

RICE UNIVERSITY

**Inference of Parsimonious Species Phylogenies  
from Multi-locus Data**

by

**Cuong V. Than**

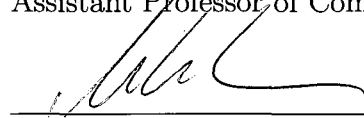
A THESIS SUBMITTED  
IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS FOR THE DEGREE

**Doctor of Philosophy**

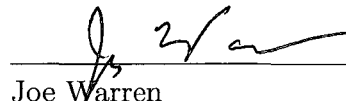
APPROVED, THESIS COMMITTEE:



Luay K. Nakhleh, Chair  
Assistant Professor of Computer Science



Michael Kohn  
Assistant Professor of Ecology and  
Evolutionary Biology



Joe Warren  
Professor of Computer Science

Houston, Texas

January, 2010

UMI Number: 3421316

All rights reserved

**INFORMATION TO ALL USERS**

The quality of this reproduction is dependent upon the quality of the copy submitted.

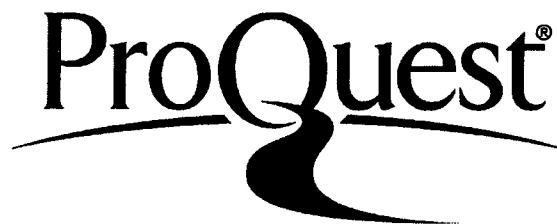
In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI 3421316

Copyright 2010 by ProQuest LLC.

All rights reserved. This edition of the work is protected against unauthorized copying under Title 17, United States Code.



ProQuest LLC  
789 East Eisenhower Parkway  
P.O. Box 1346  
Ann Arbor, MI 48106-1346

## ABSTRACT

### Inference of Parsimonious Species Phylogenies from Multi-locus Data

by

Cuong V. Than

The main focus of this dissertation is the inference of species phylogenies, i.e. evolutionary histories of species. Species phylogenies allow us to gain insights into the mechanisms of evolution and to hypothesize past evolutionary events. They also find applications in medicine, for example, the understanding of antibiotic resistance in bacteria. The reconstruction of species phylogenies is, therefore, of both biological and practical importance.

In the traditional method for inferring species trees from genetic data, we sequence a single locus in species genomes, reconstruct a gene tree, and report it as the species tree. Biologists have long acknowledged that a gene tree can be different from a species tree, thus implying that this traditional method might infer the wrong species tree. Moreover, reticulate events such as horizontal gene transfer and hybridization make the evolution of species no longer tree-like. The availability of multi-locus data provides us with excellent opportunities to resolve those long standing problems. In this dissertation, we present parsimony-based algorithms for reconciling species/gene tree incongruence that is assumed to be due solely to lineage sorting. We also describe a unified framework for detecting hybridization despite lineage sorting.

To address the first problem of species/gene tree incongruence caused by lineage sorting, we present three algorithms. In Chapter 3, we present an algorithm based

on an integer-linear programming (ILP) formula to infer the species tree's topology and divergence times from multiple gene trees. In Chapter 4, we describe two methods that infer the species tree by minimizing deep coalescences (MDC), a criterion introduced by Maddison in 1997. The first method is also based on an ILP formula, but it eliminates the enumeration phase of candidate species trees of the algorithm in Chapter 3. The second algorithm further eliminates the dependence on external ILP solvers by employing dynamic programming. We ran those methods on both biological and simulated data, and experimental results demonstrate their high accuracy and speed in species tree inference, which makes them suitable for analyzing multi-locus data.

The second problem this dissertation deals with is reticulation (e.g., horizontal gene transfer, hybridization) detection despite lineage sorting. The phylogeny-based approach compares the evolutionary histories of different genomic regions and test them for incongruence that would indicate hybridization. However, since species tree and gene tree incongruence can also be due to lineage sorting, phylogeny-based hybridization methods might overestimate the amount of hybridization. We present in this dissertation a framework that can handle both hybridization and lineage sorting simultaneously. In this framework, we extend the MDC criterion to phylogenetic networks, and use it to propose a heuristic to detect hybridization despite lineage sorting. Empirical results on a simulated and a yeast data set show its promising performance, as well as several directions for future research.

## Acknowledgements

First of all, I would like to thank Professor Luay Nakhleh, my advisor. I joined with Professor Nakhleh in the early 2006, and since then I have been introduced by him to so many interesting problems in phylogenetics, as well as in other parts of bioinformatics. By raising interesting problems, he has helped me shape my research in bioinformatics. Luay, thank you for your guidance, for your patience and for so many thoughtful advices and critiques of my work.

I would also like to thank Professor Joe Warren and Professor Michael Kohn for agreeing to be members of my thesis committee, for taking the time to review and evaluate my dissertation, and for their helpful comments on my dissertation, and my presentation and communication skills, as well as advices on my research and career.

Finally, I would like to thank all members in our group. My sincere thanks go to Derek Ruths and Dr. Guohua Jin, with whom I have worked closely. Derek developed basic data structures for the PhyloNet software package, which is part of this dissertation. He has helped me extend and improve the package. Dr. Jin has also collaborated with me on PhyloNet and a number of other projects.

# Contents

Abstract	ii
List of Illustrations	ix
List of Tables	xviii
<b>1 Introduction</b>	<b>1</b>
1.1 Contributions of the Dissertation . . . . .	4
1.2 Outline of the Dissertation . . . . .	5
<b>2 Background</b>	<b>8</b>
2.1 Phylogenetic Trees . . . . .	8
2.1.1 Trees and Phylogenetic Trees . . . . .	8
2.1.2 Clades, Clusters and Cluster Compatibility . . . . .	11
2.1.3 Phylogenetic Tree Comparison . . . . .	12
2.2 Phylogenetic Networks . . . . .	14
2.3 Species Tree, Gene Tree and their Incongruence . . . . .	16
2.3.1 Lineage Sorting . . . . .	17
2.3.2 Hybridization and Horizontal Gene Transfer . . . . .	19
2.4 Methods for Inferring the Species Trees despite Lineage Sorting . . . . .	21
2.4.1 Concatenation . . . . .	22
2.4.2 Consensus Methods . . . . .	23
2.4.3 Democratic Vote . . . . .	24
2.4.4 Maximum Likelihood . . . . .	24
2.4.5 GLASS . . . . .	25
2.4.6 BEST . . . . .	26

2.5	Existing Phylogeny-based HGT Detection Methods . . . . .	27
2.5.1	The T-REX Package . . . . .	27
2.5.2	LatTrans . . . . .	29
2.5.3	HorizStory . . . . .	30
2.5.4	EEEP . . . . .	31
2.5.5	RIATA-HGT . . . . .	32
<b>3</b>	<b>Species Tree Inference from Gene Trees Using Their Topologies and Coalescence Times</b>	<b>37</b>
3.1	Reconciling Gene Trees within Species Trees . . . . .	38
3.2	An ILP-based Method for Inferring Species Trees . . . . .	40
3.2.1	Inferring Species Tree Topology Candidates . . . . .	40
3.2.2	Estimating Species Tree Divergence Times . . . . .	42
3.2.3	Species/Gene Tree Reconciliation and Optimality . . . . .	47
3.3	Empirical Study . . . . .	47
3.3.1	Materials and Analysis . . . . .	47
3.3.2	Results and Discussion . . . . .	52
<b>4</b>	<b>Species Tree Inference from Gene Trees Using Their Topologies Alone</b>	<b>60</b>
4.1	Extra Lineages and Inferring the Species Tree by Minimizing Deep Coalescences . . . . .	61
4.1.1	Extra Lineages . . . . .	62
4.1.2	Inferring the Species Tree under the MDC Criterion . . . . .	65
4.2	Current Methods for Inferring the Species Tree under the MDC Criterion . . . . .	66
4.2.1	Brute-force Algorithm . . . . .	66
4.2.2	Mesquite's Heuristic . . . . .	67

4.3	Counting the Number of Extra Lineages . . . . .	67
4.4	Inferring Species Trees: An ILP Approach . . . . .	69
4.4.1	Constructing the Weighted Compatibility Graph . . . . .	71
4.4.2	Finding the Optimal Tree in the Compatibility Graph . . . . .	72
4.5	Inferring Species Trees: A DP Algorithm . . . . .	73
4.6	Extra Lineages for Non-binary and Multiple-Allele Gene Trees . . . . .	76
4.6.1	Multiple Individuals per Species . . . . .	76
4.6.2	Non-binary Trees . . . . .	77
4.7	Experimental Verification . . . . .	78
4.7.1	Analysis of the <i>Apicomplexan</i> Data Set . . . . .	81
4.7.2	Analysis of the Yeast Data Set . . . . .	83
4.7.3	Analysis of the Synthetic Data . . . . .	85
4.8	Discussions . . . . .	91
<b>5</b>	<b>Detection of Hybridization despite Lineage Sorting</b>	<b>94</b>
5.1	Current Methods for Simultaneous Modeling of Lineage Sorting and Reticulation . . . . .	96
5.1.1	The Method of Than <i>et al.</i> . . . . .	96
5.1.2	The Method of Meng <i>et al.</i> . . . . .	98
5.1.3	The Method of Joly <i>et al.</i> . . . . .	99
5.2	Lineage Sorting in Phylogenetic Networks . . . . .	99
5.3	Experimental Study . . . . .	103
5.3.1	Data . . . . .	103
5.3.2	Results on Simulated Data . . . . .	106
<b>6</b>	<b>PhyloNet</b>	<b>114</b>
6.1	Phylogenetic Network Representation . . . . .	115
6.2	Evolutionary Network Characterization . . . . .	118
6.3	Evolutionary Network Comparison . . . . .	122



6.3.1	Tree-based Comparison . . . . .	125
6.3.2	Cluster-based Comparison . . . . .	126
6.3.3	Tripartition-based Comparison . . . . .	128
6.3.4	Which Measure to Use? . . . . .	128
6.3.5	Parsimony of Evolutionary Networks . . . . .	130
6.4	Inferring Species Trees From Gene Trees . . . . .	132
6.5	Reconstructing Evolutionary Networks from Species Trees and Gene Trees . . . . .	135
6.6	Phylogenetic Tree Utilities . . . . .	138
6.7	Implementation . . . . .	139
6.8	The Command Line Interface . . . . .	140
6.8.1	Programmatic Interface . . . . .	142
6.9	Conclusions . . . . .	142
<b>7</b>	<b>Conclusions</b>	<b>144</b>
7.1	Discussion . . . . .	144
7.2	Future Research . . . . .	145
	<b>Bibliography</b>	<b>148</b>

# Illustrations

1.1	Approaches for inferring species trees. In the combined analysis approach (top), the sequences of the four loci are concatenated, generating one sequence data set, which is then analyzed by any of a host of phylogenetic tree reconstruction methods. In the separate analysis approach (bottom), a gene tree is reconstructed for each locus, and a species tree that reconciles their incongruence is inferred. . . . .	2
2.1	Rooted phylogenetic tree (a) and unrooted phylogenetic tree (b) over 4 taxa $a, b, c$ and $d$ . In Figure (b), we do not say if the parent of $a$ and $b$ and the parent of $c$ and $d$ are from a common ancestor, as in (a). . . . .	9
2.2	Illustration of an SPR operation for unrooted (a) and rooted trees (b). In (a), edge $(u, v)$ is cut, and subtree $t_2$ is regrafted to vertex $u'$ . For rooted trees in (b), an edge $(u, v)$ is also cut, but there are two ways to regraft $t_2$ . . . . .	15
2.3	An illustration of a network (a), and its set of induced trees (b) and (c). . . . .	16
2.4	An illustration of lineage sorting. There are two alleles, one in red and the other in blue, in the ancestral population at time $\tau_2$ . In this figure, the current genes $g_b$ and $g_c$ both derived from the blue gene lineage, resulting a gene tree where $b$ is closer to $c$ than to $a$ . . . . .	19
2.5	Illustration of horizontal gene transfer. Figure (a): the gene tree in thin lines disagree with the species tree shown in tubes, because gene $g_b$ is transferred from $c$ , making in the gene tree $b$ appear closer to $c$ than to $a$ . Figure (b): a phylogenetic network representing HGT. . . . .	21

2.6	Illustration of strict- and majority-consensus and democratic vote methods. Figures (a), (b), and (c) are input gene trees. The strict consensus tree is shown in (d), while (e) shows the majority consensus tree. The species tree of the democratic vote method is either (a), (b) or (c) since for this set of input gene trees, all of them appear with the same frequency $1/3$ . . . . .	22
2.7	Illustration of the GLASS method. Two input gene trees are in (a) and (b). The inferred tree is shown in (c). . . . .	26
2.8	Three possible scenarios where the addition of an HGT branch ( $a, b$ ) can change the minimum distance between two taxa $i$ and $j$ [66]. . . . .	29
2.9	One horizontal gene transfer scenario found by LatTrans for gene $rbcL$ [44].	30
2.10	Edith paths. The species tree is on the left, and the gene tree is on the right. The gene tree differs from the species tree by only one HGT move. Edges $E_0$ and $E_1$ , which induce clusters discordant with the gene tree, form a path. Similarly, edges $E_{(1+3)}$ and $E_{(3+4)}$ form a connected subgraph of the gene tree [45]. . . . .	32
2.11	Illustrating RIATA-HGT. Figures (a) and (b) are the species tree and gene tree, respectively. Figure (c) depicts a scenario to reconcile the species and gene trees [47]. . . . .	33
3.1	Illustration of discordance between species and gene trees. Tree $T'$ is a species tree, and $T_1$ and $T_2$ are two gene trees, which are different from $T'$ . In gene tree $T_1$ , gene lineages $a$ and $b$ coalesce at time $\tau_1$ prior the divergence time $\tau_s$ . Gene tree $T_2$ is different from tree $T'$ topologically. . . . .	38

3.2	Illustration of the first phase in our method. At the top are three gene trees, which are the input to the algorithm. The set of all clusters occurring in these gene trees are then computed, and their compatibility graph is built. Finally, the set of all maximal cliques are computed, and each defines a species tree topology candidate. . . . .	42
3.3	Algorithm ESTIMATEDIVERGENCETIMES. The complete ILP formulation for estimating the divergence times of a species tree topology $T'$ given a set $\mathcal{G}$ of gene trees with times at internal nodes. Solving this ILP yields the divergence time $\tau_v$ , for every node $v$ in the species tree $T'$ . . . . .	48
3.4	Algorithm COMPUTESPECIESTREE( $\mathcal{G}$ ). The algorithm for computing the species tree topology and divergence times from an input set of gene trees with coalescence times at internal nodes $\mathcal{G}$ . . . . .	49
3.5	The Robinson-Foulds (RF) distances between every pair of the 1898 gene trees. RF distance of 0 indicates the two trees are identical, and RF distance of 1 indicates that the two trees do not share any clades in common.	54
3.6	The distributions of coalescence times of all 36 clusters of taxa in the 1898 gene trees, as calculated by Equation (3.13), but without division by $r_s \approx 10^{-8}$ . . . . .	54
3.7	The number of gene tree clades that do not appear in the species tree. Trees 1 to 24 are built from maximal cliques. The first 24 trees are built from the compatibility graph for $\mathcal{G}$ , while trees 25, 26, 27, 28, and 29 are $T_{\text{conc}}$ , $T_{\text{hf}}$ , $T_{\text{avgds}}$ , $T_{\text{avghd}}$ , and $T_{\text{majcons}}$ , respectively. . . . .	56
3.8	The cost of deep coalescences, $\sum_{T \in \mathcal{G}} \sum_{v \in \hat{V}(T)} f_v$ , and the cost of shallow coalescences, $\sum_{T \in \mathcal{G}} \sum_{v \in \hat{V}(T)} g_v$ , for all 29 species tree candidates. The first 24 trees are built from the compatibility graph for $\mathcal{G}$ , while trees 25, 26, 27, 28, and 29 are $T_{\text{conc}}$ , $T_{\text{hf}}$ , $T_{\text{avgds}}$ , $T_{\text{avghd}}$ , and $T_{\text{majcons}}$ , respectively. . . . .	58

3.9	Species trees with times assigned by Algorithm 3.3. The lengths of the “shortened” branches were divided by $10^5$ , so that the resolution of the trees can be shown clearly. . . . .	59
4.1	Illustration of the concept of extra lineages. We are given a gene tree $(a, (b, (c, d)))$ . Then, tree $T_1 = (((d, b), c), a)$ requires one extra lineage to reconcile the gene tree within its branches, while tree $T_2 = (((a, d), b), c)$ requires three extra lineages. . . . .	62
4.2	Fitting a gene tree $T$ into a species tree $T'$ . In the figure, only mappings of internal nodes of $T$ are shown, as each leaf in $T$ is mapped to a leaf with the same label in $T'$ . . . . .	64
4.3	Algorithm COMPUTEOPTIMALTREE. An approach to find the optimal tree for a set of gene trees $\mathcal{G}$ . Note that all (nonempty) clusters are used to find the optimal tree. . . . .	69
4.4	Compatibility graph constructed from three gene trees $T_1, T_2$ , and $T_3$ . A maximum vertex-weighted clique consisting of clusters $\{bc\}, \{abc\}, \{de\}, \{def\}$ is highlighted. . . . .	70
4.5	Algorithm DP-SPECIESTREEINFERENCE. . . . .	75
4.6	MDC for gene trees with multiple alleles/individuals. On the left, the species tree is shown in tubes, while the thin lines show how the gene tree, on the right, is fitted within the branches of the species tree. On the right, a gene tree with four leaves, two of which correspond to two individuals of species $A$ . . . . .	77
4.7	MDC for non-binary trees. On the left, the species tree is shown in tubes, while the thin lines show how the non-binary gene tree, on the right, is fitted within the branches of the species tree. . . . .	78

- 4.8 The species tree for the *Apicomplexan* data as inferred using the majority consensus method and reported in [29]. The species *Tt* (*Tetrahymena thermophila*) is the outgroup. The numbers on the tree branches are bootstrap support values based on maximum likelihood, maximum parsimony and neighbor joining methods, respectively. . . . . 79
- 4.9 The species tree for the yeast data set as inferred using the concatenation method and reported in [25]. All branches in the tree have 100% support values. . . . . 80
- 4.10 (a) The optimal (species) tree inferred by our method for the *Apicomplexan* data set; this tree requires 440 deep coalescences to reconcile all 268 gene trees. The two sub-optimal species trees with 469 and 542 deep coalescences are shown in (b) and (c), respectively. The value on each branch is the numbers of extra lineages within that branch, when reconciling all 268 gene trees. . . . . 81
- 4.11 Plot of the number of extra lineages for each of the binary (fully resolved) 247 species tree candidates identified as maximal cliques in the compatibility graph of the gene trees. The first three lowest values are 440, 469 and 542. The trees corresponding to these numbers are shown in Figure 4.10, respectively. . . . . 83
- 4.12 (a) The species tree inferred by our method for the yeast data set. The values on its branches are the numbers of extra lineages within them. (b) Plot of the number of extra lineages for all 48 species tree candidates . . . 84
- 4.13 The six best sub-optimal trees for the yeast data set. These trees, from left to right and top down, have in total 134, 163, 170, 186, 191 and 193 extra lineages. The values on the branches are the numbers of extra lineages within them. . . . . 85

- 4.14 Accuracy of the inferred species tree as measured by the Robinson-Foulds distance when all clusters (there are  $2^8 - 1 = 255$  of them) are used.  
 (a) Recent divergence (total branch length is  $1N_e$ ); (b) Deep divergence (total branch length is  $10N_e$ ). We note that the  $y$ -axes in (a) and (b) are on different scales to make the difference between the curves more visible. . . . . 86
- 4.15 Average numbers of extra lineages required to reconcile the inferred species tree and gene trees when all clusters (there are  $2^8 - 1 = 255$  of them) are used for the inference. (a) Recent divergence (total branch length is  $1N_e$ ); (b) Deep divergence (total branch length is  $10N_e$ ). We note that the  $y$ -axes in (a) and (b) are on different scales to make the difference between the curves more visible. . . . . 88
- 4.16 Average rates of species tree clusters that do not appear in any gene trees. (a) for data with recent divergence (total branch length is  $1N_e$ ); (b) for data with deep divergence (total branch length is  $10N_e$ ). We note that the  $y$ -axes in (a) and (b) are on different scales to make the difference between the curves more visible. . . . . 89
- 4.17 Accuracy of the inferred species tree as measured by the Robinson-Foulds distance when only clusters induced by gene trees are used. (a) for data with recent divergence (total branch length is  $1N_e$ ); (b) for data with deep divergence (total branch length is  $10N_e$ ). We note that the  $y$ -axes in (a) and (b) are on different scales to make the difference between the curves more visible. . . . . 90
- 4.18 Average numbers of extra lineages required to reconcile the inferred species tree and gene trees when only clusters induced by genes trees are used for the inference. (a) for data with recent divergence (total branch length is  $1N_e$ ); (b) for data with deep divergence (total branch length is  $10N_e$ ). We note that the  $y$ -axes in (a) and (b) are on different scales to make the difference between the curves more visible. . . . . 90

4.19	Average numbers of clusters induced by gene trees, excluding single-element and all-element clusters. Note that the total number of nonempty clusters is $2^8 - 1 = 255$ , as there are eight species. (a) Recent divergence (total branch length is $1N_e$ ); (b) Deep divergence (total branch length is $10N_e$ ). We note that the $y$ -axes in (a) and (b) are on different scales to make the difference between the curves more visible. . . . .	91
4.20	A counterexample where the optimal tree cannot be built from gene tree clusters. (a) is the input gene trees. (b) is compatibility graph built from clusters induced by gene trees in (a), where the maximal clique with the smallest weight is highlighted. (c) is another tree that requires a fewer number of extra lineages to reconcile three trees in (a). . . . .	93
5.1	A three bacterial species model with an HGT event. In (a), there is no lineage sorting, and hence resulting a gene tree that is different from the species tree. In (b), there is a deep coalescence event between the lineage in $b$ and the lineage transferred to $c$ , making the gene tree congruent (topologically) with species tree. . . . .	97
5.2	The hybrid speciation model. The network is shown on the left, and the two induced trees are shown in the right. In this network, a gene in $b$ is either from $a$ with probability $\gamma$ , or from $c$ with probability $1 - \gamma$ . . . . .	98
5.3	Two gene trees that differ in the placement of $b$ . . . . .	100
5.4	An optimal tree and an optimal network for the two gene trees in Figure 5.3. (a) An optimal species tree under the MDC criterion, which requires a single deep coalescence event to reconcile the two gene trees of Figure 5.3. (b) A phylogenetic network that requires <i>no</i> deep coalescence events to reconcile both gene trees of Figure 5.3. . . . .	102



- 5.5 Simulation scenario. A hundred gene trees were simulated under the coalescent model within the branches of the network  $N$  by evolving  $(1 - \frac{1}{2}e^{-t_2})\gamma$  of them within the branches of  $T_1$ ,  $(1 - \frac{1}{2}e^{-t_2})(1 - \gamma)$  within the branches of  $T_2$ ,  $\frac{1}{4}e^{-t_2}$  within the branches of  $T_3$  and  $\frac{1}{4}e^{-t_2}$  within the branches of  $T_4$ . Times are given in coalescent units (number of generations divided by population size). . . . . 104
- 5.6 (a) The single optimal tree under the MDC criterion for the data set. The number of extra lineages resulting from reconciling all 106 gene trees within the branches of this tree is 127. (b) The best sub-optimal tree under the MDC criterion. The number of extra lineages resulting from reconciling all 106 gene trees within the branches of this tree is 134, which is just 7 extra lineages away from the optimal value of 127 achieved by the tree in (a). The number on a branch indicates the number of extra lineages along that branch once all 106 gene trees are reconciled within the branches of the tree. . . . . 106
- 5.7 Average numbers extra lineages for each of the constituent trees. The  $x$ -axis lists the 15 possible (rooted) tree topologies on the four taxa, and the  $y$ -axis denotes the number of extra lineages resulting from reconciling all 100 gene trees within each of the 15 trees. Left: the “easy” case of very long times; right: the “hard” case of very short times. . . . . 107
- 5.8 Optimal trees vs. optimal networks. The  $x$ -axis shows the values of  $\gamma$ , and the  $y$ -axis shows the number of extra lineages. The optimal value for network is computed by exhaustively considering *all* networks on 4 taxa. . . 109
- 5.9 Three hybridization scenarios for the yeast data set. Each of the networks requires 69 extra lineages to reconcile all 106 gene trees, and depicts a slightly different hybridization scenario. The number on a branch indicates the number of extra lineages along that branch once all 106 gene trees are reconciled within the branches of the network. . . . . 113

- 6.1 Two evolutionary networks  $N_1$  and  $N_2$ , each with eight leaves (labeled  $a, \dots, h$ ) and two network nodes  $W$  and  $Z$ . Shown are the orientation of the network edges; all other edges are directed away from the root (toward the leaves) Notice that the difference between the two networks is that node  $W$  in  $N_1$  has lineage  $g$  as one of its parents, whereas node  $W$  in  $N_2$  has lineage  $h$  as one of its parents. . . . . 115
- 6.2 Three trees,  $N'$ ,  $W$ , and  $Z$ , along with their Newick representation. These trees form the tree decomposition  $\mathcal{F}$  of the phylogenetic network  $N_1$  in Figure 6.1. The eNewick representation of  $N$  is the triplet  $\langle N'; W; Z \rangle$ . . . 116
- 6.3 A modified Newick format for representing phylogenetic networks. This example is from [106]. . . . . 116
- 6.4 The sets  $\mathcal{T}(N_1) = \{T_1^1, T_1^2, T_1^3, T_1^4\}$  and  $\mathcal{T}(N_2) = \{T_2^1, T_2^2, T_2^3, T_2^4\}$  of all eight trees induced by the two networks  $N_1$  and  $N_2$ , respectively, in Figure 6.1. . . . . 120
- 6.5 Illustration of the tree-based network comparison measure. (a) The weighted bipartite graph  $G$  that is constructed from the two networks  $N_1$  and  $N_2$  in Figure 6.1. On the left are four nodes that correspond to the four trees in  $\mathcal{T}(N_1)$  and on the right are four nodes that correspond to the four trees in  $\mathcal{T}(N_2)$ . The weight of an edge between  $T_1^i$  and  $T_2^j$  is the values of the Robinson-Foulds (RF) distance between the two trees, which is computed as the number of clusters present in one but not both of the trees, divided by 2. (b) The edges that comprise the minimum-weight edge cover of the bipartite graph  $G$ . The weight of this cover is 2, which is the sum of the weights of the edges in the cover; therefore,  $m^{tree}(N_1, N_2) = 2$ . 127
- 6.6 (a) Screen captures of the graphical output of RIATA-HGT on the pair of trees  $((a, b), c), (d, (e, f))$  and  $((a, c), b), ((d, f), e)$ . (b) The eNewick representations of the two selected networks. . . . . 137

# Tables

3.1	Information of nine strains of the <i>Staphylococcus aureus</i> bacteria. . .	49
4.1	The number of extra lineages for each of seven clusters induced by gene trees $T_1$ , $T_2$ , and $T_3$ in Figure 4.4. The last column is the weight assigned to vertices in the compatibility graph according to Equation 4.4, where $m = 2$ . . . . .	72
6.1	A table of the (nontrivial) clusters of the two networks $N_1$ and $N_2$ in Figure 6.1, denoted by $\mathcal{C}(N_1)$ and $\mathcal{C}(N_2)$ , respectively, in the text. Highlighted are rows corresponding to clusters that differ between the two networks. . . . .	121
6.2	A table of the (nontrivial) tripartitions of the two networks $N_1$ and $N_2$ in Figure 6.1, denoted by $\theta(N_1)$ and $\theta(N_2)$ , respectively, in the text. Highlighted are rows corresponding to tripartitions that differ between the two networks. . . . .	123
6.3	A table of the tools currently implemented in PhyloNet. With the exception of the three phylogenetic trees tools <code>lca</code> , <code>mast</code> , and <code>rf</code> , all the other tools are for analyzing reticulate evolutionary relationships. . . . .	141

# Chapter 1

## Introduction

The study of inferring phylogenies, or evolutionary histories of species, started when Charles Darwin published his famous book, “On the Origin of Species by Means of Natural Selection,” where he realized and presented a hypothesis that all species have evolved from a common ancestor. Phylogenies, which are often represented by trees, allow us to gain insights into the mechanisms of evolution and to hypothesize past evolutionary events. Traditionally, a phylogeny was inferred by using morphological features. Since the 1960s when amino acid sequences were first widely available [1], molecular data have become the main source for phylogenetic analysis [2].

However, since the early days of molecular phylogenetics, researchers already noted the difference between a phylogeny of species (a species tree) and a phylogeny of a gene (a gene tree) [3, 4]. A gene tree can be different from a species tree (on the same group of species) for various reasons. First, we do not know the true gene tree, and therefore the estimated one that we build from molecular data might be incorrect due to both random and phylogeny reconstruction errors. Second, there are biological processes such as lineage sorting, horizontal gene transfer, and gene duplication and loss that cause species/gene tree incongruence [5]. They also recognized that ultimately we are concerned with reconstructing species trees rather than gene trees. Tateno *et al.* [6] stated that “the primary objective of molecular taxonomy or phylogenetics is to construct a species trees rather than a gene tree,” and that we can achieve higher accuracy in species tree estimation only by using more gene trees.

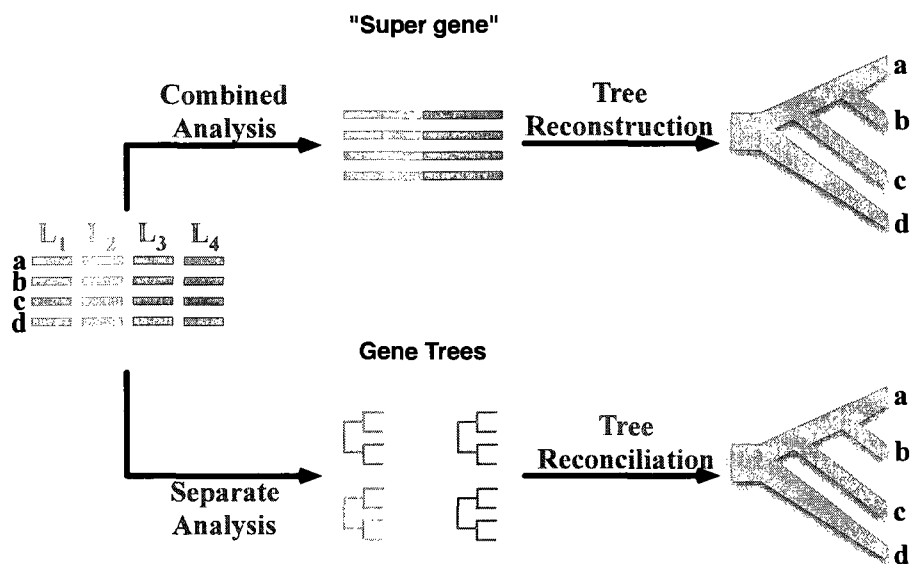


Figure 1.1 : Approaches for inferring species trees. In the combined analysis approach (top), the sequences of the four loci are concatenated, generating one sequence data set, which is then analyzed by any of a host of phylogenetic tree reconstruction methods. In the separate analysis approach (bottom), a gene tree is reconstructed for each locus, and a species tree that reconciles their incongruence is inferred.

The first genome to be sequenced was that of the bacteriophage virus  $\phi$ X174; it was sequenced in 1977 [7]. The genome of a bacterium (*Haemophilus influenzae*) was first sequenced in 1995 [8], and a eukaryotic genome (*Saccharomyces cerevisiae*) was sequenced in 1997 [9]. A decade later, there were 543 sequenced genomes for eubacteria, 47 for archaeal species and 23 for eukaryotes [10]. The availability of whole-genome data provides an unprecedented opportunity for studying organismal evolutionary relationships, while it also poses computational and methodological challenges. We discuss briefly here current methods in phylogeny inference based on genome data (or phylogenomic tree inference for short).

Broadly speaking, methods for inferring phylogenomic trees fall into two categories: (a) methods that use information above the sequence level and (b) primary

sequence-based methods [11]. Methods in the first category use whole-genome features such as frequencies of oligonucleotides or oligopeptides [12], gene order [13, 14, 15], or gene content [16, 17, 18]. These methods are clearly better than the traditional approach of equating a species trees to a single gene tree, because in general gene trees can disagree with their containing species tree for various causes. Several researchers, in fact, gave them strong support [19, 20]. However, they do have their shortcomings. For example, methods based on the distribution of oligonucleotides or oligopeptides have no model of evolution, while gene-order methods are computationally expensive as the search space is huge [21], and those using gene content are affected by big/small gene attraction [22], producing phylogenies conflicting with previous studies [23].

Two approaches mainly used in primary sequence-based methods are: (a) total evidence (or combined analysis) and (b) separate analysis; see Figure 1.1. In the combined analysis approach, sequences from multiple loci are concatenated, and the resulting “supergene” data set is analyzed, using traditional phylogenetic methods such as maximum parsimony and maximum likelihood; e.g., [24, 25]. Although combined analysis methods are preferred in practice [26, 27, 25], we should note that “no rational systematist would suggest combining genes with different histories to produce a single reconstruction” [28].

In the separate analysis approach, the sequence data from each locus is first analyzed individually, and a reconciliation of the gene trees is then sought. One way to reconcile the gene trees is by taking their majority consensus [19, 29]. Another way is the “democratic vote” method, which entails taking the tree topology occurring with the highest frequency among all gene trees as the species tree. Shortcomings of those methods have been analyzed by various researchers [30, 31]. Recently, Bayesian methods following the separate analysis approach were developed [32, 33]. While

Bayesian methods are accurate, they are very time consuming, taking hours and days even on moderate-size data sets, which limits their scalability.

## 1.1 Contributions of the Dissertation

The first contribution of this dissertation is algorithms for inferring species trees from input gene trees despite lineage sorting. Our first algorithm infers the species tree topology and its branch lengths by seeking a tree that minimizes the amount of incongruence and deep coalescence required to reconcile the input gene trees within the species tree. It divides the inference process into three phases. In the first phase, it computes a set of species tree topology candidates based on clusters (i.e., subsets of a taxon set) induced by the input gene trees. In the second phase, it assigns divergence times to the internal nodes of each of those tree candidates based on an integer-linear programming (ILP) formulation such that the time assignment results in the least amount of deep coalescence. Among those time-assigned trees, the optimal tree is chosen during the third phase, which is then reported as the species tree.

The other two algorithms infer the species tree topology using the minimizing deep coalescence (MDC) criterion. This criterion was introduced in 1997 in a paper by Maddison [5], but so far there have been only approximation heuristics for it, e.g., [34]. They are also slow as they employ a strategy that basically performs a hill-climbing search in the space of all phylogenetic trees (on the same set of taxa). We show that under the MDC criterion, it is possible to work with clusters to find an optimal tree. This allows us to develop our second ILP-based algorithm to infer the species tree that avoids the enumeration of species tree topology candidates required in the first method. Furthermore, it also allows us to develop an efficient dynamic programming algorithm, thus eliminating the dependence on ILP solvers, and more

importantly making it more applicable to large data sets. Those two algorithms are the first exact solutions for the MDC criterion, and we hope that their introduction would help to have a more comprehensive evaluation of this criterion in species tree inference despite lineage sorting.

The other main contribution of this dissertation is about detection of reticulation (e.g., horizontal gene transfer, hybridization) despite lineage sorting. The phylogeny-based approach compares the evolutionary histories of different genomic regions and test them for incongruence that would indicate hybridization. However, the species/gene tree incongruence can also be due to other factors, such as lineage sorting[5], which implies that phylogeny-based hybridization methods might overestimate the amount of hybridization. We present in this dissertation a framework that can handle both hybridization and lineage sorting simultaneously. In this framework, we extend the MDC criterion introduced in [5] to phylogenetic networks. Under this new criterion, we propose that the optimal network consists of the optimal tree and sub-optimal trees within a threshold of the optimal tree's score.

## 1.2 Outline of the Dissertation

Below is a summary of the chapters of this dissertation.

Chapter 2 provides a brief review of phylogenetic trees and phylogenetic networks and related concepts such as clades, clusters and compatibility of clusters. It then describes lineage sorting, and horizontal gene transfer and hybridization, biological processes that cause species/gene tree incongruence and that are of the main concern of this dissertation. An overview of current methods for inferring species trees despite lineage sorting then follows. As discussed in the previous section, a major contribution of this dissertation is a unified framework for detecting hybridization despite



lineage sorting. In this chapter, we also describe several phylogeny-based methods for detecting hybridization. We defer a discussion of recent attempts to incorporate lineage sorting into hybridization detection until Chapter 5.

Chapter 3 is about our first algorithm for inferring species trees from a set of gene trees despite lineage sorting. We first show, with the time information on both species and gene trees, how we reconcile them, and introduce a weighting scheme to measure the amount of deep coalescence, based on the “depth” of a coalescence event. We then discuss the algorithm, which is divided into three phases. In the first phase, we compute clusters induced by the the input gene trees, and build a graph based on those clusters and their compatibility. Because of the equivalence of a set of compatible clusters and a tree, maximal cliques in this graph result in species tree topology candidates, which are the input for the second phase of the algorithm. The second phase assigns divergence times to internal nodes of each of those candidate trees in such a way that the resulting tree requires the minimum cost of deep coalescence. This phase is solved by using an ILP formulation. Finally, we describe an optimality criterion that combines deep coalescence and species/gene tree incongruence for reporting the species tree. In the third phase, our algorithm chooses among the timed trees output from the second phase an optimal one that it declares as the species tree.

In Chapter 4, we describe the inference of species trees from gene trees under the MDC criterion. Under this criterion, we fit a gene tree into a species tree using the most recent common ancestor (MRCA) mapping, and then count the number of extra lineages in all branches of the species tree. A species tree is better than another if it requires a fewer number of extra lineages. We show that we can compute the number of extra lineages for each individual species tree cluster, thus eliminating the

need of prior knowledge of the species tree. This result is fundamental to the two algorithms we present in this chapter. First of all, it makes the phase of generating species tree topology candidates unnecessary. This, along with the equivalence of a set of compatible clusters and a tree, allows us to develop an elegant ILP solution to the MDC optimization problem. We further exploit this result to develop an efficient dynamic programming algorithm for this problem. Details of those algorithms and their performance study are discussed in this chapter.

We describe our framework for detecting hybridization despite lineage sorting in Chapter 5. An extension to the MDC criterion originally introduced for trees is made to take into account reticulate events in phylogenetic networks. We propose that the MDC cost for reconciling a gene tree within a network is the minimum MDC score for reconciling that gene tree with trees induced by the network. Using this extension, we propose a new heuristic to detect hybridization despite lineage sorting. Empirical results on the yeast data set [25] as well as on simulated data show promising performance of the method, as well as several directions for future research.

We discuss PhyloNet [35] in Chapter 6, a package that implements all the algorithms presented in this dissertation. In addition, the package presents a new format for representing phylogenetic networks. It also implements an array of methods for characterizing and comparing phylogenetic networks, as well as those for working with phylogenetic trees.

## Chapter 2

### Background

In this chapter, we introduce concepts and definitions relevant to this dissertation. After a review of the terminology of phylogenetic trees and phylogenetic networks, we discuss several biological processes that cause species tree and gene tree incongruence: lineage sorting, and horizontal gene transfer and hybridization. We also give a brief overview of methods for inferring the species tree from multiple gene trees whose incongruence is assumed to be due to lineage sorting. We conclude this chapter with phylogeny-based methods for detecting horizontal gene transfer.

#### 2.1 Phylogenetic Trees

##### 2.1.1 Trees and Phylogenetic Trees

The evolutionary history of a group of species is often depicted in the form of a tree (in the formal sense in computer science), called a species tree. Each internal node in the tree reflects a speciation event that splits the group into smaller subgroups, and leaves can be thought of as representing present-day organisms. As species evolve, their genes evolve, and when species are split, their gene copies are also split. Therefore, the evolution of a gene is likewise represented by a tree, called a gene tree. Species and gene trees are commonly called phylogenetic trees.

A tree  $T = (V, E)$  is a connected graph with no cycles, where  $V$ ,  $E$  are its node set and edge set (we also use  $V(T)$  and  $E(T)$  to denote the node set and edge set of a

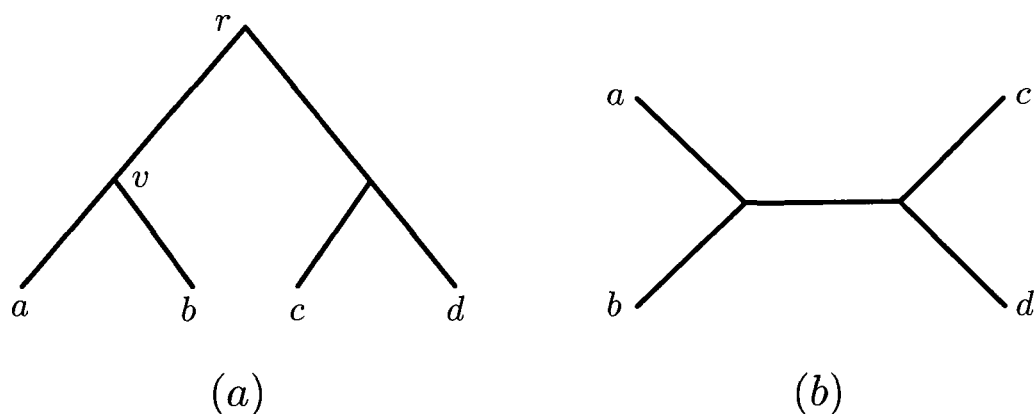


Figure 2.1 : Rooted phylogenetic tree (a) and unrooted phylogenetic tree (b) over 4 taxa  $a, b, c$  and  $d$ . In Figure (b), we do not say if the parent of  $a$  and  $b$  and the parent of  $c$  and  $d$  are from a common ancestor, as in (a).

tree  $T$ ). A node with degree (the number of incident edges) one is called a leaf, and a node of degree at least two is called an internal node. Let us denote by  $\mathcal{L}(T)$ ,  $\mathring{V}(T)$  the set of leaves, and the set of internal nodes of  $T$ , respectively. Let  $X$  be a set of taxa (i.e., species names). Then, a *phylogenetic tree* is an ordered pair  $(T, \phi)$ , where  $\phi$  is a one-to-one correspondence mapping from  $X$  to  $\mathcal{L}(T)$  (i.e., it maps each taxon to one and only one leaf of  $T$ ); see Figure 2.1 for examples of phylogenetic trees. For the sake of brevity, in this dissertation we often call  $T$  a phylogenetic tree when the mapping  $\phi$  is obvious from the context.

A phylogenetic tree can be rooted or unrooted. A tree is rooted if there is a distinguished node, called the root, with in-degree 0 (i.e., there are no edges incident into it). For a rooted tree  $T$ , we denote such a node by  $r(T)$ . In a rooted phylogenetic tree, the root corresponds to the common ancestor of all species or genes at its leaves. A rooted phylogenetic tree, therefore, shows not only the relative relationships of species, but also the direction of the evolution, from its root down to its leaves. An

unrooted phylogenetic tree, on the other hand, only shows the relationship among species. Figure 2.1(a) shows an example of a rooted phylogenetic trees, while Figure 2.1(b) is an example of an unrooted phylogenetic tree. In this dissertation all trees are rooted, unless explicitly stated.

An edge of a tree  $T$  is called a pendant edge if it is incident to a leaf, and an internal edge otherwise. If every internal node of a binary, rooted phylogenetic tree  $T$  has exactly two children, we say that  $T$  is binary. (If  $T$  is an unrooted tree, then it is binary if all internal nodes have degree three.) It is easy to see that for a rooted, binary phylogenetic tree  $T$  on an  $n$ -element taxon set  $X$ , there are exactly  $n$  pendant edges,  $n - 2$  internal edges, and  $n - 1$  internal nodes. The following result is also well known; its proof can be found in [36].

**Theorem 2.1** (Number of Binary Phylogenetic Trees). *Let  $X$  be a set of  $n$  taxa. Then, the number of binary, unrooted phylogenetic trees on  $X$  is  $(2n - 5)!!$ , and the number of binary, rooted phylogenetic trees on  $X$  is  $(2n - 3)!!$ .*

A rooted phylogenetic tree can be represented in computer-readable form, known as the Newick format [37]. This format represents a tree by making use of parentheses and commas. For example, the tree in Figure 2.1(a) is written in the Newick format as  $((a, b), (c, d))$ . We can also write an unrooted tree in the Newick format. First, we arbitrarily root it, and write the resulting rooted tree in the Newick format. Then, we add a prefix, say,  $[U]$ , to the Newick representation. For example, a Newick representation for the unrooted tree in Figure 2.1(b) can be  $[U] (a, (b, (c, d)))$  (here, we root it on the pendant edge incident to  $a$ , and add a prefix  $[U]$  to the rooted tree's Newick string representation).

For a phylogenetic tree  $T$ , we can also associate it with a time function  $\tau: V(T) \rightarrow \mathbb{R}^+ \cup \{0\}$  to indicate divergence times of its internal nodes. In this dissertation, we

use the conventions that if  $v$  is a leaf then  $\tau(v) = 0$ , and that if  $u$  is an ancestor of  $v$ ,  $u \neq v$ , then  $\tau(u) > \tau(v)$ .

### 2.1.2 Clades, Clusters and Cluster Compatibility

Let  $u$  be a node of a rooted tree  $T$ . A node  $v$  is a descendant of  $u$  if  $u$  is on the (unique) path from  $r(T)$  to  $v$ . We also say that  $u$  is an ancestor of  $v$ . Note that  $u$  is both an ancestor and descendant of itself.

A subtree of  $T$  is a connected subgraph of  $T$ . For a node  $v$  of  $T$ , the subtree of  $T$  rooted at  $v$ , or a clade induced by  $v$ , denoted by  $T(v)$ , is the connected subgraph of  $T$  on the set of descendants of  $v$ .

A cluster is defined as a nonempty subset of an  $n$ -taxon set  $X$ . Clearly, there are  $2^n - 1$  clusters for a given taxon set  $X$ . A cluster is called trivial if it is either  $X$  or it has exactly one element. For a node  $v$  of a (rooted) phylogenetic tree  $T$  on  $X$ , the label set of  $\mathcal{L}(T(v))$  is called an induced cluster, denoted by  $C_T(v)$ . For the rooted tree in Figure 2.1(a),  $T(v) = (a, b)$ , and  $C_T(v) = \{a, b\}$ . If  $T$  is binary, then there are  $2n - 1$  induced clusters,  $(n - 2)$  of which are nontrivial. We denote by  $\mathcal{C}(T)$  the set of all nontrivial clusters induced by  $T$ . For example, for the tree in Figure 2.1(a), the set  $\mathcal{C}(T)$  is  $\{\{a, b\}, \{c, d\}\}$ .

Given two clusters  $A$  and  $B$  of a taxon set  $X$ , we say that they are compatible if either  $A \subseteq B$ ,  $B \subseteq A$ , or  $A \cap B = \emptyset$ . Informally, we say that  $A$  and  $B$  are compatible if there exists a rooted phylogenetic tree such that it induces both  $A$  and  $B$ . If none of the three conditions hold, we say that  $A$  and  $B$  are incompatible. As an example, clusters  $\{a, b\}$  and  $\{c, d\}$  are compatible since they are both induced clusters of the tree in Figure 2.1(a). On the other hand,  $\{a, b\}$  and  $\{a, c\}$  are incompatible since all the they do not satisfy all three conditions above. A set of clusters is called pairwise

compatible if every pair of member clusters is compatible. The following theorem shows the relationship between phylogenetic trees and cluster compatibility.

**Theorem 2.2** ([38, 36]). *A nonempty set of pairwise compatible clusters uniquely defines a tree, and vice versa.*

Finally, for a (nonempty) cluster  $A$  of  $X$ , we call a node  $v$  of a rooted phylogenetic tree  $T$  on  $X$  the most recent common ancestor of  $A$  in  $T$ , denoted by  $\text{MRCA}_T(A)$ , if: (1)  $A \subseteq C_T(v)$ ; and (2) for any descendant  $w$  of  $v$ ,  $w \neq v$ ,  $A \not\subseteq C_T(w)$ .

### 2.1.3 Phylogenetic Tree Comparison

In this subsection, we review some common measures for comparing phylogenetic trees. We discuss the Robinson-Foulds distance [39] and the SPR (subtree prune and regraft) distance.

#### Robinson-Foulds distance

For two rooted phylogenetic trees  $T_1$  and  $T_2$ , we define their Robinson-Foulds distance as follows:

$$d_{\text{RF}}(T_1, T_2) = |\mathcal{C}(T_1) \setminus \mathcal{C}(T_2)| + |\mathcal{C}(T_2) \setminus \mathcal{C}(T_1)|. \quad (2.1)$$

It is easy to see that  $d_{\text{RF}}(T_1, T_2) = 0$  if and only if  $T_1$  and  $T_2$  are identical since a tree is uniquely defined by the set clusters it induces. The distance is also symmetric, by definition of symmetric set difference. We note that  $|A \setminus B| + |B \setminus A| = |A| + |B| - 2|A \cap B|$ , and therefore to prove that  $d_{\text{RF}}$  satisfies the triangle inequality, we need to

---

\*Robinson and Foulds in their paper [39] defines the distance for unrooted tree. The definition for rooted trees given here follows their definition for unrooted trees.

show that

$$|\mathcal{C}(T_1) \cap \mathcal{C}(T_2)| + |\mathcal{C}(T_3) \cap \mathcal{C}(T_2)| \leq |\mathcal{C}(T_2)| + |\mathcal{C}(T_1) \cap \mathcal{C}(T_3)|, \quad (2.2)$$

for all phylogenetic trees  $T_1, T_2, T_3$  on the same taxon set  $X$ . However, every cluster  $A \in \mathcal{C}(T_2)$  appears at least once in the right-hand side of Equation (2.2). If it appears twice on the left-hand side of this equation, then  $A$  must be in both  $\mathcal{C}(T_1), \mathcal{C}(T_3)$ , and hence in  $\mathcal{C}(T_1) \cap \mathcal{C}(T_3)$ , which means that it also appears twice in the right-hand side of the equation. Therefore, Equation (2.2) holds for all  $T_1, T_2$ , and  $T_3$ , and  $d_{\text{RF}}$  is a distance measure.

The normalized Robinson-Foulds distance is defined as

$$\tilde{d}_{\text{RF}}(T_1, T_2) = \frac{1}{2} \times \left( \frac{|\mathcal{C}(T_1) \setminus \mathcal{C}(T_2)|}{|\mathcal{C}(T_1)|} + \frac{|\mathcal{C}(T_2) \setminus \mathcal{C}(T_1)|}{|\mathcal{C}(T_2)|} \right), \quad (2.3)$$

which is always between 0 and 1, inclusively; a distance of zero means two trees are identical, while a distance of one means they are completely different, i.e., they have no induced clusters in common.

### Subtree prune and regraft (SPR) distance

Let  $T$  be an unrooted binary phylogenetic tree on  $X$ ,  $|X| \geq 3$ , and let  $e = (u, v)$  be an edge of  $T$ . By deleting edge  $e$  we obtain two connected subtrees  $t_1$  to whom  $u$  belongs and  $t_2$  to whom  $v$  belongs. We can suppose that  $t_1$  has at least two leaves (since  $|X| \geq 3$ ), and hence there exists another edge  $e' \in E(t_1)$ . We add a new vertex to subdivide  $e'$  and add a new edge between it and  $v$ , and suppress<sup>†</sup> all 2-degree vertices. The new tree is said to be obtained from  $T$  by an SPR operation. See Figure 2.2(a) for an illustration.

---

<sup>†</sup>Suppressing a 2-degree node  $v$  means that we delete two edges  $(u, v)$  and  $(v, w)$  and then adjoin  $u$  with  $w$  by a new edge.



It has been proved that for any pair of unrooted binary trees, one can always be reached from the other by applying a sequence of SPR operations [40, 41]. The SPR distance between two unrooted trees is defined as the minimum number of SPR operations required to transform one to the other. The problem of computing this distance is NP-hard [42].

For rooted binary trees, an SPR operation can be defined in a similar way, except that we also allow for creating a new root and adjoining an edge between it and  $v$  in order to make the rooted SPR distance a metric [43]; see Figure 2.2(b) for an example.

The importance of the rooted SPR distance comes from the fact that it can be used to simulate a horizontal gene transfer (HGT) event (see Section 2.3 for more detail). However, the problem of computing the rooted SPR distance between two rooted binary trees is NP-hard [43]. There are a number of heuristics that compute this distance, for example, LatTrans [44], EEEP [45], HorizStory [46], RIATA-HGT [47, 48]. Recently, there is an integer linear programming (ILP)-based algorithm that computes the exact rooted SPR distance [49].

## 2.2 Phylogenetic Networks

The evolutionary history of a group of species is not always tree-like. When biological processes such as hybridization and horizontal gene transfer occur, it might be more appropriate to represent the evolution of species by a phylogenetic network (see Section 2.3 for more detail). A (rooted) phylogenetic network is a rooted directed acyclic graph (or rooted DAG for short)  $N = (V, E)$  whose leaves are labeled with labels from a taxon  $X$  by a bijective function  $\phi$ . As with phylogenetic trees, we call  $N$  a phylogenetic network when the labeling function is clear from the context.

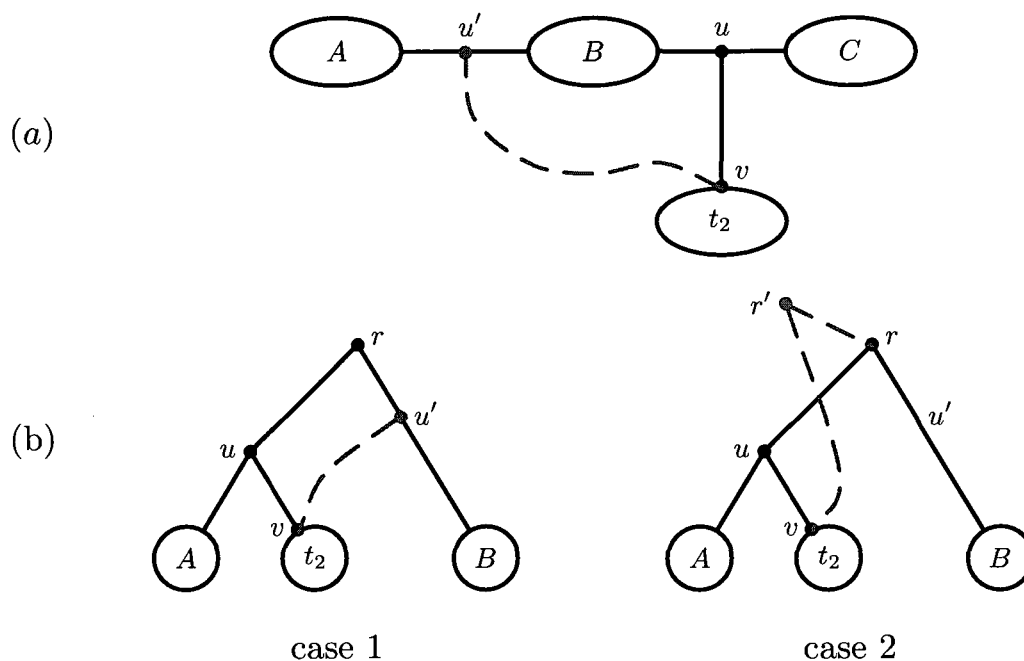


Figure 2.2 : Illustration of an SPR operation for unrooted (a) and rooted trees (b). In (a), edge  $(u, v)$  is cut, and subtree  $t_2$  is regrafted to vertex  $u'$ . For rooted trees in (b), an edge  $(u, v)$  is also cut, but there are two ways to regraft  $t_2$ .

Figure 2.3(a) shows an example of a phylogenetic network on  $X = \{a, b, c\}$ . The set of nodes  $V$  consists two disjoint subsets:  $V_N$ , the set of nodes with indegree at least two (called network nodes), and  $V_T$ , the set of nodes with indegree at most one (the set of tree nodes). We denote by  $r(N)$  the root of  $N$ . The set  $V_T$  is further divided into two subsets:  $\overset{\circ}{V}_T$ , the set of internal tree nodes, and  $\mathcal{L}(N)$ , the set of leaves of  $N$ . Similarly, an edge incident into a network node is called a network edge; an edge incident to a tree node is called a tree edge; and an edge incident into a leaf is called a pendant edge.

A phylogenetic network induces a set of trees, called induced trees, each of which is obtained as follows:

1. For each node of indegree at least two, remove all but one of the network edges

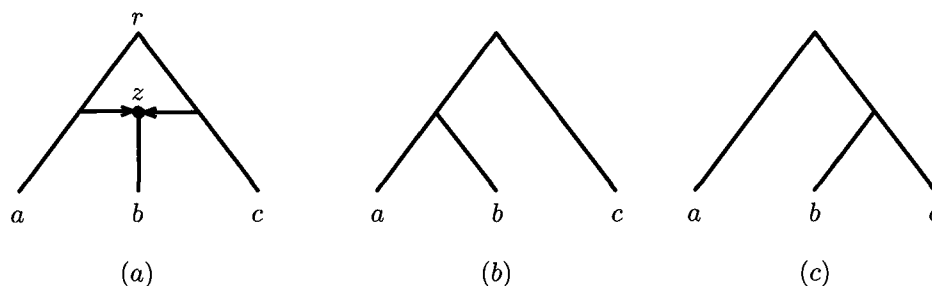


Figure 2.3 : An illustration of a network (a), and its set of induced trees (b) and (c).

incident into it; and

2. Suppress all nodes with indegree and outdegree one.

We denote by  $\mathcal{T}(N)$  the set of all trees induced by  $N$ . Note that for each network node  $v$ , there are exactly  $\text{indeg}(v)$  choices, and hence, the number of induced trees is bounded by  $\prod_{v \in V_N} \text{indeg}(v)$ .

As with phylogenetic trees, a phylogenetic network  $N$  induces a set of (nontrivial) clusters, which we define as  $\mathcal{C}(N) = \bigcup_{T \in \mathcal{T}(N)} \mathcal{C}(T)$ . Since a phylogenetic tree on  $X$  can have at most  $|X| - 2$  nontrivial clusters, the number of clusters induced by  $N$  is bounded by  $(|X| - 2) \times \prod_{v \in V_N} \text{indeg}(v)$ . For example, Figure 2.3(b) and (c) are two (and only two) trees induced by the network in Figure 2.3(a). The set of induced clusters of this network is  $\{\{a, b\}, \{b, c\}\}$ .

### 2.3 Species Tree, Gene Tree and their Incongruence

Although both species trees and gene trees can be represented by phylogenetic trees, they are conceptually different: a gene tree shows the evolutionary history of a single gene, while a species tree shows the evolution of species via the process of speciation. During the course of evolution of species, a number of biological events can

cause a gene tree different from its containing species tree [5]. We describe in this section lineage sorting, and hybridization and horizontal gene transfer, on which this dissertation focuses.

### 2.3.1 Lineage Sorting

Lineage sorting occurs at a population-level. If in an ancestral population, there exist several variants of some gene (alleles), then due to random genetic drift the evolutionary history of that gene is incongruent topologically with the species tree [50, 51, 52]. Consider an example in Figure 2.4 where two gene variants, one in red lines and the other one in blue lines, exist in the ancestral population of species  $a$ ,  $b$  and  $c$ . During the evolution, alleles in the next generation are a random sample of those of the previous generation. By chance,  $g_b$  and  $g_c$  are sampled from individuals that have the blue allele, while  $g_a$  is from the red allele. Therefore, in the gene tree,  $b$  and  $c$  are siblings, while in the species tree, which is shown in tubes,  $a$  and  $b$  are siblings.

Looking backward in time, in the tradition of the coalescent theory [53], the topological disagreement between a species tree and gene tree is due to the fact that some lineages fail to coalesce at their MRCA; instead, they coalesce deeper in the past. In the case in Figure 2.4, when tracing back in time, we find that  $g_b$  and  $g_c$  coalesce before they together coalesce with  $g_a$ , making  $b$  and  $c$  sibling taxa in the gene tree. If instead,  $g_a$  and  $g_b$  coalesce at sometime between  $\tau_1$  and  $\tau_2$ , then the gene tree is topologically identical to the species tree.

Assuming the Wright-Fisher model where ancestors are chosen randomly with replacement from previous generations, we can compute the probability that two gene lineages from  $a$  and  $b$  coalesce at time  $\tau$ ,  $\tau_1 < \tau \leq \tau_2$ , as follows. Let  $N_e$  denote the effective population size of the (haploid) ancestral population of  $a$  and  $b$ , and we

assume that  $N_e$  is constant through time. Then, the probability that two lineages have the same ancestor in the immediate previous generation is  $1/N_e$ . Therefore, the probability two gene lineages in  $a$  and  $b$  coalesce at time  $\tau$  is

$$\Pr(\tau) = \frac{1}{N_e} \prod_{\tau'=\tau_1+1}^{\tau-1} \left(1 - \frac{1}{N_e}\right) \approx \frac{1}{N_e} e^{-(\tau-\tau_1)/N_e} \quad \text{for large } N_e. \quad (2.4)$$

From this probability, we see that the chance of gene lineages from  $a$  and  $b$  not coalescing on the branch marked by  $\tau_1$  and  $\tau_2$  depends on the length of that branch,  $\tau_2 - \tau_1$ , and on the effective population size  $N_e$ , represented in the Figure 2.4 as the branch width. If the branch is short and wide, then they are less likely to coalesce before the speciation event at time  $\tau_2$ . If the branch is long and short, they are more likely to coalesce first before the resulting gene lineage coalesce with the gene lineage from  $c$ .

Similarly, the probability that those gene lineages do not coalesce on the branch marked by  $\tau_1$  and  $\tau_2$  is

$$\prod_{\tau=\tau_1+1}^{\tau_2} \left(1 - \frac{1}{N_e}\right) \approx e^{(\tau_2-\tau_1)/N_e} \quad \text{for large } N_e. \quad (2.5)$$

Based on this formula, we have the probability of obtaining the gene tree  $(a, (b, c))$  given the species tree  $((a, b), c)$  [54]

$$\Pr[(a, (b, c))] = \frac{1}{3} e^{-(\tau_2-\tau_1)/N_e}; \quad (2.6)$$

the probability of obtaining the gene tree  $((a, c), b)$

$$\Pr[((a, c), b)] = \Pr[(a, (b, c))] = \frac{1}{3} e^{-(\tau_2-\tau_1)/N_e}; \quad (2.7)$$

and the probability of obtaining the gene tree  $((a, b), c)$

$$\Pr[((a, b), c)] = 1 - [\Pr[(a, (b, c))] + \Pr[((a, c), b)]] = 1 - \frac{2}{3} e^{-(\tau_2-\tau_1)/N_e}. \quad (2.8)$$

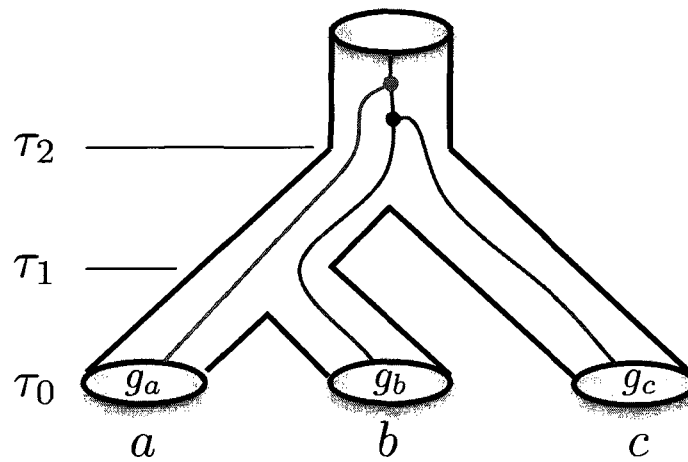


Figure 2.4 : An illustration of lineage sorting. There are two alleles, one in red and the other in blue, in the ancestral population at time  $\tau_2$ . In this figure, the current genes  $g_b$  and  $g_c$  both derived from the blue gene lineage, resulting a gene tree where  $b$  is closer to  $c$  than to  $a$ .

We have discussed in this subsection lineage sorting as a source of species/gene tree incongruence, and have demonstrated the use of the coalescent theory in computing the probability of a gene tree given a species tree in the simple case in Figure 2.4. The paper by Degnan and Salter [55] contains the formulae for the case of binary species tree with an arbitrary number of leaves. The books by Hein [56] and Wakeley [57] also contain an excellent treatment of the coalescent theory.

### 2.3.2 Hybridization and Horizontal Gene Transfer

In addition to lineage sorting, reticulate events such as hybridization and horizontal gene transfer (HGT) can also make a gene tree different from its containing species tree. In several groups of species, especially in plant and fish [58, 59], hybridization can occur between two species, resulting a new species that carries genetic material from both parents. Hybridization can be either:

- diploid: Each parent contributes a chromosome to the hybrid species, and therefore, the it has the same number of chromosomes as one of its parents;
- polyploid: The hybrid species combines all the chromosomes from its parents, and therefore.

However, whether hybridization is diploid or polyploid, the evolutionary relationships of species is no longer tree-like since different regions in the genome of a hybrid species can have different paths of evolution.

HGT also results in non-tree like evolution. It is a process in which a species receives genetic material from another species. HGT is believed to be rampant among bacteria [60], and as such it plays an important role in their evolution and genetic diversity. Three common mechanisms through which HGT occurs are [61]

- transformation: the uptake of free DNA (of a dead bacterium, for instance) from the surrounding environment;
- conjugation: the process in which genetic material is transferred from one bacterium to another through direct physical contact; and
- transduction: the process in which a bacterial virus, commonly called a phage, inserts genetic material (taken from one bacterium) to another bacterium.

We illustrate HGT from the phylogenetic point of view as in Figure 2.5. In the species tree, which is shown in tubes in Figure 2.5(a),  $a$  and  $b$  are sister taxa whose least common ancestor is a sibling of  $c$ . Consider the gene in thin lines. Through one of the mechanisms above,  $g_b$  in species  $b$  is transferred from  $c$ , instead of evolving from a common ancestor with gene copy  $g_a$  in  $a$ . Therefore, the evolution of that gene

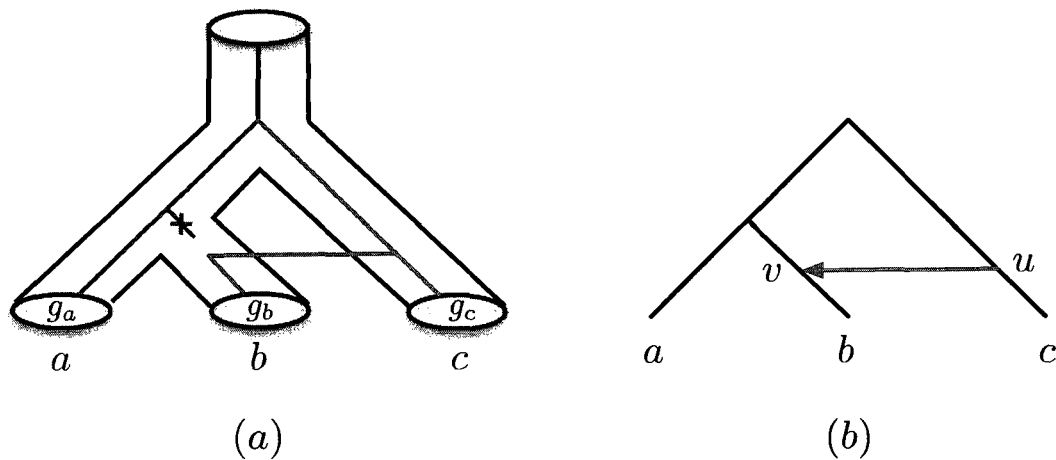


Figure 2.5 : Illustration of horizontal gene transfer. Figure (a): the gene tree in thin lines disagree with the species tree shown in tubes, because gene  $g_b$  is transferred from  $c$ , making in the gene tree  $b$  appear closer to  $c$  than to  $a$ . Figure (b): a phylogenetic network representing HGT.

is incongruent with that of the species; as the figure shows,  $b$  and  $c$  are now sister taxa. Figure 2.5(b) shows how we represent such an HGT graphically.

Hybridization and HGT are examples of reticulate events, and when they occur the phylogenies of species cannot be represented by phylogenetic trees. Instead, they are represented by phylogenetic networks. As defined in Section 2.2, a phylogenetic network is a DAG consisting of tree nodes and network nodes. Network nodes and network edges of a phylogenetic network represent reticulate events. For example, the network in Figure 2.3 is the phylogeny for three species  $a$ ,  $b$  and  $c$ , where  $b$  is a hybrid species of  $a$  and  $c$ . In this network, network node  $z$  and its two incident network edges represent the hybridization of  $a$  and  $c$ . As another instance, network edge  $(u, v)$  in Figure 2.5(b) represents the transfer of a genetic material from  $c$  to  $b$ .



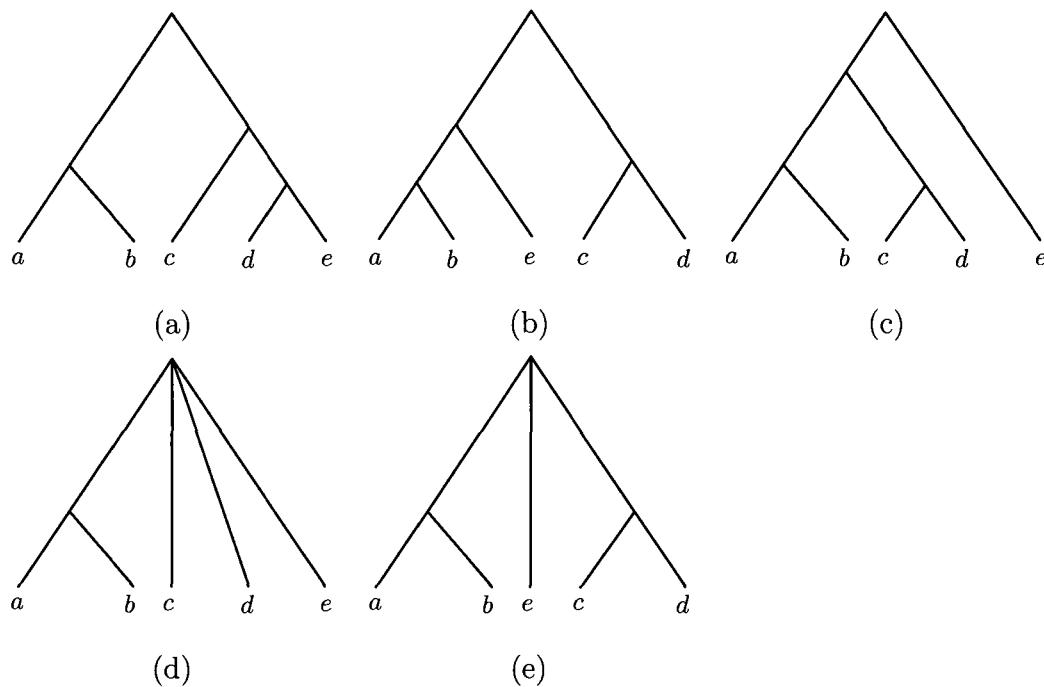


Figure 2.6 : Illustration of strict- and majority-consensus and democratic vote methods. Figures (a), (b), and (c) are input gene trees. The strict consensus tree is shown in (d), while (e) shows the majority consensus tree. The species tree of the democratic vote method is either (a), (b) or (c) since for this set of input gene trees, all of them appear with the same frequency  $1/3$ .

## 2.4 Methods for Inferring the Species Trees despite Lineage Sorting

Due to those processes discussed in the previous section, one cannot equate a gene tree to its containing species trees. With the availability of multiple locus data, how can we use them to infer the species tree? In this section, we discuss some of the methods commonly used for species tree inference despite lineage sorting. Generally speaking, those methods can be divided into two groups: (a) total evidence, and (b) separate analysis. Let us begin with a total evidence method.

### 2.4.1 Concatenation

In this method all DNA/protein sequences are concatenated together. Then, any classic phylogenetic methods such as maximum parsimony or maximum likelihood is used to build a single tree from the concatenated sequence. Advantages of the method include its simplicity, and the availability of an array of well-established methods for the analysis of the concatenated sequence. In fact, it is commonly used in practice [26, 27, 25]. However, we should not ignore its main weakness that it treats all genes equally. Different genes may have different courses of evolution, either having different mutation rates or involving biological events such as hybridization or HGT. When molecular sequences are concatenated, all the differences among genes are averaged away, leading to incorrect phylogenetic estimates [31].

### 2.4.2 Consensus Methods

We now consider simple attempts of the separate analysis approach at reconciling incongruence among gene trees. Instead of concatenating all gene sequences, we reconstruct a gene tree for each individual gene. We now have a set of gene trees, which may exhibit conflicting phylogenetic signals among themselves. In the first variant of those methods, the strict-consensus method, only clusters that appear in all gene trees are used to build the species tree. To illustrate, consider three trees in Figures 2.6(a), (b), and (c). Only cluster  $\{a, b\}$  appears in all those three trees, and hence we obtain the species tree in Figure 2.6(d). That strict-consensus tree, as we might notice, is highly unresolved; its root has four children. This is the main disadvantage of the strict-consensus tree, which makes it not frequently used in practice.

We can lessen the requirement that only clusters occurring in all gene trees appear

in the species tree. If two clusters that both appear in more than 50% of all gene trees, then we know that there must be at least one tree inducing both of them. Therefore, the set of clusters, each of which appears in more than 50% of all gene trees, allows us to uniquely build a tree. Consider the three gene trees in Figure 2.6 again. Besides cluster  $\{a, b\}$  that all gene trees have, cluster  $\{c, d\}$  appears in 2 of them. Hence, we have the majority consensus tree as in Figure 2.6(e). We note that we still have a non-binary tree, although the severity of irresolution is alleviated, compared to the strict-consensus method.

### 2.4.3 Democratic Vote

Another method for inferring the species tree from multiple gene trees is by “democratic voting.” As its name suggests, this method takes the gene tree that occurs with the highest frequency as the species tree. At first, it seems to be reasonable to declare such a gene tree as the species tree. However, it was shown that such a gene tree might disagree with the true species tree [30]. We also note that there can be more than one such gene tree. Figure 2.6 is an example. All three gene trees are different, and so each one of them is equally probable to be chosen as the species tree by this method.

### 2.4.4 Maximum Likelihood

In this approach, we infer the species tree (its topology and branch lengths) from a set of input gene trees by seeking a tree that maximizes its likelihood, which is defined as

$$\prod_{\text{loci}} \Pr(\text{gene trees} \mid \text{species tree}), \quad (2.9)$$

where we assume that gene trees at different loci evolve independently. The probability of a gene tree given a species tree can be computed using the coalescent theory. We show the computation for trees with three leaves in Subsection 2.3.1. Trees with four and five taxa were treated in [62], and binary trees with an arbitrary number of leaves were treated in [55].

We note that in inferring the species tree by maximizing its likelihood, we assume the gene trees are correct. However, we can eliminate this assumption by simultaneously inferring both the species tree and gene trees. Let  $\Pr(\text{sequence} \mid \text{gene tree})$  be the probability of observing a sequence given a gene tree. Then, we infer the species tree by maximizing the quantity [5]

$$\prod_{\text{loci}} \sum_{\text{gene trees}} \Pr(\text{sequence} \mid \text{gene tree}) \times \Pr(\text{gene trees} \mid \text{species tree}), \quad (2.10)$$

where we also assume that different loci evolve independently.

#### 2.4.5 GLASS

GLASS, short for Global LAteSt Split, was introduced by Mossel and Roch [63]. It is a clustering approach. Suppose we are given a set of gene trees, along with times assigned to their internal nodes. To simplify the description, we assume that one individual is sampled per gene per species, although GLASS can handle multiple-allele gene trees. For two taxon clusters  $A$  and  $B$ , GLASS defines a distance between them as

$$d(A, B) = \min_{GT} \{\tau_{GT}(a, b) : a \in A, b \in B\}, \quad (2.11)$$

where  $\tau_{GT}(a, b)$  is the time of the MRCA of taxa  $a$  and  $b$  in gene tree  $GT$ . Initially, each taxon is considered as a single-element cluster, and GLASS finds among all pairs of taxa the one whose most common ancestor's time is smallest. It then groups those

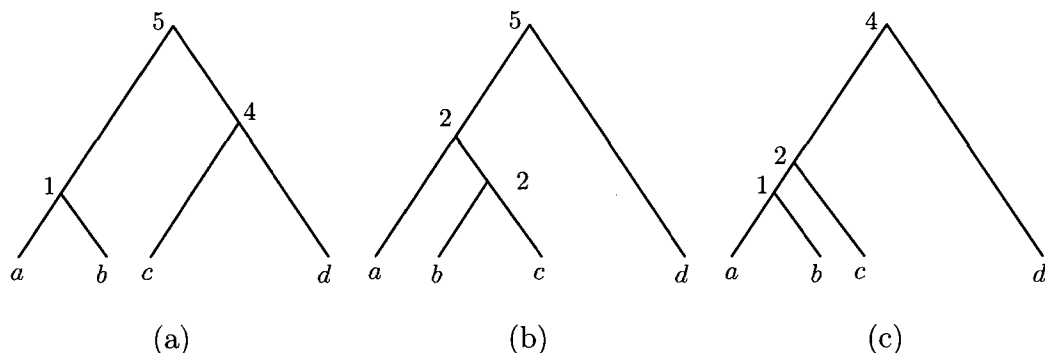


Figure 2.7 : Illustration of the GLASS method. Two input gene trees are in (a) and (b). The inferred tree is shown in (c).

two taxa into a new cluster, and recompute the distance between this new cluster and all the other clusters according to Equation (2.11). This process is repeated until only one cluster remains (i.e., until all taxa are in one cluster). The species tree topology is reconstructed by clusters produced during this process, while the divergence times at internal nodes are the distance  $d(\cdot, \cdot)$ .

Let us illustrate GLASS with two gene trees in Figures 2.7(a) and (b). In the first step, we merge taxa  $a$  and  $b$  together, because their MRCA's time is 1, the smallest value. Next, we merge  $\{a, b\}$  with  $c$ , whose distance is 2. Finally, we obtain the cluster  $\{a, b, c, d\}$  with time 4 assigned to it. We have clusters  $\{a, b\}$ ,  $\{a, b, c\}$  and  $\{a, b, c, d\}$ , along with the distances  $d(\cdot, \cdot)$ , which allow us to build the species tree  $((a, b), c), d$  as shown in Figure 2.7(c).

#### 2.4.6 BEST

Liu and Pearl [33] introduced BEST, Bayesian Estimation of Species Trees, for inferring species trees from multi-locus data. The goal of the method is to find the species tree that maximizes the posterior probability  $\Pr(\text{species tree} | D)$ , where  $D$  is

the multi-locus data available to us. They show in their paper that

$$\Pr(\text{species tree} | D) = \int_{\mathbf{G}} f(\mathbf{G} | D) f(\text{species tree} | \mathbf{G}) d\mathbf{G}, \quad (2.12)$$

where  $\mathbf{G}$  is the input gene tree vector. The BEST algorithm is developed based on this formula. The algorithm consists of three steps. In the first step, it computes an estimate of  $f(\mathbf{G} | D)$ , denoted as  $K(\mathbf{G} | D)$ . In order to do this, BEST first obtains an estimate of  $f(\mathbf{G})$ , the prior distribution of gene trees by considering only “maximum species trees” with internal nodes being as deep as possible but still being compatible with all gene trees in  $\mathbf{G}$ . A sequence of  $K(\mathbf{G}_i | D)$  for  $1 \leq i \leq N$  is generated by Markov Chain Monte Carlo (MCMC) in MrBayes [64]. Prior probabilities  $K(\mathbf{G}_i)$  for  $1 \leq i \leq N$  are also recorded.

In the second step, BEST uses  $K(\mathbf{G} | D)$  to estimate  $f(\text{species tree} | G)$ . For each  $\mathbf{G}_i$ ,  $1 \leq i \leq N$ , BEST computes  $k$  samples from  $f(\text{species tree} | \mathbf{G}_i)$  using another MCMC procedure. In effect, we produce a sample of size  $k \cdot N$  from  $f(\text{species tree} | \mathbf{G})$ .

The final step is to combine those estimates computed in the first two steps to produce an estimate of  $\Pr(\text{species tree} | D)$ . However, we need to correct the fact that we use estimates of  $f(\mathbf{G} | D)$  and  $f(\text{species tree} | \mathbf{G})$ , instead of their true values. This is done by multiplying each  $K(\mathbf{G}_i | D)$  a weight  $f(\mathbf{G}_i)/K(\mathbf{G}_i)$ . Note that we do not know  $f(\mathbf{G}_i)$ 's, but each of it can be estimated using  $k$  samples of  $f(\text{species tree} | \mathbf{G}_i)$ , as shown in their paper.

## 2.5 Existing Phylogeny-based HGT Detection Methods

We discuss in this section some HGT detection methods. All of them are phylogeny-based, that is, they detect HGT based on the topological discordance between a pair of species and gene trees. The methods discussed here are T-REX [65], LatTrans [44],

HorizStory [46], EEEP [45], and RIATA-HGT [47, 48].

### 2.5.1 The T-REX Package

The T-REX package implements an HGT detection method developed by Boc and Makarenkov [66]. The method is based on the distance between taxa associated with trees to detect HGT. Suppose we have a species tree  $ST$  and gene tree  $GT$ . If  $ST$  and  $GT$  are identical, then there is no HGT. Otherwise, the algorithm will find HGT events that are likely to take place by minimizing a least-squares function  $Q$ .

Suppose we add an HGT branch  $(a, b)$  to the species tree  $ST$ . Then there are only three possible scenarios where the minimum distance between two taxa  $i$  and  $j$  can be changed (Figure 2.8). Define the function:

$$\text{dist}(i, j) = d(i, j) - \min\{d(i, a) + d(j, b); d(j, a) + d(i, b)\},$$

where  $d(i, j)$  is the minimum distance between  $i$  and  $j$  in the species tree. Denote  $\delta(i, j)$  the distance between the same taxa  $i$  and  $j$  in the gene tree  $GT$ . Then, the function  $Q$  which we seek to minimize is defined as:

$$Q(ab, l) = \sum_{\text{dist}(i, j) > l} (d(i, j) - \text{dist}(i, j) + l - \delta(i, j))^2 + \sum_{\text{dist}(i, j) \leq l} (d(i, j) - \delta(i, j))^2,$$

where  $l$  is the length of the HGT branch  $(a, b)$ . The function  $Q$  measures the topological difference between the species tree and gene tree after the addition of branch  $(a, b)$ ; in the best case  $d(i, j) = \delta(i, j)$  and  $l$  being exactly the difference  $\text{dist}(i, j)$ , then the the addition of  $(a, b)$  makes the species tree identical to the gene tree.

Once we find a branch  $(a, b)$  with optimal  $Q(ab, l)$  value, the minimum length  $d$  between taxa is recomputed, and the procedure described above is repeated. There are  $(2n-3)(2n-4)$  possible (directed) HGT branches (suppose the each of the species

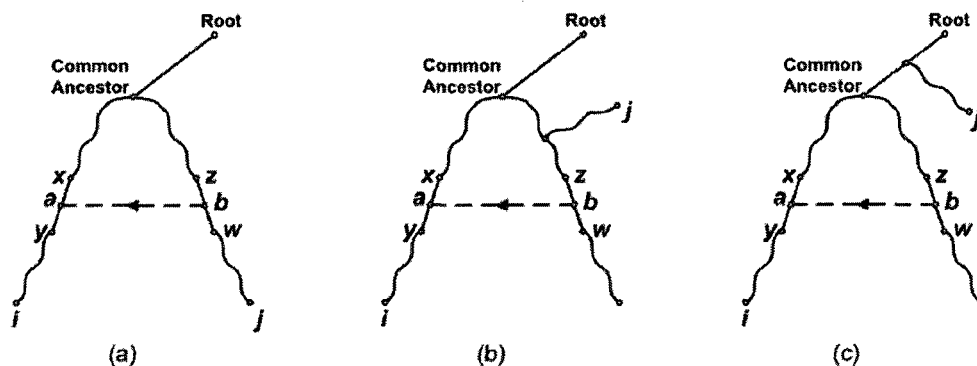


Figure 2.8 : Three possible scenarios where the addition of an HGT branch  $(a, b)$  can change the minimum distance between two taxa  $i$  and  $j$  [66].

tree and gene tree has  $n$  leaves), and hence the procedure stops after  $(2n - 3)(2n - 4)$  steps.

This method is different from all other HGT detection methods mentioned in this dissertation in the sense that, in addition to topological incongruence between the species tree and gene tree, their branch lengths are used to detect HGT.

### 2.5.2 LatTrans

Hallet and Lagergren [44] propose a model for HGT. In this model, a horizontal transfer scheme for a species tree  $ST$  is an acyclic directed graph built from  $ST$  and a set  $\Xi$  of new directed edges added to  $ST$  (so edges in  $\Xi$  represent HGT events). A horizontal transfer scenario is then defined as a triple  $(ST, \Xi, g)$ , where  $(ST, \Xi)$  is a horizontal transfer scheme and  $g$  is a procedure allowing us to obtain the gene tree  $GT$  from the scheme. The interesting point of the model by Hallet and Lagergren is that it allows more than one copy of a gene to exist at any point of the evolution. When this is the case, the algorithm views gene copies as possible HGT events.

The algorithm detects HGT by first determining at which vertices the species tree



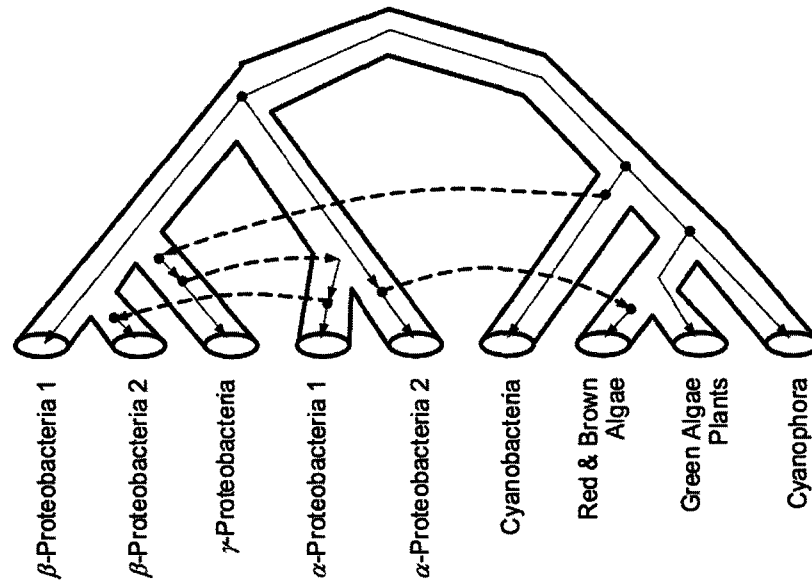


Figure 2.9 : One horizontal gene transfer scenario found by LatTrans for gene *rbcL* [44].

and gene tree can disagree. The algorithm distinguishes two such groups of nodes, called I-fat vertices and H-fat vertices. Then, the algorithm fixes those discordant vertices by adding I-moves and H-moves accordingly. The addition of those events might introduce new disagreements, so the algorithm needs to repeat this process until the two trees become identical.

### 2.5.3 HorizStory

HorizStory [46] uses a relatively simple strategy for detecting HGT. The algorithm recursively repeats two phases of consolidation and rearrangement until two trees become identical. In the first phase of consolidation, identical clades in two trees are collapsed, thus essentially reducing the size of the two trees. For example, if two trees both have a clade  $((A, B), C)$ , then that clade is replaced by a single new leaf.

After the trees are simplified, the algorithm’s second phase tries to detect HGT by cutting from the species tree one leaf and regrafting it at every possible edge. If it ever detects one such move that creates new identical clades, then the algorithm records that move, and goes back to the consolidation phase. These two steps are repeated until the two trees can be reduced to a single leaf. As there can be more than one set of HGT events that can reconcile two trees, HorizStory does try to find the “best” one by using a branch-and-bound strategy.

As stated in [46], HorizStory is limited to comparing trees that are relatively similar. This is due to the fact that there are too many ways to cut and regraft a leaf in the second phase of rearrangement ( $O(n^2)$ , where  $n$  is the number of leaves in species and gene trees). Another shortcoming of HorizStory is that it does not consider HGT events between two internal branches. Such an event moves a group of leaves, and hence it might be equivalent to several events detected by HorizStory. Therefore, the “best” scenario found by HorizStory is not necessarily optimal.

#### 2.5.4 EEEP

EEEEP [45] stands for Efficient Evaluation of Edit Path, and the concept of edit paths is central to the detection of horizontal transfers in EEEP. Consider the species tree and gene tree in Figure 2.10. One HGT move is required to reconcile the gene tree. Because of the HGT branch, clusters induced by the species tree and gene tree are not the same. For the two trees in Figure 2.10, cluster  $\{T_1, T_2, T_3\}$  induced by edge  $E_0$  appears in the species tree, but is not an induced cluster of the gene tree. Cluster  $\{T_4, T_5, T_6, T_7\}$  induced by edge  $E_1$  also only appears in the species tree.  $\{T_1, T_2\}$  and  $\{T_3\}$ , on the other hand, are induced clusters of both the species and gene tree. We see that edges  $E_0$  and  $E_1$  form a connected path. In order to reconcile those

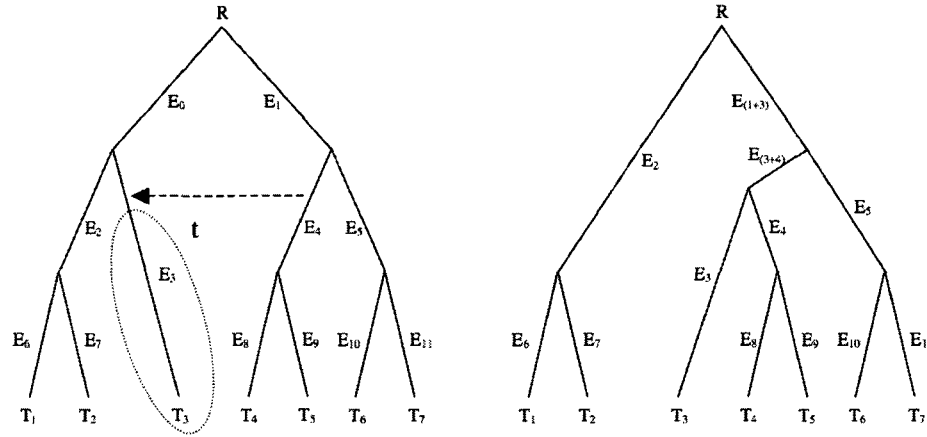


Figure 2.10 : Edith paths. The species tree is on the left, and the gene tree is on the right. The gene tree differs from the species tree by only one HGT move. Edges  $E_0$  and  $E_1$ , which induce clusters discordant with the gene tree, form a path. Similarly, edges  $E_{(1+3)}$  and  $E_{(3+4)}$  form a connected subgraph of the gene tree [45].

discordant clusters, HGT edges must move  $T_3$  from the cluster  $\{T_1, T_2, T_3\}$  to an edge in the clade under  $E_1$ .

The above observation is utilized by EEEP to detect HGT. First, it computes induced clusters of the species and gene trees, and finds which clusters are in discordance. Those discordant clusters define a connected subgraph (or a path in this specific example). HGT events that connect two branches of the clades under two leaves of this subgraph are considered as candidate transfers. To find the most parsimonious scenario, EEEP considers them in a breadth-first manner: HGT events occurring between closest leaves of the subgraph are considered first. With each HGT event applied to the species tree, its induced clusters are recomputed, and this procedure is repeated until the gene tree is obtained.

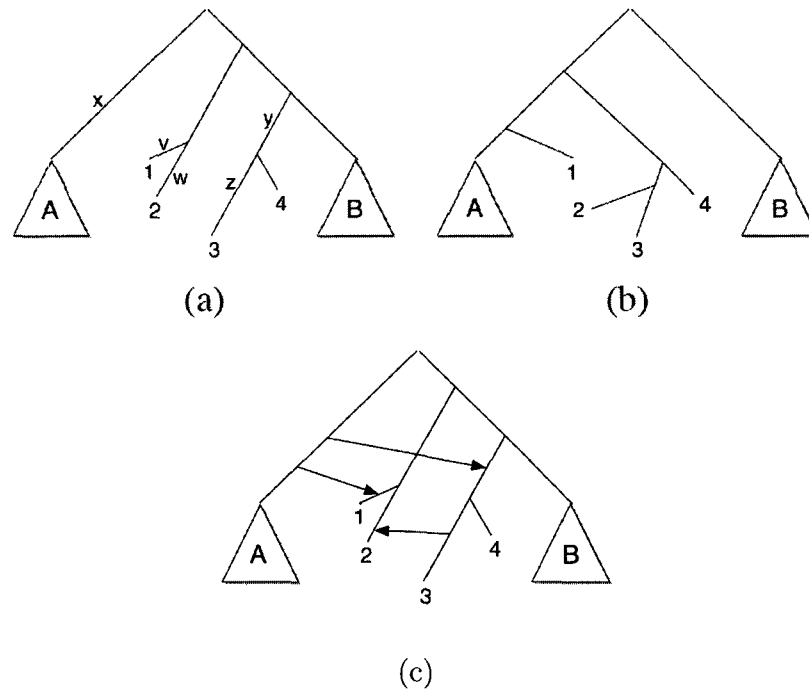


Figure 2.11 : Illustrating RIATA-HGT. Figures (a) and (b) are the species tree and gene tree, respectively. Figure (c) depicts a scenario to reconcile the species and gene trees [47].

### 2.5.5 RIATA-HGT

This subsection is a brief description of RIATA-HGT. For further details, see [47].

#### Maximum Agreement Subtrees

Let us first begin with maximum agreement subtrees (MASTs) that RIATA-HGT uses to detect HGT. Let  $T$  be a phylogenetic tree on a set of taxa  $X$ , and let  $A$  be a nonempty subset of  $X$ . We denote by  $T(A)$  the minimal subtree of  $T$  whose leaf set is  $A$ . Further, we denote by  $T|_A$  the restriction of  $T$  on  $A$  that is obtained from  $T(A)$  by suppressing all nodes of degree 2, except for its root. For example, the restriction of the species tree in Figure 2.11 (a) on the set  $\{1, 2, 3, 4\}$  is the subtree  $((1, 2), (3, 4))$ .

Let  $T'$  be another phylogenetic tree over the same set of taxa  $X$ . If  $T|_A = T'|_A$ , then this subtree is called an agreement subtree of  $T$  and  $T'$ . A maximum agreement subtree, denoted by  $\text{MAST}(T, T')$ , is an agreement subtree with the maximum number of taxa. Consider the two example trees in Figure 2.11 again. Two of their agreement subtrees are  $((3, 4), B)$  and  $(A, B)$ . If each of the subtrees represented by  $A$  and  $B$  has at least three leaves, then one can easily verify that the MAST is  $(A, B)$ .

We note that if there is no HGT, then maximum agreement subtree  $T$  and  $T'$  is definitely  $T$  because  $T$  and  $T'$  are identical topologically. When there is HGT,  $T$  and  $T'$  can disagree with each other. From the graph-theoretic point of view, each HGT event can be simulated by an SPR move. Therefore, by extracting out their MAST, we can know which part of  $T$  must be pruned and regrafted in order to obtain  $T'$ , and hence we can find HGT events.

We also note that computing the SPR distance between two trees is NP-hard [43], while computing the maximum agreement subtree is polynomial. One such an algorithm is developed by Steel and Warnow [67] that finds the MAST by working bottom-up from the leaves and by employing dynamic programming. For leaves  $a$  in  $T$  and  $a'$  in  $T'$ , their MAST is  $a$  if  $a = a'$ . Consider two nodes  $p$  in  $T$  and  $p'$  in  $T'$ . In the case both  $T$  and  $T'$  are binary,  $p$  and  $p'$  have two children, and so there are four possibilities to combine their children's MASTs to obtain their MAST. In the case internal nodes of  $T$  and  $T'$  have arbitrary degrees, the problem of finding a MAST for  $p$  and  $p'$  is converted to finding a maximum weighted matching in a bipartite graph. See [67] for a complete description of the algorithm.

---

**Algorithm 1** ComputeHGT( $ST, GT$ )

---

```

1: compute MAST( $ST, GT$ );
2: if  $ST = \text{MAST}(ST, GT)$  then
3:   return;
4: else
5:   call Decompose to “refine” subtrees in  $ST$  and  $GT$  that are not in
     MAST( $ST, GT$ );
6: end if
7: for all pair of subtree  $st$  and  $gt$  returned by Decompose do
8:   call ComputeHGT( $st, gt$ ) to reconcile  $st$  and  $gt$ ;
9:   call AddSingleHGT to add an HGT event for  $gt$ ;
10: end for

```

---

**Description of RIATA-HGT**

The algorithm has three components. In the main procedure, **ComputeHGT**, it computes the MAST of the two input trees  $ST$  and  $GT$  by using the algorithm described in [67], decomposes the remaining discordant subtrees by calling the second component **Decompose**, detects HGT events that move the subtrees (**AddSingleHGT**), and then recursively calls **ComputeHGT** to reconcile the subtrees. Algorithm 1 is a high-level description of RIATA-HGT.

We illustrate the algorithm on the two example trees in Figure 2.11. Their MAST is the tree  $(A, B)$  (assuming  $A$  and  $B$  have at least three leaves), and those discordant subtrees are  $st_1 = (1, 2)$ ,  $st_2 = (3, 4)$  (in the species tree), and  $gt_1 = (1)$ ,  $gt_2 = ((2, 3), 4)$  (in the gene tree). We note that we have  $gt_2 = ((2, 3), 4)$  in the gene tree, while leaf 2 is in the subtree in  $st_1$  and leaf 3 is in  $st_2$ . This means that an HGT

involving  $gt_2$  cannot place both leaves 2 and 3 in the correct place in the gene tree. The purpose of **Decompose** is to resolve this problem. In this example,  $gt_2$  is broken down further into two smaller subtrees (2) and (3, 4). An HGT edge involving one of those decomposed subtrees, for example (3, 4), can now be handled by **AddSingleHGT** by noting that its head is the least common of (3, 4) and that its tail is a node in the species tree that corresponds to the least common ancestor of the subtree's siblings in the gene tree (in this example, the root of the subtree  $A$ ). An example of a complete set of HGT transfers is given in Figure 2.11 (c).

The original algorithm RIATA-HGT [47] computes only a single set of HGT events, while in fact there can be more than one [54]. We recently extended it to compute multiple solutions, and introduced a refinement procedure for dealing with trees with non-binary trees [48]. For full details of these new improvements, see [48, 68].

## Chapter 3

# Species Tree Inference from Gene Trees Using Their Topologies and Coalescence Times

This chapter is about our first method for inferring species trees from multi-locus data [69]. We note that gene trees can be different from each other in terms of both topologies and branch lengths, e.g., trees in Figure 3.1. The method presented in this chapter allows us to infer both the species tree topology and branch lengths from a set of gene trees whose incongruence is due to lineage sorting.

In Section 3.1, we describe how we reconcile a gene tree within a species tree where their incongruence is due to lineage sorting. We then introduce a cost function to measure the severity of deep coalescence events. We note that our model for reconciling a pair of trees here is similar to that in the paper by Maddison [5], except for the fact that ours uses branch length information of both trees.

Based on this model, we developed an algorithm for finding an optimal tree. The algorithm operates in three phases, the first of which computes a set of species tree topologies, the second of which estimates divergence times of those candidate trees using an ILP formulation, and the third of which selects the optimal tree under a criterion that combines deep coalescence and species/gene tree incongruence. Those three phases are discussed in detail in Section 3.2.

The remaining section of this chapter contains an empirical study of our method on nine strains of the *Staphylococcus aureus* bacteria. We analyzed 1898 genes in the data set, and used the reconstructed gene trees to create 24 candidate species tree topology



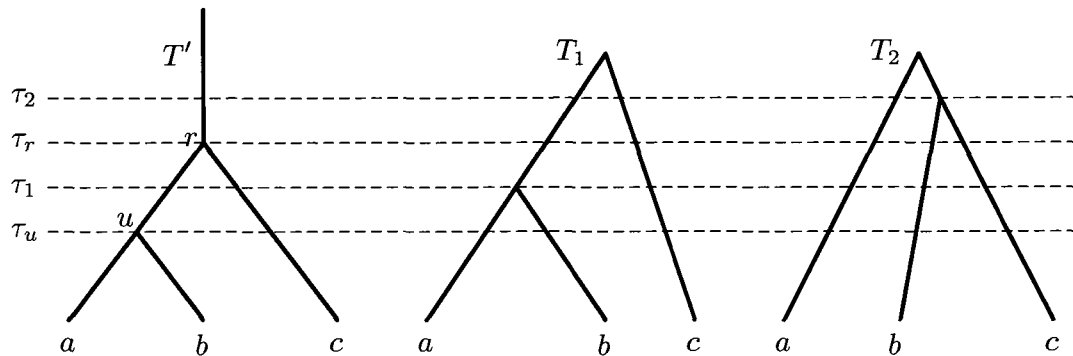


Figure 3.1 : Illustration of discordance between species and gene trees. Tree  $T'$  is a species tree, and  $T_1$  and  $T_2$  are two gene trees, which are different from  $T'$ . In gene tree  $T_1$ , gene lineages  $a$  and  $b$  coalesce at time  $\tau_1$  prior the divergence time  $\tau_s$ . Gene tree  $T_2$  is different from tree  $T'$  topologically.

candidates. The divergence time was then inferred, which took approximately 1 hour for each of those tree candidates. (The inference algorithm was run on a 3.2 GHz Intel Pentium 4 machine with 1 GB of RAM.) Despite the high degree of sequence identity at the nucleotide level in the data set, our method is still able to return a binary tree in a reasonable amount of time. This affirms its suitability for analyzing very closely related organisms.

### 3.1 Reconciling Gene Trees within Species Trees

As illustrated in Figure 3.1, a gene tree can be different from a species tree in terms of both shapes and branch lengths. In coalescent theory, this incongruence is caused by the failure of gene lineages to coalesce at their MRCA. For example, in Figure 3.1 although the gene tree  $T_1$  is identical topologically to the species tree  $T'$ , they are not the same if branch lengths are taken into account; in  $T_1$ , gene lineages  $a$  and  $b$  do not coalesce at  $\tau_s$ , but instead at a deeper time  $\tau_1$ . As the coalescence time of a group of gene lineages cannot be later than their MRCA time, any time assignment

to a candidate species tree must satisfy this requirement. We state this requirement formally as follows. Let  $T$  be a gene tree and let  $T'$  be a species tree. Further, we denote  $\tau_T$  and  $\tau_{T'}$  two time functions for  $T$  and  $T'$  as defined in Chapter 2. Then, for any internal node  $v$  of  $T$ , we require that

$$\tau_{T'}(v') \leq \tau_T(v), \quad (3.1)$$

where  $v' = \text{MRCA}_{T'}(C_T(v))$  is the MRCA in the species tree  $T'$  of taxa in the cluster  $C_T(v)$  induced by  $v$ .

In Figure 3.1, in order for  $T'$  to reconcile both  $T_1$  and  $T_2$ , we must have  $\tau_u \leq \tau_1$  and  $\tau_r \leq \tau_2$ . Although in the figure,  $\tau_r$  is greater than  $\tau_1$ , it can happen that  $\tau_u < \tau_r \leq \tau_1$  without violating the condition in Equation (3.1). In this case, however, lineages  $a$  and  $b$  in the tree  $T_1$  do not coalesce on the branch  $(r, u)$ , but instead they coalesce deeper on the branch incident into the node  $r$ . In our model for reconciling a pair of species and gene trees, we penalize such scenarios. More precisely, suppose that we are reconciling a gene tree  $T$  within a species tree  $T'$ . Then if lineages of a cluster of  $T$  coalesce on the species tree branch incident into its MRCA, we call this a correct coalescence event and assign it weight zero. If they instead coalesce  $k$  species tree branches deeper than their MRCA, we call this a deep coalescence event and assign it weight  $k$ . With this weighting scheme, we seek to assign times to internal nodes of a candidate species tree topology in such a way that the total weight of all coalescence events is minimum.

We note that if we assume lineage sorting is the only cause of species/gene tree incongruence, then the condition in Equation (3.1) cannot be violated. However, there can still be genetic exchange between species after their divergence time, e.g., via horizontal gene transfer. Moreover, the estimated branch length in reconstructed gene trees is not always 100% accurate. Therefore, to make our model more flexible,

we relax the requirement that Equation (3.1) must always be satisfied, provided that a small number of violations of this condition (we call them shallow coalescence events) can lead to a significant decrease in the number of deep coalescence events.

## 3.2 An ILP-based Method for Inferring Species Trees

In this section, we present our method for inferring the species tree based on the model described in the previous section. Our method works in three phases:

1. Construction of species tree topology candidates from input gene trees;
2. Assignment of times to nodes of each of those candidate trees, based on the coalescence times of the gene trees;
3. Reconciliation of the gene trees within branches of each of those candidate trees so as to find an optimal tree among them under a criterion that combines deep coalescence, shallow coalescence and species tree/gene tree incongruence. This optimal tree is reported as the species tree for the input gene trees.

In the following subsections, we describe those phases in more detail.

### 3.2.1 Inferring Species Tree Topology Candidates

Despite lineage sorting, a gene tree still carry phylogenetic signals [34]. Moreover, from our empirical study, we see that the species tree topology is almost always an agglomeration of compatible clusters induced by gene trees; see Section 4.7 of Chapter 4. Therefore, in our method, we use only clusters induced by input gene trees to generate species tree candidates, instead of considering all  $(2n - 3)!!$  binary rooted trees (assuming that the taxon set has  $n$  elements). Based on this observation

---

**Algorithm 2** EstimateSpeciesTreeTopologies( $\mathcal{G}$ )

---

- 1: compute  $\mathcal{C} = \bigcup_{T \in \mathcal{G}} \mathcal{C}(T)$ ;
  - 2: construct the compatibility graph  $H$  for  $\mathcal{C}$ :
    - each vertex of  $H$  represents an element of  $\mathcal{C}$ ;
    - two vertices are adjacent if two clusters they represent are compatible;
  - 3: compute all maximal cliques in  $H$ , and use them to build species tree topology candidates;
- 

and the relationship between clusters and trees, we formulate a heuristic for finding candidate tree topologies from the set  $\mathcal{G}$  of gene trees as in Algorithm 2.

Steps 1 and 2 of the heuristic are quite straightforward. For Step 1, we simply visit each internal branch of each tree in  $\mathcal{G}$ , and compute the cluster induced by it. To build the compatibility graph  $H$  in Step 2, we add an edge between two vertices representing clusters  $C_1$  and  $C_2$  if and only if either  $C_1 \subseteq C_2$ ,  $C_2 \subseteq C_1$  or  $C_1 \cap C_2 = \emptyset$ . The number of clusters induced by a rooted phylogenetic tree is exactly  $n - 2$ , where  $n$  is the cardinality of its leaf set. Therefore,  $|\mathcal{C}| = O((n - 2)|\mathcal{G}|)$ , and those two steps can be carried out in polynomial time. Step 3 is more involved, as it seeks to enumerate all maximal cliques in the graph  $H$ , but there are already several efficient algorithms for doing this, e.g., [70, 71, 72, 73].

Figure 3.2 illustrates the algorithm on three input gene trees. Under each gene tree are clusters induced by that tree. In total, there are seven distinct clusters:  $\{b, c\}$ ,  $\{a, b, c\}$ ,  $\{d, e\}$ ,  $\{d, e, f\}$ ,  $\{e, f\}$ ,  $\{a, b\}$ , and  $\{d, f\}$ . The compatibility graph  $H$  is then constructed from those clusters. For this graph, there are six maximal cliques, all of which have four vertices. Those maximal cliques allow us to build six rooted,

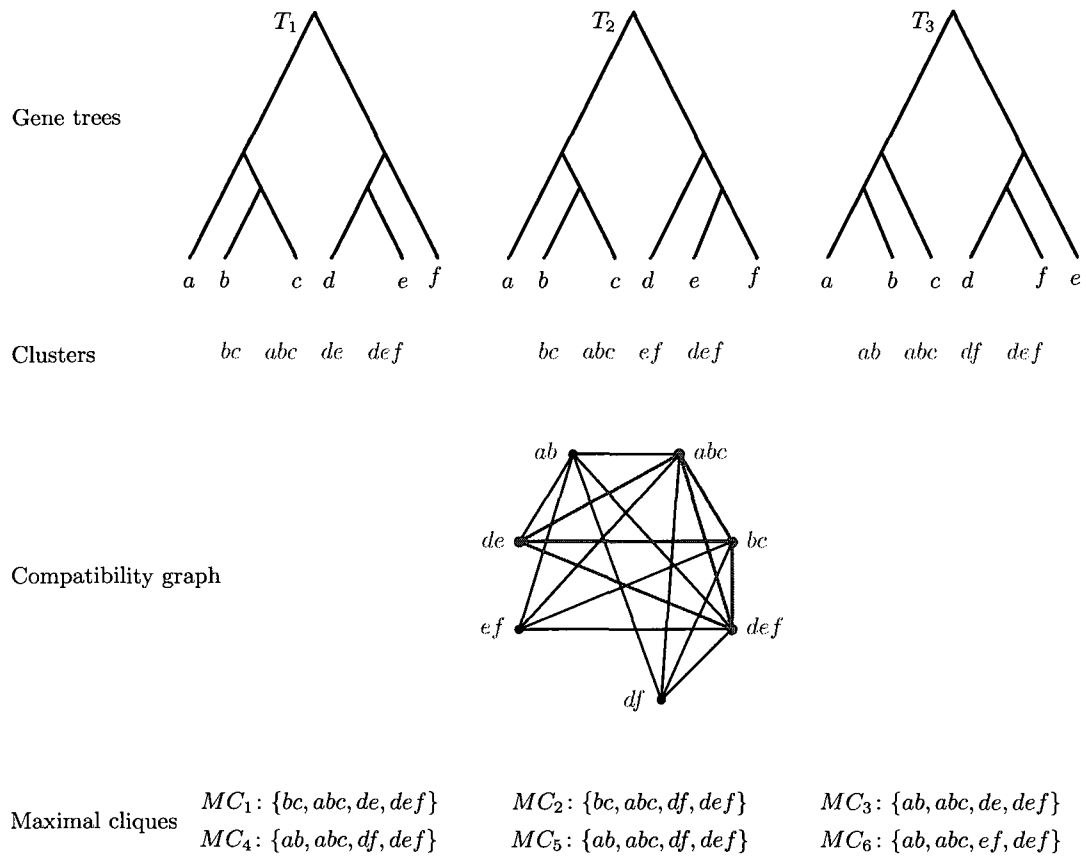


Figure 3.2 : Illustration of the first phase in our method. At the top are three gene trees, which are the input to the algorithm. The set of all clusters occurring in these gene trees are then computed, and their compatibility graph is built. Finally, the set of all maximal cliques are computed, and each defines a species tree topology candidate.

binary species tree topology candidates, which will be used in the second stage of our inference method.

### 3.2.2 Estimating Species Tree Divergence Times

Our next task entails estimating the divergence times at internal nodes of each of the species tree topology candidates that we computed. As we discussed in Section 3.1, different assignments of times to internal nodes of a species tree lead to different cost

of deep coalescence events. If we relax the condition in Equation (3.1), then some shallow coalescence events can occur. Our objective is to find a time assignment that minimizes a weighted combination of deep and shallow coalescence cost. This is done through the use of an ILP formulation that involves three elements: (1) temporal constraints on internal nodes of the species tree candidate; (2) constraints relating deep/shallow coalescence to temporal information of the internal nodes; and (3) an objective function. We now elaborate those elements, but before doing so we describe a special labeling of branches of a species tree candidate that facilitates their formulation using the language of linear and integer programming.

### **Labeling branches of the species tree candidate**

In our model for reconciling a gene tree within a species tree in Section 3.1, when a coalescence event occurs  $k$  branches deeper than its MRCA, we penalize it with a weight  $k$ . To easily convert this relation to a linear constraint, we propose to label branches of a species tree as follows. Let us be given a species tree  $T'$  and a cluster  $C_T(v)$  of a gene tree tree  $T$ , and let  $v' = \text{MRCA}_{T'}(C_T(v))$ . We define a chain  $E_v$  of edges on the path from  $v'$  to the root of  $T'$ , plus the edge incident to this root node, and assign positive integers  $0, 1, \dots, |E_v| - 1$  to those edges in the same order. Denote by  $\ell_v(e)$  the integer value assigned to an edge  $e \in E_v$ . (Note here that we use the subscript  $v$  instead of  $v'$  because  $E_v$  and  $\ell_v$  are defined for each node  $v$  of  $T$ , and in general the MRCA mapping is not one-to-one (several different nodes  $v$  can be mapped to the same node  $v'$  in  $T'$ ).)

This labeling is essential for our ILP formulation, since it will be used to compute the weight of coalescence events. For example, since the MRCA of cluster  $\{a, b\}$  of  $T_1$  is node  $u$  in the species tree  $T'$ , we label (for this cluster) branch  $(r, u)$  of  $T'$  number

0 and the branch incident into  $r$  number 1. If  $\tau_u \leq \tau_1 < \tau_r$  as in Figure 3.1, then lineages  $a$  and  $b$  coalesce on branch  $(r, u)$  and this coalescence event has weight zero. On the other hand, if it occurs the branch incident into  $r$  because  $\tau_r \leq \tau_1$ , then its weight is 1, which is also the label of this branch.

### Temporal constraints

The topology of the species tree  $T'$  defines a partial order on the times of its internal nodes. This can be represented using linear constraints as

$$\tau_{T'}(u') > \tau_{T'}(v') \quad (3.2)$$

for every branch  $(u', v')$  of the species tree. For species tree  $T'$  in Figure 3.1, for example, we require that  $\tau_{T'}(r) > \tau_{T'}(u)$ .

Further, lineages in a cluster  $C_T(v)$  induced by a node  $v$  of the gene tree  $T$  may coalesce on any branch of the species tree  $T'$  above their MRCA, including the branch incident to  $r(T')$ . Temporally, this imposes a linear constraint

$$\tau_{T'}(v') \leq \tau_T(v), \quad (3.3)$$

where  $v' = \text{MRCA}_{T'}(C_T(v))$ , the most recent common ancestor of  $C_T(v)$  in  $T'$ . For the cluster  $\{a, b\}$  of  $T_1$  in Figure 3.1, for example, we have  $\tau_{T'}(u) \leq \tau_1$ , as  $u$  is the MRCA in  $T'$  of cluster  $\{a, b\}$ .

We note that since the coalescence times may be underestimated or horizontal gene transfer may have occurred after divergence of species, we relax this constraint by allowing the coalescence time of certain clades to be smaller than the time of their MRCA in the species tree. Let us designate a binary variable  $g_v$  for each node  $v$  in  $T$  to indicate whether the coalescence event for all lineages under  $v$  is shallow (i.e.,

it occurs after its corresponding divergence time) or not (i.e., it occurs prior to its corresponding divergence time):

$$\text{if } \tau_{T'}(v') \leq \tau_T(v), \quad \text{then } g_v = 0$$

$$\text{if } \tau_{T'}(v') > \tau_T(v), \quad \text{then } g_v = 1,$$

where  $v' = \text{MRCA}_{T'}(C_T(v))$ . Defining  $M$  to be any positive real number that is larger than the time of the root of any gene tree  $T$ , we can convert those constraints into linear ones:

$$\tau_T(v) - (1 - g_v)M < \tau_{T'}(v') \tag{3.4}$$

$$(1 - g_v)\tau_T(v) + g_vM \geq \tau_{T'}(v') \tag{3.5}$$

$$g_v \in \{0, 1\}. \tag{3.6}$$

Because of the choice of  $M$ , the Equation (3.4) forces  $g_v$  to be assigned 0 if  $\tau_{T'}(v') \leq \tau_T(v)$ , while Equation (3.5) forces  $g_v = 1$  if  $\tau_{T'}(v') > \tau_T(v)$ . We note that we might need to subtract a small value (e.g.,  $10^{-8}$ ) from the right-hand side of the strict inequality (3.4) so that it can be entered as input to an ILP solver.

### Associating times with branches through their labels

The next set of constraints associate divergence times with the weight of coalescence events. Consider a node  $v$  of tree  $T$ . Let  $v' = \text{MRCA}_{T'}(C_T(v))$ , and let  $E_{v'}$  be the chain of edges on the path from  $v'$  to  $r(T')$ , plus the edge incident into  $r(T')$ . Suppose further that the edges  $e \in E_{v'}$  are labeled the labeling function  $\ell_c(e)$ . When reconciling  $T$  within  $T'$ , if lineages in  $C_T(v)$  coalesce on a branch  $e = (t_e, h_e) \in E_{v'}$ , then we must have

$$\text{if } \tau_{T'}(h_e) \leq \tau_T(v) < \tau_{T'}(t_e), \quad \text{then } f_v = \ell_v(e), \tag{3.7}$$



where the integer variable  $f_v$  is introduced for each node  $v$  and used to compute the weight of coalescence events. In order to convert this “if-then” constraint into linear constraints, we associate with each edge  $e = (t_e, h_e) \in E_v$  a binary variable  $\alpha_e$ , and rewrite this constraint as:

$$\tau_T(v) - (1 - \alpha_e)M < \tau_{T'}(t_e) \quad \forall e \in E_v, \quad (3.8)$$

$$\tau_T(v) + (1 - \alpha_e)M \leq \tau_{T'}(h_e) \quad \forall e \in E_v, \quad (3.9)$$

$$g_v + \sum_{e \in E_v} \alpha_e = 1, \quad (3.10)$$

$$f_v - \sum_{e \in E_v} \ell_v(e) \cdot \alpha_e = 0, \quad (3.11)$$

$$\alpha_e \in \{0, 1\} \quad \forall e \in E_v. \quad (3.12)$$

Constraints Equations (3.8) and (3.9) force the condition  $\tau_{T'}(h_e) \leq \tau_T(v) < \tau_{T'}(t_e)$  if lineages in the cluster  $C_T(v)$  coalesce on the branch  $e = (t_e, h_e)$ ; they are vacuously true if those lineages do not. The third constraint in Equation (3.10) ensures that a coalescence event can occur on exactly one branch. It is either a shallow event with  $g_v = 1$ , or a deep event on exactly one branch  $e$  with  $\alpha_e = 1$ . In the former case, the value of  $f_v$  should be zero, and in the latter case, the value of  $f_v$  is exactly  $\ell_v(e)$ , as guaranteed by Equation (3.11).

### The complete ILP formulation

Now that we have described the constraints and how to write them as linear constraints, we are in a position to introduce the complete ILP formulation for solving the problem of estimating divergence times in a species tree  $T'$ , given a set  $\mathcal{G}$  of gene trees with coalescence times at internal nodes. In our formulation, we seek to minimize a weighted combination of the costs of deep coalescence and shallow coalescence

events. The formulation is given in Algorithm 3.3.

### 3.2.3 Species/Gene Tree Reconciliation and Optimality

After we have assigned times to internal nodes of species tree candidates  $T'$ , we can now seek among them an optimal tree that we declare as the species tree for the set of input gene trees  $\mathcal{G}$ . The optimality criterion,  $\eta(T', \mathcal{G})$ , is defined as the sum of three terms:

1. the weighted number of gene tree clusters that are missing from  $T'$ ,  $w_{il} \sum_{T \in \mathcal{G}} |C \in \mathcal{C}(T): C \notin \mathcal{C}(T')|$ ,
2. the weighted number of deep coalescence events,  $w_{dc} \sum_{T \in \mathcal{G}} \sum_{v \in \dot{V}(T)} f_v$ ,
3. and the weighted number of shallow coalescence events,  $w_{sc} \sum_{T \in \mathcal{G}} \sum_{v \in \dot{V}(T)} g_v$ .

The weights  $w_{il}$ ,  $w_{dc}$ , and  $w_{sc}$  can be set in a way to reflect the significance given to each of the three terms in the criterion. For example, if only topological difference among the gene trees and species tree matters,  $w_{dc}$  and  $w_{sc}$  can be set to 0. Using this optimality criterion, we now give the description of the algorithm for inferring the species tree from a set of gene trees  $\mathcal{G}$  as in Algorithm 3.4.

## 3.3 Empirical Study

### 3.3.1 Materials and Analysis

In our experimental study, we used the *Staphylococcus aureus* bacteria, which infect humans in the community and hospitals and cause a variety of diseases. We obtained all the sequence data from the site <ftp://ftp.ncbi.nih.gov/genomes/>. Table 3.1 summarizes the nine strains we used.

<b>ESTIMATEDIVERGENCETIMES(<math>T', \mathcal{G}</math>)</b>	
minimize:	
$w_{dc} \sum_{T \in \mathcal{G}} \sum_{v \in \dot{V}(T)} f_v + w_{sc} \sum_{T \in \mathcal{G}} \sum_{v \in V(T)} g_v + \sum_{e \in T'} (\tau_{T'}(t_e) - \tau_{T'}(h_e)),$	
subject to:	
$\tau_{T'}(h_e) \leq \tau_{T'}(t_e)$	$e \in E(T')$
$\tau_T(v) - (1 - g_v)M < \tau_{T'}(v')$	$\forall v \in \dot{V}(T), T \in \mathcal{G}, v' = \text{MRCA}_{T'}(C_T(v))$
$(1 - g_v)\tau_T(v) + g_vM \geq \tau_{T'}(v')$	$\forall v \in \dot{V}(T), T \in \mathcal{G}, v' = \text{MRCA}_{T'}(C_T(v))$
$\tau_T(v) - (1 - \alpha_e)M < \tau_{T'}(t_e)$	$\forall v \in \dot{V}(T), T \in \mathcal{G}, \forall e \in E_v,$
$\tau_T(v) + (1 - \alpha_e)M \geq \tau_{T'}(h_e)$	$\forall v \in \dot{V}(T), T \in \mathcal{G}, \forall e \in E_v$
$g_v + \sum_{e \in E_v} \alpha_e = 1$	$\forall v \in \dot{V}(T), T \in \mathcal{G},$
$f_v - \sum_{e \in E_v} \ell_v(e) \cdot \alpha_e = 0$	$\forall v \in \dot{V}(T), T \in \mathcal{G},$
$g_v, \alpha_e \in \{0, 1\}$	$\forall v \in \dot{V}(T), T \in \mathcal{G}, \forall e \in E_v.$

Figure 3.3 : Algorithm ESTIMATEDIVERGENCETIMES. The complete ILP formulation for estimating the divergence times of a species tree topology  $T'$  given a set  $\mathcal{G}$  of gene trees with times at internal nodes. Solving this ILP yields the divergence time  $\tau_v$ , for every node  $v$  in the species tree  $T'$ .

COMPUTESPECIESTREE( $\mathcal{G}$ )				
1. $\mathcal{T} \leftarrow \text{ESTIMATESPECIESTREETOPOLOGY}(\mathcal{G});$				
2. best $\leftarrow \infty;$				
3. <b>for each</b> $T' \in \mathcal{T}$				
(a) ESTIMATEDIVERGENCETIMES( $T', \mathcal{G}$ );				
(b) best $\leftarrow \min\{\text{best}, \eta(T', \mathcal{G})\};$				
4. <b>end for</b>				
5. return the tree with the smallest $\eta$ value as the species tree;				

Figure 3.4 : Algorithm COMPUTESPECIESTREE( $\mathcal{G}$ ). The algorithm for computing the species tree topology and divergence times from an input set of gene trees with coalescence times at internal nodes  $\mathcal{G}$ .

Table 3.1 : Information of nine strains of the *Staphylococcus aureus* bacteria.

Refseq	<i>subsp. aureus</i> ~	Genome size (nt)	Annotated gene#	Reference
NC_002745	N315	2,814,816	2669	[74]
NC_002758	Mu50	2,878,529	2775	[75]
NC_002951	COL	2,809,422	2724	[76]
NC_002952	MRSA252	2,902,619	2845	[77]
NC_002953	MSSA476	2,799,802	2723	[77]
NC_003923	NW2	2,820,462	2712	[78]
NC_007622	RF122	2,742,531	2665	[79]
NC_007793	USA300	2,872,769	2648	[80]
NC_007795	NCTC 8325	2,821,361	2969	–

To identify orthologous genes, we used the information of both DNA sequence identity and synteny (gene order) as follows. All-against-all BLASTN search with default parameters [81] was performed for the genes in NC\_002745 v.s. all others. Then, we produced a list of BLASTN hits of the 2669 genes in NC\_002745 for each of the other strains. The lists include genes that have at least 90% sequence identity to the reference gene in NC\_002745 and the length of the BLASTN hit region covers more than 50% of the entire gene. We excluded BLASTN hits when there are more than one hit for each reference gene. As there were not many such cases, this restriction did not result in much loss of data.

In order to identify orthologous genes conservatively, we considered that orthologous genes should be in a large block of a region in which the gene order is well conserved for all investigated strains. A block is defined such that genes from all strains are continuously located on their genomes with less than three gene skips, which could be created by small indels and annotation errors. To detect such blocks, we performed a synteny survey from the first gene in NC\_002745 (NC\_002745.1) to downstream genes. Then, we identified 222 such blocks, which covered in total 1898 genes.

For each gene, we built a maximum parsimony (MP) tree from its DNA sequences by using PAUP\* 4.0 [82], and rooted the tree using the midpoint method. When the MP heuristic identified more than one tree for a given gene, we used the strict consensus of these trees. We inferred coalescence times at internal nodes in the gene trees using the formula

$$\tau_y = \frac{1}{|B(y)|} \times \left( \sum_{(a,b) \in B(y)} \frac{d_s(a,b)}{2r_s} \right) \quad (3.13)$$

for coalescence time of node  $y$  in a gene tree, where  $B(y) = \{(a, b) : \text{MRCA}(a, b) = y\}$ ,

$d_s$  is the number of synonymous substitutions per synonymous sites, and  $r_s$  is the rate of synonymous substitutions. In other words,  $\tau_y$  is the average of all coalescence times of every pair of genes whose MRCA is node  $y$ . Given that the rate of synonymous substitutions is similar across genes [83], this allowed us to compare the coalescence times across gene trees and use them to infer divergence times in the species tree. We used  $r_s = 10^{-8}$ , following the findings of [84].

It has been suggested that  $d_s$  may not be constant across the genome due to different codon bias among genes [85]. We found that  $d_s$  and the *codon adaptation index* (CAI) are in a negative correlation, therefore, we used a linear regression method to correct  $d_s$  for bias caused by non-random usage of codons. The correction is made such that a corrected  $d_s$  corresponds to that with the mean CAI. However, the corrected  $d_s$  measure did not change the relative times we obtained for the species trees.

To get the species tree candidates, we used Algorithm 2. Additionally, we considered five other candidate tree topologies:

1.  $T_{\text{conc}}$ : the tree topology obtained by the maximum parsimony heuristic, as implemented in PAUP\*, on the concatenation of all 1898 gene data sets;
2.  $T_{\text{hf}}$ : the topology of the gene tree that is compatible with the largest number of other gene trees (this tree, shown in Figure 3.9, is compatible with 1645 of the gene trees);
3.  $T_{\text{avgds}}$ : a tree topology built using the neighbor joining method [86] from the average  $d_s$  distances among nine strains;
4.  $T_{\text{avghd}}$ : a tree topology built using the neighbor joining method from the average Hamming distances among nine strains; and

5.  $T_{\text{majcons}}$ : the topology of the majority consensus tree of all 1898 gene trees.

In total, we have 29 candidate species tree topologies.

We then estimated the divergence times of each of the species tree topology candidates, using the CPLEX tool to solve ILP programs described in Algorithm 3.3. We have implemented a software tool for generating the ILP program from a set of gene trees with coalescence times, following the formulation in Algorithm 3.3, in the PhyloNet software package, which is available publicly at <http://bioinfo.cs.rice.edu/phyloNet/>. In the 9-genome data set that we considered in this study, each MILP program had approximately 4,000 variables and 30,000 constraints. Nonetheless, CPLEX solved each program in about one hour.

### 3.3.2 Results and Discussion

Our first task was to measure the “heterogeneity” in the data, which consisted of the  $9 \times 1898$  gene sequences and 1898 gene trees. In this task, we considered two measures of heterogeneity: topological differences among the gene trees, and distributions of coalescence times of each cluster of genes across all gene trees. Figure 3.5 shows the topological differences between every pair of the 1898 gene trees, as computed by the Robinson-Foulds (RF) measure [39]. The RF measure quantifies, for a given pair of trees, the average number of clades that appears in one, but not both, of the trees. Hence, if two trees are identical, the RF distance between them is 0; if they do not share any clades, then the RF distance is 1; and, trees with varying degrees of shared clades have RF distance values between 0 and 1.

As shown in Figure 3.5, while blue (low RF values) is the dominating color, there are many pairs of trees that have RF distance of at least 0.3. In fact, among the 1898 gene trees, there were over 400 different topologies. Given our conservative selection of

the orthology groups, which almost eliminates the possibility of gene tree discordance due to events such as horizontal gene transfer and gene duplication/loss, this result indicates massive gene tree discordance due to stochastic effects of incomplete lineage sorting.

Furthermore, it is important to point out that the majority of the gene trees were not binary, since the percent identity among the orthologous sequences was very high. This lack of resolution of the gene tree topologies may give a false indication of high concordance (low RF values) among the gene trees, even though this may not be the case in reality. Alternatively, one may quantify the “compatibility”, rather than “similarity” (as measured by the RF distance), among gene trees. However, this suffers from the fact that compatibility measures are not true metrics, and in particular do not satisfy the triangle inequality property, which may distort the picture emerging from such an analysis.

As illustrated in Figure 3.1, it may be the case the gene trees have the same topology, yet they disagree in their coalescence times (times at their internal nodes). Therefore, what we studied next was the distribution of coalescence times of each cluster of taxa across all gene trees in which the cluster occurs (recall that a cluster occurs in a tree if the tree contains a clade whose leaves are the only members of that cluster); the results are shown in Figure 3.6. The figure shows that, even with the exclusion of possible outliers, each cluster of taxa has a wide distribution of coalescence times across all gene trees in which it occurs. Further, what makes the computational analysis of such a data set particularly challenging is that large extent of overlap of distributions of the different clusters. Dealing with this overlap is where most of the computational time of solving our MILP formulation is spent.

After we characterized the heterogeneity in the data, we turned to the main issue:



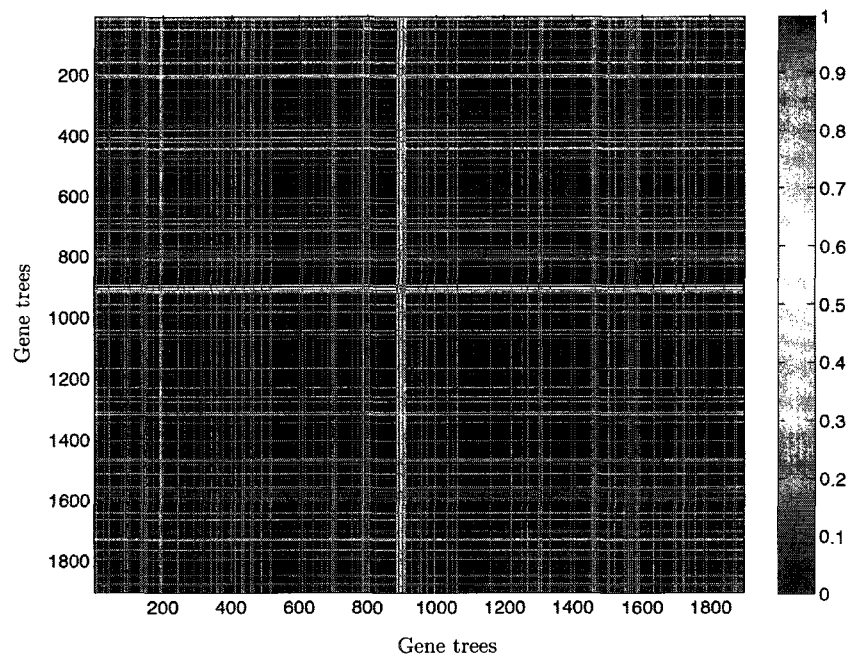


Figure 3.5 : The Robinson-Foulds (RF) distances between every pair of the 1898 gene trees. RF distance of 0 indicates the two trees are identical, and RF distance of 1 indicates that the two trees do not share any clades in common.

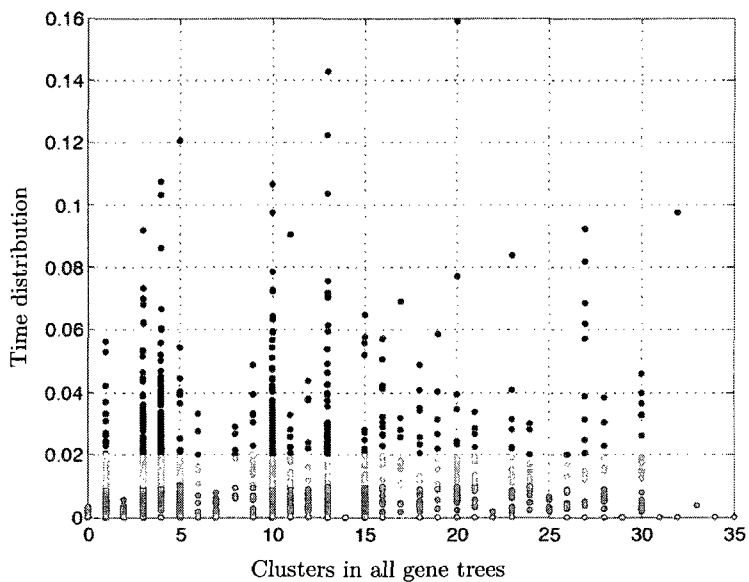


Figure 3.6 : The distributions of coalescence times of all 36 clusters of taxa in the 1898 gene trees, as calculated by Equation (3.13), but without division by  $r_s \approx 10^{-8}$ .

estimating the species tree topology and divergence times from the set of 1898 gene trees. As described in the previous subsection, we considered 29 species tree topology candidates. For each of these 29 topology candidates, we solved the ILP formulation as outlined in Algorithm 3.3, once with  $w_{dc} = w_{sc} = 1$ , and another with  $w_{sc} = 5w_{dc}$ . In both cases, the same tree topology candidate of all 24 maximal cliques emerged as the optimal one, yet with differing times. Therefore, we report the results of only the optimal solution under  $w_{dc} = w_{sc} = 1$ .

For a clearer presentation, we show each of the three terms in the optimality criterion described in Subsection 3.2.3 individually, with Figure 3.7 showing the number of missing (or, discordant) clades, and the stacked bars in Figure 3.8 showing the sum of the depths of deep coalescence events (the blue bars) and the number of shallow coalescence events (the red bars).

Figure 3.7 shows that the first tree out of the 24 maximal clique trees has the least disagreements with the set of 1898 gene trees, with trees 8 and 9 differing from it by about 70 clades. The other 21 maximal clique trees are much less optimal in this context, with the best of them disagreeing with the gene trees in at least 400 more clades. We denote by  $T_{\text{optm}}$  the first tree, which is the best in this context among all 24 maximal clique trees. Out of the additional five trees,  $T_{\text{hf}}$  is clearly the best in this context, and the only one that is better than  $T_{\text{optm}}$ . Both trees  $T_{\text{optm}}$  and  $T_{\text{hf}}$  are shown in Figure 3.9. The tree  $T_{\text{optm}}$  is a *refinement* of the tree  $T_{\text{hf}}$ ; that is,  $T_{\text{optm}}$  contains all the clades in  $T_{\text{hf}}$ , plus additional ones. In this case,  $T_{\text{hf}}$  has the clade (USA300, NCTC8325, COL) unresolved, while  $T_{\text{optm}}$  has it resolved as (NCTC8325, (USA300, COL)).

When considering the optimality of both trees,  $T_{\text{optm}}$  and  $T_{\text{hf}}$ , as measured by the cost of deep coalescence and shallow coalescence events, as shown in Figure 3.8, they

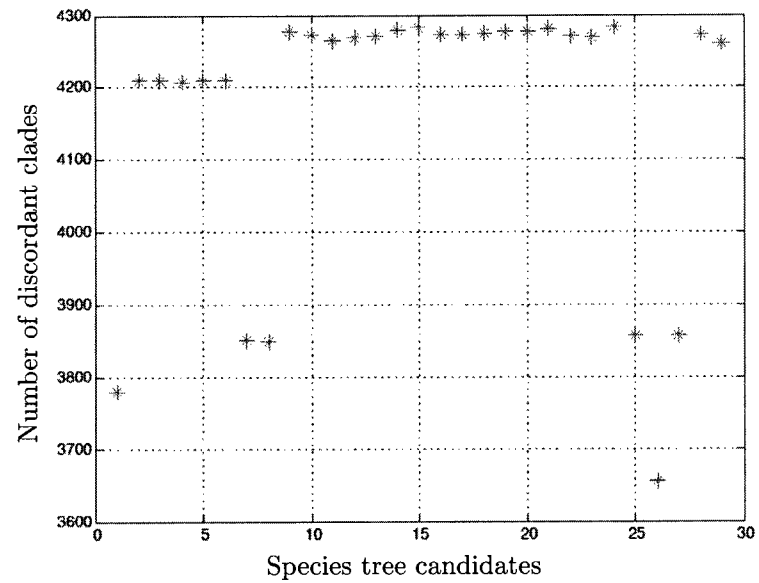


Figure 3.7 : The number of gene tree clades that do not appear in the species tree. Trees 1 to 24 are built from maximal cliques. The first 24 trees are built from the compatibility graph for  $\mathcal{G}$ , while trees 25, 26, 27, 28, and 29 are  $T_{\text{conc}}$ ,  $T_{\text{hf}}$ ,  $T_{\text{avgds}}$ ,  $T_{\text{avghd}}$ , and  $T_{\text{majcons}}$ , respectively.

are nearly identical. The significance of this result comes from the fact that, while the unresolved clade (USA300, NCTC8325, COL) has three possible refinements:

- (NCTC8325, (USA300, COL)),
- ((NCTC8325, USA300), COL),
- ((NCTC8325, COL), USA300),

the ILP formulation led to a fully binary species tree that has exactly the same combined cost of deep and shallow coalescence events.

We note that the majority consensus tree  $T_{\text{majcons}}$  is the optimal among all 29 trees in terms of the costs of deep and shallow coalescences. However, this tree has two problems. First, in terms of missing clades, it is one of the least optimal, as shown in Figure 3.7. Second, it is highly unresolved, containing only two internal branches, as shown in Figure 3.9.

For the concatenation tree  $T_{\text{conc}}$ , it is the best in terms of the cost of shallow coalescence events, yet the worst in terms of the cost of deep coalescence events. Further, it is the only tree that had the wrong outgroup. This indicates that concatenation of gene sequences and reconstructing a strain tree from the resulting “supergene” may result in very inaccurate trees, particularly when there is a massive extent of discordance among gene trees, a fact that has already been established through extensive experimental studies [31]. While it seems from Figure 3.9 that  $T_{\text{conc}}$  indicates very large divergence time between N315 and Mu50, this is but a reflection of time estimation given that these two strains did not form a single clade in the concatenation tree. To solve this problem, we will consider in future development of our tool all possible refinements of any non-binary strain tree topology candidate.

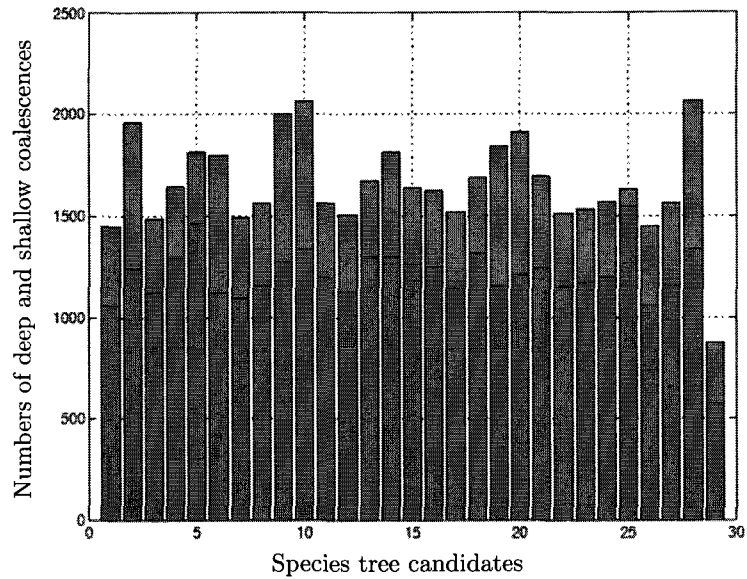


Figure 3.8 : The cost of deep coalescences,  $\sum_{T \in \mathcal{G}} \sum_{v \in \hat{V}(T)} f_v$ , and the cost of shallow coalescences,  $\sum_{T \in \mathcal{G}} \sum_{v \in \hat{V}(T)} g_v$ , for all 29 species tree candidates. The first 24 trees are built from the compatibility graph for  $\mathcal{G}$ , while trees 25, 26, 27, 28, and 29 are  $T_{\text{conc}}$ ,  $T_{\text{hf}}$ ,  $T_{\text{avgds}}$ ,  $T_{\text{avghd}}$ , and  $T_{\text{majcons}}$ , respectively.

The other two trees,  $T_{\text{avgds}}$  and  $T_{\text{avghd}}$  are very similar in terms of topology, as shown in Figure 3.9, and both fall “in the middle” in terms of optimality (Figures 3.7 and 3.8). Therefore, our proposed evolutionary history of all nine strains of *Staphylococcus aureus* is the tree  $T_{\text{optm}}$ , shown in Figure 3.9.

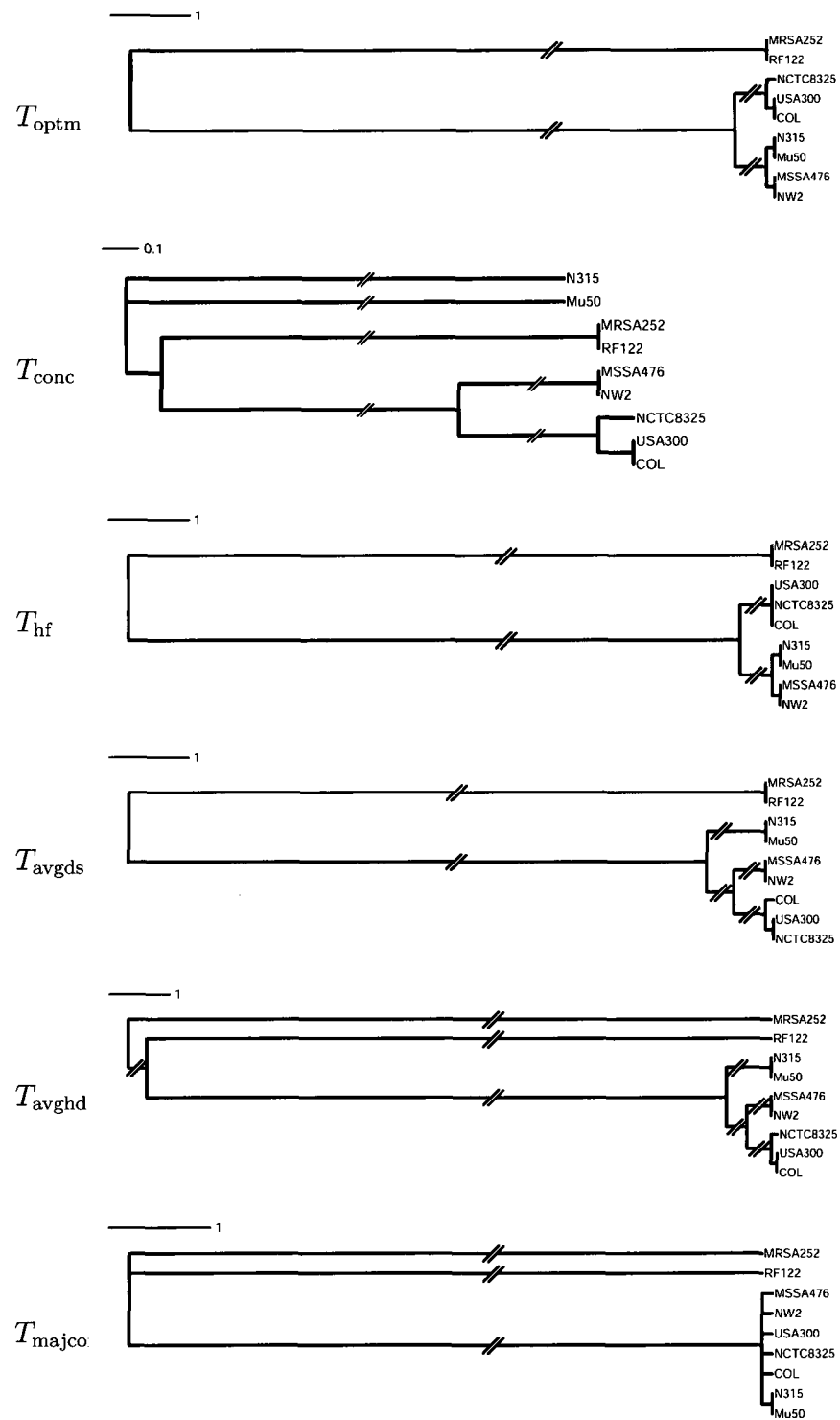


Figure 3.9 : Species trees with times assigned by Algorithm 3.3. The lengths of the “shortened” branches were divided by  $10^5$ , so that the resolution of the trees can be shown clearly.

## Chapter 4

# Species Tree Inference from Gene Trees Using Their Topologies Alone

In the previous chapter, we present a method for inferring the species tree, both the topology and divergence times of its internal nodes, from a set of gene trees. For assigning divergence times to the internal nodes of a species tree topology candidate, we introduced an optimality criterion that is a combination of the cost of deep and of shallow coalescence events; the cost of a deep coalescence event was defined simply as the number of edges that this event occurs deeper than its MRCA (also called its depth), while the cost of any shallow coalescence event was 1.

Maddison proposed another parsimony criterion, called *minimizing deep coalescences* (MDC), for inferring the species tree from multiple gene trees [5], when we also assume that the incongruence is exclusively due to lineage sorting. An empirical study in [34] shows that the criterion allows for reasonable recovery of species trees from phylogenetic signals in gene trees, despite the fact it makes no use of gene tree branch lengths. However, there have been so far heuristics for finding the tree minimizing deep coalescences, e.g. the one implemented in Mesquite [87]. This hinders a more comprehensive evaluation of the MDC criterion as well as its applicability to practice.

We recently devised two exact (i.e., guaranteed to find the optimal tree) and more efficient methods than the heuristic in Mesquite [88]. One method is also integer linear programming (ILP)-based, but unlike the method presented in the previous

chapter it does not require a separate phase that enumerates species tree topology candidates. Instead, it finds the tree minimizing deep coalescences directly from clusters, based on an observation that the MDC cost can be computed for individual clusters without the prior knowledge of the species tree. This observation also led to a more efficient dynamic programming algorithm. We describe the ILP-based method and the dynamic programming algorithm in Sections 4.4 and 4.5, respectively.

In Section 4.7, we show the performance of our algorithms. We analyzed a data set of 106 loci from eight yeast species [25], a data set of 268 loci from eight *Apicomplexan* species [29], and several simulated data sets. We show that the MDC criterion provides very accurate estimates of the species tree topologies, and that our methods are very fast, thus allowing for the accurate analysis of genome-scale data sets. We also show that searching for the species tree from clusters induced by input gene trees might be sufficient in practice, a finding that helps to ameliorate the computational requirements of computing the optimal tree. Further, we study the statistical consistency and convergence rate of the MDC criterion as well as its optimality in inferring the species tree.

## 4.1 Extra Lineages and Inferring the Species Tree by Minimizing Deep Coalescences

Maddison introduced the number of extra lineages to measure the severity of deep coalescences [5]. In this section, we review this concept, and then formalize it as it is necessary for the computation of this number and for the methods presented in Sections 4.4, 4.5. We also define the problem of inferring the species tree by *minimizing deep coalescences* (MDC).



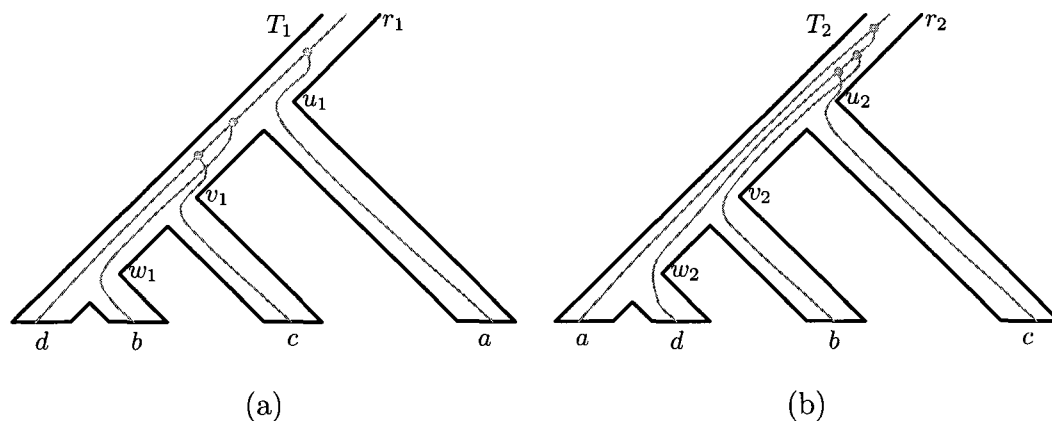


Figure 4.1 : Illustration of the concept of extra lineages. We are given a gene tree  $(a, (b, (c, d)))$ . Then, tree  $T_1 = (((d, b), c), a)$  requires one extra lineage to reconcile the gene tree within its branches, while tree  $T_2 = (((a, d), b), c)$  requires three extra lineages.

#### 4.1.1 Extra Lineages

Let us assume that we are given a gene tree  $(a, (b, (c, d)))$ , and we want to reconcile it within species tree  $T_1$  in Figure 4.1(a). In order to create clade  $(c, d)$  in the gene tree, lineages  $b$  and  $d$  must fail to coalesce on branch  $(v_1, w_1)$ . Those lineages can coalesce on any of branches  $(u_1, v_1)$  and  $(r_1, u_1)$ . However, as we ignore the branch lengths, we can force a coalescence event to occur as soon as possible. In this case, we require  $b$  and  $d$  coalesce on branch  $(u_1, v_1)$ , where  $v_1$  is their most recent common ancestor (MRCA) in  $T_1$ . Similarly, lineages  $b$ ,  $c$ , and  $d$  coalesce on this branch before they coalesce with  $a$  on branch  $(r_1, u_1)$ . After completing the reconciliation between the gene tree and  $T_1$ , we visit every internal branch of  $T_1$  and count the number of extra lineages as follows. In branch  $(v_1, w_1)$ , there are two lineages exiting it, and so we count the number of extra lineages as one. In a similar fashion, the numbers of extra lineages in  $(u_1, v_1)$  and  $(r_1, u_1)$  are 0. In total, the number of extra lineages required to reconcile the gene tree within  $T_1$  is  $1 + 0 + 0 = 1$ .

If instead we have the species tree  $T_2$  as in Figure 4.1(b), then in order to reconcile the gene tree lineages  $a$ ,  $d$  fail to coalesce along branch  $(v_2, w_2)$ , and they also fail to coalesce with each other and with lineage  $B$  on the branch  $(u_2, v_2)$ . All coalescence events occur on branch  $(r_2, u_2)$ . We now count the number of extra lineages for  $T_2$ : there is one in  $(v_2, w_2)$ , two in  $(u_2, v_2)$ , and 0 in  $(r_2, u_2)$ , and hence three in total. Maddison in his paper proposed that a tree with a smaller number of extra lineages is better than a tree with a larger number. For those two species trees  $T_1$  and  $T_2$ , we prefer  $T_1$  to  $T_2$ , since  $T_1$  needs one extra lineage while  $T_2$  needs three.

We now formalize the concept of extra lineages as it is necessary to devise a formula for counting the number of extra lineages. Suppose we are given a gene tree  $T$  and a species tree  $T'$ . Suppose further that both  $T$  and  $T'$  are binary and have the same set of leaves. The gene tree  $T$  is reconciled within the species tree  $T'$  by mapping each node of  $v$  in  $T$  according to three rules below:

1. Each taxon (labeled leaf) in  $T$  is mapped to the corresponding taxon in  $T'$ .
2. Let  $v' = \text{MRCA}_{T'}(C_T(v))$ , and let  $u'$  be the parent node of  $v'$ . Then,  $v$  is mapped to any point  $p_v$ , excluding node  $u'$ , in branch  $(u', v')$  in  $T'$ .
3. If  $w$  is a proper descendant of  $v$ , and  $w, v$  are mapped to  $p_w, p_v$  in  $T'$ , then  $p_w$  must also be a proper descendant of  $p_v$ .

Figure 4.2 shows an example of such a mapping. In the figure, we can see that for branch  $(u', v')$  there are two lineages, one being the lineage of the common ancestor of species  $a, b, c$ , and one being lineage  $d$ . In the case where  $T$  and  $T'$  are identical topologically, then we can easily see that there is only one lineage in  $(u', v')$ , that is one lineage for the common ancestor of  $a, b, c$  and  $d$ . Therefore, for the branch  $(u', v')$  in Figure 4.2, the number of extra lineages is  $2 - 1 = 1$ .

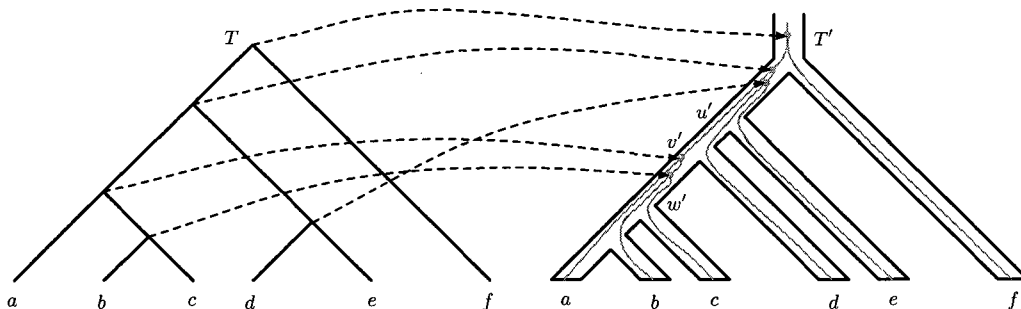


Figure 4.2 : Fitting a gene tree  $T$  into a species tree  $T'$ . In the figure, only mappings of internal nodes of  $T$  are shown, as each leaf in  $T$  is mapped to a leaf with the same label in  $T'$ .

**Definition 4.1** (Number of Extra Lineages). *Assuming that a gene tree  $T$  is mapped into a species tree  $T'$  according to the three rules above, the number of extra lineages in a branch of  $T'$  is defined to be the number of lineages exiting that branch minus one. The number of extra lineages required to reconcile  $T$  within  $T'$  is equal to the sum of the numbers of extra lineages in all branches of  $T'$ .*

Each  $p_v$  in  $T'$  that is the image of the mapping of an internal node  $v$  in  $T$  is a coalescence event. In Figure 4.2, there are two coalescence events in branch  $(v', w')$ , but there are no coalescent events in branch  $(u', v')$ . We can establish a relationship between the number of extra lineages and the number of coalescence events as follows. Consider a branch  $(u', v')$  of  $T'$ . There are exactly  $|C_{T'}(v')|$  species in the subtree  $T'(v')$ . If there were no coalescence among those species, then there would be  $|C_{T'}(v')|$  lineages exiting  $(u', v')$ . However, each coalescence event merges two lineages into one, and we note that under the mapping's conditions whenever there is a coalescence among lineages from species in  $C_{T'}(v')$ , it must occur either in a branch of  $T'(v')$  or in  $(u', v')$ . Therefore, the actual number of lineages exiting  $(u', v')$  is equal to  $|C_{T'}(v')|$  minus the total number of coalescence events among species in  $T'(v')$ . We have the

following lemma:

**Lemma 4.1.** *Let  $n(v')$  be the number of coalescence events occurring among species in  $C_{T'}(v')$ . Then, the number of extra lineages in branch  $(u', v')$  is*

$$|C_{T'}(v')| - n(v') - 1. \quad (4.1)$$

We note that this lemma may not be true without the conditions of the mapping defined above. If we do not have Rules 2 and 3, then lineages  $a$ ,  $b$ , and  $c$  in Figure 4.2, for example, need not coalesce in branch  $(v', w')$ . They can coalesce at a branch above  $u'$ , and in this case there are four lineages (and therefore, three extra ones instead of one) in  $(u', v')$ .

#### 4.1.2 Inferring the Species Tree under the MDC Criterion

Given a set of gene trees and a species tree topology candidate, we can compute the number of extra lineages this candidate tree requires to reconcile all the gene trees, which is considered as the parsimony for it. Inferring the species tree under the MDC criterion is to choose the most parsimonious tree as the species tree [5].

**Problem 4.1** (Species Tree Inference under the MDC Criterion).

**Input:** *A set of gene trees  $\mathcal{G}$ .*

**Output:** *A tree  $T'$  such that the total number of extra lineages required to reconcile all gene trees of  $\mathcal{G}$  within  $T'$  is minimized. The optimal tree  $T'$  is reported as the species tree.*

In Section 4.3, we prove a theorem that is fundamental to the methods for solving this problem that are described in Sections 4.4, 4.5.

## 4.2 Current Methods for Inferring the Species Tree under the MDC Criterion

We describe in this section two methods for solving Problem 4.1. One method is a brute-force algorithm, while the other is a heuristic implemented in Mesquite [87].

### 4.2.1 Brute-force Algorithm

The obvious way to find the tree minimizing deep coalescences is to compute the number of extra lineages for all possible species tree topology candidates, and choose the one requiring the smallest number of extra lineages as the species tree.

What is the complexity of this method? Computing the number of extra lineages for a pair of species tree and gene tree needs to find the MRCA in the species of all clusters in the gene tree. The problem of finding the MRCA in a tree can be solved in constant time, plus a preprocessing on the tree that takes linear time [89, 90]. Therefore, given a set  $\mathcal{G}$  of gene trees the complexity for computing the number of extra lineages for a candidate tree can be performed in  $O(n + |\mathcal{G}|n) = O(|\mathcal{G}|n)$ , where  $n$  is the number of leaves of the candidate tree (and also of a gene tree as we assume that both species tree and gene tree have the same leaf set). However, we note that there are  $(2n - 3)!!$  binary rooted trees whose leaves are labeled by an  $n$ -taxon set. Using Stirling's approximation, we have

$$\begin{aligned}
 (2n - 3)!! &= \frac{(2n - 2)!}{2^{n-1}(n - 1)!} \\
 &\sim \frac{\sqrt{2\pi(2n - 2)}((2n - 2)/e)^{2n-2}}{2^{n-1}\sqrt{2\pi(n - 1)}((n - 1)/e)^{n-1}} \\
 &\sim \sqrt{2} \left(\frac{2n - 2}{e}\right)^{n-1}.
 \end{aligned} \tag{4.2}$$

Therefore, the complexity of the brute-force method is  $O(((2n - 2)/e)^{n-1}|\mathcal{G}|n)$ , which

implies that the method is infeasible for trees with  $n \geq 10$  leaves.

#### 4.2.2 Mesquite’s Heuristic

Mesquite also implemented a heuristic for finding the optimal tree under the MDC criterion [34, 87]. The heuristic works as follows. It starts with a random tree, and computes the MDC cost for it. The heuristic then applies an SPR operation to it to obtain a new tree, and recomputes the MDC cost for the newly obtained tree. The heuristic records the best MDC score computed so far, and it stops when applying an SPR to a tree does not improve that score. As stated in [34], this method does not guarantee to compute the optimal tree. It is also quite slow since there are about  $O(n^2)$  possible SPR moves that can be applied to a tree. Further, it requires the computation of the MDC cost for every derived tree.

### 4.3 Counting the Number of Extra Lineages

In computing the number of extra lineages for reconciling a gene tree  $T$  within the branches of a species tree  $T'$  based on Definition 4.1, we map each node of  $T$  to a node in  $T'$  using the MRCA mapping. In this way, we need to know  $T'$ . However, we show that the number of extra lineages in a branch of  $T'$  depends only on the cluster it induces (and on  $T$ )—it does not depend on where this branch is placed in  $T'$  nor on the shape of  $T'$  (provided that this branch is present in  $T'$ ). This result implies that we can compute the number of extra lineages for each individual cluster *without* a prior knowledge of the species tree as in the standard way of computing this number based on Definition 4.1.

The theorem we state in this section makes use of the notion of a maximal clade with respect to a given cluster. Let us be given a tree  $T$  and a cluster  $A$ . We call a

clade  $t$  of  $T$  maximal with respect to  $A$  if: (1)  $\mathcal{L}(t) \subseteq A$ ; and (2)  $t$  is not a proper subtree of another tree  $t'$  such that  $\mathcal{L}(t') \subseteq A$ .

**Theorem 4.1** (Number of Extra Lineages for a Cluster). *Let  $T$  and  $T'$  be a gene tree and  $T'$  be species tree, respectively, and let  $(u', v')$  be a branch of  $T'$ . Further, let  $k$  be the number of clades of  $T$  that are maximal with respect to  $C_{T'}(v')$ . Then, the number of extra lineages in branch  $(u', v')$ , which we denote as  $\alpha(C_{T'}(v'), T)$ , is:*

$$\alpha(C_{T'}(v'), T) = k - 1. \quad (4.3)$$

*Proof.* Let us denote those  $k$  maximal clades of  $T$  as  $t_1, \dots, t_k$ . Consider a clade  $t_i$ ,  $1 \leq i \leq k$ . First of all, because  $t_i$  is clade of  $T$  all species in  $t_i$  must coalesce into a single lineage (and they must coalesce either in a branch of  $T'(v')$  or  $(u', v')$  under the mapping's conditions in Section 4.1). Second, because  $t_i$  is a maximal clade of  $T$  with respect to  $C_{T'}(v')$ , that lineage will not coalesce with any other lineages in  $T'(v')$  or in branch  $(u', v')$  (for otherwise, we will obtain a bigger clade in  $T$  whose leaf set is still a subset of  $C_{T'}(v')$ , a contradiction). By Lemma 4.1, the number of coalescence events occurring among species of  $t_i$  is  $|\mathcal{L}(t_i)| - 1$ . We also note that  $\bigcup_{i=1}^k \mathcal{L}(t_i) = C_{T'}(v')$ . So, by applying this lemma again, we obtain

$$\begin{aligned} \alpha(C_{T'}(v'), T) &= |C_{T'}(v')| - \sum_{i=1}^k (|\mathcal{L}(t_i)| - 1) - 1 \\ &= k - 1. \end{aligned}$$

□

As an example, consider trees  $T$  and  $T'$  in Figure 4.2. From the figure, we see that there are no extra lineages in branch  $(v', w')$ . The cluster under  $w'$  is  $\{A, B, C\}$ . The clade  $(A, (B, C))$  is a maximal clade of  $T$  with only species from  $\{A, B, C\}$ . Therefore,

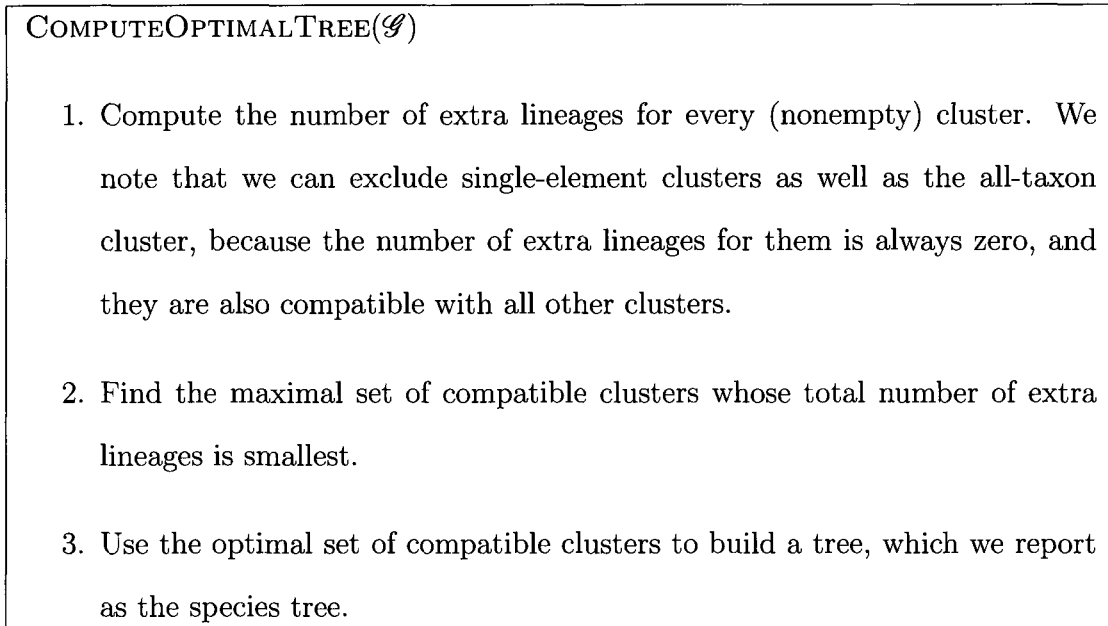


Figure 4.3 : Algorithm COMPUTEOPTIMALTREE. An approach to find the optimal tree for a set of gene trees  $\mathcal{G}$ . Note that all (nonempty) clusters are used to find the optimal tree.

the number of extra lineages is  $1 - 1 = 0$ . On the other hand, consider branch  $(u', v')$ . There are two maximal clades in  $T$  with species from  $\{A, B, C, D\}$ :  $(A, (B, C))$  and  $D$ . So, the number of extra lineages in  $(u', v')$  is  $2 - 1 = 1$ .

#### 4.4 Inferring Species Trees: An ILP Approach

Using Theorem 4.1, we can solve Problem 4.1 by finding a maximal set of compatible clusters whose total number of extra lineages is smallest. The reason for seeking a maximal set of compatible clusters is that such a set defines a rooted binary tree. (We clearly do not want to choose a star tree as a species tree although it requires zero extra lineages to reconcile a gene tree.) Therefore, we propose an approach to solving (exactly) Problem 4.1 as in Figure 4.3.



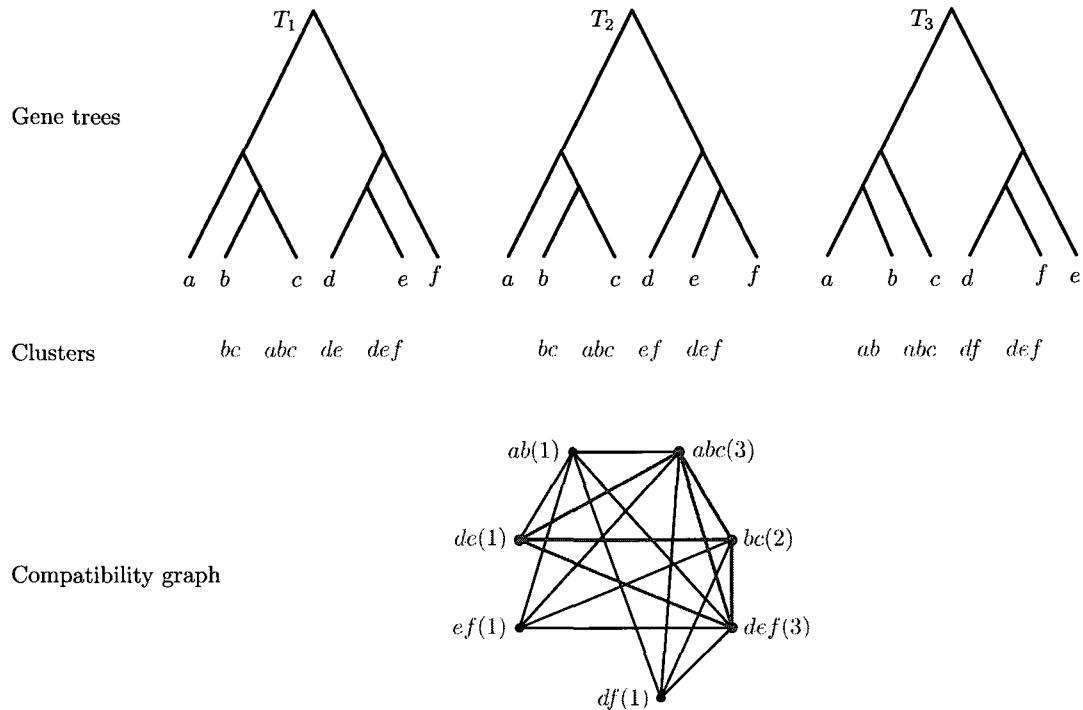


Figure 4.4 : Compatibility graph constructed from three gene trees  $T_1$ ,  $T_2$ , and  $T_3$ . A maximum vertex-weighted clique consisting of clusters  $\{bc\}$ ,  $\{abc\}$ ,  $\{de\}$ ,  $\{def\}$  is highlighted.

In Section 4.7, we show that the percentage of species tree clusters that are not present in any of input gene trees is negligible, and decreases as more gene trees are available. A species tree can, therefore, be recovered from clusters induced by gene trees, and hence, we can focus on those gene tree clusters, instead of working with the set of all possible clusters. In the following subsections, we describe how to find the optimal tree composed of only gene tree clusters by using an ILP formulation. We see that this approximation produces very accurate estimates of species trees, even though in some cases it might not return the actual optimal tree when all clusters are used (Section 4.8).

#### 4.4.1 Constructing the Weighted Compatibility Graph

Given that a collection of pairwise compatible clusters uniquely defines a tree, we construct the *compatibility graph*  $G$  of all clusters and focus on the cliques in this graph. Let  $\mathcal{C}$  be the collection of all clusters of a set  $\mathcal{G}$  of gene trees. The vertex set of  $G$  represents clusters in  $\mathcal{C}$ . Two vertices are adjacent if the two corresponding clusters are compatible. Since we seek the clique that is simultaneously maximal in terms of size and minimizes the amount of deep coalescence events, we assign weights to the vertices of  $G$  in a special way. Let  $v$  be a vertex in the graph  $G$  and let  $C$  be the cluster it represents. For each gene tree  $T \in \mathcal{G}$ , we count the number of extra lineages contributed by  $A$  as in Equation (4.3). In total, cluster  $C$  contributes  $\sum_{T \in \mathcal{G}} \alpha(C, T)$  extra lineages. Let  $m$  be the maximum value of  $\sum_{T \in \mathcal{G}} \alpha(C, T)$  over all  $A \in \mathcal{C}$ . We assign vertex  $v$  the weight

$$w(v) = m + 1 - \sum_{T \in \mathcal{G}} \alpha(C, T). \quad (4.4)$$

The reason we define  $w(v)$  in this manner, instead of  $\sum_{T \in \mathcal{G}} \alpha(C, T)$ , will be clear next, where we describe an efficient ILP formulation for identifying the clique in the compatibility graph that corresponds to a tree that minimizes the MDC cost of all coalescence events.

Let us illustrate the construction of  $G$  from three gene trees  $T_1$ ,  $T_2$ , and  $T_3$  in Figure 4.4. Those three gene trees induce seven clusters,  $\{ab\}$ ,  $\{bc\}$ ,  $\{abc\}$ ,  $\{de\}$ ,  $\{df\}$ ,  $\{ef\}$ , and  $\{def\}$ . Next, we compute the number of extra lineages for each of them, and compute the weight according to Equation 4.4; those numbers are given in Table 4.1. The corresponding compatibility graph  $G$  constructed from those clusters is shown in Figure 4.4.

Table 4.1 : The number of extra lineages for each of seven clusters induced by gene trees  $T_1$ ,  $T_2$ , and  $T_3$  in Figure 4.4. The last column is the weight assigned to vertices in the compatibility graph according to Equation 4.4, where  $m = 2$ .

cluster	$\alpha(\cdot, T_1)$	$\alpha(\cdot, T_2)$	$\alpha(\cdot, T_3)$	$w(\cdot)$
$\{ab\}$	1	1	0	1
$\{bc\}$	0	0	1	2
$\{abc\}$	0	0	0	3
$\{de\}$	0	1	1	1
$\{df\}$	1	1	0	1
$\{ef\}$	1	0	1	1
$\{def\}$	0	0	0	3

#### 4.4.2 Finding the Optimal Tree in the Compatibility Graph

A clique in the compatibility graph  $G$  defines a tree, and we seek a clique in  $G$  such that, on one hand, it has as many vertices as possible (to obtain maximal resolution of the species tree), and on the other hand, the number of extra lineages contributed by its vertices, as defined above, is as small as possible. The way we assign weights to vertices of the compatibility graph  $G$  allows us to achieve both goals simultaneously.

In the compatibility graph  $G$ , we will find a maximum vertex-weighted clique. This clique is clearly a maximal one, because each vertex  $v$  is assigned a positive weight by function  $w(v)$  in Equation (4.4), which will guarantee having the maximal number possible of compatible clusters in the species tree. Moreover, because we maximize the clique weight, by the definition of function  $w(v)$ , we in fact minimize the total number of extra lineages (among all cliques of the same size). Finding a maximum vertex-weighted clique in a graph can be converted to a linear programming

formulation [91]:

$$\begin{aligned}
& \text{maximize} && \sum_{v \in V(G)} w(v)x_v, \\
& \text{subject to} && x_u + x_v \leq 1, \forall (u, v) \notin E(G), \\
& && x_v \in \{0, 1\}, \forall v \in V(G).
\end{aligned} \tag{4.5}$$

This formulation allows us to solve our problem by using CPLEX. From empirical observations, we find that the compatibility graph  $G$  is often very sparse. Therefore, the above formulation results in a very large number of constraints  $x_u + x_v \leq 1$ . The following method can reduce the number of constraints to exactly  $|V(G)|$ . For a vertex  $u \in V(G)$ , let  $N(u)$  be the set of vertices that are adjacent to  $u$ . The constraint

$$|V(G) \setminus N(u)| \times x_u + \sum_{v \notin N(u)} x_v \leq |V(G) \setminus N(u)|$$

means that if  $u$  is included in the clique (i.e.,  $x_u = 1$ ), then no vertices in  $G$  that are not adjacent to  $u$  are included in the clique (all  $x_v$ 's not in  $N(u)$  are 0), and that if any of those vertices is included in the clique, then  $u$  cannot be in the clique (i.e.,  $x_u$  must be 0). Therefore, the above linear programming formulation is equivalent to

$$\begin{aligned}
& \text{maximize} && \sum_{v \in V(G)} w(v)x_v, \\
& \text{subject to} && |V(G) \setminus N(u)| \times x_u + \sum_{v \notin N(u)} x_v \leq |V(G) \setminus N(u)|, \forall u \in V(G), \\
& && x_v \in \{0, 1\}, \forall v \in V(G).
\end{aligned} \tag{4.6}$$

## 4.5 Inferring Species Trees: A DP Algorithm

We can find the optimal species tree without the need to find a maximum vertex-weighted clique in the compatibility graph  $G$  by employing dynamic programming.

Dynamic programming (DP) is a divide-and-conquer algorithmic technique that breaks a problem into sub-problems, solves the sub-problems, and then uses those solutions in an efficient way to form the solution to the main problem. For a problem to be amenable to a DP solution, it must exhibit an optimal substructure [92].

Let  $t'$  be a rooted binary phylogenetic tree on a fixed taxon subset  $C = \mathcal{L}(t')$  of  $X$ . Given a collection  $\mathcal{G}$  of gene trees, let us denote  $l(t', \mathcal{G})$  the sum of  $\sum_{T \in \mathcal{G}} \alpha(B, T)$  for all clusters  $B$  in  $t'$ , including  $C$ . Further, let  $l^*(C, \mathcal{G})$  be the minimum value of  $l(t', \mathcal{G})$  over all possible binary trees  $t'$  on  $C$ . If  $t'_1$  and  $t'_2$  are the two subtrees whose roots are the children of  $t'$ , then clearly we have

$$l(t', \mathcal{G}) = l(t'_1, \mathcal{G}) + l(t'_2, \mathcal{G}) + \sum_{T \in \mathcal{G}} \alpha(C, T). \quad (4.7)$$

The quantity  $\sum_{T \in \mathcal{G}} \alpha(C, T)$  is fixed for each  $C$ , and therefore, if  $t'$  is an optimal tree on  $C$  such that  $l(t', \mathcal{G})$  is minimum, then  $l(t'_1, \mathcal{G})$  and  $l(t'_2, \mathcal{G})$  must also be minimum. This optimal substructure allows us to compute  $l^*(C, \mathcal{G})$  recursively as in Algorithm 4.5.

**Remarks.** Although the algorithm described above only returns the number of extra lineages, we can easily modify it so that we can actually reconstruct the optimal species tree. For each  $i$ ,  $3 \leq i \leq |X|$ , in Step 3, we also record two pointers to optimal subclusters  $C_1$  and  $C_2$ . By backtracking those pointers starting with cluster  $X$ , we can obtain the optimal set of compatible clusters.

Any tree  $T \in \mathcal{G}$  induces exactly  $|X| - 2$  nontrivial clusters. Therefore,  $|\mathcal{C}| = O(|\mathcal{G}| \cdot (|X| - 2))$ . For every  $C \subseteq X$ , there are at most  $|\mathcal{C}|$  subsets of  $C$  to look at, and hence Step 3 is executed at most  $|\mathcal{C}|^2$  times. The running time of the algorithm is then  $O(|\mathcal{G}|^2 \cdot (|X| - 2)^2)$ .

**DP-SPECIES TREE INFERENCE( $\mathcal{G}$ )**

1. Let  $\mathcal{C}$  be a collection of nontrivial clusters induced by trees in  $\mathcal{G}$  plus cluster  $X$  and all single-element clusters. We partition  $\mathcal{C}$  into subsets  $\mathcal{C}_1, \dots, \mathcal{C}_{|X|}$ , where  $\mathcal{C}_i$ ,  $1 \leq i \leq |X|$ , is the collection of all clusters of size  $i$  in  $\mathcal{C}$ .

2. **for each**  $C \in \mathcal{C}_1$ , set  $l^*(C, \mathcal{G}) = 0$ ,

3. **for each**  $C \in \mathcal{C}_2$ , set  $l^*(C, \mathcal{G}) = \sum_{T \in \mathcal{G}} \alpha(C, T)$ .

4. **for each**  $C \in \mathcal{C}_i$ ,  $3 \leq i \leq |X|$ , set

$$l^*(C, \mathcal{G}) = \min \{l^*(C_1, \mathcal{G}) + l^*(C_2, \mathcal{G}) : C_1 \cap C_2 = \emptyset \text{ and } C = C_1 \cup C_2\} \\ + \sum_{T \in \mathcal{G}} \alpha(C, T).$$

5. **return**  $l^*(X, \mathcal{G})$ .

Figure 4.5 : Algorithm DP-SPECIES TREE INFERENCE.

The collection  $\mathcal{C}$  described in the algorithm only contains clusters induced by gene trees in  $\mathcal{G}$ . However, we can replace it by the collection of all nonempty subsets of  $X$  (there are  $2^{|X|} - 1$  such subsets). In this case, the running time of the algorithm is bounded by  $\sum_{i=0}^{|X|} \binom{|X|}{i} 2^i = 3^{|X|}$ . Although it is exponential, it is significantly better than a brute-force approach that examines all  $(2^{|X|} - 3)!!$  binary rooted phylogenetic trees on  $X$ .

## 4.6 Extra Lineages for Non-binary and Multiple-Allele Gene Trees

Thus far, we have discussed the MDC criterion and presented algorithms for finding the optimal tree under this criterion only for the case when gene trees are binary, and have exactly a single individual (or, allele) per locus per species. We now discuss how the MDC criterion, and algorithms, can be extended for the case of multiple individuals and/or non-binary trees.

### 4.6.1 Multiple Individuals per Species

Suppose that we sample more than one individual per species when reconstructing a gene tree. We can extend the MDC criterion as follows. All taxa in the gene trees are considered distinct, even if they are from the same species. When fitting the gene tree into the species tree, we simply draw as many lineages originated backwards from a species as the number of individuals sampled for that species, and the remaining process is carried out in a similar manner as in [5]. For instance, consider the species tree and gene tree in Figure 4.6. There are three species  $a$ ,  $b$  and  $c$ , and for species  $a$ , we sample two individuals, represented as  $a_1$  and  $a_2$  in the gene tree. Because we sample two individuals for  $a$ , there are two lineages within the branch incident with

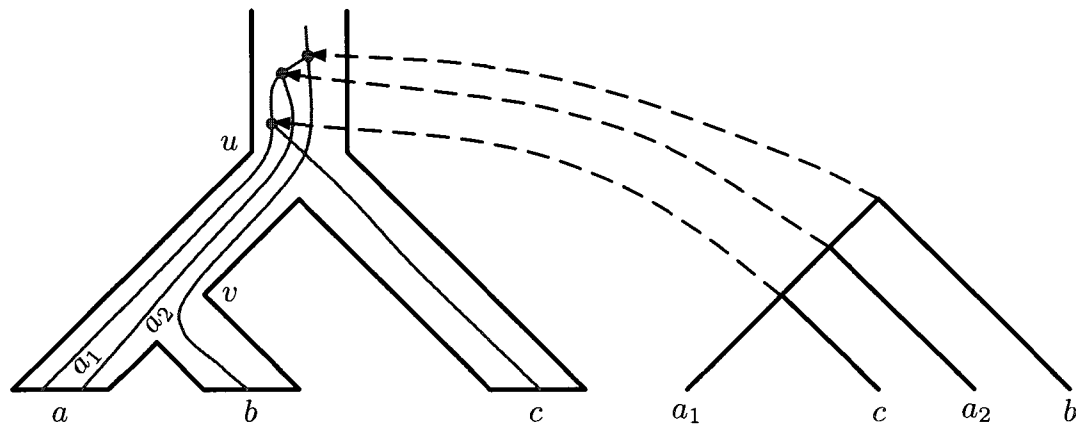


Figure 4.6 : MDC for gene trees with multiple alleles/individuals. On the left, the species tree is shown in tubes, while the thin lines show how the gene tree, on the right, is fitted within the branches of the species tree. On the right, a gene tree with four leaves, two of which correspond to two individuals of species  $A$ .

the leaf  $a$ . As we trace the evolution backwards in time, we find that  $a_1$  coalesces first with  $c$ , then with  $a_2$ , and finally with  $b$ . All of those coalescence events occur on the branch incident into the root of the species tree. For this example, there is one extra lineage on the branch incident with the leaf  $a$ , and two extra lineages on the branch  $(u, v)$ , accounting for a total of three extra lineages.

#### 4.6.2 Non-binary Trees

The extension of the MDC criterion for non-binary trees is quite straightforward. A non-binary node (a node with out-degree higher than 2) in the gene tree indicates that the lineages in the subtree rooted at that node all coalesce together. Fitting a gene tree into a species tree can be carried out in exactly the same way as in [5]. Figure 4.7 provides an illustration. Here, lineages from  $a$ ,  $b$ , and  $d$  fail to coalesce along the branch  $(u, v)$ , resulting in  $3 - 1 = 2$  extra lineages on that branch. We note here that a non-binary node in the species tree does not affect the way we count the



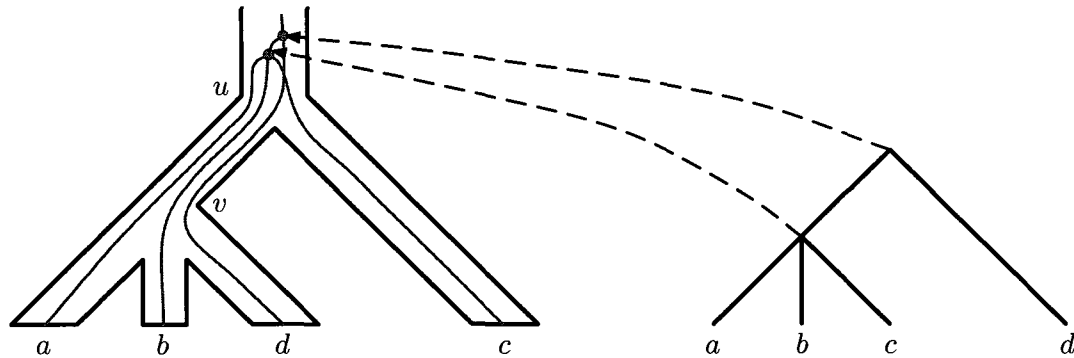


Figure 4.7 : MDC for non-binary trees. On the left, the species tree is shown in tubes, while the thin lines show how the non-binary gene tree, on the right, is fitted within the branches of the species tree.

number of extra lineages on the branch incident into it. In this example, we have a node with out-degree three in the species tree corresponding to the cluster  $\{a, b, d\}$ . In the gene tree, we have exactly three subtrees  $(a)$ ,  $(b)$  and  $(d)$  such that their leaf sets are subsets of  $\{a, b, d\}$ .

## 4.7 Experimental Verification

To study the performance of our algorithms and the MDC criterion, we analyzed biological as well as synthetic data sets. For the biological data, we used two data sets: the *Apicomplexan* data set of [29] and the yeast data set of [25]. The *Apicomplexan* data set contains eight species: *Babesia bovis* (Bb), *Cryptosporidium parvum* (Cp), *Eimeria tenella* (Et), *Plasmodium falciparum* (Pf), *Plasmodium vivax* (Pv), *Theileria annulata* (Ta), *Toxoplasma gondii* (Tg), and *Tetrahymena thermophila* (Tt). Kuo *et al.* identified 268 single-copy genes suitable for phylogenetic inference [29]. For each gene, they reconstructed its tree using three methods (maximum parsimony, maximum likelihood, and neighbor joining). Among the 268 gene trees, there were 48 different gene-tree topologies, the most frequent of which appears with about

18% frequency. They inferred the species tree using two different methods: the concatenation method and the majority consensus method, both of which produced the same tree, shown in Figure 4.8, which the author presented as their hypothesis for the species tree of these eight *Apicomplexan* species.

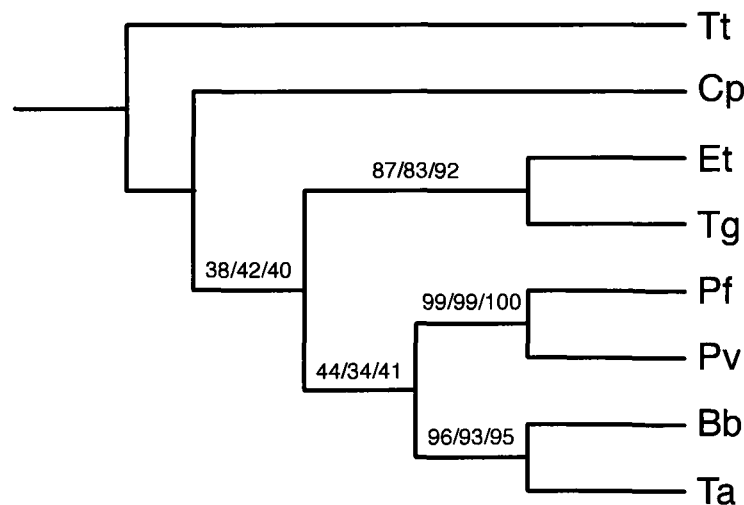


Figure 4.8 : The species tree for the *Apicomplexan* data as inferred using the majority consensus method and reported in [29]. The species *Tt* (*Tetrahymena thermophila*) is the outgroup. The numbers on the tree branches are bootstrap support values based on maximum likelihood, maximum parsimony and neighbor joining methods, respectively.

The yeast data set contains seven *Saccharomyces* species *S. cerevisiae* (Scer), *S. paradoxus* (Spar), *S. mikatae* (Smik), *S. kudriavzevii* (Skud), *S. bayanus* (Sbay), *S. castellii* (Scas), *S. kluyveri* (Sklu), and the outgroup fungus *Candida albicans* (Calb). Rokas *et. al.* [25] identified 106 genes, which are distributed throughout the *S. cerevisiae* genome on all 16 chromosomes and comprise about 2% of predicted genes. For each gene, they reconstructed its tree using the maximum likelihood and maximum parsimony methods. Among the 106 trees, more than 20 different gene-tree topologies were observed. They inferred the species tree using the concatenation

method on the the sequences of the 106 genes. The resulting tree had 100% bootstrap support for each of its branches, and the tree topology is shown in Figure 4.9.

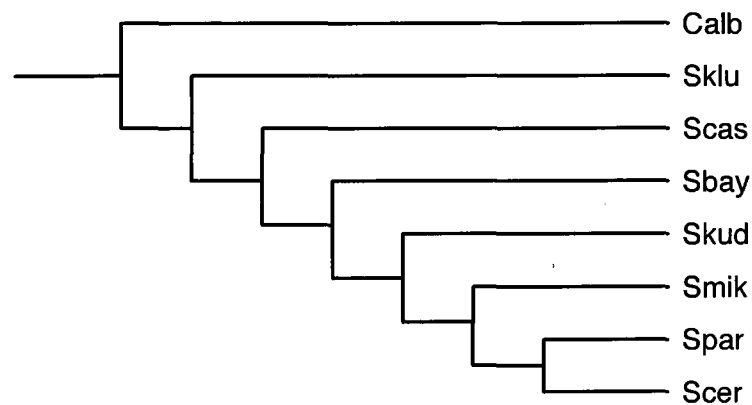


Figure 4.9 : The species tree for the yeast data set as inferred using the concatenation method and reported in [25]. All branches in the tree have 100% support values.

We generated synthetic data set by using Mesquite [87], and the same procedure and parameters in [34]. Species trees were simulated by using the “Uniform Speciation” (Yule) module in Mesquite. Two sets species trees were generated: one for those with a total branch length of 100,000 ( $1N_e$ ) generations, and one for 1,000,000 ( $10N_e$ ) generations. Each data set has 500 species trees. Within the branches of each species tree, the script generated 1, 3, 9, or 27 gene trees using the module “Coalescence Contained within Current Tree” with the effective population size  $N_e$  equal 100,000. For each gene tree, 1, 3, 9, or 27 alleles (individuals) were sampled per species. Since the species tree is known for simulated data, we studied the performance of our methods and the MDC criterion by comparing the inferred species tree against the true species tree. For this comparison, we used the normalized Robinson-Foulds (RF) measure [39], which quantifies the average proportion of branches present in one, but not both, of the trees. A value 0 of the RF distance indicates the two trees are iden-

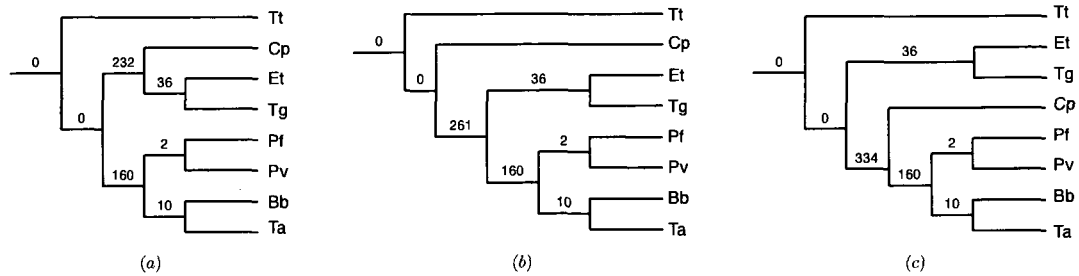


Figure 4.10 : (a) The optimal (species) tree inferred by our method for the *Apicomplexan* data set; this tree requires 440 deep coalescences to reconcile all 268 gene trees. The two sub-optimal species trees with 469 and 542 deep coalescences are shown in (b) and (c), respectively. The value on each branch is the numbers of extra lineages within that branch, when reconciling all 268 gene trees.

tical, and a value of 1 indicates the two trees and completely different (they disagree on every branch).

#### 4.7.1 Analysis of the *Apicomplexan* Data Set

Applying our method to the *Apicomplexan* data set, by using the 268 gene trees reported by Kuo *et. al.* [29], there was a single optimal tree, which is shown in Figure 4.10(a). The inferred tree requires in total 440 extra lineages to reconcile all 268 gene trees. This tree differs from their tree (Figure 4.8) with respect to only the single clade  $(Cp, (Et, Tg))$ . As Figure 4.8 shows, their tree places  $Cp$  as a sibling of the clade  $((Et, Tg), ((Pf, Pv), (Bb, Ta)))$ . However, it is important to note that as the authors reported, this placement of  $Cp$  has very low bootstrap support values of 38, 42, and 40 based on maximum likelihood, maximum parsimony and neighbor joining methods, respectively. Therefore, this grouping is not well-supported, even though both the concatenation and majority consensus methods compute it. Our method differed by placing  $Cp$  as a sibling of the clade  $(Et, Tg)$ . In fact, this grouping was advocated by [93].

To investigate this data set further, and particularly the placement of *Cp*, we employed our methods in an exploratory mode: we computed all maximal cliques in the compatibility graph of this data set, and for each maximal clique it computed the optimal fitting of all gene trees by minimizing the deep coalescences. The compatibility graph has 37 vertices (which means there are 37 different clusters induced by all gene trees) and 297 edges. In this graph, there are 247 maximal cliques, all of which have 6 vertices. This allows us to construct 247 fully binary species tree candidates. Figure 4.11 plots the number of extra lineages for all 247 species tree candidates, sorted from the lowest (which is the optimal one with 440 extra lineages) to the least optimal, which is a maximal clique requiring about 2200 extra lineages to reconcile all gene trees.

We observed that next to the optimal maximal clique with 440 extra lineages, the next two sub-optimal maximal cliques within 100 lineage counts from the optimal one had 469 and 542 extra lineages, respectively. In other words, in addition to the optimal maximal clique, whose corresponding species tree is shown in Figure 4.10(a), there were two additional trees very close in terms of the optimality criterion (minimizing deep coalescences). These two trees are shown in Figure 4.10(b) and 4.10(c). It is worth noting that the tree in Figure 4.10(b) is exactly the tree reported in [29], and that the tree in Figure 4.10(c) is the third way to group *Cp*,  $(Et, Tb)$  and  $((Bb, Ta), (Pf, Pv))$ . In other words, while our method identified a single optimal tree, this tree along with the two close sub-optimal trees differ from each other by the placement of *Cp*. This fact is already reflected in the community by having two different hypotheses about this placement reported by [93] and [29].

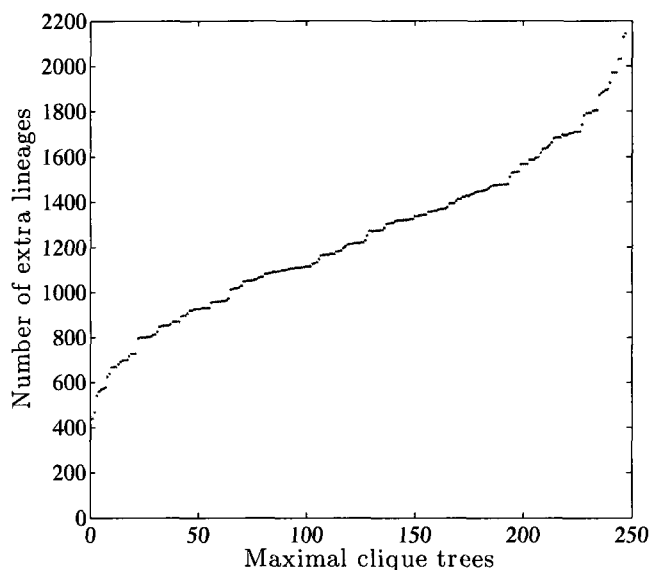


Figure 4.11 : Plot of the number of extra lineages for each of the binary (fully resolved) 247 species tree candidates identified as maximal cliques in the compatibility graph of the gene trees. The first three lowest values are 440, 469 and 542. The trees corresponding to these numbers are shown in Figure 4.10, respectively.

#### 4.7.2 Analysis of the Yeast Data Set

The yeast data set contains 106 genes from eight species, with massive discordance among the gene trees, as reported by [25]. The authors concatenated all gene sequences and ran maximum likelihood and maximum parsimony methods to reconstruct the species tree, and produced a species tree all of whose branches had 100% bootstrap support; this tree is shown in Figure 4.9.

For our analysis, we reconstructed the gene trees using a maximum parsimony heuristic, and ran our method on them to infer the species tree. There was a single optimal tree found by our method, which is shown in Figure 4.12(a). Clearly, the tree is identical to the one reported by [25]. This tree requires 127 extra lineages to reconcile all 106 gene trees. Edwards *et al.* also reported the same species tree using

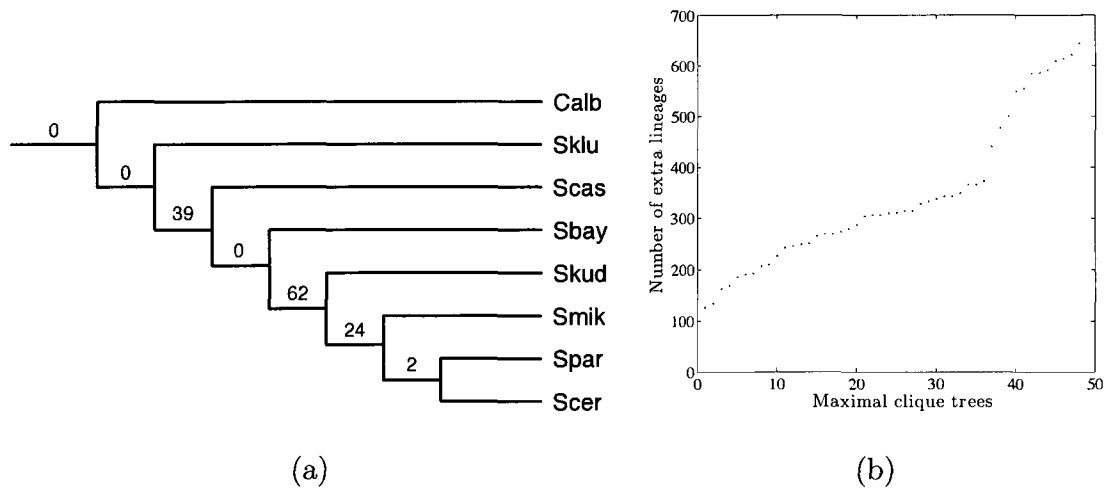


Figure 4.12 : (a) The species tree inferred by our method for the yeast data set. The values on its branches are the numbers of extra lineages within them. (b) Plot of the number of extra lineages for all 48 species tree candidates

the tool Bayesian Estimation of Species Trees (BEST) [32, 33]. However, while our method took a fraction of a second to infer this species tree, the BEST tool took several days.

As we did with the *Apicomplexan* data set, we also generated all species tree candidates from the compatibility graph built from gene trees. The compatibility graph for this yeast data has 17 vertices and 94 edges. We then built 48 binary trees from the 48 maximal cliques in the compatibility graph, and scored the minimum number of deep coalescences required to reconcile all gene trees with each of the trees; these values are shown in Figure 4.12(b). The majority of those species tree candidates require more than 200 extra lineages. The first seven best trees have 127, 134, 163, 170, 186, 191 and 193, respectively. The best tree (the one with 127 extra lineages) is the one shown in Figure 4.12(a), while the other six are shown in Figure 4.13. A very important point to make here is that these seven trees, while produced by our

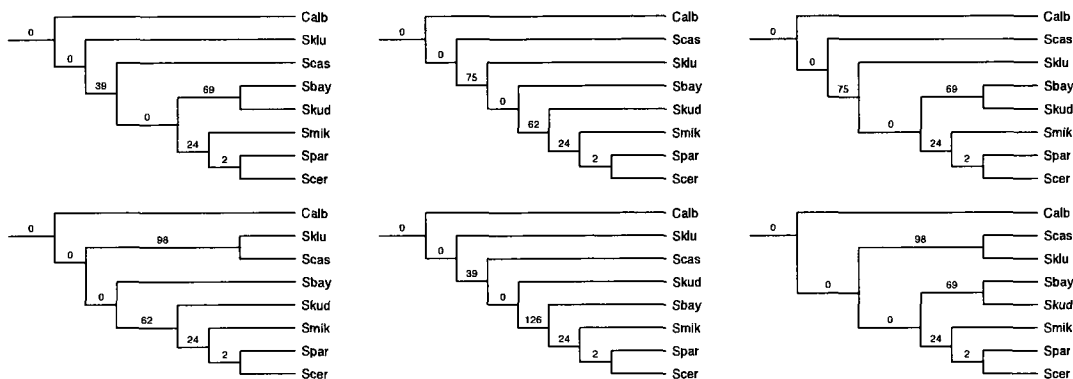


Figure 4.13 : The six best sub-optimal trees for the yeast data set. These trees, from left to right and top down, have in total 134, 163, 170, 186, 191 and 193 extra lineages. The values on the branches are the numbers of extra lineages within them.

non-parametric method, include all six maximum posterior probability trees found by BEST in [32].

### 4.7.3 Analysis of the Synthetic Data

The simulated data allowed us to investigate other aspects of the performance of our method, since the true species tree is known and we could compare the inferences made by our method against the true trees. In the first analysis, we use all  $2^8 - 1 = 255$  clusters (since there are eight species) to compute the optimal trees. Figure 4.14 shows the normalized RF distance between the inferred species tree and the true one. Clearly, for a given number of loci and alleles, the performance of MDC is better for the case of deep divergence (total branch length of  $10N_e$  than the case of recent divergence (total branch length of  $1N_e$ ). However, the difference in performance shrinks as the number of individuals sampled increases. For example, when only a single individual is sampled per species and a single locus is used, MDC has an error rate of about 19% in the case of deep divergence, whereas it has an error rate of



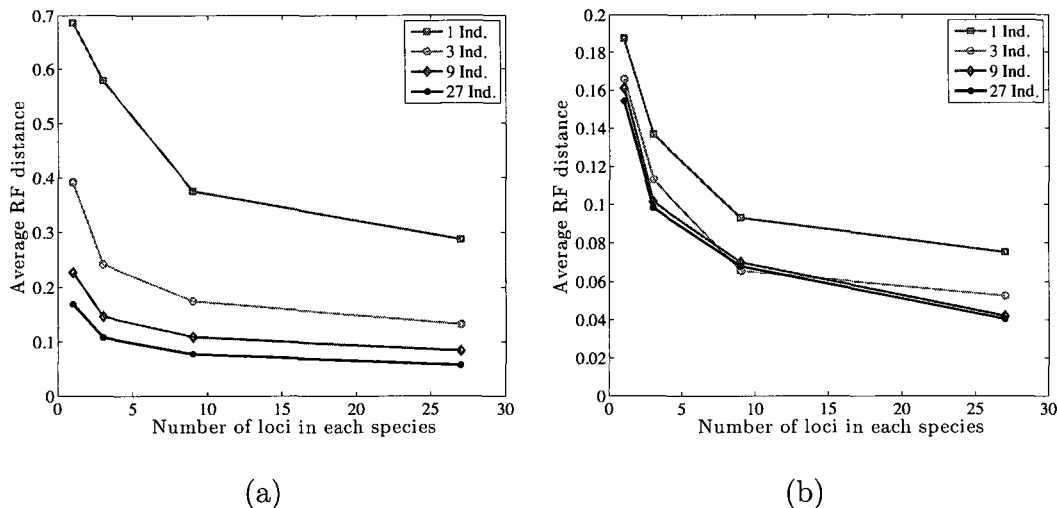


Figure 4.14 : Accuracy of the inferred species tree as measured by the Robinson-Foulds distance when all clusters (there are  $2^8 - 1 = 255$  of them) are used. (a)Recent divergence (total branch length is  $1N_e$ ); (b) Deep divergence (total branch length is  $10N_e$ ). We note that the  $y$ -axes in (a) and (b) are on different scales to make the difference between the curves more visible.

about 70% in the case of recent divergence. However, this gap closes as the number of individuals and number of loci increase.

In general, we observe the MDC's performance improves as the number of loci and individuals increases, regardless of the level of divergence. However, in the case of recent divergence, we observe that increasing the number of individuals yields a higher gain in performance than an increase in the number of loci (see also [34]). Further, under this divergence, the gain from increasing the number of loci becomes much smaller as the number of individuals sampled is larger. For example, for the case of 27 individuals, there is hardly any gain from increasing the number of loci from nine to 27.

It is important to note that when a single gene tree is used as the input to MDC, the method returns a species tree that is identical to the gene tree, since that is the

tree with the minimum (zero, in this case) number of extra lineages. We observe that the performance, in the case of a single locus and single individual, is much better in the deep divergence case—this is simply because the gene tree in this case has a smaller degree of incongruence with the species tree. However, even in the case of recent divergence, using only one locus but with increasing the number of alleles from one to 27, results in a drastic improvement in performance. Last but not least, Figure 4.14 indicates statistical consistency of MDC under the simulation conditions.

The amount of incongruence in a data set may be reflected in the optimal number of extra lineages required to reconcile all the gene trees within the branches of a species tree, over all possible species tree. Figure 4.15 shows the average number of extra lineages required to reconcile all gene trees in the input within the branches of the optimal (under MDC) tree. We can see that the average number of extra lineages is much smaller in the case of deep divergence—we would expect much less incongruence in this case than in the case of recent divergence. Further, we observe that for small numbers of individuals, the increase in the number of extra lineages is much slower than for the case of large numbers of individuals. This indicates that a large extent of the incongruence is caused by the multiplicity of individuals, rather than from the size of the set  $\mathcal{G}$  of gene trees. This has a practical implication on the running time of inference methods: when analyzing genome-scale data sets, the number of loci, particularly for small numbers of individuals, may not be the crucial factor affecting the performance (in terms of time and memory requirements) of the inference method.

In the second analysis, we used only clusters induced by gene trees to infer the species tree. Given that under the coalescent model, the gene tree is a random variable conditional on the species tree, gene trees are expected to contain the signal for the

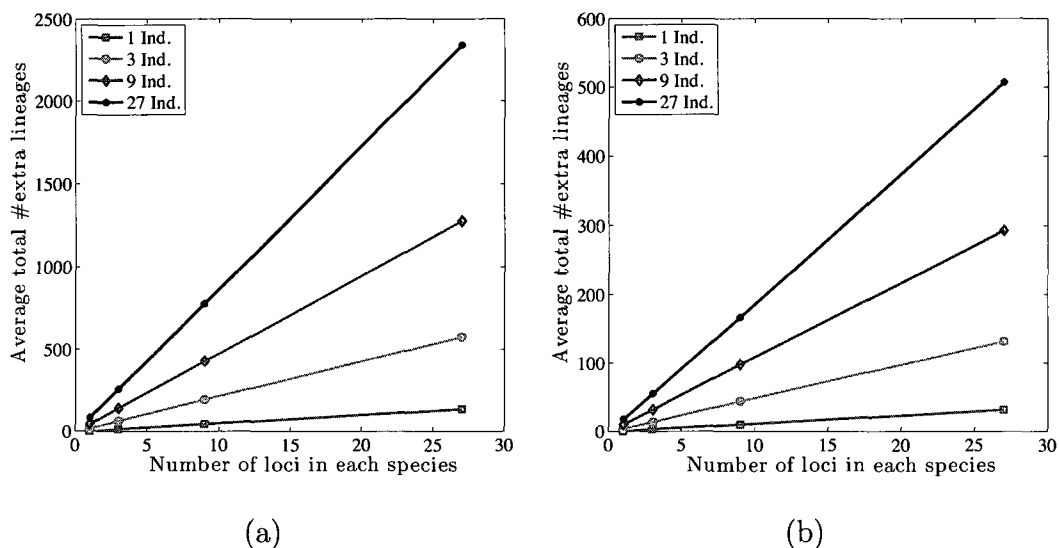


Figure 4.15 : Average numbers of extra lineages required to reconcile the inferred species tree and gene trees when all clusters (there are  $2^8 - 1 = 255$  of them) are used for the inference. (a) Recent divergence (total branch length is  $1N_e$ ); (b) Deep divergence (total branch length is  $10N_e$ ). We note that the  $y$ -axes in (a) and (b) are on different scales to make the difference between the curves more visible.

phylogenetic relationship of the species. Figure 4.16 plots the average rate of species tree clusters that would be missing from clusters induced by gene trees. Clearly, the number of missing species tree clusters decreases as the numbers of loci and of individuals increase, and no clusters are missing when nine loci are used and at least three individuals are sampled. When a single individual is sampled, using all 27 loci guarantees that almost all clusters of the species tree would be included in gene tree clusters.

Figure 4.17 shows the RF distance between the true species tree and the tree inferred from only clusters induced by gene trees. When comparing the results in this figure with those in Figure 4.14, we observe that there is almost no loss in accuracy of our algorithms and the MDC criterion. Further, the average number of extra lineages

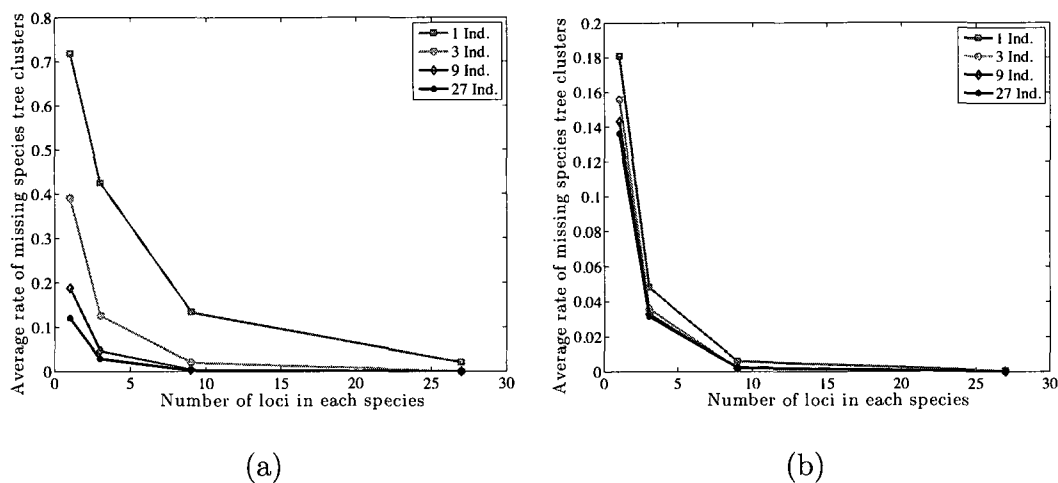


Figure 4.16 : Average rates of species tree clusters that do not appear in any gene trees. (a) for data with recent divergence (total branch length is  $1N_e$ ); (b) for data with deep divergence (total branch length is  $10N_e$ ). We note that the  $y$ -axes in (a) and (b) are on different scales to make the difference between the curves more visible.

for the inferred tree in this case is almost the same as the optimal value, as evident from comparing Figures 4.15 and 4.18.

We note that the settings for generating our generating synthetic data are reasonable for many organisms [34], and that the two analyses above show the high accuracy of our algorithms in inferring the species tree, whether only gene tree clusters or all (nonempty) clusters are used. We also note further that their accuracy in inferring the species tree for the *Apicomplexan* and yeast data sets was not affected if only clusters induced by gene trees are used. We can therefore say that it is sufficient to infer the species tree by considering only gene tree clusters, which is often much smaller than  $2^n - 1$ , the total number of nonempty clusters (Figure 4.19). Thus, this observation has a significant impact on the actual running time of our methods. More importantly, it can help to boost other methods such as BEST [33] as they can narrow the search space to candidate trees built from clusters induced by gene trees.

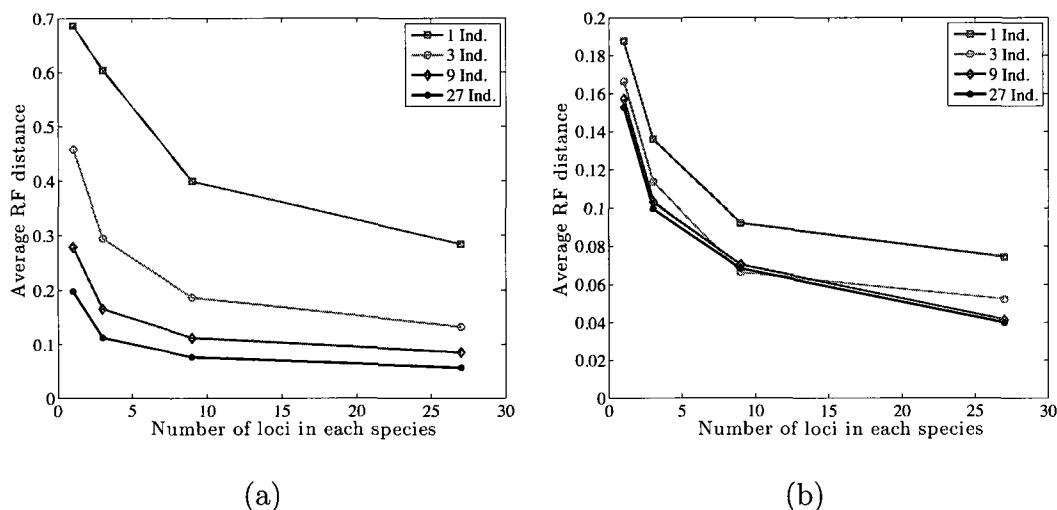


Figure 4.17 : Accuracy of the inferred species tree as measured by the Robinson-Foulds distance when only clusters induced by gene trees are used. (a) for data with recent divergence (total branch length is  $1N_e$ ); (b) for data with deep divergence (total branch length is  $10N_e$ ). We note that the  $y$ -axes in (a) and (b) are on different scales to make the difference between the curves more visible.

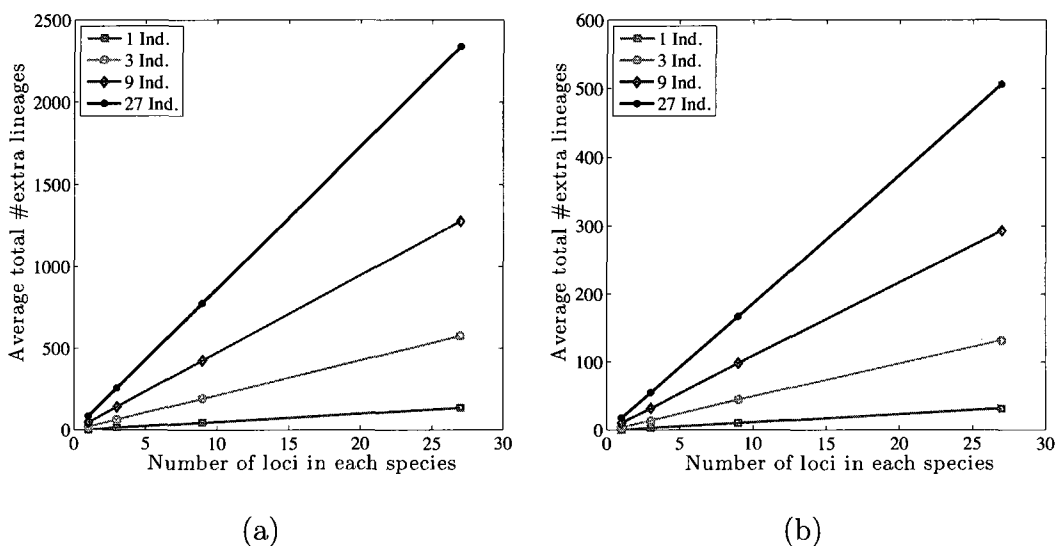


Figure 4.18 : Average numbers of extra lineages required to reconcile the inferred species tree and gene trees when only clusters induced by genes trees are used for the inference. (a) for data with recent divergence (total branch length is  $1N_e$ ); (b) for data with deep divergence (total branch length is  $10N_e$ ). We note that the  $y$ -axes in (a) and (b) are on different scales to make the difference between the curves more visible.

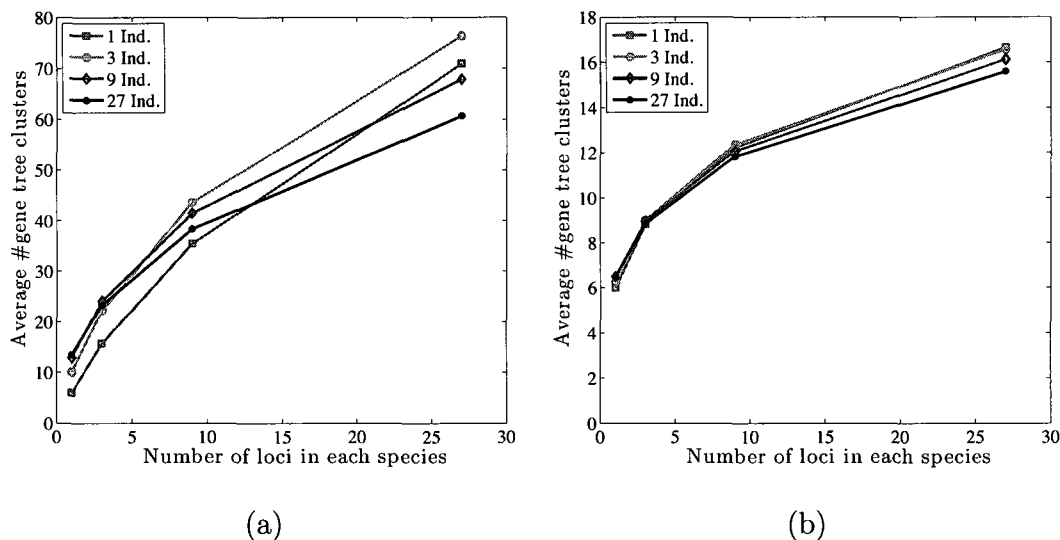


Figure 4.19 : Average numbers of clusters induced by gene trees, excluding single-element and all-element clusters. Note that the total number of nonempty clusters is  $2^8 - 1 = 255$ , as there are eight species. (a) Recent divergence (total branch length is  $1N_e$ ); (b) Deep divergence (total branch length is  $10N_e$ ). We note that the  $y$ -axes in (a) and (b) are on different scales to make the difference between the curves more visible.

## 4.8 Discussions

In this chapter, we show that the MDC cost (i.e., the number extra lineages) for a branch in a species tree depends only on the cluster it induces (and gene trees), and how to compute it. Based on this, we present an ILP and DP algorithms for inferring species trees from multiple gene trees under the MDC criterion. The experimental results we present in this chapter demonstrate that our algorithms compute very accurate species trees. They also show that we can use just clusters induced by gene trees for inferring the species tree, an important observation that has a significant impact on the actual running of our algorithms. However, we must also note that the MDC criterion may still not identify the true species tree; for example, in Figures 4.14 and 4.17, the RF distance does not drop to zero, even in the case of 27 loci and 27

individuals. This indicates the number of extra lineages required by the true species tree is larger than the optimal one—a phenomenon encountered by all parsimony-based criteria.

Finally, although the empirical results show that there is almost no difference between the optimal tree and the tree inferred by using only gene tree clusters (Figures 4.14, 4.17), we can come up with a counterexample where those two trees are in fact different. Consider the gene trees in Figure 4.20(a). Figure 4.20(b) is the compatibility graph constructed from clusters induced by those trees, where each vertex represents a cluster and is assigned the number of extra lineages for that cluster. In this graph, the maximal clique with the smallest total weight is highlighted: it consists of three vertices representing clusters  $\{a, b\}$ ,  $\{a, b, c\}$  and  $\{a, b, c, d\}$ , and it has weight seven. Therefore, the optimal tree inferred from this graph requires *seven* extra lineages to reconcile the gene trees. However, Figure 4.20(c) shows a tree that requires only *six* extra lineages to reconcile them.

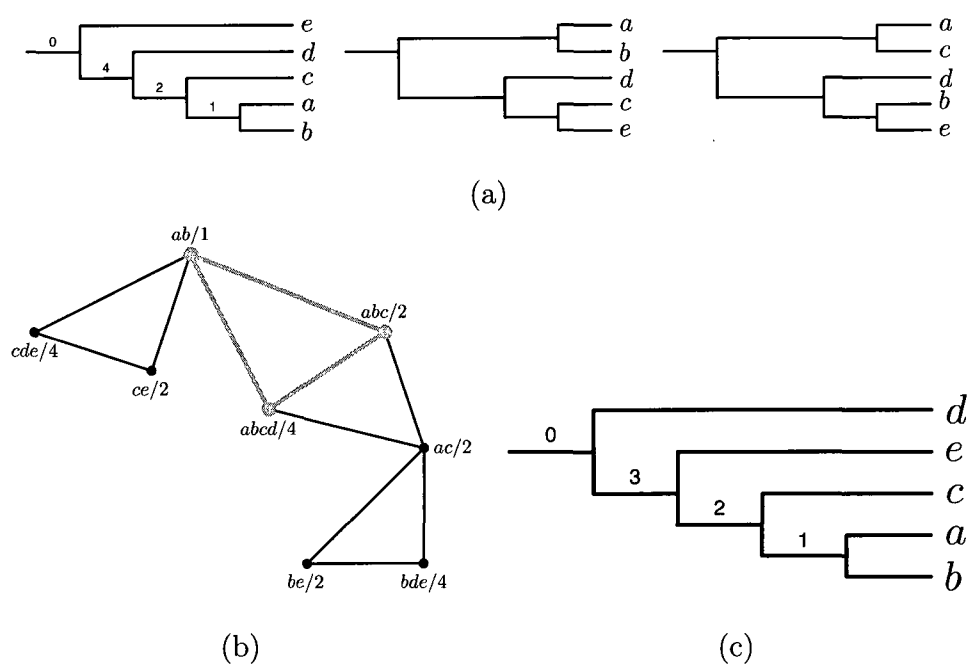


Figure 4.20 : A counterexample where the optimal tree cannot be built from gene tree clusters. (a) is the input gene trees. (b) is compatibility graph built from clusters induced by gene trees in (a), where the maximal clique with the smallest weight is highlighted. (c) is another tree that requires a fewer number of extra lineages to reconcile three trees in (a).



## Chapter 5

### Detection of Hybridization despite Lineage Sorting

The main focus of this dissertation is to infer the evolutionary relationships of species from gene trees despite their incongruence. In Chapters 3 and 4, we present methods for reconstructing the species tree when we assume that the incongruence is due only to lineage sorting. Another biological process that causes species/gene tree incongruence is hybridization—the “crossing” of genetic material from one species to another. Hybridization is believed to play an important role in the speciation and evolutionary innovations of several groups of plant and animal species [94, 59]. Whether hybridization is polyploid or diploid, the evolutionary histories of different marker alleles in a hybrid species take different paths through the two parents. This evolutionary fact is the basis for a large class of phylogeny-based methods for detecting hybridization (or, reticulate evolution in general) in a group of taxa. These methods compare the evolutionary histories of different genomic regions, and take incongruence in their individual evolutionary histories to indicate hybridization, for example, those described in Chapter 2; see also [95] for a recent survey.

A major factor that confounds the performance, in terms of the accuracy of the inferred species evolutionary history, of hybridization detection methods is that species/gene tree incongruence may be caused by other factors, such as lineage sorting (also referred to as deep coalescence) [5]. Indeed, several recent studies have reported on massive amounts of incongruence in various data sets due to lineage sorting; e.g., see [69, 25, 29, 96, 97]. Therefore, incongruence among evolutionary histories of ge-

nomous regions may be partly due to lineage sorting, partly due to hybridization, and distinguishing between the two factors is hard [58].

Existing methods for phylogenetic network reconstruction choose to ignore lineage sorting as a cause of incongruence, while those for inferring species trees despite lineage sorting ignore hybridization. When either assumption holds, it has been shown that they compute accurate estimates of species phylogenies [45, 48, 34, 88]. However, under most circumstances, such assumptions cannot be made *a priori*, and hence applying hybridization detection methods results in an overestimation of the amount of hybridization, while species tree inference methods (despite lineage sorting) fail to detect hybridization events that might have occurred. Therefore, a more appropriate model is a phylogenetic network that allows for deep coalescence events, since such a framework allows for simultaneously capturing vertical and horizontal inheritance of genetic material [98].

In this chapter, we present a heuristic for detecting hybridization despite incomplete lineage sorting [99]. Our heuristic is parsimony-based, and extends the MDC criterion for inferring the species tree that we discuss in Chapter 4. It infers phylogenetic networks to explicitly model hybridization, while simultaneously accommodating lineage sorting. Since trees are a special case of networks, our method infers a tree when there is no support for hybridization in the data.

We have studied the performance of our method on simulated data, and found that the discrete quantity of extra lineages captures to a certain degree the amount of hybridization, and that the divergence time of the two species involved in hybridization, as well as the time between hybridization and a consecutive divergence, affect the detectability of hybridization, particularly the latter. In addition, we reanalyzed the 106-locus yeast data set in [25]. In this new analysis, we show that a phylogenetic

network with a single hybridization fits the data much better than the tree reported in Section 4.7 does [99]. We therefore propose a hypothesis of the occurrence of a hybridization event involving *S. kudriavzevii*, *S. bayanus* and the clade of *S. mikatae*, *S. cerevisiae*, and *S. paradoxus*.

As evidence of hybridization in plants and animals continues to accumulate, and its role in speciation and evolutionary innovations continue to be elucidated, our framework will help to analyze systematically the evolution of groups of species in which hybridization may have occurred.

## 5.1 Current Methods for Simultaneous Modeling of Lineage Sorting and Reticulation

In this section, we describe four current methods that attempt to detect hybridization or horizontal gene transfer despite lineage sorting.

### 5.1.1 The Method of Than *et al.*

Than *et al.* introduced a stochastic framework for detecting horizontal gene transfer, given a species tree and a gene tree, despite lineage sorting [54]. This framework is based on the *coalescent* model, and assumes knowledge of the population parameters (branch lengths, population size, etc.). Consider a the model in Figure 5.1 for three bacteria  $a$ ,  $b$  and  $c$ , where an HGT event occurs at time  $\tau_h$ . If there is no lineage sorting (Figure 5.1(a)), then this event results in the gene tree  $(a, (b, c))$  that is different from the species tree  $((a, b), c)$ . However, lineage sorting can cancel the effect of an HGT event, as illustrated in Figure 5.1(b). We note that in the figure,  $\tau_h \leq t_1$ , the speciation time of  $a$  and  $b$ , but it can happen that  $\tau_h > t_1$ , in which case we interpret it as having an HGT event occurring before the speciation time of  $\tau_1$  of  $a$  and  $b$ .

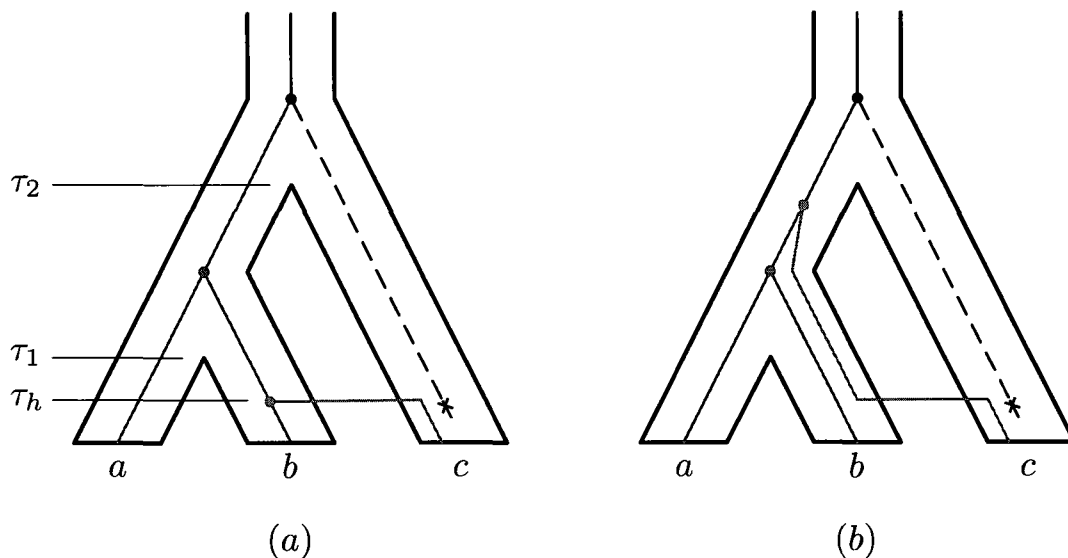


Figure 5.1 : A three bacterial species model with an HGT event. In (a), there is no lineage sorting, and hence resulting a gene tree that is different from the species tree. In (b), there is a deep coalescence event between the lineage in  $b$  and the lineage transferred to  $c$ , making the gene tree congruent (topologically) with species tree.

By the same analysis as in Subsection 2.3.1, we obtain the following probabilities of obtaining three different gene trees, given the species tree  $((a, b) : \tau_1, c) : \tau_2$ .

$$\Pr[((a, b), c)] = \begin{cases} \frac{1}{3}e^{-(\tau_1 - \tau_h)/N_e} & \text{if } \tau_h \leq \tau_1, \\ 1 - \frac{2}{3}e^{-(\tau_h - \tau_1)/N_e} & \text{if } \tau_h > \tau_1 \end{cases}, \quad (5.1)$$

$$\Pr[((a, c), b)] = \begin{cases} \frac{1}{3}e^{-(\tau_1 - \tau_h)/N_e} & \text{if } \tau_h \leq \tau_1, \\ \frac{1}{3}e^{-(\tau_h - \tau_1)/N_e} & \text{if } \tau_h > \tau_1 \end{cases}, \quad (5.2)$$

$$\Pr[(a, (b, c))] = \begin{cases} 1 - \frac{2}{3}e^{-(\tau_1 - \tau_h)/N_e} & \text{if } \tau_h \leq \tau_1, \\ \frac{1}{3}e^{-(\tau_h - \tau_1)/N_e} & \text{if } \tau_h > \tau_1. \end{cases} \quad (5.3)$$

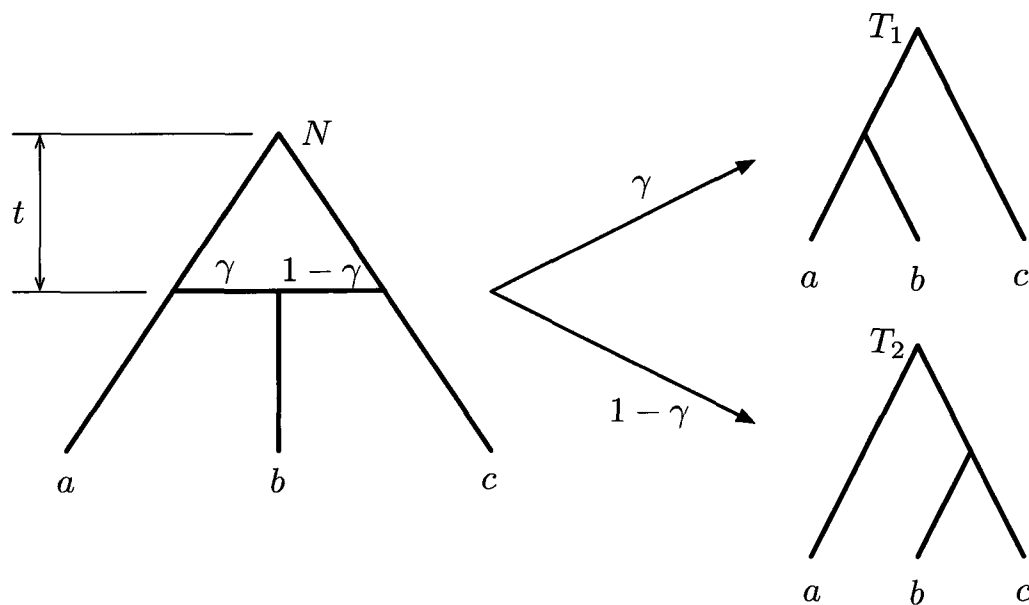


Figure 5.2 : smallThe hybrid speciation model. The network is shown on the left, and the two induced trees are shown in the right. In this network, a a gene in  $b$  is either from  $a$  with probability  $\gamma$ , or from  $c$  with probability  $1 - \gamma$ .

### 5.1.2 The Method of Meng *et al.*

Meng and Kubatko introduced another coalescent-based framework for detecting hybridization despite lineage sorting, also assuming knowledge of the population parameters [100]. Consider the model in Figure 5.2 on three species, where  $b$  is a hybrid species of  $a$  and  $c$ . In this model, a gene in  $b$  is either from  $a$  with probability  $\gamma$ , or from  $c$  with probability  $1 - \gamma$ . For a given gene, we choose either tree  $T_1$  (with probability  $\gamma$ ) or tree  $T_2$  (with probability  $(1 - \gamma)$ ), and then apply the coalescent process on the chosen tree. We can compute the likelihood of a species tree  $T'$  given a collection  $\mathcal{G}$  of gene trees by

$$\Pr(\gamma, T' | \mathcal{G}) = \prod_{T \in \mathcal{G}} (\gamma \Pr(T | T_1) + (1 - \gamma) \Pr(T | T_2)), \quad (5.4)$$

where  $\Pr(T | T_1)$  and  $\Pr(T | T_2)$  are computed as in Subsection 2.3.1. The species tree is the tree, along with  $\gamma$ 's value, that maximizes this quantity. The value of  $\gamma$  is also used to determine whether there is a hybridization or not in the data.

We note that Than *et al.*'s and Meng *et al.*'s methods require knowledge of the reticulation scenario, and the frameworks can be used to assess its support in terms of the observed gene trees. In other words, they do not attempt to detect the location of the reticulation events.

### 5.1.3 The Method of Joly *et al.*

More recently, Joly *et al.* introduced a statistical framework for the same task, which distinguishes hybridization from lineage sorting based on the genetic distance between sequences [101]. Assuming a null hypothesis where we assume that the incongruence between species and gene trees is due solely to lineage sorting, we can obtain a distribution of this distance using the coalescent theory. Then, we compute the distance between every pair of sequences in the data, and compare it with the distance derived from the null hypothesis: if the observed distance is smaller than  $(1 - \alpha)$  of the null hypothesis distance for some threshold  $\alpha$  (for example, 0.5%), the null hypothesis is rejected and we conclude there is hybridization.

This framework, as well, requires knowledge of the population parameters, since it conducts coalescent-based simulations for testing the null hypothesis of only lineage sorting and no hybridization.

## 5.2 Lineage Sorting in Phylogenetic Networks

In this section, we describe how to extend the MDC criterion, originally defined for phylogenetic trees, to phylogenetic networks, which yields a framework for detecting

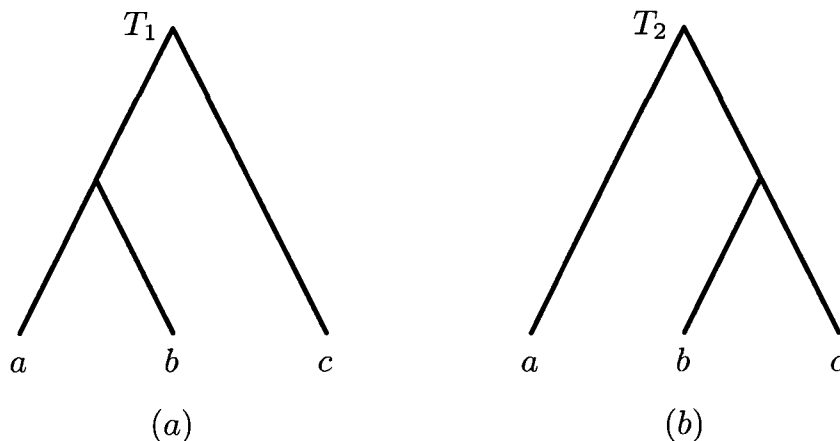


Figure 5.3 : Two gene trees that differ in the placement of  $b$ .

hybridizations as well as other reticulate events despite lineage sorting in the data. Let  $N$  be a phylogenetic network, and  $T$  a gene tree. Denote  $\mathcal{T}(N)$  the set of trees induced by  $N$  as defined in Chapter 2. We define the number of extra lineages required to reconcile  $T$  within the branches of  $N$  to be

$$\alpha(N, T) = \min\{\alpha(T', T) : T' \in \mathcal{T}(N)\}, \quad (5.5)$$

where  $\alpha(T', T)$  is the number of extra lineages for reconciling  $T$  within the branches of  $T'$ . This definition is generalized in a straightforward manner to a set  $\mathcal{G}$  of gene trees:

$$\alpha(N, \mathcal{G}) = \sum_{T \in \mathcal{G}} \alpha(N, T). \quad (5.6)$$

Consider two trees in Figure 5.3. They are different only in the placement of  $b$ . In Figure 5.4(a), we show an optimal tree for reconciling them under the MDC criterion that requires a single extra lineage. On the other hand, if we allow for a reticulate event between species  $a$  and  $c$ , resulting in species  $b$ , (Figure 5.4(b)) then this network induces both gene trees, and hence the number of extra lineages for it

is zero, according to Equation (5.6).

An analogous definition to the problem of inferring the phylogenetic tree under the MDC criterion in the context of networks would be the following:

**Definition 5.1** (Network Inference Using the MDC Criterion).

**Input:** *A set of gene trees  $\mathcal{G}$ .*

**Output:** *A network  $N$  such that the total number of extra lineages required to reconcile all gene trees of  $\mathcal{G}$  within  $N$  is minimized.*

This definition suffers from a major drawback: Without controlling the number of reticulations in the network, identifying an optimal network  $N$  becomes a trivial task, since we can always find a network that reconciles the entire set of gene trees with no extra lineages, e.g., [95]. This overfitting phenomenon is an issue that plagues phylogenetic network reconstruction in general: The more reticulations in the network, the better it fits the data. Therefore, without a close inspection of the improvement to fitting the data, one may end up with a network that grossly overestimates the amount of hybridization [95]. Even worse, one can always find a network that reconciles each gene tree without resulting in any deep coalescence events. This is analogous to the problem of inferring phylogenetic networks to model sequence evolution, where one can always find a network under which each site in the sequences evolves with no homoplasy [102].

More precisely, assume  $\alpha(N, \mathcal{G}) = k$  for a set of  $\mathcal{G}$  of gene trees reconciled within the branches of a phylogenetic network  $N$ . If  $N'$  is a phylogenetic network obtained by adding extra hybridization events to  $N$ , then we always have  $\mathcal{T}(N) \subseteq \mathcal{T}(N')$ . Consequently, and based on Equation (5.6), this implies that  $\alpha(N', \mathcal{G}) \leq \alpha(N, \mathcal{G}) = k$ . In other words, adding more hybridization events to a network can never hurt



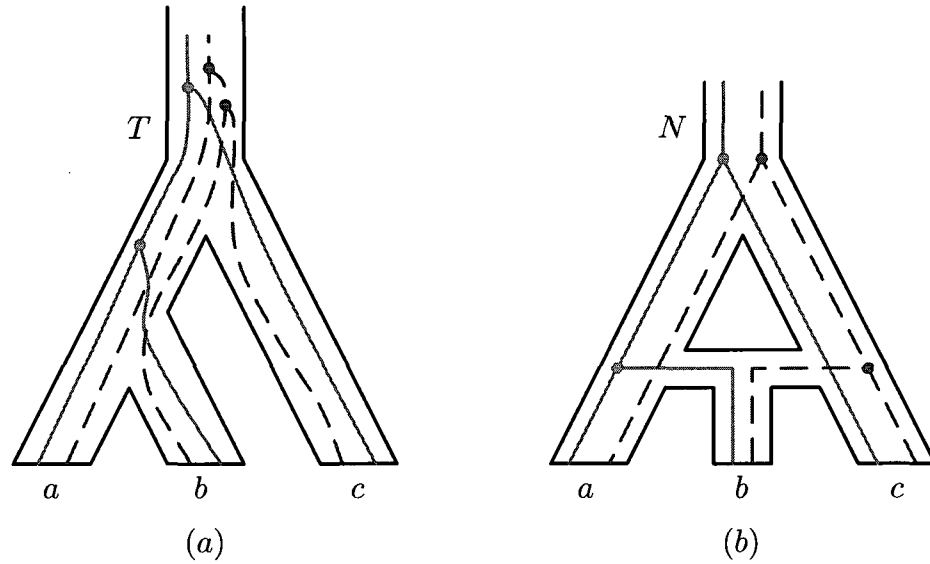


Figure 5.4 : An optimal tree and an optimal network for the two gene trees in Figure 5.3. (a) An optimal species tree under the MDC criterion, which requires a single deep coalescence event to reconcile the two gene trees of Figure 5.3. (b) A phylogenetic network that requires *no* deep coalescence events to reconcile both gene trees of Figure 5.3.

fitting the gene trees in  $\mathcal{G}$  to the network under the MDC criterion as given by Equation (5.6); it either improves it or keeps it the same. As a result, if optimizing the MDC criterion is the objective, it is always “safe” to keep adding hybridization events. Clearly, this is problematic, and it would be problematic for a probabilistic approach as well, unless a criterion that penalizes the model’s complexity (in this case, this complexity includes the number of reticulations in the network) is used.

The “quick fix” of minimizing  $\alpha(N, \mathcal{G}) + h_N$ , where  $h_N$  is the number of hybridizations in  $N$  does not work well in general, since as the number of loci increases, an improvement in the number of deep coalescence events may be obtained by adding an arbitrary hybridization event. We propose a solution based on an empirical observation that we have made by analyzing simulated data. The observation is that

when hybridization occurs, there is no clear optimal species tree estimate; rather, some sub-optimal trees are very close to the optimal one. Based on this observation, we propose the following method for detecting hybridization despite lineage sorting:

1. Find the optimal tree  $T^*$  under MDC criterion;
2. Compute the set

$$S = \{T: (\alpha(T, \mathcal{G}) - \alpha(T^*, \mathcal{G}))/\alpha(T^*, \mathcal{G}) \leq p\%\} \quad (5.7)$$

of trees whose MDC cost is within  $p\%$  of the optimal score  $\alpha(T^*, \mathcal{G})$ ; and,

3. Infer a phylogenetic network  $N$  that reconciles the trees in  $S$ .

When there is no hybridization in the data, we would expect the set  $S$  to contain a single tree. When there is a single hybridization in the data, we would expect the set  $S$  to contain two trees. An important question concerns  $p$ : what value should be used? We show that an answer to this question depends on the proportion of loci involved in hybridization, out of all loci in the data set.

## 5.3 Experimental Study

### 5.3.1 Data

To study the performance of our method, we conducted simulation studies, and re-analyzed the yeast data set of Rokas *et al.* [25]. For the simulated data, we used the scenario depicted in Figure 5.5. In this scenario, there is a hybridization event involving  $a$ ,  $d$ , and the clade  $(b, c)$ . The value  $\gamma$  denotes the proportion of loci in  $(b, c)$  that are inherited from  $a$ , and  $1 - \gamma$  denotes the proportion of loci inherited from  $d$ . The scenario we investigate is more complex than that investigated in [54, 100, 101]

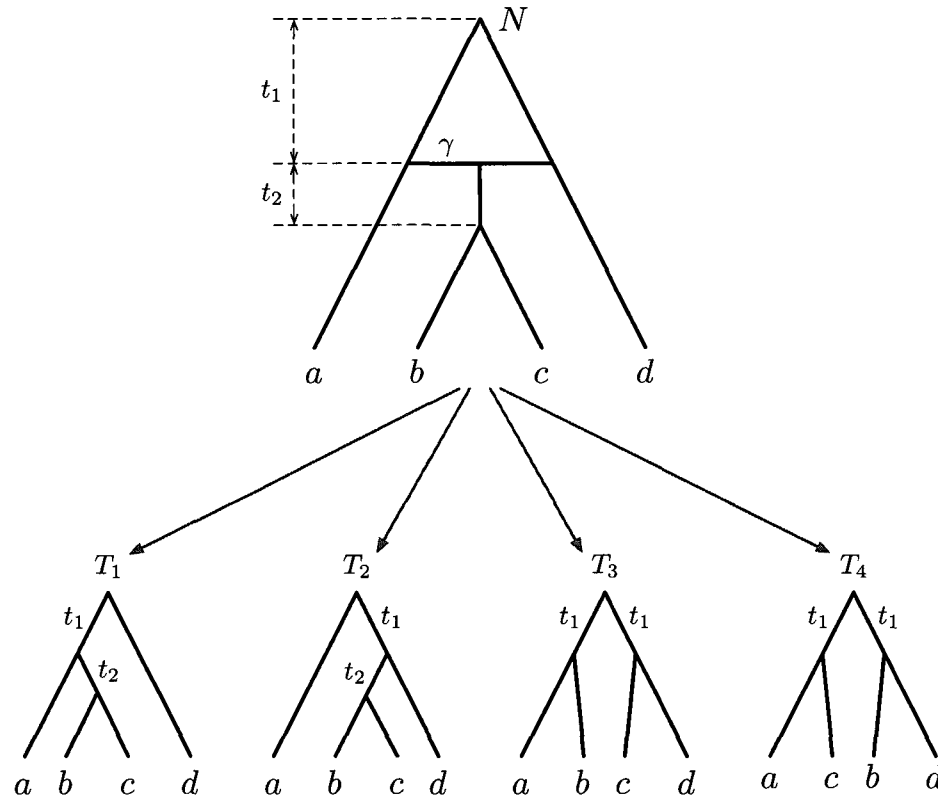


Figure 5.5 : Simulation scenario. A hundred gene trees were simulated under the coalescent model within the branches of the network  $N$  by evolving  $(1 - \frac{1}{2}e^{-t_2})\gamma$  of them within the branches of  $T_1$ ,  $(1 - \frac{1}{2}e^{-t_2})(1 - \gamma)$  within the branches of  $T_2$ ,  $\frac{1}{4}e^{-t_2}$  within the branches of  $T_3$  and  $\frac{1}{4}e^{-t_2}$  within the branches of  $T_4$ . Times are given in coalescent units (number of generations divided by population size).

in that we allow for divergence after hybridization; this allows us to study the effect of the divergence time between the two “parents” ( $t_1$  in Figure 5.5) as well as the time between the hybridization and subsequent divergence ( $t_2$  in Figure 5.5) on the ability to detect hybridization.

The simulation flow proceeds as follows for generating  $\ell$  gene trees (or, loci, in general). The probability of two alleles, one from  $b$  and another from  $c$ , not coalescing within time  $t_2$  is  $e^{-t_2}$ . Further, if such two alleles have equal probability of one coming

from  $a$  and the other coming from  $d$ , this implies that we can simulate  $\frac{1}{4}\ell e^{-t_2}$  gene trees within the branches of each of the two trees  $T_3$  and  $T_4$  in Figure 5.5. The remaining  $\ell(1 - \frac{1}{2}e^{-t_2})$  gene trees can be simulated with proportions  $\gamma$  and  $1 - \gamma$  within the branches of the trees  $T_1$  and  $T_2$ , respectively. In our simulations, we used  $\ell = 100$ , and varied the times  $t_1$  and  $t_2$  to take on the values 0.5, 1, 2, and 4, ranging from the very short (and hence extensive deep coalescence) to the very long (and hence almost no deep coalescence), respectively. Further, we used values 0, 0.1, 0.2, 0.3, 0.4, and 0.5 for  $\gamma$ , to simulate cases with amount of hybridization ranging from none to equal contribution of both parents, respectively. For each combination of values of  $\ell$ ,  $t_1$ ,  $t_2$ , and  $\gamma$ , we generated 100 data sets, and averaged the results.

The yeast data set of [25] contains seven *Saccharomyces* species *S. cerevisiae* (Scer), *S. paradoxus* (Spar), *S. mikatae* (Smik), *S. kudriavzevii* (Skud), *S. bayanus* (Sbay), *S. castellii* (Scas), *S. kluyveri* (Sklu), and the outgroup fungus *Candida albicans* (Calb). Rokas *et al.* in [25] identified 106 genes, which are distributed throughout the *S. cerevisiae* genome on all 16 chromosomes and comprise about 2% of the predicted genes. For each gene, they reconstructed its tree using the maximum likelihood and maximum parsimony methods. Among the 106 trees, more than 20 different gene-tree topologies were observed. Rokas *et al.* inferred the species tree using the concatenation method on the the sequences of the 106 genes. The resulting tree had 100% bootstrap support for each of its branches; this tree topology is shown in Figure 5.6(a). Further, various studies of the same data set, using different criteria and methods, have inferred this same tree as the species tree best supported by the 106 gene trees; e.g., [32, 88].

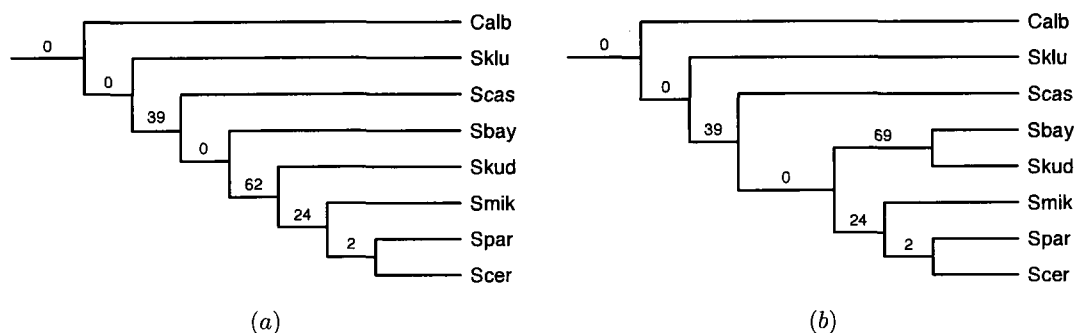


Figure 5.6 : (a) The single optimal tree under the MDC criterion for the data set. The number of extra lineages resulting from reconciling all 106 gene trees within the branches of this tree is 127. (b) The best sub-optimal tree under the MDC criterion. The number of extra lineages resulting from reconciling all 106 gene trees within the branches of this tree is 134, which is just 7 extra lineages away from the optimal value of 127 achieved by the tree in (a). The number on a branch indicates the number of extra lineages along that branch once all 106 gene trees are reconciled within the branches of the tree.

### 5.3.2 Results on Simulated Data

For the simulated data, we mainly investigated the effect of the times  $t_1$  and  $t_2$  on two questions. The first question is: How does the optimal tree under MDC compare to sub-optimal ones? For this question, our hypothesis was that for very low proportion of hybridization (indicated by low values of  $\gamma$ ), a clear species tree estimate would emerge, while an increasing proportion of hybridization would result in the emergence of more than a single species tree candidate, such that a network reconciling those candidates would be a more appropriate evolutionary history of the species. Given that the number of rooted trees on 4 taxa is only 15, we investigated all of them. Results of this investigation are shown in Figure 5.7.

As the figure shows for the case of  $t_1 = t_2 = 4$ , when there is no hybridization ( $\gamma = 0$ ), tree  $T_2$  reconciles all 100 gene trees with almost zero extra lineages, while the next sub-optimal trees (in this case, those are  $T_1$ ,  $T_5$ ,  $T_{14}$ , and  $T_{15}$ ) require 100

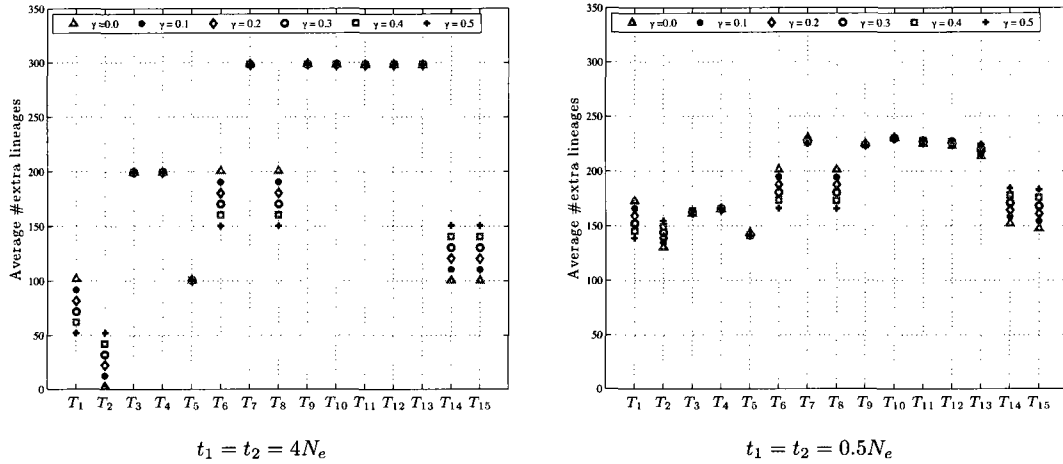


Figure 5.7 : Average numbers extra lineages for each of the constituent trees. The  $x$ -axis lists the 15 possible (rooted) tree topologies on the four taxa, and the  $y$ -axis denotes the number of extra lineages resulting from reconciling all 100 gene trees within each of the 15 trees. Left: the “easy” case of very long times; right: the “hard” case of very short times.

extra lineage, each, to reconcile all 100 gene trees. In other words, tree  $T_2$  is a clear candidate for the species tree estimate. It is worth mentioning that the trees labeled  $T_1$  and  $T_2$  in this figure are identical (in terms of topology) to  $T_1$  and  $T_2$ , respectively, in Figure 5.5. Further, notice that in this case, as  $\gamma$  increases, the gap in optimality between the two trees  $T_1$  and  $T_2$  starts decreasing, until it closes completely for the value of  $\gamma = 0.5$  (as indicated the two “level” + signs for  $T_1$  and  $T_2$ ). Therefore, in this case, if we set  $p$  in Equation (5.7) stringently to a value close to 0, we would detect hybridization perfectly in the case of  $\gamma = 0.5$  and detect no hybridization in the other cases. However, as we relax the value of  $p$ , we the method would start detecting even lower proportions of hybridization. This is very similar to the performance of the probabilistic approach of [100], where detecting the value of  $\gamma$  is the main objective of the method.

This trend becomes very blurry in the hard case of  $t_1 = t_2 = 0.5$ . In this case,

tree  $T_2$  is still the clear species phylogeny candidate when no hybridization took place. However, while the optimality gap between  $T_1$  and  $T_2$  is closing as  $\gamma$  increases, other trees, particularly  $T_5$ , blur the picture and make the detection of hybridization harder. In particular, in the case of  $\gamma = 0.5$ , trees  $T_1$  and  $T_5$  are a better pair of trees to reconcile into a network than the pair  $T_1$  and  $T_2$ .

It is important to note that we observed, under the conditions of our simulation study, that the value of  $t_2$  affects the detectability of hybridization more than the value of  $t_1$ . For example, the trends, and hence indication of hybridizations, are better in the case of  $(t_1 = 0.5, t_2 = 4)$  than in the case of  $(t_1 = 4, t_2 = 0.5)$ . This implies that the time between hybridization and a consequent divergence affects the detectability of hybridization significantly. Again, given the simplified simulation scenarios in other studies, this effect was not reported.

The second question is somehow related: how does the optimal network compare to the optimal tree? For this question, our hypothesis was that the number of extra lineages computed over the optimal network would constitute a greater improvement over that computed over the optimal tree. In particular, our hypothesis was that for the case of no hybridization ( $\gamma = 0$ ), the optimal network and optimal tree would be equally good models, while for the case of extensive hybridization ( $\gamma = 0.5$ ), the optimal network would result in a much improved reconciliation in terms of the number of extra lineages. Results of this investigation are shown in Figure 5.8.

The figure clearly shows that in the case of  $t_1 = t_2 = 4$ , for each value of  $\gamma$ , there is an optimal network with a single reticulation, that reconciles all 100 gene trees and yielding almost zero deep coalescences. While this may imply that a network is a better representation, even in the case of  $\gamma = 0$ , the number of extra lineages in the optimal tree helps address this issue. In the case of  $\gamma = 0$ , the optimal tree and

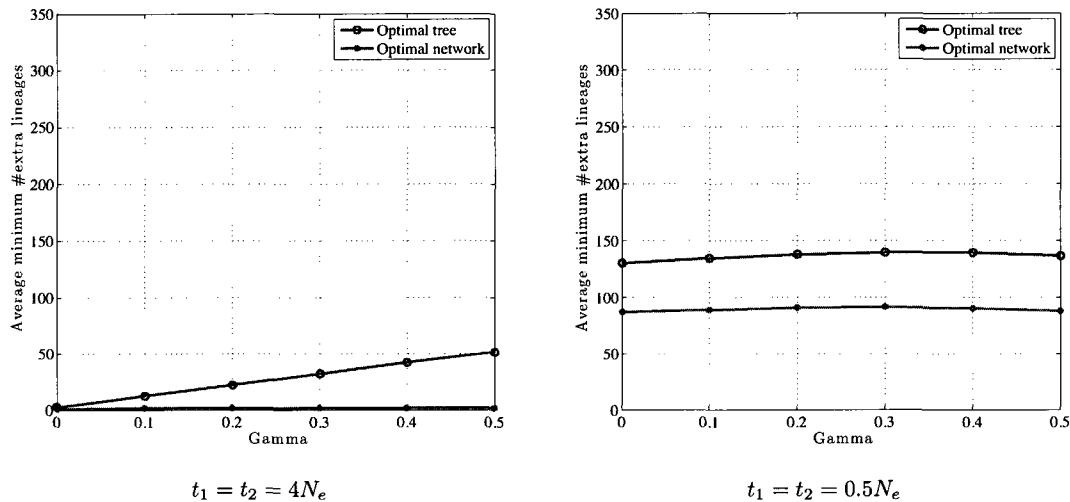


Figure 5.8 : Optimal trees vs. optimal networks. The  $x$ -axis shows the values of  $\gamma$ , and the  $y$ -axis shows the number of extra lineages. The optimal value for network is computed by exhaustively considering *all* networks on 4 taxa.

optimal network are almost equally optimal, and the tree is selected as the estimate of the species phylogeny, as it is the simpler explanation. As the value of  $\gamma$  increases, the gap in optimality between the best tree and network increases, giving more support for the hypothesis that a network is a more appropriate model and that hybridization took place. It is worth mentioning that in this case, the difference in the number of extra lineages between the optimal tree and optimal network, when normalized by the number of gene trees in the data set (100 in this case), gives an almost perfect indication of the value of  $\gamma$ —again, analogously to the probabilistic approach of [100]. As in the previous case, the performance suffers when both times  $t_1$  and  $t_2$  decrease significantly, since the amount of deep coalescences increases significantly. What the figure shows clearly is that when the extent of deep coalescences becomes massive, a network becomes a much better representation of the data, even in the absence of any hybridization. In this case, we would expect a more sophisticated approach,



such as a stochastic method that also attempts to estimate times, population sizes, etc., would do much better than a parsimony-based method such as the one we present here. It may be possible to improve the performance of the parsimony-based method by coupling it with coalescent-based simulations under the null hypothesis of no hybridization. However, once again, the performance of such an approach would heavily depend on the accurate estimate of population parameters that inaccuracies in these estimates may lead to wrong predictions.

### Results on the Yeast Data Set

For our analysis of the yeast data set, we reconstructed the gene trees using a maximum parsimony heuristic, and used our method [88] to infer the optimal species tree under the MDC criterion. There was a single optimal tree, which is identical to that proposed by Rokas *et al.* [25], and is shown in Figure 5.6(a). This tree results in 127 extra lineages when optimally reconciling all 106 gene trees in the data set.

Next, we exhaustively searched all 126 networks obtained by adding a single hybridization event to this optimal tree. There are three equally optimal networks, each resulting in 69 extra lineages when reconciling all 106 gene trees. The improvement in the number of extra lineages that is achieved by each of these three networks is  $d^* = 127 - 69 = 58$ . All other networks led to almost no improvement over the optimal species tree (i.e., all other networks required a number of extra lineages that was close to 127). If we take consider all 126 networks in calculating the  $z$ -score of the improvement of  $d^*$  in extra lineages, we have  $\mu = 4.35$  and  $\sigma = 10.94$ , which results in a  $z$ -score of 4.9. If we exclude the three optimal networks and do the calculation with the 123 sub-optimal networks, we have  $\mu = 3.04$  and  $\sigma = 7.08$ , which results in a  $z$ -score of 7.76. Either way, the  $z$ -score is very high, supporting the hypothesis pre-

sented by each of the three networks. These three networks are shown in Figure 5.9.

The network in Figure 5.9(a) illustrates a scenario in which hybridization occurred between *S. bayanus* and the clade of *S. mikatae*, *S. cerevisiae*, and *S. paradoxus* to give rise to hybrid species *S. kudriavzevii*. The network in Figure 5.9(b) illustrates a scenario in which hybridization occurred between *S. kudriavzevii* and *S. bayanus*. The network in Figure 5.9(c) seemingly illustrates a scenario in which hybridization occurred between an ancestor of all five species and the clade of *S. mikatae*, *S. cerevisiae*, and *S. paradoxus*. Such a scenario may at first sound implausible given that it violates the natural constraint that hybridization involves two species that *co-exist* in time. However, this is not necessarily the case, as this type of violation can be explained through incomplete taxon sampling or extinction [95]. This scenario can be explained, for example, by the scenario in which the hybridization occurred between the clade of *S. mikatae*, *S. cerevisiae*, and *S. paradoxus* and a sibling of the clade of all five species that was not sampled, or became extinct.

A striking point about all three networks that they all induced exactly the same pair of trees. Ignoring the hybridization event in all of them, we obtain the optimal tree shown in Figure 5.6(a). However, if we take in each of the three networks the hybridization event, we obtain the tree shown in Figure 5.6(b). Interestingly, this tree is second only to the optimal tree in terms of the number of extra lineages it requires when optimally reconciling all 106 gene trees. Further, it is very close, in terms of the optimality value, to the optimal one: only seven extra lineages separate the two.

Notice that the difference between the two trees in Figures 5.6(a) and 5.6(b) is the grouping of the three groups: (1) *S. kudriavzevii*, (2) *S. bayanus*, and (3) the clade of *S. mikatae*, *S. cerevisiae*, and *S. paradoxus*. While the optimal tree groups (2) with (3),

each of the three hybridization scenarios shown in Figure 5.9 indicates that hybridization could be a better supported hypothesis than that given by the optimal tree. Of the 106 gene trees, 65 have the clade  $((S. paradoxus, S. cerevisiae), S. mikatae)$ , and 38 have the clade  $(S. bayanus, S. kudriavzevii)$ . Further, each of the 106 loci in all five species have coalesced at the most recent common ancestor (MRCA) of these five species, as indicated by the value zero on the branch above the MRCA in all three scenarios in Figure 5.9.

It is worth mentioning that while the difference in numbers of extra lineages between the optimal tree and best sub-optimal tree is very small ( $134 - 127 = 7$ ), this difference is much larger between the optimal network and best sub-optimal network is much larger ( $92 - 69 = 23$ ). This has at least two implications. First, while previous studies proposed the tree in Figure 5.6(a) as the species tree for this group, our analysis shows that it is not really a clear candidate, given that the sub-optimal tree shown in Figure 5.6(b) reconciles all 106 gene trees almost equally well. Second, from a practical perspective, to analyze the data for hybridization scenarios, it may be worth focusing first on the set of all of trees within a certain threshold from optimality, as given by Equation (5.7).

Finally, it is not surprising to propose hybridization scenarios as evolutionary hypotheses for the data set when Rokas *et al.* and others have proposed a *tree* as the evolutionary history. Several studies have reported on the presence of hybridization in yeast; e.g., [103, 104]. In particular, Dunn and Sherlock have recently reported on a hybridization between *S. cerevisiae* and *S. bayanus*-related yeasts to form *Saccharomyces pastorianus* [105].

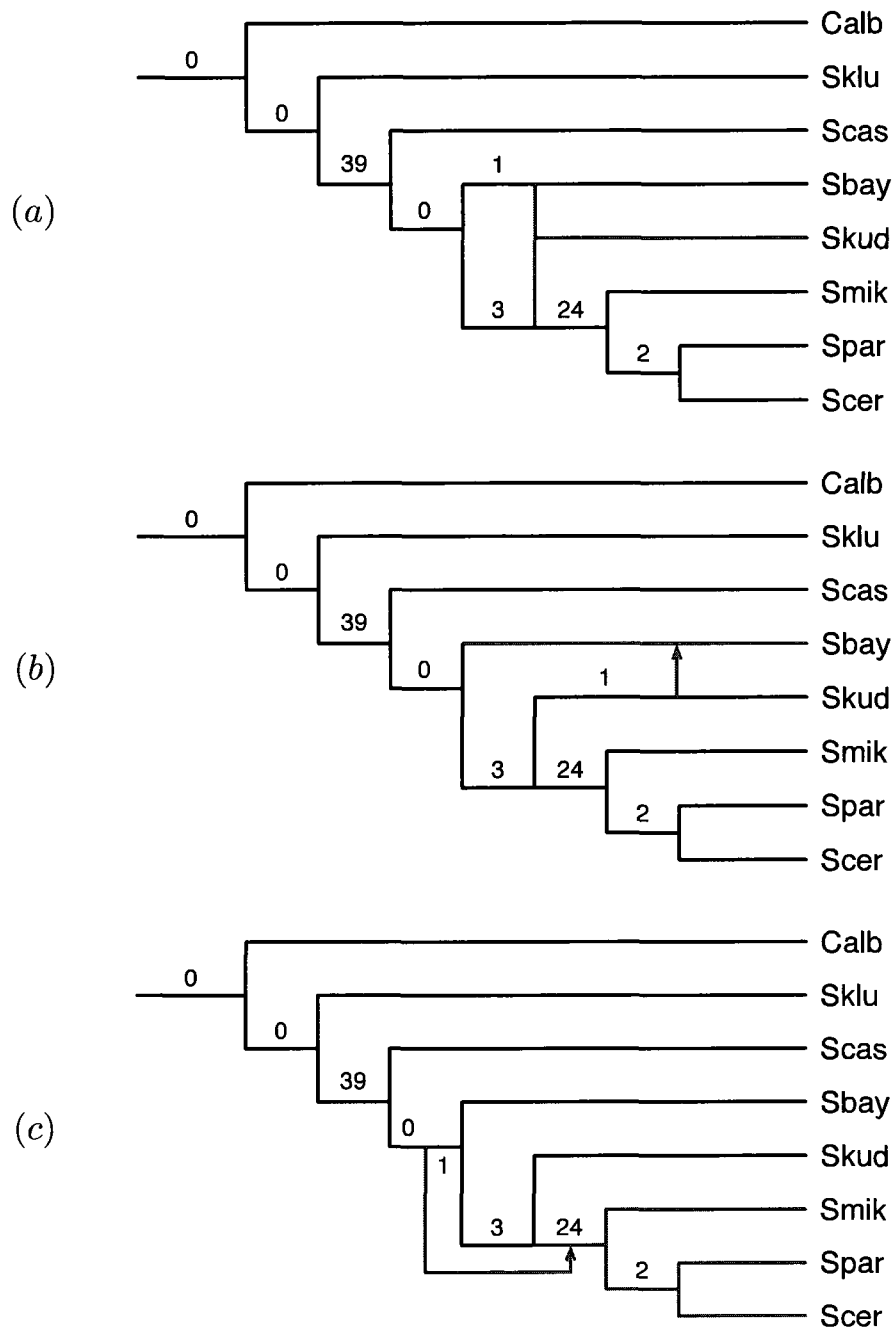


Figure 5.9 : Three hybridization scenarios for the yeast data set. Each of the networks requires 69 extra lineages to reconcile all 106 gene trees, and depicts a slightly different hybridization scenario. The number on a branch indicates the number of extra lineages along that branch once all 106 gene trees are reconciled within the branches of the network.

## Chapter 6

### PhyloNet

In the previous chapters, we present methods for inferring species phylogenies from gene trees despite lineage sorting. Those methods are implemented in software package PhyloNet [35], which is freely available at: <http://bioinfo.cs.rice.edu/phyloNet/>. In addition to those methods, PhyloNet implements methods for comparing and charactering reticulate networks, which include:

1. RIATA-HGT [47, 54]: reconciling a pair of species tree and gene tree;
2. evolutionary network representation: reading/writing evolutionary networks in a newly devised compact form;
3. evolutionary network characterization: analyzing evolutionary networks in terms of three basic building blocks—trees, clusters, and tripartitions;
4. evolutionary network comparison: comparing two evolutionary networks in terms of topological dissimilarities, as well as fitness to sequence evolution under a *maximum parsimony criterion*; and
5. evolutionary network construction: reconstructing an evolutionary network from a species tree and a set of gene trees.

Furthermore, since various evolutionary network utilities use functionalities from the phylogenetic tree domain, PhyloNet provides a set of standalone phylogenetic tree analysis tools.

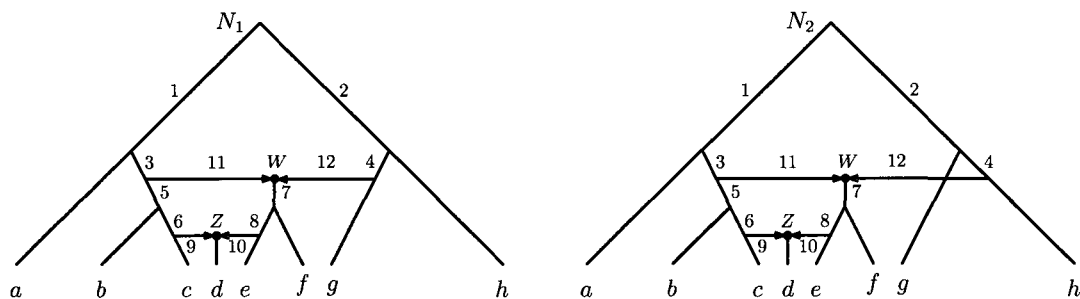


Figure 6.1 : Two evolutionary networks  $N_1$  and  $N_2$ , each with eight leaves (labeled  $a, \dots, h$ ) and two network nodes  $W$  and  $Z$ . Shown are the orientation of the network edges; all other edges are directed away from the root (toward the leaves) Notice that the difference between the two networks is that node  $W$  in  $N_1$  has lineage  $g$  as one of its parents, whereas node  $W$  in  $N_2$  has lineage  $h$  as one of its parents.

## 6.1 Phylogenetic Network Representation

The Newick format for representing and storing phylogenetic trees was adopted in 1986 [37], and it has been the standard for almost all phylogeny software packages ever since. This format captures an elegant correspondence between leaf-labeled trees and matched parentheses, where the leaves are represented by their names and the internal nodes by a matched pair of parentheses that contains a list of the Newick representation of all its children. Shown in Figure 6.2 are three trees along with their representations in the Newick format.

Existing phylogenetic network software tools store these networks as adjacency lists of their underlying graphs, which are usually very large and necessitate translation of representations among the different tools. Morin and Moret [106] proposed a modified version of the Newick format for representing reticulate networks. In their format, hybrid nodes are represented by nodes labeled with  $\#H$ , and those nodes are considered as two separate nodes in the normal Newick format for trees; see the Figure 6.3 for an example.

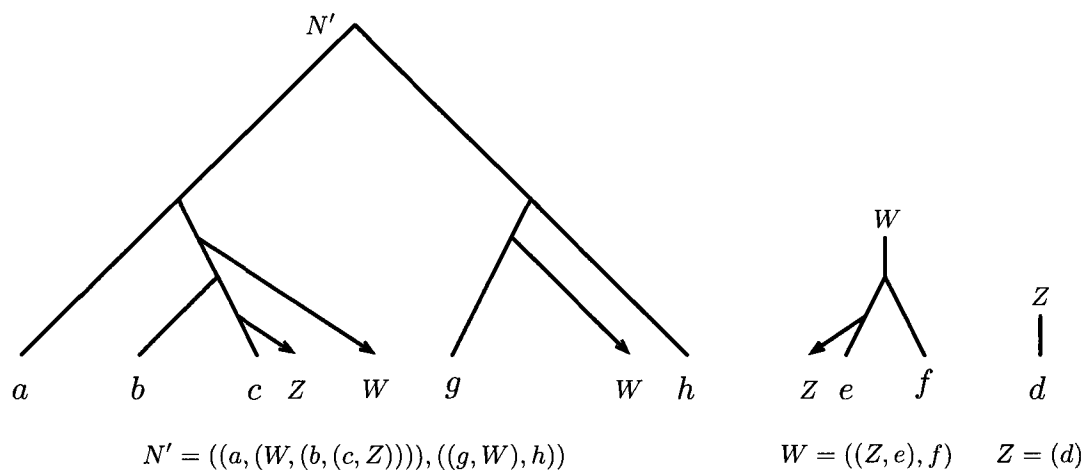


Figure 6.2 : Three trees,  $N'$ ,  $W$ , and  $Z$ , along with their Newick representation. These trees form the tree decomposition  $\mathcal{F}$  of the phylogenetic network  $N_1$  in Figure 6.1. The eNewick representation of  $N$  is the triplet  $\langle N'; W; Z \rangle$ .

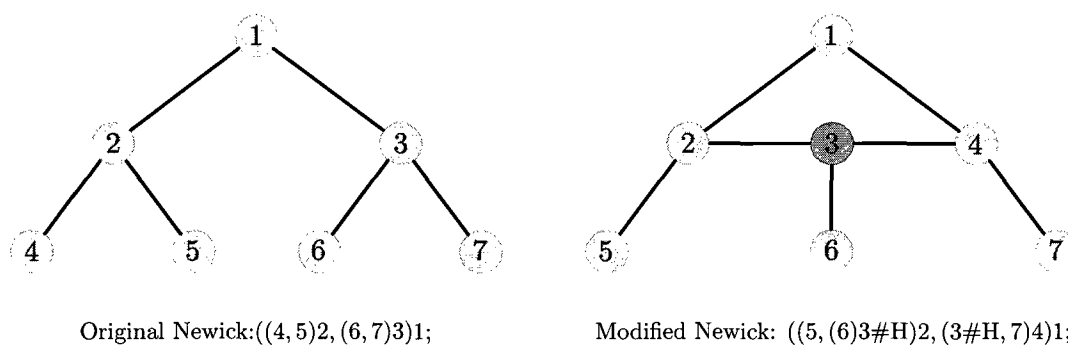


Figure 6.3 : A modified Newick format for representing phylogenetic networks. This example is from [106].

We have independently proposed a new method of *tree decomposition* of phylogenetic networks, which provides the basis for a new format, *extended Newick* (or eNewick for short), and used it as a compact representation of phylogenetic networks. The idea in our method is to break the network into a set of trees, and then represent the network as a collection of Newick representations of those trees. Since the eNewick format is nothing but a collection of trees in the Newick format, it follows that eNewick can represent unrooted networks. However, in the PhyloNet utilities, rooting is assumed, since different ways of rooting the same evolutionary networks may imply different evolutionary relationships.

Let  $N = (V, E)$  be a phylogenetic network, where  $V$  is the union of two disjoint sets  $V_N$ , the set of network nodes, and  $V_T$ , the set of tree nodes. We create a forest of  $|V_N| + 1$  trees as follows. For every  $u_i \in V_N$ :

- Compute the set  $\{v_1, \dots, v_k\}$  of nodes in  $V$  such that  $\{e_1 = (v_1, u_i), \dots, e_k = (v_k, u_i)\}$  is the set of all network edges incident into node  $u_i$ ;
- Create  $k$  new leaves, all labeled with  $x_i$  ( $x_i \cap \mathcal{L}(N) = \emptyset$ );
- Delete all  $k$  edges  $e_1, \dots, e_k$  incoming into  $u_i$ ;
- For each node  $v_j$ ,  $1 \leq j \leq k$ , add an edge from  $v_j$  to a unique leaf labeled with  $x_i$ .
- Assign  $x_i$  as the name of the tree rooted at node  $u_i$ ;

(Note here that each network node in  $V_N$  results in a tree. In the case  $V_N = \emptyset$ , we have one tree that is the original network.) The result is a forest of trees  $\mathcal{F} = \{t_1, \dots, t_{|V_N|+1}\}$  such that: (1)  $|\mathcal{L}(t_i)| \geq 1$  for every  $1 \leq i \leq |V_N| + 1$ ; (2)  $\bigcup_{i=1}^{|V_N|+1} \mathcal{L}(t_i) \setminus V_N = \mathcal{L}(N)$ ; and (3)  $\mathcal{L}(t_i) \cap \mathcal{L}(t_j) = \emptyset$  for every  $1 \leq i, j \leq |V_N| + 1$



and  $i \neq j$ . We call  $\mathcal{F}$  the tree decomposition of  $N$ . Then, the eNewick representation of  $N$  is the  $(|V_N| + 1)$ -tuple  $\langle n(t_1); \dots; n(t_{|V_N|+1}) \rangle$ , where  $n(t_i)$  is the Newick representation of tree  $t_i$ . Figure 6.2 shows the tree decomposition and eNewick representation of the network  $N_1$  in Figure 6.1.

In the case of modeling networks with horizontal gene transfer events, it is often very helpful to the biologist to know what the species tree is and what the additional set of HGT events are. Such information is “lost” in an eNewick representation, unless the representation is extended further to keep a record of the “species tree parent” of each network node. Therefore, in this case (which is the output of RIATA-HGT) we opt for the format of a species tree  $T$ , in Newick format, followed by a list of the HGT edges, each written as  $u \rightarrow v$ , where  $u$  and  $v$  are two nodes in  $T$ .

## 6.2 Evolutionary Network Characterization

As we described in Chapter 2, a phylogenetic network induces, or contains, a set of trees. The set of induced trees can be used to characterize phylogenetic networks. A tree  $T$  is induced by a network  $N$  if  $T$  is obtained from  $N$  as follows:

1. for each node of in-degree larger than one, remove all but one of the network edges incident into it; and
2. for every node of in-degree and out-degree 1, and whose parent is  $u$  and child is  $v$ , remove the two edges incident with it, and add an edge from  $u$  to  $v$ . We denote by  $\mathcal{T}(N)$  the set of all trees induced by  $N$ .

Figure 6.4 shows the sets  $\mathcal{T}(N_1)$  and  $\mathcal{T}(N_2)$  for the two networks  $N_1$  and  $N_2$  in Figure 6.1. It is important to note that this set of trees is completely different from the set of trees obtained by the tree decomposition we introduced to facilitate the

eNewick format. A phylogenetic network  $N$  induces at most  $\prod_{v \in V_N} \text{indeg}(v)$ , where the product is one if  $V_N$  is empty.

Given an evolutionary network  $N$ , the set  $\mathcal{T}(N)$  is unique. Further, this set informs about the possible gene histories that the network reconciles.

In addition to characterizing phylogenetic networks by the set of trees they induce, we consider a *cluster*-based characterization. This view of phylogenetic networks is very important for understanding the relationships among the “evolutionary perspective” of phylogenetic networks and the “clustering perspective”, which is adopted in various methods [107, 108]. Let  $T = (V, E)$  be a phylogenetic tree on set  $X$  of taxa and rooted at node  $r$ . Each node  $v \in V$  induces a cluster  $C_T(v)$ . The (nontrivial) clusters of tree  $T$  are the set  $\mathcal{C}(T) = \{C_T(v) : v \text{ is an internal node and } v \neq r\}$ . A straightforward way to extend this concept to phylogenetic networks is to define the set of clusters of phylogenetic network  $N$  as  $\mathcal{C}(N) = \bigcup_{T \in \mathcal{T}(N)} \mathcal{C}(T)$ . The clusters of the two networks  $N_1$  and  $N_2$  in Figure 6.1 are listed in Table 6.1.

In this form of cluster-based characterization, clusters are unweighted; equivalently, all clusters are weighted equally. One option of weighting the clusters is by considering the fraction of trees in which it appears. In other words, the weight of a cluster  $A$  can be computed as

$$w(A) = \frac{|\{T \in \mathcal{T}(N) : A \in \mathcal{C}(T)\}|}{|\mathcal{T}(N)|}. \quad (6.1)$$

This weighting scheme informs not only about the clusters of taxa that the network represents, but also how many gene trees in the input share each cluster. It is important to note here that this weighting of a cluster should not be confused with, or used in lieu of, support values of clusters, since a cluster may appear in only one gene tree and have a high support (e.g., by having a high bootstrap value on the edge that defines it) whereas a poorly supported cluster may appear in several trees.

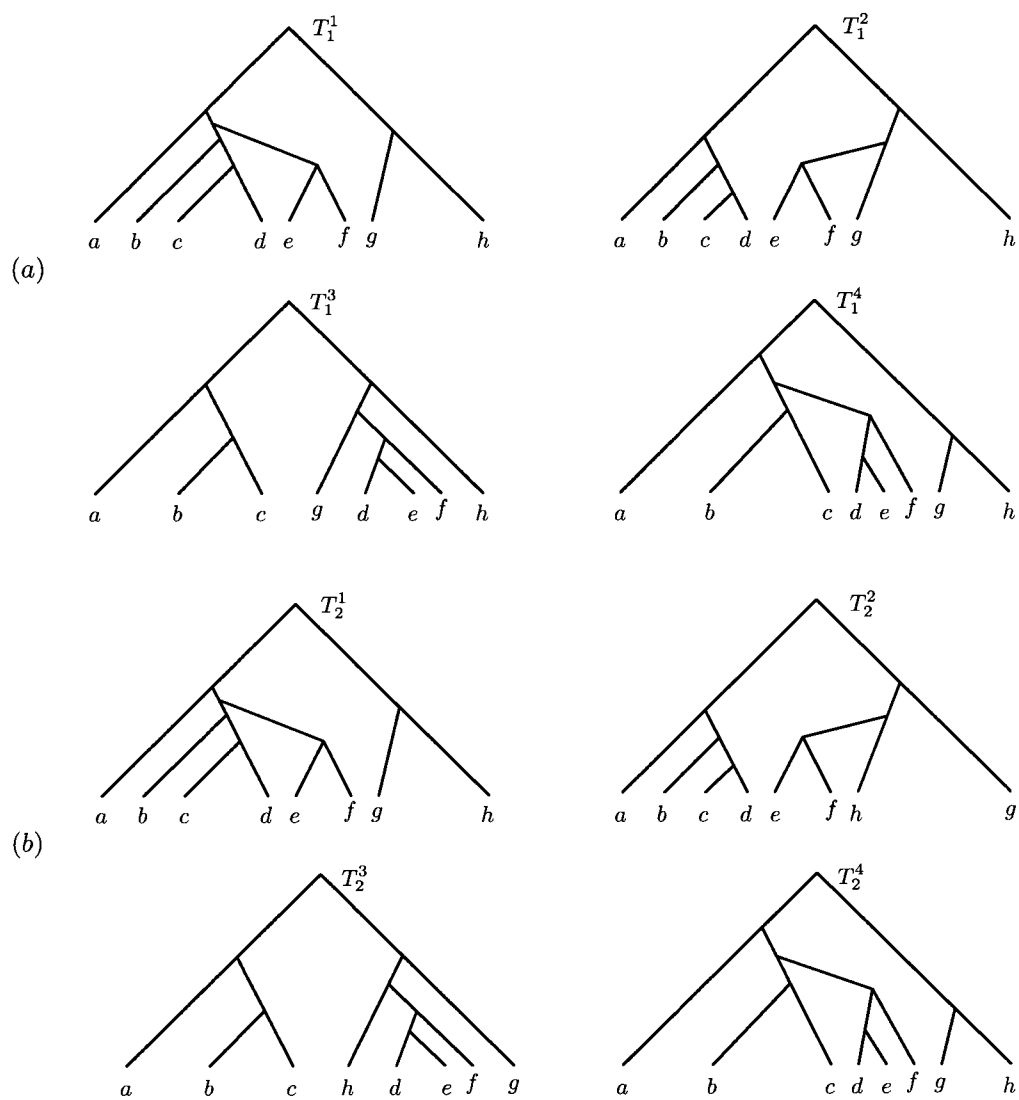


Figure 6.4 : The sets  $\mathcal{T}(N_1) = \{T_1^1, T_1^2, T_1^3, T_1^4\}$  and  $\mathcal{T}(N_2) = \{T_2^1, T_2^2, T_2^3, T_2^4\}$  of all eight trees induced by the two networks  $N_1$  and  $N_2$ , respectively, in Figure 6.1.

Table 6.1 : A table of the (nontrivial) clusters of the two networks  $N_1$  and  $N_2$  in Figure 6.1, denoted by  $\mathcal{C}(N_1)$  and  $\mathcal{C}(N_2)$ , respectively, in the text. Highlighted are rows corresponding to clusters that differ between the two networks.

Network $N_1$	Network $N_2$
$\{b, c\}$	$\{b, c\}$
$\{c, d\}$	$\{c, d\}$
$\{b, c, d\}$	$\{b, c, d\}$
$\{d, e\}$	$\{d, e\}$
$\{e, f\}$	$\{e, f\}$
$\{d, e, f\}$	$\{d, e, f\}$
$\{b, c, d, e, f\}$	$\{b, c, d, e, f\}$
$\{a, b, c\}$	$\{a, b, c\}$
$\{a, b, c, d\}$	$\{a, b, c, d\}$
$\{a, b, c, d, e, f\}$	$\{a, b, c, d, e, f\}$
$\{e, f, g\}$	$\{e, f, h\}$
$\{d, e, f, g\}$	$\{d, e, f, h\}$
$\{g, h\}$	$\{g, h\}$
$\{d, e, f, g, h\}$	$\{d, e, f, g, h\}$

Nakhleh and colleagues have recently introduced a new characterization of phylogenetic networks based on the *tripartitions* of their edges [109]. Let  $e = (u, v)$  be an edge in a phylogenetic network on set  $X$  of taxa and rooted at node  $r$ . We define three disjoint sets  $A_e = \{x \in X : r \rightsquigarrow^{[v]} x\}$ ,  $B_e = \{x \in X : r \rightsquigarrow^{[v]} x\}$ , and  $C_e = \{x \in X : r \not\rightsquigarrow^{[v]} x\}$ . Then, the tripartition induced by edge  $e$ , denoted  $\theta_e$ , is the triplet  $\langle A_e; B_e; C_e \rangle$ . Roughly speaking, the tripartition induced by an edge is the three sets of taxa reachable from the root only through that edge ( $A_e$ ), reachable through that edge but not exclusively ( $B_e$ ), and not reachable through that edge ( $C_e$ ). The set of (nontrivial) tripartitions induced by a phylogenetic network  $N$ , denoted by  $\theta(N)$ , is  $\{\theta_e \mid e \text{ is an internal edge in } E\}$ . As an example, tripartitions of the two networks  $N_1$  and  $N_2$  in Figure 6.1 are listed in Table 6.2.

Tripartition-based characterization of an evolutionary network helps to identify clades across which no genetic transfer occurred. If  $A_e = X$  and  $B_e = \emptyset$  for an edge  $e = (u, v)$ , this implies that the clade rooted at node  $v$  has set  $X$  of leaves, and there does not exist any exchange or transfer of genetic material between any organism in  $X$  and another organism that is not in  $X$ . Equivalently, an evolutionary network can be partitioned into a collection  $\{N_1, N_2, \dots, N_k\}$  of evolutionary networks that result from  $N$  by deleting every edge  $e$  for which  $B_e = \emptyset$ . Such a partition informs about the “locality” of reticulation events: each event in  $N$  is local to one of the  $k$  components in  $\{N_1, N_2, \dots, N_k\}$ . Further, this partition implies that each of the trees in  $\mathcal{T}(N)$  has  $k$  clades that have the sets  $\{\mathcal{L}(N_1), \mathcal{L}(N_2), \dots, \mathcal{L}(N_k)\}$  of leaves.

### 6.3 Evolutionary Network Comparison

Researchers are often interested in quantifying the similarities and differences between two phylogenies reconstructed either from two different sources of data or from

Table 6.2 : A table of the (nontrivial) tripartitions of the two networks  $N_1$  and  $N_2$  in Figure 6.1, denoted by  $\theta(N_1)$  and  $\theta(N_2)$ , respectively, in the text. Highlighted are rows corresponding to tripartitions that differ between the two networks.

Edge Label	Network $N_1$	Network $N_2$
1	$\langle \{a, b, c\}, \{d, e, f\}, \{g, h\} \rangle$	$\langle \{a, b, c\}, \{d, e, f\}, \{g, h\} \rangle$
2	$\langle \{g, h\}, \{d, e, f\}, \{a, b, c\} \rangle$	$\langle \{g, h\}, \{d, e, f\}, \{a, b, c\} \rangle$
3	$\langle \{b, c\}, \{d, e, f\}, \{a, g, h\} \rangle$	$\langle \{b, c\}, \{d, e, f\}, \{a, g, h\} \rangle$
4	$\langle \{g\}, \{d, e, f\}, \{a, b, c, h\} \rangle$	$\langle \{h\}, \{d, e, f\}, \{a, b, c, g\} \rangle$
5	$\langle \{b, c\}, \{d\}, \{a, e, f, g, h\} \rangle$	$\langle \{b, c\}, \{d\}, \{a, e, f, g, h\} \rangle$
6	$\langle \{c\}, \{d\}, \{a, b, e, f, g, h\} \rangle$	$\langle \{c\}, \{d\}, \{a, b, e, f, g, h\} \rangle$
7	$\langle \{e, f\}, \{d\}, \{a, b, c, g, h\} \rangle$	$\langle \{e, f\}, \{d\}, \{a, b, c, g, h\} \rangle$
8	$\langle \{e\}, \{d\}, \{a, b, c, f, g, h\} \rangle$	$\langle \{e\}, \{d\}, \{a, b, c, f, g, h\} \rangle$
9	$\langle \{d\}, \{\}, \{a, b, c, e, f, g, h\} \rangle$	$\langle \{d\}, \{\}, \{a, b, c, e, f, g, h\} \rangle$
10	$\langle \{d\}, \{\}, \{a, b, c, e, f, g, h\} \rangle$	$\langle \{d\}, \{\}, \{a, b, c, e, f, g, h\} \rangle$
11	$\langle \{e, f\}, \{d\}, \{a, b, c, g, h\} \rangle$	$\langle \{e, f\}, \{d\}, \{a, b, c, g, h\} \rangle$
12	$\langle \{e, f\}, \{d\}, \{a, b, c, g, h\} \rangle$	$\langle \{e, f\}, \{d\}, \{a, b, c, g, h\} \rangle$

two different reconstruction methods. Such a quantification provides insights into agreements and disagreements among analyses, confidence values for different parts of the phylogenies, and metrics for comparing the performance of phylogenetic reconstruction methods. In the context of phylogenetic trees, this quantification is most commonly done based on one of two criteria:

- *Topological differences.* The topologies, or shapes, of two phylogenetic trees are compared, and their differences are quantified. Several measures have been introduced to quantify topological differences and similarities between a pair of trees, such as the Robinson-Foulds measure and the SPR distance; see [110, 36] for a description of several such measures.
- *Fitness to sequence evolution.* When two phylogenies are reconstructed from the same sequence data set, it is common to compare them in terms of how well they model the evolution of the sequences. The most commonly used criteria for measuring such fitness are maximum parsimony, maximum likelihood, and the Bayesian posterior probability; see [110] for a detailed discussion of all three criteria.

In this section, we report on the capabilities in PhyloNet for comparing two evolutionary networks in terms of their topological differences and similarities, as well as in terms of their fitness to sequence evolution based on the maximum parsimony criterion.

For quantifying the dissimilarity between two evolutionary network topologies  $N_1$  and  $N_2$ , we want a measure  $m(\cdot, \cdot)$  that satisfies three conditions:

*Identity:*  $m(N_1, N_2) = 0$  if and only if  $N_1$  and  $N_2$  are *equivalent*;

*Symmetry:*  $m(N_1, N_2) = m(N_2, N_1)$ ; and

*Triangle inequality:*  $m(N_1, N_3) + m(N_3, N_2) \geq m(N_1, N_2)$  for any evolutionary network  $N_3$ .

This issue of evolutionary network equivalence was discussed in [109]. The three characterizations of evolutionary networks that we described above induce three measures which we now define. Let  $N_1$  and  $N_2$  be two evolutionary networks on the same set  $X$  of leaves; we define the three measures as follows.

### 6.3.1 Tree-based Comparison

Let  $\mathcal{T}(N_1)$  and  $\mathcal{T}(N_2)$  be the two sets of all trees induced by the two networks, and let  $d(\cdot, \cdot)$  be a distance metric on trees (see [36] for examples of such metrics). The idea is to compare the two networks based on how similar their corresponding sets of trees are. We formalize this as follows. Construct a weighted complete bipartite graph  $G(U_1, U_2, E)$ , where  $|U_i| = |\mathcal{T}(N_i)|$ , and there are two bijections  $f_i : U_i \rightarrow \mathcal{T}(N_i)$  for  $i = 1, 2$ . The weight of an edge  $e = (u, v) \in E$  for  $u \in U_1$  and  $v \in U_2$ ,  $w(e) = d(f_1(u), f_2(v))$ . Then, the tree-based measure  $m^{tree}(N_1, N_2)$  is defined as the weight of a minimum-weight edge cover of  $G$ . In its current implementation, PhyloNet uses the Robinson-Foulds distance measure [39] for  $d$ . For example, for the two networks in Figure 6.1, PhyloNet will return their tree-based distance as  $(0 + 2)/2 = 1.0$  because the network  $N'$  induces only two trees that are also induced by the network  $N$ . The tree-based measure was first introduced by Nakhleh *et al.* [111].

An illustration of tree-based comparison of the two networks  $N_1$  and  $N_2$  in Figure 6.1 is given in Figure 6.5. Shown on the left of the figure is the bipartite graph  $G$  built from the sets  $\mathcal{T}(N_1)$  and  $\mathcal{T}(N_2)$  of trees induced by the two networks; these two sets are shown in Figure 6.4. The weight of each edge connecting two nodes in  $G$  is the RF distance between the two trees corresponding to these two nodes. These



weights can be normalized by the number of internal edges in the trees. Since each of the eight trees has six internal edges, the weight of each edge in  $G$  can be divided by six to normalize it.

Shown on the right of Figure 6.5 is the minimum-weight edge cover of  $G$ , which is the set of edges that satisfies two conditions: (1) each node in  $G$  must be the endpoint of at least one edge in the set, and (2) the sum of the weights of the edges in the set is minimum among all sets of edges satisfying condition (1). In this case, the four edges shown are a cover, since each node in  $G$  is “covered” by at least one edge (here, each node is covered by exactly one edge). Further, it is of minimum weight, which equals 2, since a simple inspection yields that every other cover has a weight larger than 2. Since the cover has four edges in it, we have  $m^{tree}(N_1, N_2) = (0 + 0 + 1/6 + 1/6)/4 = 1/12$ . If we use the raw RF values, then  $m^{tree}(N_1, N_2) = (0 + 0 + 1 + 1)/4 = 1/2$ .

### 6.3.2 Cluster-based Comparison

We define the measure based on these two sets to be

$$m^{cluster}(N_1, N_2) = \left( \frac{|\mathcal{C}(N_1) \setminus \mathcal{C}(N_2)|}{|\mathcal{C}(N_1)|} + \frac{|\mathcal{C}(N_2) \setminus \mathcal{C}(N_1)|}{|\mathcal{C}(N_2)|} \right) / 2. \quad (6.2)$$

The rationale behind this measure is that it is the sum of the ratios of clusters present in one but not both networks. The cluster-based measure was first introduced by Nakhleh *et al.* [112]. The sets  $\mathcal{C}(N_1)$  and  $\mathcal{C}(N_2)$  of the two networks  $N_1$  and  $N_2$  in Figure 6.1 are listed in Table 6.1, with  $|\mathcal{C}(N_1)| = |\mathcal{C}(N_2)| = 14$ . Since  $|\mathcal{C}(N_2) \setminus \mathcal{C}(N_1)| = |\mathcal{C}(N_1) \setminus \mathcal{C}(N_2)| = 2$  (the two highlighted clusters in Table 6.1), we have  $m^{cluster}(N_1, N_2) = 1/7$ .

A similar weighting scheme to that described in the previous section can be used to incorporate the fraction of trees in which a cluster appears into the measure calculation.

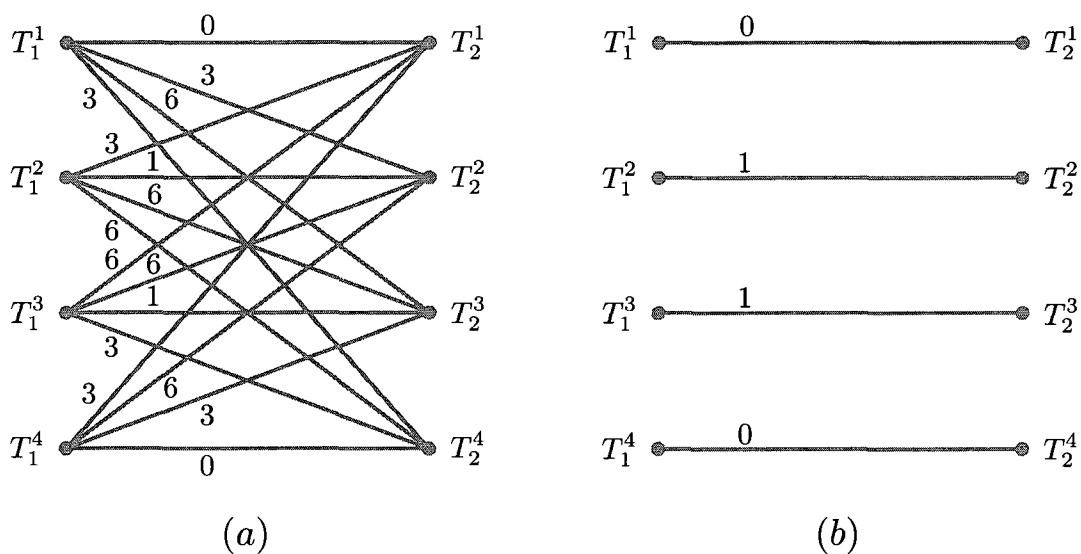


Figure 6.5 : Illustration of the tree-based network comparison measure. (a) The weighted bipartite graph  $G$  that is constructed from the two networks  $N_1$  and  $N_2$  in Figure 6.1. On the left are four nodes that correspond to the four trees in  $\mathcal{S}(N_1)$  and on the right are four nodes that correspond to the four trees in  $\mathcal{S}(N_2)$ . The weight of an edge between  $T_1^i$  and  $T_2^j$  is the values of the Robinson-Foulds (RF) distance between the two trees, which is computed as the number of clusters present in one but not both of the trees, divided by 2. (b) The edges that comprise the minimum-weight edge cover of the bipartite graph  $G$ . The weight of this cover is 2, which is the sum of the weights of the edges in the cover; therefore,  $m^{tree}(N_1, N_2) = 2$ .

### 6.3.3 Tripartition-based Comparison

We define the measure based on these two sets to be

$$m^{\text{tripartition}}(N_1, N_2) = \frac{|\theta(N_1) - \theta(N_2)|}{|\theta(N_1)|} + \frac{|\theta(N_2) - \theta(N_1)|}{|\theta(N_2)|}. \quad (6.3)$$

This measure views the two networks in terms of the sets of edges they define (where an edge is in a 1-1 correspondence with a tripartition) and computes the sum of the ratios of edges present in one but not both networks. The tripartition-based measure was devised by Moret *et al.* [109]. The sets  $\theta(N_1)$  and  $\theta(N_2)$  of the two networks  $N_1$  and  $N_2$  in Figure 6.1 are listed in Table 6.2, with  $|\theta(N_1)| = |\theta(N_2)| = 12$ . Since  $|\theta(N_1) - \theta(N_2)| = |\theta(N_2) - \theta(N_1)| = 1$  (the highlighted tripartition in Table 6.2), we have  $m^{\text{tripartition}}(N_1, N_2) = 1/12$ .

### 6.3.4 Which Measure to Use?

Several distance measures, such as the Robinson-Foulds measure and the Subtree Prune and Regraft (SPR) distance, have been introduced over the years to quantify the difference between the topologies of a pair of phylogenetic trees; e.g., see [110, 36] for description of many of these measures. Even though these measures may compute different distance values on the same pair of trees, there has been no consensus as to which measure should be used in general [113]. It may be the case that the Robinson-Foulds measure is more commonly used than the others, but this may be a mere reflection of its very low time requirements as compared to the other, more compute-intensive, measures.

Regarding the three measures for comparing networks, a scenario analogous to that in phylogenetic trees arises here: each measure gives a different quantification of the dissimilarity between two networks based on one of the three ways to characterize

a given network. As shown in the examples above, some or all of these measures may compute the same value for a given pair of networks, but that may not always be the case. Tree-based comparison of networks can be viewed as a method to quantify how similar, or dissimilar, two networks are in terms of their quality as a summary of a collection of trees. In some cases, even though two networks “look different,” they may be identical in terms of the trees they induce—this is the issue of indistinguishability of networks from a collection of trees that Nakhleh and colleagues discussed in [109]. In such a case, using the tree-based comparison, or equivalently the cluster-based comparison, is most appropriate. However, if the similarity/dissimilarity of two networks means something close to an *isomorphism*, then the tripartition-based measure is more appropriate. However, it is important to note that none of the three measures described here is a metric on the general space of all evolutionary networks labeled by a given set of taxa.

A practical distinction among the three measures can be derived based on the methods used to infer the evolutionary history of the set of species under study. Methods such as SplitsTree [107] and NeighborNet [108] represent the evolutionary history as a set of splits, or clusters, hence making it more natural to use cluster-based comparison to study their performance. Methods such as RIATA-HGT [47] and LatTrans [44] compute evolutionary networks that are rooted, directed, acyclic graphs, where internal nodes have an evolutionary implication in terms of ancestry. For these two methods, all three measures are appropriate. When the evolutionary history of a set of species is represented as a collection of its constituent gene trees, the tree-based measure is most appropriate.

Finally, a clear distinction can be made among the methods in terms of computational requirements. In their current implementations, the tripartition-based

measure is very fast in practice, taking time that is polynomial in the size of the two networks. On the other hand, the tree- and cluster- based measures are much slower, taking time that is exponential in the number of network nodes in the two networks (since these measures compute explicitly all trees inside each of the two networks). In light of recent complexity results that we obtained [114], it is very likely that no polynomial-time algorithms exist for computing the tree- and cluster-based measures in general.

### 6.3.5 Parsimony of Evolutionary Networks

Nakhleh and colleagues have recently formalized a maximum parsimony (MP) criterion for evolutionary networks [115] and demonstrated its utility in reconstructing evolutionary networks on both biological and synthetic data sets [102]. In this section, we describe a PhyloNet utility that allows for comparing two evolutionary networks in terms of their fitness to the evolution of a sequence data set, based on the MP criterion. We first begin with a brief review of the MP criterion, based on the exposition in [115].

The relationship between an evolutionary network and its constituent trees, as described in the background section, is the basis for the MP extension to evolutionary networks.

**Definition 6.1.** *The Hamming distance between two equal-length sequences  $x$  and  $y$ , denoted by  $H(x, y)$ , is the number of positions  $j$  such that  $x_j \neq y_j$ .*

Given a fully-labeled tree  $T$ , i.e., a tree in which each node  $v$  is labeled by a sequence  $s_v$  over some alphabet  $\Sigma$ , we define the Hamming distance of an edge  $e = (u, v) \in E(T)$ , denoted by  $H(e)$ , to be  $H(s_u, s_v)$ . We now define the parsimony score of a tree  $T$ .

**Definition 6.2.** *The parsimony score of a fully-labeled tree  $T$ , is  $\sum_{e \in E(T)} H(e)$ . Given a set  $S$  of sequences, a maximum parsimony tree for  $S$  is a tree leaf-labeled by  $S$  and assigned labels for the internal nodes, of minimum parsimony score.*

The parsimony definitions can be extended in a straightforward manner to incorporate different site substitution matrices, where different substitutions do not necessarily contribute equally to the parsimony score, by simply modifying the formula  $H(x, y)$  to reflect the weights. Let  $\Sigma$  be the set of states that a site can take (e.g.,  $\Sigma = \{\text{A, C, T, G}\}$  for DNA sequences), and  $W$  the site substitution matrix such that  $W(\sigma_1, \sigma_2)$  is the weight of replacing  $\sigma_1$  by  $\sigma_2$ , for every  $\sigma_1, \sigma_2 \in \Sigma$ . In particular, the *identity* site substitution matrix satisfies  $W(\sigma_1, \sigma_2) = 0$  when  $\sigma_1 = \sigma_2$ , and  $W(\sigma_1, \sigma_2) = 1$  otherwise. The weighted Hamming distance between two sequence is  $H(x, y) = \sum_{1 \leq i \leq k} W(x_i, y_i)$ , where  $k$  is the length of the sequences  $x$  and  $y$ . The rest of the definitions are identical to the simple Hamming distance case.

As described above, the evolutionary history of a single (non-recombining) gene is modeled by one of the trees contained inside the evolutionary network of the species containing that gene. Therefore the evolutionary history of a site  $p$  is also modeled by a tree contained inside the evolutionary network. A natural way to extend the tree-based parsimony score to fit a dataset that evolved on a network is to define the parsimony score for each site as the minimum parsimony score of that site over all trees contained inside the network.

**Definition 6.3** ([115]). *The parsimony score of a network  $N$  leaf-labeled by a set  $S$  of taxa, is*

$$NCost(N, S) = \sum_{p \in S} \min\{T \in \mathcal{T}(N) : TCost(T, p)\} \quad (6.4)$$

where  $TCost(T, p)$  is the parsimony score of site  $p$  on tree  $T$ .

Notice that as usually large segments of DNA, rather than single sites, evolve together, Definition 6.3 can be extended easily to reflect this fact, by partitioning the sequences  $S$  into non-overlapping blocks  $b$  of sites, rather than sites  $p$ , and replacing  $p$  by  $b$  in Definition 6.3. This extension may be very significant if, for example, the evolutionary history of a gene includes some recombination events, and hence that evolutionary history is not a single tree. In this case, the recombination breakpoint can be detected by experimenting with different block sizes.

The MP utility in PhyloNet allows the user to specify two evolutionary networks (either or both of which can be a tree)  $N_1$  and  $N_2$  and a sequence data set  $S$ , and computes the parsimony scores  $NCost(N_1, S)$  and  $NCost(N_2, S)$ . The user can then compare the two scores and evaluate the fitness of the networks to the data set  $S$  based on the difference in the scores. Further, the utility allows the user, for example, to evaluate the significance of each network edge in a network  $N$  by comparing the parsimony scores of two different versions of  $N$  that contain different subsets of the network edges in  $N$ .

## 6.4 Inferring Species Trees From Gene Trees

In this section, we describe tools that implement algorithms in Chapters 3 and 4 for inferring species trees from gene trees. In Chapter 3, we present an ILP-based algorithm for inferring the species tree's topology and its branch lengths. As the algorithm involves a number of substeps, PhyloNet divide the implementation of this algorithm into several tools:

- **genst**: For generating species tree topologies that correspond to maximal compatible cliques in the compatibility graph constructed from clusters induced by input gene trees.

- **gencplex**: For generating CPLEX programs. The tool reads a species tree candidate and a set of gene trees, and it creates an ILP program as described in Algorithm 3.3.
- **compute\_st**: For computing the species tree—both its topology and branch lengths—from gene trees. This tool implements Algorithm 3.4, and is written as a Perl script. Essentially, the script calls **genest** to generate species tree topology candidates, for each of them it calls CPLEX to solve an ILP program produced by tool **gencplex**, and finally choose the best tree according to the optimality criterion  $\eta$  described in Subsectionsubsec:ch3-algorithm.

There are two tools in PhyloNet for inferring the species tree using the MDC criterion: `coal_infer_st`, which is an implementation of the ILP algorithm (Section 4.4), and `dpcoal_infer_st`, which is an implementation of the DP algorithm (Section 4.5). To use the first tool, the user invokes PhyloNet as follows:

```
java -jar phylonet.jar coal_infer_st cplexpath gt
```

In this case, `cplexpath` is the path to CPLEX (the ILP solver) on the user's computer, and `gt` is the name of the file that contains all input gene trees (each gene tree written in the Newick format on a separate line). For the second tool, the user invokes PhyloNet as follows:

```
java -jar phylonet.jar dpcoal_infer_st gt
```

In this case, only file `gt`, which contains all gene trees, needs to be specified.

As an example, suppose we have a file named `input` that contains the gene trees:

```
T1 = (((a, b), c), d), e);
T2 = ((a, b), (d, (c, e)));
T3 = ((a, c), (d, (b, e)));
```



Then, to infer the species tree under MDC, from these gene trees, by using the DP algorithm, the user can type the command:

```
java -jar phylonet.jar dpcoal_infer_st input
```

which returns  $((((a, b), c), d), e)$  as the species tree.

Those tools also implement our extension to MDC to handle non-binary and multiple-allele gene trees (Section 4.6). For non-binary trees, the input to those tools are unchanged. However, in the case where multiple individuals per species may be sampled, the user needs to supply a mapping between gene tree taxa and species tree taxa in a separate file. If a total of  $k$  individuals are sampled from all species, then this mapping file contains  $k$  lines, each line containing two entries:

```
ind    sp
```

where *ind* is the label of an individual and *sp* is the label of the species to which *ind* belongs. For example, suppose we have a file *gt* that contains two gene trees:

```
T1 = ((a1, a2), ((b1, c1), (b2, c2)));
```

```
T2 = (((a1, b1), (c1, b2)), (a2, c2));
```

where  $a_1, a_2$  are two sampled individuals of species  $a$ ;  $b_1, b_2$  are two sampled individuals of species  $b$ ; and  $c_1, c_2$  are two sampled individuals of species  $c$ . Then, in order to reconstruct the species tree for the three species  $a, b$ , and  $c$ , using the DP algorithm for solving MDC, the user invokes the command:

```
java -jar phylonet.jar dpcoal_infer_st gt -m map
```

where file *map* contains the following lines

a1	a
a2	a
b1	b
b2	b
c1	c
c2	c

For this example, the inferred species tree estimate is  $(a, (c, b))$ .

## 6.5 Reconstructing Evolutionary Networks from Species Trees and Gene Trees

Assuming incongruence among gene and species trees is the result of HGT events only, the *HGT Reconstruction Problem* is to find the smallest number of HGT events to reconcile the incongruence. This problem has been shown to be NP-complete [43]. In [47], Nakhleh *et al.* introduced an accurate, polynomial-time heuristic, RIATA-HGT, for solving the HGT Reconstruction Problem for a pair of species and gene trees. In a nutshell, the method computes the maximum agreement subtree [67] of the species tree and each of the gene trees, and adds HGT edges to connect all subtrees that do not appear in the maximum agreement subtree. Theoretically, RIATA-HGT may not compute the minimum-cardinality set of HGT events; nonetheless, experimental results show very good empirical performance on synthetic as well as biological data [47].

RIATA-HGT was designed originally to compute a single solution to the problem, and was mainly aimed at binary trees. Later, Than *et al.* [48] extended the method

to compute multiple solutions and to handle non-binary trees. These two features are very significant: the former allows biologists to explore multiple potential HGT scenarios, whereas the latter allows for analyzing trees in which some edges were contracted due to inaccuracies (see [116], for example). We have conducted an experimental study to compare the performance of RIATA-HGT with LatTrans [54]. Although RIATA-HGT and LatTrans [44] have almost the same performance in terms of the number of HGT solutions and the solution size, the former runs much faster than the latter.

For a compact representation of multiple solutions, we introduce four terms:

- An *event*: this is a single HGT edge, written in the form of  $u \rightarrow v$ , where  $u$  and  $v$  are two nodes in the species tree.
- A *subsolution*: this is an *atomic* set of events, which forms a part of an overall solution. In other words, either all or none of the events of a subsolution are taken in a solution.
- A *component*: a set of components and/or subsolutions. Two components at the same level of decomposition are independent, in that an element of each component is needed to form a solution.
- A *solution*: the union of a single element from each component at the highest level.

To illustrate these concepts, consider species tree  $((a, b), c), (d, (e, f))$  and the gene tree  $((a, c), b), ((d, f), e)$ . Observe, that each HGT event required to reconcile the two trees has both endpoints in the subtree  $((a, b), c)$  or both endpoints in the subtree  $(d, (e, f))$ , and no HGT event has endpoints in both subtrees. In this case, RIATA-HGT divides the pair of trees into two pairs:

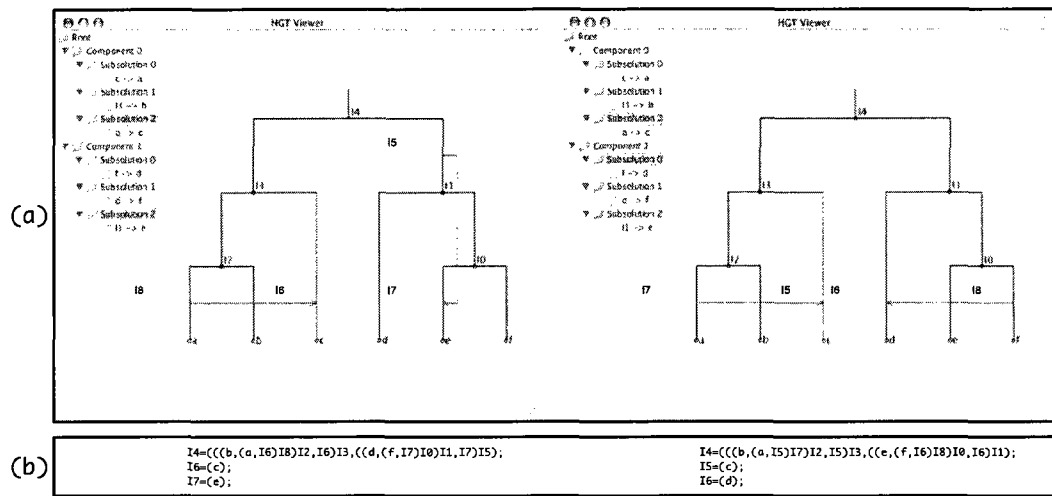


Figure 6.6 : (a) Screen captures of the graphical output of RIATA-HGT on the pair of trees  $((a, b), c)$  and  $((a, c), b)$  and  $((d, e), f)$  and  $((d, f), e)$ . (b) The eNewick representations of the two selected networks.

- Pair 1:  $((a, b), c)$  and  $((a, c), b)$ ,
- Pair 2:  $((d, e), f)$  and  $((d, f), e)$ ,

and solves the HGT Reconstruction Problem on each of the two pairs *independently*. The set of solutions of each pair is a component. Notice that for each pair there are three possible ways to reconcile them; each such way is called a subsolution. Each subsolution is a set of events, which in this case is only one event. Figure 6.6(a) shows the screen captures of two graphical outputs that correspond to two solutions on this pair of trees. Notice that if a component can be further divided into independent components, RIATA-HGT would do so, which will result in components at different levels, with the largest components being at the highest level.

The compact representation of RIATA-HGT's output in terms of subsolutions and components is especially helpful when the number of solutions is large. RIATA-HGT also has an option to display all complete solutions. RIATA-HGT enumerates

all complete solutions that are compactly represented as described in the preceding paragraphs. Each solution, which is a set of HGT events, along with the species tree defines an evolutionary network, which RIATA-HGT displays in the eNewick format. For example, for the trees  $((a, b), c), (d, (e, f))$  and  $((a, c), b), ((d, f), e)$ , RIATA-HGT outputs 9 different networks in the eNewick format, if RIATA-HGT's option for displaying complete solutions is on. Figure 6.6(b) shows the corresponding eNewick representations.

From the multiple comparisons between a species and a set of trees, RIATA-HGT offers a (strict) consensus network. For each pair of species tree and gene tree, RIATA-HGT computes a set of HGT events for reconciling them. To obtain the consensus network, RIATA-HGT retains only HGT events that appear in every set of solutions for every pair of species tree and gene tree. Those events are then added to the species to build the consensus network.

We note here that while offering a simple summary of solutions, this way of computing consensus networks may not be appropriate in general; work is under way to address this issue more properly.

## 6.6 Phylogenetic Tree Utilities

As evident from the description of the methods above, there are fundamental correlations between phylogenetic trees and networks. Hence, many of the phylogenetic network utilities use functionalities from the phylogenetic trees domain, which we have implemented and provide as standalone tools in PhyloNet:

- A tool for computing the maximum agreement subtree (MAST) of a pair of trees using the algorithm of Steel and Warnow [67]. We also extended the algorithm so that it computes *all* MASTs of a pair of tree, and this feature is implemented

as well.

- A tool for computing the Robinson-Foulds distance measure between two phylogenetic trees [39].
- A tool for computing the *least common ancestor* (lca) of a set of nodes in a phylogenetic tree [89].

## 6.7 Implementation

A major goal for the PhyloNet package was to make its functionality platform-independent and accessible both to end users for data analysis and to researchers designing new computational methods and techniques. In order to encompass as many platforms as possible, PhyloNet was implemented in Java. As a result, any system with the Java 2 Platform (Version 5.0 or higher) installed can run PhyloNet.

PhyloNet can be used in two ways, depending on how the functionality needs to be accessed. A command-line interface exposes all of PhyloNet's tools on a Unix or DOS command-line. Each command accepts input from and writes output to text files. This allows PhyloNet's functionality to be used for manual data analysis or integrated into scripts for performing larger-scale processing. Additionally, a rich and thoroughly documented object model allows the incorporation of any of PhyloNet's functionality into existing Java programs. Also bundled are various programmatic utilities that represent trees, networks, and that read and write these various data structures to and from files.

## 6.8 The Command Line Interface

PhyloNet has a consistent and easy-to-use command line interface. A detailed discussion of this interface and all available options is available in the documentation that accompanies a download of the tool. Here we provide a brief overview of the design of the command-line tool and the tools that can be accessed.

Table 6.3 lists all the commands that are currently available from the command-line. Each of these commands accepts a set of parameters as command-line arguments. All trees, networks, sequences, and other major data structures are read in either from *standard in* or from text files. Similarly all results can be written either to *standard out* or to a desired text file. All trees are read and written in Newick format. Networks are read and written in eNewick format. These design features allow the easy use of PhyloNet for manual data analysis or as a tool used within a larger scripted automated analysis.

To run a tool in PhyloNet, invoke the executable `.jar` file downloaded from the PhyloNet project homepage with appropriate tool and its arguments, for example,

```
java -jar phylonet.jar charnet -i net.in -m tree
```

where `phylonet.jar` is the executable jar downloaded from the project homepage (the file is assumed to be in the current directory where the user invokes this command), `charnet` is the name of the tool that decomposes the network contained in file `net.in` into a set of trees. The reference manual included with the executable jar provides very detailed instructions regarding how to run each tool in the PhyloNet package.

Table 6.3 : A table of the tools currently implemented in PhyloNet. With the exception of the three phylogenetic trees tools `lca`, `mast`, and `rf`, all the other tools are for analyzing reticulate evolutionary relationships.

<b>Tool name</b>	<b>Purpose</b>
<code>charnet</code>	Computing clusters, trees and tripartitions in a network
<code>cmpnets</code>	Computing the distance between two networks
<code>lca</code>	Finding the least common ancestor of a set of nodes
<code>mast</code>	Computing the maximum agreement subtree
<code>netpars</code>	Scoring the parsimony of sequences on a network
<code>recomp</code>	Detecting interspecific recombination breakpoints in a sequence alignment
<code>riatahgt</code>	Reconstructing HGT events from a pair of species/gene trees
<code>rf</code>	Computing the Robinson-Foulds tree measure
<code>compute_st</code>	Computing the species tree's shape and branch lengths from gene trees
<code>coal_infer_st</code>	Inferring the species tree topology from gene trees, using an ILP formulation
<code>dpcoal_infer_st</code>	Inferring the species tree topology from gene trees, using a dynamic programming algorithm.



### 6.8.1 Programmatic Interface

Many phylogenetic methods comprise critical, but intermediate, steps in much larger methods. As a result, there is also a need for the functionality in PhyloNet to be available for incorporation into larger programs. As a result, all of PhyloNet's functionality is exposed through an extensive set of Java classes. Each tool is contained within its own Java class and exposes a carefully constructed set of public methods that will be preserved and maintained even as PhyloNet grows. This modular design allows for the easy addition functionality in the future without effecting existing programs that use PhyloNet as a programmatic library.

In addition to exposing a consistent API, PhyloNet also provides implementations of the most common phylogenetic data structures: trees and networks. Utility classes are also included that read and write these data structures to and from files. These classes can accelerate not only incorporation of PhyloNet's functionality, but also the development of new phylogenetic functionality within other applications.

As PhyloNet grows, programmatic interfaces will be added to provide access to new functionality and tools. Detailed documentation of these libraries is available in JavaDoc form on the PhyloNet website.

## 6.9 Conclusions

Analyzing and understanding reticulate evolutionary relationships have been hindered by the lack of software tools for conducting these tasks. The software package, PhyloNet, offers an array of utilities to allow for efficient and accurate analysis of such evolutionary relationships. These utilities allow for reconstructing phylogenetic networks from pairs of species/gene trees, detecting interspecific recombination in a

sequence alignment, scoring the parsimony of a phylogenetic network with sequences at its leaves, characterizing phylogenetic networks in terms of their basic units, and comparing the topologies of phylogenetic networks to quantify their similarities.

The software package will help significantly in analyzing large data sets, as well as in studying the performance of phylogenetic network reconstruction methods. Further, the software package offers the novel eNewick format for compact representation of phylogenetic networks, a feature that allows for efficient interoperability of phylogenetic network software tools.

## Chapter 7

### Conclusions

#### 7.1 Discussion

In this dissertation, we present three methods for inferring species trees from multiple gene trees despite lineage sorting. Their elegance lies in the fact that they explicitly model the process of species/gene tree incongruence during the inference process, but at the same time they do not introduce too much complexities as maximum likelihood and Bayesian methods do. Instead, they all use a simple parsimony score to measure the severity of deep coalescence in inferring the species tree. In the first algorithm, the score is the depth of coalescence events. In the second and third algorithms, the criterion used is the MDC, or minimizing deep coalescences, first introduced in [5]. Although they are simple, the experimental study on both biological and simulated data that we carried out shows that they have accuracy competitive with probabilistic methods, while they run significantly faster.

Our investigation of the MDC criterion also results in a simple, but interesting, formula for computing the number of extra lineages for individual clusters. With this formula, we are able to replace the problem of finding an optimal tree under the MDC criterion by the the problem of finding an optimal set of compatible clusters. The improvement is huge here, since we know that the total number of rooted binary trees on  $n$  taxa is  $(2n - 3)!!$ , which is  $O(\sqrt{2\pi n}(n/e)^n)$  using Stirling's approximation, while the number of all possible clusters is  $O(2^n)$ . Certainly, finding an optimal

set of compatible clusters among those  $O(2^n)$  might be as bad as  $O(\sqrt{2\pi n}(n/e)^n)$ . However, this is not the case, since we show in Chapter 4 that by using the dynamic programming algorithm, the complexity is  $O(3^n)$ . Besides, we made an empirical observation that it is sufficient to work with clusters induced by gene trees, as those clusters almost always contain species tree clusters (see Chapter 4). This observation makes the dynamic programming algorithm polynomial of the number of input gene trees and of the number of taxa.

We also show how to extend the MDC criterion to phylogenetic networks; see Chapter 5. This extension to the MDC provides us with a means to detect hybridization despite lineage sorting. By combining lineage sorting into hybridization detection, the new technique overcomes limitations in traditional phylogeny-based hybridization detection methods as they often overestimate the amount of hybridization. Using this method, we proposed an interesting evolutionary history on the yeast data set.

## 7.2 Future Research

There are several open questions and interesting research projects related to the content of this dissertation. One of them is the complexity of inferring species trees using the MDC criterion. This question is still unresolved at the time of writing, but we doubt that it is in class P (i.e., the class of polynomial algorithms). The reason is that we might need to look at all possible clusters in order to find the optimal tree, and that number is already exponential. Furthermore, a closely related problem (in terms of mathematical modeling) of inferring species trees from gene trees by minimizing gene duplications and extinctions has been shown to be NP-complete.

An immediate project that would be of interest is to improve the inference of

species trees using the MDC criterion with time included. The current MDC criterion for phylogenetic trees allows for elegant approaches to solve the MDC optimization problem; see Chapter 4. Including time to the MDC criterion is not a problem, since we always count the number of extra lineages by visiting the species tree's branches. And as described in Chapter 3, we did have an algorithm for inferring species trees when time information is used. The question is: Can we eliminate the phase of generating species tree topology candidates as we did in Chapter 4 with the original MDC criterion? If this question is solved, then it would be of huge significance as branch lengths contain a wealth of information that is valuable to the inference process. And being a parsimony method, it can be expected to run faster than maximum likelihood and Bayesian methods.

We note that the efficiency we achieve in the dynamic programming algorithm for the MDC criterion stems from the fact that we reconstruct the optimal tree from combining gradually compatible optimal clusters. This is a powerful approach, since the space of clusters is much smaller than the space of rooted trees. In [55], Degnan and Salter provides a general formula for computing the probability of a gene tree given a species tree under the coalescent model. In this formula, the probabilities for each valid coalescent history is computed (a valid coalescent history is an ordering of coalescence events on the species tree's branches such that both the MRCA mapping and the gene tree topology are respected), and then they are summed together. The complexity here is that the number of valid coalescent histories is often huge [114]. Our preliminary investigation of the formula shows that it might be possible to use the cluster approach for its computation.

Concerning the MDC for phylogenetic networks, we note that in Chapter 5 we compute the number of extra lineages required to reconcile a gene tree within a

network by taking the minimum of those numbers for reconciling that gene tree within trees induced by the network. This implies that we might need to look at all trees induced by the network, which can be exponential of the number of reticulate events. We currently do not have an efficient algorithm to carry out this computation, but it is worth investigating since it would make our proposed method for hybridization method application for large data sets.

## Bibliography

- [1] R. V. Eck and M. O. Dayhoff, *Atlas of protein sequence and structure*. Silver Spring, MD: National Biomedical Research Foundation, 1966.
- [2] B. Pierce, *Genetics: A conceptual approach*. W. H. Freeman, 2008.
- [3] W. M. Fitch, "Distinguishing homologous from analogous proteins," *Systematic Zoology*, vol. 19, pp. 99–113, 1970.
- [4] M. Nei, "Standard error of immunological dating of evolutionary time," *Journal of Molecular Evolution*, vol. 9, pp. 203–11, 1977.
- [5] W. P. Maddison, "Gene trees in species trees," *Systematic Biology*, vol. 46, no. 3, pp. 523–536, 1997.
- [6] Y. Tateno, N. Takezaki, and M. Nei, "Accuracy of estimated phylogenetic trees from molecular data," *Journal of Molecular Evolution*, vol. 18, no. 387–404, 1982.
- [7] F. Sanger, G. M. Air, B. G. Barrell, N. L. Brown, A. R. Coulson, C. A. Fiddes, C. A. Hutchison, P. M. Slocombe, and M. Smith, "Nucleotide sequence of bacteriophage  $\phi$ X174," *Nature*, vol. 265, pp. 687–695, 1977.
- [8] R. Fleischmann, M. Adams, O. White, R. Clayton, E. Kirkness, A. Kerlavage, C. Bult, J. Tomb, B. Dougherty, J. Merrick, *et al.*, "Whole-genome random se-

- quencing and assembly of haemophilus influenza rd.," *Science*, vol. 269, pp. 496–498, 507–512, 1995.
- [9] A. Goffeau, et al., "The yeast genome directory," *Nature*, vol. 387, no. suppl., pp. 1–105, 1997.
- [10] B. Rannala and Z. Yang, "Phylogenetic inference using whole genomes," *Annual Review of Genomics and Human Genetics*, vol. 9, pp. 217–231, 2008.
- [11] H. Philippe, F. Delsuc, H. Brinkmann, and N. Lartillot, "Phylogenomics," *Annual Review of Ecology, Evolution, and Systematics*, vol. 36, pp. 541–562, 2005.
- [12] B. E. Blaisdell, "A measure of the similarity of sets of sequences not requiring sequence alignment," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 83, pp. 5155–5159, 1986.
- [13] D. Sankoff, G. Leduc, N. Antoine, B. Paquin, B. F. Lang, and R. Cedergren, "Gene order comparisons for phylogenetic inference: Evolution of the mitochondrial genome," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 89, pp. 6575–6579, 1992.
- [14] G. Bourque and P. A. Pevzner, "Genome-scale evolution: reconstructing gene orders in the ancestral species," *Genome Research*, vol. 12, pp. 26–36, 2002.
- [15] M. Blanchette, G. Bourque, and D. Sankoff, "Breakpoint phylogenies," in *Genome Informatics* (S. Miyano and T. Takagi, eds.), pp. 25–34, Tokyo: Universal Academy Press, 1997.
- [16] S. T. Fitz-Gibbon and C. H. House, "Whole genome-based phylogenetic analysis of free-living microorganisms," *Nucleic Acids Research*, vol. 27, no. 4218–4222,



1990.

- [17] S. Yang, R. F. Doolittle, and P. E. Bourne, "Phylogeny determined by protein domain content," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 373-378, 2005.
- [18] J. Lin and M. Gerstein, "Whole-genome trees based on the occurrence of folds and orthologs: Implications for comparing genomes on different levels," *Genome Research*, vol. 10, pp. 808–818, 2000.
- [19] M. Miyamoto and W. Fitch, "Testing species phylogenies and phylogenetic methods with congruence," *Systematic Biology*, vol. 44, no. 1, pp. 64–76, 1995.
- [20] Y. I. Wolf, I. B. Rogozin, N. V. Grishin, and E. V. Koonin, "Genome trees and the tree of life," *Trends in Genetics*, vol. 18, pp. 472–479, 2002.
- [21] B. M. E. Moret, J. Tang, and T. Warnow, "Reconstructing phylogenies from gene-content and gene-order data," in *Mathematics of evolution and phylogeny* (O. Gascuel, ed.), pp. 321–352, Oxford University Press, 2005.
- [22] J. A. Lake and M. C. Rivera, "Deriving the genomic tree of life in the presence of horizontal gene transfer: Conditioned reconstruction," *Molecular Biology and Evolution*, vol. 21, pp. 681–690, 2004.
- [23] R. R. Copley, P. Aloy, R. B. Russell, and M. J. Telford, "Systematic searches for molecular synapomorphies in model metazoan genomes give support for Ecdysozoa after accounting for the idiosyncrasies of *Caenorhabditis elegans*," *Evolution and Development*, vol. 6, pp. 164–169, 2004.

- [24] A. G. Kluge, "A concern for evidence and a phylogenetic hypothesis of relationships among Epicrates (Boidae, Serpentes)," *Systematic Zoology*, vol. 38, pp. 7–25, 1989.
- [25] A. Rokas, B. L. Williams, N. King, and S. B. Carroll, "Genome-scale approaches to resolving incongruence in molecular phylogenies," *Nature*, vol. 425, pp. 798–804, 2003.
- [26] W. J. Murphy, E. Eizirik, W. E. Johnson, Y. P. Zhang, O. A. Ryder, and S. J. O'Brien, "Molecular phylogenetics and the origins of placental mammals," *Nature*, vol. 409, pp. 614–618, 2001.
- [27] H. Philippe, N. Lartillot, and H. Brinkmann, "Multigene analyses of bilaterian animals corroborate the monophyly of Ecdysozoa, Lophotrochozoa, and Protostomia," *Molecular Biology and Evolution*, vol. 22, pp. 1246–1253, 2005.
- [28] J. Bull, J. Huelsenbeck, C. Cunningham, D. Swofford, and P. Waddell, "Partitioning and combining data in phylogenetic analysis," *Systematic Biology*, vol. 42, no. 3, pp. 384–397, 1993.
- [29] C. H. Kuo, J. P. Wares, and J. C. Kissinger, "The Apicomplexan whole-genome phylogeny: an analysis of incongruence among gene trees," *Molecular Biology and Evolution*, vol. 25, no. 12, pp. 2689–2698, 2008.
- [30] J. H. Degnan and N. A. Rosenberg, "Discordance of species trees with their most likely gene trees," *PLoS Genetics*, vol. 2, pp. 762–768, 2006.
- [31] L. S. Kubatko and J. H. Degnan, "Inconsistency of phylogenetic estimates from concatenated data under coalescence," *Systematic Biology*, vol. 56, no. 1, pp. 17–24, 2007.

- [32] S. V. Edwards, L. Liu, and D. K. Pearl, “High-resolution species trees without concatenation,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, pp. 5936–5941, 2007.
- [33] L. Liu and D. K. Pearl, “Species trees from gene trees: reconstructing Bayesian posterior distributions of a species phylogeny using estimated gene tree distributions,” *Systematic Biology*, vol. 56, no. 3, pp. 504–514, 2007.
- [34] W. P. Maddison and L. L. Knowles, “Inferring phylogeny despite incomplete lineage sorting,” *Systematic Biology*, vol. 55, no. 1, pp. 21–30, 2006.
- [35] C. V. Than, D. Ruths, and L. K. Nakhleh, “PhyloNet: a software package for analyzing and reconstructing reticulate evolutionary relationships,” *BMC Bioinformatics*, vol. 9, no. 1, p. 322, 2008.
- [36] C. Semple and M. Steel, *Phylogenetics*. Oxford Lecture Series in Mathematics and its Applications 24, Oxford University Press, 2003.
- [37] J. Felsenstein, “The newick tree format,” 1986. <http://evolution.genetics.washington.edu/phylip/newicktree.html>.
- [38] P. Buneman, “The recovery of trees from measures of dissimilarity,” in *Mathematics in the Archaeological and Historical Sciences*, pp. 387–395, 1971.
- [39] D. R. Robinson and L. R. Foulds, “Comparison of phylogenetic trees,” *Mathematical Biosciences*, vol. 53, no. Mathematical Biosciences, pp. 131–147, 1981.
- [40] D. F. Robinson, “Comparison of labeled trees with valency three,” *Journal of Combinatorial Theory*, vol. 11, pp. 105–119, 1971.

- [41] B. Allen and M. Steel, “Subtree transfer operations and their induced metrics on evolutionary trees,” *Annals of Combinatorics*, vol. 5, pp. 1–13, 2001.
- [42] G. Hickey, F. Dehne, A. Rau-Chaplin, and C. Blouin, “SPR distance computation for unrooted trees,” *Evolutionary Bioinformatics Online*, vol. 4, pp. 17–27, 2008.
- [43] M. Bordewich and C. Semple, “On the computational complexity of the rooted subtree prune and regraft distance,” *Annals of Combinatorics*, pp. 1–15, 2005.
- [44] M. Hallett and J. Lagergren, “Efficient algorithms for lateral gene transfer problems,” in *Proceedings of the 5th Annual International Conference on Research in Computational Molecular Biology*, (New York), pp. 149–156, ACM Press, 2001.
- [45] R. Beiko and N. Hamilton, “Phylogenetic identification of lateral genetic transfer events,” *BMC Evolutionary Biology*, vol. 6, 2006.
- [46] D. MacLeod, R. Charlebois, F. Doolittle, and E. Baptiste, “Deduction of probable events of lateral gene transfer through comparison of phylogenetic trees by recursive consolidation and rearrangement,” *BMC Evolutionary Biology*, vol. 5, 2005.
- [47] L. K. Nakhleh, D. Ruths, and L. S. Wang, “RIATA-HGT: a fast and accurate heuristic for reconstructing horizontal gene transfer,” in *Proceedings of the Eleventh International Computing and Combinatorics Conference (COCOON 05)* (L. Wang, ed.), pp. 84–93, 2005. LNCS #3595.
- [48] C. V. Than and L. K. Nakhleh, “SPR-based tree reconciliation: Non-binary

- trees and multiple solutions,” in *Proceedings of the Sixth Asia Pacific Bioinformatics Conference (APBC)*, pp. 251–260, 2008.
- [49] Y. Wu, “A practical method for exact computation of subtree prune and regraft distance,” *Bioinformatics*, vol. 25, no. 2, pp. 190–196, 2009.
- [50] J. C. Avise, J. F. Shapiro, S. W. Daniel, C. F. Aquadro, and R. A. Lansman, “Mitochondrial DNA differentiation during the speciation process in *Peromyscus*,” *Molecular Biology and Evolution*, vol. 1, pp. 38–56, 1983.
- [51] F. Tajima, “Evolutionary relationship of DNA sequences in finite populations,” *Genetics*, vol. 105, pp. 437–460, 1983.
- [52] N. Takahata and M. Nei, “Gene genealogy and variance of interpopulation nucleotide differences,” *Genetics*, vol. 110, pp. 325–344, 1985.
- [53] J. F. C. Kingman, “The coalescent,” *Stochast. Proc. Appl.*, vol. 13, pp. 235–248, 1982.
- [54] C. V. Than, D. Ruths, H. Innan, and L. K. Nakhleh, “Confounding factors in HGT detection: statistical error, coalescent effects, and multiple solutions,” *Journal of Computational Biology*, vol. 14, no. 4, pp. 517–535, 2007.
- [55] J. Degnan and L. Salter, “Gene tree distributions under the coalescent process,” *Evolution*, vol. 59, pp. 24–37, 2005.
- [56] J. Hein, M. H. Schierup, and C. Wiuf, *Gene genealogies, variation and evolution: A primer in coalescent theory*. Oxford University Press, 2005.
- [57] J. Wakeley, *Coalescent Theory: An Introduction*. Roberts & Company Publishers, 2009.

- [58] J. Mallet, “Hybridization as an invasion of the genome,” *Trends in Ecology and Evolution*, vol. 20, no. 5, pp. 229–237, 2005.
- [59] M. L. Arnold, “Natural hybridization as an evolutionary process,” *Annual Review of Ecology and Systematics*, vol. 23, pp. 237–261, 1992.
- [60] F. de la Cruz and J. Davies, “Horizontal gene transfer and the origin of species: lessons from bacteria,” *Trends Microbiology*, vol. 8, pp. 128–133, 2000.
- [61] P. Planet, “Reexamining microbial evolution through the lens of horizontal transfer,” in *Molecular Systematics and Evolution: Theory and Practice* (R. DeSalle, G. Giribet, and W. Wheeler, eds.), pp. 247–270, Birkhauser Verlag, 2002.
- [62] P. Pamilo and M. Nei, “Relationship between gene trees and species trees,” *Molecular Biology and Evolution*, vol. 5, pp. 568–583, 1998.
- [63] E. Mossel and S. Roch, “Incomplete lineage sorting: consistent phylogeny estimation from multiple loci,” *IEEE Transactions on Computational Biology and Bioinformatics*, 2009.
- [64] J. P. Huelsenbeck and F. Ronquist, “MrBayes: Bayesian inference of phylogenetic trees,” *Bioinformatics*, vol. 17, no. 8, pp. 754–755, 2001.
- [65] V. Makarenkov, “T-REX: reconstructing and visualizing phylogenetic trees and reticulation networks,” *Bioinformatics*, vol. 17, no. 7, pp. 664–668, 2001.
- [66] A. Boc and V. Makarenkov, “New efficient algorithm for detection of horizontal gene transfer events,” *Algorithms in Bioinformatics: Third International Workshop*, Jan 2003.

- [67] M. Steel and T. Warnow, “Kaikoura tree theorems: computing the maximum agreement subtree,” *Information Processing Letters*, vol. 48, pp. 77–82, 1993.
- [68] C. V. Than, “Reconstruction of phylogenetic networks and their relationships with trees and branches,” Master’s thesis, Rice University, 2008.
- [69] C. V. Than, R. Sugino, H. Innan, and L. K. Nakhleh, “Efficient inference of bacterial strain trees from genome-scale multi-locus data,” *Bioinformatics*, vol. 24, pp. i123–i131, 2008. Proceedings of the 16th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB ‘08).
- [70] C. Bron and J. Kerbosch, “Finding all cliques of an undirected graph,” *Communication of ACM*, vol. 16, no. 9, pp. 575–577, 1973.
- [71] I. Koch, “Enumerating all connected maximal common subgraphs in two graphs,” *Theoretical Computer Science*, vol. 250, no. 1-2, pp. 1–30, 2001.
- [72] E. Tomita, A. Tanaka, and H. Takahashi, “The worst-case time complexity for generating all maximal cliques and computational experiments,” *Theoretical Computer Science*, vol. 363, no. 1, pp. 28 – 42, 2006.
- [73] S. Tsukiyama, M. Ide, H. Ariyoshi, and I. Shirakawa, “A new algorithm for generating all the maximal independent sets,” *Siam Journal on Computing*, vol. 6, pp. 505–517, 1977.
- [74] M. Kuroda, T. Ohta, I. Uchiyama, T. Baba, H. Yuzawa, I. Kobayashi, L. Cui, A. Oguchi, K. Aoki, Y. Nagai, J. Lian, T. Ito, M. Kanamori, H. Matsumaru, A. Maruyama, H. Murakami, A. Hosoyama, Y. Mizutani-Ui, N. Takahashi, T. Sawano, R. Inoue, C. Kaito, K. Sekimizu, H. Hirakawa, S. Kuhara,

- S. Goto, J. Yabuzaki, M. Kanehisa, A. Yamashita, K. Oshima, K. Furuya, C. Yoshino, T. Shiba, M. Hattori, N. Ogasawara, H. Hayashi, and K. Hiramatsu, "Whole genome sequencing of methicillin-resistant *Staphylococcus aureus*," *Lancet*, vol. 357, no. 9264, pp. 1225–40, 2001.
- [75] T. Ohta, H. Hirakawa, K. Morikawa, A. Maruyama, Y. Inose, A. Yamashita, K. Oshima, M. Kuroda, M. Hattori, K. Hiramatsu, S. Kuhara, and H. Hayashi, "Nucleotide substitutions in *Staphylococcus aureus* strains, Mu50, Mu3, and N315.," *DNA Research*, vol. 11, no. 1, pp. 51–6, 2004.
- [76] S. Gill, D. Fouts, G. Archer, E. Mongodin, R. Deboy, J. Ravel, I. Paulsen, J. Kolonay, L. Brinkac, M. Beanan, R. Dodson, S. Daugherty, R. Madupu, S. Angiuoli, A. Durkin, D. Haft, J. Vamathevan, H. Khouri, T. Utterback, C. Lee, G. Dimitrov, L. Jiang, H. Qin, J. Weidman, K. Tran, K. Kang, I. Hance, K. Nelson, and C. Fraser, "Insights on evolution of virulence and resistance from the complete genome analysis of an early methicillin-resistant *Staphylococcus aureus* strain and a biofilm-producing methicillin-resistant *Staphylococcus epidermidis* strain.," *Journal of Bacteriology*, vol. 187, no. 7, pp. 2426–38, 2005.
- [77] M. Holden, E. Feil, J. Lindsay, S. Peacock, N. Day, M. Enright, T. Foster, C. Moore, L. Hurst, R. Atkin, A. Barron, N. Bason, S. Bentley, C. Chillingworth, T. Chillingworth, C. Churcher, L. Clark, C. Corton, A. Cronin, J. Doggett, L. Dowd, T. Feltwell, Z. Hance, B. Harris, H. Hauser, S. Holroyd, K. Jagels, K. James, N. Lennard, A. Line, R. Mayes, S. Moule, K. Mungall, D. Ormond, M. Quail, E. Rabbino-witsch, K. Rutherford, M. Sanders, S. Sharp, M. Simmonds, K. Stevens, S. Whitehead, B. Barrell, B. Spratt, and J. Parkhill, "Complete genomes of two clinical *Staphylococcus aureus* strains: evidence for



- the rapid evolution of virulence and drug resistance.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 26, pp. 9786–91, 2004.
- [78] T. Baba, F. Takeuchi, M. Kuroda, H. Yuzawa, K. Aoki, A. Oguchi, Y. Nagai, N. Iwama, K. Asano, T. Naimi, H. Kuroda, L. Cui, K. Yamamoto, and K. Hiramatsu, "Genome and virulence determinants of high virulence community-acquired MRSA.," *Lancet*, vol. 359, no. 9320, pp. 1819–27, 2002.
- [79] L. Herron-Olson, J. Fitzgerald, J. Musser, and V. Kapur, "Molecular correlates of host specialization in staphylococcus aureus.," *PLoS One*, vol. 2, no. 10, p. e1120, 2007.
- [80] B. Diep, S. Gill, R. Chang, T. Phan, J. Chen, M. Davidson, F. Lin, J. Lin, H. Carleton, E. Mongodin, G. Sensabaugh, and F. Perdreau-Remington, "Complete genome sequence of USA300, an epidemic clone of community-acquired methicillin-resistant *Staphylococcus aureus*.," *Lancet*, vol. 367, no. 9512, pp. 731–9, 2006.
- [81] S. Altschul, T. Madden, A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. Lipman, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.," *Nucleic Acids Research*, vol. 25, no. 17, pp. 3389–402, 1997.
- [82] D. L. Swofford, "PAUP\*: Phylogenetic analysis using parsimony (\* and others methods). version 4. sinauer associates, sunderland, massachusetts.," 2003.
- [83] M. Nei and S. Kumar, *Molecular evolution and phylogenetics*. Oxford: Oxford University Press, 2000.

- [84] H. Ochman and A. Wilson, “Evolution in bacteria: evidence for a universal substitution rate in cellular genomes,” *Journal of Molecular Evolution*, vol. 26, no. 1-2, pp. 74–86, 1987.
- [85] A. Retchless and J. Lawrence, “Temporal fragmentation of speciation in bacteria,” *Science*, vol. 317, pp. 1093–1096, 2007.
- [86] N. Saitou and M. Nei, “The neighbor-joining method: a new method for reconstructing phylogenetic trees,” *Molecular Biology and Evolution*, vol. 4, pp. 406–425, 1987.
- [87] W. P. Maddison and D. R. Maddison, “Mesquite: a modular system for evolutionary analysis. Version 1.01. <http://mesquiteproject.org>,” 2004.
- [88] C. V. Than and L. K. Nakhleh, “Species tree inference by minimizing deep coalescences,” *PLoS Computational Biology*, vol. 5, no. 9, p. e1000501, 2009.
- [89] B. Schieber and U. Vishkin, “On finding lowest common ancestors: simplification and parallelization,” *Siam Journal on Computing*, vol. 17, no. 6, pp. 1253–1262, 1988.
- [90] O. Berkman and U. Vishkin, “Recursive star-tree parallel data structure,” *Siam Journal on Computing*, vol. 22, no. 2, pp. 221–242, 1993.
- [91] G. L. Nemhauser and L. E. Trotter, Jr., “Vertex packings: structural properties and algorithms,” *Mathematical Programming*, vol. 8, pp. 232–248, 1975.
- [92] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. The MIT Press, 2 ed., 2001.

- [93] N. D. Levine, "Taxonomy and review of the coccidian genus *Cryptosporidium* (Protozoa, Apicomplexa)," *Journal of Protozool*, vol. 31, pp. 94–98, 1984.
- [94] J. Mallet, "Hybrid speciation," *Nature*, vol. 446, pp. 279–283, 2007.
- [95] L. K. Nakhleh, "Evolutionary phylogenetic networks: models and issues," in *The Problem Solving Handbook for Computational Biology and Bioinformatics* (L. Heath and N. Ramakrishnan, eds.), Springer, 2009.
- [96] J. Syring, A. Willyard, R. Cronn, and A. Liston, "Evolutionary relationships among *Pinus* (*Pinaceae*) subsections inferred from multiple low-copy nuclear loci," *American Journal of Botany*, vol. 92, pp. 2086–2100, 2005.
- [97] D. A. Pollard, V. N. Iyer, A. M. Moses, and M. B. Eisen, "Widespread discordance of gene trees with species tree in *Drosophila*: evidence for incomplete lineage sorting," *PLoS Genetics*, vol. 2, no. 10, p. e173, 2006.
- [98] C. R. Linder and L. H. Rieseberg, "Reconstructing patterns of reticulate evolution in plants," *American Journal of Botany*, vol. 91, pp. 1700–1708, 2004.
- [99] C. V. Than and L. K. Nakhleh, "Coalescent histories on phylogenetic networks and detection of hybridization despite lineage sorting," Under review, 2009.
- [100] C. Meng and L. S. Kubatko, "Detecting hybrid speciation in the presence of incomplete lineage sorting using gene tree incongruence: A model," *Theoretical Population Biology*, vol. 75, no. 1, pp. 35–45, 2009.
- [101] S. Joly, P. A. McLenachan, and P. J. Lockhart, "A statistical approach for distinguishing hybridization and incomplete lineage sorting," *The American Naturalist*, vol. 174, no. 2, pp. E54–E70, 2009.

- [102] G. Jin, L. K. Nakhleh, S. Snir, and T. Tuller, “Inferring phylogenetic networks by the maximum parsimony criterion: a case study,” *Molecular Biology and Evolution*, vol. 24, no. 1, pp. 324–337, 2007.
- [103] S. Gonzalez, L. Gallo, M. Climent, E. Barrio, and A. Querol, “Ecological characterization of natural hybrids from *Saccharomyces cerevisiae* and *S. kudriavzevii*,” *International Journal of Food Microbiology*, vol. 116, pp. 11–18, 2007.
- [104] S. Gonzalez, E. Barrio, and A. Querol, “Molecular characterization of new natural hybrids of *Saccharomyces cerevisiae* and *S. kudriavzevii* in brewing,” *Applied Environmental Microbiology*, vol. 74, pp. 2314–2320, 2008.
- [105] B. Dunn and G. Sherlock, “Reconstruction of the genome origins and evolution of the hybrid lager yeast *saccharomyces pastorianus*,” *Genome Research*, vol. 18, pp. 1610–1623, 2008.
- [106] M. M. Morin and B. M. E. Moret, “Netgen: generating phylogenetic networks with diploid hybrids,” *Bioinformatics*, vol. 22, no. 15, pp. 1921–1923, 2006.
- [107] D. H. Huson, “SplitsTree: a program for analyzing and visualizing evolutionary data,” *Bioinformatics*, vol. 14, no. 1, pp. 68–73, 1998.
- [108] D. Bryant and V. Moulton, “NeighborNet: an agglomerative method for the construction of planar phylogenetic networks,” *Molecular Biology and Evolution*, vol. 21, no. 2, pp. 255–265, 2004.
- [109] B. Moret, L. Nakhleh, T. Warnow, C. Linder, A. Tholse, A. Padolina, J. Sun, and R. Timme, “Phylogenetic networks: modeling, reconstructibility, and accuracy,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 1, no. 1, pp. 13–23, 2004.

- [110] J. Felsenstein, *Inferring Phylogenies*. Sunderland, MA: Sinauer Associates, Inc., 2003.
- [111] L. K. Nakhleh, J. Sun, T. Warnow, R. Linder, B. Moret, and A. Tholse, "Towards the development of computational tools for evaluating phylogenetic network reconstruction methods," in *Proceedings of the 8th Pacific Symposium on Biocomputing*, pp. 315–326, World Scientific Pub., 2003.
- [112] L. Nakhleh, T. Warnow, and C. Linder, "Reconstructing reticulate evolution in species—theory and practice," in *Proceedings of the 8th Annual International Conference on Research in Computational Molecular Biology*, pp. 337–346, 2004.
- [113] D. Penny and M. D. Hendy, "The use of tree comparison metrics," *Systematic Zoology*, vol. 34, no. 1, pp. 75–82, 1985.
- [114] I. Kanj, L. K. Nakhleh, C. V. Than, and G. Xia, "Seeing the trees and their branches in the network in hard," *Theoretical Computer Science*, vol. 401, no. 153–164, 2008.
- [115] L. Nakhleh, G. Jin, F. Zhao, and J. Mellor-Crummey, "Reconstructing phylogenetic networks using maximum parsimony," in *Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conference (CSB2005)*, pp. 93–102, 2005.
- [116] D. Ruths and L. Nakhleh, "Techniques for assessing phylogenetic branch support: A performance study," in *Proceedings of the 4th Asia Pacific Bioinformatics Conference*, pp. 187–196, 2006.