithm finds the optimal constrained species tree in $O(n^3k^2)$ (since the number of subtree-bipartitions $|\mathcal{SBP}|$ in $\mathcal{G}$ is bounded by $O(kn)$ as we will show in Theorem 4.4.1 and Corollary 4.4.2), where n is the number of species and k is the number of gene trees.*

4.4 MGD and MGDL on unrooted, binary, and incomplete gene trees

We now extend our DP-based approach for MGD_r and $MGDL_r$ to MGD_{ru} and $MGDL_{ru}$. The optimal solution for MGD_r is characterized graph-theoretically as follows. First, every binary rooted tree on n taxa can be represented by its set of subtree-bipartitions. To solve MGD_r, each possible subtree-bipartition is represented by a node in a graph, and edges exist between pairs of nodes whose subtree-bipartitions are compatible. It is proved that whenever a set of subtree-bipartition is given that are all pairwise compatible, then a rooted tree exists with precisely that set of subtree-bipartitions [6] (see Observation 3.3.1 in Chapter 3). Thus, a set of $n-1$ pairwise compatible subtree-bipartitions, where n is the number of species, defines a binary rooted species tree for that set of subtree-bipartition.

Bayzid *et al.* [6] (described in Chapter 3) showed that it is possible to weight the nodes in the graph so that the total weight of any $(n-1)$-clique is the number of speciation nodes (in the input gene trees) for the species tree defined by that clique, so that solving the MGD_r problem is equivalent to finding a maximum (maximizing speciation nodes minimizes the number of duplication nodes) weight $(n-1)$-clique.

This problem formulation seems to be particularly expensive, since MaxClique is NP-hard and the graph has an exponential number of vertices, but Bayzid *et al.* [6] (see Chapter 3) also showed that finding the minimum weight clique of size $n-1$ can be obtained in time that is polynomial in the number of nodes in the graph, using dynamic programming (DP). We also presented a "heuristic" version that only uses subtree-bipartitions that appear in the input gene trees, and so runs in polynomial time. We also showed how to extend this to the $MGDL_r$ problem, using the same graph but with different weights on the edges. We refer to Chapter 3 for details.

The approaches we present here for solving MGD_{ru} or $MGDL_{ru}$ are based on these ideas. We first show how to root an unrooted gene tree with respect to a species tree in an optimal way under duplication and duplication and loss criteria. Then we modify our weight assignment accordingly so that the dynamic programming technique can still be applied to unrooted gene trees.

Given a species tree and a set of unrooted gene trees, it is easy to compute the optimal rooting of each gene tree with respect to the given species tree, since there are only $O(n)$ possible locations for the root in a gene tree with n leaves, and for each possible rooting we can compute the score of that solution in $O(n)$ time. Thus, we can compute the optimal rooting in $O(n^2)$ time. Here we present a more efficient way of solving this problem by finding the optimal rooting in $O(n)$ time that saves a factor of n. Next, we will extend our algorithms for MGD_r and $MGDL_r$ to unrooted gene trees that are more efficient than the algorithms presented in [50].

Extending the concept of subtree-bipartition The set of subtree-bipartitions of a tree T depends on whether or not T is rooted. Each internal node of a rooted tree T defines one subtree-bipartition. However, for an unrooted tree T, the set of subtree-bipartitions contains all the subtree-bipartitions for all possible rooted versions of T. That is, if T is an unrooted tree, $\mathcal{SBP}_T = \cup_i \mathcal{SBP}_{T_i}$, where T_i ranges over all possible rooted versions of T.

For any binary unrooted tree, three edges are incident on any internal node u (see Fig 4.1(a)). The tree can be rooted at any of these three edges. Figures 4.1(b)–(d) demonstrate these rooted versions. Throughout this chapter, we denote by A, B and C the three clusters associated with an internal node u of an unrooted gene tree gt (see Figure 4.1(a)). Then, the set of all subtree-bipartitions at an internal node u is $\{A|B, (A \cup B)|C, A|C, (A \cup C)|B, B|C, (B \cup C)|A\}$. Therefore, for an unrooted tree gt, $\mathcal{SBP}_{gt}(u)$ is this set of six elements. Among them, $(A \cup B)|C, (A \cup C)|B, (B \cup C)|A$ contains all the taxa. For any node u (as shown in Fig 4.1(a)), we define $\mathcal{SBP}^*_{gt}(u) = \{A|B, B|C, A|C\}$. Hence, $\mathcal{SBP}^*_{gt}(u) \subsetneq \mathcal{SBP}_{gt}(u)$. For an unrooted gene tree gt, we define $\mathcal{SBP}_{gt}$

as follows.

$$\mathcal{SBP}_{gt} = \cup_{u \in V_{int}(gt)} \mathcal{SBP}_{gt}(u), \tag{4.1}$$

where $\mathcal{SBP}_{gt}(u)$ is the set of six subtree-bipartitions as described above. Therefore,

Theorem 4.4.1. *Let gt be an unrooted gene tree with n leaves. Then, $|\mathcal{SBP}_{gt}| = 5n - 9$.*

Proof. Note that for any two internal nodes u_1 and u_2, $\mathcal{SBP}^*_{gt}(u_1) \cap \mathcal{SBP}^*_{gt}(u_2) = \phi$. For any internal node u, the three subtree-bipartitions in $\mathcal{SBP}_{gt}(u)$, that contain all the taxa are the splits defined by the three edges incident on u. Therefore, since an unrooted tree with n taxa contains $n - 2$ internal nodes and $2n - 3$ edges, $|\mathcal{SBP}_{gt}| = 3(n - 2) + 2n - 3 = 5n - 9$. □

Corollary 4.4.2. *For a set $\mathcal{G} = \{gt_1, gt_2, \ldots, gt_k\}$ of unrooted gene trees, where $L(gt_i) \subseteq \mathcal{X}$ and $|\mathcal{X}| = n$,*

$$|\mathcal{SBP}_{\mathcal{G}}| = \sum_{i=1}^{k} |\mathcal{SBP}_{gt_i}| = O(kn).$$

Proof. Follows immediately from Theorem 4.4.1. □

We now extend the notion of domination to unrooted gene trees. We say an internal node u of an unrooted gene tree gt is a dominated node, with respect to a species tree T, if a subtree-bipartition in $\mathcal{SBP}_{gt}(u)$ is dominated by a subtree-bipartition in T.

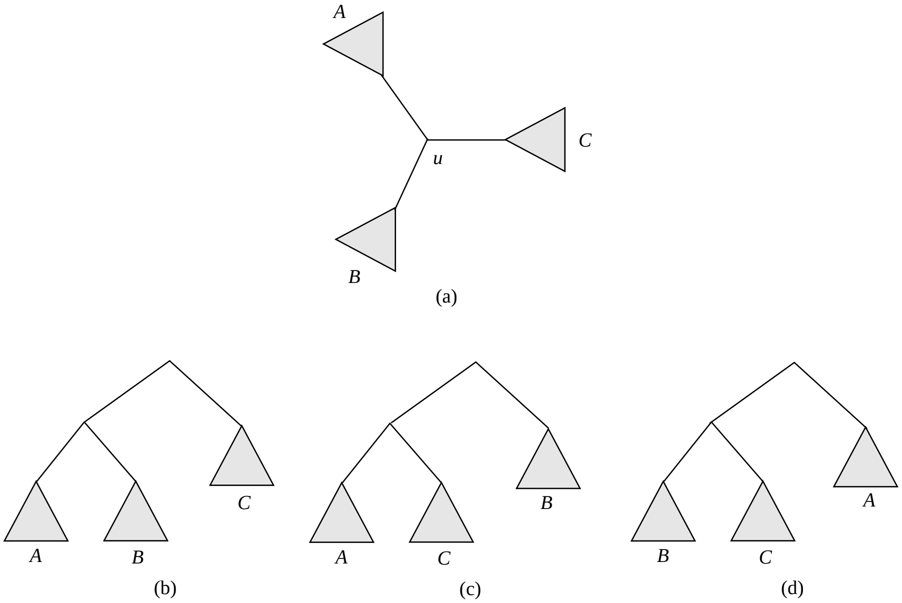


Figure 4.1: **Illustration of different rooted versions of an unrooted tree around a particular internal node.** (a) An unrooted gene tree gt, (b)–(d) three possible rooted versions of gt on the three edges incident on u.

4.4.1 Optimal rooting

We now show how to root an unrooted, binary gene tree gt with at most single copy per species with respect to a rooted binary species tree ST to minimize the number of duplications. For clarity, we will assume that $L(gt) = L(ST)$. In general, if $L(gt) \subset L(ST)$ then we can simply set ST to be $ST|_{L(gt)}$.

Lemma 4.4.3. *Let gt be a binary unrooted gene tree and ST be a binary* *rooted species tree, both on the same set of taxa $\mathcal{X}$. If gt does not contain the root subtree-bipartition of ST ($\mathcal{SBP}_{ST}(root(ST))$), then for any node u in gt, at most one of its subtree-bipartitions can be dominated.*

Proof. Let A, B and C be the clusters associated with u. Note that $A|BC$, $AB|C$, and $AC|B$ contain all the taxa. Therefore, only the root subtree-bipartition of ST can dominate them. Moreover, it is quite easy to see that the root subtree-bipartition of ST cannot dominate more than one of these three subtree-bipartition. Thus, if the root subtree-bipartition of ST dominate any of these three subtree-bipartition, then the dominated subtree-bipartition is exactly the same subtree-bipartition defined at the root of ST. Therefore, if gt does not contain $\mathcal{SBP}_{ST}(root(ST))$, then none of these three subtree-bipartitions can be dominated. Now consider the other three possible subtree-bipartitions ($A|B$, $B|C$, and $A|C$). We assume for a contradiction that two of them are dominated. Without loss of generality, assume that both $A|B$ and $B|C$ are dominated. Then the only subtree-bipartition that can dominate these two is $AC|B$ which contains all the taxa and must be $\mathcal{SBP}_{ST}(root(ST))$, contradicting our hypothesis that gt does not have the root subtree-bipartition of ST. □

Lemma 4.4.4. *Let gt be a binary unrooted gene tree and ST be a binary rooted species tree, both on the same set of taxa $\mathcal{X}$. Assume that gt contains the root subtree-bipartition of ST ($\mathcal{SBP}_{ST}(root(ST))$), and let $e \in E(gt)$ be the edge that defines that subtree-bipartition. Then, for an internal vertex u which is incident on e, two of the subtree-bipartitions in $\mathcal{SBP}^*_{gt}(u)$ are* *dominated by $\mathcal{SBP}_{ST}(root(ST))$. Also, the bipartition defined by e will be dominated. Therefore, three subtree-bipartitions in $\mathcal{SBP}_{gt}(u)$ are dominated by $\mathcal{SBP}_{ST}(root(ST))$. For all other internal nodes v, none of its subtree-bipartitions is dominated by $\mathcal{SBP}_{ST}(root(ST))$.*

Proof. Let $\mathcal{SBP}_{ST}(root(ST)) = X|Y$. Then gt is as illustrated in Fig 4.2(a). Let u_1 and u_2 be the two vertices incident on e. First, we prove that two subtree-bipartitions from each of the two sets $\mathcal{SBP}^*_{gt}(u_1)$ and $\mathcal{SBP}^*_{gt}(u_2)$ are dominated by $X|Y$. We assume that $X = X_1 \cup X_2$ and $Y = Y_1 \cup Y_2$. Note that since u_1 and u_2 are internal node, X_1, X_2, Y_1, Y_2 are non-empty. Here, $\mathcal{SBP}^*_{gt}(u_1) = \{X_1|Y, X_2|Y, X_1|Y\}$; and $\mathcal{SBP}^*_{gt}(u_2) = \{Y_1|X, Y_2|X, Y_1|Y_2\}$. It is easy to see that $X|Y$ dominates $X_1|Y, X_2|Y \in \mathcal{SBP}^*_{gt}(u_1)$, and it also dominates $Y_1|X, Y_2|X \in \mathcal{SBP}^*_{gt}(u_2)$. Moreover, $X|Y \in \mathcal{SBP}_{gt}(u_1)$, and $X|Y \in \mathcal{SBP}_{gt}(u_2)$.

Next, we prove that for an internal node v (other than u_1 and u_2), no subtree-bipartition in $\mathcal{SBP}_{gt}(v)$ is dominated by $X|Y$. First, note that $X|Y \notin \mathcal{SBP}_{gt}(v)$. Thus, the subtree-bipartitions of $\mathcal{SBP}_{gt}(v)$ that contain all the taxa cannot be dominated by $X|Y$. Next, we prove that no subtree-bipartition in $\mathcal{SBP}^*_{gt}(v)$ is dominated by $X|Y$. The reason is, for any internal node v (other than u_1 and u_2), if v is inside the subtree induced by X, then for any subtree-bipartition $P|Q \in \mathcal{SBP}^*_{gt}(v)$, $P \cap X \neq \phi, Q \cap X \neq \phi$. So they cannot be dominated by $X|Y$. Again, if v is inside the subtree induced by Y, then for any subtree-bipartition $P|Q \subset \mathcal{SBP}^*_{gt}(v)$, $P \cap Y \neq \phi, Q \cap Y \neq \phi$. Hence they cannot be dominated by $X|Y$. For an example, in Fig 4.2(b),

$\mathcal{SBP}^*_{gt}(v) = \{Y_1|(Y_3 \cup X), Y_2|(Y_3 \cup X), Y_1|Y_2\}$. It is easy to see that none of them is dominated by $X|Y$. It is also easy to see that $X|Y \notin \mathcal{SBP}_{gt}(v)$. □

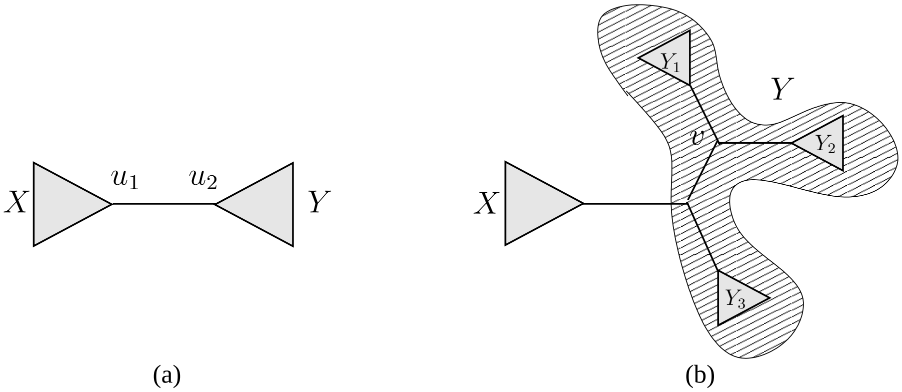


Figure 4.2: (a) An unrooted gene tree containing the root subtree-bipartition of ST ($\mathcal{SBP}_{ST}(root(ST)) = X|Y$), (b) an internal node inside the subtree induced by Y.

Lemma 4.4.5. *Suppose gt does not contain the root subtree-bipartition of ST ($\mathcal{SBP}_{ST}(root(ST)) = X|Y$). Let $\mathcal{DS}$ be the set of dominated subtree-bipartitions in $\mathcal{SBP}_{gt}$ with respect to ST. Let $gt_{A|B}$ be the restriction of gt into $A|B$, which means $gt_{A|B}$ is the subtree of gt induced by the cluster $A \cup B$. Then the tree gt^* produced by rooting gt on an edge $e \in (E(gt) - \cup_{X|Y \in \mathcal{DS}} E(gt_{X|Y}))$ satisfies $Dup(gt^*, ST) \leq Dup(gt', ST)$, where gt' is a rooted version of gt.*

Proof. Let an internal node u in gt be dominated by ST. By Lemma 4.4.3, at most one subtree-bipartition in $\mathcal{SBP}_{gt}(u)$, $u \in V(gt)$ can be dominated. Then, without loss of generality, let the subtree-bipartition $A|B$ at u be dominated by ST (see Figure 4.1(a)). Let $gt_{A|B}$ be the restriction of gt into $A|B$. Then the tree gt', rooted on an edge $e \in E(gt) - E(gt_{A|B})$, will contain the subtree-bipartition $A|B$. Let $\mathcal{DS}$ be the set of dominated subtree-bipartitions in $\mathcal{SBP}_{gt}$ by ST. Then the tree gt^* rooted on an edge $e \in (E(gt) - \cup_{X|Y \in \mathcal{DS}} E(gt_{X|Y}))$ will contain all the subtree-bipartitions in $\mathcal{DS}$. Now we show that there is at least one edge in $E(gt) - \cup_{X|Y \in \mathcal{DS}} E(gt_{X|Y})$. Note that the root subtree-bipartition of ST contains all the taxa and hence the subtree induced by this subtree-bipartition contains all the edges in gt. Since gt does not have the root subtree-bipartition, $\mathcal{DS}$ does not contain any subtree-bipartition that contains all the taxa. Pick a subtree-bipartition $S_1 = X|Y \in \mathcal{DS}$ which is maximal (i.e., it does not contain any subtree-bipartition in $\mathcal{DS}$). We will show that the parent edge e of S_1 is not inside any subtree of gt induced by a subtree-bipartition in $\mathcal{DS}$. Suppose, for the way of contradiction, e is inside the subtree induced by a subtree-bipartition $S_2 = P|Q \in \mathcal{DS}$. Since S_1 is maximal, it follows that $S_1 \not\subseteq S_2$. Moreover, since S_2 contains e, it follows that $S_2 \not\subseteq S_1$. Thus, either S_1 and S_2 are disjoint, or one of the following two holds.

- Either $P = X$, or ($P = Y$ and $Q \notin X \cup Y$).
- Either $Q = X$, or ($Q = Y$ and $P \notin X \cup Y$).

That means, $S_1 \cap S_2 = \emptyset$, or $S_1 \cap S_2 = \mathcal{SS}$ where $\mathcal{SS} = X$, or $\mathcal{SS} = Y$ ($\mathcal{SS}$ cannot contain taxa from both X and Y, otherwise S_2 cannot be dominated by ST). For the former case, where S_1 and S_2 are disjoint, S_2 cannot contain

e. For the later case, without loss of generality, we assume that $P = Y$. Then, $X|Y \cup Q \in \mathcal{DS}$ which contradicts our assumption that S_1 is maximal.

We now show that for any rooted version gt' of gt, $Dup(gt^*, ST) \leq Dup(gt', ST)$. Note that $\mathcal{DS}$ contains all the dominated subtree-bipartitions in all possible rooted versions of gt. Then clearly, $|\mathcal{DS}| \geq |\mathcal{DS}'|$, where $\mathcal{DS}'$ is the set of dominated subtree-bipartitions in any rooted version gt' of gt. Since gt^* has all the subtree-bipartitions in $\mathcal{DS}$,

$$\begin{aligned} Dup(gt^*, ST) &= n - 1 - |\mathcal{DS}| \\ &\leq n - 1 - |\mathcal{DS}'| \\ &= Dup(gt', ST). \end{aligned}$$

□

Lemma 4.4.6. *Suppose gt contains the root subtree-bipartition of ST ($\mathcal{SBP}_{ST}(root(ST))$), and let $e^* \in E(gt)$ be the edge that defines that subtree-bipartition. Let gt^* be the rooted tree obtained by rooting gt on e^*. Then for a rooted version gt' of gt, $Dup(gt^*, ST) \leq Dup(gt', ST)$.*

Proof. Let $\mathcal{SBP}_{ST}(root(ST)) = X|Y$. Then $\mathcal{SBP}_{gt^*}(root(gt^*)) = X|Y$, and hence $root(gt^*)$ is a speciation node. Let gt_X and gt_Y be the subtrees of gt induced by X and Y, respectively. Then clearly, $e^* = E(gt) - E(gt_X) - E(gt_Y)$. Let gt' be a rooted version of gt produced by rooting gt on an edge other than e^*. Then $\mathcal{SBP}_{gt'}(root(gt')) \neq \mathcal{SBP}_{ST}(root(ST))$. Hence $root(gt')$ is a duplication node (since $root(gt')$ can only be dominated by $root(ST)$).

Note that gt_X or gt_Y must have at least one edge in it, otherwise there are only two taxa in the gene tree gt and only one possible rooting and hence $gt^* = gt'$. Without loss of generality, we assume that gt' is rooted on the edge $e \in E(gt_X)$. Then it is easy to see that all the subtree-bipartitions in gt'_Y are also present in gt^*_Y. All other subtree-bipartitions in gt^* (other than those in gt'_Y and $\mathcal{SBP}_{gt'}(root(gt'))$ are of the form $P|Q$ such that *case 1*) $P, Q \in X$, or *case 2*) either $Y \subseteq P$ or $Y \subseteq Q$ (without loss of generality we assume that $Y \subseteq P$). It is easy to see that all the subtree-bipartitions satisfying *case 1* are also present in gt^*. We now consider the subtree-bipartitions $P|Q$ satisfying *case 2*. Notice that only $\mathcal{SBP}_{ST}(root(ST))$ can dominate such subtree-bipartitions, since $Y \subseteq P$. Let u be the internal node in gt_X which is incident on e^*. By Lemma 4.4.4, $\mathcal{SBP}_{gt'}(u)$ is dominated by $\mathcal{SBP}_{ST}(root(ST))$, and any subtree-bipartition $\mathcal{SBP}_{gt'}(u_1)$ satisfying *case 2*, where $u_1 \neq u$, is not dominated. For an example, let $X = \{u\} \cup X_1 \cup X_2$ as illustrated in Fig. 4.3. If gt' is obtained by rooting on $e \in E(gt_{\{u\} \cup X_1})$, then $SBP_{gt'}(u) = X_2|Y$ is dominated by $root(ST)$. If gt' is obtained by rooting on $e \in E(gt_{\{u\} \cup X_2})$, then $SBP_{gt'}(u) = X_1|Y$ is dominated by $root(ST)$. Therefore, regardless of whether u is a dominated node in gt^*, the number of dominated nodes in gt^* is greater than or equal to the number of dominated nodes in gt'. Therefore, $Dup(gt^*, ST) \leq Dup(gt', ST)$. □

Consider the scenario described in Lemma 4.4.6. Clearly $e^* \notin (E(gt) - \cup_{X|Y \in \mathcal{DS}'} E(gt_{X|Y}))$, where $\mathcal{DS}' = \mathcal{DS} - \mathcal{SBP}_{ST}(root(ST))$ and $\mathcal{DS}$ is the set of dominated subtree-bipartitions in gt with respect to ST. Now the following theorem directly follows.

Theorem 4.4.7. *Let gt be an unrooted, binary gene tree and ST be a rooted, binary species tree. Let $\mathcal{DS}$ be the set of dominated subtree-bipartitions in $\mathcal{SBP}_{gt}$ with respect to ST, and $\mathcal{DS}' = \mathcal{DS} - \mathcal{SBP}_{ST}(root(ST))$. Note that $\mathcal{DS} = \mathcal{DS}'$ when gt does not contain the root subtree-bipartition of ST. The tree gt^*, produced by rooting gt on an edge $e \in (E(gt) - \cup_{X|Y \in \mathcal{DS}'} E(gt_{X|Y}))$, is the optimal rooted version of gt under the MGD criterion (i.e., $Dup(gt^*, ST) \leq Dup(gt', ST)$, where gt' is any rooted version of gt).*

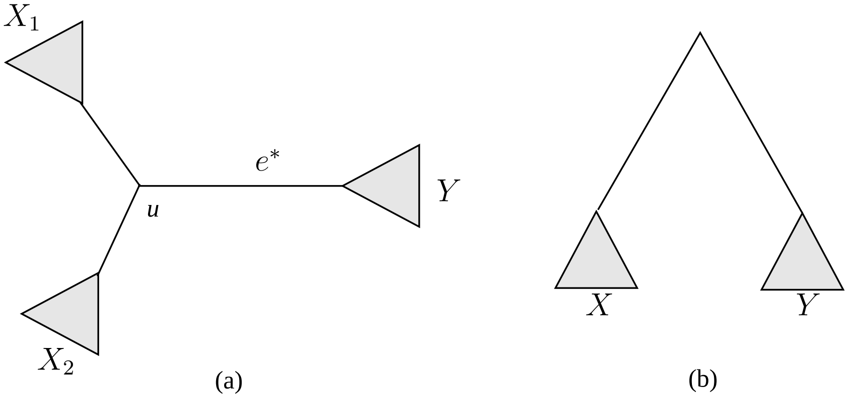


Figure 4.3: (a) An unrooted gene tree gt and (b) a rooted tree ST. Here $X = X_1 \cup X_2 \cup \{u\}$.

4.4.2 Solving MGD_{ru}

Given the extended definition of domination and Theorem 4.4.7, we can apply our clique based DP algorithms [6] (described in Chapter 3) to unrooted gene trees by modifying the weight calculation of a subtree-bipartition appropriately. For rooted gene trees, the weight of a subtree-bipartition $v = X|Y$ is defined as follows [6] (described in Chapter 3).

$$W_{dom}(v) = \sum_{gt \in \mathcal{G}} |\{bp : bp \in \mathcal{SBP}_{gt} \text{ and } dom(bp, v) = 1\}|$$

By definition, the set $\mathcal{SBP}_{gt}$ of subtree bipartitions in an unrooted tree gt contains the subtree bipartitions present in all possible rooted versions of gt. In MGD_{ru}, we count the number of duplications required to reconcile the optimal rooted version gt^* of gt. Therefore, we need to modify $W_{dom}(v)$ as it counts dominated subtrees in all possible rooted versions. By Lemma 4.4.3 and Lemma 4.4.4, a subtree-bipartition $v = X|Y$ that does not contain all the taxa of a gene tree gt cannot dominate more than one subtree-bipartition of an internal node in gt. According to Lemma 4.4.4, if the subtree-bipartition v contains all the taxa in gt and gt contains v, then v dominates three subtree-bipartitions of $\mathcal{SBP}_{gt}(u)$, where u is an internal node incident on the edge that defines v in gt. Using these results, we now modify $W_{dom}(v)$ for unrooted gene trees as follows.

For a subtree-bipartition $v = X|Y$, let $N_{gt}(v)$ denote the number of gene trees containing this subtree-bipartition v, and with $L(gt) = X \cup Y$. Note that, the later condition ($L(gt) = X \cup Y$) is due to the reason that gene trees may be incomplete ($L(gt) \subset L(ST)$). The corrected weight $W^*_{dom}(v)$ for unrooted gene trees can be defined as follows.

$$W^*_{dom}(v) = \begin{cases} W_{dom}(v) & \text{if both } X \text{ and } Y \text{ contain exactly one taxon,} \\ W_{dom}(v) - 4N_{gt}(v) & \text{if both } X \text{ and } Y \text{ contain more than one taxon,} \\ W_{dom}(v) - 2N_{gt}(v) & \text{if either } X \text{ or } Y \text{ contains more than one taxon.} \end{cases} \tag{4.2}$$

We now discuss these weight corrections with a simple example. Consider the unrooted tree gt shown in Fig. 4.4 (a), and the rooted species tree ST in Fig. 4.4 (b). Here $\mathcal{SBP}_{gt} = \{a|b, b|c, a|c, ab|c, bc|a, ac|b\}$, and $\mathcal{SBP}_{ST} = \{a|b, ab|c\}$. For each subtree-bipartition $v = X|Y \in \mathcal{SBP}_{ST}$, we need to find the number of dominated subtree-bipartitions in $\mathcal{SBP}_{gt}$. For $v = a|b$, there is only one subtree-bipartition ($a|b \in \mathcal{SBP}_{gt}$) which is dominated by v. Therefore, $W_{dom}(a|b) = 1$. Since both $X = \{a\}$, and $Y = \{b\}$ contain only one taxon, $W^*_{dom}(a|b) = W_{dom}(a|b) = 1$. For $v = ab|c$, $X = \{a, b\}$ and $Y = \{c\}$; and hence $N_{gt}(ab|c) = 1$ since gt contains this subtree-bipartition and $L(gt) = \{a, b, c\}$. It is easy to see that three subtree-bipartitions ($\{a|c, b|c, ab|c\}$) in $\mathcal{SBP}_{gt}$ are dominated by $ab|c$. Therefore, $W_{dom}(ab|c) = 3$. However, a rooted version of gt cannot contain all these three subtree-bipartitions. The optimal rooted version $gt^* = ((a, b), c)$ contains $ab|c$ but does not contain $a|c$ and $b|c$. Therefore, according to Eqn. 4.2, $W^*_{dom}(ab|c) = W_{dom}(ab|c) - 2 * N_{gt}(ab|c) = 3 - 2 = 1$. Similarly, when both X and Y contain more than one taxon, we need to correct $W_{dom}(v)$ by subtracting $4 * N_{gt}(v)$ as defined in Eqn. 4.2.

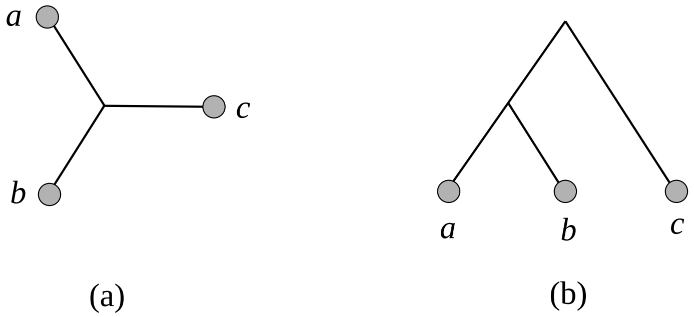


Figure 4.4: (a) An unrooted gene tree gt, (b) a rooted binary species tree ST.

Therefore, we can compute the score of any candidate species tree with respect to a set of unrooted gene trees as follows, and thus our DP technique can still be applied to unrooted gene trees.

Theorem 4.4.8. *Let ST be a species tree and $\mathcal{G} = \{gt_1, gt_2, \ldots, gt_k\}$ be a set of unrooted, binary gene trees with at most single copy per species. Let $\mathcal{G}^* = \{gt_1^*, gt_2^*, \ldots, gt_k^*\}$ be a set of binary gene trees such that gt_i^* is an optimally rooted version of gt_i that minimizes the number of duplications with respect to ST. Then $Dup(\mathcal{G}^*, ST) = \sum_{i=1}^{k} |V_{int}(gt_i^*)| - \sum_{v \in \mathcal{SBP}_{ST}} W_{dom}^*(v)$. Furthermore, the optimal $\mathcal{G}^*$ can be computed in $O(nk)$ time.*

Therefore, we have the following:

Theorem 4.4.9. *Let $\mathcal{G} = \{gt_1, gt_2, \ldots, gt_k\}$ be a set of unrooted binary gene trees on the n taxa in $\mathcal{X}$ (with at most single copy per species). Let $CG(\mathcal{G})$ be the compatibility graph with vertex weights defined by $W_{dom}^*(v)$. Let $\mathcal{C}$ be an $(n-1)$-clique in $CG(\mathcal{G})$ maximizing $W_{dom}^*(\mathcal{C})$, and let ST be the species tree defined by the clique (so that $\mathcal{SBP}_{ST}$ corresponds to $\mathcal{C}$). Then ST is a binary species tree that optimizes MGD_{ru} with respect to $\mathcal{G}$.*

Proof. The proof is similar to that of Theorem 3.4.1 in Chapter 3. □

For any set $\mathcal{SBP}$ of subtree-bipartition, we can define the constrained version of MGD_r by limiting the solution space such that $\mathcal{SBP}(ST) \in \mathcal{SBP}$ (in a similar way which we discussed in Chapter 3). We can obtain the exact solution by setting $\mathcal{SBP}$ to be the set of all possible subtree-bipartitions.

4.4.3 Extension to $MGDL_{ru}$

We now describe $MGDL_{ru}$ problem. The input to this problem is a set $\mathcal{G} = \{gt_1, gt_2, \ldots, gt_k\}$ of unrooted and binary gene trees such that $L(gt_i) \subseteq \mathcal{X}$, where $i \in 1, 2, \ldots, k$, and at most one copy per species is present in each of the gene trees. $MGDL_{ru}$ problem asks to find a rooted and binary species tree ST on $\mathcal{X}$ and set $\mathcal{G}^* = \{gt_1^*, gt_2^*, \ldots, gt_k^*\}$, where gt_i^* is a rooted version of gt_i such that $Duploss(\mathcal{G}^*, ST)$ is minimized.

Yu *et al.* [155] describes the optimal rooting of an unrooted gene tree gt with respect to a binary rooted species tree ST under the minimize deep coalescence (MDC) criterion. Let $C(gt)$ and $C(ST)$ be the set of all clusters in gt and ST, respectively. We say that a cluster A in gt is *ST-maximal* if there is a cluster $B \in C(ST)$ such that $B \neq \mathcal{X}$ and A is *B-maximal*. The optimal rooting under the MDC criterion can be obtained as described in the following theorem [155].

Theorem 4.4.10. *(From [155]) Let gt be an unrooted gene tree on $\mathcal{X}$ and ST be a species tree on $\mathcal{X}$. Let $\mathcal{C}^*$ be the set of ST-maximal clusters in gt. Let*

e be any edge of gt such that $\forall Y \in \mathcal{C}^*, e \notin E(gt_Y)$ (i.e., e is not inside any subtree of gt induced by one of the clusters in $\mathcal{C}^*$). Then the tree gt^* produced by rooting gt on edge e is the optimal rooted version of gt under MDC criteria. Furthermore, there is at least one such edge in gt.

We now prove the following theorem.

Theorem 4.4.11. *Let gt be an unrooted binary gene tree (single copy) and ST a species tree both on $\mathcal{X}$. There is an edge e such that the tree gt^*, produced by rooting gt on edge e, is the optimal rooted version of gt under both the MDC_r and MGD_r criteria.*

Proof. Let $\mathcal{C}^*$ be the set of ST-maximal clusters in gt and $\mathcal{DS}$ be the set of dominated subtree-bipartitions in DS. Let $\mathcal{DS}' = \mathcal{DS} - \mathcal{SBP}_{ST}(root(ST))$. Let e be any edge of gt such that $\forall Y \in \mathcal{C}^*, e \notin E(gt_y)$ (i.e., e is not inside any subtree of gt induced by one of the clusters in $\mathcal{C}^*$). According to Theorem 4.4.10, the tree gt^* produced by rooting gt on edge e is optimal under the MDC criterion. We will argue that gt^* is also an optimal rooted version of gt under the MGD criterion. Let S_1 be a subtree-bipartition in ST and a subtree-bipartition S_2 in gt is the maximal subtree-bipartition in gt which is dominated by S_1. We denote by $Cluster(S)$ the cluster associated with a subtree bipartition S (i.e., $Cluster(S)$ is the set of leaves in S). Then, from the definition of domination, it is clear that $Cluster(S_2) \subseteq Cluster(S_1)$. It follows that $Cluster(S_2) \subseteq cl^*$, where $cl^* \in \mathcal{C}^*$. Therefore, for any subtree-bipartition $S \in \mathcal{DS}', Cluster(S) \subseteq cl^*$ where cl^* is a cluster in $\mathcal{C}^*$. Hence,

$e \in E(gt) - \cup_{X|Y \in \mathcal{DS}'} E(gt_{X|Y})$. Therefore, according to Theorem 4.4.7, gt^* is an optimal rooted version of gt under the MGD criterion. Furthermore, by Theorem 4.4.10, there is at least one such edge in gt.

□

Theorem 4.4.12. *(From Zhang [158]) Let gt be a rooted binary gene tree and ST a rooted binary species tree such that $L(gt) \subseteq L(ST)$. Then, under the restriction-based analysis, $Duploss(gt, ST) = MDC(gt, ST) + 3 * Dup(gt, ST)$.*

From Theorem 4.4.11 and Theorem 4.4.12, we now have the following theorem.

Theorem 4.4.13. *Let gt be an unrooted gene tree and ST a species tree both on $\mathcal{X}$. There is an edge e such that the tree gt^*, produced by rooting gt on edge e, is the optimal rooted version of gt under all three optimization criteria (MDC_r, MGD_r and $MGDL_r$).*

4.4.4 Solving $MGDL_{ru}$

We have already shown how to solve MGD_{ru}. Yu *et al.* showed how to solve MDC for unrooted gene trees [155] and showed that the score function remains unchanged when we consider unrooted gene trees instead of rooted (see corollary 2 of [155]). As a result we can use all the techniques used for $MGDL_r$ (see Chapter 3) with necessary modifications in the weight calculation as described in Sec. 4.4.2.

Theorem 4.4.14. *Let $\mathcal{G} = \{gt_1, gt_2, \ldots, gt_k\}$ be a set of binary rooted gene trees on set $\mathcal{X}$ of n species, and let $CG(\mathcal{G})$ be the compatibility graph with vertex weights defined by $W^*_{MGDL}(v) = W_{xl}(v) - 3W^*_{dom}(v)$, where $W_{xl}(v)$ is as defined in [6], and described in Section 3.5 in Chapter 3. The set of bipartitions in an $(n-1)$-clique of minimum weight in $CG(\mathcal{G})$ defines a binary species tree ST that optimizes $MGDL_{ru}$.*

Proof. The proof is similar to that of Theorem 3.5.4 in Chapter 3. □

Advantage over Hallett and Lagergren [50] Hallett and Lagergren [50] solved MGD and MGDL for unrooted gene trees (allowing the gene trees to have multiple copies of the taxa, and even to miss some taxa) using a similar DP based approach. They choose a set $\mathcal{P}$ of partitions (similar to our concept of subtree-bipartition) from input gene trees (see [50] for details), and then find the optimal species tree for MGD and MGDL subject to the constraint that $\mathcal{P}_{\mathcal{S}} \subseteq \mathcal{P}$, where $\mathcal{P}_{\mathcal{S}}$ is the set of partitions in S. The running time of their algorithm is $O(p^2ral^3)$, where $p = |\mathcal{P}|$, l is the size of the leafset, r is the number of gene trees and a is the time needed to access a set $A \in \mathcal{P}$ (Theorem 1 in [50]). If $\mathcal{P}$ contains only the partitions from k input gene trees on n taxa (identical to the constrained version we consider), their running time becomes $O(n^5k^3a)$, since $p = O(nk)$ and $r = k$. But, as we showed in Theorem 4.3.2, running time of our method is $O(n^3k^2a)$ saving a factor of $O(n^2k)$.

4.5 Conclusion

Phylogenetic methods for estimating species trees from a collection of gene trees often assume that the gene trees are rooted. However, estimated gene trees are often unrooted and rooting the gene trees correctly is very difficult. Estimating species tree by summarizing gene trees considering the reasons for gene tree discordance is a difficult task. Absence of rooted gene trees makes the inference of rooted binary species tree even more complicated. In this chapter we proposed exact and heuristic algorithms for solving MGD and MGDL problems for the cases where gene trees can be unrooted. We showed how to root a gene tree with respect to a species tree by minimizing gene duplication and loss. Next we showed how to estimate a binary, rooted species tree from a collection of gene trees under MGD and MGDL criteria. We proposed exact solutions as well as constrained versions where the solution space is defined by the input gene trees.

Chapter 5

Gene Tree Parsimony for Incomplete Gene Trees under Gene Duplication and Loss

Species tree estimation from gene trees can be complicated by gene duplication and loss, and gene tree parsimony (GTP) is one approach for estimating species trees from multiple gene trees. In its standard formulation, the objective is to find a species tree that minimizes the total number of gene duplications and losses with respect to the input set of gene trees. Although much is known about GTP, little is known about how to treat inputs containing some *incomplete gene trees* (i.e., gene trees lacking one or more of the species). In this chapter, we present new theory for GTP when incompleteness results from gene birth and death (i.e., true biological loss), and a dynamic programming algorithm that can be used for an exact solution for small numbers of taxa, or as a heuristic for larger numbers of taxa. We also prove that the "standard" calculations for duplications and losses exactly solve GTP when incompleteness results from taxon sampling, although they can be incorrect when incompleteness results from true biological loss.

The results presented in this chapter are not published yet. Md. Shamsuzzoha Bayzid designed the study, developed the algorithms and proved all the theoretical results. Bayzid and Tandy Warnow wrote this chapter. Siavash Mirarab implemented the algorithms as a software tool.

5.1 Introduction

The estimation of species trees is often performed by estimating multiple sequence alignments for some collection of genes, concatenating these alignments into one super-matrix, and then estimating a tree (often using maximum likelihood or a Bayesian technique) on the resultant super-matrix. However, this approach cannot be used when the species' genomes contain multiple copies of some gene, which can result from gene duplication. Since gene duplication and loss is a common phenomenon, the estimation of species trees requires a different type of approach in this case.

Gene Tree Parsimony (GTP) is an optimization problem for estimating species trees from multiple gene trees. In its most typical formulations, only gene duplication and loss are considered, so that GTP depends upon two parameters: c_d (the cost for a duplication) and c_l (the cost for a loss). The two most popular versions of GTP are MGD (minimize gene duplication), for which $c_d = 1$ and $c_l = 0$, and MGDL (minimize gene duplication and loss), for which $c_d = c_l = 1$. The version of GTP which seeks the tree minimizing the total number of losses has also been studied; for this, $c_d = 0$ and $c_l = 1$. These variants of GTP are NP-hard optimization problems [83], but software such as DupTree [150] and iGTP [15] for GTP are in wide use.

Basic to all these problems is the ability to compute the number of duplications and losses implied by a species tree and gene tree. This problem is called the reconciliation problem, surveyed in [32], and intensively studied in the literature (see, for example, [43, 44, 47, 50, 83, 95, 103, 105, 107, 132, 157, 158]). The mathematical formulation of the reconciliation problem was derived for the case where the gene tree and the species tree have the same set of taxa, and then extended to be able to be used on *incomplete* gene trees, i.e., trees that can miss some taxa.

Incomplete gene trees are quite common, and can arise for two different reasons: (1) *taxon sampling*: the gene may be available in the species' genome, but the biologist did not sample it when he/she estimated the gene tree, or (2) *gene birth/death*: as a result of gene birth and death (true biological gene loss), the species does not have the gene in its genome.

Given an incomplete gene tree gt and a species tree ST, two formulations for the number of losses have been defined. One, described in [16, 145], correctly computes the number of losses when incompleteness is a result of true gene loss, as we will prove. The other, and most commonly used one, computes the number of losses by first computing the homeomorphic subtree $ST(gt)$ of ST induced by gt, and then computing the number of losses required to reconcile gt with $ST(gt)$ (see, for example, [47, 83, 158]). Although this second formulation (described in Chapter 2) is in wide use (and is the basis of both iGTP [15] and Duptree [150], two popular methods for "solving" GTP), the theoretical basis for this approach has not yet been established. We refer to this second formulation as the "standard" approach because of this widespread use in both software and the theoretical literature on GTP.

This chapter addresses the GTP problem for the case where some of the input gene trees may be incomplete due to either taxon sampling strategies or true biological loss. The main results are as follows:

- We formalize the duploss reconciliation problem when gene trees are incomplete due to taxon sampling as the "optimal completion of a gene tree" (Section 5.2.4), and we prove (Theorem 5.2.1) that the standard calculation correctly computes losses for this case.

- We show by example that the standard calculation for losses in GTP can be incorrect when incompleteness is due to true biological loss (Section 5.2.5).

- We show how to compute the number of losses implied by a gene tree and species tree, when incompleteness is due to true biological loss (Section 5.3).

- We formulate variants of the GTP problem (when gene tree incompleteness is due to true biological loss) as minimum weight maximum clique problems (see Theorem 5.4.10 for one duploss variant), and show how to solve these problems efficiently using dynamic programming (Section 5.4).

5.2 Basics

5.2.1 Notation and terminology

Throughout this chapter we will assume that gene trees and species trees are rooted binary trees, and we let gt denote a gene tree (with any number of copies of each taxon) and ST denote a species tree. We let $L(t)$

denote the set of taxa at the leaves of t, and require that $L(gt) \subseteq L(ST)$. We let n denote the number of leaves in gt and n' denote the number of leaves in ST; note that we cannot infer any relationship about the relative magnitudes of n and n'. If $L(gt) = L(ST)$ we say that gt is complete, and otherwise we say that gt is incomplete. Given a node u in a rooted binary tree, we let r denote the right child of u and l denote the left child of u.

$ST(gt)$ and $ST^*(gt)$. $ST(gt)$ is the homeomorphic subtree of ST induced by the taxon set of gt, and is produced as follows: ST is restricted to the taxon set of gt, and then nodes with in-degree and out-degree 1 are suppressed. $ST^*(gt)$ is the tree obtained by restricting ST to the taxon set of gt, but not suppressing nodes of in-degree and out-degree 1.

Maximal missing clades, UMMC, and LMMC. We say that clade cl in ST is a *missing clade* with respect to gt if $L(gt) \cap L(cl) = \emptyset$, and a *maximal missing clade* if it is not contained in any other missing clade. Maximal missing clades that lie below $M(r(gt))$ are called the "lower" maximal missing clades, and those that do not lie below $M(r(gt))$ are called the "upper" maximal missing clades. We denote by $LMMC(gt, ST)$ (or $LMMC$), the set of lower maximal missing clades, and $UMMC(gt, ST)$ (or $UMMC$), the set of upper maximal missing clades. Note $UMMC(gt, ST) = \emptyset$ iff $M(r(gt)) = r(ST)$.

5.2.2 Reconciliation of gene trees and species trees

There are several equivalent definitions of reconciliation of gene trees and species trees in the presence of gene duplication and loss; of these, we find the definition based on "DS-trees" the easiest to understand, and so our discussion is based on this. *DS-trees* are gene trees that explain the evolution of the gene in terms only of gene duplications and speciations; thus, no gene losses are needed in the evolutionary history (see, for example, [16]).

DS-trees. We say that T is a DS-tree for the species tree ST, or equivalently that T is *DS-consistent* with ST, iff for every internal vertex v of T (with children l and r), there exists a vertex v' of ST (with children l' and r') such that $L(T_v) = L(ST_{v'})$ and one of the following conditions holds:

- $L(T_r) = L(T_l)$, in which case v is said to be a duplication node, or
- $L(T_r) = L(ST_{r'})$ and $L(T_l) = L(ST_{l'})$ (or the condition obtained by swapping r' and l'), in which case v is said to be a speciation node.

How DS-trees define evolutionary scenarios. A *subtree insertion* [16] in a tree t is obtained by grafting a new subtree onto an existing branch of t, and a tree t' is said to be an *extension of* t if it can be obtained from t by a sequence of subtree insertions. A *reconciliation* between a gene tree gt and a species tree ST is an extension T of gt that is DS-consistent with ST. Note that in a DS-tree T, every node is labelled either as a speciation node or a duplication node. The creation of the tree T from gt allows us to identify the specific leaves in T that correspond to each leaf in gt (despite the fact that gt can be multi-copy), and thus define the MRCA mapping from the internal nodes of gt to the internal nodes of T; therefore, we can identify the duplication nodes in gt. Finally, the gene losses implied by this reconciliation are defined by the subtree insertions used to create T from gene tree gt. Thus, the number of duplications in this reconciliation is the number of duplication nodes in T, and the number of gene losses is the number of subtree insertions. Given a gene tree gt, species tree ST, and DS-tree T that is a reconciliation for gt with ST, there is a mapping from the nodes of gt to ST defined by first mapping each node v to the MRCA v' in T of $L(gt_v)$, and then mapping v' to the MRCA v'' in ST of $L(T_{v'})$. Thus, a reconciliation defines a mapping from the nodes of the gene tree to the nodes of the species tree. For this reason, in the literature a reconciliation is identified by a mapping.

5.2.3 The standard formula for computing losses

The most commonly used approach of reconciliation and calculating losses, which we call the *standard* approach, is based upon the homeomorphic subtree of the species tree. The *standard* formula (see, for example, [6, 44, 47, 83, 158]) for computing the minimum number of losses of a (potentially incomplete) gene tree gt with respect to a species tree ST is discussed in Chapter 2. Due to its relevance to this chapter and for our convenience, we define the standard loss, denoted by $L_{std}(gt, ST)$, as follows.

$$L_{std}(gt, ST) = \sum_{u \in V_{int}(gt)} F(u, ST(gt)),$$

where $F(u,T)$ is defined for internal nodes u with children l and r (which can be interchanged in the formula below) by:

$$F(u,T) = \begin{cases} d(\mathcal{M}(r), \mathcal{M}(u)) + 1 & \text{if } \mathcal{M}(r) \neq \mathcal{M}(u) \ \& \\ & \mathcal{M}(l) = \mathcal{M}(u), \\ d(\mathcal{M}(l), \mathcal{M}(u)) + 1 & \text{if } \mathcal{M}(l) \neq \mathcal{M}(u) \ \& \\ & \mathcal{M}(r) = \mathcal{M}(u), \\ d(\mathcal{M}(r), \mathcal{M}(u)) & \\ +d(\mathcal{M}(l), \mathcal{M}(u)) & \text{if } \mathcal{M}(r) \neq \mathcal{M}(u) \ \& \\ & \mathcal{M}(l) \neq \mathcal{M}(u), \\ 0 & \text{if } \mathcal{M}(r) = \mathcal{M}(l) = \\ & \mathcal{M}(u). \end{cases} \tag{5.1}$$

where $d(s, s')$ is the number of internal nodes in T on the path from s to s'. When gt is complete, then $ST(gt) = ST$, and this formula follows from [16]. Note that Eqn. 5.1 is equivalent to Eqn. 2.2.

5.2.4 Incompleteness due to sampling strategies

When gene trees are incomplete due to taxon sampling strategies used to define the gene dataset rather than due to true biological loss, a natural optimization problem is to add the missing taxa into each gene tree so as to produce complete gene trees, and then try to estimate the species tree. This yields two problems: one for reconciling (and hence scoring) a gene tree gt with a species tree ST, and the other for finding an optimal species tree from a set of incomplete gene trees. Since the second problem depends upon the first, we discuss the first problem:

Optimal completion of a gene tree:

- Input: rooted binary gene tree gt and rooted binary species tree with $L(gt) \subseteq L(ST)$.
- Output: complete gene tree $T_{samp}(gt, ST)$ that is an extension of gt such that $T_{samp}(gt, ST)$ implies a minimum number of losses with respect to ST.

In other words, we add all the missing taxa into gt (each taxon added at least once, but perhaps several times) so as to produce a complete binary gene tree that has a minimum number of losses with respect to ST. Let $L_{samp}(gt, ST) = L_{std}(T_{samp}(gt, ST), ST)$. Thus, $L_{samp}(gt, ST)$ denotes the total number of losses needed for an optimal completion of gt. Similarly, we can define $DL_{samp}(gt, ST)$ to be the total number of duplications and losses needed for a completion of gt that minimizes the duploss score.

Theorem 5.2.1. *Given a binary rooted gene tree gt and a binary rooted species tree ST such that $L(gt) \subseteq L(ST)$, the MRCA mapping defines a reconciliation that minimizes the number of duplications, the number of losses, and hence also the total number of duplications and losses, where we treat losses as due to taxon sampling strategies. Furthermore, $L_{std}(gt, ST) = L_{samp}(gt, ST)$, which means the standard formula correctly computes the number of losses when we treat incompleteness as due to taxon sampling strategies.*

Proof. Consider $ST(gt)$, the homeomorphic subtree of ST defined by the taxon set of gt. Since gt is complete with respect to $ST(gt)$, the optimal reconciliation that minimizes duplications, losses, and their sum, is defined by M, the MRCA mapping from gt to ST, and the standard formula correctly computes the number of losses for this reconciliation [16]. Note that for any completion t of gt, $L_{std}(t, ST) \geq L_{std}(gt, ST)$; in other words, the number of losses cannot decrease by making gt complete. Similarly, the number of duplications for t with respect to ST cannot be less than the number of duplications of gt with respect to ST. We will show we can add all the remaining taxa into gt to produce a complete gene tree t^* such that $L_{std}(t^*, ST) = L_{std}(gt, ST)$ and so that t^* has the same number of losses and duplications with respect to ST as t^* has. Therefore, t^* will be an optimal completion. Furthermore, this will also imply that $L_{std}(gt, ST) = L_{samp}(gt, ST)$, as desired.

Recall the definition of the sets $UMMC$ and $LMMC$, the upper and lower maximal missing clades, respectively. Since gt is not complete, there must be at least one missing taxon, and hence at least one maximal missing clade. If $M(r(gt)) = r(ST)$ then $UMMC = \emptyset$ and we set $gt' = gt$. Otherwise, $M(r(gt)) \neq r(ST)$ and $UMMC \neq \emptyset$. Consider the path in ST from $r(ST)$ down to $M(r(gt))$, and the $m \geq 1$ subtrees that hang off that path before we reach $M(r(gt))$. Note that each of these subtrees is an upper maximal missing clade. Let gt' be the tree created by starting with ST and replacing the subtree of ST rooted at $M(r(gt))$ by gt. Note also that the number of duplications has not changed, and that $L_{std}(gt', ST) = L_{std}(gt, ST)$.

If $LMMC = \emptyset$ we are done; otherwise, we now add the lower maximal missing clades to gt' one at a time. Let $LMMC = \{t_1, t_2, \ldots, t_p\}$, so that

$p \geq 1$. We will define a sequence of gene trees $gt_1, gt_2, \ldots, gt_p = t^*$, so that gt_1 is the result of adding clade t_1 to gt', and gt_i is the result of adding clade t_i to gt_{i-1} for $p \geq i \geq 2$. We will show that $L_{std}(gt_i, ST) = L_{std}(gt', ST)$ for $p \geq i \geq 2$, and that the number of duplications in gt_i is the same as the number of duplications in gt'. Since $gt_p = t^*$ is a completion of gt, our theorem will be proven.

So consider $t = t_1$, the first lower maximal missing clade, and let q be the node in ST that is the parent of $r(t)$ (i.e., $q = p(r(t))$). Consider the edges (x, y) in gt' with $y = p(x)$, such that q lies in the path between $M(x)$ and $M(y)$. Subdivide each such edge (creating a new node), and add t to gt' by making its root the child of each such newly created node. Note that there must be at least one such edge in gt' but there can be several such edges, and hence this step adds t at least once (and perhaps several times) to gt'. Note that when we add t_1 to gt', we do not change the image under the MRCA mapping for any node v that is in gt'.

We now show that t_1 has the same number of duplications as gt with respect to ST. Clearly, any node in a copy of t is a speciation node (since t is a subtree of ST, which only has speciation nodes). Now consider a node u created by subdividing an edge (x, y), where y is the parent of x in gt'. One child of u is the root of t and the other child has an entirely disjoint leaf set; thus u is a speciation node. When we subdivide edge (x, y) we make y the parent of u. Therefore, $M(u) \neq M(y)$. Thus, y is a duplication node in gt_1 if and only if $M(z) = M(y)$ where z is the other child of y in gt'.

But then y is a duplication node in gt' if and only if y is a duplication node in gt_1, since the MRCA mapping does not change. Hence, no node in gt' changes duplication/speciation status, and the newly added nodes are always speciation nodes. Therefore the number of duplication nodes does not change.

We now show that the number of losses does not change, i.e., $L_{std}(gt', ST) = L_{std}(gt_1, ST)$. Now consider an edge (x, y) that is subdivided through the addition of a node u that is made the parent of the subtree t_1. Then x, y, and u all map (under M) to different vertices in $ST(gt_1)$. Also, a simple case analysis (using the standard formula) verifies that $F(y, ST(gt')) = F(y, ST(gt_1)) + F(u, ST(gt_1))$. Since $F(z, ST(gt')) = F(z, ST(gt_1))$ for all other vertices $z \in V(gt')$, this means that the total number of losses does not change.

Therefore, the result of adding each lower maximal missing clade to gt' does not imply any additional losses nor any additional duplications, and so also the total number of duplications and losses does not change. Let $t^* = t_p$ be the tree obtained after adding in all the missing maximal clades, and return t^*. The result then follows by induction on p.

□

5.2.5 Incompleteness due to gene birth and death

As we will see, while the MRCA mapping is still an optimal reconciliation when gene trees are incomplete due to gene birth and death, the standard formula does not correctly compute the number of losses.

We begin by summarizing some results that have already been established:

Theorem 5.2.2. *(From [16, 45]) Given a binary rooted gene tree gt and a binary rooted species tree ST such that $L(gt) \subseteq L(ST)$, the MRCA mapping defines a reconciliation that minimizes the number of duplications and the number of losses where we treat losses as due to gene birth and death. The set of speciation nodes in gt are those vertices $v \in V_{int}(gt)$ that satisfy $M(v) \notin \{M(l), M(r)\}$, where l and r are the two children of v and M is the MRCA mapping from gt to ST; all other nodes are duplication nodes. Furthermore, we can compute the MRCA mapping, the set of duplication nodes, and the set of speciation nodes, in $O(n + n')$ time, where ST has n leaves and gt has n' leaves.*

Proof. Chauve *et al.* [16] proved that the MRCA mapping minimizes the losses required to reconcile gt with $ST(gt)$ for complete gene trees, but the proof also applies to incomplete gene trees, treating incompleteness as due to gene birth and death. Górecki *et al.* [45] showed that the MRCA mapping minimizes the number of duplications required to reconcile gt with $ST(gt)$, treating incompleteness as due to gene birth and death. Therefore, the MRCA mapping is optimal for all three scores (number of duplications, number of losses, and number of duplications plus losses), when treating incompleteness as due to gene birth and death.

It is easy to see that the duplication nodes in gt are those nodes that have $M(v) = M(l)$ or $M(v) = M(r)$ (where l and r are the two children of

v, and M is the MRCA mapping), and all other nodes are speciation nodes. Since the MRCA mapping M can be computed in $O(n + n')$, where ST has n leaves and gt has n' leaves, it follows that all these can be computed in $O(n + n')$ time. □

However, the standard calculation for the number of losses can be incorrect when incompleteness is due to true biological loss! Consider the simple example $gt = ((a, b), c)$ and $ST = ((a, (b, d)), c)$. Under the standard formula, $L_{std}(gt, ST) = 0$, since $ST(gt) = gt$. Under the assumption that incompleteness is due to true biological loss, the genome for d does not have the gene. Because d is sister to b and all the other taxa have the gene, the gene must have been present in the parent of d, and lost on the branch leading to d. *Therefore, the standard formula for the number of losses can be incorrect when gene trees are incomplete due to gene birth and death (i.e., true biological loss).*

5.3 How to calculate losses

We now show how to calculate the number of losses for an incomplete gene tree gt and species tree ST, treating incomplete gene trees as due to gene birth and death. How this is defined will depend upon whether one assumes, *a priori*, that the gene is present in the genome of the common ancestor of the species in ST (i.e., at the root of ST). Thus, this section shows how to calculate the following values:

- $L^*_{bd}(gt, ST)$, the minimum number of losses, under the assumption the gene is present in the common ancestor of the species in ST ($DL^*_{bd}(gt, ST)$ is defined similarly for the total number of duplications and losses), and

- $L_{bd}(gt, ST)$ the minimum number of losses *without* assuming the gene is present in the common ancestor of the species in ST ($DL_{bd}(gt, ST)$ is defined similarly for duplications and losses).

Because the MRCA mapping is optimal for duplications and also for losses (and hence also for their sum) when interpreting missing taxa as due to gene birth and death (as shown in Theorem 5.2.2), the reconciliation that optimizes $L_{bd}(gt, ST)$ will optimize $DL_{bd}(gt, ST)$, $L^*_{bd}(gt, ST)$, and $DL^*_{bd}(gt, ST)$. Therefore, we will focus on how to compute the *number* of losses (i.e., $L_{bd}(gt, ST)$ and $L^*_{bd}(gt, ST)$), using the fact that the MRCA mapping defines an optimal reconciliation.

Theorem 5.3.1. *Let gt be a gene tree and ST a species tree such that $L(gt) \subseteq L(ST)$. Then,*

$$L_{bd}(gt, ST) = \sum_{u \in V_{int}(gt)} F(u, ST),$$

and

$$L^*_{bd}(gt, ST) = L_{bd}(gt, ST) + |UMMC(gt, ST)|.$$

Furthermore, these values can be calculated in $O(n + n')$ time, where ST has n leaves and gt has n' leaves.

Proof. Note that we use a modification of the standard formula, $F(u, ST)$, so that we do not replace ST by $ST(gt)$ as was done in [16, 145].

Derivation of $L_{bd}(gt, ST)$. Recall that $L_{bd}(gt, ST)$ does not assume that the most recent common ancestor of the species in ST has the gene. Since gene birth can happen only once (although loss can happen repeatedly), we begin by determining the location of the gene birth. If $M(r(gt)) = r(ST)$, then the gene is born before $r(ST)$, and is present at the root of ST. Otherwise, it is easy to see that the location of the gene birth that minimizes the number of losses is the edge above $M(r(gt))$. Now consider the modification of the standard formula (i.e., using ST instead of $ST(gt)$):

$$L_{bd}(gt, ST) = \sum_{u \in V_{int}(gt)} F(u, ST). \tag{5.2}$$

It is easy to see that this correctly returns the number of inserted subtrees, and hence the number of losses.

Derivation of $L^*_{bd}(gt, ST)$. By definition of $L^*_{bd}(gt, ST)$, the gene is assumed to be present at the root of the species tree ST. If $M(r(gt)) = r(ST)$, then $UMMC(gt, ST) = \emptyset$, and the result follows. However, if $M(r(gt)) \neq r(ST)$, the gene must be present on the path between $r(ST)$ and $M(r(gt))$. Since the gene is not present in any leaf that is not below $M(r(gt))$, to minimize losses, the gene must be lost on every edge off that path, since such edges lead to subtrees that do not have the gene present in any leaf. Note that if $M(r(gt)) \neq r(ST)$, then the number of edges that lead off that path is $|UMMC(gt, ST)| = d(M(r(gt)), r(ST)) + 1$. Since the gene must be lost on each of those edges, and the total number of losses is the sum of this value and the number of losses that occur within the subtree rooted at $M(r(gt))$, it follows that

$$L^*_{bd}(gt, ST) = L_{bd}(gt, ST) + |UMMC(gt, ST)|. \tag{5.3}$$

The running time follows easily from the fact that the MRCA mapping can be computed in linear time [40]. □

Figure 5.1 illustrates an example distinguishing $L_{bd}(gt, ST)$ and $L^*_{bd}(gt, ST)$.

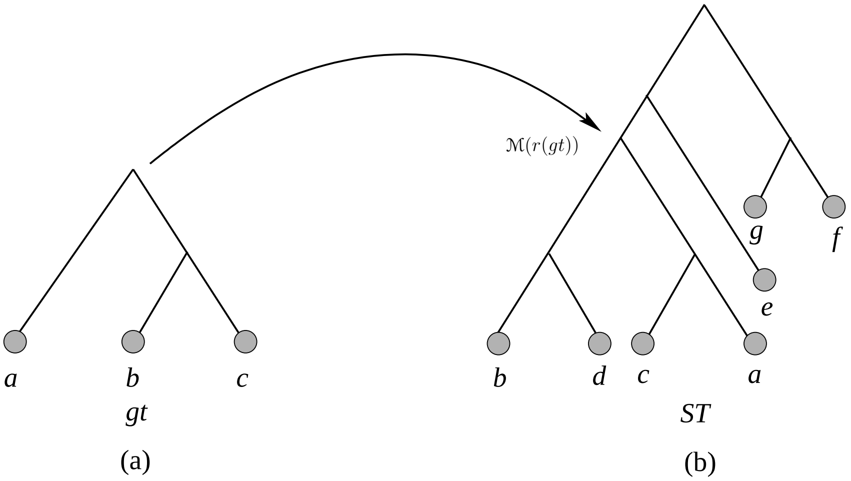


Figure 5.1: (a) A gene tree $gt = ((b, c), a)$, (b) a species tree $ST = ((((a, c), (b, d)), e), (f, g))$. Here, $\mathcal{M}(r(gt)) \neq r(ST)$, and $UMMC(gt, ST) = \{\{e\}, \{f, g\}\}$. $L_{bd}(gt, ST)$ is the number of losses required to reconcile gt with ST according to the Eqn. 5.2, and we get $L^*_{bd}(gt, ST)$ by adding $|UMMC(gt, ST)|$ to $L_{bd}(gt, ST)$.

Now the most important question in terms of estimating the optimal species tree is – given a set $\mathcal{G}$ of (possibly incomplete) gene trees, is the species tree that minimizes $\sum_{gt \in \mathcal{G}} L^*_{bd}(gt, ST)$ or $\sum_{gt \in \mathcal{G}} L_{bd}(gt, ST)$ is different than the one that minimizes $\sum_{gt \in \mathcal{G}} L_{std}(gt, ST)$? If the same species tree optimizes both ways of calculating losses, then defining loss differently is not that important in the context of phylogenomic analyses. But this is not necessarily true as we will show in the following theorem.

Theorem 5.3.2. *Let $\mathcal{G}$ be a set of incomplete gene trees and ST_{bd}, ST^*_{bd} and ST_{std} are the species trees that minimizes $\sum_{gt \in \mathcal{G}} L_{bd}(gt, ST)$, $\sum_{gt \in \mathcal{G}} L^*_{bd}(gt, ST)$ and $\sum_{gt \in \mathcal{G}} L_{std}(gt, ST)$, respectively. Then ST_{std} is not necessarily identical to ST_{bd} or ST^*_{bd}.*

Proof. We prove this by showing an example. Consider the two gene tree topologies tp_1 and tp_2 as shown in Fig. 5.2(a) and Fig. 5.2(b). Let $\mathcal{G}$ be a set of 14 gene trees, with 8 gene trees having topology tp_1 and the rest 6 gene trees having topology tp_2. It is easy to verify that the species tree with topology tp_2 minimizes $\sum_{gt \in \mathcal{G}} L_{std}(gt, ST)$. Here, $\sum_{gt \in \mathcal{G}} L_{std}(gt, tp_2) = 8 * 3 + 6 * 0 = 24$. Any other species tree will result into more than 24 losses. The reason is as follows. There are three tree topologies on leaf-est $\{a, b, c\}$: $((a, b), c)$, $((a, c), b)$ and $((b, c), a)$. Reconciling tp_1 with $((a, c), b)$ or $((b, c), a)$ requires 3 losses. Therefore, any species tree ST, such that $ST(tp_1)$ is not identical to $tp_1 = ((a, b), c)$, requires $8 * 3 = 24$ losses to reconcile the eight gene trees (having topology tp_1) with ST. Therefore, to achieve less than 24 losses, $ST(tp_1)$ should be identical to tp_1. We now estimate the number of losses required to reconcile tp_2 with a ST such that $ST(tp_1) = ((a, b), c)$. Note that, $tp_2(tp_1) = ((a, c), b)$. Reconciling $((a, c), b)$ with $((a, b), c)$ requires 3 losses. Then taking

$\{d, e, f\}$ into consideration, it is quite easy to verify that it requires more than 4 losses to reconcile tp_2 with a species tree ST such that $ST(tp_1) = ((a, b), c)$. Hence, there is no species tree ST so that $\sum_{gt \in \mathcal{G}} L_{std}(gt, ST) < 24$. Therefore, $ST = tp_2$ minimizes $\sum_{gt \in \mathcal{G}} L_{std}(gt, ST)$. However, the species tree $ST = (((((a, b), c), d), e), f)$ minimizes $\sum_{gt \in \mathcal{G}} L_{bd}(gt, ST)$. Here $\sum_{gt \in \mathcal{G}} L_{bd}(gt, ST) = 8 * 3 + 6 * 6 = 60$, which is less than $\sum_{gt \in \mathcal{G}} L_{bd}(gt, tp_2) = 8 * 9 + 6 * 0 = 72$. Therefore, ST_{std} is not necessarily same as ST_{bd}. Then the fact that ST_{std} is not necessarily identical to ST^*_{bd} immediately follows. □

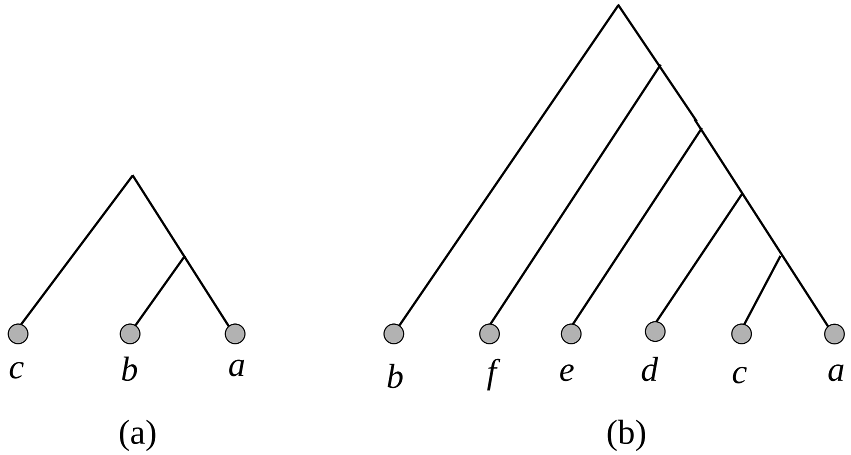


Figure 5.2: (a) Gene tree topology tp_1, and (b) gene tree topology tp_2.

5.4 Algorithms to find species trees

Here we address the problem of finding a species tree that has a minimum total number of duplications and losses, treating incompleteness as due to true biological loss. Prior results on this problem include a branch-and-bound algorithm for this problem in [31], based on techniques from [16]. However, the branch-and-bound algorithm in [31] cannot be used on even moderate-sized datasets.

In this section, we derive a different approach for the GTP problems, treating incomplete gene trees as due to true biological loss (i.e., minimizing $L_{bd}(gt, ST)$ or $L^*_{bd}(gt, ST)$). The techniques we propose can be used to solve GTP exactly for small datasets, or approximately (though without any guaranteed error bounds) on larger datasets. The approach we take here is based on Chapter 3 [6] (see also [50, 141, 154, 155], which use very similar techniques). Bayzid *et al.* [6] provided a graph-theoretic formulation for $MGDL_{std}$, whereby an optimal solution to $MGDL_{std}$ corresponded to finding a minimum weight maximum clique inside a graph called the compatibility graph. The nodes of the compatibility graph correspond to subtree-bipartitions. Bayzid *et al.* [6] showed how to find a minimum weight max clique using a dynamic programming approach. We will use the same graph-theoretic formulation as in [6], but modify the weights appropriately, to show that the optimal solution to $MGDL^*_{bd}$ still corresponds to a minimum weight max clique. The DP algorithm in [6] can then be used directly to find the optimal solution to $MGDL^*_{bd}$.

To achieve this, we first derive an efficient formula for $L_{bd}(gt, ST)$ (and $L^*_{bd}(gt, ST)$, similar to the one derived in [158] for $L_{std}(gt, ST)$, but somewhat more involved. We now show how we derive these formulas.

By Theorem 5.2.2, the number of duplications for a given gene tree and species tree pair does not depend upon how one interprets missing taxa in gene trees, and can be inferred directly from the MRCA mapping. Therefore, we will let $\mathcal{D}_{gt,ST}$ denote the set of duplication nodes in gt with respect to ST and $\mathcal{S}_{gt,ST}$ denote the set of speciation nodes in gt with respect to ST. When gt and ST are known, we may write these as $\mathcal{D}$ and $\mathcal{S}$.

However, the calculation for the number of losses does depend on how we interpret incompleteness in gene trees. Therefore, rather than having a single optimization problem like $MGDL$, we have variants of this problem depending on how we treat incompleteness. As shown in Theorem 5.2.1, the term $MGDL$ in the literature refers to $MGDL_{std}$, which (by Theorem 5.2.1) is identical to $MGDL_{samp}$. Here, we consider the optimization problems $MGDL^*_{bd}$, where we treat incompleteness as due to gene birth and death. And therefore, we also consider $MGDL_{bd}$, MGL^*_{bd}, and MGL_{bd}.

5.4.1 Basic material

Definition 5.4.1. *For a gene tree gt and a species tree ST such that $L(gt) \subseteq L(ST)$, the number of extra lineages (summing over all edges) is defined to be*

$$XL(gt, ST) = \sum_{e' \in E(ST^*(gt))} XL(gt, e'),$$

where $XL(gt, e')$ is the number of extra lineages on e'.

Definition 5.4.2. *(From [155]) For $B \subseteq \mathcal{X}$ and gene tree gt, we set $k_B(gt)$ to be the number of B-maximal clusters in gt (see Chapter 2).*

Definition 5.4.3. *We define $W_{xl}(x, gt)$ for x either a subtree-bipartition or a subset of $\mathcal{X}$, as follows. If $x \subseteq \mathcal{X}$, then we set $W_{xl}(x, gt) = 0$ if $x \cap L(gt) = \emptyset$ and otherwise $W_{xl}(x, gt) = k_x(gt) - 1$. If x is a subtree-bipartition, then we let $B = p \cup q$ for $x = (p|q)$, and we set $W_{xl}(x, gt) = 0$ if $B \cap L(gt) = \emptyset$, and otherwise $W_{xl}(x, gt) = k_B(gt) - 1$. For a set $\mathcal{G}$ of gene trees and ST a species tree, we set $W_0 = \sum_{gt \in \mathcal{G}} \sum_{x \in \mathcal{X}} W_{xl}(\{x\}, gt)$.*

[155] showed that for any edge e in ST, where B is the cluster below e, then $k_B(gt)$ is the number of lineages going through edge e, and so $k_B(gt) - 1$ is the number of extra lineages going through e. They defined weights on potential species tree clusters B by $W_{mdc}(B, gt) = 0$ if $B \cap L(gt) = \emptyset$ and otherwise $W_{mdc}(B, gt) = k_B(gt) - 1$ (i.e., W_{mdc} is defined for clusters while W_{xl} is defined for subtree-bipartitions), and extended this to a set $\mathcal{G}$ of gene trees by $W'_{mdc}(B) = \sum_{gt \in \mathcal{G}} W_{mdc}(B, gt)$, and then to a set C of clusters by $W''_{mdc}(C) = \sum_{B \in C} W'_{mdc}(B)$. From this, it follows easily that a set C of $n - 1$ compatible clusters minimizing $W''_{mdc}(C)$ defines a rooted binary species tree with a minimum MDC score.

Theorem 5.4.4. *(From Bayzid et al. [6] (presented in Chapter 3; see Corollary 3.3.3 and Theorems 3.3.4, 3.4.1 and 3.5.4))*

- *For any subtree bipartition $(A|B)$ and any species tree T, there is at most one subtree bipartition in T that dominates $(A|B)$.*

- *Let gt and ST be a gene tree and species tree pair, and $(A|B)$ be a subtree bipartition in gt associated with internal vertex v. Then v is a speciation* *node in gt if and only if $(A|B)$ is dominated by ST.*

- *A set Z of $n-1$ subtree-bipartitions on set $\mathcal{X}$ of n taxa is pairwise-compatible if and only if there is a rooted binary tree T such that $\mathcal{SBP}_T = Z$. Hence, every clique of size $n-1$ in $CG(\mathcal{G})$ defines a species tree on $\mathcal{X}$. Furthermore, if the nodes in $CG(\mathcal{G})$ are weighted by $W_{dom}(v)$, then a maximum weight clique with $n-1$ vertices defines an optimal solution to MGD. If the nodes are weighted by $W_{xl} - 3W_{dom}(v)$, then the minimum weight clique with $n-1$ vertices defines an optimal solution to $MGDL_{std}$.*

5.4.2 Deriving $L_{bd}(gt, ST)$ and $L^*_{bd}(gt, ST)$

We begin with the following theorem:

Theorem 5.4.5. *(From [158]) Let gt be a rooted binary gene tree, ST a rooted binary species tree and $\mathcal{D}$ the set of duplication nodes in gt with respect to ST. Then*

$$L_{std}(gt, ST) = XL(gt, ST(gt)) + 2|\mathcal{D}| + |V(gt)| - |V(ST(gt))|.$$

We now derive formulas for $L_{bd}(gt, ST)$ and $L^*_{bd}(gt, ST)$; to obtain formulas for $DL_{bd}(gt, ST)$ and $DL^*_{bd}(gt, ST)$, simply add $|\mathcal{D}(gt, ST)|$.

Recall that in the definition of $F(u, T)$ given in Eqn. 5.1, losses are associated with internal nodes, and the total number of losses is defined as the sum of losses associated to each internal node. However, the definition of the number of losses corresponding to a node can be rewritten in terms of edges, as we now show. Let $D(s, s') = d(s, s') + 1$; i.e., $D(s, s')$ is the number of edges in the path in ST between s and s'. Then, for a vertex u in gt with children r and l, we can rewrite Eqn. 5.1 as follows:

$$F(u, ST) = \begin{cases} D(\mathcal{M}(r), \mathcal{M}(u)) & \\ +D(\mathcal{M}(l), \mathcal{M}(u)) & \text{if } \mathcal{M}(r) \neq \mathcal{M}(u) = \mathcal{M}(l), \\ (D(\mathcal{M}(r), \mathcal{M}(u)) - 1) & \\ +(D(\mathcal{M}(l), \mathcal{M}(u)) - 1) & \text{if } \mathcal{M}(u) \notin \{\mathcal{M}(l), \mathcal{M}(r)\}, \\ D(\mathcal{M}(r), \mathcal{M}(u)) & \\ +D(\mathcal{M}(l), \mathcal{M}(u)) & \text{if } \mathcal{M}(r) = \mathcal{M}(u) = \mathcal{M}(l). \end{cases}$$

It is easy to see that in all three branches of the equation above, the two terms of the sum correspond to the edges connecting u to its two children l and r. (The second term in the first branch and both terms in the third branch are 0, but we wrote them in terms of the function $D(.,.)$ for convenience.) Therefore, we can associate gene losses to edges $e = (x, p(x))$ instead of nodes, as follows:

$$\mathcal{M}D(e) = D(\mathcal{M}(x), \mathcal{M}(p(x)), and$$

$$edgeloss_{ST}(e) = \begin{cases} \mathcal{M}D(e) & \text{if } p(x) \in \mathcal{D}_{gt,ST}, \\ \mathcal{M}D(e) - 1 & \text{otherwise.} \end{cases}$$

We use the subscript ST in $edgeloss_{ST}(e)$ to emphasize the fact that the distance is taken within the tree ST and not within $ST(gt)$. Note therefore

$$\sum_{u \in V_{int}(gt)} F(u, ST) = \sum_{e \in E(gt)} edgeloss_{ST}(e).$$

Lemma 5.4.6. *For all gene trees gt and species trees ST with $L(gt) \subseteq L(ST)$,*

$$L_{bd}(gt, ST) = \sum_{e \in E(gt)} \mathcal{M}D(e) - |E(gt)| + 2|\mathcal{D}|, \tag{5.4}$$

and for all sets $\mathcal{G}$ of gene trees,

$$\begin{aligned} L_{bd}(\mathcal{G}, ST) &= \sum_{gt \in \mathcal{G}} L_{bd}(gt, ST) \\ &= \sum_{gt \in \mathcal{G}} \sum_{e \in E(gt)} \mathcal{M}D(e) - \sum_{gt \in \mathcal{G}} |E(gt)| \\ &\quad +2 \sum_{gt \in \mathcal{G}} |\mathcal{D}_{gt,ST}|. \end{aligned} \tag{5.5}$$

Finally, equalities concerning $DL_{bd}(gt, ST)$ and $DL_{bd}(\mathcal{G}, ST)$ can be obtained from these equalities by adding $|\mathcal{D}(gt, ST)|$ and $|\mathcal{D}(gt, \mathcal{G})|$.

Proof. We partition all the non-root nodes in gt into two sets: CD (children of duplications), consisting of those nodes whose parents are duplication nodes, and CS (children of speciations), consisting of those nodes whose parents are speciation nodes. Note that every edge $(x, p(x)) \in E(gt)$ can be associated with the set containing x. Therefore,

$$\begin{aligned} L_{bd}(gt, ST) &= \sum_{e \in E(gt)} edgeloss_{ST}(e) \\ &= \sum_{x \,\in\, CD} \mathcal{M}D(x, p(x)) \\ &\quad + \sum_{x \,\in\, CS} (\mathcal{M}D(x, p(x)) - 1) \\ &= \sum_{e \in E(gt)} \mathcal{M}D(e) - |CS|. \end{aligned} \tag{5.6}$$

Since each internal node has two children, clearly the number of vertices x for which $p(x)$ is a speciation node is twice the number $|\mathcal{S}|$ of speciation nodes; therefore

$$L_{bd}(gt, ST) = \sum_{e \in E(gt)} \mathcal{M}D(e) - 2|\mathcal{S}|.$$

Since each internal node is a speciation node or a duplication node, it follows that $2(|\mathcal{D}| + |\mathcal{S}|) = |E(gt)|$, and the result follows. □

Let $L(gt, e)$ be the number of lineages that go through edge $e \in E(ST)$; thus, $XL(gt, e) = L(gt, e) - 1$, and so (by Definition 5.4.1)

$$XL(gt, ST) = \sum_{e' \in E(ST^*(gt))} L(gt, e') - |E(ST^*(gt))|. \tag{5.7}$$

Lemma 5.4.7. *For any gene tree gt and species tree ST,*

$$\sum_{e \in E(gt)} \mathcal{M}D(e) = \sum_{e' \in E(ST^*(gt))} L(gt, e').$$

and (by Equation 5.7)

$$XL(gt, ST) = \sum_{e \in E(gt)} \mathcal{M}D(e) - |E(ST^*(gt))|. \tag{5.8}$$

Thus, for a set $\mathcal{G}$ of gene trees and species tree ST,

$$\begin{aligned} XL(\mathcal{G}, ST) &= \sum_{gt \in \mathcal{G}} XL(gt, ST) \\ &= \sum_{gt \in \mathcal{G}} \sum_{e \in E(gt)} \mathcal{M}D(e) - \sum_{gt \in \mathcal{G}} |E(ST^*(gt))|. \end{aligned}$$

Proof. We establish the first equality, since the remaining ones follow directly from it. Consider the lists of edges in paths in ST from $\mathcal{M}(x)$ to $\mathcal{M}(p(x))$, as x ranges over the internal vertices in gt. It is easy to see that the number of occurrences of an edge $e' \in E(ST^*(gt))$ in these lists is $L(gt, e')$ (the number of lineages through e'). Also, the edges $e \in E(ST) - E(ST^*(gt))$ will not be present in these lists, since these are the edges incident on the missing clades in ST with respect to gt. Therefore, the sum of the lengths of these lists is equal to $\sum_{e\in E(gt)} \mathcal{M}D(e)$ and also equal to $\sum_{e\in ST^*(gt)} L(gt, e)$. □

Theorem 5.4.8. *For all gene trees gt, sets $\mathcal{G}$ of gene trees, and species trees ST,*

$$L_{bd}(gt, ST) = XL(gt, ST) + 2|\mathcal{D}| + |E(ST^*(gt))| - |E(gt)|$$

$$\begin{aligned} L_{bd}(\mathcal{G}, ST) = XL(\mathcal{G}, ST) + 2\sum_{gt\in\mathcal{G}} |\mathcal{D}_{gt,ST}| \\ + \sum_{gt\in\mathcal{G}} (|E(ST^*(gt))| - |E(gt)|). \end{aligned} \tag{5.9}$$

Proof. Follows from Lemma 5.4.6 and Lemma 5.4.7. □

Corollary 5.4.9. *For all gene trees gt and species trees ST,*

$$\begin{aligned} L^*_{bd}(gt, ST) &= L_{bd}(gt, ST) + |UMMC(gt, ST)| \\ &= XL(gt, ST) + 2|\mathcal{D}_{gt,ST}| \\ &\quad +|E(ST^*(gt))| - |E(gt)| \\ &\quad +|UMMC(gt, ST)|. \end{aligned} \tag{5.10}$$

and

$$\begin{aligned} DL^*_{bd}(gt, ST) &= L_{bd}(gt, ST) + |UMMC(gt, ST)| \\ &\quad +|\mathcal{D}_{gt,ST}| \\ &= XL(gt, ST) + 3|\mathcal{D}_{gt,ST}| \\ &\quad +|E(ST^*(gt))| - |E(gt)| \\ &\quad +|UMMC(gt, ST)| \qquad (5.11) \end{aligned}$$

Proof. The equalities concerning L^*_{bd} follow from Theorem 5.3.1 and Theorem 5.4.8. The equalities concerning DL^*_{bd} follow by adding $|\mathcal{D}(gt, ST)|$. □

5.4.3 Assigning weights to subtree-bipartitions

To use the graph-theoretic formulation of $MGDL^*_{bd}$, we have to assign weights to each node in the compatibility graph, $CG(\mathcal{G})$, where $\mathcal{G}$ is the input set of gene trees, so that a minimum weight clique of $n-1$ vertices defines an optimal solution to $MGDL^*_{bd}(\mathcal{G})$. We will define weights $W_{xl}(v), W_{dom}(v), W_{EC}(v)$, and $W_{MMC}(v)$ to each subtree-bipartition (i.e., node in the compatibility graph), and set

$$W_{MGDL^*_{bd}}(v) = W_{xl}(v) - 3W_{dom}(v) + W_{EC}(v) + W_{MMC}(v).$$

We then prove (see Theorem 5.4.10) that a set of $n-1$ compatible subtree-bipartitions that has minimum total weight defines a species tree that optimizes $MGDL^*_{bd}$. Note that weights $W_{xl}(v)$ and $W_{dom}(v)$ have already been defined (in Section 4.1.1 and Section 4.1.2, respectively). Hence, all that remains is to define $W_{EC}(v)$ and $W_{MMC}(v)$, and then to prove Theorem 5.4.10.

Calculating $W_{EC}(v)$ and $|E(ST^*(gt))|$: We now show how to define weight $W_{EC}(v, gt)$ for every vertex v in the compatibility graph $CG(\mathcal{G})$ so that for all species trees ST, $|E(ST^*(gt))|$ is the sum of the vertex weights for the $n-1$ clique $\mathcal{C}$ in $CG(\mathcal{G})$ corresponding to ST. To count the number of edges in $E(ST^*(gt))$, we need to exclude those edges from $E(ST)$ that are incident on a clade that is missing in gt. For a vertex v associated with the subtree-bipartition $(p|q)$, we define $W_{EC}(v, gt)$ as follows (swapping p and q as needed):

$$W_{EC}(v, gt) = \begin{cases} 0 & \text{if } p \cap L(gt) = \emptyset \text{ and} \\ & q \cap L(gt) \in \{L(gt), \emptyset\} \\ 1 & \text{if } p \cap L(gt) = \emptyset \text{ and} \\ & \emptyset \neq q \cap L(gt) \subsetneq L(gt) \\ 2 & \text{otherwise.} \end{cases} \tag{5.12}$$

Then, $|E(ST^*(gt))| = \sum_{u \in \mathcal{SBP}_{ST}} W_{EC}(u, gt)$. We set $W_{EC}(v) = \sum_{gt \in \mathcal{G}} W_{EC}(v, gt)$. Then, for any species tree ST and set $\mathcal{G}$ of gene trees,

$$\sum_{gt \in \mathcal{G}} |E(ST^*(gt))| = \sum_{v \in \mathcal{C}} W_{EC}(v), \tag{5.13}$$

where $\mathcal{C}$ is the clique in $CG(\mathcal{G})$ that corresponds to ST.

Calculating $W_{MMC}(v)$ and $|UMMC(gt, ST)|$: We now show how to assign the weight $W_{MMC}(v, gt)$ to each vertex v of the compatibility graph so that for all species trees ST, $|UMMC(gt, ST)|$ is the sum of the weights over all the vertices of the clique $\mathcal{C}$ in $CG(\mathcal{G})$ corresponding to ST. Recall that

$UMMC(gt, ST)$ is the set of upper maximal missing clades in ST. For a vertex v associated with the subtree-bipartition $(p|q)$, we define $W_{MMC}(v, gt)$ as follows (swapping p and q as needed):

$$W_{MMC}(v, gt) = \begin{cases} 1 & \text{if } p \cap L(gt) = \emptyset \text{ and } q \cap L(gt) = \\ & L(gt) \text{ (or vice-versa)} \\ 0 & \text{otherwise.} \end{cases} \tag{5.14}$$

Then $|UMMC(gt, ST)| = \sum_{u \in \mathcal{SBP}_{ST}} W_{MMC}(u, gt)$. Finally, we set $W_{MMC}(v) = \sum_{gt \in \mathcal{G}} W_{MMC}(v, gt)$. Then, for any species tree ST and set $\mathcal{G}$ of gene trees,

$$\sum_{gt \in \mathcal{G}} |UMMC(gt, ST)| = \sum_{v \in \mathcal{C}} W_{MMC}(v), \tag{5.15}$$

where $\mathcal{C}$ is the clique in $CG(\mathcal{G})$ that corresponds to ST.

We can extend the $MGDL^*_{bd}$ techniques to allow for losses and duplications to have different costs, as follows. Let c_d be the cost of a duplication and assume the cost of a loss (c_l) is 1. Let $\mathcal{D}_{\mathcal{G},ST} = \sum_i^k \mathcal{D}_{gt_i,ST}$, and set $DL^*_{bd}(\mathcal{G}, ST, c_d) = c_d * \mathcal{D}_{\mathcal{G},ST} + L^*_{bd}(\mathcal{G}, ST)$. Let $MGDL^*_{bd}(\mathcal{G}, c_d)$ be the problem that takes a set $\mathcal{G}$ of gene trees and duplication cost c_d as input, and finds the species tree that minimizes the weighted duploss score $DL^*_{bd}(\mathcal{G}, ST, c_d)$. Let $W^{c_d}_{MGDL^*_{bd}}(v) = W_{xl}(v) - (c_d + 2)W_{dom}(v) + W_{EC}(v) + W_{MMC}(v)$. (If $c_d = 1$, we omit the superscript c_d and write $W_{MGDL^*_{bd}}(v)$.)

Theorem 5.4.10. *Let $\mathcal{G} = \{gt_1, gt_2, \ldots, gt_k\}$ be a set of binary rooted gene trees on set $\mathcal{X}$ of n species, and set the weights on the vertices in the compatibility graph using $W^{c_d}_{MGDL^*_{bd}}(v)$. (a) A set of subtree-bipartitions in an $(n-1)$-clique of minimum weight in $CG(\mathcal{G})$ defines a binary species tree ST that minimizes $DL^*_{bd}(\mathcal{G}, ST, c_d)$. Furthermore, the weighted duploss score of ST is given by $W_0 + W^{c_d}_{MGDL^*_{bd}}(\mathcal{C}) + c_d(N-k)$. (b) If we reset the weights to be $W_{MGL^*_{bd}}(v) = W_{MGDL^*_{bd}}(v) + W_{dom}(v)$, then a set of subtree-bipartitions in an $(n-1)$-clique of minimum weight in $CG(\mathcal{G})$ defines a binary species tree ST that minimizes $L^*_{bd}(\mathcal{G}, ST)$.*

Proof. We prove (a), since (b) follows directly from (a). Let $\mathcal{C}$ be a clique of size $n-1$ in $CG(\mathcal{G})$ and ST the associated species tree. Let $\mathcal{SBP}_{dom}(gt, ST)$ be the set of subtree-bipartitions in gt that are dominated by a subtree-bipartition in ST. Note that $|\mathcal{SBP}_{dom}(gt, ST)|$ is the number of speciation nodes in gt with respect to ST. Therefore, the total number of speciation nodes in $\mathcal{G}$ is $\sum_{i=1}^{k} |\mathcal{SBP}_{dom}(gt_i, ST)| = \sum_{v \in V_{int}(ST)} W_{dom}(v)$. Also, $\sum_{v \in \mathcal{C}} W_{xl}(v) = \sum_{i=1}^{k} XL(gt_i, ST)$, and $\sum_{i=1}^{k} |\mathcal{D}_{gt_i, ST}| = \sum_{i=1}^{k}(n_i - 1) - \sum_{v \in \mathcal{C}} W_{dom}(v)$, where n_i is the number of leaves in gt_i. Finally, since all gene trees are rooted binary trees, $|E(gt_i)| = 2n_i - 2$ and $|V_{int}(gt_i)| = n_i - 1$. Recall that W_0 is the number of extra lineages contributed by the leaf set of the species tree (Definition 5.4.3). Therefore,

$$\begin{aligned}
DL^*_{bd}(\mathcal{G}, ST, c_d) &= \sum_{i=1}^{k}(c_d * \mathcal{D}(gt_i, ST) + L^*_{bd}(gt_i, ST)) \\
&= +\sum_{i=1}^{k}[XL(gt_i, ST) + (c_d+2)|\mathcal{D}_{gt_i,ST}| \\
&\quad + |UMMC(gt_i, ST)| \\
&\quad +|E(ST^*(gt_i))| - |E(gt_i)|] \text{ (by Cor. 5.4.9)} \\
&= W_0 + \sum_{v\in\mathcal{C}} W_{xl}(v) + \sum_{i=1}^{k}(c_d+2)(n_i-1) \\
&\quad -(c_d+2)\sum_{v\in\mathcal{C}} W_{dom}(v) + \sum_{v\in\mathcal{C}} W_{MMC}(v) \\
&\quad + \sum_{v\in\mathcal{C}} W_{EC}(v) - \sum_{i=1}^{k}(2n_i - 2) \\
&\quad \text{(by Equations 5.13 and 5.15.)} \\
&= W_0 + W^{c_d}_{MGDL^*_{bd}}(\mathcal{C}) + c_d(N-k)
\end{aligned}$$

where $N = \sum_{i=1}^{k} n_i$. Note that W_0 does not depend on the topology of the species tree. Hence, the $(n-1)$-clique $\mathcal{C}$ with minimum weight defines a tree ST that minimizes $DL^*_{bd}(\mathcal{G}, ST, c_d)$. The proof for (b) follows trivially. □

5.4.4 Dynamic programming algorithm

Let $\mathcal{SBP}$ be a set of subtree-bipartitions, with $\mathcal{SBP}$ equal to all possible subtree-bipartitions if an exact solution is desired, and otherwise a proper subset if a faster algorithm is desired or necessary. We present the DP algorithm for the $MGDL^*_{bd}(\mathcal{G}, c_d)$ problem.

Algorithm $MGDL^*_{bd}(\mathcal{G}, c_d)$

```
For $A \in \mathcal{SBP}$
  if $|A| = 1$ then $score(A) = W_{XL}(A)$
```

```
else
    $score(A) = max\{score(A_1) + score(A - A_1)$
    $+W^{c_d}_{MGDL^*_{bd}}(A_1|A - A_1) : (A_1|A - A_1) \in \mathcal{SBP}\}$
```

The running time is $O(n|SBP|^2)$. The optimal number of duplications and losses is given by $score(\mathcal{X}) + c_d(N - k)$, by Theorem 5.4.10. We can construct the optimal set of compatible clusters and hence the optimal species tree (subject to the constraint that all the subtree bipartitions in the output tree are in $\mathcal{SBP}$) using backtracking.

Theorem 5.4.11. *Let $\mathcal{G}$ be a set of rooted binary gene trees, $\mathcal{SBP}$ a set of subtree-bipartitions. The DP algorithm finds the species tree ST minimizing the total weighted GTP cost where c_d is the cost of a duplication and losses have unit cost, treating incomplete gene trees as resulting from gene birth and death, subject to the constraint that $\mathcal{SBP}_{ST} \subseteq \mathcal{SBP}$ in $O(n|\mathcal{SBP}|^2)$ time. Therefore, if $\mathcal{SBP}$ is all possible subtree-bipartitions, we have an exact but exponential time algorithm. However, if $\mathcal{SBP}$ contains only those subtree-bipartitions from the input gene trees, then the DP algorithm finds the optimal constrained species tree in $O(d^2n^3k^2)$ time, where n is the number of species, k is the number of gene trees, and d the maximum number of times that any taxon appears in any gene tree.*

Proof. Note that $|SBP_{ST}|$ is $O(dkn)$. The running time analysis follows the same argument as given in [6], since $W_{XL}(v)$ and $W_{dom}(v)$ can be computed in

$O(1)$ time after the preprocessing (as described in [6]). The proof of correctness follows from the observation that the DP algorithm correctly computes the minimum weight of all maximum cliques. □

Note that this theorem allows all values of c_d, and so can solve the standard $MGDL_{bd}$ problem in which losses and duplications have the same cost, or where they have different costs.

5.4.5 Extensions

It is trivial to extend the theory for $MGDL^*_{bd}$ and MGL^*_{bd} to $MGDL_{bd}$ and MGL_{bd}, as we now show. Recall that $L_{bd}(gt, ST) = L^*_{bd}(gt, ST) - |UMMC(gt, ST)|$ and that $DL_{bd}(gt, ST) = DL^*_{bd}(gt, ST) - |UMMC(gt, ST)|$. Therefore, to extend the algorithmic approach to solve MGL_{bd} and $MGDL_{bd}$, we define

$$W_{MGL_{bd}}(v, gt) = W_{MGL^*_{bd}}(v, gt) - W_{MMC}(v, gt)$$

and

$$W_{MGDL_{bd}}(v, gt) = W_{MGDL^*_{bd}}(v, gt) - W_{MMC}(v, gt),$$

and then seek a minimum weight maximum clique in the compatibility graph with these modified weights.

5.5 Conclusion

In this chapter we investigated how different reasons for gene tree incompleteness affects the mathematical formulation of gene loss. We showed that the standard definition of loss is appropriate when the reason for missing taxa is taxon sampling strategies, and can be incorrect if the incompleteness is due to true biological gene loss. We present the first mathematical formulation to model gene loss due to true biological loss, and distinguish this from incompleteness due to taxon sampling. We show that the optimal species tree by minimizing gene duplications and losses can be different based on different reasons for missing taxa. We propose exact and heuristic algorithms to infer species trees from a set of incomplete gene trees by minimizing gene duplications and losses by assuming that the incompleteness is due to true biological loss.

Chapter 6

Gene Tree Parsimony for Incomplete Gene Trees under ILS

The estimation of species trees typically involves the estimation of trees and alignments on many different genes, so that the species tree can be based on many different parts of the genome. This kind of *phylogenomic* approach to species tree estimation has the potential to produce more accurate species tree estimates, especially when gene trees can differ from the species tree due to processes such as incomplete lineage sorting (ILS), gene duplication and loss, and horizontal gene transfer. Because ILS (also called deep coalescence) is a frequent problem in systematics, many methods have been developed to estimate species trees from gene trees or alignments that specifically take ILS into consideration. In this chapter we consider the problem of estimating species trees from gene trees and alignments for the general case where the gene trees and alignments can be *incomplete*, which means that not all the

Much of the material in this chapter is taken without alteration from the following paper.

- M. S. Bayzid and T. Warnow. Estimating optimal species trees from incomplete gene trees under deep coalescence. *Journal of Computational Biology*, 19(6):591–605, 2012

TW designed the study; MSB performed the study; MSB and TW proved the theoretical results, analyzed the data, and wrote the paper.

genes contain sequences for all the species.

We formalize optimization problems for this context and prove theoretical results for these problems. We also present the results of a simulation study evaluating existing methods for estimating species trees from incomplete gene trees.

Our simulation study shows that *BEAST [52], a statistical method for estimating species trees from gene sequence alignments, produces by far the most accurate species trees. However, *BEAST can only be run on small datasets. The second most accurate method, MRP [3] (a standard supertree method), can analyze very large datasets and produces very good trees, making MRP a potentially acceptable alternative to *BEAST for large datasets.

6.1 Introduction

Over the last two decades, there have been dramatic improvements in the mathematical foundations of phylogenomic analyses. Many methods have been developed to construct a species tree from a collection of gene trees. However, little is known (in terms of both theoretical and empirical results) about the impact of incomplete estimated gene trees, by which we mean the case where the gene trees might not contain any individuals for some species. In this chapter, we consider the problem of estimating species trees from estimated gene trees when the true gene trees can differ from the true species trees due to incomplete lineage sorting. We focus our attention on the problem of estimating species trees from incomplete estimated gene trees. In this case, methods that require that all the gene trees have the same set of taxa cannot be applied. In addition, results from prior studies that evaluated methods on inputs in which all gene trees have at least one individual from each species are not necessarily applicable, since performance on incomplete gene trees could be different.

We begin with a study of the Minimize Deep Coalescence (MDC) problem introduced in [84]. This problem takes as input a set of rooted binary gene trees, each on the same set of taxa, and seeks the species tree for which there is a minimum total number of deep coalescences. Although this approach to species tree estimation is not statistically consistent when gene trees can differ from the species tree due to ILS [142], it is one of the most popular techniques for estimating species trees when ILS is suspected. We show how to extend MDC to the case where the gene trees are incomplete, and we prove that Phylonet-MDC [143] solves this computational problem exactly. We then report on a simulation study we performed to evaluate methods for estimating species trees from incomplete gene trees or alignments for datasets with multiple genes and with 11, 17, or 100 taxa. We compare *BEAST [52], a Bayesian method for estimating species trees from gene sequence alignments when genes can differ from species trees due to ILS, to methods based on MDC (iGTP-MDC [15] and Phylonet-MDC). We also make comparisons to a heuristic for MRP (matrix representation with parsimony, a standard supertree method) [2, 115] known to be one of the most accurate supertree methods [72, 135] and to heuristics to minimize duplications or duplications+losses in iGTP [15], none of which consider ILS when estimating species trees. We compare these methods on datasets simulated on gene trees that can differ from species trees due to ILS and report the missing branch rates of each species tree that we compute.

Although we did not attempt to run *BEAST on the 100-taxon datasets (due to its excessive computational requirements on large datasets), it produced the most accurate trees on the datasets with 11 or 17 taxa. Comparisons between other methods showed that generally MRP gave the most accurate results, and that (when it could be run), the exact version of Phylonet-MDC produced the next most accurate results. In addition, MRP was very fast on these datasets, producing results in under a minute on all datasets. These results suggest that at least for some conditions involving incomplete gene trees, methods that attempt to solve MRP may be computationally tractable ways of producing reasonably accurate species trees, and perhaps better than methods that optimize the MDC criterion. However, for those datasets for which statistical methods (such as *BEAST) can be run, they may be able to produce substantially more accurate trees than all other methods.

6.2 Theoretical results for MDC

We begin by defining the MDC problem in the context of complete rooted, binary gene trees. We then show how to extend MDC to incomplete gene trees.

Name	Meaning	Comments
ILS	Incomplete Lineage Sorting	Also called “deep coalescence”
MBMC	Minimizing B-maximal clusters	A computational problem for estimating species trees from complete gene trees, shown to be equivalent to MDC in [155]
MBMC$_{inc}$	MBMC for incomplete gene trees	Extension of MBMC to incomplete gene trees, shown here to be equivalent to MDC$_{inc}$.
MDC	Minimize Deep Coalescence	Optimization problem for species tree estimation in the presence of ILS, defined only for complete gene trees
MDC$_{inc}$	MDC for incomplete gene trees	MDC$_{inc}$ seeks completions of all gene trees and a species tree, so that the species tree optimizes MDC with respect to the completed gene trees.
MRP	Matrix Representation with Parsimony	Standard optimization problem for supertree computation, known to be NP-hard.

Table 6.1: **Acronyms used in this chapter**

Name	Summary	Reference
*BEAST	Bayesian co-estimation of gene trees and species trees, in the presence of ILS	[52]
FastTree-2 (FT)	Fast maximum likelihood phylogeny estimation. FT-75 refers to the tree obtained by running FastTree-2 and then collapsing all branches with support below 75%.	[114]
iGTP	Gene Tree Parsimony software, implementing a heuristic search to construct species trees from sets of gene trees, under three criteria: MDC, duplications, and duplications plus losses	[15]
PAUP*	Phylogenetic Analysis using Parsimony (*and Other Methods). We use heuristics in PAUP* for parsimony, applied to an MRP matrix we compute.	[137]
Phylonet	Software package that performs several functions related to species phylogeny estimation from sets of gene trees. In this paper we use Phylonet to find solutions (exact or heuristic) to the MDC problem.	[143]

Table 6.2: **Software used in this study**

6.2.1 MDC for complete gene trees

The MDC problem is as follows:

- Input: A set $\mathcal{T} = \{t_1, t_2, \ldots, t_k\}$ of rooted, binary gene trees with each tree t_i on the same set S of taxa.
- Output: a rooted, binary species tree T that minimizes the number of extra lineages with respect to $\mathcal{T}$, denoted by $XL(T, \mathcal{T}) = \sum_i XL(T, t_i)$

To define the MDC problem, therefore, we need to define $XL(T, t_i)$, i.e., the number of extra lineages of a species tree T with respect to a gene tree t_i. Visually, this is defined by embedding the gene tree t_i into the species tree T, and then counting how many lineages there on each edge of the species tree; for a given edge, the number of extra lineages is one less than the total number of lineages on the edge [84].

An alternative definition is given in terms of the B-maximal clusters (defined in Chapter 2). For a cluster B of T, we define $k_B(t)$ to be the number of B-maximal clusters of t, and we let $w_B(t) = k_B(t) - 1$. It is now known that the embedding of the gene tree t into the species tree T that maps every node in t to MRCA (most recent common ancestor) in T of the leafset below v optimizes the MDC cost. Furthermore, for this embedding, the number of lineages “leaving” the parent edge of the cluster B (i.e., the edge between B and the root of the tree T) is $k_B(t)$; therefore, the number of extra lineages on the parent edge is $w_B(t)$ (one less than the number of lineages) (see, for example, [155]). Note that $w_B(t) \geq 0$ since t and T have the same set of taxa, and that $XL(T,t) = \sum_B w_B(t)$, where the sum is taken over all clusters B in the tree T, is the number of extra lineages implied by the pair t, T. This is what is meant by the MDC cost for T with respect to gene tree t.

The MDC problem can then be restated as follows:

- Input: set $\mathcal{T} = \{t_1, t_2, \ldots, t_k\}$ of binary rooted trees on leafset S.
- Output: binary rooted species tree T on S such that $XL(T, \mathcal{T}) = \sum_i \sum_B w_B(t_i)$ is minimized, where i ranges from 1 to k and B ranges over the clusters in the species tree T.

Than and Nakhleh noted that this problem could be solved exactly by finding a minimum weight clique of size $n-2$ in a graph in which there is a node for every possible cluster in the species tree (i.e., subset of taxa), an edge between nodes where their clusters are compatible (meaning that they can co-exist in a rooted tree), and where the weight of the node for cluster B is $\sum_i w_B(t_i)$ [141]. This observation yielded the exact version of Phylonet-MDC [143]. By restricting the set of nodes to those clades that appear in the input set of gene trees, they produced the heuristic version of Phylonet-MDC; this method solves the MDC problem exactly when constrained to species trees whose clades are drawn only from the input gene tree clades. Finally, [155] showed how to modify the Phylonet-MDC algorithm so that it could work with unrooted, partially resolved gene trees and find optimal rooted refinements and species trees that minimize the MDC score.

6.2.2 Extension to incomplete gene trees

We now discuss how to extend the MDC criterion to handle incomplete gene trees, where the gene tree leaf sets may not contain all the species. We begin with a definition: If S is the full set of taxa and t is a binary rooted tree on a subset of S, then we say that t' is a *completion* of t if t' is a binary rooted tree that contains all the taxa in S and that agrees with t when restricted to the taxa in t. Thus, a completion t' is obtained by adding additional leaves to t so that it contains all the taxa it is missing. With this, we can now define MDC for incomplete gene trees.

MDC for incomplete gene trees (MDC_{inc}).

- Input: set $\mathcal{T} = \{t_1, t_2, \ldots, t_k\}$ with each t_i a rooted binary tree on leafset S_i, with $S_i \subseteq S$ (i.e., each t_i is an incomplete rooted binary tree)
- Output: binary rooted species tree T and completions t'_i of t_i so as to minimize $XL(T, \mathcal{T}')$, where $\mathcal{T}' = \{t'_1, t'_2, \ldots, t'_k\}$.

We will refer to this problem as MDC-incomplete, and we will denote a solution to MDC-incomplete on input set $\mathcal{T}$ by $MDC_{inc}(\mathcal{T}) = (T, \mathcal{T}')$.

Recall that $k_B(t)$ is defined for the case where the gene tree t is rooted and has the same set of taxa as the species tree; in this case, it equals the number of B-maximal clusters of t. Furthermore, again for the case where the gene trees all have the same set of taxa, we have defined $XL(T, \mathcal{T}) =$

$\sum_B \sum_i w_B(t)$, where B ranges over all clusters of T, i ranges from 1 to k, and $w_B(t) = k_B(t) - 1$. However, we will modify the definition of $w_B(t)$ to appropriately reflect the possibility that the cluster B may contain taxa that do not appear in t. That is, we set

- $w_B(t) = 0$ if $B \cap \mathcal{L}(t) = \emptyset$ (where $\mathcal{L}(t)$ denotes the leafset of t), and
- $w_B(t) = k_B(t) - 1$, otherwise.

In other words, we generally use the same definition for $w_B(t)$, except when B is entirely disjoint from the leafset of t. This definition ensures that $w_B(t) \geq 0$ for all clusters B and all gene trees t.

Minimizing B-maximal clusters ($MBMC_{inc}$).

- Input: set $\mathcal{T} = \{t_1, t_2, \ldots, t_k\}$ of binary rooted trees, with t_i on leafset S_i, for $i = 1, ..., k$.
- Output: binary rooted species tree T on $S = \cup_i S_i$ such that $\sum_i \sum_B w_B(t_i)$ is minimized, where i ranges from 1 to k and B ranges over all clusters in T.

We refer to this problem as MBMC-incomplete, and the optimal tree given input $\mathcal{T}$ is given by $MBMC_{inc}(\mathcal{T})$. Note that when $S_i = S_j$ for all i, j, then all the gene trees are complete (on the same set of taxa), and the problem is identical to the MDC problem (optimal solutions to this problem minimize the number of extra lineages).

The main result in this chapter is the following:

Theorem 1: Let $\mathcal{T}$ be a set of incomplete, rooted, binary gene trees. If $T = MBMC_{inc}(\mathcal{T})$ then there exists extensions t'_i for each t_i so that $MDC_{inc}(\mathcal{T}) = (T, \mathcal{T}')$, where $\mathcal{T}' = \{t'_1, t'_2, \ldots, t'_k\}$. Also, if $MDC_{inc}(\mathcal{T}) = (T, \mathcal{T}')$, then $T = MBMC_{inc}(\mathcal{T})$.

In other words, the species tree T that optimizes the $MBMC_{inc}$ criterion is the species tree component of the optimal solution to MDC_{inc}.

Phylonet-MDC and iGTP-MDC. The software packages Phylonet [143] and iGTP [15] handle incomplete gene trees differently when attempting to solve MDC, in that they can compute MDC scores differently. In particular, Phylonet defines the MDC score using the $MBMC_{inc}$ cost, as described above (i.e., the cost of a species tree T is $\sum_i \sum_B w_B(t_i)$, where B ranges over the clusters in T and i ranges from 1 to k). Theorem 1 thus shows that Phylonet-MDC computes the MDC score correctly. By contrast, there are inputs for which iGTP-MDC does not return this score, indicating that iGTP-MDC defines the MDC score differently for incomplete gene trees. One particular instance in which this occurs is as follows:

- Input gene trees: T_1 = (((a, b), c), d), T_2 = ((b, c), (d, e)), and T_3 = ((a, d), (b, e)).

- Output of Phylonet-MDC exact version: ((d,e),(a,(b,c,))), claiming 3 extra lineages.

- Output of iGTP(MDC): ((d,e),(c,(a,b))), and claims 2 extra lineages.

By our calculation and definition for MDC_{inc}, Phylonet-MDC correctly computes the number of extra lineages, but iGTP does not. We conjecture that iGTP seeks the species tree T that minimizes $\sum_i XL(T_i, t_i)$, where T_i is the subtree of T induced by S_i. Therefore, iGTP-MDC and Phylonet solve different problems when given gene trees that are incomplete.

6.3 Establishing the relationship between MDC_{inc} and $MBMC_{inc}$

In this section we establish the relationship between optimizing the MDC_{inc} and $MBMC_{inc}$ problems. As a result of this theorem, it will follow that the exact formulation of Phylonet-MDC solves the MDC problem optimally. That is, given an input of incomplete, binary rooted gene trees, to find an optimal species tree and completions of the binary gene trees it will suffice to find a minimum weight clique containing $n - 2$ vertices (where n is the number of taxa) in the graph defined by Phylonet, which has one vertex for each possible cluster, edges between vertices exist if and only if their clusters are compatible (either disjoint or one contains the other), and the weight on the vertex for cluster B set to w_B.

We begin with the following lemma.

Lemma 6.3.1. *Let T and t be rooted binary trees with $\mathcal{L}(t) \subset \mathcal{L}(T)$, and let X be a maximal cluster in T with $X \cap \mathcal{L}(t) = \emptyset$. Let B_0 be the sibling cluster of X in T (i.e., $X \cup B_0$ is the smallest cluster in T that properly contains X), and let A_0 be any B_0-maximal cluster in t. Let t' be the rooted binary tree obtained by modifying t by inserting the clade for X as the sibling to the clade on A_0. Then for all clusters B of T, $w_B(t) = w_B(t')$.*

Proof. We consider the four cases that can occur in a species tree T in which B, B_0 and X are clusters:

- Case 1: $B \subseteq X$
- Case 2: $B \subseteq B_0$
- Case 3: $B_0 \cup X \subseteq B$
- Case 4: $(B_0 \cup X) \cap B = \emptyset$

We take each case in turn.

Case 1: $B \subseteq X$. In this case, B is a cluster in the clade on X, and hence a cluster in t'. Therefore, $w_B(t') = 0$. Since $B \subseteq X$, it follows that $B \cap \mathcal{L}(t) = \emptyset$, and so (by definition) $w_B(t) = 0$.

Case 2: $B \subseteq B_0$. First, if $B \cap \mathcal{L}(t) = \emptyset$, then $B \cap \mathcal{L}(t') = \emptyset$ and $w_B(t) = w_B(t') = 0$. Hence, assume that $B \cap \mathcal{L}(t) \neq \emptyset$. We will show that A is a B-maximal cluster in t if and only if A is a B-maximal cluster in t', and so $w_B(t) = w_B(t')$. Suppose A is B-maximal in t. Then A is a cluster of t

and (since $A \subseteq B \subseteq B_0$) also a cluster of t'. Hence A will be B-maximal for t' unless the parent cluster in t' of A is a subset of B. Since A is B-maximal in t, the parent cluster of A in t is not a subset of B. Note that A's parent cluster in t' is either the same cluster as in t, or else the parent cluster in t' contains $A_0 \cup X$; in either case, the parent cluster of A in t' is not a subset of B. Therefore, A is also B-maximal in t'.

Conversely, suppose A is a B-maximal cluster in t'. Since $A \subseteq B \subseteq B_0$, A is a cluster in t. If A is B-maximal in t, then we are done. Else, suppose A is not B-maximal in t. Note that A cannot have the same parent cluster in t and t', since otherwise A is also B-maximal in t' (contradicting our hypothesis), and so A's parent cluster in t' must contain $A_0 \cup X$. Hence, A's parent cluster in t must be defined by an internal node on the path from the root of A_0 to the root of t. Label the nodes on that path $root(A_0) = v_0, v_1, \ldots, v_t = root(t)$, and let the "other" child of each $v_i, i = 1, 2, \ldots, t$ be w_i, defining cluster A_i. Note that A_1 is the sibling cluster to A_0 in t. Then $A = A_i$ for some i. Note that if $A = A_0$, then A is B-maximal in t (since A_0 is B_0-maximal in t and $B \subseteq B_0$). Note also that $A \neq A_1$, since otherwise A_1 is B-maximal, and so $A_1 \subseteq B \subseteq B_0$, contradicting that A_0 is B_0-maximal. Now suppose that $A = A_i$, for some $i \geq 2$. Then the parent cluster of A in t contains X, and so is not a subset of B, establishing that A is B-maximal in t as well. Therefore, $w_B(t) = w_B(t')$.

Case 3: $B_0 \cup X \subseteq B$. Our first observation is that A_0 is B-maximal in t if and only if $A_0 \cup X$ is B-maximal in t'. Hence, we need only concern ourselves with the B-maximal clusters in t other than A_0, and (equally) with the B-maximal clusters in t' other than $A_0 \cup X$. However, when $A \neq A_0$, it is easy to see that A is a B-maximal cluster in t if and only if A is a B-maximal cluster in t'. Hence, $w_B(t) = w_B(t')$.

Case 4: $(B_0 \cup X) \cap B = \emptyset$. It is easy to see that for any cluster A, A is B-maximal in t if and only if A is B-maximal in t', and so $w_B(t) = w_B(t')$. $\square$

The following lemma is obvious and the proof is omitted:

Lemma 6.3.2. *Let t be an incomplete gene tree, T a species tree, and t' a completion of t to the taxon set of T. Then $w_B(t) \leq w_B(t')$ for all clusters B of T.*

Theorem 6.3.3. *Let $\mathcal{T}$ be a set of incomplete, rooted, binary gene trees. If $T = MBMC_{inc}(\mathcal{T})$ then there exists extensions t'_i for each t_i so that $MDC_{inc}(\mathcal{T}) = (T, \mathcal{T}')$, where $\mathcal{T}' = \{t'_1, t'_2, \ldots, t'_k\}$. Also, if $MDC_{inc}(\mathcal{T}) = (T, \mathcal{T}')$, then $T = MBMC_{inc}(\mathcal{T})$.*

Proof. Let $t \in \mathcal{T}$ be given, and let $T = MBMC_{inc}(\mathcal{T})$. By Lemma 6.3.2, for any completion t' of t and any cluster B of T, $w_B(t') \geq w_B(t)$. By Lemma 6.3.1, there is a completion t' of t that achieves $w_B(t) = w_B(t')$ for all clusters B of T. Since t was arbitrary, we can let $\mathcal{T}'$ denote the set of completions of each $t \in \mathcal{T}$ so that $w_B(t) = w_B(t')$ for all clusters B of T. Hence, the number of extra lineages in T with respect to $\mathcal{T}'$ is $\sum_B \sum_i w_B(t)$, where B ranges over the clusters B of T and i ranges from 1 to k, where $\mathcal{T} = \{t_1, t_2, \ldots, t_k\}$. It follows, by Lemma 6.3.2, that T has the minimum number of extra lineages with respect to any set of completions of $\mathcal{T}$, and so $(T, \mathcal{T}')$ is a solution to $MDC_{inc}(\mathcal{T})$.

For the converse, let $(T, \mathcal{T}')$ be a solution to $MDC_{inc}(\mathcal{T})$, with $\mathcal{T}' = \{t'_1, t'_2, \ldots, t'_k\}$ (each t'_i a completion of t_i). Then since $\mathcal{T}'$ is a set of rooted, binary, complete gene trees (i.e., all on the same set of taxa as T), it follows that $XL(T, \mathcal{T}') = \sum_i \sum_B w_B(t'_i)$, as B ranges over the clusters of T and i ranges from $1...k$, and that this is the minimum possible among all species trees T and set $\mathcal{T}'$ of completions of the gene trees. Therefore, for all clusters B in T and for all i, $w_B(t_i) = w_B(t'_i)$, since otherwise we could complete t_i differently. Now suppose the tree T isn't an optimal solution to $MBMC_{inc}(\mathcal{T})$. Therefore, for some other binary rooted species tree T^* on the same set of taxa, $\sum_B \sum_i w_B(t_i) < XL(T, \mathcal{T}')$, where B ranges over the clusters of T^*. But then there is a completion $\mathcal{T}^*$ of the gene trees in $\mathcal{T}$ so that $XL(T^*, \mathcal{T}^*) < XL(T, \mathcal{T}')$, contradicting our hypothesis. □

6.4 Materials and methods

6.4.1 Overview

The simulation study used gene sequences that evolve down gene trees that can differ from the true species tree due to ILS. To produce these sequence datasets, we used sequences used in previous studies and provided to us by the authors of these studies–the 11-taxon datasets from [19], the 17-taxon datasets from [141], and the 100-taxon datasets from [152]. We summarize the simulation protocols used in these studies here, and direct the reader to the relevant publication for the details of how the data were generated.

In each case, a model species tree was generated (typically using a birth-death process). Then a set of gene trees within each species tree was produced under a coalescent process, so that for each gene one individual was sampled for each species. This produces gene trees with branch lengths that can differ topologically from their associated species tree due to ILS. DNA sequences were then simulated down each gene tree. For the 11-taxon and 17-taxon datasets, these simulations were done under a substitution-only model, and for the 100-taxon datasets these simulations were done under GTR+Gamma+gap models with varying gap lengths; thus, the 100-taxon datasets evolved with indels as well as with substitutions. Many replicates were generated for each model condition, and each replicate consisted of true sequence alignments for each gene.

For each replicate dataset we had the true alignment as well as the true tree. We then deleted taxa randomly, varying the number of taxa removed, from each gene sequence alignment, thus producing incomplete gene sequence alignments. On each resultant gene sequence alignment we estimated trees using FastTree-2 [114]; this produces a tree as well as branch support estimations. We produced a 75%-branch support version of each estimated gene tree by contracting all edges with support below 75%.

For each replicate of each model, we thus have three types of datasets (each consisting of a collection of gene sequence alignments and trees): the true gene sequence alignment, the binary trees estimated by FastTree-2 on the true gene sequence alignment, and the 75%-branch support FastTree-2 trees estimated on each true gene sequence alignment.

For each such dataset, we estimated species trees using the following techniques:

- iGTP v. 1.1. We explore all three optimization criteria (deep coalescence, duplications, duplication-loss) available in iGTP. We ran iGTP on 75% support version of the input binary trees, although it is not guaranteed to give meaningful outputs for non-binary gene trees.
- Phylonet v. 2.4. We explore both heuristic and exact version of Phylonet used to solve the MDC problem on both binary and unresolved gene trees. However, the exact version can only be run on small datasets, and so we used it only on the 11-taxon datasets.
- Matrix Representation with Parsimony (MRP). We ran MRP heuristics on the FastTree-2 trees (both binary and 75%-support versions), using a Python script to run a parsimony ratchet analysis using PAUP*, with 100 iterations, followed by taking the greedy consensus of the set of trees.
- *BEAST v. 1.6.2. We ran *BEAST on the true alignments for each dataset using its default settings.

We recorded the average (over all replicates) missing branch rate and running time for each method. When computing the missing branch rate, we compare the estimated species tree to the subtree of the true species tree induced by those species present in at least one gene tree.

6.4.2 Datasets

We ran our experiments on datasets that evolve with ILS. We used 11-taxon datasets, each with 10 genes, obtained from [19]. We also used 17-taxon datasets with 8 genes each, used previously in [141]. Finally, we used 100-taxon datasets with 25 genes each, used in [152].

6.5 Results

6.5.1 Missing branch rates

We begin by discussing performance with respect to missing branch rates.

6.5.1.1 Results on 11-taxon datasets

For these datasets, we were able to run the exact version of Phylonet-MDC, and hence solve the MDC problem exactly. As before, we ran the heuristic version of Phylonet-MDC, the three iGTP methods (for the MDC score, duplication score, and duplication plus losses score), and MRP. We explored results with two, three, and five missing taxa; see Figure 6.1.

The first observation is that *BEAST produced the most accurate species trees, for all percents of missing taxa. The second best method varied depending on the percentage of missing taxa, with MRP on the 75%-support trees best for 20% missing taxa, Phylonet-exact on the 75%-support trees best for 30% missing taxa, and MRP best for 50% missing taxa. Thus, there was no clear second best method. Furthermore, although these three methods generally gave reasonably good results, they were not always among the next most accurate. Between the iGTP methods, iGTP-dup had the worst results, and iGTP-MDC and iGTP-duploss were sometimes reasonably accurate. A noteworthy trend was that Phylonet-heuristic gave the worst results at all percents of missing taxa, whether applied to the fully resolved trees or the 75%-support trees. Finally, using the 75%-support trees instead of the fully resolved trees improved MRP and Phylonet (both exact and heuristic) for small numbers of missing taxa, but not when the number of missing taxa was large. Also, using the 75%-support trees did not help the other methods.

6.5.1.2 Results on 17-taxon datasets

Performance on 17-taxon datasets with 8 genes showed similar results, see Figure 6.2. Because of the number of taxa, we did not run Phylonet-exact. However, the results we saw here are similar to what we saw on the 11-taxon datasets. As before, *BEAST was the most accurate, for all percents of missing taxa. The next best methods were MRP and iGTP-MDC (on either binary or 75%-support trees), and sometimes also iGTP-duploss on binary trees, but all had at least 7% higher missing branch rates than *BEAST. The worst results were obtained using Phylonet-heuristic and iGTP-dup on either the binary or 75%-support trees.

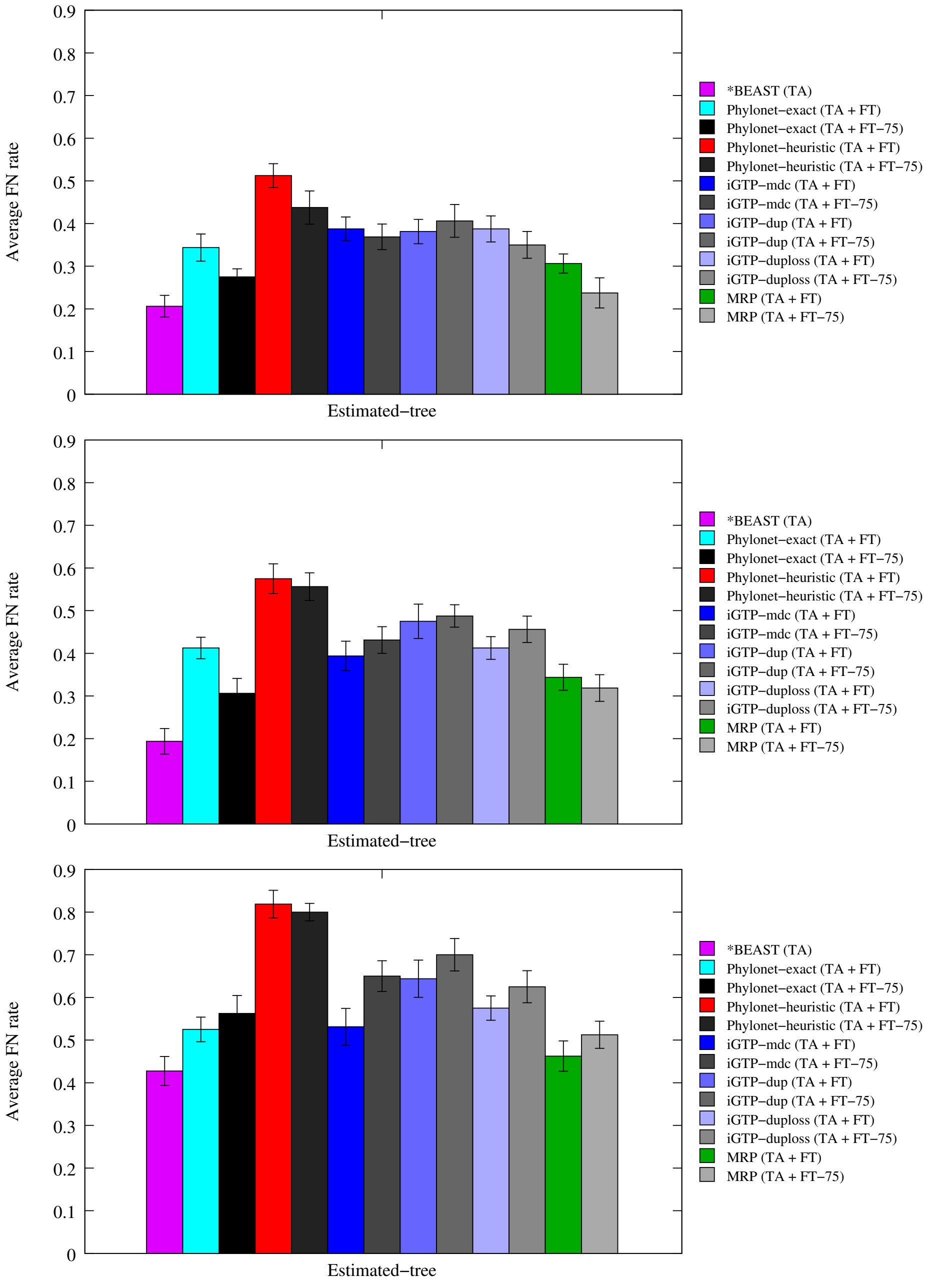


Figure 6.1: **Average missing branch rates of methods on twenty (20) 11-taxon 10-gene datasets on true alignments (TA)**. Gene trees are estimated using FastTree-2 (FT), and in some cases the branches with support less than 75% are contracted (FT-75). From top to bottom, the number of missing taxa is 2, 3, and 5.

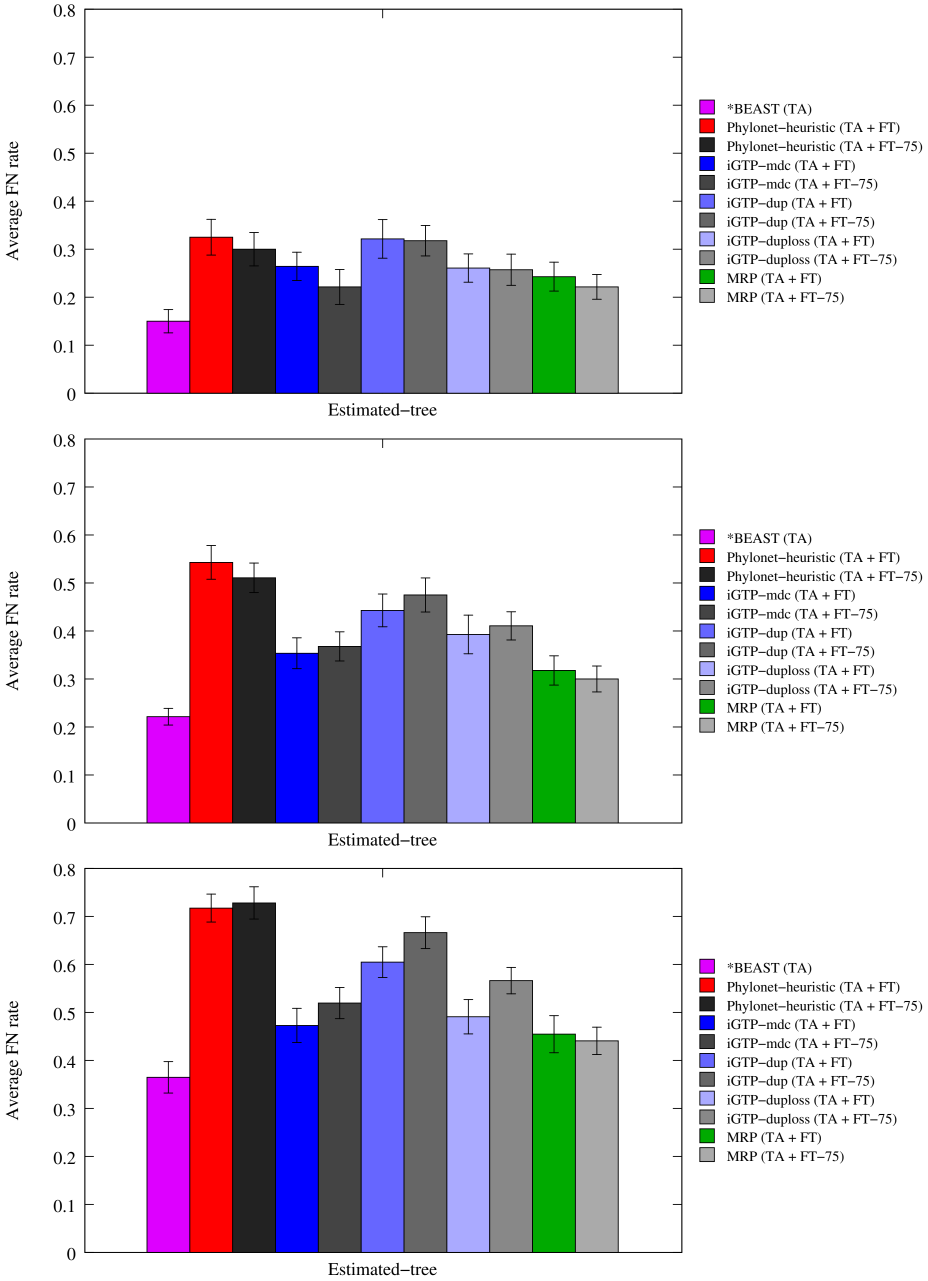


Figure 6.2: **Average missing branch rates of methods on twenty (20) 17-taxon 8-gene datasets on true alignments (TA)**. Gene trees are estimated using FastTree-2 (FT), and in some cases the branches with support less than 75% are contracted (FT-75). From top to bottom, the number of missing taxa is 1, 5, and 8.

6.5.1.3 Results on 100-taxon datasets

We now describe results on the 100-taxon datasets. Because of the number of taxa, we did not run *BEAST (running long enough to reach convergence was infeasible for this experimental study), nor Phylonet-exact. However, these data allow us to compare the other methods, Phylonet-heuristic, the three variants of iGTP, and MRP, on both binary and 75%-support trees; see Figure 6.3.

On the estimated gene trees, MRP on the 75%-support trees gives the most accurate trees, but MRP on binary trees comes quite close. The least accurate method is Phylonet-heuristic on binary trees, and Phylonet-heuristic on 75%-support trees is only slightly better (and much less accurate than all the other methods). A comparison between the iGTP methods no longer shows no reliable differences: for example, sometimes iGTP-MDC is the best and sometimes it is the worst of the three.

6.5.1.4 Overall results

For all levels of missing data, certain trends were clearly seen. Results for all methods improved when given more estimated gene trees rather than fewer; these trends are to be expected, and consistent with prior studies (see, for example, [152]). In addition, we saw that for each species tree estimation method, the missing branch rate increased with increased levels of taxon deletion, but the increase in error was particularly large for the heuristic version of Phylonet-MDC.

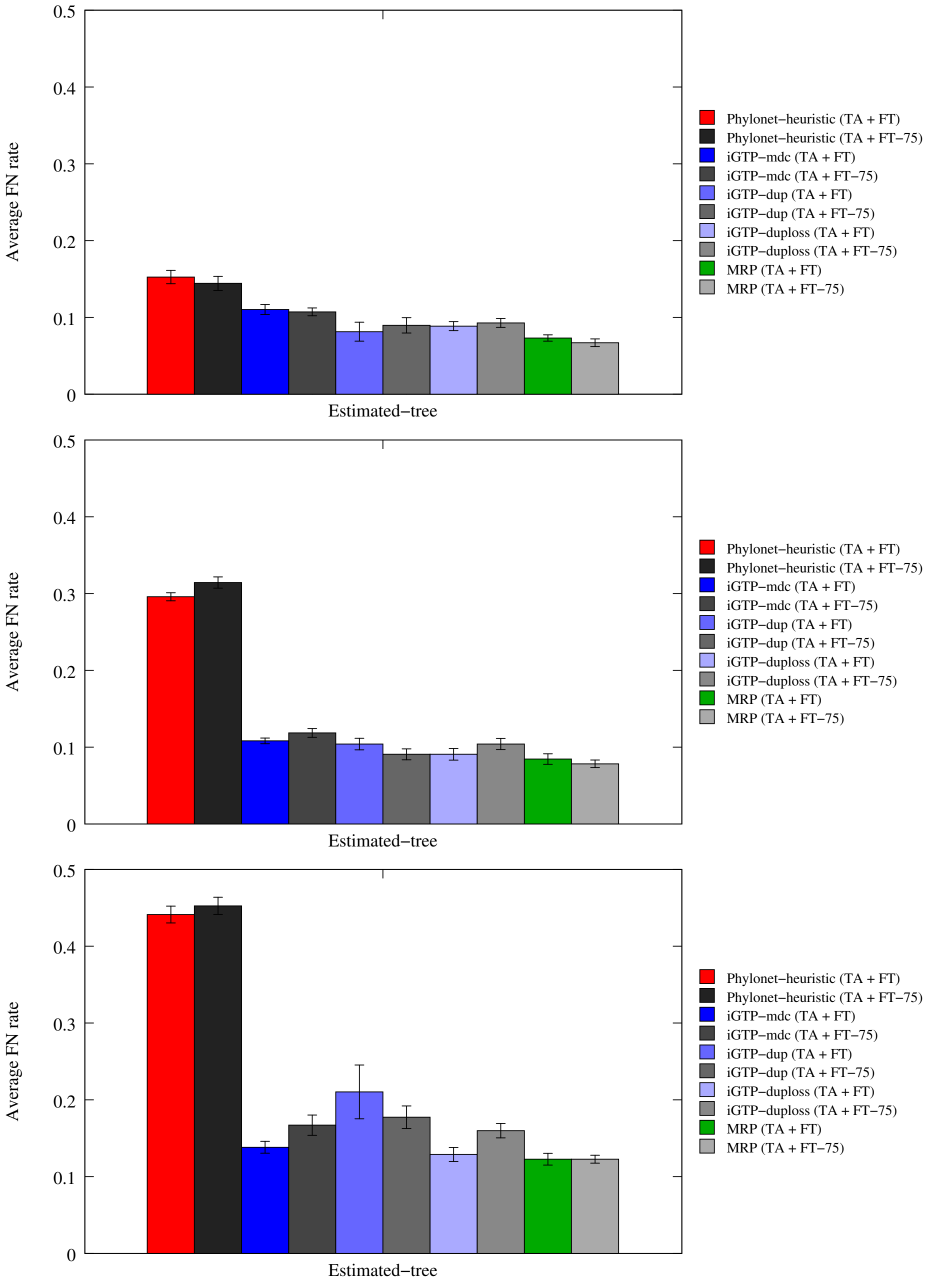


Figure 6.3: **Average missing branch rates of methods on ten (10) 100-taxon 25-gene datasets on true alignments (TA)**. Gene trees are estimated using FastTree-2 (FT), and in some cases the branches with support less than 75% are contracted (FT-75). From top to bottom, the number of missing taxa is 10, 30, and 50.

The relative performance between methods showed clearly that when analyzing estimated gene trees, *BEAST produced the most accurate results (as indicated by the lowest missing branch rate). MRP, especially on the 75%-support trees, typically came in second or close to second, and when Phylonet-exact could be run, it gave results that were close to that of MRP. However, in all the experiments, Phylonet-heuristic gave the least accurate results or tied for last. Comparisons between the iGTP methods depended on the model condition, and no overall trends could be observed.

Using 75%-support trees had a variable effect on the different methods we explored. First, on low taxon deletion levels, Phylonet-MDC (in either the exact or heuristic version) and MRP were improved by using the 75%-support trees, but this changed for the highest level of taxon deletion. Why this is happened is unclear, although it could be that the branches on the estimated gene trees had low support when estimated on very sparse taxon sets (such as would be obtained by deleting many taxa), leading to more loss of information when using the 75%-support trees. On the other hand, the iGTP methods did not show any advantage when used with 75%-support trees, and were often hurt.

6.5.2 Computational Issues

We also evaluated the running time and memory usage of the different methods we studied. Phylonet-exact uses time that is exponential in the number of taxa, and so could only be run on the 11-taxon datasets; however, on these datasets it completed in less than 2 seconds. The next most expensive method is *BEAST, which must be run long enough to converge to the stationary distribution. Therefore, we only ran *BEAST on the 11-taxon and 17-taxon datasets. On average, *BEAST finished its analyses in 15 minutes on the 11-taxon datasets and 20 minutes on the 17-taxon datasets. The remaining methods were much faster: all finished in under a second on the 11-taxon and 17-taxon datasets, and in under a minute on the 100-taxon datasets. Some differences in running time were evident on the 100-taxon datasets, where Phylonet-heuristic finished in 6 seconds, MRP finished in 20 seconds, but the three iGTP methods took between 20-64 seconds. Peak memory usage by these methods all differed, but only *BEAST used any substantial memory – about 1GB on the 17-taxon datasets.

6.6 Discussion

We begin with some observations about methods that attempt to optimize the MDC criterion. First, it is clear that iGTP-MDC generally gives more accurate trees than Phylonet-MDC run in its heuristic mode; however, when the exact version of Phylonet-MDC can be run, it produces more accurate trees than its heuristic version, and also more accurate trees than iGTP-MDC. The reason for this is likely due to the improved MDC scores produced by using the exact version of Phylonet-MDC (which are mathematically guaranteed), compared to the other methods. It is worth noting that the substantial reduction in topological accuracy by using the heuristic version instead of the exact version of Phylonet-MDC is almost certainly a result of the fact that *all* the gene trees are incomplete, with randomly deleted taxa. This greatly impairs the ability of Phylonet-MDC's heuristic to score trees that are topologically similar to the true tree, since all the clades in any estimated tree must be drawn from the input gene tree clades in this case. However, the heuristic used in iGTP-MDC explicitly searches through treespace and so is not impaired in the same way. Given that previous research [152] has shown very good trees resulting from Phylonet-MDC's heuristic version when the input gene trees are all complete, it seems likely that Phylonet-MDC might give better results when the taxon deletion is not random, or when at least some of the gene trees are based on complete taxon sets. Thus, although this study showed poor accuracy for Phylonet-MDC's heuristic, this trend may not hold under other circumstances, including those that might better represent systematic practice. Future work will investigate this possibility.

We also note that contracting low support branches in estimated gene trees typically (but not always) benefited Phylonet-MDC and MRP, but not the iGTP methods. This difference is likely due to differences in the treatment of unresolved gene trees within the iGTP, Phylonet-MDC, and MRP software. For example, it seems likely that iGTP-MDC and Phylonet-MDC do not score proposed species trees identically when the input gene trees are unresolved (Phylonet-MDC scores species trees with respect to optimal refinements of unresolved gene trees [155], a guarantee that may not be true of iGTP-MDC).

This study establishes that there is currently no computationally feasible solution for estimating highly accurate species trees from incomplete gene trees for large numbers of taxa. That is, only *BEAST was able to produce highly accurate species trees; all other methods had much higher error rates. Therefore, for small enough numbers of taxa so that *BEAST can be run properly without huge running times, very accurate species trees can be computed. Although this study did not investigate the feasibility of running *BEAST on larger datasets, other studies with Bayesian methods have shown that proper analyses of datasets (even small ones) can take weeks of analysis to reach convergence [152]. Therefore, the poor results of the other methods on larger datasets suggests that highly accurate species tree estimations from incomplete gene trees and alignments may be beyond what current methods can achieve.

This study also suggests some limitations to analyses based on MDC. Unsurprisingly, we saw that optimizing MDC generally gave better results than optimizing duplications or duplications and losses. We also observed (as had been noted earlier in [152]) that optimizing the total number of duplications and losses produced more accurate trees than optimizing duplications alone.

Finally, and perhaps most interestingly, we noted that optimizing MDC produced generally less accurate trees than optimizing MRP. This is a very interesting result, given that MRP is agnostic about the cause of incongruence between gene trees, and MDC explicitly addresses ILS as the cause for incongruence. However, there is no mathematical explanation for why MRP would perform well, and so this remains only an empirical observation.

This study shows that the standard heuristics (the parsimony ratchet as implemented in PAUP*) for the supertree method MRP produces highly accurate species tree estimations, even though it does not consider ILS, and can do so reasonably quickly, even on large datasets. These observations, combined with the observation that none of the methods we studied (other than *BEAST) that explicitly take into account events such as ILS or duplication and loss produced trees as accurate as MRP, suggest that optimizing MRP *may* be a reasonable approach to species tree estimation for large datasets, when statistical methods (such as *BEAST) cannot be run for computational reasons. Therefore, other supertree methods, such as SuperFine [101, 102, 136], a new supertree method that has been shown to produce better MRP scores and more accurate trees than standard MRP heuristics (while also being faster than standard MRP heuristics), should also be investigated. Finally, for complete gene trees, the greedy consensus produced highly accurate species trees, despite not being statistically consistent (at least when used on rooted gene trees, as shown in [26]). Given this observation, other consensus methods (see [12], [63], and [113] for some entries into the literature on consensus methods) might also be useful for estimating species trees for large numbers of taxa.

6.7 Conclusion

Species tree estimation from incomplete gene trees that can differ from the true species tree due to ILS presents many interesting theoretical and empirical challenges: excellent results can be obtained using *BEAST, a statistical approach that explicitly models the processes that cause incongruence between gene trees and species trees, but *BEAST is too computationally intensive for even moderately large datasets. In contrast, a very simple supertree method, MRP, is able to provide reasonably good results on very large datasets, even though it does not provide statistical guarantees. Thus, while it seems likely that methods based on sound statistical models will produce the most accurate species trees, the current methods that can analyze incomplete gene trees are either limited to small datasets, or are not based upon statistical models of gene tree incongruence. Given the increased use of multi-marker analyses for species tree estimation, methods that are both statistically-based and can run on large datasets (and can analyze incomplete gene datasets), are likely to have high impact. Future work will hopefully produce methods that are scalable and statistically-based, and that produce highly accurate trees on datasets with large, incomplete gene trees.

Chapter 7

Naive Binning

Species tree estimation in the presence of incomplete lineage sorting (ILS) is a major challenge for phylogenomic analysis. Although many methods have been developed for this problem, little is understood about the relative performance of these methods when estimated gene trees are poorly estimated, due to inadequate phylogenetic signal.

We explored the performance of some methods for estimating species trees from multiple markers on simulated datasets in which gene trees differed from the species tree due to ILS. We included *BEAST, concatenated analysis, and several summary methods: BUCKy, MP-EST, MDC, MRP, and the greedy consensus. We found that *BEAST and concatenation gave excellent results, often with substantially improved accuracy over the other methods. We observed that *BEAST's accuracy is largely due to its ability to co-estimate the gene trees and species tree. However, *BEAST is computationally inten-

Much of the material in this chapter is taken without alteration from the following paper.

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

MSB and TW designed the study; MSB performed the study; MSB and TW analyzed the data, and wrote the paper.

sive, making it challenging to run on datasets with 100 or more genes or with more than 20 taxa.

We propose a new approach to species tree estimation in which the genes are partitioned into sets, and the species tree is estimated from the resultant "supergenes." We show that this technique improves the scalability of *BEAST without affecting its accuracy and improves the accuracy of the summary methods. Thus, naive binning can improve phylogenomic analysis in the presence of ILS. We also developed an improved version of the binning technique called weighted statistical binning (WSB) [5], which we will describe in Chapter 8.

7.1 Introduction

Species tree estimation from multiple genes is often performed using concatenation (also called "combined analysis"): alignments are estimated for each gene and concatenated into a supermatrix, which is then used to estimate the species tree. When gene trees are identical, concatenation can give very accurate results; however, this approach to species tree estimation is potentially problematic when gene trees differ from the species tree (and hence from each other) due to several biological factors, including gene duplication and loss, horizontal gene transfer, and incomplete lineage sorting.

The best studied of these problems is species tree estimation in the presence of incomplete lineage sorting (ILS), which is based on the multi-species coalescent [146]. Many methods have been developed to estimate species trees in the presence of ILS, beginning with the MDC (minimize deep coalescence) approach suggested in [84], and now including many different types of methods (see [28, 67] for a discussion of some methods). Some of these new methods (for example, MP-EST [80] and the population tree from BUCKy [1, 73]) have been proven to be statistically consistent in the presence of ILS. In contrast, the greedy consensus, majority consensus, the concordance tree from BUCKy, and MDC [1, 26, 142] can be inconsistent in the presence of ILS (i.e., there are some parameter settings under which these methods are inconsistent). The Bayesian method *BEAST [33] may produce a statistically consistent point estimate (e.g., the MAP tree) of the species tree, but a formal proof has not yet been provided (however, see [131], which proves the statistical consistency of gene tree estimation using Bayesian MCMC methods). When true gene trees differ due to ILS, combined analyses can be statistically inconsistent [26, 119], and can produce incorrect estimates of the species tree, sometimes with high confidence [24, 36, 69, 73, 74, 80, 124].

As a result of these studies and the growing awareness that ILS can be present in many phylogenomic datasets, there is great interest in using ILS-based estimation of species trees instead of concatenated analysis [28, 35, 53, 67]. However, only a few studies have been published comparing ILS-based methods and even fewer have compared concatenated analyses to ILS-based methods. Performance in simulation has been mixed, with ILS-based methods outperforming concatenation in some cases but not all [24, 33, 36, 74, 80]. The performance of ILS-based methods on biological datasets has also been mixed, with concatenation often producing trees with very high bootstrap support that may not be completely correct, but ILS-based methods often producing trees with very low bootstrap support [88, 129]. Thus, we still do not know very much about the relative performance of ILS-based methods, how they compare to methods (such as concatenation) that do not take ILS into account, and what factors impact the absolute and relative performance of methods.

In this chapter, we report on a simulation study to evaluate a collection of methods for estimating species trees and gene trees in the presence of ILS. Our simulation study includes datasets generated under three model conditions from prior studies [19, 155]. One model condition has 17-taxon datasets that evolve under a strong molecular clock, and the other two model conditions have 11-taxon datasets that do not evolve under a clock. The amount of ILS varies between the three model conditions, ranging from relatively low amounts to very high amounts. Finally, estimated gene trees on these datasets have low average bootstrap support due to insufficient phylogenetic signal, reflecting conditions often encountered when sampling genes from throughout the genome. We study a wide range of methods for estimating species trees from multiple markers, including *BEAST [33], both the population and concordance trees returned by BUCKy [1, 73], MP-EST [80], Phylonet-MDC [143, 155], greedy consensus (GC) (also called the extended majority consensus), matrix representation with parsimony (MRP) [3], and concatenation using maximum likelihood (CA-ML).

Our study revealed that many methods *have poor accuracy when the* *individual gene sequence alignments have low phylogenetic signal.* This vulnerability to poor signal affects all methods, but especially those that combine estimated gene trees; by comparison, *BEAST and CA-ML are relatively less impacted.

We developed an approach to address the vulnerability of species tree methods to low phylogenetic signal. We randomly partitioned the genes into subsets (which we call "supergenes"), estimated trees from these supergene alignments, and then used methods to estimate the species tree from the supergene trees. This approach did not produce statistically significant changes in accuracy on the 17-taxon datasets, but improved the accuracy of the trees estimated by combining estimated gene trees, often very substantially, on the 11-taxon datasets. Running *BEAST on the binned supergene alignments did not impact its accuracy, but did improve its scalability. Furthermore, when used with binning, several methods came close to being as accurate as *BEAST, while being orders of magnitude faster than *BEAST. Thus, this study suggests that highly accurate large-scale phylogenomic analyses may be achievable through a naive binning technique.

7.2 Materials and methods

See Appendix A for details.

Datasets: We used simulated 11-taxon [19] and 17-taxon [155] multi-gene datasets. The 11-taxon datasets have 100 genes, and the 17-taxon datasets have 32 genes. The protocols used in the two studies were fairly similar, however, the 11-taxon datasets reflect more heterogeneity, and hence are less idealized than the 17-taxon datasets. In each case, a model species tree was generated and a set of gene trees within each species tree (with one haploid individual sampled per species) produced under a coalescent process. This produces gene trees that can differ topologically from their associated species tree due to ILS. DNA sequences were then simulated down each gene tree under the Jukes-Cantor model. 100 replicates were generated for each model condition, and each replicate consisted of a set of true sequence alignments (i.e., one alignment for each gene).

The 11-taxon and 17-taxon datasets differ in some regards. First, the 17-taxon datasets evolved under a molecular clock, but the 11-taxon datasets did not. Second, the 11-taxon datasets have very short sequences (only 500 nucleotides), but the 17-taxon datasets have long sequences (2000 nucleotides). In the 11-taxon model conditions, there is substantial rate variation between the gene trees and species tree, but this is not true for the 17-taxon model conditions. Finally, the model conditions also varied in the amount of ILS, as we now discuss.

We calculated two statistics to evaluate the level of ILS in each model condition: the "average clade distance" between the true species tree and the true gene trees and the percentage of the true gene trees that have the same topology as the true species tree. The clade distance between two rooted trees (i.e., the rooted analog of the bipartition distance) is the total number of unique non-trivial clades (in one tree but not in both) divided by $2n-4$. Thus, if two rooted trees on 7 leaves share exactly 2 clades in common, the clade distance is 60%. Using this metric, the 17-taxon datasets have the highest amount of ILS (avg. clade distance 25.7%). The 11-taxon datasets came in two forms, one with somewhat lower (but still high) amounts of ILS (avg. clade distance 14.8%), and one with very low amounts of ILS (avg. clade distance 2.9%). We refer to the two 11-taxon models as strongILS and weakILS, accordingly. The percent of gene trees that match the species tree also fits with this relative ranking: 73.1% for the 11-taxon weakILS datasets, 21.3% for the 11-taxon strongILS datasets, and only 1.7% for the 17-taxon datasets. Thus, the 17-taxon datasets have extremely high levels of ILS, but the 11-taxon strongILS also have a high level of ILS.

Selecting subsets of genes. For the 17-taxon datasets, we used the provided 8-gene and 32-gene datasets; for the 11-taxon datasets, we sampled from the 100 genes to produce subsets with the desired number of genes.

Gene tree estimation: We compared *BEAST, RAxML v. 7.3.1 [130], and FastTree-2 [114], as gene tree estimators. We used 20 runs of RAxML on each of the alignments, and retained the tree with the best ML score; for FastTree-2, we used it with only one run (since it is deterministic, it is not improved by multiple runs). For *BEAST, we ran it as described below. We used RAxML with 400 bootstrap replicates for BUCKy and for Phylonet.

Species tree methods: We include *BEAST, MP-EST, BUCKy-pop, BUCKy-con, CA-ML, the greedy consensus (GC), Phylonet-MDC, and matrix representation with parsimony (MRP); see below for details. With the exception of *BEAST and CA-ML, these methods estimate the species tree by combining estimated gene trees; we refer to these as summary methods. For MP-EST, MRP, and GC, we use the binary gene trees as input (these methods either require binary gene trees or have not been shown to improve by contracting low support branches [152]).

We used *BEAST v. 1.6.2 [33] in its default setting, and used the default point estimates for the gene trees and species tree. For a given *BEAST analysis, we discarded the first 10% of the trees returned by the analysis, and then sampled one (1) out of each 1000 of the remaining trees. We ran *BEAST long enough to return ESS values that were large enough to suggest possible convergence. Even after 150 hours of analysis, the ESS statistics for *BEAST on the 11-taxon 100-gene strongILS datasets were very poor, suggesting that *BEAST had not converged; therefore, we omit results of *BEAST for these datasets.

We used MP-EST [80] in its default setting, using MAXROUND=100000, and with RAxML gene trees rooted at the provided outgroup.

We used BUCKy [1] with the default setting to compute two species tree estimations - the population tree (BUCKy-pop) and the concordance tree (BUCKy-con). We computed gene tree distributions using RAxML with bootstrapping and also using *BEAST as input to BUCKy. On each model condition and number of genes, we ran BUCKy using a sufficiently large number of MCMC iterations to reach sufficiently low standard deviations for the concordance factors to suggest possible convergence.

We used Phylonet v. 2.4 [143] for a version of the NP-hard MDC (Minimize Deep Coalescence) problem that takes gene tree branch support values into consideration. Although MDC is not statistically consistent [142], Phylonet-MDC can produce highly accurate species trees [152] when applied to gene trees in which all the low support branches are collapsed. Phylonet provides a technique to solve this version of MDC exactly, even for unrooted gene trees [7, 155], which can be used on datasets with a small enough number of taxa; we used this exact method for MDC for the 11-taxon datasets, and Phylonet's heuristic method (which restricts the solution space to those trees all of whose bipartitions come from the input set of trees) for the 17-taxon datasets. We used Phylonet on the ML gene trees with all branches having bootstrap support less than 75% collapsed.

We used PAUP* to estimate MRP (matrix representation with parsimony), using the standard heuristic search, and also to compute a greedy consensus (GC) (also called the "extended majority consensus") of the estimated gene trees. Both of these analyses are performed on the binary gene trees estimated by maximum likelihood. We also studied CA-ML, using RAxML to infer a species trees from the super-alignment (without partitioning), and using 10 independent runs (-N 10).

Criteria: We report tree error using the missing branch rate (also known as the FN or "false negative" rate), which is the proportion of internal branches in the true tree defining bipartitions that are missing in the estimated tree. The use of FN rates rather than Robinson-Foulds (RF) rates is due to the observation that some of the methods for estimating trees produce unresolved trees, and the RF rates would be biased in favor of these methods [117]. We tested for statistical significance using the Wilcoxon signed rank test.

Experiments: The first experiment compared the "fast" methods (all methods except *BEAST and BUCKy) on 100 replicates of the 11-taxon and 17-taxon datasets, varying the number of genes, using RAxML to estimate gene trees. The second experiment compared the full set of methods on 20 replicates of these model conditions, again using RAxML to estimate gene trees. We explored the accuracy of gene trees estimated by RAxML, FastTree, and *BEAST in the third experiment. The fourth experiment evaluated the accuracy of species trees computed for gene trees estimated by *BEAST. The fifth experiment then examined the impact of binning genes into supergenes, using a simple "naive" binning technique.

7.3 Results

We show results evaluating computational aspects of the different methods, and then results of the five basic experiments exploring accuracy. See Appendix A for additional details.

Computational issues. The phylogenomic pipelines we studied differed dramatically in terms of their running times, making some methods infeasible to use on some datasets within the limits of this study. Due to space limitations, we present a brief discussion of the computational requirements of the different methods, and direct the interested reader to Appendix A for full results.

Pipelines that used *BEAST took the most time, with running times of 80-150 hours for the 50-gene datasets with 11 taxa; analyses of the 100-gene datasets with 11 taxa did not converge, even in 150 hours. The pipelines with BUCKy, when used with distributions computed using RAxML bootstrapping, took up to 5 hours, but were able to be run on even the 100-gene 11-taxon datasets. Pipelines with Phylonet when used with the RAxML bootstrap trees (restricted to the high support edges) took up to 2 hours per dataset (almost all of that for running RAxML). Pipelines with MRP, GC, and CA-ML took just a few minutes per dataset.

Because of these computational issues, we only ran BEAST on unbinned datasets with at most 50 genes (and even these were very computationally intensive). We also did not run *BEAST or unbinned BUCKy on more than 20 replicates for any model condition. Therefore, in the remaining study, we show results for the "fast" methods (everything but *BEAST and BUCKy) on 100 replicates of the model conditions, and we examine *BEAST and BUCKy on only 20 replicates of the model conditions. We do, however, show results using BUCKy with binning on 100 replicates of some model conditions. In total, we estimate that we used at least 5000 CPU hours, just for the *BEAST runs.

Experiment 1: The first experiment explored the accuracy of the "fast" methods for estimating species trees, i.e., CA-ML, MP-EST, MRP, Phylonet, and GC; see Figures 7.1, 7.2, and 7.3. CA-ML had the best accuracy, with very large improvements over other methods on the 11-taxon datasets and small improvements on the 17-taxon datasets. All the improvements are statistically significant: $p < 0.003$ for the 11-taxon strongILS with up to 100 genes and the 11-taxon weakILS with up to 25 genes, and $p \leq 0.04$ for the 17-taxon datasets.

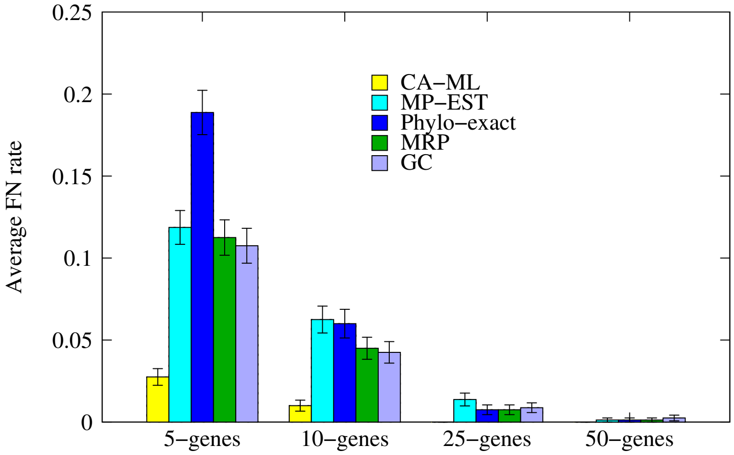


Figure 7.1: **Results for "fast" methods on 100 replicate 11-taxon weakILS models.** CA-ML uses just the input alignments, and the other methods use gene trees estimated using RAxML. We show average FN rates with standard error bars.

Experiment 2: We then evaluated BUCKy-pop, MP-EST, *BEAST, CA-ML, and BUCKy-con. Because *BEAST is computationally intensive, the

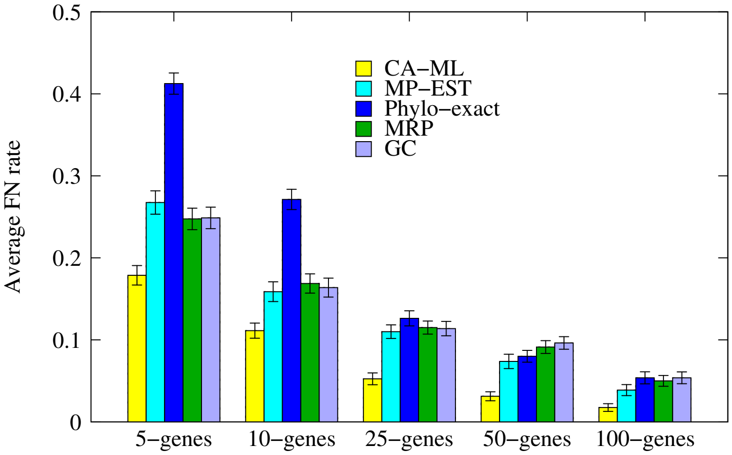


Figure 7.2: **Results for “fast” methods on 100 replicate 11-taxon strongILS models.** We show results (average FN rate with standard error bars) for up to 100 genes.

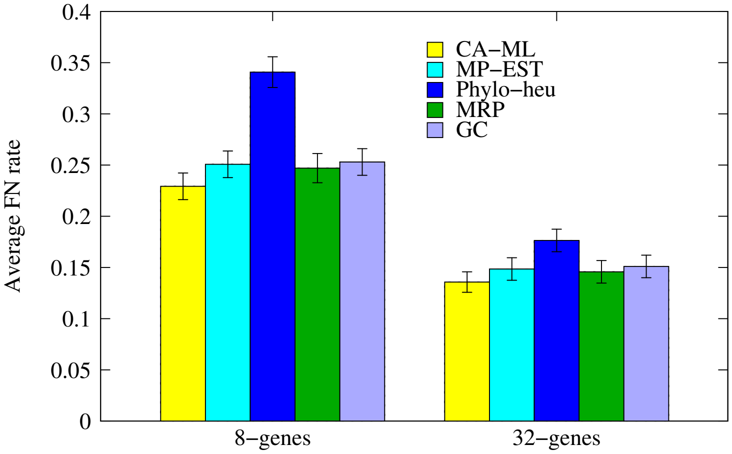


Figure 7.3: **Results for “fast” methods on 100 replicate 17-taxon models.** We show average FN rates with standard error bars.

analyses were limited to 20 replicates per datapoint. See Figures 7.4, 7.5, and 7.6.

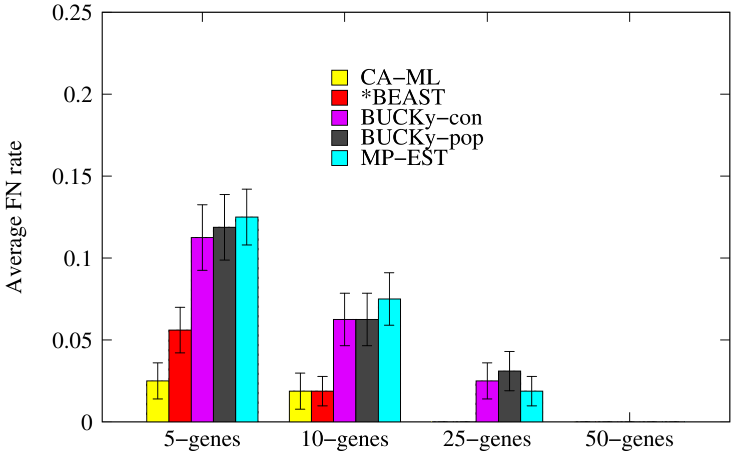


Figure 7.4: **Results for *BEAST, BUCKy, MP-EST, and CA-ML on 20 replicate 11-taxon weakILS datasets.** We show average FN rates with standard error bars. CA-ML and *BEAST return the true tree on the 25-gene case, and all methods shown return correct trees on the 50-gene case. Therefore, no results are shown for datasets with 100 genes.

Note that *BEAST and CA-ML are the two most accurate methods on these data, with the greatest improvement over the other methods on the 11-taxon weakILS datasets and the least improvement on the 17-taxon datasets. The relative performance between CA-ML and *BEAST varied, with CA-ML better in some cases and worse in others, and often the difference was small.

BUCKy-pop is in third place, and even matched the accuracy of *BEAST on the 11-taxon strongILS datasets with 25 genes. A comparison between BUCKy-pop and BUCKy-con shows that they had very close accuracy in most cases, but that BUCKy-pop was sometimes more accurate than BUCKy-con

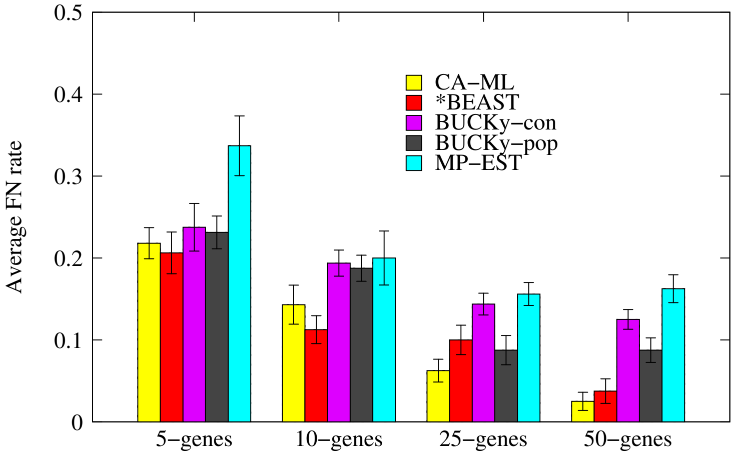


Figure 7.5: **Results for *BEAST, BUCKy, MP-EST, and CA-ML on 20 replicate 11-taxon strongILS datasets.** We show average FN rates with standard error bars.

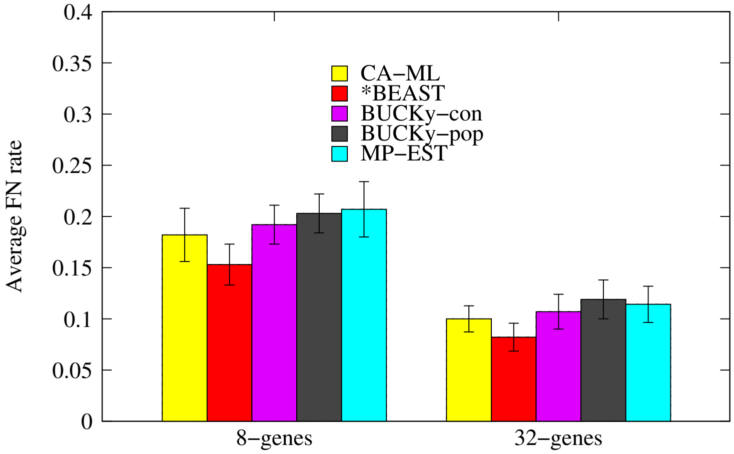


Figure 7.6: **Results for *BEAST, BUCKy, MP-EST, and CA-ML on 20 replicate 17-taxon datasets.** We show average FN rates with standard error bars.

(e.g., on the 11-taxon strongILS datasets with 25 or 50 genes), with statistically significant differences ($p = 0.003$ and $p = 0.035$, respectively).

Of these various observations, the most important here are the following: CA-ML and *BEAST had the best accuracy on these data; the gap between methods was least on the 17-taxon datasets, and greatest on the 11-taxon weakILS datasets; and all methods became less accurate with increases in the amount of ILS. It is easy to understand why the methods that are not statistically consistent under the multi-species coalescent model increase in error with the degree of ILS, but not that easy to understand why *BEAST, MP-EST, and BUCKy-pop decrease in accuracy with increases in ILS. Here we offer a possible explanation for this trend.

Recall that the conditions that favor ILS are very short branches in the species tree. Thus, the conditions that increase the amount of ILS (i.e., short branches) also make it challenging to estimate the gene trees. In fact, the weakILS model trees have long branches (and are called "LB" in [19]), and the strongILS model trees have short branches (and are called "SB" in [19]), and gene trees estimated using RAxML have lower error on the 11-taxon weakILS model conditions than on the strongILS model conditions (30% vs. 40%, respectively). Therefore, it's not at all surprising that species trees estimated by combining gene trees under the highILS model conditions would have higher error than species trees estimated by combining gene trees under the lowILS model condition. Finally, the 17-taxon datasets had the highest level of ILS, and on these data the summary methods perform the worst. Note that this vulnerability applies to all summary methods, even to the statistically consistent methods like MP-EST and BUCKy-pop.

Experiments 3 and 4: The next two experiments attempted to understand why *BEAST was so much more accurate than the summary methods. In Experiment 3, we evaluated the accuracy of the gene trees estimated by *BEAST, FastTree-2, and RAxML for all three model conditions, and observed that *BEAST produces substantially more accurate gene trees than FastTree-2 and RAxML. For example, under the 11-taxon weakILS model condition with 50 genes, gene trees estimated by *BEAST had only 3.3% error while gene trees estimated by RAxML had 31.9% error - a reduction of roughly 90%. More generally, the greatest improvement was for the model condition with the lowest rate of ILS (11-taxon weakILS), and the least improvement was for the model condition with the highest rate of ILS (17-taxon datasets). However, even on the 17-taxon datasets, the reduction was at least 50%. Results for the 17-taxon datasets are given in Figure 7.7; see Appendix A for the other results. These analyses also show that RAxML has a small but statistically significant advantage over FastTree (differences in missing branch rate of at most 1.7% on the 11-taxon weakILS conditions, 2.5% on the 11-taxon strongILS conditions, and 1.1% on the 17-taxon conditions).

In Experiment 4 (see Appendix A), we examine the results of using the summary methods (i.e., BUCKy-con, BUCKy-pop, Phylo-MDC, MP-EST, MRP, and GC) on inputs of gene trees estimated by *BEAST. These experi-

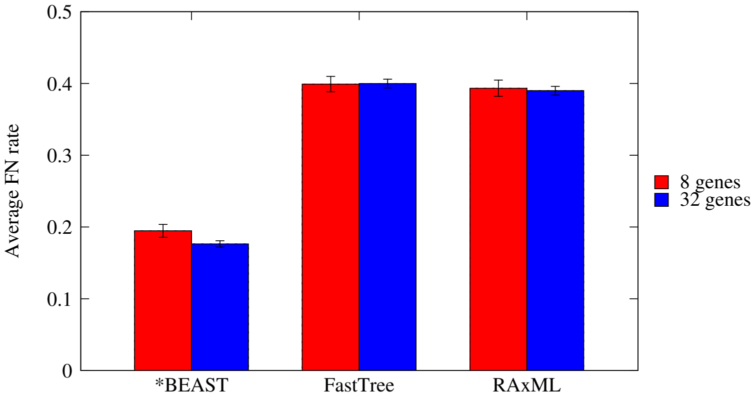


Figure 7.7: **Gene tree estimation error rates on 17-taxon datasets.** Average and standard error bars (over 20 replicates) of *BEAST, RAxML, and FastTree-2.

ments show that species trees estimated by combining gene trees estimated by *BEAST are essentially as accurate as the species trees estimated by *BEAST, and there are no statistically significant differences. This suggests that the accuracy obtained by *BEAST is primarily due to its improved gene tree accuracy, rather than to some sophisticated way of combining accurate gene trees.

Experiment 5: Since reduced phylogenetic signal in individual gene sequence alignments impacts the summary methods, we considered the following approach.

- Step 1: Partition the genes into bins,
- Step 2: Within each bin, compute a "supergene" alignment, by concatenating the alignments for the genes in the bin,

- Step 3: Compute a “supergene tree” using ML on each supergene alignment, and

- Step 4: Estimate the species tree from the set of supergene trees (using one of the summary methods), or from the set of supergene alignments (using *BEAST, for example).

Since this binning technique can put genes into the same bin that may not share the same history, this approach is a blend of CA-ML and the species tree estimation technique used in Step 4.

Our motivation for this approach is empirical. The hope is that since each supergene has more sites, ML trees estimated on each supergene might be more resolved than ML trees estimated on the individual genes. If the genes placed in the same bin have the same gene tree topology, then this approach could potentially lead to higher accuracy gene trees. If the genes placed in the same bin have different gene tree topologies, then they may not represent any gene tree that appears in the dataset, but may be closer to the species tree. In either case, summary methods applied to these supergene trees might be more accurate than summary methods applied to the individual gene trees.

Evaluating binning on fast methods. In our initial experiment, we explored the impact of binning on the fast methods on 100 replicate datasets of each model condition. We used bins with 5 genes each for the 11-taxon datasets, and bins with 4 genes each for the 17-taxon datasets. We do not present results for *BEAST (unbinned or binned) or BUCKy on unbinned datasets due to computational issues; however, we do show results for BUCKy on binned datasets. Note also that because we ran CA-ML *without partitioning*, binning has no impact on CA-ML.

Results for the 11-taxon strongILS datasets are shown in Figures 7.8, 7.9, and 7.10. See Appendix A for results on the 11-taxon weakILS datasets and 17-taxon 32-gene datasets. Binning improved accuracy for all methods for the 11-taxon datasets (both weakILS and strongILS), but not always statistically significantly. Results on the 17-taxon datasets showed that binning did not have any statistically significant impact on any method ($p > 0.22$).

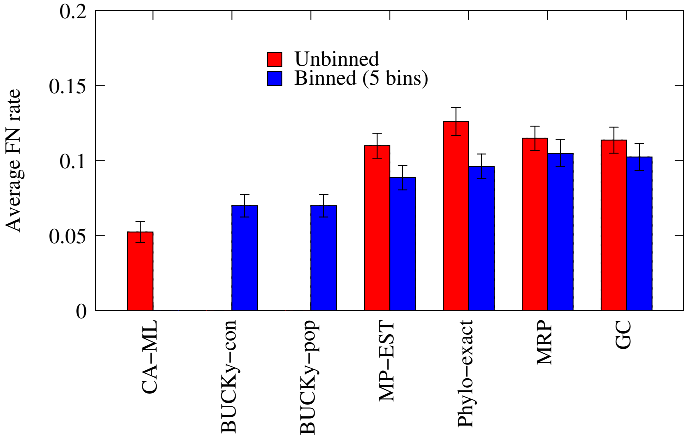


Figure 7.8: **Results of binning experiments of the fast methods on 100 replicates of the 11-taxon 25-gene strongILS datasets.** We show average FN rate with standard error bars. Each bin contains 5 genes. We omit BUCKy on unbinned genes and *BEAST (binned or unbinned) because these are too slow to run on all 100 replicates within our time limits. CA-ML is not impacted by binning because it uses an unpartitioned analysis.

On the 11-taxon weakILS datasets with 25 genes, all methods improved

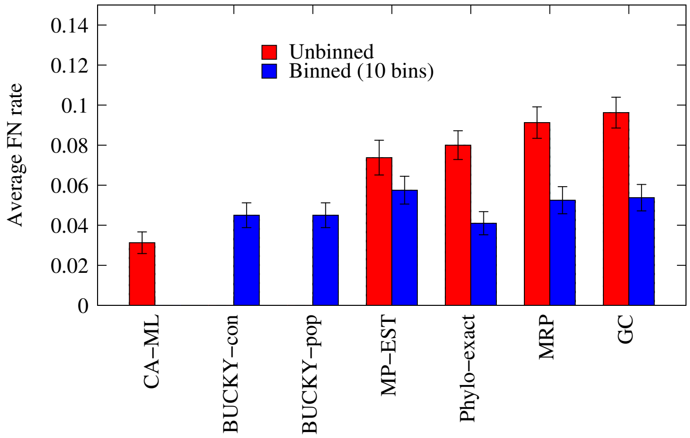


Figure 7.9: **Results of the binning experiment for the fast methods on 100 replicates of the 11-taxon 50-gene strongILS datasets.** We show average FN rates with standard error bars. Each bin contains 5 genes. We omit BUCKy on unbinned genes and *BEAST (whether binned or unbinned) because these are too slow to run on all 100 replicates within our time limits. CA-ML is not impacted by binning, because it uses an unpartitioned analysis.

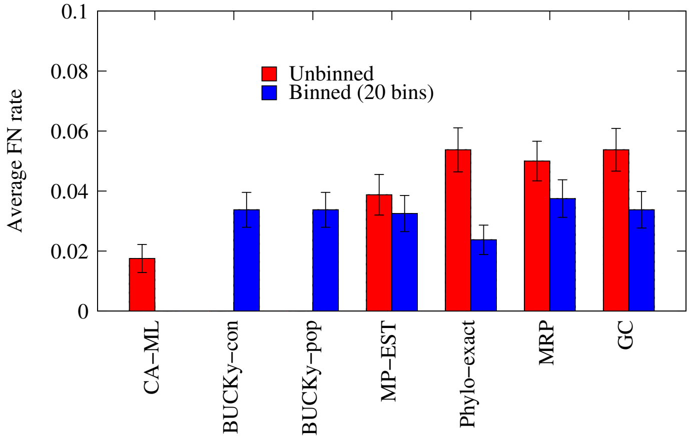


Figure 7.10: **Results of the binning experiment for the fast methods on 100 replicates of the 11-taxon 100-gene strongILS datasets.** We show average FN rates with standard error bars. Each bin contains 5 genes. We omit BUCKy on unbinned genes and *BEAST (whether binned or unbinned) because these are too slow to run on all 100 replicates within our time limits. CA-ML is not impacted by binning because it uses an unpartitioned analysis.

in accuracy. These improvements were statistically significant for MP-EST and Phylonet ($p = 0.002$ and $p = 0.016$, respectively), but not for the other summary methods. However, all methods were already highly accurate without binning.

Binning produced large reductions in error for many methods on the 11-taxon strongILS datasets with 25 genes. Phylonet-MDC showed the largest improvement (reduction from 12.6% to 9.6%, $p = 0.002$), MP-EST showed the second largest improvement (reduction from 11.0% to 8.8%, $p = 0.021$), and GC and MRP showed the least improvement (reductions of at least 1%, but not statistically significant, $p = 0.16$ and $p = 0.18$, respectively).

On 50 genes, all methods had reductions in error, with Phylonet-MDC showing the largest improvement (reduction from 8.9% to 4.1%, $p < 10^{-5}$), GC showing the next largest improvement (reduction from 9.6% to 5.4%, $p < 10^{-4}$), MRP the next largest improvement (reduction from 9.1% to 5.3%, $p < 10^{-4}$), and MP-EST with the smallest improvement (reduction from 7.3% to 5.7%, but not statistically significant, $p = 0.057$).

On 100 genes, all methods had reductions in error, and again Phylonet-MDC had the largest improvement (reduction from 5.4% to 2.4%, $p < 10^{-3}$), GC had the next largest (reduction from 5.4% to 3.4%, $p = 0.007$), and MRP and MP-EST showing smaller improvements that were not statistically significant ($p > 0.07$).

Thus, binning improved the accuracy of all methods on the 11-taxon model conditions, with large reductions for the strongILS conditions and smaller (but still significant) reductions on the weakILS conditions. The greatest improvements were for intermediate numbers of genes, in which the methods used without binning still had some error (and hence could be improved), but had enough genes so that binning produced a reasonable number of supergenes. Binning had no statistically significant impact on the 17-taxon model conditions with 100 replicates ($p > 0.22$). CA-ML was still the most accurate of all tested methods, but some methods came close to the accuracy of CA-ML when used with binning.

Evaluating binning on all methods. Due to the computational effort in using *BEAST, we limited the analysis to only 20 replicates of each model condition. We limit this discussion to the impact of binning on *BEAST and BUCKy, since the analysis on 100 replicate datasets allowed us to evaluate binning on the other methods with a higher number of replicates. Results on the 11-taxon weakILS datasets with 25 genes are shown in Appendix A; all methods improved, but the improvement was statistically significant only for BUCKy-pop (reduction from 3.1% to 0.0%, $p = 0.03$). Results on the 20-replicate 17-taxon datasets (see Appendix A) show no statistically significant differences ($p > 0.3$) for BUCKy-pop, BUCKy-con, and *BEAST, and all differences were very small (at most 0.5%). Results on 11-taxon strongILS datasets are shown in Figures 7.11 and 7.12. BUCKy-pop generally improved with binning, but the results were not statistically significant ($p > 0.06$).

BUCKy-con also improved using binning (reduction in error from 14.3% to 9.4% on 25 genes, from 12.5% to 5% on 50 genes, and from 5.6% to 2.5% on 100 genes), and the changes on 25 and 50 genes were statistically significant ($p = 0.018$ and $p = 0.005$, respectively).

The impact of binning on *BEAST is interesting. On the 100-gene datasets, we were unable to run *BEAST to convergence without binning even with 150 hours of analysis; however, *BEAST was able to reach acceptable ESS values in only 10 hours using 4 threads when run on 20 bins with 5 genes each. Thus, the use of binning did not impact the accuracy of *BEAST, but it made it feasible to use *BEAST on datasets with large numbers of genes.

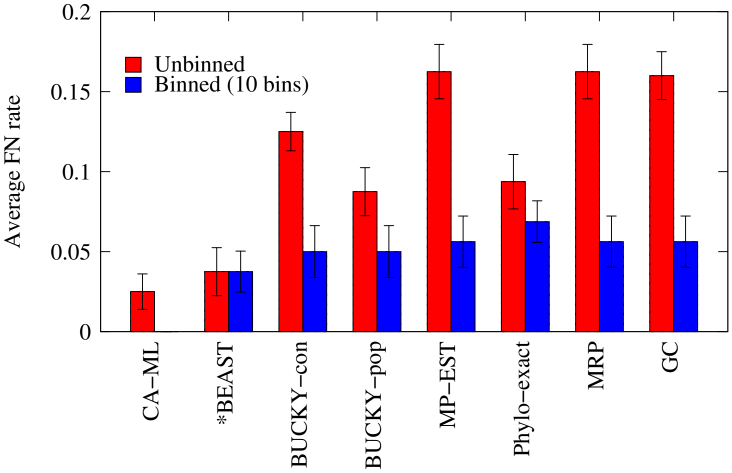


Figure 7.11: **Results of the binning experiment for all methods on 20 replicate 11-taxon 50-gene strongILS datasets.** CA-ML is not impacted by binning since it is an unpartitioned analysis. Each bin contains 5 genes. Average and standard error bars shown.

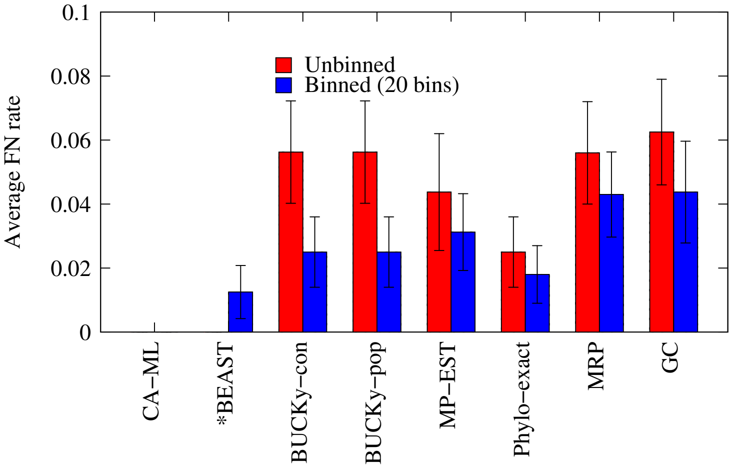


Figure 7.12: **Results of the binning experiment for all methods (except *BEAST) on 20 replicate 11-taxon 100-gene strongILS datasets.** Each bin contains 5 genes. Average and standard error bars shown. We omit *BEAST on unbinned genes because it could not run to convergence on this dataset within the time limit; however, we show results for *BEAST on the binned datasets. CA-ML returns the true tree on these data.

7.4 Discussion

The main purpose of this study was to evaluate methods for estimating species trees in the presence of ILS under realistic conditions. Since many real world phylogenomic analyses have to contend with genes with poor phylogenetic signal [124], we specifically examined conditions in which estimated gene trees were only partially resolved. As expected, the number of genes and amount of ILS impacted the accuracy of the methods we tested, so that all methods returned more accurate trees with increasing numbers of genes and decreasing levels of ILS. However, in addition to these expected results, we make the following observations:

First, all the summary methods we studied were impacted by gene tree estimation error. In contrast, although *BEAST and CA-ML were also affected by the amount of phylogenetic signal in the multiple sequence alignments, the impact was generally less.

Second, CA-ML and *BEAST had similar accuracy, and were generally more accurate than the summary methods we tested.

Third, *BEAST produced dramatically more accurate gene trees than ML analyses on the alignments, and summary methods on these gene trees produced species trees as accurate as *BEAST species trees, explaining why *BEAST produces more accurate species trees than other methods.

Fourth, the naive binning technique we tested generally improved coalescent-based methods. It improved the scalability of *BEAST without impacting its accuracy, making it feasible to use *BEAST on datasets with many genes. Binning also improved the accuracy of species trees estimated using the summary methods we tested on the 11-taxon conditions, although the degree of impact depended on the number of genes and the level of ILS. Finally, binning had no statistically significant impact on the 17-taxon conditions.

The observation that summary methods are vulnerable to poor phylogenetic signal in the gene sequences is consistent with the empirical studies reported by [124] and [88], and this study would seem to suggest that naive binning would be helpful for species tree estimation under these circumstances. However, naive binning could have unforeseen negative consequences if it puts genes with different histories into the same bin. The good performance of naive binning under the 11-taxon strongILS condition we explored suggests that it may be somewhat robust in practice, even under relatively high rates of ILS. However, since naive binning did not improve the accuracy of summary methods on the 17-taxon datasets (which had the highest rate of ILS), this suggests that naive binning could reduce accuracy when the amount of ILS is very large. There is also a possibility that binning will only be helpful when concatenation is more accurate than the coalescent-based methods. Therefore, further research is needed in order to assess the conditions in which binning improves or reduces accuracy.

One of the most interesting results in this study is the observation that CA-ML outperformed all the ILS-based methods that operate by combining estimated gene trees. This observation would seem to run counter to other simulation studies that have shown that concatenation can return incorrect trees with high confidence and can also produce trees that are less accurate than trees estimated by ILS-based methods [24, 33, 36, 73, 74, 80]. However, these studies used simulated datasets that evolve under a strong molecular clock (a condition that may benefit some coalescent-based methods more than concatenation [24]), few taxa, and generally had many genes relative to the number of taxa (and estimated gene trees on these alignments may have been fairly accurate). In contrast, our study had 11- and 17-taxon datasets, at most 100 genes, and poorly estimated gene trees. Thus, it seems that there are conditions under which some ILS-based methods might outperform CA-ML, and other conditions under which CA-ML might outperform the ILS-based methods. In particular, it is possible that the critical issue is the number of genes, and that ILS-based methods will have better accuracy than concatenation when the number of genes is large enough. Clearly, further research is needed in order to understand which conditions favor each type of approach.

7.5 Additional discussion

7.5.1 Previous studies comparing concatenation to coalescent-based estimation of species trees

One of the interesting results in this study is that concatenation using maximum likelihood produced better results than the summary coalescent-based methods, and was often more accurate than *BEAST. Since this result seems to run counter to the literature about coalescent-based methods, we discuss this in some detail.

While many papers have used simulations to evaluate coalescent-based methods, most of these papers only compared coalescent-based methods to each other, rather than to concatenation. Thus, to the best of our knowledge, only [24, 33, 36, 69, 73, 74, 80] present results of simulation studies that compare concatenated analysis (either based on a Bayesian or a maximum likelihood method) to coalescent-based methods. We discuss each of these in turn.

[24]: This study introduces Supermatrix Rooted Triplets (SMRT), a coalescent-based method that is statistically consistent under the multi-species coalescent model when sequences evolve under the two-state CFN molecular clock model. They compare SMRT to maximum likelihood in an extensive simulation study with model trees having at most 6 taxa (most have only 4 or 5 taxa). Almost all of the simulations were performed under a strong molecular clock. In their simulations, concatenation was generally, but not always, outperformed by SMRT. However, the relative performance was clearly impacted by the amount of ILS (as determined by parameter settings), with concatenation performing as well (or better) when ILS was very low. The relative performance was also impacted by the number of genes, so that under some models where SMRT outperformed concatenation for large numbers of genes, concatenation outperformed SMRT for small numbers of genes. They also explored the impact of violating the molecular clock in the simulation, but inferring under the clock; this study showed that concatenation was less impacted by the model violation than SMRT.

The most interesting part of this analysis is that it showed that the relative performance of concatenation using maximum likelihood and SMRT depended on several conditions, including whether sequences evolved under a strong molecular clock, the amount of ILS, and the number of genes.

[74]: This paper reports on a very extensive comparison several coalescent-based methods (STEM, BUCKy, and BEST) to two concatenation methods (one using MrBayes and one using maximum parsimony implemented in PAUP*) on 5-taxon model species trees. Sequence evolution on each gene was under Jukes-Cantor with a strong molecular clock, and produced sequences of length 1000 bp. They also report the percentage of time the true tree is returned by the given analysis.

One focus of their study was evaluating the impact of the model tree topology (balanced vs. unbalanced) on the relative performance of methods; they observed that BEST generally had the highest accuracy on the asymmetric model species trees and BUCKy generally had the best accuracy on the symmetric model species trees. There were, however, some model conditions (reflecting the amount of ILS) in which MrBayes was either first or tied for first, and many conditions in which MrBayes was only slightly less accurate than BEST and BUCKy.

[73]: This paper presents a comparison of concatenated analysis using a consensus tree output by MrBayes [55] to the BUCKy-pop and BUCKy-con trees, on three model conditions with rooted species trees and 5 taxa. Every model species tree has the strong molecular clock, and sequences with 500 bp evolve under the Jukes-Cantor model. They report only the percentage of times that each method recovers the true tree exactly. Two of the three models are in the anomaly zone, and one of these is in the "too greedy" zone. The analysis shows that BUCKy-pop generally had the best results of all three methods. Results on the easiest of the three model conditions show all methods had roughly the same accuracy (though BUCKy-pop does better at 10 and 30 genes than the other methods), and all methods converged to the true species tree at 100 genes. Results on the two trees in the anomaly zone distinctly show the improvement of BUCKy-pop over the other methods.

[80]: This paper presents the MP-EST method, and reports results for several simulation studies in which MP-EST is compared to other coalescent-based method. However, they also provide a simulation study comparing MP-EST and concatenation. The model tree here is a 5-taxon species tree in the anomaly zone, and sequences of length 500 evolve under the Jukes-Cantor model with the strong molecular clock. They report the frequency of returning the correct tree. Their study suggests that the two methods have roughly the same accuracy at the smallest number of genes they studied (100), but that MP-EST converges to the correct tree at 2500 genes, while Bayesian analysis

(MrBayes) converges to the wrong tree at 500 genes.

[36]: This paper introduced the coalescent-based method BEST, which co-estimates gene trees and species trees. They provide a simulation study comparing BEST to MrBayes from 30 genes that evolve within an 8-taxon model species tree. Sequence evolution on these genes is under the Jukes-Cantor model and a strong molecular clock and had 500 bp. For this analysis, they report that the species tree had 98% of the posterior probability under the BEST analysis, but that MrBayes converged to the wrong tree as the number of genes increased.

[33]: This paper introduced *BEAST, a method for co-estimating gene trees and species trees. They compared *BEAST to BEST (another coalescent-based co-estimation method) and also to BEAST, a Bayesian concatenation method for estimating species trees. They performed a simulation study using 7-taxon species trees with 4 genes that evolved under the Jukes-Cantor model and a strong molecular clock. The sequence alignments each had 1600 bp. They evaluated performance with respect to the how often the true species tree appeared in the 95% credible set of tree topologies. They observed that *BEAST had the best results, with BEST not too far below - but that BEAST had by far the worst accuracy.

Discussion: These studies clearly indicate that coalescent-based methods can be more likely to produce the true species tree than concatenation under some circumstances. However, all these studies shared some features: small numbers of taxa, generally large numbers of genes, and all genes evolving under a strong molecular clock. Some of these studies also primarily focused on model species trees in the anomaly zone. These features are likely to make it easier for coalescent-based methods (possibly especially ones that combine estimated gene trees) to perform better than concatenation-based methods that do not take ILS into account. For example, [24] observed that the presence of a strong molecular clock favors SMRT, a coalescent-based method that assumes the molecular clock; since many other coalescent-based methods assume the strong molecular clock, this would suggest that simulations under a strong molecular clock may be biased in favor of the coalescent-based methods. Also, summary methods (i.e., methods that combine estimated gene trees) are impacted by the accuracy of the estimated gene trees, and the simulation conditions in these studies may have all had sufficient sequence length and rates of evolution (relative to the number of taxa) to provide fairly accurate gene trees. Finally, most of these papers (though not all!) focused on accuracy on large numbers of genes, and the results in [24] show that the relative accuracy concatenation and coalescent-based methods can change with the number of genes (with concatenation sometimes being as good or better on small numbers of genes, but coalescent-based methods being better than concatenation on larger numbers of genes).

Taken as a whole, these studies do show that coalescent-based methods can be more accurate than concatenation. However, these studies primarily explored performance only for very small numbers of taxa, large numbers of genes, high amounts of ILS, and a strong molecular clock, while also demonstrating that these model conditions can impact the relative accuracy of concatenation and coalescent-based methods. Like these studies, our study focuses on performance under high amounts of ILS (the 11-taxon strongILS and 17-taxon conditions both have high amounts of ILS), and we also use sequences that evolved under the Jukes-Cantor model. However, there are several key difference between these studies and our study. First, we explore performance on small numbers of genes (at most 100) rather than on large numbers of genes. Second, our conditions produce estimated gene trees that are generally not that accurate as a result of inadequate sequence length, and we conjecture that the other studies had more accurate gene trees than our study. Third, the 11-taxon model conditions do not evolve sequences under a strong molecular clock. Fourth, we use 11-taxon and 17-taxon datasets instead of smaller datasets.

These differences may be sufficient to explain the different conclusions between this study and the others, but additional research will be needed to understand the impact of these model conditions on the relative accuracy of concatenation and coalescent-based estimation. Finally, we note that the performance criterion used in our study is different from that used in these other studies; they explored the percentage of the datasets in which the true species tree was recovered by each method, while we reported the average False Negative (missing branch) rate. While these criteria are equal for very small trees (4-taxon unrooted trees or 3-taxon rooted trees), they are not identical for larger trees, and it is possible that relative performance between two methods could change depending on the choice of criterion.

7.5.2 Limitations on binning

One of the findings of this study is that naive binning is helpful for coalescent-based methods. However, the conditions in which we explored the use of naive binning were either cases where concatenation was more accurate than binning (the 11-taxon datasets with not too many genes) or where the difference between concatenation and coalescent-based methods was very small (the 17-taxon datasets, and the 11-taxon datasets with sufficiently many genes so that all methods recovered the true tree). Therefore, it is possible that the naive binning technique we used is helping only because it creates a hybrid method that falls somewhere between concatenation and coalescent-based estimation, and therefore has accuracy that falls between these two.

In other words – does this naive binning technique help because it brings the coalescent-based method closer to concatenation, or does it help for some other reasons as well (such as addressing the vulnerability to poor signal gene trees)? Understanding the reasons that naive binning helps, and the conditions under which it helps, requires additional study.

We close our discussion with a basic question about phylogenetic estimation, suggested by this study. Given that summary methods are impacted by error in the estimated gene trees (resulting from inadequate phylogenetic signal in the sequence alignments), what is the optimal binning strategy? More generally, what is the best trade-off between data quantity (number of estimated gene trees) and quality (accuracy of estimated gene trees) for summary methods? Understanding the trade-off between data quantity and quality for each summary method will help inform binning strategies (e.g., how to pick the size of the bins), even if these strategies are statistically-based. This topic is subtle and statistically complex, and is only beginning to be studied, but see [53] for further discussion.

7.6 Conclusion

Under the conditions of our experiments (at least 11 taxa, at most 100 genes, and low signal per gene sequence alignment) we observed relatively poor species tree estimations using standard summary methods, and more accurate results from concatenation or from *BEAST, a method that co-estimates gene trees and species trees. However, the current co-estimation methods (including *BEAST) are computationally intensive and may not be feasible for use with more than 100 genes or more than 20 species. This study showed that a simple binning technique was able to make dramatic improvements in scalability for *BEAST, and generally improve the accuracy of summary methods, thus making some of these methods nearly as accurate as *BEAST.

This study should not be interpreted as recommending the use of naive binning, but instead as an indication of the potential for binning techniques to improve species tree estimation. For example, statistical techniques could be used to estimate whether a set of genes is likely to have a common tree, so that bins would only include genes expected to have a common history. Also, while concatenation performed well in this study, we conjecture that new techniques designed to handle markers with limited phylogenetic signal, might outperform concatenation even under these model conditions. Whether these new techniques will employ binning, or other ways of working with poorly estimated gene trees, the potential for substantial advances in species tree estimation could be great.

Chapter 8

Weighted Statistical Binning

Because biological processes can result in different loci having different evolutionary histories, species tree estimation requires multiple loci from across multiple genomes. While many processes can result in discord between gene trees and species trees, incomplete lineage sorting (ILS), modeled by the multi-species coalescent, is considered to be a dominant cause for gene tree heterogeneity. Coalescent-based methods have been developed to estimate species trees, many of which operate by combining estimated gene trees, and so are called summary methods. Because summary methods are generally fast (and much faster than more complicated coalescent-based methods that co-estimate gene trees and species trees), they have become very popular techniques for estimating species trees from multiple loci. However, recent studies have established that summary methods can have reduced accuracy in the

Much of the material in this chapter is taken without alteration from the following paper.

- M. S. Bayzid, S. Mirarab, B. Boussau, and T. Warnow. Weighted statistical binning: Enabling statistically consistent genome-scale phylogenetic analyses. *PLoS ONE*, 10(6):e0129183, 2015

SM and TW designed the study, MSB ran experiments on weighted statistical binning, SM and MSB ran experiments on statistical binning, BB generated the simulated datasets, all authors analyzed the data. MSB wrote the first complete draft of this paper, and TW, SM and BB wrote the final version.

presence of gene tree estimation error, and also that many biological datasets have substantial gene tree estimation error, so that summary methods may not be highly accurate in biologically realistic conditions.

Bayzid *et al.* [8] presented a novel technique called naive binning to address the problem of gene tree estimation error (discussed in Chapter 7). Mirarab *et al.* [90] presented the "statistical binning" technique, which is an improvement over naive binning, to improve gene tree estimation in multi-locus analyses, and showed that it improved the accuracy of MP-EST, one of the most popular coalescent-based summary methods. Statistical binning, which uses a simple heuristic to evaluate "combinability" and then uses the larger sets of genes to re-calculate gene trees, has good empirical performance, but using statistical binning within a phylogenomic pipeline does not have the desirable property of being *statistically consistent* [5]. We proposed a statistically consistent variant of binning technique called weighted statistical binning (WSB) [5]. However, with respect to the statistical consistency, the current mathematical theory does not suggest any advantage will be gained using WSB within a phylogenomic pipeline compared to an unbinned analysis (i.e., the use of the summary method without binning), because (unbinned) summary methods are also statistically consistent with sufficiently large numbers of true gene trees. Hence, the more important question is to investigating the empirical performance of WSB within a phylogenomic pipeline – how it impacts the accuracy of the estimated species trees; and so our study focused on whether WSB tends to increase or decrease the accuracy of summary methods, and how the model conditions impact the relative performance of binned and unbinned analyses.

In this chapter, we describe the phylogenomic pipeline based on weighted statistical binning. We report on an extensive experimental study (based on both simulated and biological datasets) that weighted statistical binning substantially improves the accuracy of phylogenomic analyses. We refer to [5, 89] for theoretical results on statistical consistency.

8.1 Introduction and background

Empirical studies suggest that summary methods are impacted by gene tree estimation error, and can produce less accurate estimated species trees than concatenation when gene tree estimation error is high enough (see [8, 41, 90, 111, 120] for examples of these studies on summary methods and further discussion). In a genome-scale analysis, it is unlikely that all the loci will have substantial phylogenetic signal, and so this vulnerability to gene tree estimation error means that coalescent-based summary methods may not be highly accurate techniques for estimating species trees from genome-scale data.

Bayzid *et al.* [8], and Mirarab *et al.* [90] proposed a new type of phylogenomic species tree estimation pipeline to handle gene tree estimation error, that has four steps instead of two (where the extra two steps are partitioning the genes into bins, and computing supergene trees for each bin using a fully partitioned maximum likelihood analysis). This pipeline showed very promising results when used with MP-EST. However, we did not address the theoretical properties of these pipelines, we only examined model trees with 37 or more species, and we only analyzed one coalescent-based summary method, MP-EST.

In this chapter, we report on an extended evaluation of statistical binning. Specifically,

- We describe a variant of statistical binning that we call weighted statistical binning.
- We evaluate the impact of statistical binning (both weighted statistical binning and the original unweighted statistical binning technique) on biological and simulated datasets. We examine pipelines using two coalescent-based summary methods, ASTRAL and MP-EST. We include results on simulated and biological datasets studied in [90], and also on additional simulated datasets with 10 and 15 taxa.

This study shows that weighted and unweighted statistical binning have very similar results across most datasets, and also that both ASTRAL and MP-EST tend to improve in accuracy when used with binning. However, there was one condition (characterized by a very high level of ILS, low average bootstrap support for the gene trees, and only ten species) in which statistical binning reduced accuracy for both MP-EST and ASTRAL. Thus, this study shows that binning is often beneficial, but also that there are some conditions under which binning can increase rather than decrease species tree error. Finally, we conclude with suggestions for further research.

8.2 Weighted statistical binning

The statistical binning technique presented in [90] operates as follows. The input is a multiple sequence alignment on each of p given genes, and a user-specified “threshold support” value $B < 1$. The role of the threshold B is to specify which branches in the gene trees are considered reliable, and which ones have support that is so low that the branches may be due to estimation error. Therefore, if the trees on two genes differ only in their low support edges, the differences are considered potentially consistent with estimation error, and the two genes are considered “combinable” or “compatible”.

Statistical binning computes maximum likelihood (ML) gene trees and bootstrap support on the branches for each gene, and then uses a simple heuristic based on bootstrap support values so that two genes can only be in the same bin if their ML gene trees do not have conflicting branches, each with bootstrap support of at least B. This is the combinability test, so that two genes are not considered combinable if they have highly supported conflicting branches, and otherwise are considered combinable. (Equivalently, two genes are combinable if their ML gene trees, after collapsing all branches with support less than B, share a common refinement.) Finally, because pairwise compatibility ensures setwise compatibility [48], if a set of gene trees *can* be all put in the same bin, then there is a tree that combines all the highly supported branches in any of the trees in the set.

Computing and using the incompatibility graph to bin the genes. The first step in statistical binning creates a graph based on the input, and uses a graph-theoretic optimization to bin the genes into subsets. Each gene is represented by a single vertex in the graph, and an edge is placed between two genes if their gene trees are not combinable, based on the heuristic described above. Determining if two genes are combinable can be computed in linear time [148], and so this graph, which we call the incompatibility graph, can be computed in time linear in the number of taxa and quadratic in the number of genes.

Since longer sequences tend to produce more accurate gene trees, having the bins be as large as possible is desirable; this is accomplished indirectly by seeking a coloring with as few colors as possible (i.e., a minimum vertex coloring), which is an NP-hard problem [64]. However, summary methods, such as MP-EST, use the distribution of the gene trees to estimate the species tree. Assuming gene tree reconstruction error only results in low-support branches, binning the genes so that the bins have nearly the same size means that the supergene tree frequency will be close to the true gene tree distribution (assuming that binning combines genes with the same tree, and that we can compute correct supergene trees). Note also that with such a constraint, frequent true gene tree topologies will be represented in several bins, while each of the rarest gene trees will be represented in a smaller number of bins (and perhaps in only one bin). Therefore, the objective is a coloring of the vertices, using a small number of colors, so that every color class contains about the same number of colors. To achieve such a coloring, [90] modified the Brélaz heuristic [10] for minimum vertex coloring, so that during the greedy coloring, a node is added to the smallest bin for which it has no conflicts. This coloring produces a partitioning of the vertices of the graph into subsets based on the color classes; thus, all vertices with the same color are in the same bin.

Computing a supergene tree for each bin. Once the vertex coloring is computed, the genes in a given color class form a bin, and their alignments are concatenated into a supergene alignment. Then, a maximum likelihood tree is computed (perhaps with bootstrapping) on each supergene alignment. For estimating supergenes, we use a *fully partitioned analysis* where each gene is assigned a separate partition, and all numeric model parameters are allowed to differ between partitions. We call the trees that are computed on the supergene alignments “supergene trees”. Because using a fully partitioned analysis is key to the theoretical guarantees of statistical binning, we specifically discuss this step in the pipeline.

Concatenated ML analyses of alignments from different loci can be performed in many ways, but their theoretical properties depend on the details of how they are performed, and in particular whether they are performed using an unpartitioned analysis, or a partitioned analysis. In an unpartitioned analysis, all the sites in the concatenated alignment are assumed to evolve down a single model tree (i.e., topology and numeric parameters), and the model tree maximizing the likelihood is sought for the matrix. In contrast, fully partitioned analyses of concatenated alignments assume that the different loci all evolve down the same tree topology, but allow the different parts within the concatenated alignment to have different values for all of the numeric parameters of the model. In the context of the GTR model, a fully partitioned maximum likelihood analysis would allow each locus to have its own 4×4 substitution matrix and gene tree branch lengths. Thus, if there are 10 loci within the concatenated alignment, a single tree topology is returned, but also ten different lengths for each branch, and ten different 4×4 substitution matrices. Fully partitioned and unpartitioned maximum likelihood analyses can result in different trees, and these analyses have very different theoretical properties; see the example provided in the Methods section, below.

Applying summary methods to the supergene trees. The supergene trees are used by a coalescent-based summary method (e.g., MP-EST) to estimate the species tree. In other words, by recalculating the gene trees, statistical binning changes the input to the coalescent-based summary method. Hence, statistical binning is a technique to re-estimate gene trees used within the coalescent-based pipeline for species tree estimation, as shown in Fig. 8.1.

Fig. 8.1 describes the three possible pipelines (unbinned, unweighted binned, and weighted binned) for use with a summary method. In the unbinned analysis, each gene is analyzed independently, a gene tree is estimated for each gene, and then a summary method, such as MP-EST, uses the gene trees to estimate the species tree. In both the weighted and unweighted binned analyses, the gene trees are computed independently, and then the incompatibility graph is formed with one vertex for each gene. In the shown example, there are 12 genes, and so the graph has 12 vertices. The 12 vertices of the incompatibility graph are then assigned colors, with two vertices colored purple, three vertices colored green, three vertices colored red, and four vertices colored blue. Note that no two vertices of the same color have an edge between them. For each color class, the sequence alignments for the associated genes are concatenated into one long supergene alignment, and a supergene tree is computed on the supergene alignment using a fully partitioned maximum likelihood analysis. After this point, the weighted and unweighted binning methods have different strategies. In the unweighted binning method, exactly one copy of each supergene tree is given as input to the summary method, but in the weighted binning method multiple copies of the supergene trees are given as input. Hence, in this example, MP-EST analyzes only four supergene trees in the unweighted binning pipeline, but it analyzes 12 supergene trees in the weighted binning pipeline.

By design, if the bin sizes are exactly the same, then the statistical binning pipelines produced using weighted and unweighted statistical binning produce the same results; hence, these two approaches can only produce different results when the binning is unbalanced.

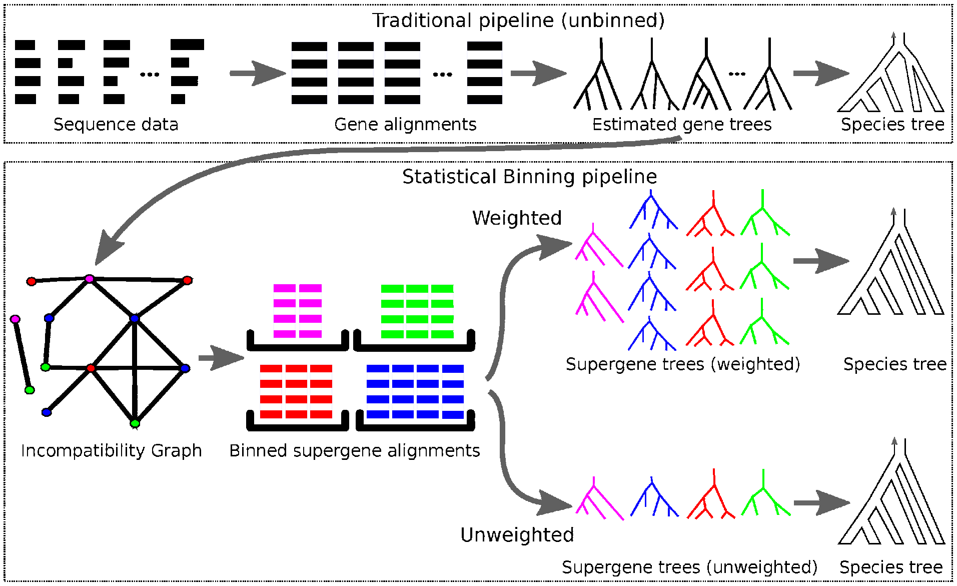

Traditional pipeline (unbinned)
Sequence data
Gene alignments
Estimated gene trees
Species tree
Statistical Binning pipeline
Weighted
Incompatibility Graph
Binned supergene alignments
Supergene trees (weighted)
Species tree
Unweighted
Supergene trees (unweighted)
Species tree

Figure 8.1: **Pipeline for unbinned analyses, unweighted statistical binning, and weighted statistical binning.** The input to the pipeline is a set of sequences for different loci across different species. In the traditional pipeline, a multiple sequence alignment and gene tree is computed for each locus, and then these are given to the preferred coalescent-based summary method, and a species tree is returned. In the statistical binning pipeline, the estimated gene trees are used to compute an incompatibility graph, where each vertex represents a gene, and an edge between two genes indicates that the differences between the trees for these genes is considered significant (based on the bootstrap support of the conflicting edges between the trees). The vertices of the graph are then assigned colors, based on a heuristic for balanced minimum vertex coloring, so that no edge connects two vertices of the same color. The vertices with a given color are put into a bin, and the sequence alignments for the genes in a bin are combined into a supergene alignment. A (supergene) tree is then computed for each supergene alignment using a fully partitioned analysis. In the unweighted binning approach (presented in [90]), these supergene trees are then given to the preferred summary method, and a species tree is returned. In the weighted binning approach presented here, each supergene tree is repeated as many times as the number of genes in its bin, and this larger set is then given to the preferred summary method.

8.3 Experimental study

8.3.1 Datasets

We use the avian and mammalian simulated datasets studied in [90] (each based on MP-EST analyses of biological datasets, and having at least 37 taxa) and two other collections of simulated datasets with 10 and 15 taxa. The simulated datasets range from moderately low ILS (the lowest ILS mammalian condition) to extremely high ILS conditions (the higher ILS 10-taxon and 15-taxon model conditions), and range in terms of average gene tree bootstrap support (from very low to moderately high). Thus, the simulated datasets provide a range of conditions in which we explore the impact of statistical binning. We also analyzed two biological datasets (a 48-species avian dataset and a 37-species mammalian dataset) studied in [90].

We used biologically-based simulated datasets that were studied in [90], and are based on species trees estimated using MP-EST on the avian dataset of [62] and the mammalian dataset of [129]. In the avian simulation, the markers vary in sequence length (250bp, 500bp, 1000bp, and 1500bp) in order to produce bootstrap support values similar to those we observed in the biological dataset. In the mammalian simulation, we again explored the impact of phylogenetic signal by varying the sequence length (250bp, 500bp, and 1000bp) for the markers. In both cases, three levels of ILS are simulated by multiplying or dividing all internal branch lengths in the model species tree by two, and we also explore various numbers of genes. The mammalian datasets range in ILS level from relatively low (18% average distance between true gene trees and the species tree) for the 2X branch length level to relatively high (54% average distance between true gene trees and the species tree) for the 0.5X branch length level, and the average bootstrap support on the estimated gene trees ranges from low (43%) for the shorter (250bp) sequences to moderately high (79%) for the longest (1000bp) sequences. The avian datasets have higher ILS levels than the mammalian datasets, and range from moderate (35% average distance between true gene trees and the species tree) for the 2X branch length condition to high (59% average distance between true gene trees and the species tree) for the 0.5X branch length condition. The estimated gene trees range in average bootstrap support from very low (27%) for the shortest (250bp) sequences to moderate (60%) for the longest (1500bp) sequences.

We also used a 15-taxon model species tree with a caterpillar-like (also known as a pectinate, or ladder-like) topology, which has 12 short internal branches (0.1 in coalescence units) in succession, a condition that gives rise to high levels of ILS [69, 122]. Ultrametric gene trees were simulated down this tree using the multi-species coalescent process. Unlike the biologically-based model conditions, no transformations of branch lengths were used, and therefore, gene trees follow a strict molecular clock. Sequence data were simulated down each gene tree, and we built four model conditions by trimming gene data to 100 or 1000 sites, and by using 100 or 1000 genes. This dataset is very homogeneous since all 10 replicates we simulated are based on the same species tree, and gene trees differ in topology and branch length only due to the coalescence process. The 15-taxon datasets have very high ILS levels (82% average topological distance between true gene trees and the species tree), and so represent a rather extreme condition. The gene trees estimated on the shorter sequences (100bp) had only 35% average bootstrap support, and the combination of very high ILS and very low average bootstrap support represents a very challenging condition. Gene trees estimated on the longer sequences have better average bootstrap support (70%), and so represent a somewhat easier condition.

We also generated 10-taxon simulated datasets using simPhy [86]. In this simulation protocol, we simulated a different species tree for each replicate, and simulated 200 gene trees for each species tree using the multi-species coalescent process. We simulated two model conditions, one with very high ILS and another with somewhat lower (but still high) ILS. The simPhy procedure uses a host of various distributions to make the gene trees heterogeneous in various aspects, such as sequence lengths, deviation of branch lengths from the strict molecular clock, and rate variation across different genes. We used these gene trees to simulate sequence data with 100 sites using Indelible [38]. Therefore, our 10-taxon datasets are very heterogeneous: different replicates have different species trees, and within each replicate, various genes have different rates of evolution. The ILS levels of the 10-taxon datasets range from moderately high (40% average distance from true gene trees to the species tree) for the "lower ILS" condition to extremely high (84% average distance) for the "higher ILS" condition. The average bootstrap support on the estimated gene trees ranged from 37% for the higher ILS condition to 45% for the lower ILS condition, and so both have very poor average bootstrap support. Thus, the 10-taxon and the 15-taxon datasets with short sequences represent the hardest model conditions in that they have very high ILS and very low average bootstrap support.

The simulated datasets we studied varied in many respects (sequence length per locus, whether the sequence evolution is ultrametric or not, and the ILS level). Table 8.1 presents data about the ILS level, as reflected in the average topological distance between the true gene trees and the true species tree. Note that two of the model conditions (the 10-taxon higher ILS and 15-taxon datasets) have extremely high ILS, reflected in average topological distances between the true gene trees and the species tree. In fact, most of the model conditions have high ILS levels (with 30% or more average topological distance between the true gene trees and the species tree), and only one model condition has low levels of ILS (the mammalian datasets with 2X branch lengths, which have 18% average topological distance between the true gene trees and true species tree). It is likely that the "1X" ILS levels for the mammalian and avian simulated datasets are larger than the ILS levels for the respective biological datasets, since the model trees that were used to generate these data are based on MP-EST analyses of the datasets, and results in [90] suggest that MP-EST estimations tend to under-estimate branch lengths, and hence inflate estimated ILS levels.

Dataset	ILS level	Discordance (%)
Avian	2X	35
Avian	1X	47
Avian	0.5X	59
Mammalian	2X	18
Mammalian	1X	32
Mammalian	0.5X	54
10-taxon	Lower ILS	40
10-taxon	Higher ILS	84
15-taxon	High ILS	82

Table 8.1: **Topological discordance between true gene trees and true species tree**. For each collection of simulated datasets (defined by the type of simulation and the ILS level), we show the average topological distance between true gene trees and the species tree.

8.3.2 Methods

We computed coalescent-based species trees using summary methods with MLBS gene trees in three ways: without binning, with weighted statistical binning and with unweighted statistical binning. Our main focus is on MP-EST, but we explore results with ASTRAL on a subset of the data. ASTRAL estimates species trees given unrooted gene trees, and can analyze very large datasets (such as the plant transcriptome dataset with approximately 100 species and 600 loci [151]); hence, ASTRAL can analyze larger datasets than MP-EST, and so understanding the impact of binning on ASTRAL's accuracy is of practical importance.

We perform statistical binning using both weighted and unweighted pipelines and using two support thresholds (B): 50% and 75%. Due to the extremely large computational effort involved, on our two large biologically-based simulated datasets, we explore one threshold for most of our results; we follow the protocol used in [90] and set $B = 50\%$ for the avian datasets, and $B = 75\%$ for the mammalian datasets. However, we also study the impact of B on one model condition for avian and mammalian datasets.

We compute gene trees and concatenation species trees using RAxML [130] maximum likelihood. For estimating supergene trees, we use fully partitioned RAxML analyses (using the $-M$ option to vary branch lengths across genes) for smaller (10- and 15-taxon) simulated datasets and for all biological analyses. However, since partitioned analyses are expensive, we use unpartitioned analyses to compute supergene trees for our studies on the avian and mammalian simulated datasets (because these studies are very extensive). We compare results using coalescent-based summary methods to concatenation, also using unpartitioned maximum likelihood. Note that the binned methods and the concatenation analysis would potentially become more accurate if fully partitioned analyses were employed.

8.3.3 Measurements

For the simulated datasets, we explore species tree accuracy with respect to the true (model) species tree topology (the missing branch rate, or false negative rate (FN)) and branch lengths, and also examine the branch support of both true positive and false positive branches. We also explore the error in the estimated gene trees and gene tree distribution estimated using binning (weighted and unweighted), compared to unbinned analyses. We analyze these simulated datasets using weighted statistical binning with MP-EST and ASTRAL, to determine if there are differences between weighted and unweighted statistical binning. Since ASTRAL does not produce branch lengths, we only use MP-EST to evaluate branch length estimation. In addition, we examine the bootstrap support on the branches of estimated species trees produced using MP-EST, as false positive edges that have low support are not as deleterious as false positive edges with high support. The bootstrap support of estimated species trees was not studied in [90], and so this study provides the first analysis of bootstrap support for MP-EST on these datasets, as well as of the impact of binning on bootstrap support values.

These aspects of phylogenomic estimation are important for different reasons. Species tree topologies indicate which species are more closely related to each other than to others, and so estimating accurate species tree topologies is the most important aspect of phylogenomic estimation. However, the improvement in species tree (coalescent-unit) branch length estimation is also biologically relevant, since these lengths are related to effective population sizes and generation times of ancestral species, and are also used to estimate the amount of ILS in the data. Bootstrap support is important, since low support branches are often ignored, but high support branches are generally assumed to be correct; hence, understanding whether a method returns high support for false positive branches (indicating incorrect relations within a tree) is particularly important. Improvements in estimating the gene tree distribution matter because the accuracy of summary methods depends on an input that captures the correct gene tree distribution.

For the biological datasets, we compare estimated species trees to the literature for each dataset, focusing on whether the estimated species tree violates known subgroups for the phylogeny.

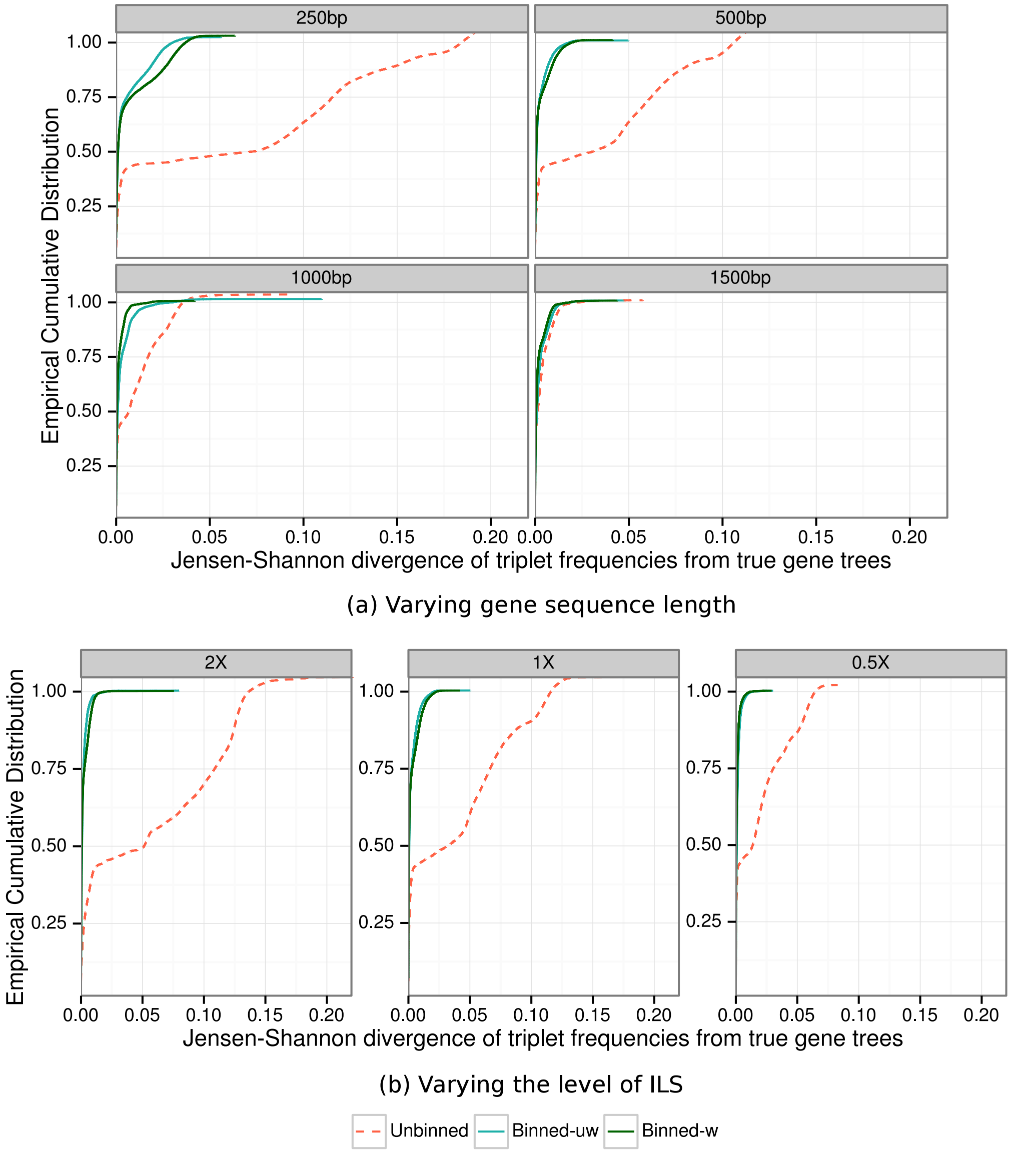
8.4 Results and discussion

8.4.1 Biologically-based simulated datasets

Gene tree error and gene tree distribution error on avian simulated datasets

We evaluated the impact of statistical binning on gene tree estimation error for the 1X (default ILS) model condition, with sequence lengths varying from 250bp to 1500bp. At the shorter sequence lengths, gene tree estimation error was reduced substantially (from 79% to 57% for 250bp, and from 69% to 57% for 500bp) (Table B.1 in Appendix B). Gene tree estimation error was reduced slightly at 1000bp (from 55% to 51%) and even less at 1500bp (from 46% to 45%). Hence, when gene tree estimation error is high due to insufficient sequence length, then binning reduces gene tree estimation error, but binning has little impact on gene tree estimation error when the sequences are long enough.

We measure the error in estimated gene tree distributions using the deviation of triplet frequencies from the triplet frequency distribution computed using the true gene trees. We express these results using a cumulative distribution over all possible triplets and all replicates; hence, if a curve for one method lies above the curve for another method, then the first method strictly improves on the second method with respect to estimating the gene tree distribution. In Fig. 8.2(a) we show results for 1000 avian genes under default ILS levels, as we vary the sequence length. In Fig. 8.2(b) we show results with 1000 genes of length 500bp, varying the ILS level. Here, true triplet frequencies are estimated based on true gene trees for each of the $\binom{n}{3}$ possible triplets, where n is the number of species. Similarly, triplet frequencies are calculated from estimated gene/supergene trees. For each of these $\binom{n}{3}$ triplets, we calculate the Jensen-Shannon divergence of the estimated triplet distribution from the true gene tree triplet distribution. We show the empirical cumulative distribution of these divergence scores. The empirical cumulative distribution shows the percentage of the triplets that are diverged from the true triplet distribution at or below the specified divergence level. Results are shown for 10 replicates. We used 50% bootstrap support threshold for binning, and estimated the supergene trees using RAxML with unpartitioned analyses. In both cases (Fig. 8.2(a) and (b)), both weighted and unweighted binning are nearly identical. Weighted and unweighted binning also show nearly identical gene tree distribution errors under other conditions (see Appendix B, Fig. B.3). Binning improves the accuracy of estimated gene tree distributions in general, but not for the longest sequences (1500bp). Also, the improvement over unbinned analyses was highest for the lowest ILS level (2X species tree branch lengths), but was high even for the highest ILS level we explored.



(a) Varying gene sequence length

(b) Varying the level of ILS

Figure 8.2: **Divergence of estimated gene tree (triplet) distributions from true gene tree distributions for MP-EST analyses of simulated avian datasets.** In (a), we vary the gene sequence length (250bp genes have the highest error, and 1500bp has the lowest error) and explore 1000 genes under default ILS levels, and in (b) we vary the amount of ILS and fix the number of genes to 1000 and sequence length to 500bp.

Species tree estimation error on avian simulated datasets

Fig. 8.3 shows results for species tree topology estimation error for analyses of avian genes of different length under the default ILS level using MP-EST and ASTRAL, for varying number of 500bp genes with default ILS using MP-EST, and for 1000 genes of 500bp with varying ILS using MP-EST. Weighted and unweighted statistical binning are essentially identical for both MP-EST and ASTRAL (no statistically significant differences were observed according to a two-way ANOVA test; see Tables 8.2 and 8.3), and both reduce species tree estimation error compared to unbinned analyses (differences were always statistically significant with p < 0.001; see Tables 8.2 and 8.3).

Dataset	Varying parameter	Weighted vs. Unweighted	WSB-50 vs. Unbinned	WSB-75 vs. Unbinned
10-taxon	ILS level	0.96	0.96	0.96
15-taxon	# of genes, seq length	0.96	0.96	**0.04**
Avian	sequence length	0.96	**<0.0001**	n.a.
Avian	ILS level	0.96	**<0.0001**	n.a.
Avian	# of genes	0.91	**<0.0001**	n.a.
Mammalian	# of genes, seq length	0.96	n.a.	**<0.0001**
Mammalian	ILS level	0.96	n.a.	**0.0003**

Table 8.2: **Statistical significance test results for choice of binning method on MP-EST.** We performed ANOVA to test the significance of the choice of methods (unbinned, weighted binned, unweighted binned, WSB-50: weighted statistical binning using 50% bootstrap support threshold and WSB-75: weighted binning using 75% bootstrap support threshold). For weighted vs. unweighted, we compared 50% bootstrap support threshold for avian, 75% for mammalian, and both 50% and 75% for 15- and 10-taxon datasets. All *p*-values are corrected for multiple hypothesis testing using the FDR correction ($n = 16$). “n.a.” stands for “not available”.

The largest improvements are for the shortest gene sequences, where

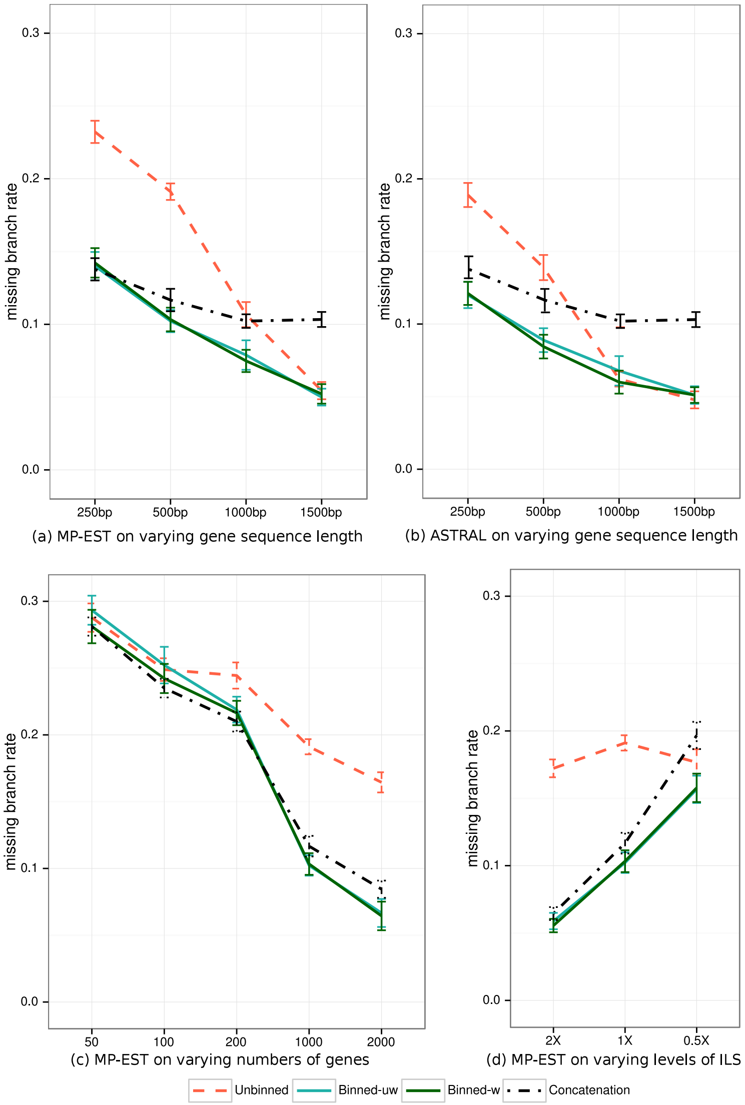


Figure 8.3: **Species tree estimation error (FN) for MP-EST and ASTRAL with MLBS on avian simulated datasets**. (a) MP-EST on 1000 genes with varying gene sequence length (controlling gene tree error) and with 1X ILS. (b) ASTRAL on the exact same conditions, (c) MP-EST on varying numbers of genes with fixed default level of ILS (1X level) and 500bp sequence length, and (d) MP-EST on varying levels of ILS and 1000 genes of length 500bp. We show results for 20 replicates everywhere, except for 2000 genes that are based on 10 replicates. Binning was performed using 50% bootstrap support threshold.

Dataset	Varying parameter	Weighted vs. Unweighted	WSB-50 vs. Unbinned	WSB-75 vs. Unbinned
10-taxon	ILS level	1	1	0.91
15-taxon	# of genes, seq length	0.91	0.57	**0.008**
Avian	sequence length	0.91	**<0.0001**	n.a.
Avian	sequence length	1	n.a.	0.57
Mammalian	ILS level	0.57	n.a.	**0.0009**
Mammalian	# of genes	0.91	n.a.	**<0.0001**

Table 8.3: **Statistical significance test results for choice of binning method on ASTRAL.** We performed ANOVA to test the significance of the choice of methods (unbinned, weighted binned, unweighted binned, WSB-50: weighted statistical binning using 50% bootstrap support threshold and WSB-75: weighted binning using 75% bootstrap support threshold). For weighted vs. unweighted, we compared 50% bootstrap support threshold for avian, 75% for mammalian, and both 50% and 75% for 15- and 10-taxon datasets. All p-values are corrected for multiple hypothesis testing using the FDR correction ($n = 14$). "n.a." stands for "not available".

error is reduced from 23% to 14% using MP-EST and from 19% to 13% using ASTRAL. The difference between binned and unbinned analyses is lower for 1000bp sequences, and there are no noteworthy differences for 1500bp sequences (sequence length has a statistically significant impact; see Tables 8.4 and 8.5). When the number of genes is changed (see Fig. 8.3(c)), the impact of binning on MP-EST ranges from neutral to highly positive, and the largest improvements are for datasets with large numbers of genes (impact of the number of genes is statistically significant; see Table 8.4). The impact of binning is also significantly impacted by ILS levels (see Table 8.4), with the largest improvements obtained for lower levels of ILS. In general, binning helps both ASTRAL and MP-EST, but MP-EST tends to be helped more than ASTRAL. For example, with 500bp genes, the error for MP-EST is reduced from 19% to

Dataset	Interaction variable	Weighted vs Unweighted	WSB-50 vs Unbinned	WSB-75 vs Unbinned
10-taxon	ILS level	0.99	0.99	0.49
15-taxon	# of genes, seq length	0.99 & 0.99	0.59 & 0.99	0.24 & 0.17
Avian	sequence length	0.99	**<0.0001**	n.a.
Avian	ILS level	0.99	**<0.0001**	n.a.
Avian	# of genes	0.99	**<0.0001**	n.a.
Mammalian	# of genes, seq length	0.99 & 0.99	n.a.	0.99 & 0.38
Mammalian	ILS level	0.15	n.a.	0.15

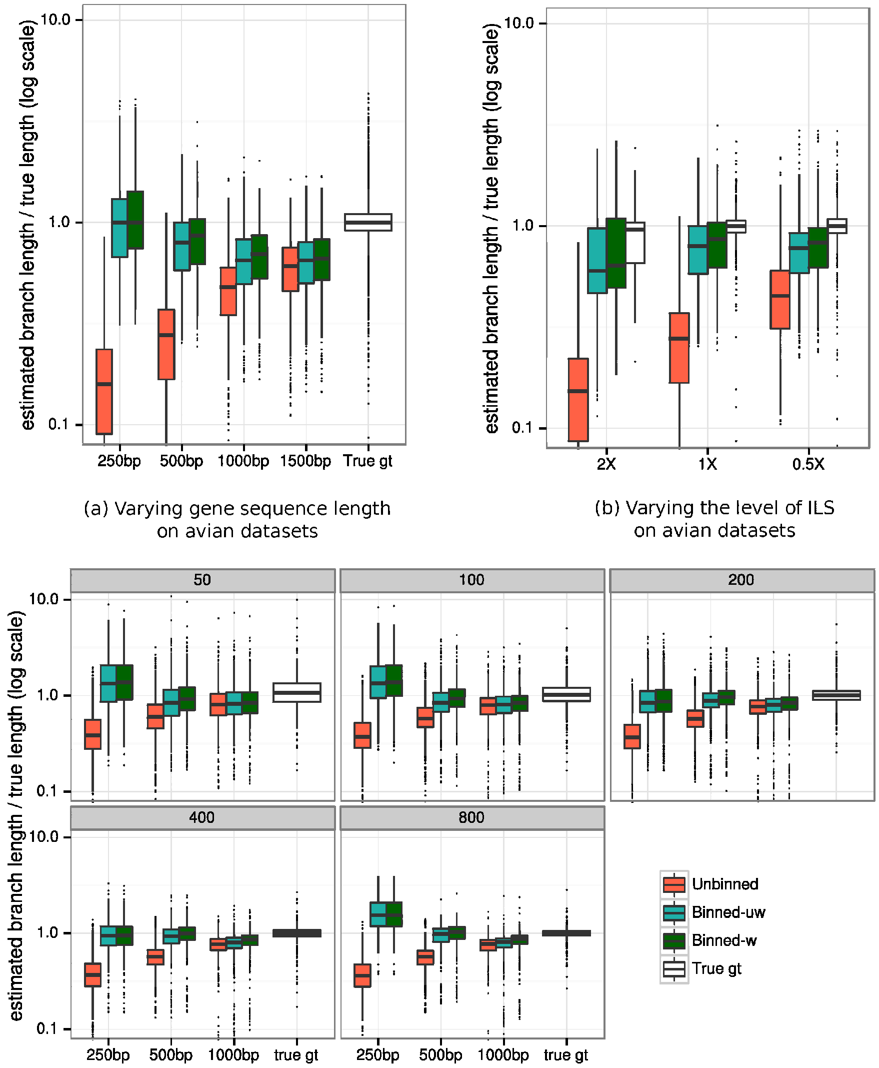
Table 8.4: **Statistical significance test results for interaction effects (binning and simulation parameter) on MP-EST.** We performed ANOVA to test the significance of whether there is an interaction between the choice of the method (unbinned, weighted binned, unweighted binned, WSB-50: weighted statistical binning using 50% bootstrap support threshold and WSB-75: weighted statistical binning using 75% bootstrap support threshold) and the variable changed in each dataset. For weighted vs. unweighted, we compared 50% bootstrap support threshold for avian, 75% for mammalian, and both 50% and 75% for 15- and 10-taxon datasets. All p-values are corrected for multiple hypothesis testing using the FDR correction ($n = 21$). "n.a." stands for "not available".

10% using binning, but the error of ASTRAL is reduced from 15% to 9%.

Fig. 8.4 shows the impact of binning on species tree branch length (in coalescent units) estimation error on the biologically-based simulations using MP-EST. We show the species tree branch length error (the ratio of estimated branch length to true branch length for branches of the true tree that appear in the estimated tree; 1 indicates correct estimation). Fig. 8.4(a) shows results on 1000 genes under default (1X) ILS levels and varying gene sequence length, and Fig. 8.4(b) shows results on 1000 genes of 500bp with varying ILS levels. Branch length estimation accuracy is reported using the ratio of the estimated branch length to the true branch length, for those true branches recovered by

Dataset	Interaction variable	Weighted vs Unweighted	WSB-50 vs Unbinned	WSB-75 vs Unbinned
10-taxon	ILS level	0.99	1	0.99
15-taxon	# of genes, seq length	0.99 & 0.99	0.99 & 0.99	0.29 & **0.02**
Avian	sequence length	0.99	**<0.0001**	n.a.
Mammalian	sequence length	0.99	n.a.	0.29
Mammalian	ILS level	0.99	n.a.	0.29
Mammalian	# of genes	0.99	n.a.	0.99

Table 8.5: **Statistical significance test results for interaction effects (binning and simulation parameter) on ASTRAL.** We performed ANOVA to test the significance of whether there is an interaction between the choice of the method (unbinned, weighted binned, unweighted binned, WSB-50: weighted statistical binning using 50% bootstrap support threshold and WSB-75: weighted statistical binning using 75% bootstrap support threshold) and the variable changed in each dataset. For weighted vs. unweighted, we compared 50% bootstrap support threshold for avian, 75% for mammalian, and both 50% and 75% for 15- and 10-taxon datasets. All p-values are corrected for multiple hypothesis testing using the FDR correction ($n = 17$). "n.a." stands for "not available".



(a) Varying gene sequence length on avian datasets

(b) Varying the level of ILS on avian datasets

(c) Varying gene tree estimation error, and number of genes on mammalian datasets

Figure 8.4: **Effect of binning on the branch lengths (in coalescent units) estimated by MP-EST using MLBS on the avian and mammalian simulated datasets**. Results are shown for (a) 1000 avian genes of 1X ILS level with varying gene sequence length, (b) 1000 avian genes of 500bp and with varying levels of ILS, and (c) varying number of mammalian genes and varying sequence length (250bp, 500bp, and 1000bp) with 1X ILS level. Results are shown for 20 replicates. We used 50% and 75% bootstrap support threshold for binning on avian and mammalian datasets, respectively.

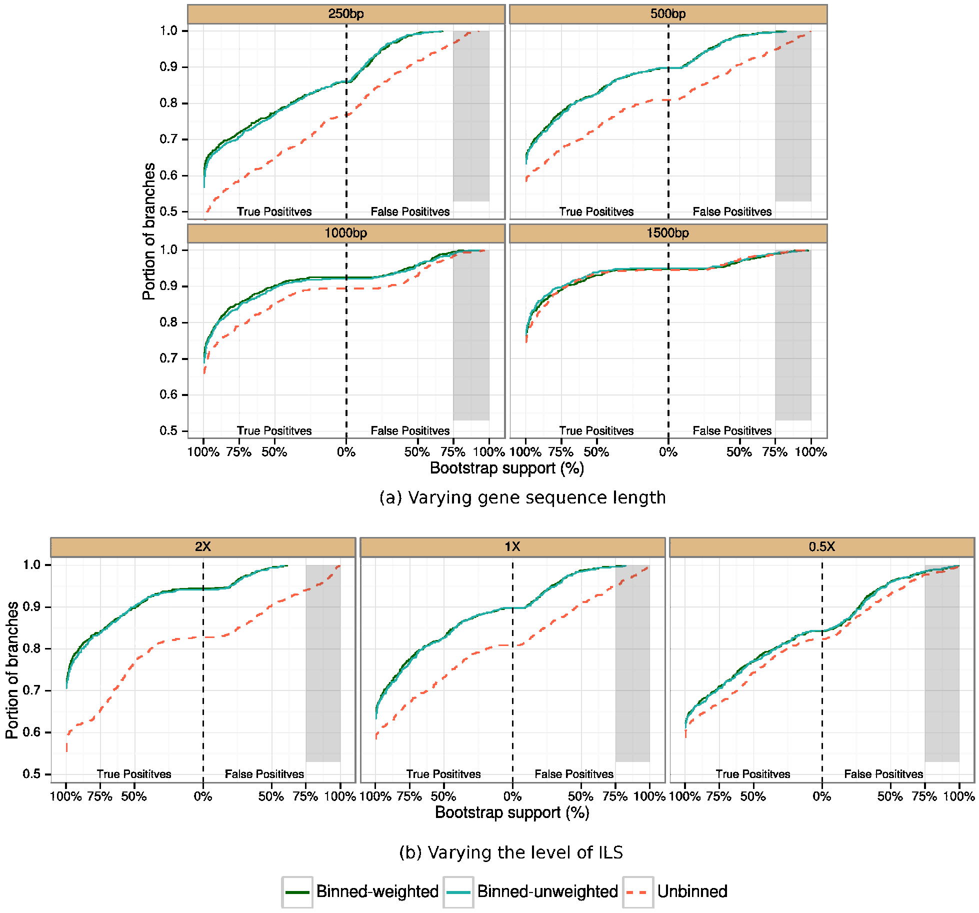
the method. Thus, values equal to 1 indicate perfect accuracy, values below 1 indicate under-estimation of branch lengths (and hence over-estimation of ILS), and values above 1 indicate over-estimation of branch lengths (and hence under-estimation of ILS).

Both types of binning (weighted and unweighted) produce nearly identical results with respect to species tree branch length estimation (with a slight advantage for weighted analyses). Unbinned analyses substantially under-estimate branch lengths, but as the sequence length increases, the branch length estimations produced by unbinned analyses improve, so that they are more accurate with 1500bp markers. The most accurate species tree branch length estimation is obtained using true gene trees. Using binning (either type) improves branch length estimation from estimated gene trees, and the improvement is very large for the shorter sequences (Fig. 8.4(a)). When levels of ILS are changed, weighted and unweighted binning are again close (with a slight advantage for weighted), and show little change in branch length estimation with changes in ILS levels; however, unbinned analyses substantially under-estimate branch lengths for the lowest ILS model condition, and then become more accurate (although still under-estimate) with increases in the ILS level. Hence, the biggest improvement obtained by binning is for the lowest ILS (2X branch lengths), and there is less improvement for the highest ILS level (0.5X). The likely explanation for this trend is that MP-EST interprets all discord as due to ILS, and produces a model tree (with branch lengths) that it considers most likely to generate the observed discordance. Hence, MP-EST tree branch lengths will be closer to the correct lengths when the ILS level is very high.

Bootstrap support on avian simulated datasets

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

Fig. 8.5 shows results on 1000 avian genes under default ILS and with varying sequence length, and also with 1000 genes of 500bp with varying ILS levels. Both types of binning are nearly identical in terms of their impact on bootstrap support, and both improve bootstrap support; in particular, using binning increases the number of highly supported true positive branches and decreases the number of highly supported false positive branches. However,



(a) Varying gene sequence length

(b) Varying the level of ILS

Figure 8.5: **Cumulative distribution of the bootstrap support values (obtained using MLBS) of true positive (TP) and false positive (FP) edges estimated by binned and unbinned MP-EST on avian datasets**. In (a) we fix the number of genes to 1000, use default ILS levels, and vary sequence length to control gene tree estimation error, and in (b) we study 1000 genes with 500bp sequence length, and vary ILS levels. We order the branches in the estimated species tree by their quality, so that the true positives with high support come first, followed by lower support true positives, then by false positives with low support, and finally by false positives with high support. The false positive branches with support above 75% are the most troublesome, and the highly supported false positives are indicated by the grey area. When the curve for a method lies above the curve for another method, then the first method has better bootstrap support. We used 50% bootstrap support threshold for binning.

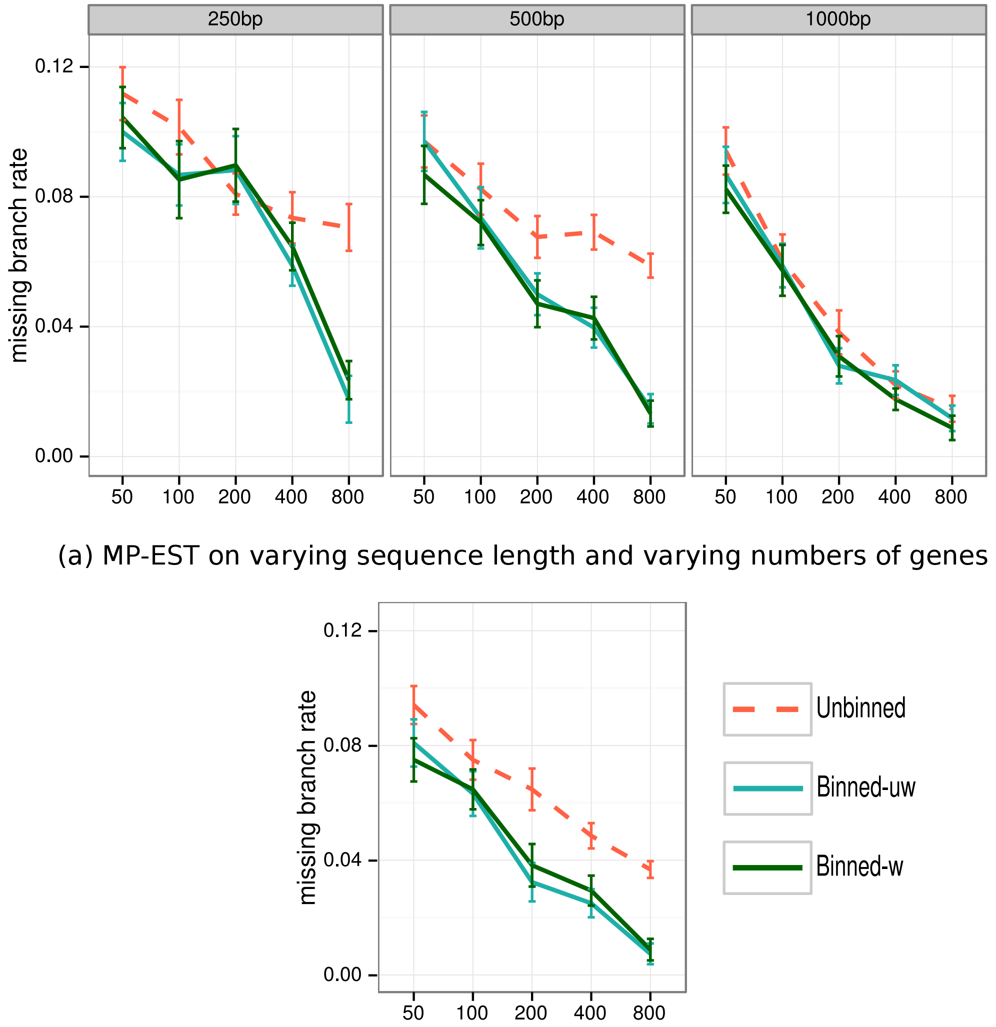
the sequence length modulates the impact of binning on bootstrap support, so that the largest impact is for the shortest sequences (250bp) and there is no discernible impact for the longest sequences (1500bp). ILS levels also impact how binning affects the bootstrap support, so that the biggest improvement in bootstrap support is obtained for the lowest ILS level (2X branch lengths). The number of genes also impacts the bootstrap support (Appendix B, Fig. B.4) so that the biggest improvement in bootstrap support is obtained for the largest number of genes (2000) (and there is little to no difference between binned and unbinned analyses on 50 or 100 genes); furthermore, weighted and unweighted binning produce very similar bootstrap support values.

Comparisons to concatenation on avian simulated datasets

On the shortest 250bp sequences, concatenation matches the accuracy of weighted and unweighted binned MP-EST methods (Fig. 8.3(a)) and is slightly less accurate than both binned ASTRAL trees (Fig. 8.3(b)). As sequence length increases, both types of binning using either ASTRAL or MP-EST become more accurate than concatenation. Unbinned analyses are less accurate than concatenation for shorter sequences and more accurate for longer sequences (the transition point depends on whether ASTRAL or MP-EST is used). Both binned analyses are more accurate than concatenation and unbinned analyses at all ILS levels (Fig. 8.3(d)). Thus, compared to concatenation, binned analyses have their largest advantage on longer gene sequences, higher ILS levels, and higher number of genes.

Results on mammalian datasets

Results on simulated mammalian datasets are similar to analyses of avian datasets. In nearly every condition, both weighted and unweighted binning show very similar results (see Fig. 8.6) and have no statistically significant differences using either ASTRAL or MP-EST (see Tables 8.2 and 8.3). As before, we evaluated the impact of statistical binning on gene tree estimation error under the 1X (default ILS) model condition with varying sequence lengths, and observed that binning substantially reduces gene tree estimation error for short sequences (250bp and 500bp) but had little impact on longer sequences (1000bp) (Table B.1 in Appendix B). Binning improves gene tree distributions, generally with very large improvements, and the improvements decrease with the sequence length and ILS level (Appendix B, Fig. B.5). Binning also improves species tree topology estimation (Fig. 8.6 and Tables 8.2 and 8.3). The impact appears to depend on the sequence length (binning seems more beneficial for shorter sequences and neutral for longer sequences) and number of genes (binning can dramatically improve species tree topologies given a large number of genes, but can be neutral or even detrimental for a small number of genes), and the choice of summary method (binning helps both ASTRAL and MP-EST, but helps MP-EST more). ILS level also seems to impact relative accuracy (Tables 8.4 and 8.5), so that binning seems most helpful for low ILS levels, and less helpful for high ILS levels (Appendix B, Fig. B.6). However, the effects of number of genes, sequence length, and the ILS level were not statistically significant for this dataset (Tables 8.4 and 8.5).



(a) MP-EST on varying sequence length and varying numbers of genes

(b) ASTRAL on varying numbers of genes

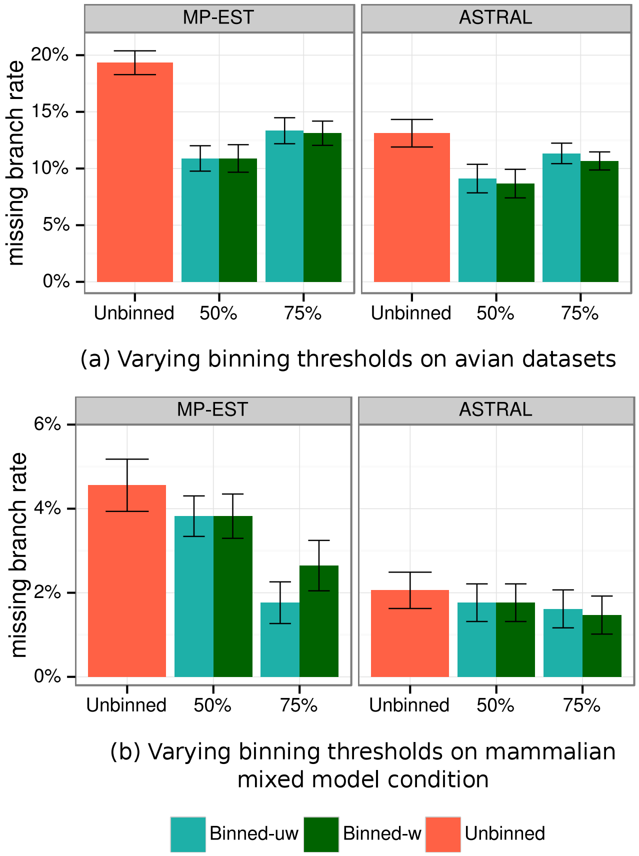
Figure 8.6: **Species tree estimation error for MP-EST and ASTRAL using MLBS on mammalian simulated datasets**. We show average FN rate over 20 replicates. (a) Results for MP-EST. We varied the number of genes (50, 100, 200, 400 and 800) and sequence length (250bp (43% BS), 500bp (63% BS) and 1000bp (79% BS)) with default amount of ILS (1X level). (b) ASTRAL on varying numbers of genes with fixed 1X ILS level and 500bp sequence length. We used 50% and 75% bootstrap support threshold for binning on avian and mammalian datasets, respectively, and estimated the supergene trees using RAxML with unpartitioned analyses.

As observed in the avian simulations, unbinned analyses substantially under-estimate species tree branch lengths (Fig. 8.4(c) and Fig. B.7 in Appendix B). Both weighted and unweighted binning produce nearly identical branch lengths for all sequence lengths, number of genes, and ILS levels, and both types of binning come closer to the true branch lengths than unbinned analyses. Finally, both weighted and unweighted binning produce nearly identical species tree branch support values, where both match or improve unbinned analyses for all tested numbers of genes, sequence lengths, and ILS levels (Figures B.8 and B.9 in Appendix B). However, improvements increase with the number of genes and decrease with the sequence length and ILS level.

Impact of support threshold B on avian and mammalian simulated datasets

In addition to varying model conditions, we use a single avian and a single mammalian model condition to study the impact of the support threshold B on binning (Fig. 8.7). We use a mixed model condition with 200 genes of 500bp and 200 genes of 1000bp for the mammalian dataset, and a model condition with 1000 genes of 500bp for the avian dataset (both with default 1X ILS level).

On the avian dataset, binning is always beneficial, but the impact is larger with $B = 50\%$ compared to $B = 75\%$ (Fig. 8.7(a)). For example, unbinned MP-EST has 19% error, and using $B = 50\%$ reduces the error to 11%, and using $B = 75\%$ reduces the error to 13%.



(a) Varying binning thresholds on avian datasets

(b) Varying binning thresholds on mammalian mixed model condition

Figure 8.7: **Species tree estimation error for MP-EST and ASTRAL using MLBS on avian and mammalian simulated datasets with two support thresholds (B).** We show average FN rate for unbinned, and weighted and unweighted binned analyses with both $B = 50\%$ and $B = 75\%$. Results are shown for (a) the avian dataset with 10 replicates of 1000 genes of length 500bp and 1X ILS level, and (b) the mammalian dataset with 20 replicates of 400 mixed genes (200 genes with 500bp and 200 genes with 1000bp) with 1X ILS level.

On the mammalian mixed data, binning is beneficial in all cases (see Fig. 8.7(b)); however, the extent of the impact depends substantially on both the threshold and the summary method. ASTRAL has high accuracy even without binning, and binning with either threshold has only a small impact on its accuracy. When MP-EST is used, binning with $B = 50\%$ leads to relatively small improvements in accuracy, whereas $B = 75\%$ results in much larger improvements. Thus, the choice of the threshold can have an impact, but for the two model conditions we studied here both choices of the threshold are beneficial.

Effects of binning on gene tree and species tree error for 15-taxon datasets

We explored the impact of statistical binning on gene tree estimation error using two sequence lengths and two values for B, the bootstrap support threshold parameter (Table B.1 in Appendix B). For the shorter sequence lengths (100bp), binning increases gene tree estimation error (from 77% to 80% when $B = 50\%$, and from 77% to 86% when $B = 75\%$). For the longer sequence lengths (1000bp), binning with $B = 50\%$ has no impact on gene tree estimation error, but using $B = 75\%$ increases error from 36% to 40%. Thus, statistical binning increases gene tree estimation error for these very high ILS 15-taxon datasets, but the amount of the increase depended on the parameter B (with larger increases for $B = 75\%$ and small increases for $B = 50\%$) and sequence length (where the impact on gene tree estimation error is much reduced for the 1000bp alignments).

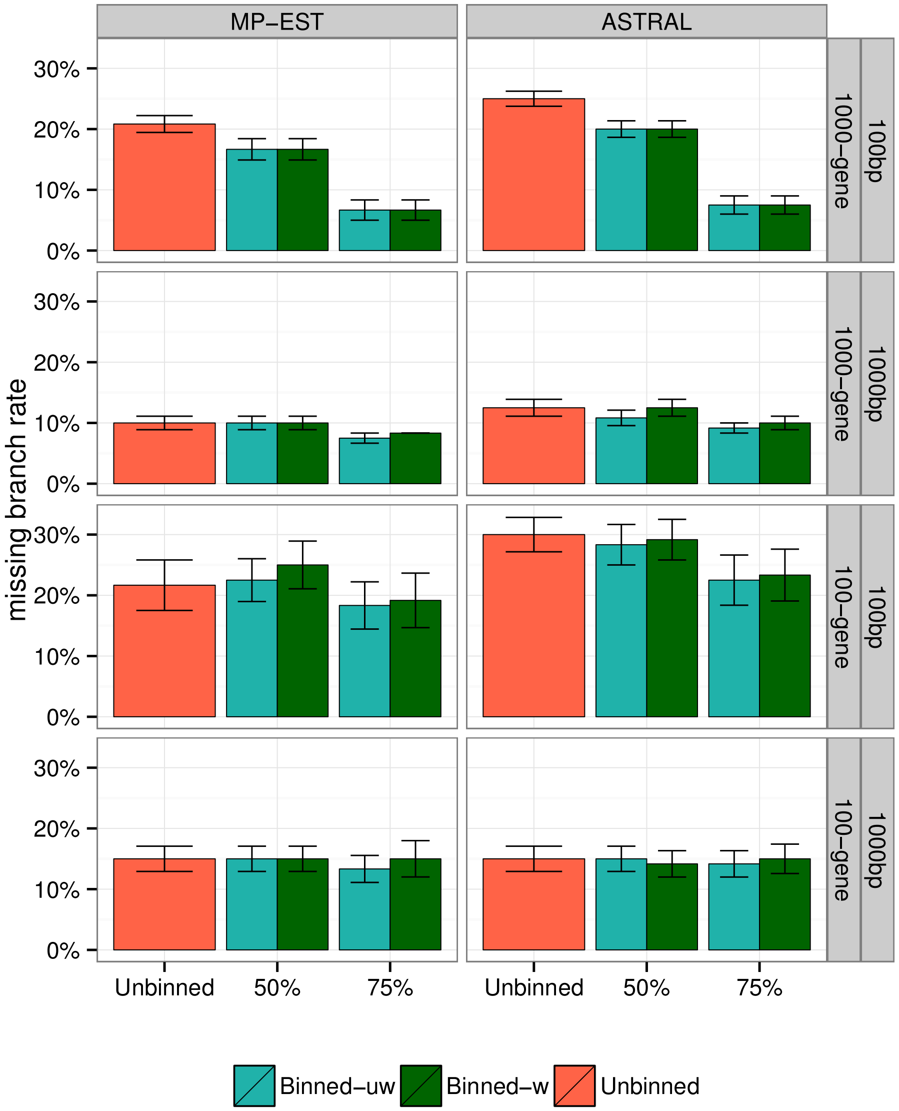


Figure 8.8: **Species tree estimation error for MP-EST and ASTRAL with MLBS on 15-taxon simulated datasets**. We show average FN rate over 10 replicates. We varied the number of genes (100 and 1000) and sequence length (100bp and 1000bp). We used 50% and 75% bootstrap support thresholds for binning, and estimated the supergene trees using RAxML with fully partitioned analyses.

Fig. 8.8 shows the impact of weighted and unweighted statistical binning on species tree accuracy for the 15-taxon dataset. We apply statistical binning with two support thresholds (50% and 75%), and we use both MP-EST and ASTRAL as the summary method. In all cases, weighted and unweighted binning have similar accuracy, with no statistically significant differences (Tables 8.2 and 8.3). The relative accuracy of unbinned and binned analyses depends on the support threshold, so that with $B = 50\%$, there are no statistically significant differences, but with $B = 75\%$, binning significantly improves accuracy ($p = 0.04$ for MP-EST and $p = 0.008$ for ASTRAL; Tables 8.2 and 8.3). The extent of the improvements seems larger for more genes and smaller alignments, but the impact of these factors are not statistically significant for MP-EST ($p = 0.24$ and $p = 0.17$ respectively) and only impact of sequence length was significant for ASTRAL ($p = 0.02$; Tables 8.4 and 8.5). The biggest gains are obtained when the 75% threshold is used with 1000 genes of 100bp, where binning reduces the error of MP-EST from 21% to only 7%. Thus, the choice of the threshold can matter, and on this dataset, the effects of binning can range from neutral to highly beneficial, depending on the threshold used, number of genes, and gene sequence length.

Effects of binning on species tree error for 10-taxon datasets

Fig. 8.9 shows the impact of binning on species tree accuracy on the 10-taxon datasets with two choices of the threshold B for the statistical binning pipeline ($B = 50\%$ and 75%), two choices of the summary method (MP-EST and ASTRAL), and two levels of ILS (high and very high). No statistically significant differences are observed on these data between weighted and unweighted binning, or between weighted binning and unbinned analyses (see Tables 8.2 and 8.3); nevertheless, some patterns can be observed in terms of the average error (Fig. 8.9). Both weighted and unweighted statistical binning are close to neutral (regardless of the choice of method or level of ILS) when applied with a 50% threshold. When the 75% threshold is used, the impact of binning depends on the level of ILS: binning improves accuracy with low ILS levels and reduces accuracy with high ILS levels, especially when MP-EST is used, but these differences are not statistically significant (Tables 8.2 and 8.3).

Analysis of biological datasets

We compared weighted and unweighted binning of MP-EST and ASTRAL on MLBS gene trees on the avian and mammalian biological datasets studied in [90].

Results for MP-EST on these datasets showed the following trends. First, for the avian dataset, there are no topological differences between MP-EST trees estimated using weighted or unweighted statistical binning, and extremely small differences in branch support (less than 3%; see Fig. 8.10). Thus, although [62] only explored unweighted statistical binning with MP-EST, the main conclusions they drew about the evolutionary history of modern birds are also found in the weighted statistical binning analysis using MP-EST. The unbinned MP-EST analysis violates several subgroups established in the

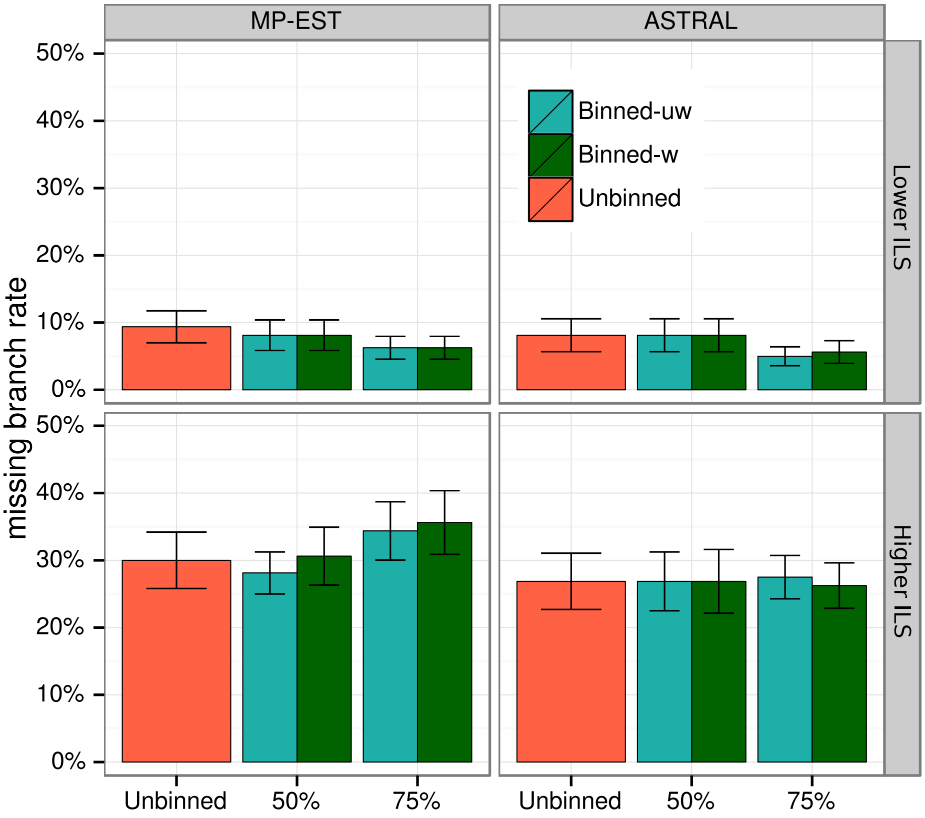


Figure 8.9: **Species tree estimation error for MP-EST and ASTRAL with MLBS on 10-taxon simulated datasets**. We show average FN rate over 20 replicates. We varied the amount of ILS and fixed the number of genes to 200 and gene sequence length to 100bp. We used 50% and 75% bootstrap support thresholds for binning, and estimated the supergene trees using RAxML with fully partitioned analyses.

avian phylogenomics project and other studies (indicated in red in Fig. 8.10), but the binned MP-EST analyses do not violate any of these subgroups. Of these violated subgroups, the failure of the unbinned MP-EST analysis to recover Australaves is the most significant, since it has been recovered in many prior analyses [65, 87, 133, 147]. On the mammalian dataset, weighted and unweighted MP-EST again produce the same exact tree, with small differences in support (less than 3%; see Appendix B, Fig. B.13). The unbinned MP-EST tree, however, has one topological difference (the position of treeshrews; compare Figs. B.12 and B.13 in Appendix B) with binned analyses, as discussed in [90].

Results for ASTRAL on the biological datasets show generally similar trends. Unbinned ASTRAL on the avian dataset (Appendix B, Fig. B.10) recovers Australaves and hence is more in line with the prior literature than unbinned MP-EST; however, just like unbinned MP-EST, the unbinned ASTRAL does not recover some key clades recovered by concatenation and other analyses reported in [62]. Using weighted and unweighted statistical binning with ASTRAL on the avian dataset produces almost identical results, and are also almost identical to the binned MP-EST tree (the only change is the position of hoatzin which has low support in all trees; see Appendix B, Fig. B.11). On the mammalian dataset, trees produced by binned ASTRAL analyses with weighted or unweighted binning pipelines are topologically identical to each other, and to the tree produced by the unbinned analysis, and have rather small differences in bootstrap support (see Appendix B, B.14). Binned and

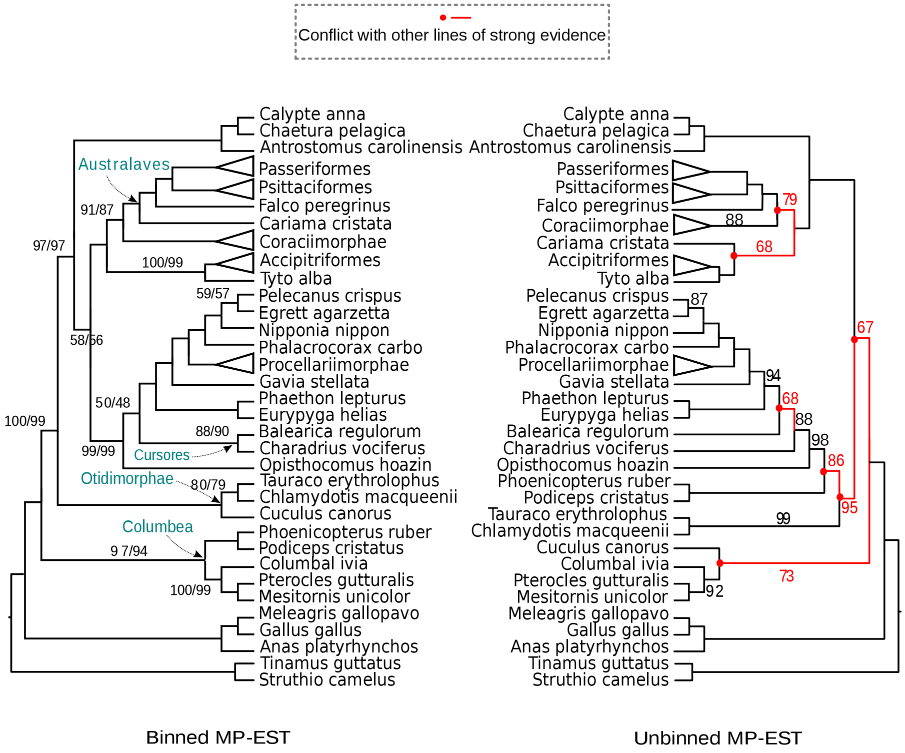


Figure 8.10: **Trees computed on the avian biological dataset using MP-EST on MLBS gene trees.** We show results with weighted and unweighted binning (left), and unbinned analyses (right). We used 50% bootstrap support threshold for binning. Supergene trees were estimated using fully partitioned analyses. MP-EST with weighted and unweighted binning returned the same tree. The branches on the binned MP-EST tree are labeled with two support values side by side: the first is for unweighted binning and the second is for weighted binning; branches without designation have 100% support. Branches in red indicate contradictions to known subgroups.

unbinned ASTRAL analyses and binned MP-EST analyses all put treeshrews as sister to Glires, while unbinned MP-EST puts them as sister to primates. The placement of treeshrews is of substantial debate, and so the differential placement is of considerable interest in mammalian systematics.

Overall, results on the two biological datasets show that weighted and unweighted statistical binning analyses produced identical species trees and nearly identical branch support values; furthermore, these binned analyses were more congruent with established subgroups than unbinned analyses.

8.4.2 Summary of observations

Across all our analyses, results for both ASTRAL and MP-EST are very similar with respect to how they responded to statistical binning. Weighted statistical binning produces nearly identical results to unweighted statistical binning on the biologically-based simulated datasets, and topologically identical results (with very similar bootstrap support values) on the biological datasets we explored in this study, and so this study generally supports the conclusions about statistical binning in [90]. In addition, because weighted and unweighted statistical binning produce topologically identical trees on the avian dataset, this study supports the findings about the avian phylogeny reported in [62]. The fact that weighted and unweighted binning typically produced similar results is not surprising, since the unweighted binning technique strives to create "balanced" bins as much as possible, and largely achieves this on the datasets we explored. Furthermore, if the bins produced by statistical binning have *exactly* the same size, then pipelines based on weighted and unweighted statistical binning will produce the same species tree, since the distributions of gene trees they produce will be identical. Since the bin sizes produced using our heuristic for balanced minimum vertex coloring are close to balanced, this explains why we observed very small differences between weighted and unweighted statistical binning in these analyses.

Under most of the model conditions we studied, both weighted and unweighted statistical binning improved the estimation of gene tree topologies, gene tree distributions, species tree topologies and branch lengths, and bootstrap support (so that statistical binning increases bootstrap support for true positive edges, and reduces the number of highly supported false positives), compared to unbinned analyses. These improvements are largest when gene sequence alignments have low phylogenetic signal, the gene trees exhibit at most moderately large ILS levels, or there are many genes.

The impact of statistical binning on the 15-taxon datasets is somewhat different than for the biologically-based simulations. Gene tree estimation accuracy is reduced for both sequence lengths (though the impact is small for the longer sequence lengths and only substantial for the short sequence lengths with $B = 75\%$). Nevertheless, the impact on species tree estimation on these data tends to be neutral, but there are also conditions where binning was beneficial.

On the lower ILS 10-taxon datasets, statistical binning reduces gene tree estimation error, and both weighted and unweighted binning reduce species tree estimation error for $B = 75\%$. However, species tree estimation error is unchanged when $B = 50\%$.

The results on the higher ILS 10-taxon datasets stand out from the other analyses: statistical binning slightly increases gene tree estimation error when B=50% but substantially increases gene tree estimation error when B=75%. Furthermore, while species tree estimation error is not increased for $B = 50\%$, when $B = 75\%$, the error increases.

The difference in impact for statistical binning in this case is interesting, and points out the significance of how B is set. To understand this, note that when B is very small, then bin sizes will tend to be very small, since any pair of incompatible branches with support above B will be considered to be evidence of statistically significant discord; thus, small settings for B produce results that are similar to unbinned analyses. Conversely, very large settings for B are more likely to bin genes together, since only the strongest supported conflicting branches will prevent genes from being binned. Therefore, if all the gene trees have low support then statistical binning could tend to produce results that are similar to concatenation. Thus, the choice of the threshold matters.

To better understand the difference in impact of statistical binning on these simulated datasets, it is helpful to consider the ILS levels and gene tree bootstrap support values for these data. As shown in Table 8.1, the average distance between the true gene trees and the species trees ranges for these datasets from as low as 18% (for the Mammalian 2X collection) to above 80%

(for the 10-taxon higher ILS collection and the 15-taxon collection). Fig. 8.3 shows how the effect of statistical binning used with MP-EST is impacted by ILS level on the avian datasets: statistical binning provided an improvement at all ILS levels, with the largest improvement for the lowest ILS level (2X branch lengths) and the smallest improvement on the highest ILS level (0.5X branch lengths). Figures B.5, B.6, B.7 in Appendix B evaluate this issue on the mammalian datasets, and shows large improvements provided by statistical binning under the lowest (2X branch lengths) ILS level, smaller improvements under the middle (1X branch length) ILS level, and then no improvement under the highest (0.5X branch lengths) ILS level. Thus, statistical binning provided an improvement except for a small number of model conditions: some of the 15-taxon conditions (which have discordance of 82%), the higher ILS 10-taxon conditions (which have discordance of 84%), and the highest ILS mammalian condition (which have discordance of 54%). Table B.2 in Appendix B shows that the average bootstrap support for the higher ILS 10-taxon datasets is quite low – only 37%. Thus, statistical binning seems to be beneficial when both ILS level and gene tree bootstrap support are not too high, will be neutral when bootstrap support values are high (so little or no binning occurs), but can be detrimental when ILS levels are extremely high but gene tree bootstrap support is low enough that binning occurs. Thus, one consequence of this study is the suggestion that when ILS levels are very high and the average gene tree bootstrap support is low, then either statistical binning should not be used, or it should be used in a very conservative fashion – with the parameter B set very low.

8.5 Conclusion

Because species trees and gene trees can differ, the estimation of species trees requires multiple loci. One approach to estimating species trees from multiple conflicting loci seeks to restrict the set of loci using principled arguments [124], but other approaches that explicitly model the discordance have also been developed. When gene tree discord is due to incomplete lineage sorting, then summary methods, such as MP-EST or ASTRAL, can be used to estimate the species tree by combining gene trees. However, this study, as well as others [8, 41, 74, 90, 93, 111], demonstrates that gene tree estimation error impacts species tree estimation, so that species trees estimated using summary methods on poorly estimated gene trees can have low accuracy. The (unweighted) statistical binning technique proposed in [90] improved the accuracy of estimated gene trees, and was shown to improve the accuracy of MP-EST when applied to MLBS gene trees. However, using unweighted statistical binning within a phylogenomic pipeline can be statistically inconsistent [5].

This study described a modification to statistical binning, obtained by replicating each supergene tree by the number of genes in its bin (equivalently, replacing each gene tree from the input set by its recalculated tree, which is the supergene tree for the bin).

On the biologically-based simulated datasets, weighted and unweighted statistical binning generally improved estimated gene tree distributions and led to improvements for MP-EST and ASTRAL estimations of species tree topologies. The use of statistical binning with MP-EST also improved estimated species tree branch lengths, increased bootstrap support for true positive edges, and reduced the number of highly supported false positives, compared to unbinned MP-EST analyses. These improvements increased when gene sequence alignments had low phylogenetic signal, the species tree had low ILS, or there were many genes.

The estimation of species tree branch lengths is biologically significant since these lengths are used to infer the amount of ILS in the data. Unbinned MP-EST analyses tended to substantially underestimate branch lengths (and thus over-estimate ILS), but both weighted and unweighted binning reduce this problem and produce branch lengths that are much closer to their true lengths. Since MP-EST tends to over-estimate ILS in the presence of gene tree estimation error, this means that predictions of ILS levels for biological datasets may have been over-estimated. Another consequence of this observation is that the biologically-based model species trees used here and in [90, 91] may have inflated levels of ILS, since they used MP-EST to construct the model species tree. If so, then performance under the lower ILS levels (species tree branch lengths of 2X or larger) might be closer to the biological dataset conditions than the default 1X condition and higher ILS conditions.

The improvement in branch support is biologically relevant, especially since unbinned MP-EST analyses sometimes produced highly supported false positive branches in the presence of poorly estimated gene trees and low levels of ILS, but binning reduced the incidence of these false positive branches with high support.

The results on small numbers of species, and in particular on the higher ILS 10-taxon datasets, show somewhat different trends. While results on the 15-taxon datasets showed binning generally being helpful or neutral, statistical binning ranged from neutral to detrimental on the higher ILS 10-taxon datasets (however, the differences were not statistically significant). Both the higher ILS 10-taxon and 15-taxon datasets had extremely high levels of ILS (the two highest we examined – average topological distance between true gene trees and the true species trees of 84% and 82%, respectively). Given that statistical binning ranged from neutral to highly beneficial for all the other model conditions, these data suggest that statistical binning may not be suitable to datasets with extremely high ILS levels. Clearly, further research is therefore needed to understand the conditions under which binning will be beneficial and where binning may reduce accuracy.

This study also did not examine model conditions in which gene tree estimation error is due to model misspecification, nor other biological causes for gene tree discord, such as gene duplication and loss or horizontal gene transfer. Furthermore, while we examined sequence datasets with varying numbers of sites for each locus (including some with 100bp), even shorter sequences may be needed to avoid loci that include any recombination [41].

This study mainly examined the impact of statistical binning on MP-EST, and examined its impact on ASTRAL only for a subset of the data and only with respect to species tree topology estimation (instead of the full set of criteria). Thus, an important direction for future study is to consider other coalescent-based methods for estimating the species tree from multiple loci. As a simple example, Mirarab *et al.* [91] showed that the accuracy of MP-EST species trees depended on whether MLBS or best maximum likelihood (BestML) gene trees were used, and that MP-EST trees based on BestML gene trees generally produced more accurate species tree topologies for datasets with large numbers of genes (such as some of the model conditions studied in this study). The explanation offered for this is that BestML gene trees are generally closer to the true gene tree than MLBS gene trees, and that this helps coalescent-based species tree estimation. Hence, the evaluation of the impact of binning on MP-EST with BestML gene trees is also needed. It is also possible that better results would be obtained using Bayesian methods (such as MrBayes [55]), rather than MLBS, to generate the distribution of gene trees [23], since the posterior distribution produced by Bayesian MCMC methods may be more closely centered around the true gene tree than the MLBS sample.

This study suggests that substantial improvement in species tree estimation could be obtained if we can develop more accurate methods for gene tree estimation. For example, methods that co-estimate gene sequence alignments and trees, such as BAli-Phy [118], SATé [76, 77], and PASTA [92], might provide improved gene tree estimation accuracy, compared to standard two-step procedures for estimating trees (first align, and then compute the tree).

Indeed, another challenge is that *if* loci are restricted to ultra-short sequences (10-50 sites), so as to decrease the probability of intra-locus recombination, then approaches based on combining estimated gene trees may not be able to provide highly accurate results, no matter what techniques are used to estimate gene trees. Hence, it is also possible that methods that construct species trees directly from the sequence data, rather than by combining gene trees, will have the best accuracy (see, for example, [18, 21]), since they can avoid the analytical and empirical challenges caused by gene tree estimation error.

However, as observed in this and other studies [41, 74], concatenation often produces more accurate trees than even the best coalescent-based methods when the level of ILS is low enough. Therefore, an important question is whether a given biological dataset has a sufficiently high level of ILS that a coalescent-based analysis is needed. Conversely, coalescent-based methods that are not only more accurate than concatenation under conditions with high ILS, but also comparably accurate even under low levels of ILS, would be very helpful tools.

Finally, since statistical binning did reduce accuracy for some of the data we examined with small numbers of species and the very highest ILS levels, an important question that needs to be addressed is whether these very high ILS simulation conditions explored here and elsewhere represent realistic levels of ILS, or whether they represent extreme conditions that are unlikely to be observed in nature. Accurate estimations of ILS levels in biological data would enable the research community to direct its efforts to developing methods that would have the greatest utility in practice.

Chapter 9

Disk Covering Methods Improve Phylogenomic Analyses

With the rapid growth rate of newly sequenced genomes, species tree inference from multiple genes has become a basic bioinformatics task in comparative and evolutionary biology. However, accurate species tree estimation is difficult in the presence of gene tree discordance, which is often due to incomplete lineage sorting (ILS), modelled by the multi-species coalescent. Several highly accurate coalescent-based species tree estimation methods have been developed over the last decade, including MP-EST. However, the running time for MP-EST increases rapidly as the number of species grows.

We present divide-and-conquer techniques that improve the scalability of MP-EST so that it can run efficiently on large datasets. Surprisingly, this technique also improves the accuracy of species trees estimated by MP-EST,

Much of the material in this chapter is taken without alteration from the following paper.

- M. S. Bayzid, T. Hunt, and T. Warnow. Disk covering methods improve phylogenomic analyses. *BMC Genomics*, 15(Suppl 6):S7, 2014

MSB and TW designed the study; MSB implemented the SSG and DACTAL decompositions, performed the analyses; TH implemented DCM1-based decomposition and performed analyses; MSB and TW wrote the paper.

as our study shows on a collection of simulated and biological datasets.

9.1 Introduction

A standard approach to species tree estimation uses multiple loci and then concatenates alignments for each locus into a super-matrix, which is then used to estimate the species tree. When genes all evolve down the same tree topology under the same well-behaved process, then statistical methods of phylogeny estimation (such as maximum likelihood) applied to the concatenated alignment are statistically consistent, and so will return the true tree with high probability given a large enough number of sites or genes. However, when the genes evolve down different tree topologies, which can happen in the presence of gene duplication and loss, horizontal gene transfer, or incomplete lineage sorting, then there are no statistical guarantees for concatenated analyses. Furthermore, simulations have shown that concatenation can return incorrect trees with high confidence in the presence of incomplete lineage sorting [69], a population-level process modelled by the multi-species coalescent [66]. Because incomplete lineage sorting is expected to occur under many biologically realistic conditions (and especially in the presence of rapid radiations), coalescent-based species tree methods with statistical guarantees of returning the true tree with high probability (as the number of genes increases) have been developed, and are increasingly popular [52, 68, 73, 78, 80, 81, 96].

Only some of these coalescent-based methods are fast enough to be used with phylogenomic datasets that contain hundreds or thousands of genes and more than 30 or so species. For example, the fully-parametric coalescent-based methods, such as BEST [78] and *BEAST [52] that co-estimate gene trees and species trees, are limited to approximately 20 species and 100 genes (and even datasets of this size can be extremely difficult) [8, 128]. The other type of coalescent-based method are called "summary methods" because they estimate species trees by combining estimated gene trees. These methods tend to be much faster than the fully-parametric methods, and some of these methods (e.g., MP-EST [80]) are able to be used with hundreds to thousands of genes.

However, even the fast summary methods can be computationally intensive on large datasets. For example, MP-EST, which has been used in many biological dataset analyses [17, 71, 129, 159], uses a heuristic search to solve an NP-hard pseudo-maximum likelihood optimization problem (based on the triplet gene tree distribution). Our evaluation of MP-EST (reported in this chapter) shows that the number of species greatly impacts the running time; thus, improving MP-EST's scalability (in terms of the number of species) is an important objective.

This chapter introduces two general techniques for improving the scalability of coalescent-based species tree estimation methods so that they can analyze datasets with large numbers of species. Each technique uses an initial tree estimated on the set of species to divide the species dataset into small overlapping subsets, applies the species tree estimation method to each subset of species to produce an estimated species tree for that subset, and then combines the estimated species trees (each on a subsets of the species) into a tree on the full set of species. Furthermore, each technique can iterate, and thus return a set of candidate species trees from which the final tree is selected. The only difference between the two techniques is how the dataset is divided into subsets, with one technique using the dataset decomposition technique from DACTAL [100] and the other using a modification of the dataset decomposition technique from Rec-I-DCM3 [123].

We evaluate these two techniques on a collection of simulated and biological datasets, and show that both reduce the running time of MP-EST, one of the most popular coalescent-based summary methods. Surprisingly, these two techniques also improve the accuracy of MP-EST. Thus, the two techniques improve the scalability of MP-EST, the leading coalescent-based species tree estimation, so that it can be run on datasets with large numbers of species and provide improved topological accuracy.

9.2 Methods

Disk-Covering Methods (DCMs) are meta-methods (employing divide-and-conquer and in some cases also iteration) designed to "boost" the performance of the existing phylogenetic reconstruction methods [58, 59, 97, 123]. The major steps of DCMs are: (i) decomposing the dataset into overlapping subsets of taxa, (ii) estimating trees on these subsets using a preferred phylogenetic method, and finally (iii) merging the subtrees to get a tree on the full set of taxa. However, DCMs have not yet been used in the context of species tree estimation from multiple gene trees, which is the focus of this study. Although the approach we present can be used with any coalescent-based method (including ones that co-estimate gene trees and species trees, such as BEST and *BEAST), we study the technique specifically for use with MP-EST.

- Step 1: Compute a starting tree from the input set of gene trees; this is the initial guide tree (we show results using MP-EST and Matrix Representation with Parsimony (MRP) [115]).
- Step 2: Repeat for a user-specified number of iterations (we show 2 and 5).
  - Step 2a: Decompose the set of species into small overlapping subsets of taxa, with target subset size specified by the user (we show 15), using the current guide tree.
  - Step 2b: For each subset, create a set of gene trees by restricting the input gene trees to the species present in the subset (each such gene tree is called a subset gene tree), and then apply MP-EST to the subset gene trees to produce a newly estimated subset species tree.
  - Step 2c: Combine the subset species trees estimated in Step 2b using a supertree method (we use SuperFine+MRL [102]) , thus producing a tree on the full set of species. This is the new guide tree, and is used in the next iteration. We also add this tree to the set of guide trees produced during the algorithm.

- Step 3: Score each of the different guide trees produced during the algorithm with respect to the selected optimization criterion and return the tree with the best score.

We provide details for Step 2a and Step 3.

Step 2a: dataset decomposition techniques

We explored three different techniques for decomposing the set of species into subsets: DCM1 [149], DACTAL [100], and a decomposition we call the short subtree graph (SSG) [123]. The DCM1 decomposition improved MP-EST but was less computationally efficient than the SSG-decomposition or the DACTAL-decomposition. Therefore, we focus the remainder of our discussion on the other two techniques.

Definitions Let T be an edge-weighted guide tree on the set S of taxa. Let e be an internal edge in T, and t_1, t_2, t_3, t_4 be the four subtrees around the edge e (i.e., removing e and its two endpoints from T breaks T into four subtrees: t_1, t_2, t_3, t_4.). A *short quartet* around e contains four leaves, one from each of these four subtrees, where each leaf is selected to be the closest (according to the edge weights) in its subtree to e. Hence, the set of short quartets of a tree are obtained by taking all short quartets around all edges in the tree. We used a “padding” technique where we find a collection of closest leaves (e.g., 2 or 3, rather than just 1) from each of the four subtrees around e, and we call this a *padded short quartet*.

DACTAL-based decomposition

DACTAL uses a padded-Recursive-DCM3 decomposition (PRD), as follows. The input is a guide tree T (without edge weights) and target subset size ms. The PRD decomposition finds a "centroid" edge (i.e., an edge that splits the guide tree into two subtrees containing roughly equal numbers of leaves). The removal of this edge and its endpoints divides the tree into four subtrees, A, B, C and D. For each of these four trees, the set of at most $p/4$ (where p is the padding size, and $p < ms$) closest leaves to the edge e are selected, and put into a set X; four leaves selected from different subtrees around the centroid edge, using this technique, are called padded short quartets, generalizing the concept of short quartets where only the nearest leaf in each subtree is selected [58, 97]. However, if there are ties (i.e., leaves that are equally close to the branch e), then all leaves at the same (very close) distance are included in the set; thus, $|X| > p$ is possible. Then, the set of leaves present in $A \cup X$, $B \cup X$, $C \cup X$ and $D \cup X$ define four overlapping subsets. If any of these sets is larger than ms, then the decomposition is repeated recursively on that set until all subsets have size at most ms and the padding size requirement is satisfied. However, if the application of the decomposition cannot reduce the subset size, then the subset is returned. Thus, both p and ms are treated as targets rather than hard constraints. For the simulated datasets studied in this study, we set $p = 4$, which means that we only used short quartets (one leaf in each subtree around a centroid edge). However, for larger datasets, increasing p might lead to improved analyses.

SSG-based decomposition

The SSG-based decomposition technique we present in this chapter is similar to DCM3 decomposition presented in [123], but modified through the use of the "padding" (described above) so that there is more overlap between subsets.

Given input guide tree T and target maximum subset size (ms), the SSG-based decomposition creates a "padded" short subtree graph $\mathcal{G} = (V, E)$ as follows. First, we compute $p = \lfloor \frac{ms}{4} \rfloor$. We then compute the set of at most p closest leaves in each subtree around a given edge in the graph, and make a clique out of this set of (at most) $4p$ species. The graph containing all these cliques is the padded short subtree graph. Equivalently, the vertex set V contains the leaves in T (i.e., the species) and $(s_i, s_j) \in E$ if and only if there is some edge e in the guide tree T such that s_i and s_j are each among the p nearest leaves to e in their respective subtrees. Because a padded short subtree graph is chordal, it contains at most $n = |V|$ maximal cliques, and these can be found in polynomial time [42]. Note that typically the number of vertices in the maximal cliques will be at most ms, but some of them can be slightly bigger than ms. Thus, as with DACTAL, ms is a target maximal value, and not a strict upper bound on the size of any subset we analyze.

Step 3: Selecting the best tree across different iterations

We explored two different optimality criteria – the maximum pseudo-likelihood score computed by MP-EST, which is based on the rooted triplet tree distribution, and a "quartet support score" [93]. The quartet support measures the similarity between a candidate tree T and the input gene trees, and is computed as follows. We decompose each input gene tree into its induced set of quartet trees (i.e., unrooted trees formed by picking four leaves). The quartet support score of a given candidate species tree T is the total, over all the input gene trees, of the number of induced quartet trees that T agrees with. As shown in [80], the tree that optimizes the maximum pseudo-likelihood score is a statistically consistent estimator of the true species tree under the multi-species coalescent model. Interestingly, the same is true of the quartet support score, as shown in [93].

9.3 Experiments

We explore the performance of MP-EST [80] and these boosted versions of MP-EST on a collection of simulated and biological datasets. We compare the estimated species trees to the model species tree (for the simulated datasets) or to the scientific literature (for the biological datasets), to evaluate accuracy. The tree error is measured using the missing branch rate (also called the false negative rate), which is the percentage of the internal edges in the model tree that are missing in the estimated tree. We measure the statistical significance of the results by Wilcoxon signed-rank test with $\alpha = 0.05$.

9.3.1 Mammalian simulated datasets

We used datasets generated in another study [93] to explore performance of coalescent-based methods for estimating species trees. These datasets have gene sequence alignments generated under a multi-stage simulation process, which begins with a species tree estimated on a mammalian dataset (studied in [129]) using MP-EST, simulates gene trees down the species tree under the multi-species coalescent model (so that the gene trees can differ topologically from the species tree), and then simulates gene sequence alignments down the gene trees under the GTRGAMMA model. We direct the reader to [93] for full details.

The basic model species tree has branch lengths in coalescent units, and we produced other model species trees by rescaling the branch lengths. This rescaling varies the amount of ILS (shorter branches have more ILS), and also impacts the amount of gene tree estimation error and the average bootstrap support (BS) in the estimated gene trees. The model condition with reduced ILS was created by uniformly doubling (2X) the branch lengths, and two model conditions with higher ILS were generated by uniformly dividing the branch lengths by two (0.5X) and five (0.2X). The amount of ILS obtained without adjusting the branch lengths is referred to as "moderate ILS", and was estimated by MP-EST on the biological data. Each model species tree was then used to generate gene trees under the multi-species coalescent model. The branch lengths in the gene trees were then modified to deviate from the strict molecular clock, and sequences were simulated down each gene tree under the

GTRGAMMA model.

Maximum likelihood (ML) gene trees were estimated on each sequence alignment using RAxML [130] under the GTRGAMMA model, with 200 bootstrap replicates to produce bootstrap support on the branches. The average bootstrap support (BS) in the biological data was 71%, and the sequence lengths were set to produce estimated gene trees with average BS bracketing that value – 500bp alignments produced estimated gene trees with 63% average BS and 1000bp alignments produced estimated gene trees with 79% average BS.

The number of genes ranged from 50 to 800 to explore both smaller and larger numbers of genes than the full biological dataset (which had roughly 400 genes). For each model condition (specified by the ILS level, the number of genes, and the sequence length), we created 20 replicate datasets.

9.3.2 Biological datasets

We analyzed two biological datasets – the mammalian dataset from [129] containing 37 species and 424 genes, and the amniota dataset from [17] containing 16 species and 248 genes – using MP-EST and both versions of boosted MP-EST. We set $ms = 15$ for the mammalian dataset, and $ms = 10$ for the amniota dataset.

9.4 Results

9.4.1 Running time on simulated datasets

Our first experiment evaluated the running time of MP-EST on different-sized subsets of the simulated mammalian datasets; see Figure 9.1. Note the fast increase in running time, so that MP-EST completed in 11 seconds on 8-taxon subsets, in 25 seconds on 10-taxon subsets, and in 150 seconds on 15-taxon subsets. Furthermore, MP-EST took 6900 seconds (115 minutes, or nearly two hours) to analyze the 37-taxon mammalian dataset.

In contrast, each iteration of boosted MP-EST requires much less time: 12 minutes per iteration for SSG-boosting and 7 minutes per iteration for DACTAL-boosting, each run sequentially.

The vast majority of the running time for both the DCM-boosted and SSG-boosted versions of MP-EST is in computing the starting tree (if it uses MP-EST or some other slow method) and when it runs MP-EST on subsets; all the other steps completed in seconds, run sequentially. The decomposition requires each subset to be no more than 15 species, but the average size of each subset under the SSG- and DACTAL-based decompositions was between 12 and 13; hence, MP-EST on each subset took about one minute to analyze. The number of subsets generated by the SSG-based decomposition ranged from 9 to 11, and used approximately 9-11 minutes. DACTAL decomposition typically generated only 4-5 subsets (two cases with 7 subsets), and used approximately 4-5 minutes. Thus, the DACTAL-based analysis and SSG-based analysis produced subsets of approximately the same size, but DACTAL-based analyses had generally half the number of subsets to analyze, and so took about half the time. We also observed (Fig. 9.2, 9.3 and 9.4) that two iterations of DACTAL-boosting achieved about the same accuracy (and sometimes better accuracy) as five iterations of SSG-boosting. Thus, DACTAL-boosting provides running time benefits compared to SSG-boosting. Finally, since using MP-EST as the starting tree is computationally expensive, we also evaluated boosting using MRP, which is a very fast method for computing the starting tree, but which is not as accurate as MP-EST for species tree estimation in the presence of ILS; see below for these results.

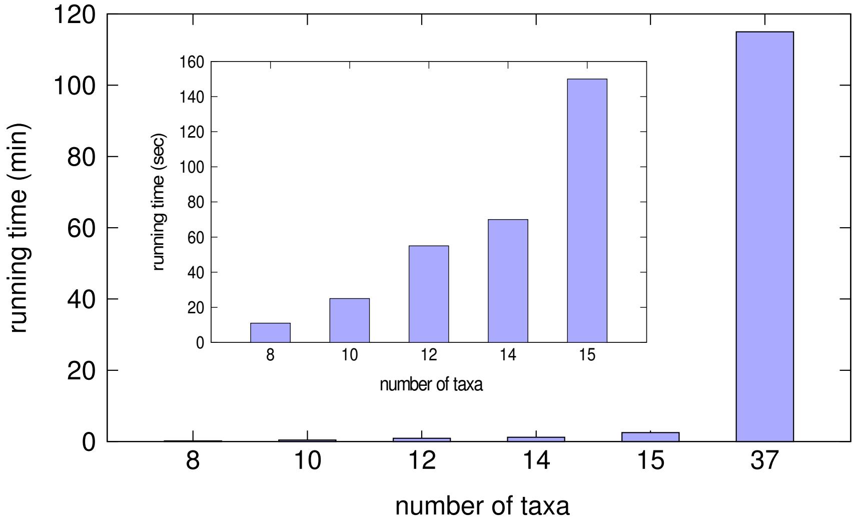


Figure 9.1: **Running time of MP-EST for varying number of taxa.** We show the running time of MP-EST on the simulated mammalian datasets for varying number of taxa on the model condition with moderate level of ILS, 200 genes and 500bp sequence length. The inset subfigure shows results in seconds for 8 to 15 taxa, and the larger figure also shows results in minutes on datasets with up to 37 taxa.

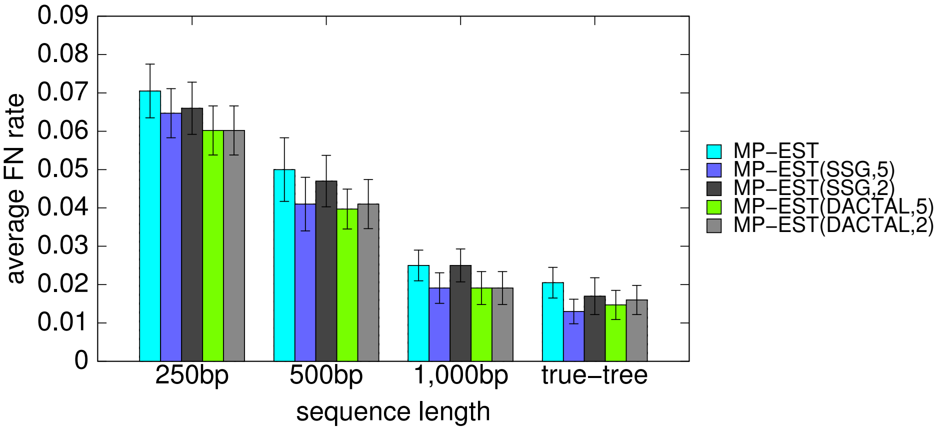


Figure 9.2: **Average FN rates of boosted MP-EST after two and five iterations.** We show the average FN rates of the best trees, with respect to the quartet support, after two and five iterations of SSG and DACTAL-based boosting on the simulated mammalian datasets with varying sequence length (200 genes, moderate amount of ILS).

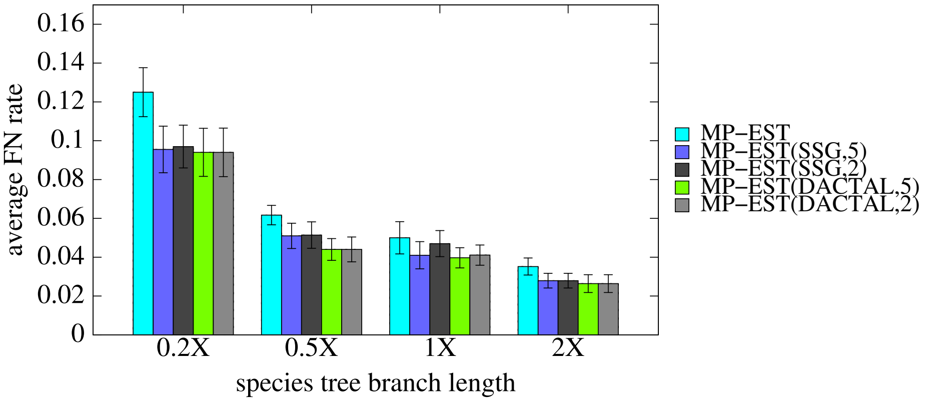


Figure 9.3: **Average FN rates of boosted MP-EST after two and five iterations.** We show the average FN rates of the best trees, with respect to the quartet support, after 2 and 5 iterations of SSG and DACTAL-based boosting on the simulated mammalian datasets with varying amount of ILS (200 genes, 500bp).

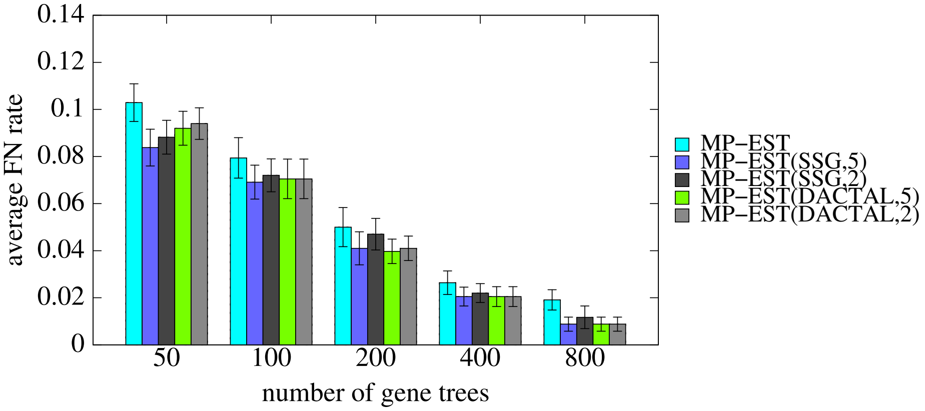


Figure 9.4: **Average FN rates of boosted MP-EST after two and five iterations.** We show the average FN rates of the best trees, with respect to the quartet support, after 2 and 5 iterations of SSG- and DACTAL-based boosting on the simulated mammalian datasets with varying numbers of gene trees (moderate amount of ILS, 500bp).

9.4.2 Impact of boosting on topological accuracy for simulated datasets

We compared the accuracy and running time for various boosting techniques. We used MP-EST to produce the starting tree, and then ran five different iterations of DACTAL-boosting and SSG-boosting, using different subset sizes (from 15 to 22), and using different criteria (maximum pseudo-likelihood as computed by MP-EST or quartet support) to select the final tree.

As noted above, DACTAL-boosting or SSG-boosting produced the same results after five iterations. Analyses based on decompositions into subsets of size 15 completed more quickly than decompositions into larger subsets, and all subset sizes we explored (15-22) produced comparable accuracy. Finally, using quartet support scores rather than maximum pseudo-likelihood scores to select the output species tree had better overall results (Fig. 9.5, 9.6 and 9.7). Based on these preliminary results, we set default algorithmic parameters as follows: DACTAL decomposition, subsets of size 15, and selecting the final tree using the quartet support score. However, we show results for different combinations of the algorithmic parameters below.

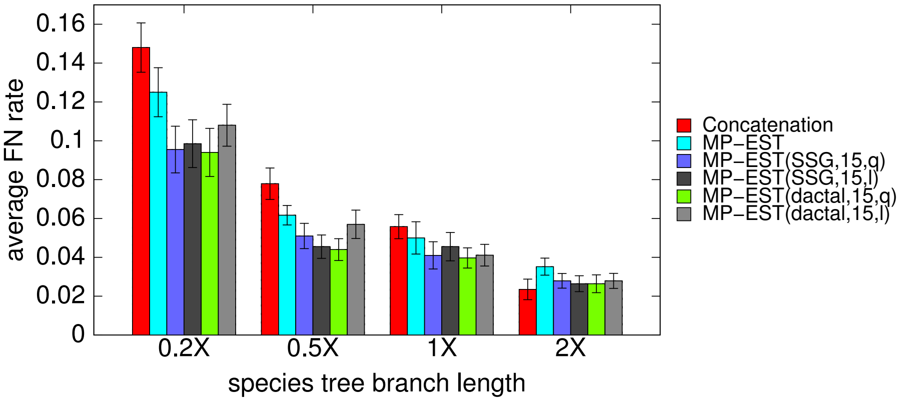


Figure 9.5: **Impact of how the final tree is selected (using quartet support or pseudo-likelihood) in boosted versions of MP-EST.** We show average FN rates of MP-EST (with and without boosting) on the simulated mammalian datasets with varying amount of ILS, using two different ways of selecting the final tree: quartet support (q) or pseudo-likelihood (l). We fixed the number of genes to 200 and sequence length to 500bp, while varied the amount of ILS. 2X model condition contains the lowest amount of ILS while 0.2X refers to the model conditions with the highest amount of ILS. We show the results for SSG and DACTAL-based decomposition with maximum subset size 15.

Figure 9.8 shows the average FN rates of concatenation using maximum likelihood, MP-EST, and boosted MP-EST (using both DACTAL and SSG-based boosting). The results for boosting are based on starting with

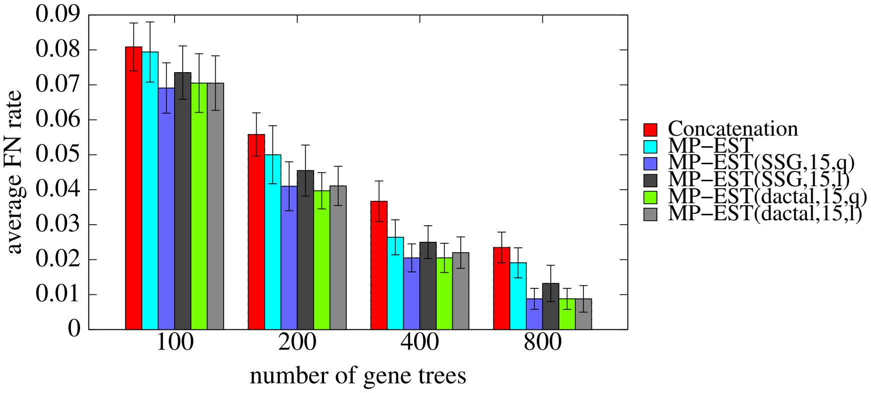


Figure 9.6: **Impact of how the final tree is selected (using quartet support or pseudo-likelihood) in boosted versions of MP-EST.** We show average FN rates of MP-EST (with and without boosting) on the simulated mammalian datasets with varying numbers of gene trees, using two different ways of selecting the final tree: quartet support (q) or pseudo-likelihood (l). We fixed the amount of ILS to moderate level (1X) and sequence length to 500bp, and varied the number of genes from 100 to 800. We show the results for SSG- and DACTAL-based decompositions with maximum subset size 15.

the MP-EST tree, then performing 5 iterations and selecting the species tree based on the quartet support. Both ways of boosting improved the accuracy of MP-EST across all levels of ILS, and were substantial on the model conditions with increased ILS (0.5X and 0.2X). We measured the statistical significance of the results using Wilcoxon signed-rank test (*p*-values given in Table 9.1). With the exception of the 1X model condition, the improvements of DACTAL-boosted MP-EST over un-boosted MP-EST were statistically significant (*p* values are 0.002, 0.009, 0.09 and 0.04 for 0.2X, 0.5X, 1X and 2X model conditions respectively). The improvements of SSG-boosted MP-EST over un-boosted MP-EST were statistically significant for the highest ILS level

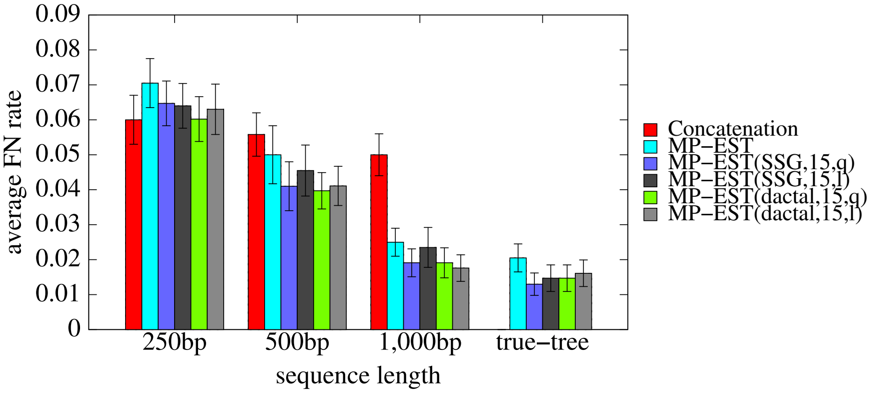


Figure 9.7: **Impact of how the final tree is selected (using quartet support or pseudo-likelihood) in boosted versions of MP-EST.** We show average FN rates of MP-EST (with and without boosting) on the simulated mammalian datasets with varying numbers of gene trees, using two different ways of selecting the final tree: quartet support (q) or pseudo-likelihood (l). We fixed the amount of ILS to moderate level (1X) and number of genes to 200, and varied the sequence lengths from 250bp to 1000bp. We show the results for SSG- and DACTAL-based decompositions with maximum subset size 15.

(0.2X, $p = 0.006$), but not for the other levels (p values were 0.13, 0.08 and 0.117 for 0.5X, 1X and 2X model conditions, respectively).

Concatenation is expected to be less accurate than coalescent-based methods when there is substantial ILS, and this is what we observed in these experiments. Thus, with the exception of the 2X model condition (which had the least ILS), concatenation was less accurate than both MP-EST and boosted MP-EST. Interestingly, the improvement of concatenation over boosted MP-EST on the 2X model condition was not statistically significant ($p = 0.33$ and $p = 0.4$ for SSG- and DACTAL-based boosting, respectively). Also, on

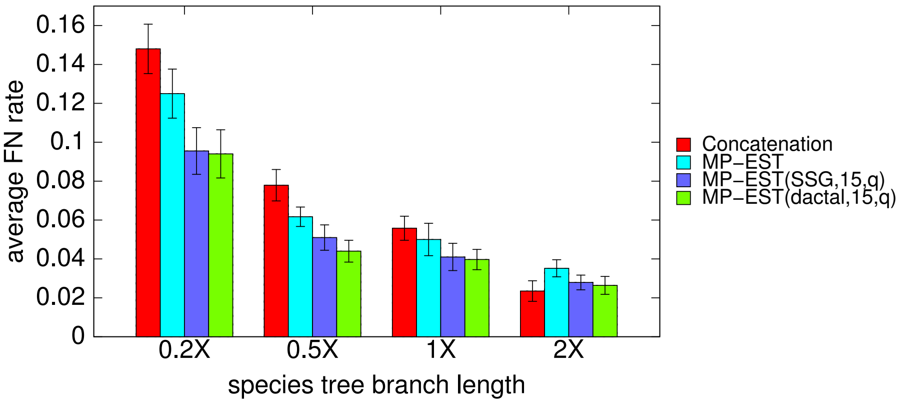


Figure 9.8: **Average FN rates of MP-EST (with and without boosting) for different levels of ILS.** Average FN rates of MP-EST (with and without boosting) over 20 replicates on the simulated mammalian datasets with varying amount of ILS. We also show the FN rate of concatenation. We fixed the number of genes to 200 and sequence length to 500bp, while varied the amount of ILS. 2X model condition contains the lowest amount of ILS while 0.2X refers to the model conditions with the highest amount of ILS. We show the results for short subtree graph (SSG) and DACTAL-based decompositions with maximum subset size 15. We show the FN rate of the best tree with respect to quartet support (as denoted by q in the figure legend) across five iterations.

the moderate level of ILS (1X), concatenation and MP-EST had very close performance, but boosted MP-EST was more accurate than concatenation. However, the differences between boosted MP-EST and concatenation were not statistically significant ($p = 0.08$ and $p = 0.11$ for DACTAL and SSG-based boosting respectively).

Figure 9.9 shows the comparison between unboosted and boosted MP-EST using both SSG- and DACTAL-based decomposition on the simulated mammalian datasets with 50 to 800 genes, moderate levels of ILS (1X), and sequence length set to 500bp. Both SSG and DACTAL-based decomposition

	p-values				
Model condition	CA vs. MP-EST	MP-EST vs. MP-EST (SSG)	MP-EST vs. MP-EST (DACTAL)	CA vs. MP-EST (SSG)	CA vs. MP-EST (DACTAL)
0.2X,200gt,500bp	0.014	0.006	0.002	0.0002	0.0001
0.5X,200gt,500bp	0.03	0.13	0.009	0.01	0.003
1X,200gt,500bp	0.433	0.08	0.09	0.11	0.08
2X,200gt,500bp	0.06	0.117	0.04	0.33	0.45
1X,50gt,500bp	0.023	0.003	0.06	0.41	0.31
1X,100gt,500bp	0.39	0.02	0.09	0.1	0.16
1X,400gt,500bp	0.08	0.09	0.09	0.02	0.02
1X,800gt,500bp	0.27	0.01	0.01	0.008	0.008
1X,200gt,250bp	0.22	0.18	0.01	0.27	0.49
1X,200gt,1000bp	0.0004	0.1	0.06	0.0002	0.0004
1X,200gt,true gene tree	NA	0.03	0.06	NA	NA

Table 9.1: ***p*-values measured by Wilcoxon signed-rank test for the simulated mammalian datasets**. We evaluate the statistical significance of differences in species tree topology using Wilcoxon signed-rank test with $\alpha = 0.05$. We show the p-values indicating whether the differences between two methods are statistically significant. We compare concatenation (CA) and MP-EST (unboosted) with SSG and DACTAL-boosted MP-EST.

improved MP-EST in all cases, sometimes substantially. The improvements of SSG-based boosting over un-boosted MP-EST were statistically significant except for the 200- and 400-gene cases (p values were 0.003, 0.02, 0.08, 0.09, and 0.01 for model conditions with 50, 100, 200, 400, and 800 genes, respectively). DACTAL-based boosting was significantly better than un-boosted MP-EST on the 800-genes case but not on the others (p values were 0.06, 0.09, 0.09, 0.09 and 0.01 for model conditions with 50, 100, 200, 400, and 800 genes, respectively).

The comparison between concatenation and (boosted) MP-EST is also interesting. For the 50-gene case, concatenation was more accurate than unboosted MP-EST, but DACTAL-boosted MP-EST matched the accuracy of concatenation, and SSG-boosted MP-EST was slightly more accurate than concatenation. For other cases (100-800 genes), the differences between concatenation and MP-EST were not statistically significant ($p > 0.05$), but both SSG-boosted and DACTAL-boosted versions of MP-EST were more accurate than concatenation. Furthermore, the improvement of boosted MP-EST over concatenation were statistically significant for 400- and 800-gene cases ($p = 0.02$ and 0.008 for the 400- and 800-gene cases, respectively, for both SSG and DACTAL-based boosting).

Figure 9.10 compares boosted and un-boosted MP-EST on the mammalian datasets with varying sequence lengths. We fixed the amount of ILS to the moderate level (1X) and number of genes to 200, while varying the sequence lengths from 250bp to 1000bp. We also show the results on true gene trees (i.e., without estimation error). Boosting improved the accuracy of MP-EST in all cases. The improvements were statistically significant for the 250bp case with DACTAL-based boosting and on the true trees for both types of boosting ($p < 0.05$). On the 250bp condition (which has the highest gene tree estimation error) concatenation was more accurate than MP-EST, and boosted MP-EST matched concatenation.

9.4.3 Results on biological datasets

Amniota dataset. We analyzed data for 248 genes on 16 amniota species from Chiari et al. [17]. Previous studies had placed turtles as the sister to birds and crocodiles (Archosaurs) [57, 60, 156]. Chiari et al. [17] used concatenation and MP-EST with multi-locus bootstrapping on two sets of gene

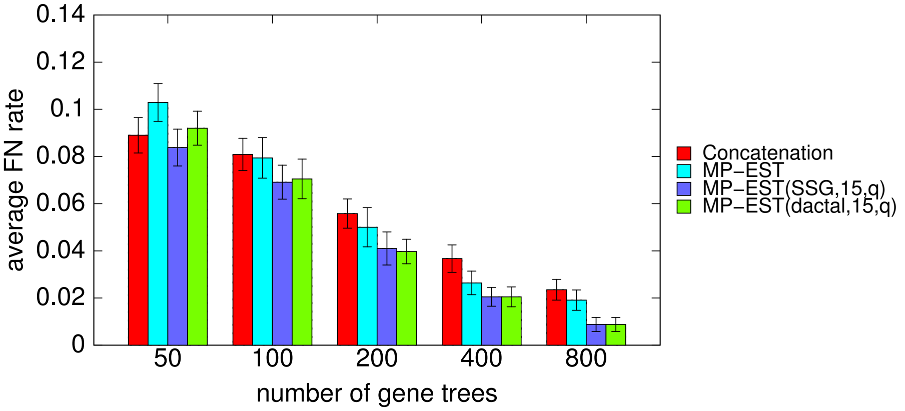


Figure 9.9: **Average FN rates of MP-EST (with and without boosting) for different number of gene trees.** Average FN rates of MP-EST (with and without boosting) over 20 replicates on the simulated mammalian datasets with varying number of gene trees. We also show the FN rate of concatenation. We varied the number of genes from 100 to 800, while set the amount of ILS to 1X level and the sequence length to 500bp. We show the results for short subtree graph (SSG) and DACTAL-based decompositions with maximum subset size 15. We show the FN rate of the best tree with respect to quartet support (as denoted by q in the figure legend) across five iterations.

trees – one based on amino acid (AA) and the other based on nucleotide (NT) alignments. Concatenation and MP-EST on the AA gene trees resolved the clade as (turtles,(birds,crocodiles)) (i.e., birds and crocodiles were considered sister taxa, consistent with the earlier studies) while MP-EST on the NT data produced (birds,(turtles,crocodiles)), and so contradicted the previous studies. Because the concatenation tree and the MP-EST(AA) tree agreed and were consistent with previous studies, the resolution with turtles as sister to birds and crocodiles was considered more likely to be correct.

We ran MP-EST on the NT datasets containing 248 gene trees with 10

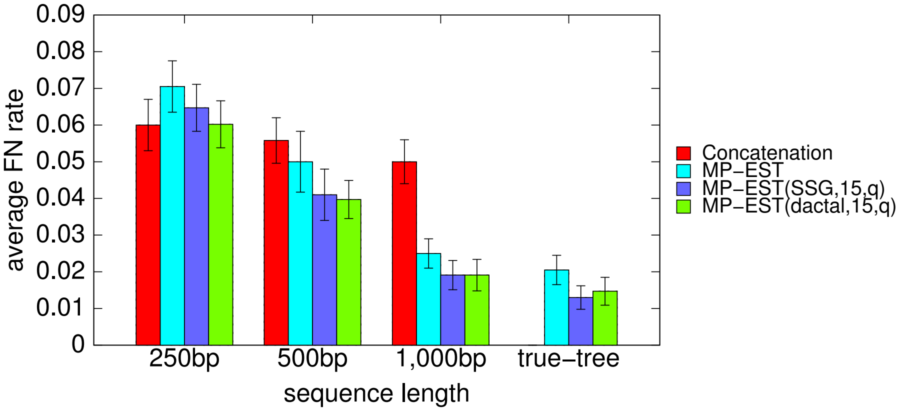


Figure 9.10: **Average FN rates of MP-EST (with and without boosting) for different sequence lengths.** Average FN rates of MP-EST (with and without boosting) over 20 replicates on the simulated mammalian datasets with different amounts of gene tree estimation error by varying the sequence lengths. We also show the FN rate of concatenation. We varied the sequence lengths from 250bp to 1000bp with 200 genes and moderate amount of ILS (1X). We show the results for short subtree graph (SSG) and DACTAL-based decompositions with maximum subset size 15. We show the FN rate of the best tree with respect to quartet support (as denoted by q in the figure legend) across five iterations.

independent runs and retained the tree with maximum likelihood value; this produced the same tree reported in [17]. We then ran four versions of boosted MP-EST, with SSG- and DACTAL-based decompositions, and using the MP-EST starting tree. For each analysis, we ran five iterations and retained the tree with the highest quartet support across the five iterations. All variants produced the same tree, resolving Archosaurs as (turtles,(birds,crocodiles)) (Fig. 9.11). Thus, the boosted MP-EST trees were consistent with concatenation and other previous studies.

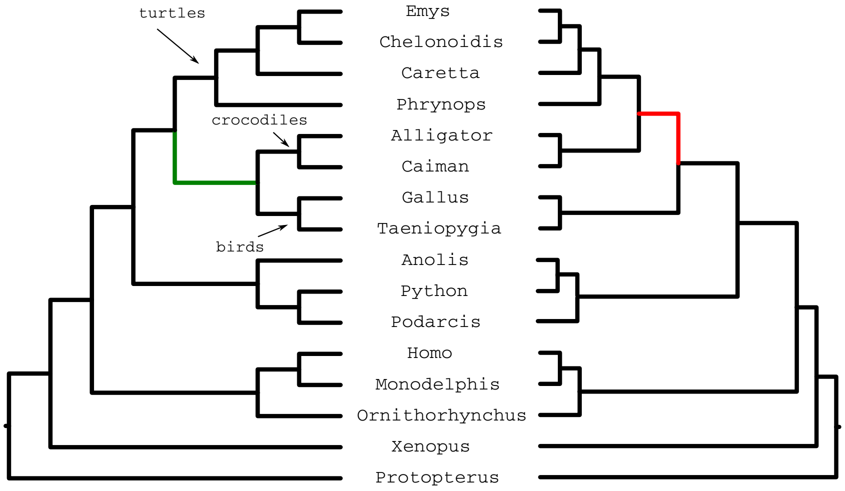


Figure 9.11: **Analyses of the amniota dataset using MP-EST (with and without boosting).** We show the trees estimated by MP-EST (right) and SSG and DACTAL-boosted MP-EST (left) using the MP-EST and MRP-estimated starting tree on the nucleotide amniota dataset from [17]. The sister relationship of crocodiles and birds is considered reliable, and is recovered in the SSG-boosted MP-EST tree. However, the MP-EST analysis of this dataset places crocodiles as sister to turtles (indicated by the red edge), and is not considered reliable.

Mammalian dataset. Song *et al.* [129] analyzed a dataset with 447 genes across 37 mammalian species using MP-EST and concatenation. In our analysis of this data we detected 21 genes with mislabelled sequences (incorrect taxon names, confirmed by the authors) which we removed from the dataset. We also identified two additional gene trees that were clearly topologically very different from all other gene trees, and removed these as well. We ran MP-EST on the 424 gene trees with SSG and DACTAL-based boosting using the MP-EST starting tree. All analyses we ran produced the same tree (see

Fig. 9.12).

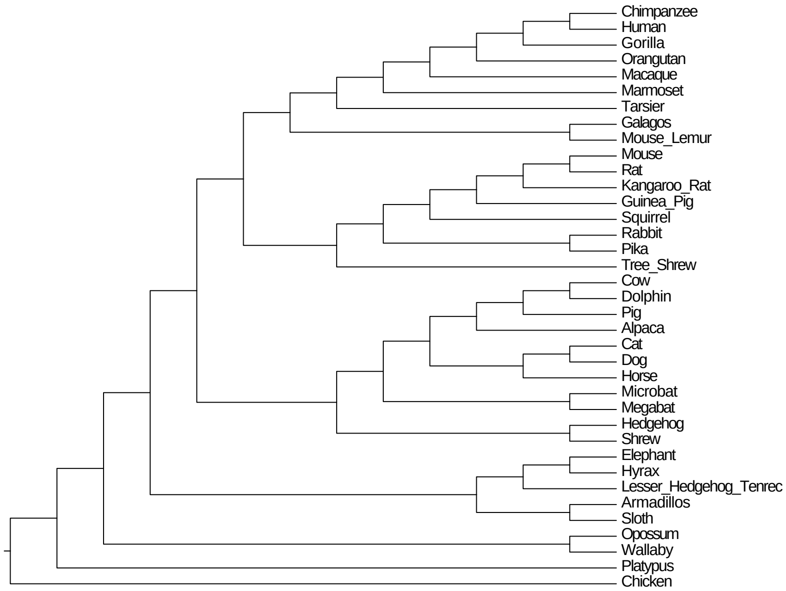


Figure 9.12: **Analyses of the mammalian dataset using MP-EST (with and without boosting).** MP-EST with SSG- and DACTAL-based boosting using both MP-EST and MRP-estimated starting tree produced the same tree as un-boosted MP-EST.

9.4.4 Pseudo-likelihood scores and quartet support values

Our analyses of the simulated and biological datasets showed that MP-EST always found trees with pseudo-likelihood scores that were at least as good as those found by any boosted MP-EST analysis, over all the iterations. In other words, the best pseudo-likelihood score was always found in the MP-EST starting tree. On the other hand, the best quartet support score was nearly always found in a subsequent iteration, for both types of boosting techniques.

The first of these observations suggests that MP-EST is doing a reasonably good job of solving its optimization problem, since boosting is not improving its search. The second of the observations is also very interesting, since the boosting techniques are not explicitly designed to optimize quartet support, and we have no explanation for this trend. Tables 9.2 and 9.3 show the log-likelihood values and the quartet supports on mammalian simulated datasets.

	Log-likelihood values			
Model condition	starting tree	best tree (SSG)	best tree (DACTAL)	model tree
0.2X,200gt,500bp	**-1338135**	-1458629	-1477619	-1338257
0.5X,200gt,500bp	**-1001218**	-1161407	-1280063	-1001269
1X,200gt,500bp	**-745312**	-903602	-1145433	-745342
2X,200gt,500bp	**-613190**	-808855	-782705	-613215
1X,100gt,500bp	**-370058**	-437613	-469161	-370154
1X,400gt,500bp	**-1486055**	-1952924	-1717131	-1486067
1X,800gt,500bp	**-2969112**	-3721911	-3407789	-2969119
1X,200gt,250bp	**-999941**	-1119309	-1145887	-1000048
1X,200gt,1000bp	**-563889**	-778484	-703988	-563904
1X,200gt,true gene tree	**-465251**	-755994	-628703	-465264

Table 9.2: **Average log likelihood values (over 20 replicates) for different species trees. We estimated the log likelihood values using MP-EST**. We show the likelihood values for the initial tree estimated by MP-EST and the true species tree (which is also estimated by MP-EST from the biological datasets). For SSG and DACTAL-based boosting, we find the best tree across the five iterations with respect to the log likelihood value estimated by MP-EST with branch length optimization. The best likelihood values are shown in bold.

	Quartet support			
Model condition	MP-EST	SSG	DACTAL	true species tree
0.2X,200gt,500bp	7818695	7819738	**7820187**	7816140
0.5X,200gt,500bp	10004913	10006656	**10006864**	10003929
1X,200gt,500bp	11269452	**11269960**	10006835	11266716
2X,200gt,500bp	11944097	11944516	**11944554**	11943759
1X,100gt,500bp	5635460	**5635757**	5635491	5630260
1X,400gt,500bp	22533544	22534293	**22534313**	22531906
1X,800gt,500bp	45095970	**45096812**	42841193	45096639
1X,200gt,250bp	10559948	**10560603**	10560467	10557321
1X,200gt,1000bp	11585974	**11586449**	11586514	11584969
1X,200gt,true gene tree	11745969	**11746174**	11746169	11744078

Table 9.3: **Average quartet supports of different species trees. We show the average (over 20 replicates) number of satisfied quartets (in the input gene trees) by different species trees for various model conditions**. For SSG and DACTAL-based boosting, we find the best tree across the five iterations with respect to the number of satisfied quartets. The best quartet support values are shown in bold.

9.4.5 Robustness to the starting trees

In the experiments shown so far, the starting tree was produced using MP-EST. We tested robustness to the starting tree by using MRP, a fast supertree technique, to compute a starting tree. However, MRP is not a statistically consistent method for estimating the species tree in the presence of ILS, and so is not likely to be as accurate as MP-EST.

Analyses of all biological datasets produced the same results, whether based on MRP or MP-EST starting trees. Results on the simulated datasets (Figs. 9.13, 9.14 and 9.15) show that MRP starting trees were generally not as accurate as MP-EST starting trees, but that five iterations of DACTAL-boosting from either starting tree produced essentially the same level of accuracy.

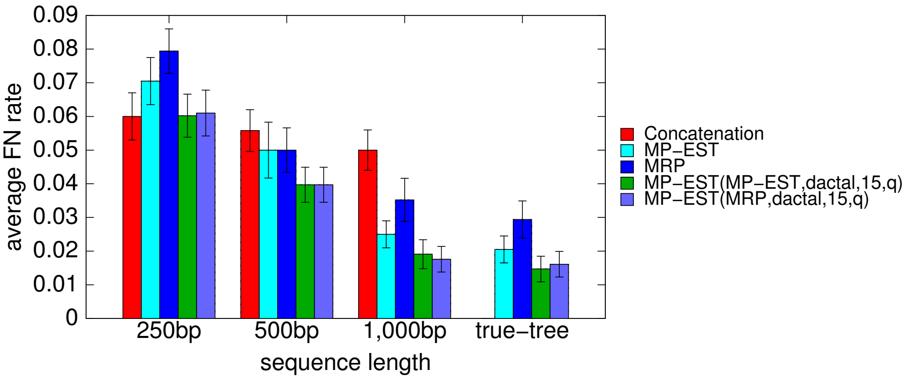


Figure 9.13: **Impact of different starting trees on DACTAL-based boosting with MP-EST.** We show the average FN rates of the best trees, with respect to the quartet support, after five iterations of DACTAL-based boosting using MP-EST and using the starting trees estimated by MRP and MP-EST on the simulated mammalian datasets with varying sequence length (200 genes, moderate amount of ILS). We ran MP-EST on the subsets produced by DACTAL-based decomposition with maximum subset size 15 using different starting trees. MP-EST(MRP,dactal,15,q) refers to the results obtained by using the MRP-estimated starting tree, while MP-EST(MP-EST,dactal,15,q) refers to the results obtained by using the starting tree estimated by MP-EST. We also show the FN rates of concatenation and the starting trees estimated by MP-EST and MRP.

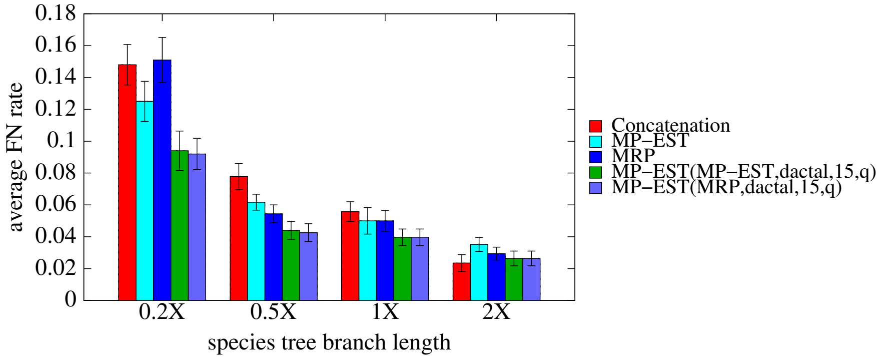


Figure 9.14: **Impact of different starting trees on DACTAL-based boosting with MP-EST.** We show the average FN rates of the best trees, with respect to the quartet support, after 5 iterations of DACTAL-based boosting using MP-EST and using the starting trees estimated by MRP and MP-EST on the simulated mammalian datasets with varying amount of ILS (200 genes and 500bp). We ran MP-EST on the subsets produced by DACTAL-based decomposition with maximum subset size 15 using different starting trees. MP-EST(MRP,dactal,15,q) refers to the results obtained by using the MRP-estimated starting tree, while MP-EST(MP-EST,dactal,15,q) refers to the results obtained by using the starting tree estimated by MP-EST. We also show the FN rates of concatenation and the starting trees estimated by MP-EST and MRP.

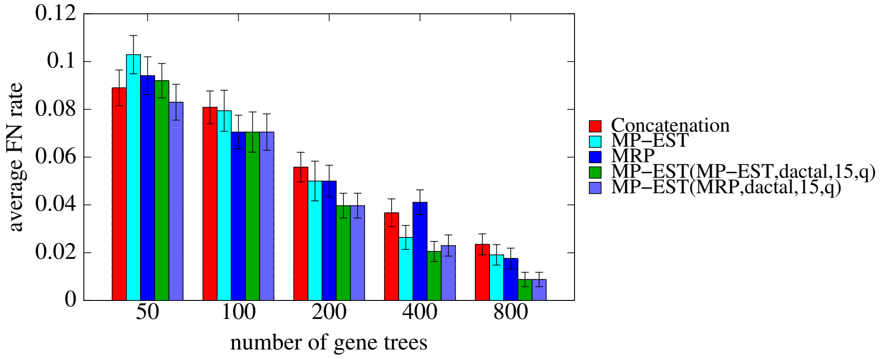


Figure 9.15: **Impact of different starting trees on DACTAL-based boosting with MP-EST.** We show the average FN rates of the best trees, with respect to the quartet support, after 5 iterations of DACTAL-based boosting using MP-EST and using the starting trees estimated by MRP and MP-EST on the simulated mammalian datasets with varying numbers of genes (500bp, moderate amount of ILS). We ran MP-EST on the subsets produced by DACTAL-based decomposition with maximum subset size 15 using different starting trees. MP-EST(MRP,dactal,15,q) refers to the results obtained by using the MRP-estimated starting tree, while MP-EST(MP-EST,dactal,15,q) refers to the results obtained by using the starting tree estimated by MP-EST. We also show the FN rates of concatenation and the starting trees estimated by MP-EST and MRP.

9.4.6 Statistical consistency

The following theorem is a direct corollary of Theorem 1 in [100].

Theorem 1: *Let T be the true species tree, and let S_1, $S_2, \ldots, S_k$ be the subsets created by a DACTAL- or SSG-decomposition with T as the starting tree. Let t_i be the true species tree on S_i, $i = 1, 2, \ldots, k$. Then the Strict Consensus Merger (and by extension also SuperFine+MRL), applied to the set $t_1, t_2, \ldots, t_k$, will return the species tree T.*

Comment: SuperFine+MRL has two steps: first it computes the Strict Consensus Merger (SCM), and then it resolves high degree nodes in the SCM tree using MRL. Therefore, if SCM produces a fully resolved tree, SuperFine+MRL returns the SCM tree.

Therefore, the following corollary can be easily proven:

Corollary 1: *If the starting tree is computed using a method that is statistically consistent under the multi-species coalescent model, then the pipeline based on either the DACTAL or SSG decomposition is statistically consistent under the multi-species coalescent model.*

9.5 Discussion

The results shown in this study suggest that using iteration and divide-and-conquer (within the DACTAL-based and SSG-based decomposition techniques) improved the topological accuracy of MP-EST. Furthermore, the specific choice of dataset decomposition technique (DACTAL-based or SSG-based) had little impact on accuracy. The improvement obtained by selecting trees based on their quartet support scores instead of their maximum pseudo-likelihood scores is very interesting, and suggests the possibility that although both optimality criteria are statistically consistent ways of searching for species trees under the multi-species coalescent, the quartet support score might have better empirical performance than the pseudo-likelihood score, at least under some conditions.

While most of the analyses were based on using MP-EST to produce the starting tree, we also showed that using MRP (a supertree method) to produce the starting tree resulted in comparable accuracy after five iterations. Since MRP generally produced less accurate starting trees than MP-EST, this suggests that the boosting techniques are robust to the starting tree. Furthermore, MRP was very fast on these datasets, completing in just ten seconds. Thus, when used with MRP as a starting tree, the entire pipeline (computing the starting tree, running five iterations of DACTAL boosting, and selecting the final tree) completes in 35 minutes. By comparison, MP-EST run without boosting takes nearly 115 minutes (nearly two hours). Thus, boosting improves the speed of MP-EST. If we use SuperFine+MRL or SuperFine+MRP to compute the starting tree, then DACTAL-boosted MP-EST should be fast, even for large numbers of species, since computing the starting tree using SuperFine is typically very fast, even on large datasets [102]. Furthermore, although we do not explore datasets with more than 37 species, the running times in Figure 1 suggest that MP-EST may be computationally infeasible for datasets with a few hundred species. By contrast, boosted versions of MP-EST are likely to scale close to linearly with the number of species, and are embarrassingly parallel. Thus, large-scale analyses of even several hundred species should be feasible using boosted MP-EST.

While the improvement in speed was expected, the improvement in accuracy was unexpected, and merits discussion. One possibility is that the performance we observed is mainly the result of some specific property of the simulation conditions we explored in this study, and that a larger study might show a difference in relative performance between boosted and unboosted MP-EST. However, both boosted versions of MP-EST gave more accurate results on the biological amniota dataset, and so that is not likely to be the answer. As noted, MP-EST is a heuristic for maximum pseudo-likelihood, and so another possible explanation is that MP-EST might have difficulty finding good solutions to its optimization problem on large datasets. However, the trees found by MP-EST had ML scores that were at least as good (and most often better) than the trees produced in any iteration by the boosted versions of MP-EST. Thus, this was clearly not the reason boosting improves MP-EST.

Instead, the data suggests that the boosting technique leads to trees with better quartet support scores, and that using quartet support scores to select the best species tree might be helping these boosted versions of MP-EST to produce more accurate trees. This hypothesis is supported by the fact that selecting the best tree based on the quartet support produced improved topological accuracy compared to selecting the best tree based on the pseudo-likelihood score, and that the quartet support optimization criterion is statistically consistent under the multi-species coalescent model [93].

9.6 Conclusion

MP-EST is one of the popular methods for estimating species trees from a collection of gene trees, and has statistical guarantees under the multi-species coalescent model. MP-EST is fast on small datasets (with not too many species) but its running time grows quickly with the number of species. We presented two iterative divide-and-conquer techniques (DACTAL-boosting and SSG-boosting) to use with MP-EST, with the goal of enabling MP-EST to analyze datasets with large numbers of species more efficiently. We tested these techniques on a collection of simulated and biological datasets, and showed that boosted versions of MP-EST were fast and highly accurate using these divide-and-conquer methods. The improvement in accuracy obtained by using these boosting techniques is not explained by any failure in MP-EST to optimize maximum likelihood effectively, but rather suggests the possibility that an alternative optimization criterion – quartet support – may be a highly effective approach to estimating species trees under the multi-species coalescent model.

Chapter 10

Conclusions

The theory of evolution indicates that every organism on earth has evolved from the "Universal Common Ancestor" (also known as *last universal ancestor* (LUA)) [144]. This theory, coupled with the advancement in molecular sequencing technology, has revolutionized the research in evolutionary biology, and has presented the grand challenge of reconstructing the *Tree of Life*. Fundamental to this reconstruction is the ability to produce, within reasonable time constraints, accurate phylogenies for large datasets (in terms of the number of genes and number of taxa). This dissertation contributes to the problem of fast and accurate species tree estimation from genes sampled throughout the whole genome, considering the presence of gene tree discordance and other challenging scenarios that frequently arise in phylogenomic analyses.

This dissertation makes significant contributions towards phylogenomic reconstruction in the presence of gene duplication and loss, and incomplete lineage sorting – two of the most important reasons for gene tree discordance. We developed efficient algorithms for estimating species trees by minimizing gene duplication and loss. We developed mathematical models for MGD and

MGDL by introducing new concepts (subtree-bipartition and domination), and presented clique-based formulations for both MGD and MGDL. We showed that the MGD and MGDL species trees are defined by a maximum weight clique and a minimum weight clique, respectively, in vertex-weighted graphs that can be computed from the subtree-bipartitions of the input gene trees. We also presented efficient polynomial time dynamic programming algorithms to find these optimal cliques by using the special structure of the graphs. We extended these algorithms for unrooted gene trees as well by considering the fact that estimated gene trees are most often unrooted.

In phylogenetic analyses gene trees are very often incomplete, meaning that genes might not have any individual for some species. Incomplete gene trees can result from sampling error, or true biological gene loss. In this dissertation we address the challenge of incomplete gene trees in phylogenomic analyses. We conducted the first empirical study to investigate the performance of different species tree estimation methods in the presence of gene tree incompleteness. We showed that incompleteness significantly reduces the accuracy of the species trees. We mathematically formalized the *optimal completion problem*, that seeks to add the missing taxa (species) into the gene trees with respect to a species tree such that the distance (in terms of ILS) between the gene tree and the species tree is minimized. We developed an efficient algorithm for solving this problem. We formalized optimization problems in the context of species tree estimation from a set of incomplete gene trees under the multi-species coalescent model, and proposed algorithms for solving these problems.

We also presented different mathematical formulations of gene loss based on different reasons for incompleteness (taxon sampling and true biological gene loss). We proved that the standard calculations for duplications and losses exactly solve gene tree parsimony (GTP) problem when incompleteness results from taxon sampling. However, they can be incorrect when incompleteness results from true biological loss. We presented new theory for gene tree parsimony when the gene trees are incomplete due to gene birth and death (true biological loss).

This dissertation investigates the impact of gene tree estimation error, which is a major challenge in phylogenomic analyses. Using extensive simulation study, we identified that existing methods are susceptible to gene tree estimation error. We proposed the first meta-method, which we call naive binning, to address the problem of poorly estimated gene trees. We showed that naive binning can dramatically improve the accuracy of the summary methods. We also showed that this technique can also be used to scale computationally expensive methods like *BEAST [52]. Statistical binning, which is an improvement over naive binning, was used in avian phylogenomics project to resolve the evolutionary history of 48 birds [62]. Being motivated by the success of the binning technique, we developed an even more improved version called weighted statistical binning. Weighted statistical binning enables highly accurate genome-scale species tree estimation, and is also statistically consistent under the multi-species coalescent model.

Finally, considering the exploding amount of molecular data systematists are willing to analyze and the high computational requirements of the existing methods, we developed divide-and-conquer based meta-methods that can make the existing techniques scalable to large numbers of taxa. Our method improves the scalability of MP-EST [80]. This technique also improves the accuracy of species trees estimated by MP-EST.

We now outline some future research directions stemming from the work in this dissertation:

- We addressed two major sources of gene tree discordance: gene duplication and loss, and incomplete lineage sorting. We always consider a single source of discordance at a time. However, evolutionary process could be more complex and a particular gene can evolve under multiple biological processes (e.g., ILS, gene duplication and loss, horizontal gene transfer etc.). Mathematical models to formulate the species tree estimation methods assuming multiple sources of gene tree discordance would enable us to better understand the gene and species evolution.
- We developed efficient algorithms for MGD and MGDL under the model condition where gene trees can be rooted or unrooted, and have a single copy per species. Extending these algorithms so that they can handle non-binary and multi-copy (multiple copies per species that can result from gene duplications) gene trees would be an important contribution.

- Throughout this dissertation, we assume that the underlying evolutionary history is treelike. However, for many organisms, a significant level of genetic exchange occurs between lineages (horizontal gene transfer), and for some groups, lineages can combine to produce new independent lineages (recombination) [22, 30, 84, 138]. These biological processes (known as *reticulate evolution*) transform a tree into a network. Future studies on investigating combinations of treelike and network-like evolutionary histories, building complex evolutionary models to capture these two modes of evolutions, and developing efficient algorithms for estimating evolutionary history under these complex models would enable us better understand the evolution.

- We developed meta-methods based on divide-and-conquer techniques, that can make species tree estimation technique scalable to large number of taxa. We report on an experimental study that it significantly improves the accuracy of MP-EST. It would be worth investigating its impact on other leading methods like ASTRAL [93, 94]. Also, future studies on reanalyzing large real biological datasets like the avian datasets [62] using this meta-method would be an important contribution.

- One of the major contributions of this dissertation is the observation that many species tree estimation methods are vulnerable to poorly estimated gene trees. We developed a novel technique called binning to address this problem. We presented weighted statistical binning, which improves phylogenomic analyses and is also statistically consistent, meaning that it can estimate the true species tree with arbitrarily high probability, given a sufficiently large number of genes and sufficiently large numbers of sites per gene. However, this assumption of having a large number of genes with unbounded numbers of sites per gene is not realistic, and we do not even need to use binning under these assumptions since a statistically consistent species tree method can reconstruct the true species tree with arbitrarily high probability, given a sufficiently large number of genes and unbounded numbers of sites per gene. Under realistic conditions with a limited number of genes and limited numbers of sites per gene, binning may result in model violation by putting genes with discordant evolutionary histories into a single bin, and thus the supergene tree distribution resulting from the binning approach may deviate from the true gene tree distribution. On the other hand, existing (unbinned) summary methods perform very poorly on poorly estimated gene trees, and therefore low signal genes are typically discarded from summary method analyses which distorts the true gene tree distribution as well. Thus both unbinned and binned analyses may be problematic with limited numbers of low signal genes (although binning is useful in reducing the problem of low signal genes, and empirically shown to be better than unbinned analyses in many cases). Therefore, in addition to improving the meta-methods like binning to handle poorly estimated gene trees, it is very important to develop new highly accurate summary methods that can handle gene tree estimation error to a certain extent.

Overall, this dissertation not only made a series of significant contributions in phylogenomic analyses, it also suggests additional avenues of important future works. We wish that one day, the present ambitious goal of constructing the *Tree of Life* would come into reality; and perhaps, that grand contribution might be traced back to the contribution of this thesis – just like a phylogeny.

Appendices

Appendix A

Supplementary Materials for Naive Binning

These supplementary materials present additional details about the methods used (Section A.1) and results obtained (Section A.2), and also present additional discussion (Section 7.4).

A.1 Methods

A.1.1 Overview

We used previously generated datasets from two studies (17-taxon datasets from [155] and 11-taxon datasets from [19]), and evaluated several pipelines for estimating species trees and gene trees for these datasets. We included three ways of estimating gene trees: RAxML and FastTree-2 to estimate maximum likelihood trees from the sequence alignments, and *BEAST to co-estimate gene trees and species trees. We explored several ways of estimating species trees: BUCKy, *BEAST, MRP, Greedy Consensus, Phylonet-MDC, MP-EST, and CA-ML. Each analysis produced a set of estimated gene trees and species trees, which we could evaluate for accuracy by comparing them to the model gene and species trees. We noted the missing branch rate (false negative, or FN error) and running time usage for each method. We compared the methods and determined which results were statistically significant using Wilcoxon signed rank T-test, with $\alpha = 0.05$.

We used 11-taxon datasets with 100 genes (100 replicates) and 17-taxon datasets with up to 32 genes (also with 100 replicates). The 11-taxon datasets were generated by model conditions that violate the molecular clock and came in two forms: datasets that were generated under a high level of ILS (called "strongILS") and datasets that were generated under a low level of ILS (called "weakILS"). The 17-taxon datasets were generated under the molecular clock and had a high level of ILS; these came in two forms: 8-gene and 32-gene datasets.

Slow methods: Pipelines that included *BEAST or BUCKy were too computationally intensive to run on all the replicates; we therefore only explored these methods on a subset of the replicates. Specifically, we never ran *BEAST on 100 replicates of any model condition. Instead, we ran *BEAST (binned and unbinned) on 20 replicates of the 11-taxon datasets with at most 50 genes, and 20 replicates of the 17-taxon datasets with 8 genes and with 32 genes. For BUCKy, we were able to run it on 20 replicates (unbinned) of all model conditions tested. In addition, when we ran BUCKy with binning, we were able to run it on 100 replicates of the 11-taxon strongILS datasets and 100 replicates of the 17-taxon 32-gene datasets. The remaining methods were all fast enough for us to run on all 100 replicates of all model conditions.

Standard error: The error bars in the figures correspond to the standard error, given by $S/\sqrt{n}$, where S is the standard deviation and n is the number of datapoints.

A.1.2 Datasets

All datasets are available online at
http://www.cs.utexas.edu/users/phylo/datasets/ILS/.

11-taxon datasets: The 11-taxon datasets were created for the study in [19], and simulated under a complex process to ensure substantial heterogeneity between genes and to deviate from the molecular clock. There were two types of model trees – ones with long branches (LB) that produce low levels of ILS, and ones with short branches (SB) that produce high levels of ILS. We have referred to these two different model conditions as weakILS and strongILS, respectively. Here we present the text from the paper, modified only to remove the references to other papers and figures.

Text from [19]:

"We generated DNA alignments from 5-taxon and 11-taxon species trees. An asymmetric tree topology was chosen on 5 taxa, as this was proven to be more difficult to reconstruct in the presence of gene-to-gene discordance [69]. Our 11-taxon tree contains two copies of our 5-taxon tree (subtree with taxa 1, 2, 3, 4 and subtree with taxa 5, 7, 9, 10, both with taxon 11 as an

outgroup). In one of the two copies, taxa 6 and 8 were added in order to detect potential effects of the number of taxa on the estimation of internal edges CFs. For each species tree topology, two sets of branch lengths were considered. One set had long internal branches (LB), whereas the other set had some short internal branches (SB). Species tree branch lengths were measured in coalescent units, as obtained by dividing the number of generations by the effective population size. Under the coalescent model, branch lengths in coalescent units determine the proportion of genes that share the species tree topology and the proportion of genes that have any given conflicting topology.

"In order to simulate multilocus data sets, 10, 50, or 100 unlinked gene trees were generated along the species trees. We used an effective size of 50,000 haploid individuals in each population. The numbers of generations between speciations were determined by multiplying branch lengths in coalescent units by the population size.

"HGTsimul was used to simulate a Poisson-distributed number of genomic rate change events (with a mean of three changes) on the species tree, for genomic departure from the molecular clock. Lineage-specific rates were simulated from a gamma distribution with mean 1 and shape parameter 2.0. For each gene, branch lengths obtained from Serial SimCoal were multiplied by these lineage-specific rates, then further multiplied by a common factor to obtain a randomly chosen gene diameter (uniform in 0.024 and 0.037 substitutions per site). Next, gene tree branch lengths were modified in a gene-specific manner: for each individual gene, a Poisson-distributed number of rate change

events (three changes on average) were placed on the gene tree, whose branch lengths were multiplied by a gamma-distributed rate (mean 1 and shape parameter 2.0) in between these gene-specific rate change events. Finally, sequences were simulated using the JukesCantor (JC) model and no site-specific rate variation, for computational feasibility.

"In summary, our simulations included important factors that contribute to heterogeneity among genes, such as heterogeneity in the overall rate of evolution, departure from clock-like evolution, and topological discordance."

17-taxon datasets: We used 17-taxon datasets that were simulated for [155], and provided to us by the authors. In this simulation, species trees were generated using the Yule module using Mesquite ([85]), and with total branch length of 800,000 generations, not counting the outgroup. Two collections of gene trees were simulated in this model: one with only 8 gene trees and one with 32 gene trees; however, the 8-gene dataset is not a subset of the 32-gene dataset. These gene trees were simulated within the species trees using the "Coalescence Contained Within Current Tree" module within Mesquite, with an effective population size of $N_e = 100,000$. Then sequences were evolved down the gene trees under the Jukes-Cantor model (without any rates-across-sites), using Seq-gen ([116]), with each sequence having length 2000.

Thus, these sequences evolve under a strong molecular clock, and there is no rate variation across sites or between different genes.

Subsampling: Our 11-taxon datasets (both strongILS and weakILS) contain 100 replicates each containing 100 genes. To evaluate the impact of the number of genes on the performance of different methods, we subsample different number of genes (5, 10, 25, and 50 genes) from our available set of 100 genes. We randomly subsample a particular number of genes (5, 10, 25 etc.) from a replicate that contains 100 genes. We generated 20 set of such subsamples from each replicate. For experiments analyzing 11-taxon datasets with up to 50 genes, we generated either 20 replicates (all from one replicate alignment) or 100 replicates (from 5 different replicate alignments). For experiments analyzing 11-taxon datasets with 100 genes, we used all 100 replicates. The 17-taxon datasets came in two collections - one with 8 genes, and one with 32 genes. Therefore, for the analyses with 17-taxon datasets, we used 20 or 100 replicates of the datasets in each collection.

A.1.3 Methods

A.1.3.1 Gene tree estimation

We used three methods for estimating gene trees: FastTree-2, RAxML, and *BEAST.

- FastTree-2 (v. 2.1.3 SSE3) ([114]). We used FastTree-2 to estimate ML gene trees from the sequence alignments, using the following command:

  ```
  FastTree -gtr -nt <sequenceAlignment>
  > <outputFile>
  ```

- RAxML: We ran RAxML v. 7.3.1 ([130]) to estimate ML gene trees from sequence alignments. We ran 20 runs of RAxML on each of the alignments, using the following command:

```
raxmlHPC-PTHREADS -T 2 -m GTRGAMMA
-s <sequenceAlignment> -n <output-name>
-N 20 -p 1234.
```

For estimating bootstrap branch support for the RAxML-estimated trees, we generated 400 bootstrap trees per each gene and then drew branch support on the edges of the ML tree by using these 400 bootstrap trees. The proportion of the bootstrap trees in which a particular split is found is taken to be the degree of support for that split. We then produced a 75%-branch support version of each estimated gene tree by contracting all edges with support below 75%.

- *BEAST: We ran *BEAST in its default setting to co-estimate gene trees and species trees; details are provided below under "Species Tree Estimation".

A.1.3.2 Species tree estimation

***BEAST:** We used *BEAST v. 1.6.2 [33] in default mode to co-estimate the gene trees and species tree on every dataset. For a given *BEAST analysis, we discarded the first 10% of the trees returned by the analysis, and then sampled one (1) out of each 1000 of the remaining trees. We return the maximum credibility species tree and gene trees from the *BEAST output. On the 11-taxon datasets with 5, 10, 25, and 50 genes, we ran *BEAST for 80M, 120M, 160M, and 200M MCMC iterations, respectively. We did not run *BEAST to convergence on the 100 gene datasets. On the 17-taxon datasets, we ran *BEAST for 200M MCMC iterations.

We were able to run *BEAST on 11-taxon datasets with up to 50 genes. We observed very high ESS values (all the ESS values were greater than 100, and many of them were in the thousands) except for 5 and 10-gene cases, where some ESS values were less than 100. On 17-taxon 8 and 32-gene datasets, we observed very high ESS values (all the ESS values are greater than 100, and many of them were in the thousands) when we ran it for 200M iterations. When used with binning on 11- and 17-taxon datasets, we ran 50M iterations on the supergenes and observed very high ESS values.

We ran *BEAST on 11-taxon 100-gene datasets with 50M iterations; each of these analyses took around 100 hours per replicate dataset, but produced very poor ESS values (we observed many parameters having less than 100 ESS). Therefore, we did not report results for *BEAST on the 11-taxon 100-gene datasets.

BUCKy: We used BUCKy v.1.4.0 [1, 73] in default mode; thus, $\alpha = 1$. As noted in Chapter 7, most of the experiments involving BUCKy were run with input gene tree distributions computed using RAxML. However, we also used *BEAST in Experiment 4. We used the following command:

```
bucky -n <numberOfGenerations> -o <outputFileRoot> <inputFiles>
```

For the analyses with distributions produced by *BEAST, we ran 80M, 120M, 160M, 200M iterations of *BEAST for 5, 10, 25, and 50 genes, respectively, and we sampled one tree out of each 1000 iterations; this produced 80K, 120K, 160K, and 200K trees in each distribution for datasets with 5, 10, 25, and 50 genes, respectively. We discarded the first 10% of these trees as *burn-in*, and used the remaining trees as the input to BUCKy. We ran BUCKy with 30M generations for 5- and 10-gene cases, 40M generations on 25-gene cases, and 50M generations for 50-gene cases. For 17-taxon datasets, we ran 40M generations. Note therefore that we did not test BUCKy on gene tree distributions estimated by *BEAST on the 100-gene datasets, because *BEAST was too expensive to run on these datasets.

We also ran BUCKy on RAxML-bootstrap trees, using 400 bootstrap trees per gene. We ran 500M generations of BUCKy for 5-, 10-, and 25-gene cases, and 200M generations for 50-gene cases. When run with binning, we ran 500M and 50M generations of BUCKy on 11-taxon strongILS and weakILS datasets, respectively. On 17-taxon datasets (both binned and unbinned), we ran 100M generations of BUCKy.

As with *BEAST, there is no strict condition for convergence of BUCKy; however, an "Average SD of mean sample-wide CF" below 0.05 may be adequate to have high confidence about the convergence. Samples of the standard deviation (SD) for the CF statistics for different BUCKy analyses follow:

- 11-taxon 50-gt, RAxML trees: SD = 0.000 to $\sim$ 0.004

- 11-taxon 50-gt, *BEAST trees: SD = 0.000

- 11-taxon 25-gt, RAxML trees: SD = 0.001 to $\sim$ 0.006

- 11-taxon 25-gt, *BEAST trees: SD = 0.000

- 11-taxon 10-gt, RAxML trees: SD = 0.000 to $\sim$ 0.007

- 11-taxon 10-gt, *BEAST trees: SD = 0.000

- 11-taxon 5-gt, RAxML trees: SD = 0.000 to $\sim$ 0.001

- 11-taxon 5-gt, *BEAST trees: SD = 0.000

- 17-taxon 32-gt, RAxML trees: SD = 0.000

- 11-taxon 32-gt, *BEAST trees: SD = 0.000 to $\sim$ 0.003

- 17-taxon 8-gt, RAxML trees: SD = 0.000

- 11-taxon 8-gt, *BEAST trees: SD = 0.000

The following statistics are for the binned analyses:

- 11-taxon 50-gt (10 bins): SD = 0.000

- 11-taxon 25-gt (5 bins): SD = 0.000

- 17-taxon 32-gt (8 bins): SD = 0.000

BUCKy returns two trees: one is the population tree (referred to as "BUCKy-pop") and the other is the concordance tree (referred to as "BUCKy-con"). BUCKy-pop is statistically consistent in the presence of ILS, but BUCKy-con is not.

MP-EST: We used MP-EST v. 1.2 [80] to estimate the species tree from input gene trees. MP-EST requires rooted gene trees as input; our datasets all include outgroups, and we root the estimated gene trees using these outgroups. MP-EST is statistically consistent in the presence of ILS, and maximizes a pseudo-likelihood function in order to estimate the species tree. We ran it in its default setting with MAXROUND=1000000.

Matrix Representation with Parsimony (MRP): MRP [115] is a supertree method that we use as a consensus method (since all the gene trees have the same set of taxa). MRP has two steps: in the first step, it encodes each input source tree as a matrix over {0,1, ?}, with one row for each taxon in the full set of taxa, and with each character corresponding to one edge bipartition in one source tree. These matrices are then concatenated together to obtain a single matrix. The MRP supertree is obtained by analyzing the character matrix using a maximum parsimony approach.

We created MRP matrices using a custom Java program, and solved MRP heuristically using the default approach implemented in PAUP* (v. 4. 0b10) [137]. By default, PAUP* generates an initial tree through random sequence addition (adding sequences one at a time in the most parsimonious position in a tree) and then performs Tree Bisection and Reconnection (TBR) moves until it reaches a local optimum. This process is repeated 1000 times, and at the end the most parsimonious tree is returned. When multiple trees are found with the same maximum parsimony score, the "extended majority consensus" of those trees is returned.

Below is the PAUP* block:

```
begin paup;
set criterion=parsimony maxtrees=1000
increase=no;
hsearch start=stepwise addseq=random
nreps=100 swap=tbr;
filter best=yes;
savetrees file = <treeFile> replace=yes
format=altnex;
contree all/ strict=yes
treefile = <strictConsensusTreeFile>
replace=yes;
tcontree all/ majrule=yes strict=no
treefile = <majorityConsensusTreeFile>
replace=yes;
contree all/ majrule=yes strict=no
le50=yes
```

```
treefile = <greedyConsensusTreeFile>
replace=yes;
log stop;
quit; end;
```

Phylonet: We use the Phylonet v. 2.4 [143] to solve MDC heuristically or exactly, depending on the dataset size. For the 11-taxon datasets, we use the version that is guaranteed to solve MDC optimally, and for the 17-taxon datasets we use the heuristic version. The input to Phylonet in each case is a set of gene trees restricted to the branches with bootstrap support at least 75% (i.e., with all low-support branches contracted). The version of Phylonet we used on these partially resolved gene tree estimates solves the following problem: Given a set of (partially resolved) unrooted gene trees $\mathcal{T} = \{t_1, t_2, \ldots, t_k\}$ (not necessarily on the same set of taxa), find binary refinements t_i^* for each t_i, and species tree T, so that the MDC score of T with respect to $\mathcal{T}^* = \{t_1^*, t_2^*, \ldots, t_k^*\}$ is minimum among all such sets $\mathcal{T}^*$ and species trees T. Thus, Phylonet solves a constrained version of MDC, taking bootstrap support into consideration. See [155] for more details and the proof of correctness. See also [7] for the proof that Phylonet handles missing taxa correctly.

Greedy Consensus: We ran the greedy consensus technique (also called the extended majority consensus) using PAUP* v. 4.0b10. The greedy consensus begins by computing the majority consensus (the tree whose edge-induced taxon bipartitions are those that appear in more than half of the input trees), and then adds compatible bipartitions, one at a time, in an order reflecting the frequency with which each bipartition appears.

Below is the PAUP* block:

```
begin paup;
set autoclose = yes warntree = no
warnreset = no notifybeep = no
monitor = yes taxlabels = full;
set criterion = parsimony;
set increase = auto;
gettrees file = <nexusFile> allblocks = yes
warntree = no unrooted = yes;
contree all / strict = no
majrule = yes le50 = yes
treefile = <greedyConsensusTreeFile>;
end;
```

Combined Analyses using Maximum Likelihood (CA-ML): This method concatenates the alignments on all genes into one super-alignment, and then estimates a tree from the super-alignment using maximum likelihood, treating the alignment as unpartitioned. We used RAxML for this analysis, using the following command:

```
    raxmlHPC-PTHREADS -T 2 -m GTRGAMMA
-s <sequence> -n <output-name> -N 10
-p 1234.
```

A.1.4 Running time

***BEAST running time:** We tested three 11-taxon datasets with 100 genes without using binning and using 50M iterations; these analyses ranged from 80 to 150 hours. Based on the ESS values, none of these came close to convergence; hence, the running times here are suggestive of lower bounds for time needed to use *BEAST. However, these datasets were run on Condor, and so running times are approximate.

The remaining analyses were on at most 50 genes, or used binning to analyze 100 genes (and so had only 20 supergenes). Each analysis is of one dataset only, and was done on a dedicated 64-bit machine with 32173 MB memory.

- Unbinned analyses
  - 11-taxon strongILS 50-gt, 200M iterations: 57 hours
  - 11-taxon strongILS 25-gt 160M iterations: 20 hours
  - 17-taxon 32-gt, 200M iterations: 35 hours
- Binned analyses (5 genes per bin)

  - 11-taxon strongILS 100-gt with 20 bins with 50M iterations: 10 hours using 4 threads
  - 11-taxon strongILS 50-gt with 10 bins (5 genes in each bin), 50M iterations: 6.4 hours
  - 11-taxon 25-gt strongILS with 5 bins, 50M iterations: 3.1 hours
  - 17-taxon 32-gt with 8 bins, 50M: 5.6 hours

BUCKy running time: We performed several BUCKy analyses for all three model conditions. These analyses showed that the running time was determined by the type of input distribution, and whether it was from one of the two 11-taxon model conditions or from the 17-taxon model condition; however, 11-taxon strongILS and 11-taxon weakILS analyses took the same amount of time.

Results on unbinned analyses with RAxML gene tree distributions:

- 11-taxon 100-gt, RAxML trees, 200M generations: 2.2 hours
- 11-taxon 50-gt, RAxML trees, 200M generations: 2.1 hours
- 11-taxon 25-gt, RAxML trees, 500M generations: 3.5 hours
- 11-taxon 10-gt, RAxML trees, 500M generations: 2.36 hours
- 11-taxon 5-gt, RAxML trees, 500M generations: 1.75 hours
- 17-taxon 8-gt, RAxML trees, 100M generations: 40 mins

- 17-taxon 32-gt, RAxML trees, 100M generations: 2.07 hours

Results on unbinned analyses with *BEAST gene tree distributions:

- 11-taxon 50-gt, *BEAST trees, 50M generations: 21 mins
- 11-taxon 25-gt, *BEAST trees, 40M generations: 11 mins
- 11-taxon 10-gt, *BEAST trees, 30M generations: 7 mins
- 11-taxon 5-gt, *BEAST trees, 30M generations: 3 mins
- 17-taxon 32-gt, *BEAST trees, 40M generations: 15 mins
- 17-taxon 8-gt, *BEAST trees, 40M generations: 6 mins

Note the difference in running time between *BEAST and RAxML distributions, indicating that BUCKy converges with fewer MCMC iterations when run with *BEAST distributions than when run with RAxML bootstrap distributions! However, *BEAST takes much more time to run, so the total running time when based on *BEAST is much longer.

Running time for binned analyses:

- 11-taxon 25-gt (5 bins), RAxML trees, 500M generations: 1.1 hours
- 11-taxon 50-gt (10 bins), RAxML trees, 500M generations: 1.75 hours
- 17-taxon 32-gt (8 bins), RAxML trees, 100M generations: 13 mins

RAxML bootstrapping: We generated 400 bootstrap replicates per gene; each analysis took under 2 minutes on each gene sequence alignment, whether it was a single gene or a supergene. Specific results are:

- 11-taxon dataset strongILS and weakILS: less than 1 minute per gene
- 17-taxon dataset: less than 2 minutes per gene
- 11-taxon 50-gt, 10 bins (5 genes in each): less than 2 minutes per supergene
- 11-taxon 25-gt, 5 bins (5 genes in each): less than 2 minutes per supergene
- 11-taxon 100-gt, 20 bins (5 genes in each): less than 2 minutes per supergene

A.2 Additional results

A.2.1 Experiment 1: Evaluating fast species tree estimation methods on 100 replicate datasets

CA-ML showed substantial improvements over the next best method (typically MP-EST, but in one case MRP) in Experiment 1 for the 11-taxon datasets, with biggest improvements on the 11-taxon weakILS datasets. CA-ML was also more accurate than the next best method on the 17-taxon datasets, but the differences were smaller. As can be seen, the improvements were statistically significant for all conditions, with $p < 0.003$ on the 11-taxon datasets (both strongILS and weakILS), and $p \leq 0.043$ on the 17-taxon datasets.

- 11-taxon strongILS 5-gt: (CA-ML vs. MRP): $p < 10^{-6}$
- 11-taxon strongILS 10-gt: (CA-ML vs. MP-EST): $p < 10^{-3}$
- 11-taxon strongILS 25-gt: (CA-ML vs. MP-EST): $p = 10^{-6}$
- 11-taxon strongILS 50-gt: (CA-ML vs. MP-EST): $p < 10^{-5}$
- 11-taxon strongILS 100-gt: (CA-ML vs. MP-EST): $p = 0.003$
- 17-taxon 8-gt: (CA-ML vs. MP-EST): $p = 0.013$
- 17-taxon 32-gt: (CA-ML vs. MP-EST): $p = 0.043$

Thus, the improvement of CA-ML over the next best method is statistically significant in all these cases.

A.2.2 Experiment 2: Evaluating species tree estimation methods on 20 replicate datasets

***BEAST vs. fast methods on RAxML gene trees:** We compared *BEAST to fast methods on RAxML gene trees on 20 replicates of all model conditions. With the exception of the 17-taxon 32-gene case, the differences were statistically significant. On 11-taxon strongILS datasets, *BEAST is significantly better than the fast methods ($p < 10^{-3}$). The difference is also significant on 17-taxon 8-gene datasets (p-values are within the range $0.02 \sim 0.03$). On 11-taxon weakILS datasets, *BEAST is significantly better than the fast methods on 5 and 10 genes ($p < 10^{-2}$), but not significantly better on 25 or 50 genes ($p > 0.1$).

CA-ML vs. *BEAST: As *BEAST is computationally intensive to run (tens to hundreds of hours for each analysis for some datasets), we compared CA-ML to *BEAST on only 20 replicate datasets of each model condition. The relative performance between the two methods was mixed, with CA-ML being more accurate in some cases and less accurate in others. However, the only statistically significant differences were for two conditions: 11-taxon 25-gene strongILS and 11-taxon 5-gene weakILS, in which CA-ML was more accurate than *BEAST ($p = 0.05$ and $p = 0.03$, respectively).

BUCKY-con vs. BUCKy-pop: The difference is statistically significant only on the 11-taxon strongILS 25-gene ($p = 0.003$) and 50-gene ($p = 0.035$) cases.

A.2.3 Experiment 3: Evaluating gene tree estimation error

Here we discuss the accuracy of gene trees estimated by maximum likelihood (by RAxML or FastTree-2) and *BEAST. Results for the 11-taxon strongILS conditions are provided in Figure A.1 and Table A.1; results for the 11-taxon weakILS conditions are provided in Figure A.2 and Table A.2. In Table A.3 we present results for the 17-taxon datasets; the figure for these data are in the main document. Note that *BEAST gives a dramatic improvement in gene tree estimation accuracy, and that the smallest improvement is on the 17-taxon datasets. However, even on these data, the improvement is at least 50%.

Method	Error 5 genes	Error 10 genes	Error 25 genes	Error 50 genes	Error 100 genes
*BEAST	0.224	0.162	0.155	0.141	-
FastTree	0.430	0.440	0.407	0.418	0.424
RAxML	0.405	0.424	0.401	0.399	0.413

Table A.1: **Average missing branch rates (over 20 replicates) of gene trees estimated by different methods on 11-taxon strongILS datasets**. *BEAST could not be run on 100-gene datasets. Experiment 3.

Method	Error 5 genes	Error 10 genes	Error 25 genes	Error 50 genes
*BEAST	0.095	0.039	0.033	0.033
FastTree	0.314	0.299	0.338	0.334
RAxML	0.311	0.283	0.321	0.319

Table A.2: **Average missing branch rates (over 20 replicates) of gene trees estimated by different methods on 11-taxon weakILS datasets**. Experiment 3.

Method	Error 8 genes	Error 32 genes
*BEAST	0.195	0.176
FastTree	0.399	0.400
RAxML	0.393	0.389

Table A.3: **Average missing branch rates over 20 replicates of gene trees estimated by different methods on 17-taxon datasets**. Experiment 3.

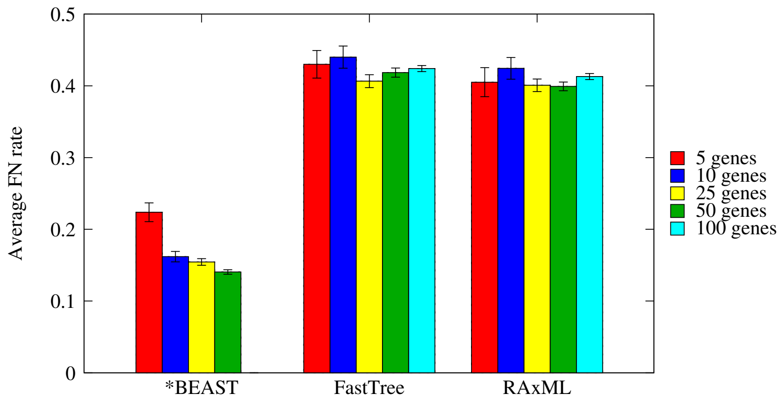


Figure A.1: **Gene tree estimation error rates on 11-taxon strongILS datasets.** Average and standard error bars (over 20 replicates) of *BEAST, RAxML, and FastTree-2. Experiment 3.

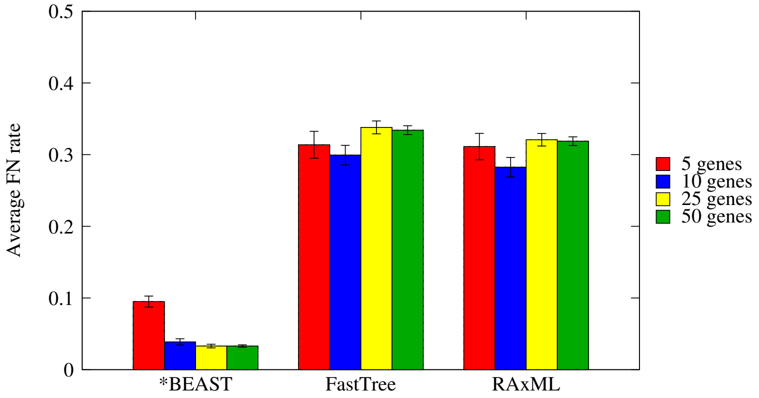


Figure A.2: **Gene tree estimation error rates on 11-taxon weakILS datasets.** Average and standard error bars (over 20 replicates) of *BEAST, RAxML, and FastTree-2. Experiment 3.

A.2.4 Experiment 4: Evaluating summary methods on gene trees estimated by *BEAST

The figures below show results of using summary methods on gene trees estimated using *BEAST, and compares them to the species trees estimated by *BEAST. There were no statistically significant differences in the accuracy of trees estimated using *BEAST as compared to using summary methods on gene trees estimated using *BEAST ($p > 0.2$ for all pairwise comparisons).

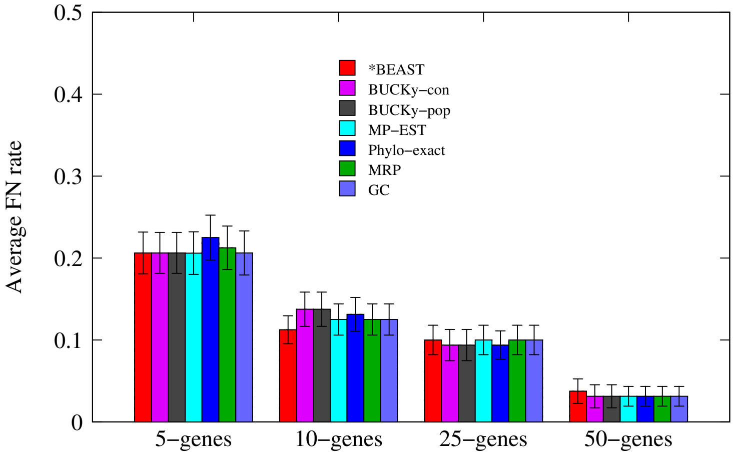


Figure A.3: **Results for summary methods on gene trees estimated using *BEAST on 11-taxon weakILS model conditions with up to 50 genes**. Results are shown for 20 replicates. Experiment 4.

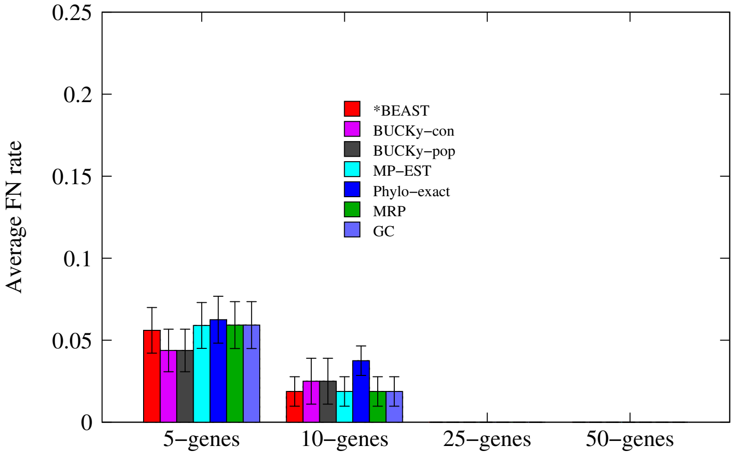


Figure A.4: **Results for summary methods with input gene tree distributions estimated using *BEAST on 11-taxon weakILS model conditions with up to 50 genes**. Results are shown for 20 replicates. Every method returns the true tree on the 25- and 50-gene datasets. Experiment 4.

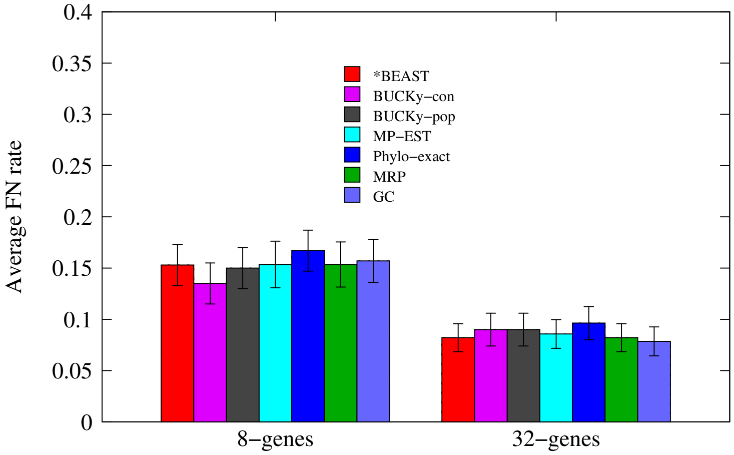


Figure A.5: **Results for methods with input gene tree distributions estimated using *BEAST on 17-taxon model conditions**. Results are shown for 20 replicates. Experiment 4.

A.2.5 Experiment 5: Evaluating the impact of naive binning on fast methods - 100 replicate datasets

We divide Experiment 5 into two parts: a comparison on 100 replicate datasets of the fast methods (all methods other than *BEAST and BUCKy), and then a comparison on 20 replicate datasets of all methods. See this subsection for results on fast methods, and the next subsection for results on all methods. Note that the impact of binning on the fast methods is best evaluated in the experiments on 100 replicate datasets, rather than on the 20 replicate datasets, especially in terms of statistical significance.

Because CA-ML is an unpartitioned analysis, it is not impacted by binning. Binning can impact all the other methods, but we do not have results for the unbinned Bayesian methods (*BEAST and BUCKy) on these 100 replicate datasets because they are too computationally expensive.

These experiments show the following trends:

- MP-EST, MRP, Phylonet, and Greedy Consensus each improved for all numbers of genes on the 11-taxon strongILS condition and on the 25-gene 11-taxon weakILS condition. The improvements on the 11-taxon weakILS conditions with 25 genes were small (at most 0.5%), but this is because all unbinned methods were highly accurate to begin with – all had error between 0.4% and 1.4%. The improvements on the 11-taxon strongILS conditions ranged from 1% to 4.8% (Phylonet on 50 genes), but differences were generally less on the 100-gene case (ranging from

0.6% to 3%) and 25-gene case (ranging from 1.1% for Greedy to 3% for Phylonet) than on the 50-gene case (ranging from 1.6% for MP-EST to 4.2% for Greedy).

- Phylonet became 0.5% more accurate on the 17-taxon condition, but the change was not statistically significant ($p > 0.25$). All other methods (MP-EST, Greedy, and MRP) became less accurate on the 17-taxon conditions, but the difference in accuracy was small (at most 1%) and the changes were not statistically significant for any of these methods.

- On the 11-taxon models, the differences for Phylonet's performance were statistically significant for every case, and tended to be larger than for the other methods. They were statistically significant for Greedy Consensus only on the 11-taxon strongILS datasets with 50 and 100 genes (and hence not for 25 genes on either strongILS or weakILS). The results were statistically significant for MP-EST on the 25-gene datasets (both strongILS and weakILS), but not for the other cases. Finally, the results were statistically significant for MRP only on the 50-gene strongILS datasets.

Thus, methods differed in their response to binning, and binning on the 11-taxon datasets generally improved accuracy and sometimes substantially, while generally reducing accuracy (but only slightly) on the 17-taxon datasets. However, the only statistically significant differences were improvements in accuracy. Phylonet in particular benefited from binning, improving even on the 17-taxon datasets, and improvement was greatest in cases where there were enough genes (at least 50), and accuracy before binning was not too great.

Method	Error 25 genes	Error 50 genes	Error 100 genes
CA-ML	0.053	0.031	0.018
BUCKy-con (binned)	0.070	0.045	0.034
BUCKy-pop (binned)	0.070	0.045	0.034
MP-EST	0.110	0.073	0.039
MP-EST (binned)	0.088	0.057	0.033
Phylonet-exact	0.126	0.089	0.054
Phylonet-exact (binned)	0.096	0.041	0.024
MRP	0.115	0.091	0.050
MRP (binned)	0.105	0.053	0.038
GC	0.114	0.096	0.054
GC (binned)	0.103	0.054	0.034

Table A.4: **Average missing branch rates for methods (unbinned and binned) on 11-taxon strongILS 25, 50 and 100-gene cases**. Results are shown for 100 replicates. Each bin contains 5 genes. BUCKy (unbinned) was not run on 100 replicates. Experiment 5.

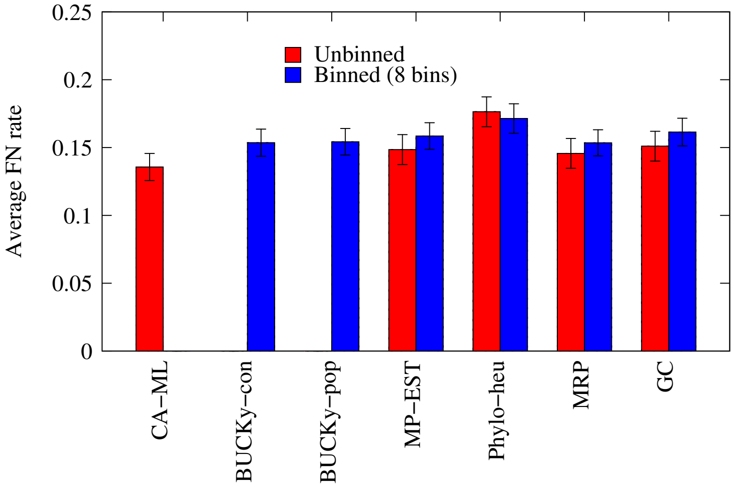


Figure A.6: **Results of binning experiment on 17-taxon datasets with 32 genes.** We show the performance (average and standard error bars) of methods other than BUCKy on unbinned genes and *BEAST. Each bin contains 4 genes; n=100 for all datapoints. Experiment 5.

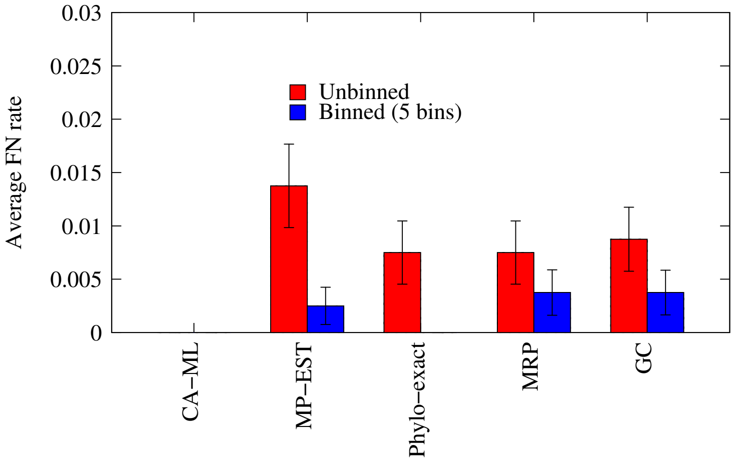


Figure A.7: **Results of the binning experiment on 11-taxon 25-gene weakILS datasets.** Each bin contains 5 genes. Average and standard error bars shown; n=100 for all datapoints. CA-ML returns the true tree on these data Experiment 5.

Method	p-values 25 genes	p-values 50 genes	p-values 100 genes
MP-EST	0.021	0.057	0.211
Phylonet	0.002	$< 10^{-5}$	$< 10^{-3}$
MRP	0.177	$< 10^{-4}$	0.079
GC	0.156	$< 10^{-4}$	0.007

Table A.5: **Evaluating the statistical significance of using binning on fast methods, when analyzing 100 replicate 11-taxon strongILS datasets**. We show *p*-values for the statistical significance of a difference between binned and unbinned analyses. Each bin has 5 genes. Experiment 5.

Method	Error
CA-ML	0.000
MP-EST	0.014
MP-EST (binned)	0.003
Phylonet	0.008
Phylonet (binned)	0.000
MRP	0.008
MRP (binned)	0.004
GC	0.009
GC (binned)	0.004

Table A.6: **Average FN rates for methods (unbinned and binned) on 11-taxon weakILS 25-gene case; n = 100**. Each bin contains 5 genes. We did not run *BEAST or BUCKy on 100 replicates. Experiment 5.

Method	p-values
MP-EST	0.002
Phylonet	0.016
MRP	0.188
GC	0.109

Table A.7: **Evaluating the impact of binning on fast methods on 100 replicate 11-taxon weakILS datasets with 25 genes**. We show p-values for the statistical significance of a difference between binned and unbinned analyses. Each bin has 5 genes. Experiment 5.

Method	Error
CA-ML	0.136
BUCKy-con (binned)	0.154
BUCKy-pop (binned)	0.154
MP-EST	0.149
MP-EST (binned)	0.159
Phylonet	0.176
Phylonet (binned)	0.171
MRP	0.146
MRP (binned)	0.153
GC	0.151
GC (binned)	0.161

Table A.8: **Average FN rates for methods (unbinned and binned) on 17-taxon 32-gene case; n = 100**. Each bin contains 4 genes. We did not run unbinned BUCKy on 100 replicates. Experiment 5.

Method	p-values
MP-EST	0.221
Phylonet	0.258
MRP	0.273
GC	0.245

Table A.9: **Evaluating the impact of binning for fast methods (binned vs. unbinned) on 100 replicates of 17-taxon 32-gene dataset**. We show *p*-values for the statistical significance of binned versus unbinned analyses. Each bin has 4 genes. Experiment 5.

A.2.6 Experiment 5: Evaluating the impact of naive binning on all methods - 20 replicate datasets

We now show results for naive binning on all methods (including BUCKy and *BEAST), but restricted to 20 replicate datasets. On these datasets, we were able to run the Bayesian methods (BUCKy and *BEAST), and so can explore the impact of binning on these methods. We do not show results for unbinned *BEAST on the 100-gene datasets, because these were too computationally intensive to run, but do show results obtained using *BEAST with binned datasets.

These results show the following trends:

- *BEAST has unchanged accuracy under all conditions where it can run in the unbinned and binned settings.
- On the 17-taxon datasets, no changes were statistically significant.
- BUCKy-con improved for the 11-taxon strongILS datasets (ranging from 3% on the 100-gene case to 7.5% on the 50-gene case) and by 2.5% on the 11-taxon weakILS 25-gene case. The changes were statistically significant for 25-genes and 50-genes, but not for 100-genes, on the strongILS datasets.
- With the exception of Phylonet (which was 100% accurate both with and without binning) all methods improved on the 11-taxon weakILS datasets as a result of binning, and the improvements ranged from 0.7%

(for MRP) to 3.1% (for BUCKy-pop). However, only BUCKy-pop had a statistically significant improvement ($p = 0.031$).

These results are similar to those observed on the 100-replicate case, except that with only 20 replicates, we do not detect statistically significant changes.

Method	Error 25 genes	Error 50 genes	Error 100 genes
CA-ML	0.062	0.025	0
*BEAST	0.100	0.038	-
*BEAST (binned)	0.100	0.038	0.012
BUCKy-con	0.143	0.125	0.056
BUCKy-con (binned)	0.094	0.050	0.025
BUCKy-pop	0.088	0.088	0.056
BUCKy-pop (binned)	0.094	0.050	0.025
MP-EST	0.156	0.163	0.044
MP-EST (binned)	0.106	0.056	0.031
Phylonet-exact	0.106	0.094	0.025
Phylonet-exact (binned)	0.077	0.069	0.018
MRP	0.143	0.163	0.056
MRP (binned)	0.138	0.056	0.043
GC	0.150	0.160	0.063
GC (binned)	0.125	0.056	0.044

Table A.10: **Average FN rates for methods (unbinned and binned) on 11-taxon strongILS 25, 50 and 100-gene cases; n = 20**. We do not show results for unbinned *BEAST on 100 genes, because it was not run to convergence. Each bin contains 5 genes. Experiment 5.

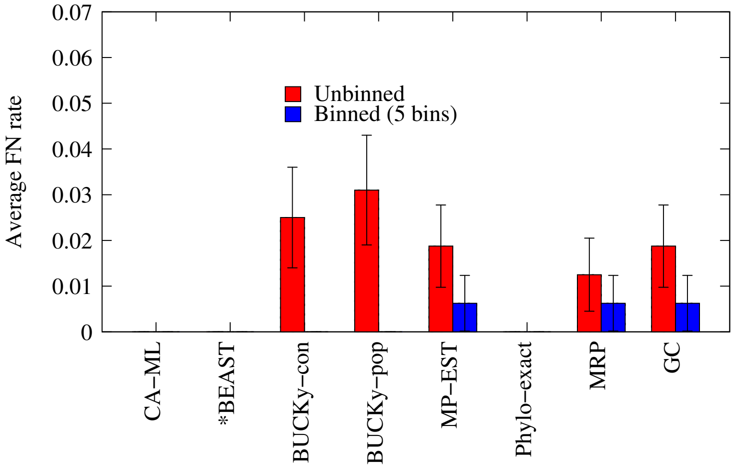


Figure A.8: **Results of the binning experiment evaluating all methods on 20 replicates of the 11-taxon 25-gene weakILS datasets.** Results are shown (average and standard error bars) for bins with 5 genes each. CA-ML, *BEAST (binned and unbinned), BUCKy-con (binned), BUCKy-pop (binned), and Phylonet-MDC (binned and unbinned) all return the true tree on these data.

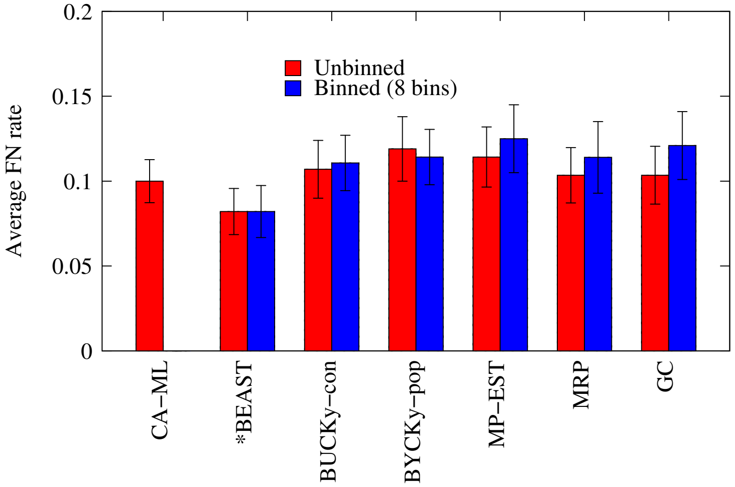


Figure A.9: **Results of binning experiment of 17-taxon datasets with 32 genes.** Average and standard error bars shown for all methods. Each bin has 4 genes; n=20 for all datapoints. No changes are statistically significant (p = 0.053 for MRP, p = 0.082 for GC, and $p > 0.2$ for all other methods). Experiment 5.

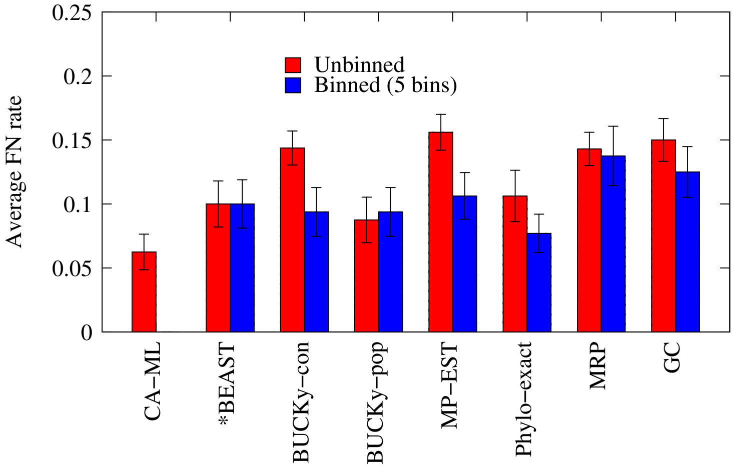


Figure A.10: **Results of the binning experiment on 11-taxon 25-gene strongILS datasets.** Each bin contains 5 genes. Average and standard error bars shown; n=20 for all datapoints. Experiment 5.

Method	p-values for 25 genes	p-values for 50 genes	p-values for 100 genes
*BEAST	0.500	0.500	-
BUCKy-con	0.018	0.005	0.089
BUCKy-pop	0.441	0.227	0.062
MP-EST	0.011	$< 10^{-4}$	0.363
Phylonet	0.113	0.179	0.500
MRP	0.307	$< 10^{-3}$	0.291
GC	0.230	$< 10^{-4}$	0.290

Table A.11: **Evaluating the impact of binning on all methods, applied to 20 replicates of the 11-taxon strongILS datasets**. We show *p*-values. We were not able to run *BEAST (unbinned) on 100-gene datasets. Experiment 5.

Method	Error
CA-ML	0.100
*BEAST	0.082
*BEAST (binned)	0.082
BUCKy-con	0.107
BUCKy-con (binned)	0.111
BUCKy-pop	0.119
BUCKy-pop (binned)	0.114
MP-EST	0.114
MP-EST (binned)	0.125
Phylonet	0.139
Phylonet (binned)	0.132
MRP	0.104
MRP (binned)	0.114
GC	0.104
GC (binned)	0.121

Table A.12: **Average FN rates for methods (unbinned and binned) on 17-taxon 32-gene case; n = 20**. Each bin contains 4 genes. Experiment 5.

Method	p-values
BUCKy-con	0.063
BUCKy-pop	0.031
MP-EST	0.250
Phylonet	0.500
MRP	0.500
GC	0.250

Table A.13: **Evaluating the impact of binning on species tree estimation methods on 20 replicates of the 11-taxon weakILS datasets with 25 genes**. We show *p*-values for methods (binned vs. unbinned methods). Each bin has 5 genes. Experiment 5.

Method	Error
CA-ML	0.000
*BEAST	0.000
*BEAST (binned)	0.000
BUCKY-con	0.025
BUCKy-con (binned)	0.000
BUCKy-pop	0.031
BUCKy-pop (binned)	0.000
MP-EST	0.019
MP-EST (binned)	0.006
Phylonet	0.000
Phylonet (binned)	0.000
MRP	0.013
MRP (binned)	0.006
GC	0.019
GC (binned)	0.006

Table A.14: **Average FN rates for methods (unbinned and binned) on 11-taxon weakILS 25-gene case; n = 20**. Each bin contains 5 genes. Experiment 5.

Method	p-values
*BEAST	0.500
BUCKy-con	0.444
BUCKy-pop	0.311
MP-EST	0.191
Phylonet	0.212
MRP	0.053
GC	0.082

Table A.15: p**-values for methods (binned vs. unbinned) on 20 replicates of 17-taxon 32-gene dataset**. Each bin has 4 genes. Experiment 5.

Appendix B

Supplementary Materials for Weighted Statistical Binning

These supplementary materials present additional details about the datasets and methods used, and the results obtained for weighted statistical binning (Chapter 8).

B.1 Evaluation

We explored the performance of MP-EST and ASTRAL with weighted and unweighted statistical binning, and also without binning. We also examine concatenation of the entire set of gene sequence alignments using an unpartitioned maximum likelihood analysis using RAxML. We explore performance on a collection of simulated and biological datasets originally studied in [90]. We applied MP-EST and ASTRAL to a set of RAxML gene trees computed on bootstrap replicates of each gene sequence alignment. With bootstrap ML gene trees for each gene, summary methods were applied with the site-only multi-locus bootstrapping (MLBS) procedure [126], implemented as follows. For each gene or supergene, 200 replicates of bootstrapping are performed using RAxML. Next, 200 replicates $(R_1, R_2, \ldots, R_{200})$ of input datasets to the summary methods are created such that R_i contains the i^{th} bootstrap tree across all genes/supergenes. The summary methods are then run on these 200 input replicates, and 200 species trees are estimated. Finally, the greedy consensus tree of these 200 estimated species tree is computed, and support values are drawn on the branches of the greedy consensus tree by counting the occurrences of each bipartition in the 200 species trees.

B.1.1 Triplet gene tree distribution error

MP-EST computes species trees using the estimated distribution on rooted triplet trees defined by its input of gene trees. We therefore evaluated the impact of binning on the estimated gene tree distribution, measuring the divergence between the triplet distribution of estimated gene trees and the triplet distribution of true gene trees. We represent the gene tree distribution by the frequency of each of the three possible alternative topologies for all the $\binom{n}{3}$ triplets of taxa, where n is the number of taxa. Therefore, we have $\binom{n}{3}$ true triplet distributions. Hence, for each triplet of taxa, we have estimated triplet distributions using the unbinned analysis, as well as weighted and unweighted binning analyses. We computed the Jensen-Shannon divergence of each of these $\binom{n}{3}$ triplet distributions and showed the empirical cumulative distribution of these divergences. The Jensen-Shanon divergence is a symmetrized and smoothed version of Kullback-Leibler divergence [70] between two distributions P and Q, and can be calculated as follows [39]:

$$JS(P, Q) = \frac{1}{2}KL(P, M) + \frac{1}{2}KL(Q, M) \tag{B.1}$$

where $M = \frac{P+Q}{2}$, and KL is the Kullback-Leibler divergence.

B.1.2 Species tree estimation error and branch support

We compared the estimated species trees to the model (i.e., true) species tree (for the simulated datasets) or to the scientific literature (for the biological datasets). We measure topological error using the missing branch rate (also known as the false negative (FN) rate), which is the proportion of branches in the true tree that are missing from the estimated tree. We also reported the error in species tree branch lengths estimated by MP-EST using the ratio of estimated branch length to true branch length for those branches of the true tree that appear in the estimated tree; thus, 1 indicates correct estimation, values above 1 indicate lengths that are too long, and values below 1 indicate branch lengths that are too short. Note that species tree branch lengths reflect the expected amount of ILS, and so under-estimation of species tree branch lengths means over-estimation of ILS, and over-estimation of branch lengths means under-estimation of ILS. We also computed the branch support of the false positive (FP) and true positive (TP) edges, where false positive edges are present in the estimated tree but not in the true tree, and edges that are present in both the estimated and true tree are true positive edges.

B.1.3 Simulated datasets

We studied four collections of simulated datasets: two based on biological datasets that were generated in a prior study [90], and two new collections with smaller numbers of species. We briefly describe the simulation protocol for the biological datasets, and direct the reader to [90] for full details.

Mammalian simulated datasets

This dataset was generated by [90], and studied there and also in [91]. Here we describe the procedure followed by [90] to generate these data. First, a species tree was computed for the full biological dataset in [129], using MP-EST (this was done before removing 23 erroneous genes), and the tree topology and branch lengths were used as the model tree. Thus, the mammalian simulation model tree has an ILS level based on an MP-EST analysis of the biological mammalian dataset. Gene trees were simulated within this species tree under the multi-species coalescent model, and then the branch lengths on the gene trees were defined using the gene trees estimated on the biological dataset.

Variants of the basic model condition were generated by varying the amount of ILS, the number of genes, and the sequence length for each gene; these modifications also impact the amount of gene tree estimation error and the average bootstrap support in the estimated gene trees, and so can be modified to produce datasets that resemble the biological data.

The amount of ILS was varied by adjusting the branch length (shorter branches increase ILS). A model condition with reduced ILS was created by uniformly doubling (2X) the branch lengths, and a model condition with higher ILS was generated by uniformly dividing the branch lengths by two (0.5X). The amount of ILS obtained without adjusting the branch lengths is referred to as "default ILS", and was estimated by MP-EST on the biological data.

The average bootstrap support (BS) in the biological data was 71%, and so [90] generated sequence lengths that produced estimated gene trees with bootstrap support bracketing that value – 500bp alignments produced estimated gene trees with 63% average BS and 1000bp alignments produced estimated gene trees with 79% BS. We also generated model conditions with very short sequence lengths (250bp), which have 43% average BS.

Here, we varied the number of genes from 50 to 800 to explore both smaller and larger numbers of genes than the biological dataset (which had roughly 400 genes). In total, we generated 17 different model conditions specified by the ILS level, the number of genes, and the sequence length. For each of these model conditions, [90] created 20 replicates.

Avian simulated datasets

Mirarab et al. [90] used the species tree estimated by MP-EST on a subset of the avian dataset with 48 species and 14,446 loci studied by [62], and simulated gene trees by varying different parameters (similar to the mammalian simulated datasets). Three types of genomic markers were studied in [62]: exons, UCEs, and introns. The average bootstrap support (BS) of the gene trees based on exons, UCEs, and introns, was 24%, 39% and 48%, respectively; the longest introns had the highest average BS (59%). Mirarab et al. varied sequence lengths (250bp, 500bp, 1000bp, and 1500bp) to produce four model conditions with patterns of average bootstrap support that resemble these four marker types. Mirarab et al. varied the number of genes from 200 to 2000, but here, we augmented the dataset to also look at fewer genes (50 and 100). Mirarab et al. varied the amount of ILS, using the same technique as was used in generating the mammalian simulated datasets.

15-taxon simulated datasets

We simulated a collection of 15-taxon datasets. The model species tree is a caterpillar-like ultrametric tree (i.e., the substitution process obeys a strict molecular clock) with 15 taxa; hence, it has two leaves x and y that are siblings in the tree. The lengths of all internal branches and the two branches incident with leaves x and y are all set to 0.005 substitutions per site; note that the assumption of ultrametricity defines the remaining branch lengths. The population size parameter ($\theta = 4N\mu$) is set to 0.05 for all branches, and this results in 12 short internal branches (0.1 in coalescence units) in succession. Ultrametric gene trees were simulated down this tree using McCoal[153] and commands given in Fig. B.1 Sequence data were simulated down each gene tree using bppseqgen [34] according to GTR+Γ parameters given in Fig. B.1. We built four model conditions (with ten replicates each) by trimming gene data to 100 or 1000 sites and by exploring 100 or 1000 genes.

10-taxon simulated datasets

We used simPhy [86] to simulate species trees using the Yule process with two different maximum tree length settings: 200K generations, resulting in short trees and high levels of ILS, and 1.8M generations, resulting in relatively longer trees and lower levels of ILS. We generated 20 species trees per model condition, and used simPhy to simulate 200 gene trees for each species trees using the multi-species coalescent process (simPhy parameters and commands are given in Fig. B.2). The gene trees (with branch lengths in substitution units) deviate from the strict molecular clock, and the rates of evolution vary across genes. We used Indelible to simulate GTR+Γ sequence evolution down these gene trees with 100 sites, with parameters given in Fig. B.2.

B.1.4 Biological datasets

We studied two biological datasets also studied in [90]: the avian dataset [62] containing 14,446 loci across 48 species, and a reduced version of the mammalian dataset studied by Song et al. [129] with 447 loci across 37 species, from which [90] deleted 23 erroneous genes and re-estimated gene trees using RAxML (see [90, 91] for discussion of these loci).

Methods and commands

Gene tree estimation: RAxML version 7.3.5 [130] was used to estimate gene trees under the GTRGAMMA model, using the following command:

```
raxmlHPC-SSE3 -m GTRGAMMA -s [input_alignment] -n [output_name]
-N 20
-p [random_seed_number]
```

The following command was used for bootstrapping:

```
raxmlHPC-SSE3 -m GTRGAMMA -s [input_alignment] -n [output_name]
-N 200
-p [random_seed_number] -b [random_seed_number]
```

Supergene tree estimation: For the biological datasets and the 10- and 15-taxon simulated datasets, we used a fully partitioned maximum likelihood analysis. All other analyses were based on unpartitioned maximum likelihood analysis, using the command given above for gene tree estimation. For the fully partitioned analysis, we used the following command:

```
raxmlHPC-SSE3 -m GTRGAMMA -s [input_alignment] -m GTRGAMMA
-n [output_name] -N 20
-M -q [partition_file] -p [random_seed_number]
```

Concatenation: We concatenate the alignments of all genes into one supermatrix, and then estimate a tree from the supermatrix using unpartitioned maximum likelihood. We computed a parsimony starting tree using RAxML version 7.3.5, and then ran RAxML-light version 1.0.6. The following commands were used:

```
raxmlHPC-SSE3 -y -s supermatrix.phylip -m GTRGAMMA -n [output_name]
-p [random_seed_number]
```

```
raxmlLight-PTHREADS -T 4 -s supermatrix.phylip -m GTRGAMMA
-n name
-t [parsimony_tree]
```

MP-EST: We used version 1.3 of MP-EST. We ran MP-EST 10 times with different random seed numbers, and selected the species tree with the best likelihood score using a custom shell script. MP-EST was run using site-only multi-locus bootstrapping, using 200 MLBS replicates, and returning the greedy consensus of the 200 MP-EST MLBS species trees as the output. The branch support on the edges of the tree represent the frequency of the bipartition in the sample of 200 species trees.

ASTRAL: We used ASTRAL version 4.7.6. in its default mode using the following command:

```
astral.4.7.6.jar -i [input_gene_trees] -o [output_file]
```

Greedy consensus: The greedy consensus (also called the "extended majority consensus") of a set of trees, all on the same set of leaves, is obtained by ordering the bipartitions that appear in one or more trees in the order of their frequency (most frequent first). Then, a tree is built from this set, beginning with the first bipartition, and then modifying the tree to include the next bipartition in the list, if the addition of the bipartition is possible. We used Dendropy version 3.12.0 [134] to compute greedy consensus trees when running MP-EST or ASTRAL with MLBS gene trees.

Data availability

Most of the datasets used in this study are available through the prior publications. The new datasets generated for this study are available on figshare, with DOI: http://dx.doi.org/10.6084/m9.figshare.1411146. (Retrieved May 13, 2015.) The weighted statistical binning software is available on github at
https://github.com/smirarab/binning (Retrieved May 14, 2015.)

B.2 Simulation protocols for 15- and 10-taxon datasets

Fig A.1 and A.2 illustrate the simulation protocols for 15- and 10-taxon datasets, respectively. The simulation protocol of generating the 10-taxon datasets is shown in Fig A.2. Gene trees were simulated using simPhy with commands given here (a) which includes all the parameter settings (important parameters are listed in panel (b)). The maximum tree length parameter is set to either 1.8M or 200K to produce two different model conditions. The speciation rate parameter is adjusted based on the maximum tree length so that maximum rate multiplied by speciation rate is always 0.2 (thus, rate is 0.000000111 and 0.000001 for 1.8M and 200K respectively). We used a perl script available at `http://www.cs.utexas.edu/~phylo/datasets/weighted-binning-datasets.html` to draw parameters for the Indelible simulations. Gene length is set to 1000 for all genes, but sequences are trimmed to their first 100bp in this study. For GTR+Γ parameters, we use a set of hyper parameters (estimated from real datasets) to drawn different parameter values for each gene in each replicate. Hyper parameters for base frequencies, GTR matrix, and the rate parameter (α) are shown in panel (c). These hyperparameters were calculated using maximum likelihood estimation form a collection of three large scale multi-locus datasets: 1KP dataset [151], Song et al Mammalian dataset [129], and Avian phylogenomics dataset [62]. The base values used for this maximum likelihood estimation and the corresponding scripts are available at `http://www.cs.utexas.edu/~phylo/software/astral/` (under the first bullet; i.e., estimate-parameters.r). Note that for the shape frequency, α, we use a heavy-tailed distribution, but to avoid unrealistic settings, values below 0.1 are discarded.

a) Control files used for MCcoal simulations (MCcoal.ctl):

```
SimulatedData.txt
9823126266
15 A B C D E F G H I J K L M N O
  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
((((((((((((((A #.05,B #.05):0.005 #.05,C #.05):0.01 #.05, D #.05):0.015
#.05,E #.05):0.02 #.05,F #.05):0.025 #.05,G #.05):0.03 #.05,H #.05):0.035
#.05,I #.05):0.04 #.05,J #.05):0.045 #.05,K #.05):0.05 #.05,L #.05):0.055
#.05,M #.05):0.06 #.05,N #.05):0.065 #.05,O #.05):0.565 #.05;
```

b) Command used to run MCcoal

```
   printf "10000 1000" PATH_TO_MCCOAL/MCcoal
```

c) Commands to run bppseqgen

```
mkdir allTrees;
split -a 4 -l 1 out.trees;
for i in x* ; do mv $i  allTrees/; done
for i in allTrees/x* ; do
 bppseqgen number_of_sites=1000 input.tree.file=$i param=opts
 output.sequence.file=$i".fasta"
done
```

d) GTR+Γ model parameters

```
model = GTR(a=1.062409952497, b=0.133307705766, c=0.195517800882,
d=0.223514845018, e=0.294405416545,
theta=0.469075709819, theta1=0.558949940165, theta2=0.488093447144)
rate_distribution = Gamma(n=4, alpha=0.370209777709)
```

Figure B.1: **Simulation parameters and commands for 15-taxon dataset**. Gene trees were simulated using MCcoal, with control files given here (a) and the command provided (b). The control files define the species tree, which is in the caterpillar form. Running MCcoal simulated 10,000 gene trees, which we divided into 10 replicates of 1000 genes or 100 genes. For each true gene tree, we then simulated alignments using bppseqgen [34], using the command given in (c). Here, the file "opts" is the same file we used in [90] and defines parameters given in (d).

a) Command used for SimPhy simulations:

```
for t in 1800000 200000; do
  b=0$(echo "scale=9; 1 / $t / 5"|bc -l}
  simphy -RS 20 -RL U:1000,1000 -RG 1 -ST U:$t,$t -SB U:$b,$b \\
            -SI U:1,1 -SL U:10,10 -CP U:400000,400000 -HS L:1.5,1 -HL L:1.2,1\\
            -HG L:1.4,1 -CU E:10000000 -SO U:1,1 -OD 1 -OR 0 -V 3  -CS 293745\\
            -O model.10.$t.$b |grep -E "[:-]"| tee log.10.$t.$b;
  for r in ‘ls -d model.$sp.$t.$b/*‘; do
     sed -i -e "s/_0_0//g" $r/g_trees*.trees;
  done
done
```

b) Parameter settings for SimPhy:

Arg.	Description	Value	Notes
ST	maximum tree length	200K or 1.8M	
SB	birth rates	0.000001 or 0.000000111	
SI	number of individuals per species	1	
SL	number of leaves	10	
P	global population sizes	400000	
HS	Species-specific branch rate heterogeneity modifiers	Log normal (1.5,1)	
HL	Locus-specific rate heterogeneity modifiers	Log normal (1.2,1)	
HG	Gene-tree-branch-specific rate heterogeneity modifiers	Log normal (1.4,1)	
U	Global substitution rate	Exponential (10000000)	
SO	Outgroup branch length relative to half the tree length	1	

c) Indelible GTR+Γ model parameters

Base frequencies $\sim Dirichlet(36, 26, 28, 32)$
GTR Matrix $\sim Dirichlet(16, 3, 5, 5, 6, 15)$
Rate parameter (α) $\sim Exponential(1.2)$ trimmed at 0.1 from below

Figure B.2: **Simulation parameters and commands for 10-taxon dataset**.

B.3 Supplementary tables

Dataset	Model condition	Gene tree error (%)		
		Unbinned	Binned-50	Binned-75
Avian	250bp	79	57	n.a.
	500bp	69	57	n.a.
	1000bp	55	51	n.a.
	1500bp	46	45	n.a.
Mammalian	250bp	60	n.a.	47
	500bp	43	n.a.	35
	1000bp	27	n.a.	26
15-taxon	100bp	77	80	86
	1000bp	36	36	40
10-taxon	Lower ILS	64	58	51
	Higher ILS	69	73	80

Table B.1: **Gene tree estimation error, with and without binning for simulated datasets**. We show the average gene tree estimation error for the simulated datasets analyzed in this study. Results are shown for fixed number of genes (1000 for avian and 200 for mammalian, 100 for 15-taxon and 100 for 10-taxon). We fixed the level of ILS to 1X for avian, mammalian and 15-taxon datasets; and varied the level of ILS for 10-taxon datasets with 100bp sequence length. Gene tree error is mean topological distance, measured using the missing branch rate between the true gene tree and all 200 bootstrap replicates of each estimated gene tree. For the supergene trees, each bootstrap replicate of each supergene tree is compared separately against each true gene tree for the genes put in that bin. “n.a.” stands for “not available”.

Dataset	Model condition	Average bootstrap support (%)
Avian	250bp	27
	500bp	31
	1000bp	51
	1500bp	60
Mammalian	250bp	43
	500bp	63
	1000bp	79
15-taxon	100bp	35
15-taxon	1000bp	70
10-taxon	Lower ILS, 100bp	45
	Higher ILS, 100bp	37

Table B.2: **Average bootstrap support**. We show the average bootstrap support values of the estimated gene trees for the simulated datasets. Results are shown for fixed number of genes (1000 for avian and 200 for mammalian, 100 for 15-taxon and 100 for 10-taxon datasets). We fixed the level of ILS to 1X for avian, mammalian and 15-taxon datasets; and varied the level of ILS for 10-taxon datasets with 100bp sequence length.

B.4 Supplementary figures

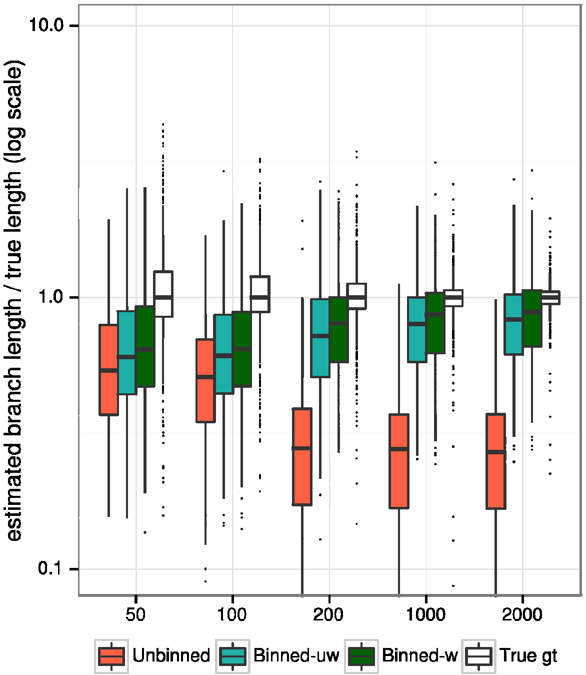


Figure B.3: **Effect of binning on the branch lengths (in coalescent units) estimated by MP-EST using MLBS on the avian simulated datasets with varying numbers of gene trees**. We show the species tree branch length error (the ratio of estimated branch length to true branch length for branches of the true tree that appear in the estimated tree; 1 indicates correct estimation). We varied the number of genes from 50 to 2000, and fixed the sequence length to 500bp with default amount of ILS (1X level). We used 50% bootstrap support threshold for binning. Supergene trees were estimated using unpartitioned analyses.

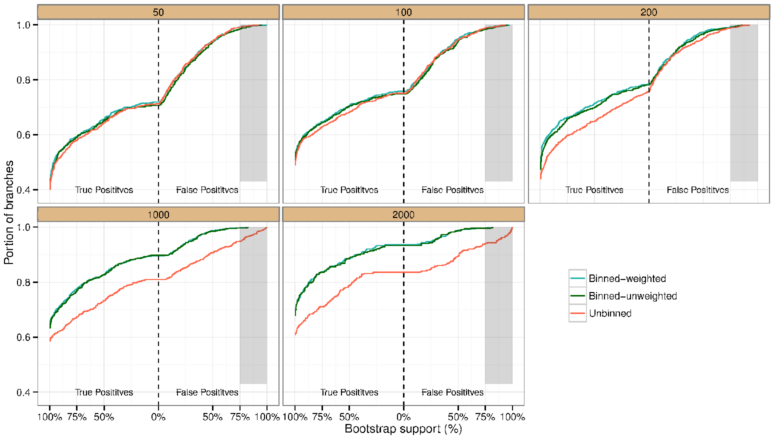
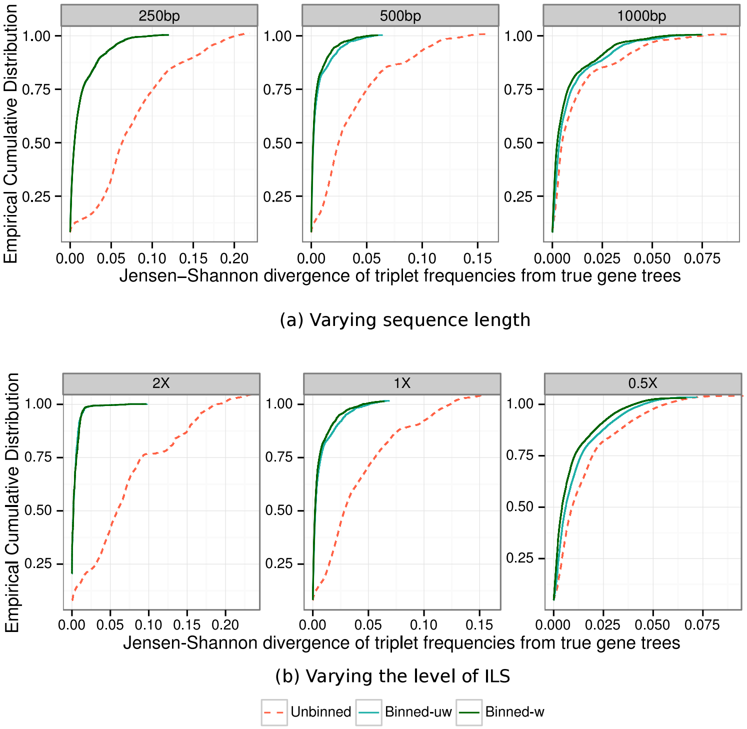


Figure B.4: **Cumulative distribution of the bootstrap support values (obtained using MLBS) of true positive (TP) and false positive (FP) edges estimated by MP-EST on avian datasets**. We varied the numbers of genes, and fixed the sequence length to 500bp (UCE-like) with default amount of ILS (1X level). We used 50% bootstrap support threshold for binning. Supergene trees were estimated using unpartitioned analyses. To produce the graph, we order the branches in the estimated species tree by their quality, so that the true positives with high support come first, followed by lower support true positives, then by false positives with low support, and finally by false positives with high support. The false positive branches with support above 75% are the most troublesome, and that fraction are indicated in the grey area. When the curve for a method lies above the curve for another method, then the first method has better bootstrap support.



(a) Varying sequence length

(b) Varying the level of ILS

Figure B.5: **Divergence of estimated gene trees triplet distributions from true gene tree distributions for simulated mammalian datasets**. (a) Varying gene sequence alignments lengths with 200 number of genes and default levels of ILS (1X); (b) varying ILS levels with fixed 200 genes and sequence length fixed to 500bp (63% BS). We used 75% bootstrap support threshold for binning. Supergene trees were estimated using unpartitioned analyses. True triplet frequencies are estimated based on true gene trees for each of the $\binom{n}{3}$ possible triplets, where n is the number of species. Similarly, triplet frequencies are calculated from estimated gene/supergene trees. For each of these $\binom{n}{3}$ triplets, we calculate the Jensen-Shannon divergence of the estimated triplet distribution from the true gene tree triplet distribution. We show the empirical cumulative distribution of these divergences. The empirical cumulative distribution shows that for a given divergence level, what percentage of the triplets are diverged from true triplet distribution at or below that level. Results are shown for 10 replicates.

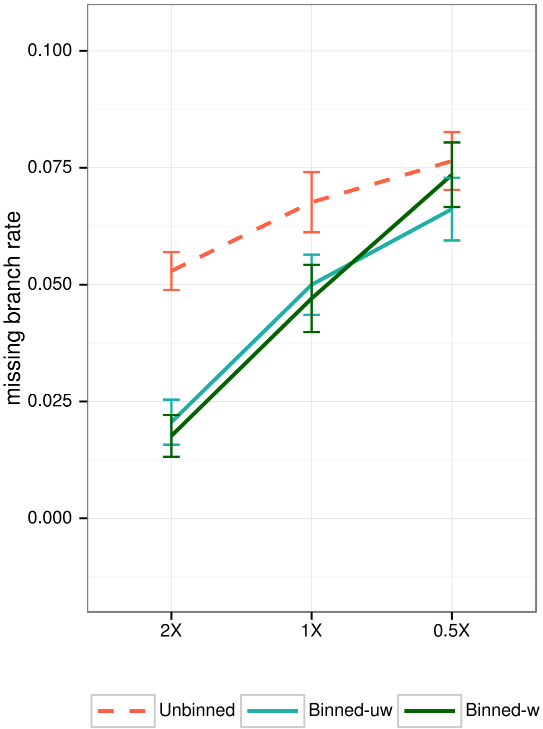


Figure B.6: **Species tree estimation error for MP-EST with MLBS on mammalian simulated datasets with varying amounts of ILS**. We show average FN rate over 20 replicates. We varied the amount of ILS, and fixed the number of genes to 200 and sequence length to 500bp (63% BS). We used 75% bootstrap support threshold for binning. Supergene trees were estimated using unpartitioned analyses.

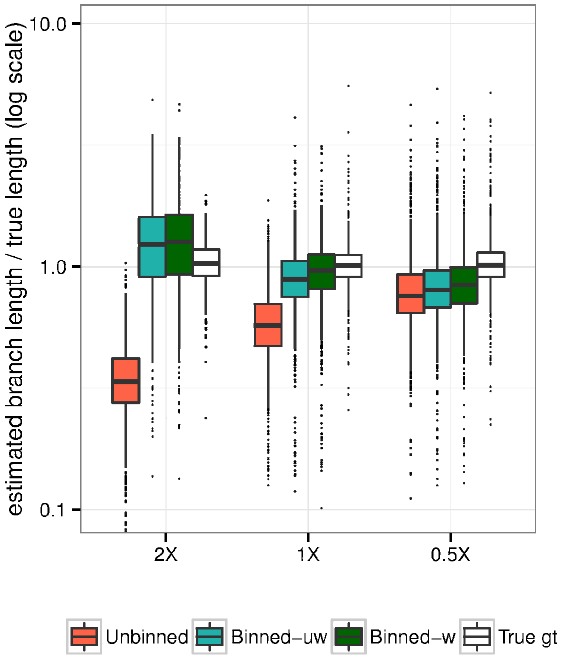


Figure B.7: **Effect of binning on the branch lengths (in coalescent unit) estimated by MP-EST using MLBS on the mammalian simulated datasets with varying amounts of ILS**. We show the species tree branch length error (the ratio of estimated branch length to true branch length for branches of the true tree that appear in the estimated tree; 1 indicates correct estimation). We varied the amount of ILS, and fixed the number of genes to 200 and sequence length to 500bp (63% BS). We used 75% bootstrap support threshold for binning. Supergene trees were estimated using unpartitioned analyses.

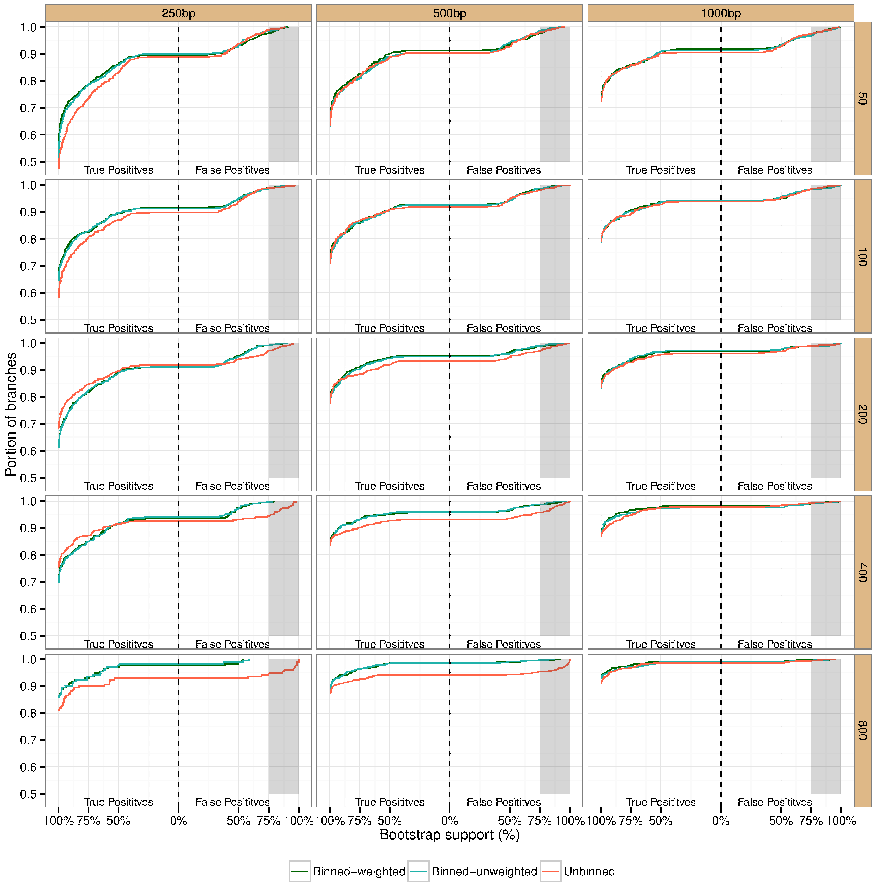


Figure B.8: **Cumulative distribution of the bootstrap support values (obtained using MLBS) of true positive (TP) and false positive (FP) edges estimated by MP-EST on mammalian datasets**. We varied the numbers of genes, and gene sequence alignments length with default amount of ILS. We used 75% bootstrap support threshold for binning. Supergene trees were estimated using unpartitioned analyses.

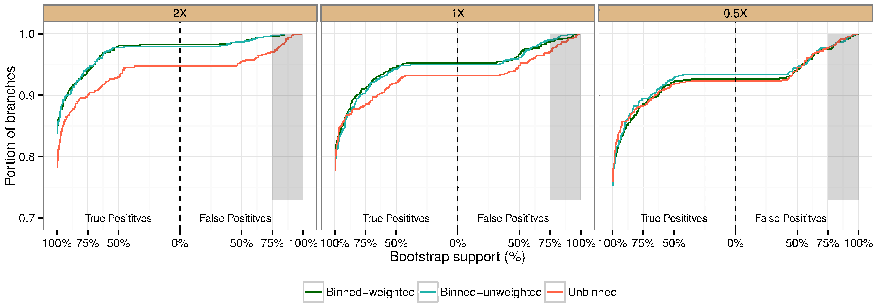


Figure B.9: **Cumulative distribution of the bootstrap support values (obtained using MLBS) of true positive (TP) and false positive (FP) edges estimated by MP-EST on mammalian datasets with varying amounts of ILS**. We varied the amount of ILS, and fixed the number of genes to 200 and sequence length to 500bp. We used 75% bootstrap support threshold for binning. Supergene trees were estimated using unpartitioned analyses. To produce the graph, we order the branches in the estimated species tree by their quality, so that the true positives with high support come first, followed by lower support true positives, then by false positives with low support, and finally by false positives with high support. When the curve for a method lies above the curve for another method, then the first method has better bootstrap support.

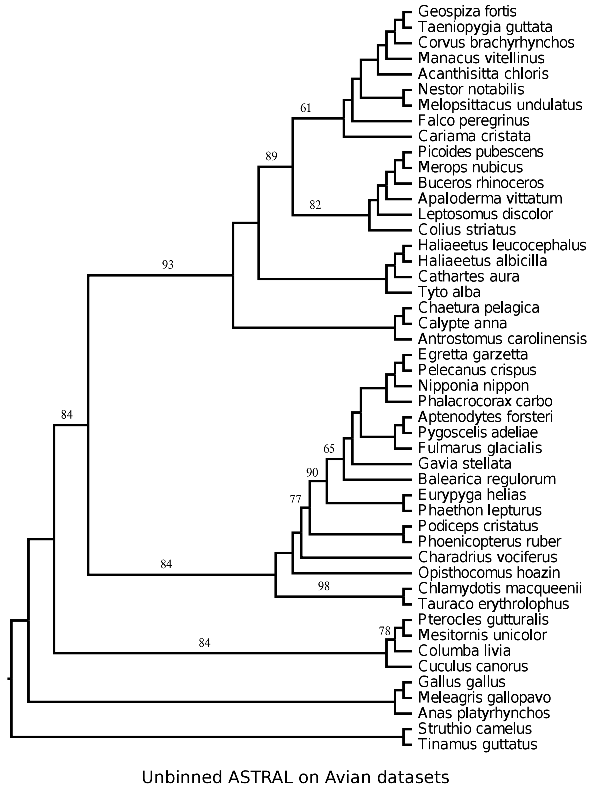


Figure B.10: **Species trees estimated by unbinned ASTRAL using MLBS on avian biological datasets.** Branches without designation have 100% support. We used 50% bootstrap support threshold for binning. Supergene trees were estimated using fully partitioned analyses.

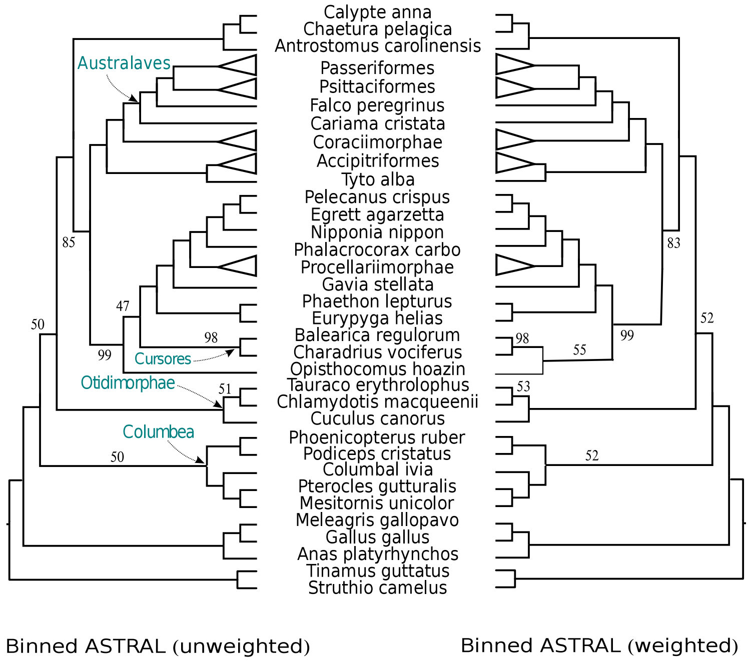


Figure B.11: **Species trees estimated by binned (with and without weighting) ASTRAL using MLBS on avian biological datasets.** (a) Unweighted binned ASTRAL, and (b) weighted binned ASTRAL. Branches without designation have 100% support. We used 50% bootstrap support threshold for binning. Supergene trees were estimated using fully partitioned analyses. Binned ASTRAL with weighting and binned ASTRAL without weighting differ only in the placement of *Opisthocomus hoazin*. However, the branches supporting different placements of *Opisthocomus hoazin* have low support values (47% for unweighted binning and 55% for weighted binning).

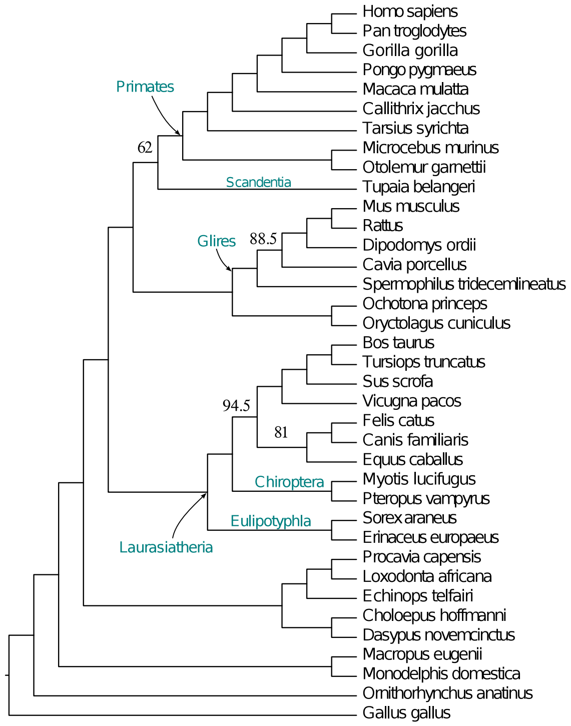


Figure B.12: **Species trees estimated by unbinned MP-EST using MLBS for mammalian biological datasets.** Branches without designation have 100% support. We used 75% bootstrap support threshold for binning. We estimated the supergene trees using fully partitioned analyses.

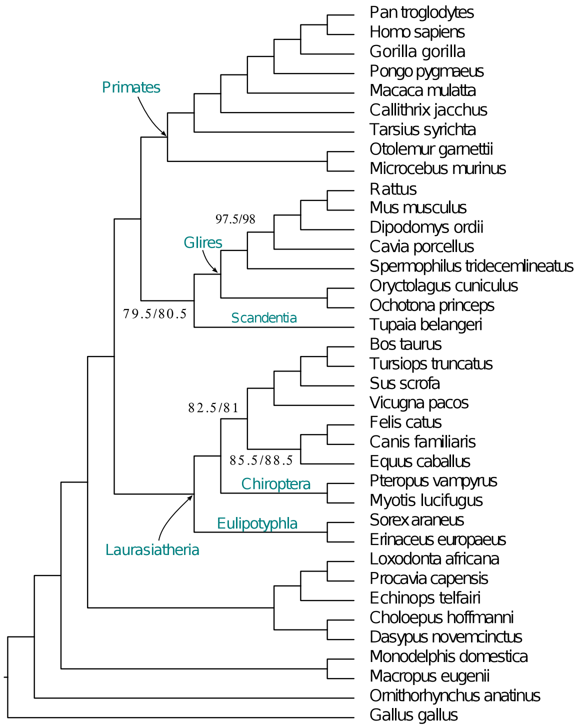


Figure B.13: **Species trees estimated by binned (with and without weighting) MP-EST using MLBS for mammalian biological datasets.** Binned and unbinned ASTRAL returned identical topology. The branches on this tree are labeled with two support values side by side: the first one corresponds to unweighted binning and the next one corresponds to weighted binning. Branches without designation have 100% support. We used 75% bootstrap support threshold for binning. Supergene trees were estimated using fully partitioned analyses.

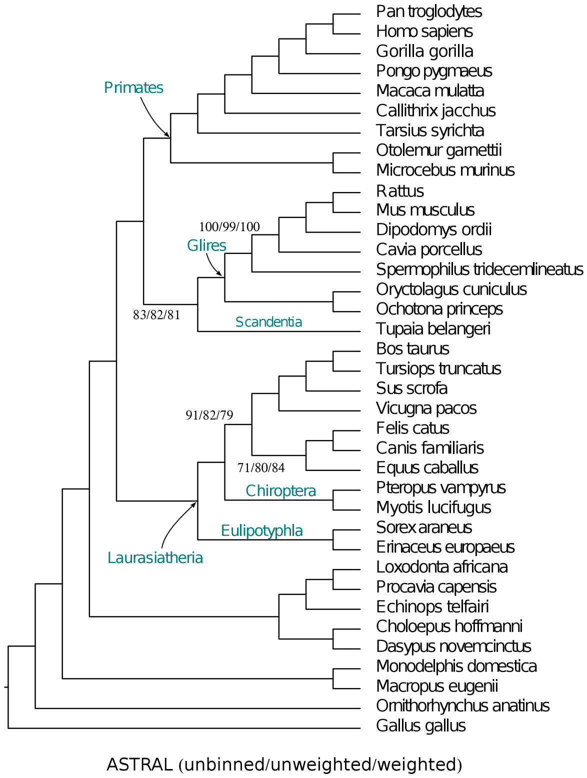


Figure B.14: **Species trees estimated by unbinned and binned (with and without weighting) ASTRAL using MLBS for mammalian biological datasets.** Binned and unbinned ASTRAL returned identical topology. The branches on this tree are labeled with three support values side by side: the first one corresponds to unbinned ASTRAL, the next one corresponds to unweighted binning, and the last one is for weighted binning. Branches without designation have 100% support. We used 75% bootstrap support threshold for binning. Supergene trees were estimated using fully partitioned analyses.

Index

*BEAST, 7, 29, 119, 120, 147, 237

Abstract, x
Acknowledgments, v
Allele, 16
ANOVA test, 204
Appendices, 277
ASTRAL, 29, 187, 198, 317

Bayesian method, 84, 120, 144
Bibliography, 366
Binning, 7, 181, 191
  Naive binning, 7, 147, 185
  Statistical binning, 7, 185
  Weighted statistical binning, 7, 184, 317
Biological datasets, 245, 255, 323
  Amniota dataset, 245, 255
  Avian dataset, 194
  Mammalian dataset, 194, 245, 257
Bipartitions, 12
Bootstrap support, 150, 187, 188, 194, 331
Bootstrapping, 295
Branch length, 10
Branch support, 319
Brélaz heuristic, 190
BUCKy, 7, 29

Clade, 12
Clique, 5, 36, 86
  Maximum weight clique, 36
  Minimum weight clique, 37
Cluster, 12
Coalescent model, 15
Combined analyses, *see also* Concatenation, 148, 291
Compatibility graph, 35, 102, 110
Completion, 42
Concatenation, *see also* Combined analyses, 28, 147, 148, 150, 256, 324
Conclusions, 270
Containment, 34
Contraction, 13

DACTAL, 238, 241
Dedication, iv
Deep coalescence, *see also* Incomplete Lineage Sorting (ILS), 22, 26, 118, 149
Disk Covering Method (DCM), 235, 238
Divide-and-conquer, 4, 235
Domination, 34, 59, 74
DS-tree, 88
Duplication node, 24, 45
DupTree, 30, 38, 85
Dynamic programming, 39, 48, 56, 61, 83, 86, 114

ESS value, 285
Extra lineages, 17, 18, 26, 103

FN rate, 31
FP rate, 33

Gene birth-and-death, 85
Gene duplication and loss, 2, 14, 36, 59, 83
Gene loss, 15

Gene tree, 2, 14, 283
Gene tree discordance, 2, 14
Gene Tree Parsimony (GTP), 26, 83
Gene tree reconciliation, 21, 88
Genes, 1, 14, 235, 245
Greedy consensus, 7, 29, 145, 147, 154, 290

Homeomorphic subtree, 24, 87
Horizontal Gene Transfer (HGT), 2, 17
Hybridization, 2

iGTP, 30, 37, 85, 120
Incompatibility graph, 189
Incomplete gene tree, 3, 83, 118
Incomplete Lineage Sorting (ILS), 2, 15, 118, 147, 184, 235
Introduction, 1

Jensen-Shannon divergence, 202, 318
Jukes-Cantor model, 152

Lateral gene transfer, 19, *see also* Horizontal Gene Transfer (HGT)

Maximum likelihood, 188, 245
- Partitioned analyses, 186
- Unpartitioned analyses, 190, 191, 324

Minimize Deep Coalescence (MDC), 39, 120
Minimize Gene Duplications (MGD), 4, 36, 60
Minimize Gene Duplications and Losses (MGDL), 4, 36, 60
Missing clade, 87
- Maximal missing clade, 87

Missing taxa, *see also* Incomplete gene tree, 135
Molecular clock, 11, 60, 244
- Strict molecular clock, 6, 323

MP-EST, 7, 147, 185, 187, 235, 317
MRCA mapping, 22, 25
MRL, 29, 239, 266
MRP, 119, 120, 239, 288
Multi-locus Bootstrapping (MLBS), 317
Multi-species coalescent model, 7, 16

Naive binning, 278
NP-hard, 64

Optimal reconciliation, 21, 22, 42
Orthologous, 15
Outgroup, 6, 11, 60

Paralogous, 15
Phylogenetic signal, 183
Phylogenetic tree, 1, 9
Phylogenomic analyses, 2, 28
Phylogeny, 9, 276
Phylonet, 120, 155, 290
Preliminaries, 9
Pseudo-likelihood score, 242, 259

Quartet support, 243, 259

RAxML, 156, 245
Recombination, 274
Refinement, 13
RF rate, 33

Short quartet, 240
- Padded short quartet, 240

Short Subtree Graph (SSG), 240, 242

simPhy, 326
Simulated datasets, 244
  10-taxon datasets, 194, 196, 323
  100-taxon datasets, 135, 139
  11-taxon datasets, 135, 152, 280
  15-taxon datasets, 194, 195, 322
  17-taxon datasets, 135, 136, 152, 282
  Avian datasets, 194, 321
  Mammalian datasets, 194, 244, 320
Speciation node, 24, 45
Species tree, 1, 14, 235
Standard error, 279
Statistical consistency, 27, 145, 266
Strict Consensus Merger (SCM), 266
Subsampling, 283
Subtree-bipartition, 5, 34, 36, 59, 102
Summary method, 2, 28, 147, 184, 191, 237
SuperFine, 239, 266, 267
Supergene, 148, 189, 190
Supergene tree, 190, 324
Supermatrix, 2, 28, 148, 175, 324
Supertree, 239

Taxon, 10, 43
Taxon sampling, 6, 83
Threshold support, 188
Topologies, 11
Tree of Life, 1, 21, 276
Triplet, 175, 201, 318
  Triplet frequency, 201
True biological loss, 6, 83
True tree, 31

Unrooted gene tree, 59, 82

Vertex coloring, 189

Wilcoxon signed-rank test, 156, 243, 254

Bibliography

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

[2] B. Baum. Combining trees as a way of combining data sets for phylogenetic inference, and the desirability of combining gene trees. *Taxon*, 41(1):3–10, 1992.

[3] B. R. Baum and M. A. Ragan. The MRP method. In *Phylogenetic Supertrees*, volume 4 of *Computational Biology*, pages 17–34, 2004.

[4] M. S. Bayzid, T. Hunt, and T. Warnow. Disk covering methods improve phylogenomic analyses. *BMC Genomics*, 15(Suppl 6):S7, 2014.

[5] M. S. Bayzid, S. Mirarab, B. Boussau, and T. Warnow. Weighted statistical binning: Enabling statistically consistent genome-scale phylogenetic analyses. *PLoS ONE*, 10(6):e0129183, 2015.

[6] M. S. Bayzid, S. Mirarab, and T. Warnow. Inferring optimal species trees under gene duplication and loss. In *Proceedings of the of Pacific Symposium on Biocomputing (PSB)*, volume 18, pages 250–261, 2013.

[7] M. S. Bayzid and T. Warnow. Estimating optimal species trees from incomplete gene trees under deep coalescence. *Journal of Computational Biology*, 19(6):591–605, 2012.

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

[9] L. M. Boykin, L. S. Kubatko, and T. K. Lowrey. Comparison of methods for rooting phylogenetic trees: A case study using orcuttieae (poaceae: Chloridoideae). *Molecular Phylogenetics and Evolution*, 54(3):687–700, 2010.

[10] D. Brélaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22(4):251–256, 1979.

[11] D. Brooks and D. McLennan. *Phylogeny, Ecology, and Behavior.* University of Chicago Press, 1991.

[12] D. Bryant. A classification of consensus methods for phylogenetics. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 1991.

[13] R. Bush, C. Bender, K. Subbarao, N. Cox, and W. Fitch. Predicting the evolution of human influenza A. *Science*, 286(5446):1921–1925, 1999.

[14] W. C. Chang, G. Burleigh, D. F. Baca, and O. Eulenstein. An ILP solution for the gene duplication problem. *BMC Bioinformatics*, 12(Suppl 1):S14, 2011.

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

[16] C. Chauve, J. P. Doyon, and N. El-Mabrouk. Gene family evolution by duplication, speciation, and loss. *Journal of Computational Biology*, 15(8):1043–1062, 2008.

[17] Y. Chiari, V. Cahais, N. Galtier, and F. Delsuc. Phylogenomic analyses support the position of turtles as the sister group of birds and crocodiles (archosauria). *BMC Biology*, 10(1):65, 2012.

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

[19] Y. Chung and C. Ané. Comparing two Bayesian methods for gene tree/species tree reconstruction: A simulation with incomplete lineage sorting and horizontal gene transfer. *Systematic Biology*, 60(3):261–275, 2011.

[20] J. Cotton and R. D. M. Page. Tangled tales from multiple markers: reconciling conflict between phylogenies to build molecular supertrees. In *Phylogenetic Supertrees*, volume 4 of *Computational Biology*, pages 107–125, 2004.

[21] G. Dasarathy, R. Nowak, and S. Roch. New sample complexity bounds for phylogenetic inference from multiple loci. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, pages 2307–2041, 2014.

[22] F. de la Cruz and J. Davies. Horizontal gene transfer and the origin of species: lessons from bacteria. *Trends in Microbiology*, 8(3):128–133, 2000.

[23] M. DeGiorgio and J. Degnan. Robustness to divergence time underestimation when inferring species trees from estimated gene trees. *Systematic Biology*, 63(1):66–82, 2014.

[24] M. DeGiorgio and J. H. Degnan. Fast and consistent estimation of species trees using supermatrix rooted triples. *Molecular Biology and Evolution*, 27(3):552–569, 2010.

[25] J. H. Degnan. Anomalous unrooted gene trees. *Systematic Biology*, 62(4):574–590, 2013.

[26] J. H. Degnan, M. DeGiorgio, D. Bryant, and N. A. Rosenberg. Properties of consensus methods for inferring species trees from gene trees. *Systematic Biology*, 58:35–54, 2009.

[27] J. H. Degnan and N. A. Rosenberg. Discordance of species trees with their most likely gene trees. *PLoS Genetics*, 2:762 – 768, 2006.

[28] J. H. Degnan and N. A. Rosenberg. Gene tree discordance, phylogenetic inference and the multispecies coalescent. *Trends in Ecology and Evolution*, 24(6):332–340, 2009.

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

[30] W. F. Doolittle. Phylogenetic classification and the universal tree. *Science*, 284(5423):2124–2128rambaut, 1999.

[31] J. P. Doyon and C. Chauve. Branch-and-bound approach for parsimonious inference of a species tree from a set of gene family trees. *Advances in Experimental Medicine and Biology*, 696:287–295, 2011.

[32] J. P. Doyon, V. Ranwez, V. Daubin, and V. Berry. Models, algorithms and programs for phylogeny reconciliation. *Brieifings in Bioinformatics*, 12(5):392–400, 2011.

[33] A. Drummond and A. Rambaut. BEAST: Bayesian evolutionary analysis by sampling trees. *BMC Evolutionary Biology*, 7(1):214, 2007.

[34] J. Dutheil and B. Boussau. Non-homogeneous models of sequence evolution in the Bio++ suite of libraries and programs. *BMC Evolutinary Biology*, 8(1):255, 2008.

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

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

[37] W. Fitch and E. Margoliash. Construction of phylogenetic trees. *Science*, 155(3760):279 – 284, 1967.

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

[39] B. Fuglede and F. Topsoe. Jensen-Shannon divergence and Hilbert space embedding. In *Proceedings of the IEEE International Symposium on Information Theory*, page 31, 2004.

[40] H. Gabow and R. Tarjan. A linear-time algorithm for a special case of disjoint set union. In *Proceedings of the 15th ACM Symposium on Theory of Computing (STOC)*, pages 246–251, 1983.

[41] J. Gatesy and M. Springer. Phylogenetic analysis at deep timescales: Unreliable gene trees, bypassed hidden support, and the coalescence/concatalescence conundrum. *Molecular Phylogenetics and Evolution*, 80:231–266, 2014.

[42] M. C. Golumbic. *Algorithmic graph theory and perfect graphs*, volume 57. Elsevier, 2004.

[43] M. Goodman, J. Czelusniak, G. Moore, E. Romero-Herrera, and G. Matsuda. Fitting the gene lineage into its species lineage: a parsimony strategy illustrated by cladograms constructed from globin sequences. *Systematic Zoology*, 28(2):132–163, 1979.

[44] P. Górecki. Reconciliation problems for duplication, loss and horizontal gene transfer. In *Proceedings of the 8th Annual International Conference on Computational Molecular Biology*, pages 316 – 325, 2004.

[45] P. Górecki and J. Tiuryn. DLS-trees: A model of evolutionary scenarios. *Theoretical Computer Science*, 359(8):378–399, 2006.

[46] 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*, 19(10):1769–1781, 2002.

[47] R. Guigo, I. Muchnik, and T. Smith. Reconstruction of ancient molecular phylogeny. *Molecular Phylogenetics and Evolution*, 6(2):189–213, 1996.

[48] D. Gusfield. Efficient algorithms for inferring evolutionary trees. *Networks*, 21(1):19–28, 1991.

[49] K. M. Halanych and L. R. Goertzen. Grand challenges in organismal biology: the need to develop both theory and resources. *Integrative and Comparative Biology*, 49(5):475–479, 2009.

[50] M. T. Hallett and J. Lagergren. New algorithms for the duplication-loss model. In *Proceedings of the International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 138–146, 2000.

[51] P. Harvey and M. Pagel. *The Comparative Method in Evolutionary Biology.* Oxford University Press, 1991.

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

[53] H. Huang, Q. He, L. S. Kubatko, and L. L. Knowles. Sources of error inherent in species-tree estimation: impact of mutational and coalescent effects on accuracy and implications for choosing among different methods. *Systematic Biology*, 59(5):573–583, 2010.

[54] R. R. Hudson. Testing the constant-rate neutral allele model with protein sequence data. *Evolution*, 37(1):203 – 217, 1983.

[55] J. Huelsenbeck and R. Ronquist. MrBayes: Bayesian inference of phylogeny. *Bioinformatics*, 17(8):754–755, 2001.

[56] J. P. Huelsenbeck, J. P. Bollback, and A. M. Levine. Inferring the root of a phylogenetic tree. *Systematic biology*, 51(1):32–43, 2002.

[57] A. F. Hugall, R. Foster, and M. S. Lee. Calibration choice, rate smoothing, and the pattern of tetrapod diversification according to the long nuclear gene rag-1. *Systematic Biology*, 56(4):543–563, 2007.

[58] D. Huson, S. Nettles, and T. Warnow. Disk-covering, a fast converging method for phylogenetic tree reconstruction. *Journal of Computational Biology*, 6(3):369–386, 1999.

[59] D. Huson, L. Vawter, and T. Warnow. Solving large scale phylogenetic problems using DCM2. In *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB'99)*, pages 118–129, 1999.

[60] N. Iwabe, Y. Hara, Y. Kumazawa, K. Shibamoto, Y. Saito, T. Miyata, and K. Katoh. Sister group relationship of turtles to the bird-crocodilian clade revealed by nuclear DNA–coded proteins. *Molecular Biology and Evolution*, 22(4):810–813, 2005.

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

[62] E. D. Jarvis, S. Mirarab, A. J. Aberer, B. Li, P. Houde, C. Li, S. Y. W. Ho, B. C. Faircloth, B. Nabholz, J. T. Howard, A. Suh, C. C. Weber, R. R. da Fonseca, J. Li, F. Zhang, H. Li, L. Zhou, N. Narula, L. Liu, G. Ganapathy, B. Boussau, M. S. Bayzid, V. Zavidovych, S. Subramanian, T. Gabaldón, S. Capella-Gutiérrez, J. Huerta-Cepas, B. Rekepalli, K. Munch, M. Schierup, B. Lindow, W. C. Warren, D. Ray, R. E. Green, M. W. Bruford, X. Zhan, A. Dixon, S. Li, N. Li, Y. Huang, E. P. Derryberry, M. F. Bertelsen, F. H. Sheldon, R. T. Brumfield, C. V. Mello, P. V. Lovell, M. Wirthlin, M. P. C. Schneider, F. Prosdocimi, J. A. Samaniego, A. M. V. Velazquez, A. Alfaro-Núñez, P. F. Campos, B. Petersen, T. Sicheritz-Ponten, A. Pas, T. Bailey, P. Scofield, M. Bunce, D. M. Lambert, Q. Zhou, P. Perelman, A. C. Driskell, B. Shapiro, Z. Xiong, Y. Zeng, S. Liu, Z. Li, B. Liu, K. Wu, J. Xiao, X. Yinqi, Q. Zheng, Y. Zhang, H. Yang, J. Wang, L. Smeds, F. E. Rheindt, M. Braun, J. Fjeldsa, L. Orlando, F. K. Barker, K. A. Jønsson, W. Johnson, K.-P. Koepfli, S. O'Brien, D. Haussler, O. A. Ryder, C. Rahbek, E. Willerslev, G. R. Graves, T. C. Glenn, J. McCormack, D. Burt, H. Ellegren, P. Alström, S. V. Edwards, A. Stamatakis, D. P. Mindell,

J. Cracraft, E. L. Braun, T. Warnow, W. Jun, M. T. P. Gilbert, and G. Zhang. Whole genome analyses resolve early branches in the tree of life of modern birds. *Science*, 346(6215):1320–1331, 2014.

[63] S. Kannan, T. Warnow, and S. Yooseph. Computing the local consensus of trees. *SIAM Journal of Computing*, 27(6):1695–1724, 1995.

[64] R. Karp. *Reducibility among combinatorial problems.* Plenum, 1972.

[65] R. T. Kimball, N. Wang, V. Heimer-McGinn, C. Ferguson, and E. L. Braun. Identifying localized biases in large datasets: a case study using the avian tree of life. *Molecular Phylogenetics and Evolution*, 69(3):1021–1032, 2013.

[66] J. F. C. Kingman. The coalescent. *Stochactic Processes and their Applications*, 13(3):235–248, 1982.

[67] L. L. Knowles. Estimating species trees: methods of phylogenetic analysis when there is incongruence across genes. *Systematic Biology*, 58(5):463–7, Oct. 2009.

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

[69] L. S. Kubatko and J. H. Degnan. Inconsistency of phylogenetic estimates from concatenated data under coalescence. *Systematic Biology*, 56(1):17–24, 2007.

[70] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[71] V. Kumar, B. M. Hallström, and A. Janke. Coalescent-based genome analyses resolve the early branches of the euarchontoglires. *PLoS ONE*, 8(4):e60019, 2013.

[72] A. Kupczok, H. A. Schmidt, and A. von Haeseler. Accuracy of phylogeny reconstruction methods combining overlapping gene data sets. *Algorithms for Molecular Biology*, 5:37, 2010.

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

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

[75] C. R. Linder and T. Warnow. An overview of phylogeny reconstruction. *Handbook of Computational Molecular Biology*, 2005.

[76] K. Liu, S. Raghavan, S. Nelesen, C. R. Linder, and T. Warnow. Rapid and accurate large-scale coestimation of sequence alignments and phylogenetic trees. *Science*, 324(5934):1561–1564, 2009.

[77] K. Liu, T. J. Warnow, M. T. Holder, S. M. Nelesen, J. Yu, A. P. Stamatakis, and C. R. Linder. SATé-II: Very fast and accurate simultaneous estimation of multiple sequence alignments and phylogenetic trees. *Systematic Biology*, 61(1):60–106, 2011.

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

[79] L. Liu and L. Yu. Estimating species trees from unrooted gene trees. *Systematic Biology*, 60(5):661–667, 2011.

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

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

[82] B. Ma, M. Li, and L. Zhang. On reconstructing species trees from gene trees in terms of duplications and losses. In *Proceedings of the International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 182 – 191, 1998.

[83] B. Ma, M. Li, and L. Zhang. From gene trees to species trees. *SIAM Journal of Computing*, 30(3):729–752, 2000.

[84] W. P. Maddison. Gene trees in species trees. *Systematic Biology*, 46(3):523–536, 1997.

[85] W. P. Maddison and D. R. Maddison. Mesquite: a modular system for evolutionary analysis.

[86] D. Mallo, L. D. O. Martins, and D. Posada. Simphy: Phylogenomic simulation of gene, locus, and species trees. *Systematic Biology*, 65(2):334–344, 2015.

[87] J. E. McCormack, M. G. Harvey, B. C. Faircloth, N. G. Crawford, T. C. Glenn, and R. T. Brumfield. A phylogeny of birds based on over 1,500 loci collected by target enrichment and high-throughput sequencing. *PLoS ONE*, 8(1):e54848, 2013.

[88] R. W. Meredith, J. E. Janečka, J. Gatesy, O. A. Ryder, C. A. Fisher, E. C. Teeling, A. Goodbla, E. Eizirik, T. L. L. Simão, T. Stadler, D. L. Rabosky, R. L. Honeycutt,

J. J. Flynn, C. M. Ingram, C. Steiner, T. L. Williams, T. J. Robinson, A. Burk-Herrick, M. Westerman, N. A. Ayoub, M. S. Springer, and W. J. Murphy. Impacts of the Cretaceous Terrestrial Revolution and KPg extinction on mammal diversification. *Science*, 334(6055):521–4, 2011.

[89] S. Mirarab. *Novel scalable approaches for multiple sequence alignment and phylogenomic reconstruction*. PhD thesis, University of Texas at Austin, 2015.

[90] S. Mirarab, M. S. Bayzid, B. Boussau, and T. Warnow. Statistical binning improves species tree estimation in the presence of gene tree incongruence. *Science*, 346(6215):1250463, 2014.

[91] 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*, doi: 10.1093/sysbio/syu063, 2014.

[92] S. Mirarab, N. Nguyen, S. Guo, L.-S. Wang, J. Kim, and T. Warnow. PASTA: ultra-large multiple sequence alignment for nucleotide and amino-acid sequences. *Journal of Computational Biology*, 22(5):377–386, 2015.

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

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

[95] B. Mirkin, I. Muchnik, and T. Smith. A biologically consistent model for comparing molecular phylogenies. *Journal of Computational Biology*, 2(4):493–507, 1995.

[96] E. Mossel and S. Roch. Incomplete lineage sorting: consistent phylogeny estimation from multiple loci. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(1):166–171, 2011.

[97] L. Nakhleh, U. Roshan, K. S. James, J. Sun, and T. Warnow. Designing fast converging phylogenetic methods. *Bioinformatics*, 17(suppl 1):S190–S198, 2001.

[98] M. Nei. Stochastic errors in DNA evolution and molecular phylogeny. *Progress in Clinical and Biological Research*, 218:133 – 147, 1986.

[99] M. Nei. *Molecular evolutionary genetics*. Columbia University Press, 1987.

[100] S. Nelesen, K. Liu, L. S. Wang, C. R. Linder, and T. Warnow. Dactal: divide-and-conquer trees (almost) without alignments. *Bioinformatics*, 28(12):i274–i282, 2012.

[101] D. T. Neves, T. Warnow, J. L. Sobral, and K. Pingali. Parallelizing superfine. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pages 1361–1367, 2012.

[102] N. Nguyen, S. Mirarab, and T. Warnow. MRL and SuperFine+MRL: new supertree methods. *Algorithms for Molecular Biology*, 7:3, 2012.

[103] R. Page. Genetree: comparing gene and species phylogenies using reconciled trees. *Bioinformatics*, 14(9):819–820, 1998.

[104] R. Page and M. Charleston. From gene to organismal phylogeny: reconciled trees and the gene tree/species tree problem. *Molecular Phylogentics and Evolution*, 7(2):231–240, 1997.

[105] R. Page and M. Charleston. Reconciled trees and incongruent gene and species trees. *Mathematical Hierarchies in Biology*, 37:57–70, 1997.

[106] R. D. Page. Genes, organisms, and areas: the problem of multiple lineages. *Systematic Biology*, 42(1):77–84, 1993.

[107] R. D. M. Page. Maps between trees and cladistic analysis of historical associations amoung genes, organisms and areas. *Systematic Biology*, 43(1):58–77, 1994.

[108] R. D. M. Page. Extracting species trees from complex gene trees: Reconciled trees and vertebrate phylogeny. *Molecular Phylogenetics and Evolution*, 14(1):89–106, 2000.

[109] R. D. M. Page and J. A. Cotton. Vertebrate phylogenomics: reconciled trees and gene duplications. In *Proceedings of the Pacific Symposium On Biocomputing*, pages 536–547, 2002.

[110] P. Pamilo and M. Nei. Relationship between gene trees and species trees. *Molecular Biology and Evolution*, 5(5):568–583, 1998.

[111] S. Patel, R. T. Kimball, and E. L. Braun. Error in phylogenetic estimation for bushes in the tree of life. *Journal of Phylogenetics and Evolutionary Biology*, 1(2):110, 2013.

[112] H. Philippe and P. Forterre. The rooting of the universal tree of life is not reliable. *Journal of Molecular Evolution*, 49(4):509–523, 1999.

[113] C. Phillips and T. Warnow. The asymmetric median tree: a new model for building consensus trees. *Discrete Applied Mathematics*, 71(1):311–335, 1996.

[114] M. Price, P. Dehal, and A. Arkin. FastTree: computing large minimum evolution trees with profiles instead of a distance matrix. *Molecular Biology and Evolution*, 26(7):1641–1650, 2009.

[115] M. A. Ragan. Phylogenetic inference based on matrix representation of trees. *Molecular Phylogenetics and Evolution*, 1(1):53–58, 1992.

[116] A. Rambaut and N. C. Grassly. Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Computer Applications in the Biosciences*, 13(3):235–238, 1997.

[117] B. Rannala, J. Huelsenbeck, Z. Yang, and R. Nielsen. Taxon sampling and the accuracy of large phylogenies. *Systematic Biology*, 47(4):702–710, 1998.

[118] B. Redelings and M. Suchard. Joint Bayesian estimation of alignment and phylogeny. *Systematic Biology*, 54(3):401–418, 2005.

[119] S. Roch and M. Steel. Likelihood-based tree reconstruction on a concatenation of aligned sequence data sets can be statistically inconsistent. *Theoretical Population Biology*, 100:56–62, 2014.

[120] S. Roch and T. Warnow. On the robustness to gene tree estimation error (or lack thereof) of coalescent-based species tree methods. *Systematic Biology*, doi: 10.1093/sysbio/syv016, 2015.

[121] N. Rosenberg. The probability of topological concordance of gene trees and species trees. *Theoretical Population Biology*, 61(2):225–247, 2002.

[122] N. A. Rosenberg. Discordance of species trees with their most likely gene trees: A unifying principle. *Molecular Biology and Evolution*, 30(12):2709–2713, 2013.

[123] U. Roshan, B. Moret, T. Williams, and T. Warnow. Rec-I-DCM3: A fast algorithmic technique for reconstructing large phylogenetic trees. In *Proceedings of the 3rd Computational Systems Bioinformatics Conference*, pages 98–109, 2004.

[124] L. Salichos and A. Rokas. Inferring ancient divergences requires genes with strong phylogenetic signals. *Nature*, 497(7449):327–31, 2013.

[125] M. Sanderson and M. McMahon. Inferring angiosperm phylogeny from est data with widespread gene duplication. *BMC Evolutionary Biology*, 7(Suppl 1):S3, 2007.

[126] T. K. Seo. Calculating bootstrap probabilities of phylogeny using multilocus sequence data. *Molecular Biology and Evolution*, 25(5):960–971, 2008.

[127] J. B. Slowinski, A. Knight, and A. P. Rooney. Inferring species trees from gene trees: a phylogenetic analysis of the elapidae (serpentes) based on the amino acid sequences of venom proteins. *Molecular Phylogenetics and Evolution*, 8(3):349–362, 1997.

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

[129] 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 of the United States of America*, 109(37):14942–7, Sept. 2012.

[130] A. Stamatakis. RAxML-VI-HPC: Maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinfomatics*, 22(21):2688–2690, 2006.

[131] M. Steel. Consistency of bayesian inference of resolved phylogenetic trees. *Journal of Theoretical Biology*, 336:246–249, 2013.

[132] U. Stege. Gene trees and species trees: The gene-duplication problem is fixed-parameter tractable. In *Proceedings of the 6th International Workshop on Algorithms and Data Structures (WADS)*, pages 166–173, 1999.

[133] A. Suh, M. Paus, M. Kiefmann, G. Churakov, F. A. Franke, J. Brosius, J. O. Kriegs, and J. Schmitz. Mesozoic retroposons reveal parrots as the closest living relatives of passerine birds. *Nature Communications*, 2:443, 2011.

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

[135] M. Swenson, R. Suri, C. Linder, and T. Warnow. An experimental study of Quartets MaxCut and other supertree methods. *Algorithms for Molecular Biology*, 6(1):7, 2010.

[136] M. S. Swenson, R. Suri, C. R. Linder, and T. Warnow. SuperFine: fast and accurate supertree estimation. *Systematic Biology*, 61(2):214–227, 2012.

[137] D. Swofford. *PAUP*: Phylogenetic analysis using parsimony (and other methods), version 4.0.* Sinauer Associates, 1996.

[138] M. Syvanen. Cross-species gene transfer; implications for a new theory of evolution. *Journal of Theoretical Biology*, 112(2):333–343, 1985.

[139] F. Tajima. Evolutionary relationship of DNA sequences in finite populations. *Genetics*, 105(2):437–460, 1983.

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

[141] C. V. Than and L. Nakhleh. Species tree inference by minimizing deep coalescences. *PLoS Computational Biology*, 5(9):e1000501, 2009.

[142] C. V. Than and N. A. Rosenberg. Consistency properties of species tree inference by minimizing deep coalescences. *Journal of Computational Biology*, 18(1):1–15, 2011.

[143] C. V. Than, D. Ruths, and L. Nakhleh. Phylonet: A software package for analyzing and reconstructing reticulate evolutionary relationships. *BMC Bioinformatics*, 9(1):322, 2008.

[144] D. L. Theobald. A formal test of the theory of universal common ancestry. *Nature*, 465(7295):219–222, 2010.

[145] B. Vernot, M. Stolzer, A. Goldman, and D. Durand. Reconciliation with non-binary species trees. *Journal of Computational Biology*, 15(8):981–1006, 2008.

[146] J. Wakeley. *Coalescent theory: an introduction*, volume 1. Roberts & Company Publishers, 2009.

[147] N. Wang, E. Braun, and R. Kimball. Testing hypotheses about the sister group of the Passeriformes using an independent 30-locus data set. *Molecular Biology and Evolution*, 29(2):737–750, 2012.

[148] T. Warnow. Tree compatibility and inferring evolutionary history. *Journal of Algorithms*, 16(3):388–407, 1994.

[149] T. Warnow, B. M. Moret, and K. St John. Absolute convergence: true trees from short sequences. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 186–195, 2001.

[150] A. Wehe, M. S. Bansal, J. G. Burleigh, and O. Eulenstein. Duptree: A program for large-scale phylogenetic analyses using gene tree parsimony. *American Journal of Botany*, 24(13):1540–1541, 2008.

[151] N. J. Wickett, S. Mirarab, N. Nguyen, T. Warnow, E. Carpenter, N. Matasci, S. Ayyampalayam, M. S. Barker, J. G. Burleigh, M. A. Gitzendanner, B. R. Ruhfel, E. Wafula, J. P. Der, S. W. Graham, S. Mathews, M. Melkonian, D. E. Soltis, P. S. Soltis, N. W. Miles, C. J. Rothfels, L. Pokorny, A. J. Shaw, L. DeGironimo, D. W. Stevenson, B. Surek, J. C. Villarreal, B. Roure, H. Philippe, C. W. dePamphilis, T. Chen, M. K. Deyholos, R. S. Baucom, T. M. Kutchan, M. M. Augustin, J. Wang, Y. Zhang, Z. Tian, Z. Yan, X. Wu, X. Sun, G. K.-S. Wong, and J. Leebens-Mack. Phylotranscriptomic analysis of the origin and early diversification of land plants. *Proceedings of the National Academy of Sciences*, 111(45):E4859–E4868, 2014.

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

[153] Z. Yang, 2015. MCCoal: software available online at http://abacus.gene.ucl.ac.uk/software/MCMCcoal.html.

[154] Y. Yu, T. Warnow, and L. Nakhleh. Algorithms for MDC-based multi-locus phylogeny inference. In *Proceedings of the International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 531–545, 2011.

[155] Y. Yu, T. Warnow, and L. Nakhleh. Algorithms for MDC-based multi-locus phylogeny inference: Beyond rooted binary gene trees on single alleles. *Journal of Computational Biology*, 18(11):1543–1559, 2011.

[156] R. Zardoya and A. Meyer. Complete mitochondrial genome suggests diapsid affinities of turtles. *Proceedings of the National Academy of Sciences*, 95(24):14226–14231, 1998.

[157] L. Zhang. On a Mirkin-Muchnik-Smith conjecture for comparing molecular phylogenies. *Journal of Computational Biology*, 4(2):177–188, 1997.

[158] L. Zhang. From gene trees to species trees II: Species tree inference by minimizing deep coalescence events. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(9):1685–1691, 2011.

[159] B. Zhong, L. Liu, Z. Yan, and D. Penny. Origin of land plants using the multispecies coalescent model. *Trends in Plant Science*, 18(9):492–495, 2013.