

RICE UNIVERSITY

**From Gene Trees to Species Trees:
Algorithms for Parsimonious Reconciliation**

by

Yun Yu

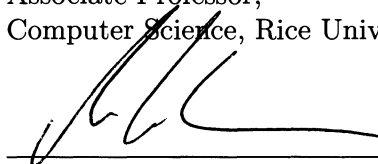
A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Master of Science

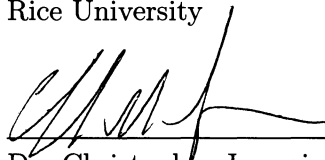
APPROVED, THESIS COMMITTEE:



Dr. Luay K. Nakhleh (Chair),
Associate Professor,
Computer Science, Rice University



Dr. Michael H. Kohn,
Associate Professor,
Ecology and Evolutionary Biology,
Rice University



Dr. Christopher Jermaine,
Associate Professor,
Computer Science, Rice University

HOUSTON, TEXAS
MARCH, 2012

Abstract

From Gene Trees to Species Trees:

Algorithms for Parsimonious Reconciliation

by

Yun Yu

One of the criteria for inferring a species tree from a collection of gene trees, when gene tree incongruence is assumed to be due to incomplete lineage sorting (ILS), is *minimize deep coalescence*, or MDC. Exact algorithms for inferring the species tree from rooted, binary trees under MDC were recently introduced. Nevertheless, in phylogenetic analyses of biological data sets, estimated gene trees may differ from true gene trees, be incompletely resolved, and not necessarily rooted. Further, the MDC criterion considers only the topologies of the gene trees. So the contributions of my work are three-fold:

1. We propose new MDC formulations for the cases in which the gene trees are unrooted/binary, rooted/non-binary, and unrooted/non-binary, prove structural theorems that allow me to extend the algorithms for the rooted/binary gene tree case to these cases in a straightforward manner.
2. We propose an algorithm for inferring a species tree from a collection of gene trees with coalescence times that takes into account not only the topology of the gene trees but also the coalescence times.

3. We devise MDC-based algorithms for cases in which multiple alleles per species may be sampled.

We have implemented all of the algorithms in the PhyloNet software package and studied their performance in coalescent-based simulation studies in comparison with other methods including democratic vote, greedy consensus, STEM, and GLASS.

Acknowledgments

This thesis would not have been possible without the help and support of many people, only some of whom it is possible to mention here.

First and foremost, I would like to thank my advisor, Dr. Luay Nakhleh. He is always very patient and encouraging and has excellent advice and unsurpassed knowledge of Phylogenetics. Dr. Nakhleh has been my inspiration whenever I have met with difficulties.

I would like to thank Dr. Tandy Warnow. Some of the work in this thesis has been in collaboration with her. I appreciate her insight and helpfulness.

I would like to thank my committee members Dr. Michael Kohn and Dr. Christopher Jermaine for their time and patience.

I also want to thank our group members, Angela Zhu, Cuong Than, Justin Park, Kevin Liu, Matt Barnett, Natalie Yudin, Troy Ruths, and Wanding Zhou for their friendship, encouragement, and the insights.

Last but not least, I would like to thank my husband, Chang Li, my son, Kevin Li, and my parents, Wende Yu and Xianping Zhou. They have been very supportive of my work, and I could not have finished this thesis without them.

Contents

Abstract	ii
Acknowledgments	iv
List of Figures	ix
1 Introduction	1
1.1 Contributions of this thesis	3
2 Background	5
2.1 Species tree/gene tree problem	5
2.2 Incomplete lineage sorting	6
2.3 Existing methods	7
3 Preliminary Material	9
3.1 Clades and clusters	9
3.2 Valid coalescent histories and extra lineages	11
3.3 MDC on rooted binary gene trees: The single-allele case	16
4 Extending MDC	21
4.1 Estimated gene trees: The single-allele case	21

4.1.1	Unrooted, binary gene trees	21
4.1.2	Rooted, non-binary gene trees	25
4.1.3	Unrooted, non-binary gene trees	28
4.2	Incorporating coalescence times	32
4.3	Multiple-allele cases	36
5	Performance	41
5.1	Simulated data	41
5.2	Methods	42
5.3	Results and discussion	45
6	PhyloNet	60
7	Conclusions	65

List of Figures

2.1	Gene/species tree incongruence due to ILS. Given species tree ST , with constant population size throughout and time t in coalescent units (number of generations divided by the population size) between the two divergence events, each of the three gene tree topologies gt_1 , gt_2 , and gt_3 may be observed, with probabilities $1 - (2/3)e^{-t}$, $(1/3)e^{-t}$, and $(1/3)e^{-t}$, respectively.	7
3.1	Under the coalescent model, time flows backward from the leaves toward the root. In a valid coalescent history, the number of lineages in edge $e = (u, v)$ equals the sum of the numbers of lineages entering e , when going backward in time, from edges (v, x) and (v, y) , minus the number of coalescent events that occur at node v .	12
3.2	Illustration of optimal and non-optimal reconciliations of a rooted, binary gene tree gt with a rooted, binary species tree ST , which yield 1, 2, and 3 extra lineages, respectively.	13
4.1	Illustration of optimal and non-optimal reconciliations of an unrooted, binary gene tree gt with a rooted, binary species tree ST , which yield 1 and 3 extra lineages, respectively.	22

4.2	Illustration of optimal and non-optimal reconciliations of a rooted, non-binary gene tree gt with a rooted, binary species tree ST , which yield 0 and 3 extra lineages, respectively.	26
4.3	Illustration of optimal and non-optimal reconciliations of an unrooted, non-binary gene tree gt with a rooted, binary species tree ST , which yield 0 and 3 extra lineages, respectively.	28
4.4	Illustration of constraints on speciation times of nodes in the species tree imposed by gene trees. Particularly, we have two gene trees gt_1 and gt_2 with branch lengths on the same taxa set.	35
4.5	Illustration of ILS and the MDC criterion in the case of multiple alleles. X_i denotes an allele of species X . In particular, species D has no alleles sampled for the given locus.	37
5.1	Results of the democratic vote (DV) method on the true gene trees of the 1Ne (left) and 10Ne (right) data sets (see text for description of the four curves).	47
5.2	Performance of GLASS on the 1Ne (left column) and 10Ne (right column) data sets. Rows from top to bottom correspond to 1, 3, 9, and 27 alleles, respectively. Distances were computed under the Jukes-Cantor model, and for each locus, the bottom $x\%$ distances were removed, for $x \in \{0, 5, 10, 20, 30, 40\}$	54

5.3	Results of the exact and heuristic solutions of MDC on 1Ne (left column) and 10Ne (right column) data sets, using the true gene trees. Rows from top to bottom correspond to 1, 3, 9, and 27 alleles, respectively.	55
5.4	Results of the exact and heuristic solutions of MDC on 1Ne (left column) and 10Ne (right column) data sets, using the gene tree reconstructed by MP. Rows from top to bottom correspond to 1, 3, 9, and 27 alleles, respectively.	56
5.5	Performance of all methods on 1Ne (left column) and 10Ne (right column) data sets, using the true gene trees. Rows from top to bottom correspond to 1, 3, 9, and 27 alleles, respectively.	57
5.6	Performance of all methods on 1Ne (left column) and 10Ne (right column) data sets, using the gene trees reconstructed by MP. Rows from top to bottom correspond to 1, 3, 9, and 27 alleles, respectively. . . .	58
5.7	Running time of methods on 1Ne (left column) and 10Ne (right column) data sets, using the true gene trees. Rows from top to bottom correspond to 1, 3, 9, and 27 alleles, respectively.	59

Chapter 1

Introduction

Use of DNA sequence data for inferring phylogenetic relationships among species is central in biology; however, the evolution of DNA is complex, and this complexity is almost never fully taken into account in phylogenetic inference from DNA sequence data. Of particular importance in phylogenomic analyses, the evolution of the DNA regions used for phylogenetic inference (often genes, but not always) need not be congruent with the evolution of the species. This is the classic gene tree/species tree problem [Mad97, DR09b]. Errors in estimating gene trees (due, for example, to inadequate sequence length, improper phylogeny estimation methods, or insufficient computational resources) can produce estimated gene trees that differ from the true gene trees and therefore from the species tree even when the true gene tree and species tree are identical. In addition, however, many biological factors can cause true gene trees to be different from the true species trees. For example, horizontal gene transfer, gene duplication/loss, and incomplete lineage sorting (ILS) can result in gene/species tree incongruence. My work here focuses on ILS as the sole biological cause of such

incongruence.

The population genetics community has long recognized the existence of gene tree incongruence [Hud83, Nei86, Taj83]. This phenomenon can sometimes cause severe errors in phylogenetic inference procedures [PN88, Tak89, Wu91]. However, for most species, it has until recently been difficult to gather data on more than a single part of the genome [DAB⁺04]. With the advent of technologies that make it possible to obtain large amounts of sequence data from multiple species, multi-locus data are becoming widely available, highlighting the issue of gene tree incongruence [DR09a, KWK08, PIME06, RWKC03, SWCL05, TSIN08].

Several methods have been introduced for inferring a species tree from a collection of gene trees under ILS-based incongruence. Summary statistics, such as the majority-rule consensus (e.g., [DDBR09, KWK08]) and democratic vote (e.g., [Daw04, DR06, Wu91, Wu92]), are fast to compute and provide a good estimate of the species tree in many cases. However, not only does the accuracy of these methods suffer under certain conditions, but also these methods do not provide explicit reconciliation scenarios, generating only summaries of the gene trees. Methods that explicitly model ILS have recently been introduced, such as Bayesian inference [ELP07, LP07], maximum likelihood [KCK09], and the maximum parsimony criterion *Minimize Deep Coalescence*, or MDC [Mad97, MK06, TSIN08]. Recently Than and Nakhleh [TN09, TN10] introduced the first exact algorithms for inferring species trees under the MDC criterion from a collection of rooted, binary gene trees.

1.1 Contributions of this thesis

In practice, when we analyze real biological data sets, we have DNA sequences from which we can infer gene trees. However, none of those tree reconstruction methods can identify the root. So we have to deal with unrooted gene trees. Furthermore, there is uncertainty about gene trees inferred from sequences. In Bayesian inference, this uncertainty is reflected by a posterior distribution of gene tree topologies. In a parsimony analysis, several equally optimal trees are computed. To account for these uncertainty so as to minimize the number of false-positive edges, we usually remove edges with low support. For example, for maximum parsimony, we take strict consensus of all optimal trees; for neighbor-joining with bootstrap, we remove edges with low bootstrap value; for bayesian analysis, we remove edges with low posterior probability. All of them may result in unresolved trees.

Here we propose an approach to estimating species trees from estimated gene trees which outcomes these problems. Instead of assuming that all gene trees are correct (and hence fully resolved, rooted trees), I consider the case in which all gene trees are modified so that they are reasonably likely to be unrooted, edge-contracted versions of the true gene trees. For example, the reliable edges in the gene trees can be identified using statistical techniques, such as bootstrapping, and all low-support edges can be contracted. In this way, the MDC problem becomes one in which the input is a set of gene trees that might not be rooted and might not be fully resolved, and the objective is a rooted, binary species tree and binary rooted refinements of the input gene trees that optimizes the MDC criterion. We provide exact algorithms and heuristics for

inferring species trees for these cases. Further, the MDC criterion considers only the topologies of the gene trees. In practice, coalescence times may be estimated for the internal nodes of gene trees. We also extend the MDC criterion and provide an exact algorithm for the case of input gene trees with coalescence times. In addition, Than and Nakhleh have extended the MDC criterion and devised an algorithm for cases in which multiple alleles (or no alleles) per species may be sampled. Here, we establish that the algorithm is exact, in that it finds a species tree that minimizes the amount of deep coalescences when multiple alleles may be involved in the analysis.

We have implemented these MDC-based algorithms in PhyloNet software package [TRN08], and evaluated their performance in comparison with four other popular methods: the greedy consensus, the democratic vote, GLASS, and STEM, in terms of both the accuracy and speed.

Chapter 2

Background

2.1 Species tree/gene tree problem

A species tree reveals the inferred evolutionary relationships among different biological species, in which every internal node represents species divergence. Within the branches of a species tree, gene trees are contained. They are formed because of gene replication. Every internal node in a gene tree is generated when a gene copy at a locus in the genome replicates and its copies are present in more than one offspring [Mad97].

The traditional approach to species tree inference entails sequencing a gene from the set of species under study, inferring the gene tree, and finally declaring the gene tree as the estimate of the species phylogenetic relationship. However, in a seminal paper, Maddison [Mad97] discussed the issue of potential incongruence among gene trees and proposed inferring species phylogenies by simultaneously accounting for mutations within a gene and incongruence across genes. He pointed out that gene

trees could disagree with their containing species tree due to a host of factors, like horizontal transfer, gene duplication/extinction, and lineage sorting. While Maddison focused mainly on topological incongruence, Edwards [Edw09] more recently revisited the issues and highlighted, in addition, incongruence in terms of branch lengths (what he termed *branch length heterogeneity*). Indeed, recent analysis of multi-locus data sets have shown large extents of gene tree incongruence in various groups of organisms [RWKC03, SWCL05, PIME06, TSIN08, KWK08]. Here, my work exclusively assumes incomplete lineage sorting (ILS) as the cause of incongruence.

2.2 Incomplete lineage sorting

ILS is best understood under the *coalescent model* [DR06, DS05, Hud83, Nei86, Nei87, Ros02, Taj83, Tak89]. The coalescent model views gene lineages moving backward in time, eventually coalescing down to one lineage. The term *coalescence* refers to the process in which, looking backward in time, two gene lineages merge at a common ancestor. In each time interval between species divergences (e.g., t in Fig. 2.1), lineages entering the interval from a more recent time period might or might not coalesce—an event whose probability is determined largely by the population size and branch lengths. ILS, or *deep coalescence*, refers to the case in which two lineages fail to coalesce before their speciation events. It is more likely to happen for a larger population or a shorter branch length.

Thus, a gene tree is viewed as a random variable conditional on a species tree. For the species tree $((A, B), C)$, with time t between species divergences, Fig. 2.1 shows

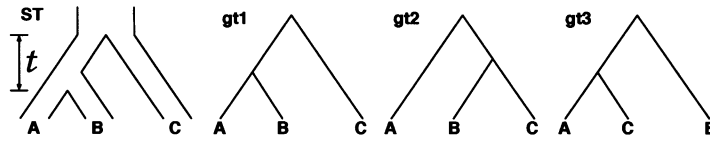


Figure 2.1: Gene/species tree incongruence due to ILS. Given species tree ST , with constant population size throughout and time t in coalescent units (number of generations divided by the population size) between the two divergence events, each of the three gene tree topologies gt_1 , gt_2 , and gt_3 may be observed, with probabilities $1 - (2/3)e^{-t}$, $(1/3)e^{-t}$, and $(1/3)e^{-t}$, respectively.

the three possible outcomes for the gene tree topology random variable, along with their probabilities.

2.3 Existing methods

The observation of gene tree incongruence in data analyses and the need to establish the phylogenetic relationship of species despite this incongruence have led to the development of a number of methods for inferring species trees despite incomplete lineage sorting (see [DR09b] and [LYK⁺09] for very recent surveys of such methods). These methods can be divided into two categories. The first category contains methods that simply concatenate sequences from the multiple loci and apply any phylogenetic tree reconstruction method to the resulting supergene. While this approach is very simple and fast, particularly when choosing a fast phylogenetic method to run on the concatenated sequences, Kubatko and Degnan [KD07] have recently shown that this approach can produce incorrect species tree estimates with strong bootstrap support.

The second category contains methods that analyze each locus individually and produce a species tree estimate from the evolutionary histories of the individual loci. The simplest two methods in this category are the democratic vote, which amounts to selecting the highest-frequency gene tree as the species tree estimate, and the majority consensus, which produces a tree in which each branch is displayed by at least 50% of the individual gene trees. Degnan and Rosenberg [DR09b] recently discussed the performance of these two methods and showed that they can produce incorrect species tree estimates, even when more data are used. More recently, four methods were added to this category, which take incongruence into account by exclusively accounting for the coalescent process [Kin82]. Mossel and Roch [MR10] introduced GLASS (Global LAtest Split), a distance-based method that infers species relationships based on pairwise distances computed for each locus independently. Liu [LP07] and Edwards [ELP07] introduced BEST (Bayesian Estimation of Species Trees), a Bayesian method for simultaneous inference of gene trees and the species tree that contains them. Kubatko, Carstens, and Knowles [KCK09] introduced STEM (Species Tree Estimation using Maximum likelihood), a maximum likelihood approach following Maddison’s proposal [Mad97]. Than and Nakhleh [TN09] introduced MDC (Minimize Deep Coalescences), a parsimony-based approach to species tree inference that builds on previous studies [Mad97, MK06, TSIN08].

Chapter 3

Preliminary Material

3.1 Clades and clusters

Throughout this section, unless specified otherwise, all trees are presumed to be rooted binary trees, bijectively leaf-labelled by the elements of \mathcal{X} (that is, each $x \in \mathcal{X}$ labels one leaf in each tree). We denote by $\mathcal{T}_{\mathcal{X}}$ the set of all binary rooted trees on leaf-set \mathcal{X} . We denote by $V(T)$, $E(T)$, and $L(T)$ the node-set, edge-set, and leaf-set, respectively, of T . For v a node in T , we define $\text{parent}(v)$ to be the parent of v in T , and $\text{Children}(v)$ to be the children of v . A *clade* in a tree T is a rooted subtree of T , which can be identified by the node in T rooting the clade. For a given tree T , we denote the subtree of T rooted at v by $\text{Clade}_T(v)$, and when the tree T is understood, by $\text{Clade}(v)$. The clade for node v is $\text{Clade}(v)$, and since nodes can have children, the children of a clade $\text{Clade}(v)$ are the clades rooted at the children of v . The set of all clades of a tree T is denoted by $\text{Clades}(T)$. The set of leaves in $\text{Clade}_T(v)$ is called a *cluster* and denoted by $\text{Cluster}_T(v)$ (or more simply by $\text{Cluster}(v)$ if the tree

T is understood). The clusters that contain either all the taxa or just single leaves are called *trivial*, and the other clusters are called *non-trivial*. The cluster of node v is $Cluster(v)$. As with clades, clusters can also have children. If Y is a cluster in a tree T , then the clade for Y within T , denoted by $Clade_T(Y)$, is the clade of T induced by Y . The set of all clusters of T is denoted by $Clusters(T)$. We say that edge e in gt is outside cluster Y if it satisfies $e \notin E(Clade_{gt}(Y))$, and otherwise that it is inside Y . Given a set $A \subseteq L(T)$, we define $MRC A_T(A)$ to be the most recent (or least) common ancestor of the taxa in A . Finally, given trees t and T , both on \mathcal{X} , we define $H : V(t) \rightarrow V(T)$ by $H_T(v) = MRC A_T(Cluster_t(v))$.

We extend the definitions of $Clades(T)$ and $Clusters(T)$ to the case in which T is unrooted by defining $Clades(T)$ to be the set of all clades of all possible rootings of T , and $Clusters(T)$ to be the set of all clusters of all possible rootings of T . Thus, the sets $Clades(T)$ and $Clusters(T)$ depend upon whether T is rooted or not.

Given a cluster $Y \subseteq \mathcal{X}$ of T , the parent edge of Y within T is the edge incident with the root of the clade for Y , but which does not lie within the clade. When T is understood by context, we will refer to this as the parent edge of Y .

A set \mathcal{C} of clusters is said to be *compatible* if there is a rooted tree T on leaf-set S such that $Clusters(T) = \mathcal{C}$. By [SS03], the set \mathcal{C} is compatible if and only if every pair A and B of clusters in \mathcal{C} are either disjoint or one contains the other.

3.2 Valid coalescent histories and extra lineages

Given gene tree gt and species tree ST , a *valid coalescent history* is a function $f : V(gt) \rightarrow V(ST)$ such that the following conditions hold:

- if w is a leaf in gt , then $f(w)$ is the leaf in ST with the same label; and,
- if w is a vertex in $Clade_{gt}(v)$, then $f(w)$ is a vertex in $Clade_{ST}(f(v))$.

Note that these two conditions together imply that $f(v)$ is a node on the path between the root of ST and the MRCA in ST of $Cluster_{gt}(v)$. Given a gene tree gt and a species tree ST , and given a function f defining a valid coalescent history of gt within ST , the *number of lineages* on each edge in ST can be computed by inspection.

Notice that in a rooted tree, each edge (u, v) is uniquely associated with, or identified by, its head node, v . Further, when multiple coalescence events occur within a branch in the species tree, the order in which these events occur does not matter under the MDC criterion. Based on these two observations, we always map coalescence events to nodes. Let $e = (u, v)$ be an edge in the species tree, and x and y be the two children of v . If l_x lineages enter edge e from the edge (v, x) , and l_y lineages enter edge e from the edge (v, y) , and q coalescence events are mapped to node v under a given valid coalescent history, then the number of lineages in edge e is $l_x + l_y - q$; see the illustration in Fig. 3.1.

An optimal valid coalescent history is one that results in the minimum number of lineages over all valid coalescent histories. We denote the number of extra lineages on an edge $e \in E(ST)$ (one less than the number of lineages on e) in an optimal

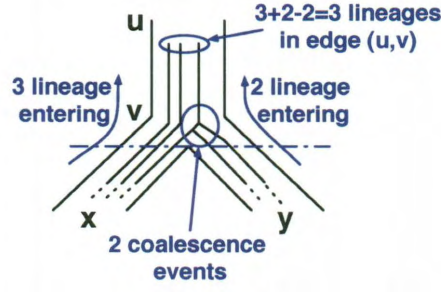


Figure 3.1: Under the coalescent model, time flows backward from the leaves toward the root. In a valid coalescent history, the number of lineages in edge $e = (u, v)$ equals the sum of the numbers of lineages entering e , when going backward in time, from edges (v, x) and (v, y) , minus the number of coalescent events that occur at node v .

valid coalescent history of gt within ST by $XL(e, gt)$, and we denote by $XL(ST, gt)$ the total number of extra lineages within an optimal valid coalescent history of gt within ST , i.e., $XL(ST, gt) = \sum_{e \in E(ST)} XL(e, gt)$; see Fig. 3.2. Finally, we denote by $XL(ST, \mathcal{G})$ the total number of extra lineages, or MDC score, over all gene trees in \mathcal{G} , so $XL(ST, \mathcal{G}) = \sum_{gt \in \mathcal{G}} XL(ST, gt)$.

For example, in Fig. 3.2, given a rooted binary gene tree gt and a rooted binary species tree ST , there are three different ways of reconciling this gene tree, as shown in the bottom row. We can see that with respect to the topology of the gene tree, lineage C and D have to coalesce above the root. However, A and B can coalesce on three different branches of the species tree, resulting in three different valid coalescent histories. From the number of extra lineages on each branch of the species tree for all three reconciliations shown in the figure, it is easy to tell that the first one is optimal under MDC criterion, which yields the smallest number of total extra lineages

$$XL(ST, gt) = 1.$$

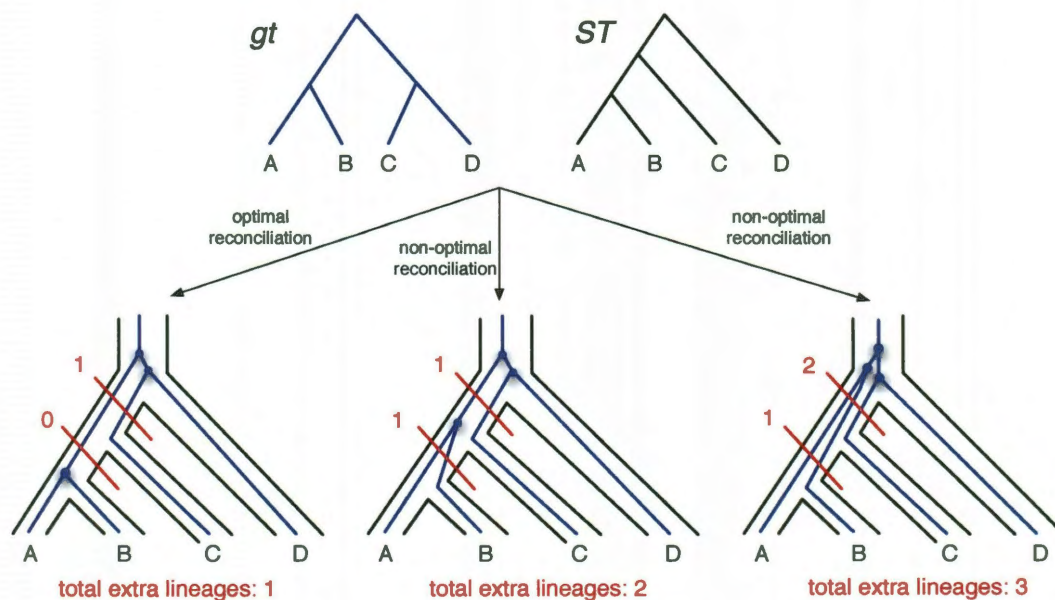


Figure 3.2: Illustration of optimal and non-optimal reconciliations of a rooted, binary gene tree gt with a rooted, binary species tree ST , which yield 1, 2, and 3 extra lineages, respectively.

Given gene tree gt and species tree ST , finding the valid coalescent history that yields the smallest number of extra lineages is achievable in polynomial time, as we now show. Given cluster A in gt and cluster B in ST , we say that A is B -maximal if (1) $A \subseteq B$ and (2) for all $A' \in Clusters(gt)$, if $A \subset A'$ then $A' \not\subseteq B$. We set $k_B(gt)$ to be the number of B -maximal clusters within gt . Finally, we say that cluster A is ST -maximal if there is a cluster $B \in Clusters(ST)$ such that $B \neq \mathcal{X}$ and A is B -maximal.

Theorem 1 (From [TN09]) Let gt be a gene tree, ST be a species tree, both binary

rooted trees on leaf-set X . Let B be a cluster in ST , and let e be the parent edge of B in ST . Then $k_B(gt)$ is equal to the number of lineages on e in an optimal valid coalescent history. Therefore, $XL(e, gt) = k_B(gt) - 1$ and $XL(ST, gt) = \sum_B [k_B(gt) - 1]$, where B ranges over the clusters of ST . Furthermore, a valid coalescent history f that achieves this total number of extra lineages can be produced by setting $f(v) = H_{ST}(v)$ (i.e., $f(v) = MRCA_{ST}(Cluster_{gt}(v))$) for all v .

In other words, we can score a candidate species tree ST with respect to a set \mathcal{G} of rooted binary trees with $XL(ST, \mathcal{G}) = \sum_{gt \in \mathcal{G}} \sum_{B \in Clusters(ST)} [k_B(gt) - 1]$. Finally,

Corollary 1 *Given collection \mathcal{G} of k gene trees and species tree ST , each tree labelled by the species in \mathcal{X} , we can compute the optimal coalescent histories relating each gene tree to ST so as to minimize the total number of extra lineages in $O(nk)$ time, and the MDC score of these optimal coalescent histories in $O(nk)$ time, where $|\mathcal{X}| = n$.*

The analysis of the running time follows from the following two lemmas.

Lemma 1 *Given a rooted gene tree gt and a rooted binary species tree ST , we can compute all $H_{ST}(v)$ (letting v range over $V(gt)$) in $O(n)$ time.*

Proof: We begin by preprocessing both gt and ST in $O(n)$ time so that each subsequent MRCA query takes constant time [HT84, BFC00]. Then, for each node $v \in V(gt)$, we can compute $H_{ST}(v)$ in $O(1)$ time. Thus, we can compute the optimal coalescent history relating gt to ST in $O(n)$ time. \square

Lemma 2 *Given a rooted gene tree gt and a rooted binary species tree ST , and assuming that $H_{ST}(v)$, for each $v \in V(gt)$ has been computed, we can compute $XL(ST, gt)$ in $O(n)$ time.*

Proof: We define the function $lins : V(ST) \rightarrow \mathbb{N}$ as follows. For a vertex $v \in V(ST)$, $lins(v)$ is the number of lineages in the parent edge of node v given an optimal valid coalescent history (i.e., under the H_{ST} mapping).

Denote by $coal_v$ the number of gene tree nodes mapped to node v under the H_{ST} mapping. That is, $coal_v = |\{u \in V(gt) : H_{ST}(u) = v\}|$. The function $lins$ can be computed recursively as

$$lins(v) = \begin{cases} 1 & \text{if } v \text{ is a leaf} \\ lins(x) + lins(y) - coal_v & \text{if } \{(v, x), (v, y)\} \subseteq E(ST) \end{cases} \quad (3.1)$$

We now prove that (1) for every node $v \in V(ST)$, $lins_v = k_B(gt)$ where $B = Cluster(v)$ in ST , and (2) $lins$ is computable in $O(n)$ time.

The proof of (1) uses strong induction on the height of node v , where the height of v is the length of the longest path from v to any leaf in $Cluster(v)$. For the base case, let v be a node of height 1, and its two children (which are leaves) x and y be labeled l_x and l_y , respectively. For this base case, $lins(x) = lins(y) = coal_x = coal_y = 1$. If $\{l_x, l_y\}$ is a cluster in gt , then $k_B(gt) = 1$, in which case the algorithm sets $lins(v)$ to $lins(x) + lins(y) - coal_v = 1 + 1 - 1 = 1$, since $coal_v = 1$. If $\{l_x, l_y\}$ is not a cluster in gt , then $k_B(gt) = 2$, and the algorithm sets $lins(v)$ to $lins(x) + lins(y) - coal_v = 1 + 1 - 0 = 2$, since no coalescence event occurs at node v . Therefore, $k_B(gt) = lins(v)$ for nodes v of height 1 in ST .

Now assume that $k_B(gt) = \text{ins}(v)$ for every node v of height at most p , and let w be a node in ST of height $p + 1$, with $B = \text{cluster}(w)$. Let x and y be w 's two children, with $B_x = \text{Cluster}(x)$ and $B_y = \text{Cluster}(y)$. Clearly, the height of x and y is smaller than p . By the induction hypothesis, $k_{B_x}(gt) = \text{ins}(x)$ and $k_{B_y}(gt) = \text{ins}(y)$. By Theorem 1, $k_B(gt)$ is equal to the number of lineages on the parent edge of B in ST , which is the sum of the numbers of lineages on the parent edges of B_x and B_y , minus the number of coalescence events that occur at node w ($k_B(gt) = k_{B_x}(gt) + k_{B_y}(gt) - \text{coal}_w$). By the strong induction hypothesis, this is identical to $\text{ins}(x) + \text{ins}(y) - \text{coal}_w$. By Equation (3.1), we have $\text{ins}(x) + \text{ins}(y) - \text{coal}_w = \text{ins}(w)$. This completes the proof that $k_B(gt) = \text{ins}(w)$.

For the second part, notice that computing the values of ins for all nodes in ST can be achieved by a bottom-up algorithm that traverses each node $v \in V(ST)$ exactly once. For each node v , the values of $\text{ins}(x)$ and $\text{ins}(y)$ of its children x and y are already computed, and the value of coal_v is already computed via the H_{ST} mapping. Thus, the algorithm takes $O(n)$ time. \square

3.3 MDC on rooted binary gene trees: The single-allele case

The MDC problem is the “minimize deep coalescence” problem; as formulated by Maddison in 1997 [Mad97], this is equivalent to finding a species tree that minimizes

the total number of extra lineages over all gene trees in \mathcal{G} . Thus, the MDC problem can be stated as follows: given a set \mathcal{G} of rooted, binary gene trees, we seek a species tree ST such that $XL(ST, \mathcal{G}) = \sum_{gt \in \mathcal{G}} XL(ST, gt)$ is minimized.

MDC is conjectured to be NP-hard, and no polynomial-time exact algorithm is known for this problem. However, it can be solved exactly using several techniques, as we now show.

Algorithms for MDC The material in this section was proposed by Than and Nakhleh [TN09]. The simplest technique to compute the optimal species tree with respect to a set \mathcal{G} of gene trees is to compute a minimum-weight clique of size $n - 2$ (where $|\mathcal{X}| = n$) in a graph which we now describe. Let \mathcal{G} be the set of gene trees in the input to MDC, and let $MDC(\mathcal{G})$ be the graph with one vertex for each non-trivial subset of \mathcal{X} (so $MDC(\mathcal{G})$ does not contain trivial clusters). Any two vertices A and B are connected by an edge if the two clusters are compatible (and so $A \cap B = \emptyset$, $A \subset B$, or $B \subset A$). A clique inside this graph therefore defines a set of pairwise compatible clusters and, hence, a rooted tree on \mathcal{X} . We set the weight of each node A to be $w(A) = \sum_{gt \in \mathcal{G}} [k_A(gt) - 1]$. We seek a clique of size $n - 2$, and among all such cliques we seek one of minimum weight. By construction, the clique will define a rooted, binary tree ST such that $XL(ST, \mathcal{G})$ is minimized.

The graph $MDC(\mathcal{G})$ contains $2^n - n - 1$ vertices, where $n = |\mathcal{X}|$, and is therefore large even for a relatively small n . We can constrain this graph size by restricting the allowable clusters to a smaller set, \mathcal{C} , of subsets of \mathcal{X} . For example, we can set $\mathcal{C} = \cup_{gt \in \mathcal{G}} Clusters(gt)$ (minus the trivial clusters), and we can define $MDC(\mathcal{C})$ to

be the subgraph of $MDC(\mathcal{G})$ defined on the vertices corresponding to \mathcal{C} . However, the cliques of size $n - 2$ in the graph $MDC(\mathcal{C})$ might not have the minimum possible weights; therefore, instead of seeking a minimum weight clique of size $n - 2$ within $MDC(\mathcal{C})$, we will set the weight of node A to be $w'(A) = Q - w(A)$, for some very large Q , and seek a *maximum weight* clique within the graph.

Finally, we can also solve the problem exactly using dynamic programming. For $A \subseteq \mathcal{X}$ and a binary rooted tree T on leaf-set A , we define

$$l_T(A, \mathcal{G}) = \sum_{gt \in \mathcal{G}} \sum_B [k_B(gt) - 1],$$

where B ranges over all clusters of T . We then set

$$l^*(A, \mathcal{G}) = \min\{l_T(A, \mathcal{G}) : T \in \mathcal{T}_A\}.$$

By Theorem 1, $l^*(\mathcal{X}, \mathcal{G})$ is the minimum number of extra lineages achievable in any species tree on \mathcal{X} , and so any tree T such that $l_T(\mathcal{X}, \mathcal{G}) = l^*(\mathcal{X}, \mathcal{G})$ is a solution to the MDC problem on input \mathcal{G} . We now show how to compute $l^*(A, \mathcal{G})$ for all $A \subseteq \mathcal{X}$ using dynamic programming. By backtracking, we can then compute the optimal species tree on \mathcal{X} with respect to the set \mathcal{G} of gene trees.

Consider a binary rooted tree T on leaf-set A that gives an optimal score for $l^*(A, \mathcal{G})$, and let the two subtrees off the root of T be T_1 and T_2 with leaf sets A_1 and $A_2 = A - A_1$, respectively. Then, letting B range over the clusters of T , we obtain

$$\begin{aligned} l_T(A, \mathcal{G}) &= \sum_{gt \in \mathcal{G}} \sum_B [k_B(gt) - 1] = \\ &= \sum_{gt \in \mathcal{G}} \sum_{B \subseteq A_1} [k_B(gt) - 1] + \sum_{gt \in \mathcal{G}} \sum_{B \subseteq A_2} [k_B(gt) - 1] + \sum_{gt \in \mathcal{G}} [k_A(gt) - 1]. \end{aligned}$$

If for $i = 1$ or 2 , $l_{T_i}(A_i, \mathcal{G}) \neq l^*(A_i, \mathcal{G})$, then we can replace T_i by a different tree on A_i and obtain a tree T' on A such that $l_{T'}(A, \mathcal{G}) < l_T(A, \mathcal{G})$, contradicting the optimality of T . Thus, $l_{T_i}(A_i, \mathcal{G}) = l^*(A_i, \mathcal{G})$ for $i = 1, 2$, and so $l^*(A, \mathcal{G})$ is obtained by taking the minimum over all sets $A_1 \subset A$ of $l^*(A_1, \mathcal{G}) + l^*(A - A_1, \mathcal{G}) + \sum_{gt \in \mathcal{G}} [k_A(gt) - 1]$. In other words, we have proven the following:

Lemma 3 $l^*(A, \mathcal{G}) = \min_{A_1 \subset A} \{l^*(A_1, \mathcal{G}) + l^*(A - A_1, \mathcal{G}) + \sum_{gt \in \mathcal{G}} [k_A(gt) - 1]\}$.

This lemma suggests the dynamic programming algorithm:

- Order the subsets of \mathcal{X} by cardinality, breaking ties arbitrarily.
- Compute $k_A(gt)$ for all $A \subseteq \mathcal{X}$ and $gt \in \mathcal{G}$.
- For all singleton sets A , set $l^*(A, \mathcal{G}) = 0$.
- For each subset with at least two elements, from smallest to largest, compute $l^*(A, \mathcal{G}) = \min_{A_1 \subset A} \{l^*(A_1, \mathcal{G}) + l^*(A - A_1, \mathcal{G}) + \sum_{gt \in \mathcal{G}} [k_A(gt) - 1]\}$.
- Return $l^*(\mathcal{X}, \mathcal{G})$.

There are $2^n - 1$ subproblems to compute (one for each set A) and each takes $O(2^n n)$ time (there are at most 2^n subsets A_1 of A , and each pair A, A_1 involves computing k_A for each $gt \in \mathcal{G}$, which costs $O(n)$ time). Hence, the running time is $O(n2^{2n})$ time. A tighter bound of $O(2^n + 3^n)$ can also be achieved. This follows from the fact that for each value of i , for $1 \leq i \leq n$, we have $\binom{n}{i}$ sets of size i , and for each of these sets, we need to consider all its subsets (there are 2^i of them) and score the number of extra lineages.

However, Than and Nakhleh showed that using only the clusters of the gene trees would produce almost equally good estimates of the species tree [TN09, TN10].

Chapter 4

Extending MDC

4.1 Estimated gene trees: The single-allele case

Estimating gene trees with high accuracy is a challenging task, particularly in cases where branch lengths are very short (which are also cases under which ILS is very likely to occur). As a result, gene tree estimates are often unrooted, unresolved, or both. To deal with these practical cases, we formulate the problems as estimating species trees and completely resolved, rooted versions of the input trees to optimize the MDC criterion. We show that the clique-based and DP algorithms can still be applied.

4.1.1 Unrooted, binary gene trees

When reconciling an unrooted, binary gene tree with a rooted, binary species tree under parsimony, it is natural to seek the rooting of the gene tree that results in the minimum number of extra lineages over all possible rootings; see the illustration in

Fig. 4.1. In this case, the MDC problem can be formulated as follows: given a set

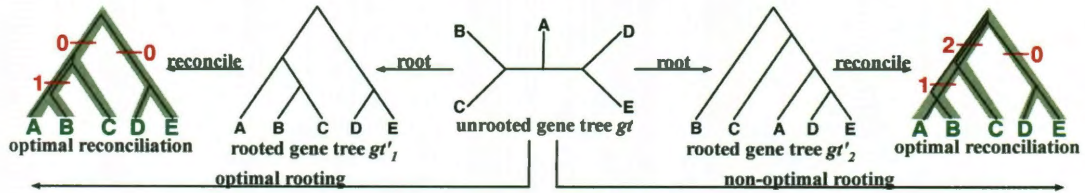


Figure 4.1: Illustration of optimal and non-optimal reconciliations of an unrooted, binary gene tree gt with a rooted, binary species tree ST , which yield 1 and 3 extra lineages, respectively.

$\mathcal{G} = \{gt_1, gt_2, \dots, gt_k\}$ of gene trees, each of which is unrooted, binary, with leaf-set \mathcal{X} , we seek a species tree ST and set $\mathcal{G}' = \{gt'_1, gt'_2, \dots, gt'_k\}$, where gt'_i is a rooted version of gt_i , so that $XL(ST, \mathcal{G}')$ is the minimum over all such sets \mathcal{G}' .

Given a species tree and a set of unrooted gene trees, it is easy to compute the optimal rootings of each gene tree with respect to the given species tree, since there are only $O(n)$ possible locations for the root in an n leaf tree, and for each possible rooting we can compute the score of that solution in $O(n^2)$ time. Thus, it is possible to compute the optimal rooting and its score in $O(n^3)$ time. Here we show how to solve this problem more efficiently – finding the optimal rooting in $O(n)$ time, and the score for the optimal rooting in $O(n^2)$ time, thus saving a factor of n . We accomplish this using a small modification to the techniques used in the case of rooted gene trees.

We begin by extending the definition of B -maximal clusters to the case of unrooted gene trees, for B a cluster in a species tree ST , in the obvious way. Recall that the set $Clusters(gt)$ depends on whether gt is rooted or not, and that $k_B(gt)$ is the number

of B -maximal clusters in gt . We continue with the following:

Lemma 4 *Let gt be an unrooted binary gene tree on \mathcal{X} , and let ST be a rooted binary species tree on \mathcal{X} . Let \mathcal{C}^* be the set of ST -maximal clusters in gt . Let e be any edge of gt such that $\forall Y \in \mathcal{C}^*, e \notin E(\text{Clade}_{gt}(Y))$ (i.e., e is not inside any subtree of gt induced by one of the clusters in \mathcal{C}^*). Then the tree gt' produced by rooting gt on edge e satisfies (1) $\mathcal{C}^* \subseteq \text{Clusters}(gt')$, and (2) $XL(ST, gt') = \sum_{B \in \text{Clusters}(ST)} [k_B(gt) - 1]$, which is the best possible rooted version of gt . Furthermore, there is at least one such edge e in gt .*

Proof: We begin by showing that there is at least one edge e that is outside Y for all $Y \in \mathcal{C}^*$. Pick a cluster $A_1 \in \mathcal{C}^*$ that is maximal (i.e., it is not a subset of any other cluster in \mathcal{C}^*); we will show that the parent edge of A_1 is outside all clusters in \mathcal{C}^* . Suppose e is inside cluster $A_2 \in \mathcal{C}^*$. Since A_1 is maximal, it follows that $A_2 \not\subseteq A_1$. However, if the parent edge of A_2 is not inside A_1 , then either A_2 is disjoint from A_1 or A_2 contains A_1 , neither of which is consistent with the assumptions that A_1 is maximal and the parent edge of A_1 is inside A_2 . Therefore, the parent edge of A_2 must be inside A_1 . In this case, $A_1 \cap A_2 \neq \emptyset$ and $A_1 \cup A_2 = \mathcal{X}$. Let B_i be the cluster in ST such that A_i is B_i -maximal, $i = 1, 2$. Then $B_1 \cap B_2 \neq \emptyset$, and so without loss of generality $B_1 \subseteq B_2$. But then $A_1 \cup A_2 \subseteq B_1 \cup B_2 = B_2$ and so $B_2 = \mathcal{X}$. But \mathcal{X} is the only \mathcal{X} -maximal cluster, contradicting our hypotheses. Hence the parent edge of any maximal cluster in \mathcal{C}^* is not inside any cluster in \mathcal{C}^* .

We now show that rooting gt on any edge e that is not inside any cluster in \mathcal{C}^* satisfies $\mathcal{C}^* \subseteq \text{Clusters}(gt')$. Let e be any such edge, and let gt' be the result

of rooting gt on e . Under this rooting, the two children of the root of gt' define subtrees T_1 , with cluster A_1 , and T_2 , with cluster A_2 . Now, suppose $\exists A' \in \mathcal{C}^*$ - $Clusters(gt')$. Since $\mathcal{C}^* \subseteq Clusters(gt)$, it follows that A' is the complement of a cluster $B \in Clusters(gt')$. If B is a proper subset of either A_1 or A_2 , then the subtree of gt induced by A' contains edge e (since $A' = \mathcal{X} - B$), contradicting how we selected e . Hence, it must be that $B = A_1$ or $B = A_2$. However, in this case, A' is also equal to either A_1 or A_2 , and hence $A' \in Clusters(gt')$, contradicting our hypothesis about A' .

We finish the proof by showing that $XL(ST, gt')$ is optimal for all such rooted trees gt' , and that all other locations for rooting gt produce a larger number of extra lineages. By Theorem 1, $XL(ST, gt') = \sum_B [k_B(gt') - 1]$, as B ranges over the clusters of ST . By construction, this is exactly $\sum_B [k_B(gt) - 1]$, as B ranges over the clusters of ST . Also note that for *any* rooted version gt^* of gt , $k_B(gt^*) \geq k_B(gt)$, so that this is optimal. Now consider a rooted version gt^* in which the root is on an edge that is inside some subtree of gt induced by $A \in \mathcal{C}^*$. Let gt^* have subtrees T_1 and T_2 with clusters A_1 and A_2 , respectively. Without loss of generality, assume that $A_1 \subset A$, and that $A_2 \cap A \neq \emptyset$. Since $A \in \mathcal{C}^*$, there is a cluster $B \in Clusters(ST)$ such that A is B -maximal. But then A_1 is B -maximal. However, since $A - A_1 \neq \emptyset$, there is also at least one B -maximal cluster $Y \subset A$ within T_2 . Hence, $k_B(gt^*) > k_B(gt)$. On the other hand, for all other clusters B' of ST , $k_{B'}(gt^*) \geq k_{B'}(gt') = k_{B'}(gt)$. Therefore, $XL(ST, gt^*) > XL(ST, gt')$. In other words, any rooting of gt on an edge that is not within a subtree induced by a cluster in \mathcal{A} is optimal, while any rooting of gt on any other edge produces a strictly larger number of extra lineages. \square

This theorem allows us to compute the optimal rooting of an unrooted binary gene tree with respect to a rooted binary species tree and, hence, gives us a way of computing the score of any candidate species tree with respect to a set of unrooted gene trees:

Corollary 2 *Let ST be a species tree, and $\mathcal{G} = \{gt_1, gt_2, \dots, gt_k\}$ be a set of unrooted binary gene trees. Let $\mathcal{G}' = \{gt'_1, gt'_2, \dots, gt'_k\}$ be a set of binary gene trees such that gt'_i is a rooted version of gt_i for each $i = 1, 2, \dots, k$, and which minimizes $XL(ST, \mathcal{G}')$. Then $XL(ST, \mathcal{G}') = \sum_i \sum_{B \in Clusters(ST)} [k_B(gt_i) - 1]$. Furthermore, the optimal \mathcal{G}' can be computed in $O(nk)$ time, and the score of \mathcal{G}' computed in $O(n^2k)$ time.*

Solving MDC given unrooted, binary gene trees. Let $\mathcal{G} = \{gt_1, gt_2, \dots, gt_k\}$, as above. We define the MDC-score of a candidate (rooted, binary) species tree ST by $\sum_i \sum_{B \in Clusters(ST)} [k_B(gt_i) - 1]$; by Corollary 2, the tree ST^* that has the minimum score will be an optimal species tree for the MDC problem on input \mathcal{G} . As a result, we can use all the techniques used for solving MDC given binary rooted gene trees, since the score function is unchanged.

4.1.2 Rooted, non-binary gene trees

When reconciling a rooted, non-binary gene tree with a rooted, binary species tree under parsimony, it is natural to seek the refinement of the gene tree that results in the minimum number of extra lineages over all possible refinements; see the illustration in Fig. 4.2. In this case, the MDC problem can be formulated as follows: given a set

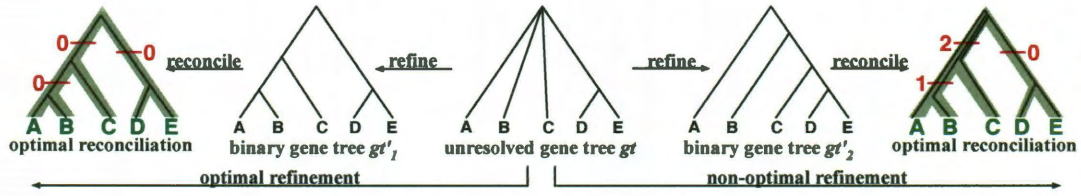


Figure 4.2: Illustration of optimal and non-optimal reconciliations of a rooted, non-binary gene tree gt with a rooted, binary species tree ST , which yield 0 and 3 extra lineages, respectively.

$\mathcal{G} = \{gt_1, gt_2, \dots, gt_k\}$ in which each gt_i may only be partially resolved, we seek a species tree ST and binary refinements gt_i^* of gt_i so that $XL(ST, \mathcal{G}^*)$ is minimized, where $\mathcal{G}^* = \{gt_1^*, gt_2^*, \dots, gt_k^*\}$. This problem is at least as hard as the MDC problem, which is conjectured to be NP-hard.

Yu, Warnow, and Nakhleh [YWN11a, YWN11b] proposed a quadratic algorithm OTR_{MDC} for optimal refinement of a given gene tree gt with respect to a given species tree ST , with both trees rooted, under MDC. It refines around each high degree node v in gt using the subtree of ST defined by the LCAs (least common ancestor) in ST of the children of v .

For a cluster B , we first define two sets of nodes in gt :

- $D_1 = \{v \in V(gt): v \text{ has at least two children whose cluster is B-maximal} \}$
- $D_2 = \{v \in V(gt): \text{Cluster}(v) \text{ is a B-maximal cluster, and } v \text{ is not a child of any node in } D_1\}$

And then we define $D_B(gt)$ to be $|D_1| + |D_2|$.

Corollary 3 *Let gt be a rooted gene tree, ST a rooted binary species tree, both on set \mathcal{X} . Then $XL(ST, gt^*) = \sum_{B \in \text{Clusters}(ST)} [D_B(gt) - 1]$, where gt^* is a binary refinement of gt that minimizes $XL(ST, gt')$ over all binary refinements gt' of gt .*

Proof: Yu, Warnow, and Nakhleh [YWN11a, YWN11b] proved that $XL(ST, gt^*) = \sum_{B \in \text{Clusters}(ST)} [F_B(gt) - 1]$, where $F_B(gt)$ is the number of nodes in gt that have at least one child whose cluster is B-maximal. We will show that $D_B(gt) = F_B(gt)$.

Let $D = D_1 \cup D_2$, so that $D_B(gt) = |D|$. And let $F = F_1 \cup F_2$, where F_1 is the set of nodes in gt that have at least two children whose cluster is B-maximal, and F_2 is the set of nodes in gt that have exactly one child whose cluster is B-maximal, so that $F_B(gt) = |F| = n$. Obviously, $D_1 = F_1$, which follows $|D_1| = |F_1|$. We will prove next $|D_2| = |F_2|$.

For any node in F_2 , its only child node whose cluster is B-maximal must be some node in D_2 , because its cluster is B-maximal and it has no sister node whose cluster is B-maximal. So $|F_2| \leq |D_2|$. For any node in D_2 , since it is the only child node of its parent node whose cluster is B-maximal, it must be some node in F_2 . So $|D_2| \geq |F_2|$. As a result, $|D_2| = |F_2|$. \square

F_B will be used instead of D_B in the rest of the thesis.

Corollary 4 *Let ST be a species tree and $\mathcal{G} = \{gt_1, gt_2, \dots, gt_k\}$ be a set of gene trees that may not be resolved. Let $\mathcal{G}^* = \{gt_1^*, gt_2^*, \dots, gt_k^*\}$ be a set of binary gene trees such that gt_i^* refines gt_i for each $i = 1, 2, \dots, k$, and which minimizes $XL(ST, \mathcal{G}^*)$. Then $XL(ST, \mathcal{G}^*) = \sum_i \sum_{B \in \text{Clusters}(ST)} [F_B(gt_i) - 1]$. Furthermore, the optimal resolution*

of each gene tree and its score can be computed in $O(n^2k)$ time.

Solving MDC given rooted, non-binary gene trees. Let $\mathcal{G} = \{gt_1, gt_2, \dots, gt_k\}$, as above. We define the MDC-score of a candidate (rooted, binary) species tree ST by $\sum_i \sum_{B \in \text{Clusters}(ST)} [F_B(gt_i) - 1]$; by Corollary 4, the tree ST^* that has the minimum score will be an optimal species tree for the MDC problem on input \mathcal{G} . As a result, we can use all the techniques used for solving MDC given binary rooted gene trees, since the score function is unchanged.

4.1.3 Unrooted, non-binary gene trees

When reconciling an unrooted and incompletely resolved gene tree with a rooted, binary species tree under parsimony, it is natural to seek the rooting and refinement of the gene tree that results in the minimum number of extra lineages over all possible rootings and refinements; see the illustration in Fig. 4.3. In this case, the MDC

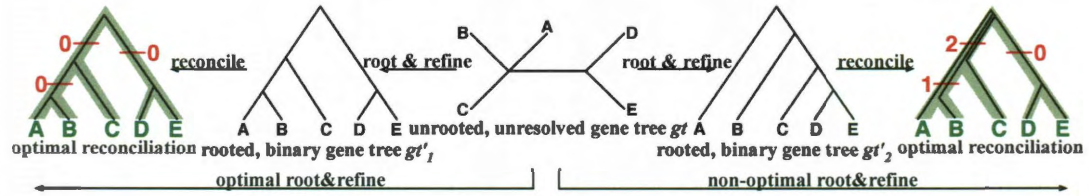


Figure 4.3: Illustration of optimal and non-optimal reconciliations of an unrooted, non-binary gene tree gt with a rooted, binary species tree ST , which yield 0 and 3 extra lineages, respectively.

problem can be formulated as follows: given a set $\mathcal{G} = \{gt_1, gt_2, \dots, gt_k\}$, with each gt_i a tree on \mathcal{X} but not necessarily rooted nor fully resolved, we seek a rooted, binary

species tree ST and set $\mathcal{G}' = \{gt'_1, gt'_2, \dots, gt'_k\}$ such that each gt'_i is a binary rooted tree that can be obtained by rooting and refining gt_i , so as to minimize $XL(ST, \mathcal{G}')$ over all such \mathcal{G}' . As before, the computational complexity of this problem is unknown, but conjectured to be NP-hard.

Observation 1 *For any gene tree gt and species tree ST , and t^* the optimal refined rooted version of gt that minimizes $XL(ST, t^*)$ can be obtained by first rooting gt at some node, and then refining the resultant rooted tree. Thus, to find t^* , it suffices to find a node $v \in V(gt)$ at which to root the tree t , thus producing a tree t' , so as to minimize $\sum_{B \in Clusters(ST)} [F_B(gt') - 1]$.*

We now show how we will find an optimal root x .

Notation Let x be a node in gt , and let $gt^{(x)}$ denote the tree obtained by rooting gt at x . Let gt be a given gene tree, ST a given species tree, and X a cluster of gt . Let $r'(X)$ be the far endpoint of the parent edge of X within gt (i.e., the parent edge of X is (r, r') , where r is the root of X in gt). For a given cluster B of ST , we will say that two clusters A and A' of gt are B -siblings if A and A' are B -maximal clusters and their roots share a common neighbor. For a fixed cluster $B \in Clusters(ST)$, we will refer to a maximal set of B -maximal clusters in gt that are pairwise siblings as a *family of B -maximal clusters in gt* . A family of B -maximal clusters will also be referred to as a family of ST -maximal clusters. We denote by $gt(A)$ the subtree of gt induced by cluster A .

Lemma 5 *Let A be a largest ST -maximal cluster in gt , and $a = r'(A)$. Then a is not internal to any ST -maximal clade in gt , nor is a the root of any ST -maximal clade that is in a family of size at least two.*

Proof: Let A be a largest ST -maximal cluster in gt , and suppose A is X -maximal for $X \in Clusters(ST)$ with $a = r'(A)$. Suppose a is inside a B -maximal cluster Y , with Y produced by edge $e \in E(gt)$. If e is inside A , then $Y \cap A \neq \emptyset$, $A - Y \neq \emptyset$, and $A \cup Y = \mathcal{X}$. Since $A \subseteq X$ and $Y \subseteq B$, it follows that $X \cup B = \mathcal{X}$ and also that $X \cap B \neq \emptyset$. Since X and B are both clusters in ST , they must be compatible, and so one must contain the other. But then one must be the entire set \mathcal{X} , which is a contradiction.

Suppose instead that a is the root of a B -maximal cluster Z of gt , with Z defined by edge e , and that Z has a sibling Z' . Thus, Z' is also B -maximal, and the roots of Z and Z' share a common neighbor. We consider first the case where the edge e defining cluster Z is the parent edge of A . In this case, the edge e in gt splits the tree into clusters A and Z . Since Z is B -maximal and A is X -maximal, and B and X are both clusters in ST , it follows that $B \cup X = \mathcal{X}$. Thus, the two subtrees off the root of ST are on leafset B and on leafset X . But then $Z = B$ and $A = X$. Since Z and Z' are disjoint, it follows that $Z' \subseteq A$, contradicting that $Z' \subseteq B$.

We now consider the case where the edge e defining Z is some other edge incident with a . But then since a is the root of Z , it follows that A is a proper subset of Z , contradicting that A is the largest ST -maximal cluster in gt .

Hence, for a , the far endpoint of the parent edge of a largest ST -maximal cluster in gt , a is not the root of any ST -maximal cluster that is in a family of size two or more, nor is a internal to any ST -maximal cluster. \square

Theorem 2 *Let A be a largest ST -maximal cluster in gt , and $a = r'(A)$. Then $F_B(gt^{(a)}) \leq F_B(gt^{(r)})$ for all clusters $B \in \text{Clusters}(ST)$ and for all nodes $r \in V(gt)$.*

Proof: By Lemma 5, a is not internal to any ST -maximal cluster of gt , and a is not the root of any cluster that is in a family of size at least two. Let B be a cluster in ST . Since a is not internal to any B -maximal cluster and not the parent of any B -maximal cluster that has a sibling, it follows that the set of B -maximal clusters of $gt^{(a)}$ is identical to the set of B -maximal clusters of gt . Hence, the number of families of B -maximal clusters of $gt^{(a)}$ is identical to the number of families of B -maximal clusters of gt , and so equal to $F_B(gt^{(a)})$. Furthermore, it is easy to see that $F_B(gt^{(r)})$ is at least the number of families of B -maximal clusters of gt , no matter what r is. Hence, $F_B(gt^{(r)}) \geq F_B(gt^{(a)})$ for all vertices r . \square

The following two corollaries follow directly from Theorem 2:

Corollary 5 *Let gt be an unrooted, not necessarily binary gene tree on \mathcal{X} , and let ST be a rooted species tree on \mathcal{X} . Let $A \in \text{Clusters}(gt)$ be a largest ST -maximal cluster, and $a = r'(A)$ the far endpoint of the parent edge for A . If we root gt at*

a , then the resultant tree $gt^{(a)}$ minimizes $\sum_{B \in \text{Clusters}(ST)} [F_B(gt') - 1]$ over all rooted versions gt' of gt .

Corollary 6 *Let \mathcal{T} be a set of gene trees that are unrooted and not necessarily binary. For $t \in \mathcal{T}$ and $B \subset \mathcal{X}$, define t^B to be the rooted version of t formed by rooting t at $r'(B)$. Then, the species tree ST that minimizes $\sum_{t \in \mathcal{T}} \sum_{B \in \text{Clusters}(ST)} [F_B(t^B) - 1]$ is an optimal species tree for \mathcal{T} .*

Solving MDC given rooted, non-binary gene trees. As a result of Corollary 6, we can solve the problem using the clique and DP formulations as in the other versions of the MDC problem.

4.2 Incorporating coalescence times

In this section, we extend the MDC criterion to the case where nodes of the gene trees have times associated with them, which correspond to coalescence times. In this case, those times constitute constraints on coalescent histories, in addition to those imposed by the topologies of the gene trees. More precisely, if a set Y of leaves coalesce at time t in gene tree gt , then t is an upper bound (recall that under the coalescent, time increases when going back from the leaves toward the root) on the speciation time of that set of leaves in the species tree. For example, assume that alleles from three species A , B , and C coalesced 100 generations ago. Then, the MRCA of these three species must have existed ≤ 100 generations ago, since otherwise there would not be a valid coalescent history for the three alleles with a coalescence time of 100

(unless, for example, gene flow occurred after the speciation; however, we do not consider such events here and focus only on ILS, as they are beyond the scope of this thesis). We now give an algorithm for inferring a rooted, binary species tree ST from a collection \mathcal{G} of gene trees, where each gene tree has coalescence times associated with its internal nodes.

Let $e = (u, v)$ be a branch in a rooted gene tree; that is, v is a child of u . Further, for a node x , let $t(x)$ denote the coalescence time associated with it. Clearly, $C(v) \subseteq C(u)$ and $t(v) \leq t(u)$ (under the coalescent, time increases as we move from the leaves toward the root). Since each node in a rooted tree is uniquely defined by the cluster of taxa under it, we use $(C(u), C(v))$ to denote the branch (u, v) . Under this setup, two branches, (c_1, c_2) and (c'_1, c'_2) , are *identical* if and only if $c_1 = c'_1$, $c_2 = c'_2$, $t(c_1) = t(c'_1)$, and $t(c_2) = t(c'_2)$. Further, We define the *compatibility* of two branches as follows.

Definition 1 (*Compatibility of two branches*)

Two branches (c_1, c_2) and (c'_1, c'_2) are compatible if one of the following three conditions holds:

- $c_1 \cap c'_1 = \emptyset$
- $c_1 = c'_1, t(c_1) = t(c'_1), c_2 \cap c'_2 = \emptyset$
- $c_1 \subseteq c'_1, t(c_1) \leq t(c'_1)$ and
 $c_1 \subseteq c'_2, t(c_1) \leq t(c'_2)$ or $c_1 \cap c'_2 = \emptyset$.

We denote by $\beta((c_1, c_2), gt)$ the number of extra lineages on branch (u, v) (i.e., $C(u) = c_1$ and $C(v) = c_2$) in species tree ST resulting from reconciling gene tree gt

into ST according to their coalescence times. Let l_1, \dots, l_k be all the maximal clusters in gt that are also subsets of c_2 and satisfy $t(l_i) < t(c_1)$ whenever $|l_i| > 1$. Then, the number of extra lineages on branch (u, v) in ST is $\beta((c_1, c_2), gt) = k - 1$.

Given a collection \mathcal{G} of rooted gene trees with coalescence times, we denote by $\ell_{time}(ST, \mathcal{G})$ the sum of $\sum_{g \in \mathcal{G}} \beta(br, g)$ for all branches br in ST , including the branch incident into the root of ST . Further, we denote by $\ell_{time}^*((A, A'), \mathcal{G})$ the minimum value of $\ell(ST, \mathcal{G})$ over all possible trees ST , with times at internal nodes, on A , where (A, A') is the branch incident into the root of ST . If $A = \mathcal{X}$, then we have $A' = \mathcal{X}$ and $t(A') = +\infty$; otherwise, ST is a subtree in a larger tree ST' and the root of ST is a child of an internal node, say v , in ST' , then $A' = C_{ST'}(v)$ and $t(A') = t(v)$. If t_1 and t_2 are the two subtrees whose roots are the two children of the root of ST , then we have $\ell_{time}(ST, \mathcal{G}) = \ell_{time}(t_1, \mathcal{G}) + \ell_{time}(t_2, \mathcal{G}) + \sum_{g \in \mathcal{G}} \beta(br, g)$, where br is the branch incident into the root of ST . The quantity $\sum_{g \in \mathcal{G}} \beta(br, g)$ is fixed for each br . Therefore, we can compute $\ell_{time}^*((A, A'), \mathcal{G})$ recursively as follows.

1. Let \mathcal{B} be a collection of all branches (with times) on \mathcal{X} . We partition \mathcal{B} into subsets $\mathcal{B}_1, \dots, \mathcal{B}_{|\mathcal{X}|}$, where \mathcal{B}_i , $1 \leq i \leq |\mathcal{X}|$, is the collection of all branches (A, A') in \mathcal{B} with $|A| = i$.
2. For every $(A, A') \in \mathcal{B}_1$, $\ell_{time}^*((A, A'), \mathcal{G}) = 0$, and for $(A, A') \in \mathcal{B}_2$,
$$\ell_{time}^*((A, A'), \mathcal{G}) = \sum_{g \in \mathcal{G}} \beta((A, A'), g).$$
3. For $(A, A') \in \mathcal{B}_i$, $3 \leq i \leq |\mathcal{X}|$,

$$\ell_{time}^*((A, A'), \mathcal{G}) = \min\{\ell_{time}^*((A_1, A), \mathcal{G}) + \ell_{time}^*((A_2, A), \mathcal{G}) : (A_j, A) \text{ and } (A, A') \text{ are compatible for } j = 1, 2, A_1 \cap A_2 = \emptyset \text{ and } A_1 \cup A_2 = A\} + \sum_{g \in \mathcal{G}} \beta((A, A'), g).$$

4. Return $\ell_{time}^*(\mathcal{X}, \mathcal{G})$.

We next show how to obtain \mathcal{B} in Step 1. As mentioned before, nodes in the species tree must satisfy the constraint that the coalescence time of a set of leaves in the gene tree must be an upper bound on the speciation time of that set of leaves in the species tree. It implies that the coalescence time of a node cannot be assigned without specifying its children. For example, assume we have two gene trees gt_1 and gt_2 , as shown in Fig. 4.4, Then for cluster BCD , if we have branch (BCD, CD) in

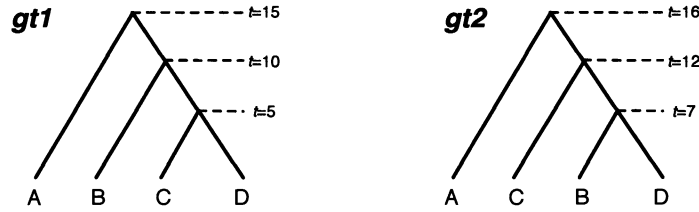


Figure 4.4: Illustration of constraints on speciation times of nodes in the species tree imposed by gene trees. Particularly, we have two gene trees gt_1 and gt_2 with branch lengths on the same taxa set.

the species tree, we will have $t(BCD) = 7$ for that branch due to the coalescence time of (B, D) in gt_2 . However, if we have branch (BCD, BD) in the species tree, we will have $t(BCD) = 5$ due to the coalescence time of (C, D) in gt_1 . Besides, the speciation time of a node in the species tree should also be no later than the speciation time of its parent node. More precisely, if we have candidate branches (c_1, c_2) and (c_2, c_3) for the species tree, the time for c_2 can be calculated as

$$t(c_2) = \min\{\min\{Coal(a, b) : a \in c_2 - c_3, b \in c_3\}, t(c_1)\},$$

where $Coal(a, b)$ is the minimum coalescence time for taxa a and b in all gene trees.

Therefore, we could compute \mathcal{B} from an empty set as follows.

1. Let \mathcal{C} be a collection of all non-empty subsets of \mathcal{X} . For every $c \in \mathcal{C}$, if $|c| = 1$, $t(c) = 0$; otherwise, $t(c) = \min\{t(MRCA_g(c')) : c \subseteq c', c' \in \mathcal{C}, g \in \mathcal{G}\}$.
2. Build a map \mathcal{M} to keep the minimal coalescence time for every pair of taxa a and b ; i.e., $\mathcal{M}(a, b) = \min\{t(c) : a, b \in c, c \in \mathcal{C}\}$.
3. Sort all clusters in \mathcal{C} in a decreasing order of size. For every $c_2 \in \mathcal{C}$ with $t_2 = t(c_2)$, let c_1^1, \dots, c_1^k be the clusters in \mathcal{C} with $c_2 \subset c_1^i$ for $1 \leq i \leq k$. Then for every c_2 and c_1^i pair, do
 - (a) Let $t_{min} = \min\{\mathcal{M}(a, b) : a \in c_1^i - c_2, b \in c_2\}$.
 - (b) Let $tlist$ be a set of all possible coalescence times for c_1^i ; i.e., $tlist = \{\min\{t(c_2'), t_{min}\} : (c_1', c_2') \in \mathcal{B}, c_1^i = c_2'\}$.
 - (c) For every $t \in tlist$, make a branch (c_1', c_2') , where $c_1' = c_1^i$ with $t(c_1') = t$, and $c_2' = c_2$ with $t(c_2') = \min\{t_2, t\}$, and add it into \mathcal{B} .
4. Add branch $(\mathcal{X}, \mathcal{X}')$ to \mathcal{B} , where $\mathcal{X}' = \mathcal{X}$ and $t(\mathcal{X}') = +\infty$, which is the branch incident into the root.

4.3 Multiple-allele cases

Thus far, we have assumed that exactly a single allele is sampled per species in the analysis. However, sequences of multiple individuals per species are becoming increasingly available. Therefore, it is necessary to develop methods that infer species trees from data sets that contain zero or more alleles per species for the different loci. Than and Nakhleh [TN10] described how to extend the algorithms from their earlier work [TN09] to the case of multiple alleles in a straightforward manner. To illustrate

this case, consider the scenario in Fig. 4.5. Here, the numbers of alleles sampled from the species A , B , C , D , and E , for the given locus are 3, 1, 2, 0, and 2, respectively. Under the MDC criterion, a cluster of alleles in the gene tree coalesce at their MRCA in the species tree, where the MRCA is taken over the set of species to which the alleles belong. For example, the cluster $\{A_1, A_2\}$ coalesces on the parent edge of species A , whereas the cluster $\{A_1, A_2, B_1\}$ coalesces on the parent edge of cluster $\{A, B\}$. We now formalize the MDC criterion for this case.

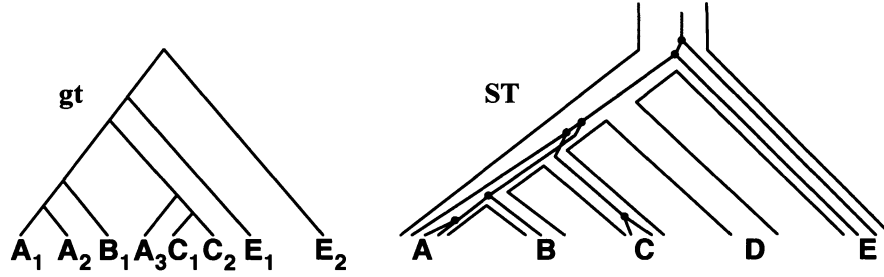


Figure 4.5: Illustration of ILS and the MDC criterion in the case of multiple alleles. X_i denotes an allele of species X . In particular, species D has no alleles sampled for the given locus.

Let ST be a rooted, binary species tree on set \mathcal{X} of taxa. Let gt be a gene tree injectively leaf-labeled by the elements of $\mathcal{A} = \cup_{x \in \mathcal{X}} a(x)$, where $a(x)$ is a set of alleles for species x . In other words, every leaf in gt is labeled uniquely by an element in \mathcal{A} , but there may be labels in \mathcal{A} to which no leaf in gt is mapped. In Fig. 4.5, we have $a(A) = \{A_1, A_2, A_3\}$, $a(B) = \{B_1\}$, $a(C) = \{C_1, C_2\}$, $a(D) = \emptyset$, and $a(E) = \{E_1, E_2\}$.

For every set of alleles W of a given locus, we denote by $\alpha(W)$ the set of all species

that have alleles in W . In Fig. 4.5, for the set $W = \{A_1, A_2, B_1, C_1, C_2, A_3\}$, we have $\alpha(W) = \{A, B, C\}$. Using the α mapping, we can now define the MRCA mapping for the multiple-allele case.

Let v be a node in gt and, as before, denote by $Cluster(v)$ its cluster. Then, the MRCA mapping $H : V(gt) \rightarrow V(ST)$ is defined by $H_{ST}(v) = MRCA_{ST}(\alpha(Cluster_{gt}(v)))$. Notice that under this mapping, if $Cluster(v)$ contains only alleles of a single species (e.g., $Cluster(v) \subseteq a(x)$ for some $x \in \mathcal{X}$), then $H_{ST}(v) = x$. Given a cluster A in gt and a cluster B in ST , we say that A is B -maximal if (1) $\alpha(A) \subseteq B$, and (2) for all $A' \in Clusters(gt)$, if $A \subseteq A'$, then $\alpha(A') \not\subseteq B$. We set $k_B(gt)$ as before to be the number of B -maximal clusters within gt . Further, we say that cluster A in gt is ST -maximal if there is a cluster $B \in Clusters(ST)$ such that $B \neq \mathcal{X}$ and A is B -maximal.

Now that we have established the definitions, Theorem 1 applies directly.

Theorem 3 *Let ST be a rooted, binary species tree on set \mathcal{X} of taxa, and gt be a rooted, binary gene tree leaf-labeled by set \mathcal{A} of alleles of \mathcal{X} . Let B be a cluster in ST , and let e be the parent edge of B in ST . Then $k_B(gt)$ is equal to the number of lineages on e in an optimal valid coalescent history. Therefore, $XL(e, gt) = k_B(gt) - 1$, and $XL(ST, gt) = \sum_B [k_B(gt) - 1]$, where B ranges over the clusters of ST . Furthermore, a valid coalescent history f that achieves this total number of extra lineages can be produced by setting $f(v) = H_{ST}(v)$ (i.e., $f(v) = MRCA_{ST}(\alpha(Cluster_{gt}(v)))$) for all v .*

Proof: Let $B = Cluster_{ST}(v)$ and e be the parent edge of node v . We prove that

$k_B(gt)$ is equal to the number of lineages on e in an optimal valid coalescent history by induction on the height of node v (as before, the height of a node is the longest distance from the node to a leaf under it). In particular, as above, we show that $k_B(gt) = k_C(gt) + k_D(gt) - \text{numcoal}(B)$, where C and D are the children clusters of B in the species tree, and $\text{numcoal}(B)$ is the number of coalescent events that occur on the parent edge of B in an optimal valid coalescent history.

For the base case, consider node v of height 0, that is, v is a leaf. Further, assume v is labeled by taxon $B \in \mathcal{X}$ (here, the cluster B has a single element, therefore, we used B as the element of the cluster itself). In this case, B has no children clusters, and the B -maximal clusters in g are exactly all maximal subtrees t_1, t_2, \dots, t_k of gt , where $L(t_i) \subseteq a(B)$ for $1 \leq i \leq k$. An optimal valid coalescent history will have all leaves of t_i , for each i , coalesce within the parent edge of B . In other words, the number of lineages in the parent edge of B under such a valid coalescent history is k , which is equal to the number of B -maximal clusters within gt . Thus, the result holds for all nodes of height 0.

For the induction hypothesis, we assume the result holds for all nodes $v \in V(ST)$ of height p , and prove the result for nodes of height $p + 1$. Let $u \in V(ST)$ be a node such that v, w are its children, and their height is p . Further, let $B = \text{Cluster}_{ST}(u)$ whose parent edge is e , $C = \text{Cluster}_{ST}(v)$ and $D = \text{Cluster}_{ST}(w)$. By the induction hypothesis, we have that $k_C(gt)$ and $k_D(gt)$ equal the number of extra lineages in an optimal valid coalescent history on the parent edges of clusters C and D , respectively. The number of lineages that “enter” edge e from below (i.e., from its endpoint that is closer to the leaves) in an optimal valid coalescent history is $k_C(gt) + k_D(gt)$. If

$numcoal(B)$ is the number of coalescent events that take place on edge e in an optimal valid coalescent history, and given that each coalescence event decreases the number of lineages by 1, then the number of edges that “exit” e from above (i.e., from its endpoint that is closer to the root) is $k_C(gt) + k_D(gt) - numcoal(B)$, which is, by definition, $k_B(gt)$. This completes the proof. \square

Chapter 5

Performance

5.1 Simulated data

To generate the data sets, we used the Mesquite tool of W.P. Maddison and D.R. Maddison [MM04] and similar procedure and parameters used by W.P. Maddison and Knowles [MK06].

Species trees were simulated by using the “Uniform Speciation” (Yule) module in Mesquite. Two sets of species trees were generated: one for those with a total branch length of 100,000 generations, and one for 1,000,000 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. Further, the evolution of DNA sequences of length 1000 base pair was simulated down each gene tree under the model of Jukes-Cantor [JC69]. Thus, sequence alignments were obtained under all settings.

For each such sequence alignment, we built a gene tree using a maximum parsimony (MP) heuristic with strict consensus in PAUP* [Swo95]. This setup allowed us to study not only the performance of methods on the true gene trees, but also those reconstructed MP, thus accounting for potential errors in the gene trees.

5.2 Methods

We studied the performance of several methods, including greedy consensus, democratic vote, STEM, GLASS, and MDC.

Summary statistic inference: greedy consensus

Greedy consensus can be viewed as an extension of *majority consensus*. Majority consensus computes from a collection of gene trees the tree whose every branch appears in at least half of the gene trees in the input. It guarantees that this collection of branches is *pairwise compatible*; that is, they can simultaneously be the branches of a tree. A salient feature of trees computed by the majority consensus method is that they typically lack resolution, and usually have a high rate of false negatives and a low rate of false positives. As an extension, the greedy consensus method continuously refines the majority consensus tree by branches with lower-than-50% frequency. Instead of using only branches with higher-than-50% frequency, it orders the rest of the branches by decreasing frequency, and adds them one by one to the consensus tree once it is compatible. Both majority consensus and greedy consensus have been implemented in PhyloNet.

In our study, we run the greedy consensus method on all data-sets.

Summary statistic inference: democratic vote

Democratic vote simply amounts to declaring the gene tree with the highest frequency in the input as the species tree estimate [DR09b]. However, an important issue that has not been addressed before is the potential non-uniqueness of a highest-frequency tree. Indeed, our coalescent simulations show that more than a single gene tree in the input may attain the highest frequency. In this case, the highest-frequency is not well-defined. Therefore, we explore below the performance of the democratic vote when a random maximum-frequency tree is used, and when the majority consensus of all maximum-frequency trees is used. Based on our findings below, we have implemented in PhyloNet a version of the democratic vote method that uses the greedy consensus of all maximum-frequency trees.

In our study, the democratic vote method was run only on single-allele data sets.

Likelihood-based inference: STEM

The tool STEM [KCK09] implements a maximum likelihood approach that infers a species tree ST using the likelihood function

$$L(ST, \tau) = \prod_{i=1}^k f(g_i | ST, \tau),$$

where τ is a vector of the branch lengths of the species tree ST , and f is the gene tree density function [RY03].

In our study, STEM was run only on the true gene trees, because of one of the

restrictions in the STEM tool that the gene trees in the input must be rooted and must satisfy the molecular clock assumption.

Distance-based inference: GLASS

GLASS [MR10] is a distance-based method that applies a clustering algorithm to pairwise distances between species computed based on the coalescence times of the input loci. Formally, let $D_{rs}^{(i)}$ denote the minimum divergence time between any two alleles of locus i , where one is sampled from species r and the other is sampled from species s , and define the distance between two species A and B as

$$D_{AB} = \min_{1 \leq i \leq k} \{D_{AB}^{(i)}\},$$

where k is the number of loci. Based on these species pairwise distances, a species tree is inferred by a clustering algorithm that joins the closest species (or, in intermediate steps, the closest clusters) and updates distances, following the algorithm of Rosenberg’s [Ros02].

We have implemented GLASS in the PhyloNet. It allows the user to specify the $D_{rs}^{(i)}$ distance matrices, and computes the species tree estimate based on the GLASS algorithm. This implementation provides the user with flexibility as to what source of data can be used, since those distances could be computed from the sequences directly, or taken from gene tree estimates, for example.

Since GLASS uses divergence times computed from all loci, in our study, to run GLASS, we used the true coalescence times from the gene trees, as well as times estimated from sequences according to the model of Jukes-Cantor [JC69].

Parsimony-based inference: MDC

For MDC without time, we run both exact and heuristic versions on all data-sets. The “exact” version uses all possible clusters on the taxon set, and the “heuristic” version uses only the clusters of the input gene trees. For the heuristic MDC, the estimated species tree may not be fully resolved. In this case, we followed this initial analysis with a search through the set of binary resolutions of the initial estimated species tree for a fully resolved tree that optimized the number of extra lineages.

The extended MDC with time method was run only on the true gene trees, since the extension works for the case of rooted trees.

5.3 Results and discussion

In this section we report on the results of the experiments we performed, in terms of both accuracy and speed. For accuracy of the species tree inference, since the species tree is known for simulated data, we compared the inferred species tree against the true species tree by the normalized [RF81] measure, which quantifies the average proportion of branches present in one, but not both, of the trees. A value 0 of the RF distance indicates that the two trees are identical, and a value of 1 indicates that the two trees are completely different (they disagree on every branch).

The democratic vote

When using the democratic vote as a method for inferring a species tree estimate, we have observed that more than a single (gene) tree may occur with the highest

frequency. To the best of our knowledge, this issue is not discussed in the literature; instead, researchers simply report using *the* democratic vote, mistakingly assuming its uniqueness. We have investigated two approaches to selecting a “representative” tree from the collection of all maximum-frequency gene trees:

- *Greedy consensus*: This amounts to taking the greedy consensus tree of all maximum-frequency trees.
- *Random*: This amounts to choosing a random tree from the collection of all maximum-frequency trees.

Given that the true species tree is known in simulations, we have also quantified the error rate in the “best” maximum-frequency gene tree (that is the tree that has the highest frequency and is closest to the true species tree in terms of topology) as well as the average error rate (averaging over all maximum-frequency gene trees). Notice that these two approaches are not applicable on biological data sets, since in this case the true species tree is unknown.

The results are plotted in Fig. 5.1. Clearly, using the greedy consensus of all maximum-frequency trees gives a more accurate estimate of the species tree than choosing a random one (on average). In the case of the 1Ne data, we observe that the greedy consensus is significantly better than randomly choosing a maximum-frequency tree. Further, we observe that choosing a maximum-frequency tree at random produce a species tree estimate that is much closer, in terms of topological accuracy, to the average than to the best maximum-frequency tree. This point indicates that there are many maximum-frequency trees, most of which are very far from the true species

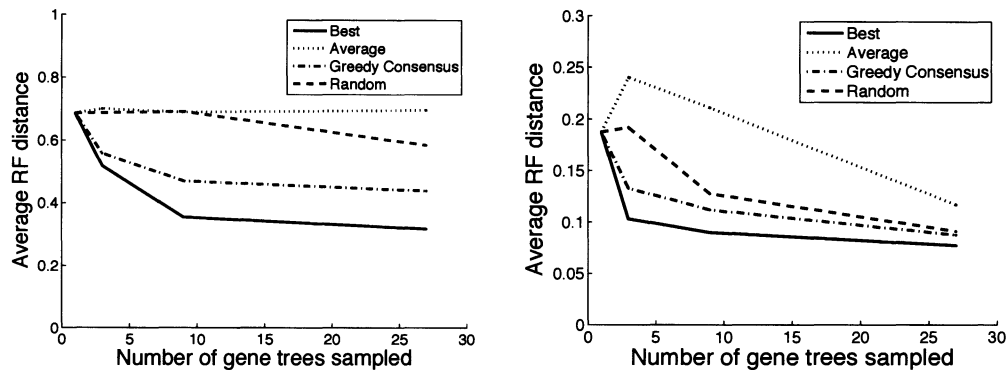


Figure 5.1: Results of the democratic vote (DV) method on the true gene trees of the 1Ne (left) and 10Ne (right) data sets (see text for description of the four curves).

tree topology.

In the case of the 10Ne data, the greedy consensus produces a more accurate species tree estimate than a random choice; however, in this case, the gain is not as significant, with both methods obtaining almost identical accuracy when all 27 gene trees are used. Further, in this case, both approaches produce trees that are very close in terms of accuracy to the best maximum-frequency gene tree. This is a reflection of the fact that, under these settings, the number of maximum-frequency gene trees is very small, and hence a random choice and the greedy consensus of all trees may produce very similar trees (identical trees in many cases in fact).

Based on these results, we recommend using the greedy consensus of all maximum-frequency gene trees as the DV tree.

GLASS

As stated above, the GLASS method [MR10] is a distance-based method that uses distance matrices, each computed from a single locus. In simulated data sets where the true gene trees, along with their coalescence times, are known, the method applies directly to those coalescence times. However, when the method is run on sequence data, distances have to be first estimated from the sequences. In our experiments, we computed pairwise distances under the model of Jukes-Cantor [JC69], since the sequences were evolved under this model. In this case, using all computed pairwise distances may negatively affect the performance of GLASS, since distances between very closely related sequences may be underestimated. We have investigated the performance of GLASS as “bottom” distances are removed for each locus; results are shown in Fig. 5.2.

The results in Fig. 5.2 clearly show that not removing bottom distances produce poor estimates of the species tree. Even worse, in the case of the 1Ne data sets, the accuracy of the method becomes worse as more loci are included in the analysis, which is rather surprising in light of the theoretical consistency results proved by Mossel and Roch [MR10].

Nonetheless, we observe that the performance of GLASS improves as bottom distances are removed, with the optimal performance achieved when the bottom 20% distances are removed for each locus (with the exception of the 10Ne, 1 allele data set, where removing the bottom 30% yielded the best results).

Therefore, we recommend removing the bottom 20% of distances computed for

each locus. It is worth mentioning, though, that these results were obtained on sequences of length 1000, evolved under the model of Jukes-Cantor [JC69], with distances computed under the same model as well. A more thorough investigation of the parameter space is needed in order to better study this issue and how it affects the performance of GLASS, since the consistency of the distance estimate is crucial to the performance of GLASS, as well as any other distance-based methods.

MDC

As discussed above, Than and Nakhleh [TN09] have recently proposed exact solutions for inferring species tree estimates under the MDC criterion. Further, they proposed a heuristic solution that considers only clusters of taxa that appear in the gene trees (that is, excludes any cluster of taxa that is not displayed by any of the gene trees). While this heuristic achieves several orders of magnitude in speedup (in fact, for data sets containing more than 15 taxa or so, it is infeasible to run the exact solution, since the number of clusters grows exponentially in the number of taxa), a question was left as to how the two compare in terms of the accuracy of the trees they compute. It is important to note that, while the exact solution is guaranteed to return a tree that is at least as good as the one computed by the heuristic in terms of the optimality criterion, this does not necessarily mean that the tree computed by the exact solution is more accurate.

Indeed, considering the results in Fig. 5.3 of both MDC solutions on the true gene trees, one observes that the error rate of the heuristic solution is slightly lower than

that of the exact solution on the 10Ne data sets. However, they are almost identical, and definitely within standard deviation from each other when true gene trees are used. Almost identical trends are observed when the solution is run on gene trees reconstructed using MP with strict consensus that are not necessarily binary (in Fig. 5.4). In light of these results, we recommend using the heuristic solution of MDC over the exact solution, and when the gene trees may contain polytomies, using the heuristic on non-binary trees is recommended.

Comparing all methods

We now turn to comparing the performance of all methods discussed above when run on the true gene trees (results in Fig. 5.5) and on the reconstructed gene trees (results in Fig. 5.6). In all these figures, we plot the error rate of methods, as measured by the distance measure of Robinson-Foulds [RF81], as a function of the number of loci sampled. Different panels correspond to different numbers of sampled alleles and/or different total branch length (1Ne and 10Ne).

Based on the above analyses, we have chosen the greedy consensus of all maximum-frequency gene trees as the implementation of the democratic vote method. While we have extended the applicability of the greedy consensus method to data sets with multiple alleles, we did not do so with the democratic vote method, since it is not as clear how, or whether, to extend it. Hence, the democratic vote method was applied only to single-allele data sets. While STEM is applicable when the true gene trees are used, we could not apply it to reconstructed gene trees, since they were not

guaranteed to be ultrametric (a requirement in the STEM implementation). When running GLASS on the sequence data, we eliminated the bottom 20% distances for each locus, per the observations made above. We used the heuristic implementation of MDC, and when run on reconstructed gene trees, we used the version that assumed unrooted (gene) trees that are not necessarily bifurcating.

In Fig. 5.5, we can see that, under the settings of our simulations, all methods exhibited trends of statistical consistency: the error rates dropped as more loci and/or alleles were sampled. STEM and GLASS, both with identical performance, produced the most accurate species trees, followed by MDC with time. Further, STEM and GLASS converge almost completely to the true species tree when 27 loci and at least 9 alleles are sampled. The error rates of the other methods, on the other hand, seem to plateau once nine loci are sampled, particularly in the 1Ne data sets. The observation that these three methods (STEM, GLASS, and MDC with time) outperform the other methods is not surprising, since these three methods make use of the topology and coalescence times of the genes, while the others make use only of the topology, and in this case, the estimates of coalescence times are accurate. As we describe below, when these times are inferred from the data, their inaccuracies significantly affect the performance of the methods that use them. Obviously, in the case when only a single locus and single allele is sampled, all methods perform identically. The performance of the democratic vote method was the worst when nine or more loci were sampled. The gap in performance among the methods seemed to close as more alleles were sampled per species. Further, this gap is much smaller in the case of the 10Ne data sets than the 1Ne data sets. While sampling more loci clearly improves the accuracy

of the methods, this improvement “slows down” with the number of loci sampled. If we denote by $\Delta RF_M(i, i + 1)$ the decrease in error rate of method M , as measured by the RF distance, when sampling 3^{i+1} loci, as opposed to 3^i loci (for $i = 0, 1, 2$), then we observe that $\Delta RF_M(0, 1) > \Delta RF_M(1, 2) > \Delta RF_M(2, 3)$, where M is any of the seven methods.

The trends observed when using the true gene trees change significantly when using gene tree estimates. Fig. 5.6 shows the performance of the greedy consensus, the democratic vote (for the single-allele case only), MDC, and GLASS on reconstructed gene trees.

While methods still exhibit trends indicating statistical consistency in these cases, MDC produces the most accurate species tree estimates under all combinations of number of alleles, number of loci, and total branch lengths. The greedy consensus performs almost similarly to MDC on gene tree estimates. Further, the accuracies of MDC and the greedy consensus do not seem to be affected when using gene tree estimates rather than the true ones, as can be seen from comparing the results from Fig. 5.5 and Fig. 5.6. A similar comparison across all three methods shows that the performance of GLASS (even under the best scenario of eliminating the bottom 20% distances for each locus) suffers significantly when using the gene tree estimates rather than the true ones. These two observations combined indicate that methods that use topology alone perform better than methods that rely on coalescence times when the gene trees may have error in them.

Finally, we show in Fig. 5.7 the running times of all methods as a function of the number of loci sampled. All tools were run on a desktop with an Intel Core 2 Duo,

2.40GH CPU, and 3 GB of RAM. Different panels correspond to different numbers of alleles sampled and/or different total branch lengths. While all methods complete the analyses in a reasonable amount of time, GLASS, the democratic vote, and the greedy consensus method were the fastest. This is not surprising as the first is distance-based and the latter two are simple summary statistics; the other methods, however, involve optimization criteria and extensive searches of the tree space. For the largest data sets (27 loci, and 27 alleles per species), STEM was by far the slowest. While the numbers of taxa, loci, and alleles affect the speed of the method, we believe the “complexity” of the data, owing to the species tree parameters (total branch lengths, etc.), may be the main factor affecting the speed of methods.

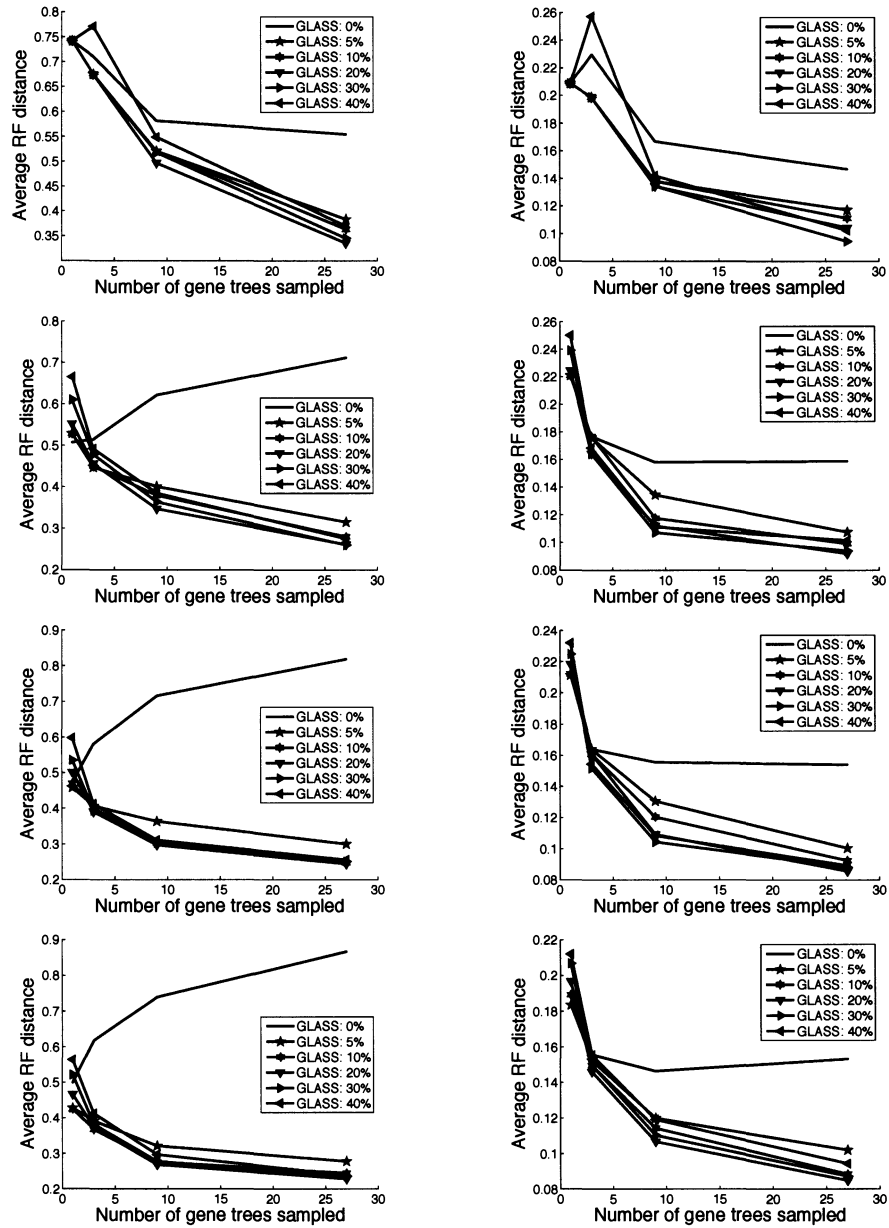


Figure 5.2: Performance of GLASS on the 1Ne (left column) and 10Ne (right column) data sets. Rows from top to bottom correspond to 1, 3, 9, and 27 alleles, respectively. Distances were computed under the Jukes-Cantor model, and for each locus, the bottom $x\%$ distances were removed, for $x \in \{0, 5, 10, 20, 30, 40\}$.

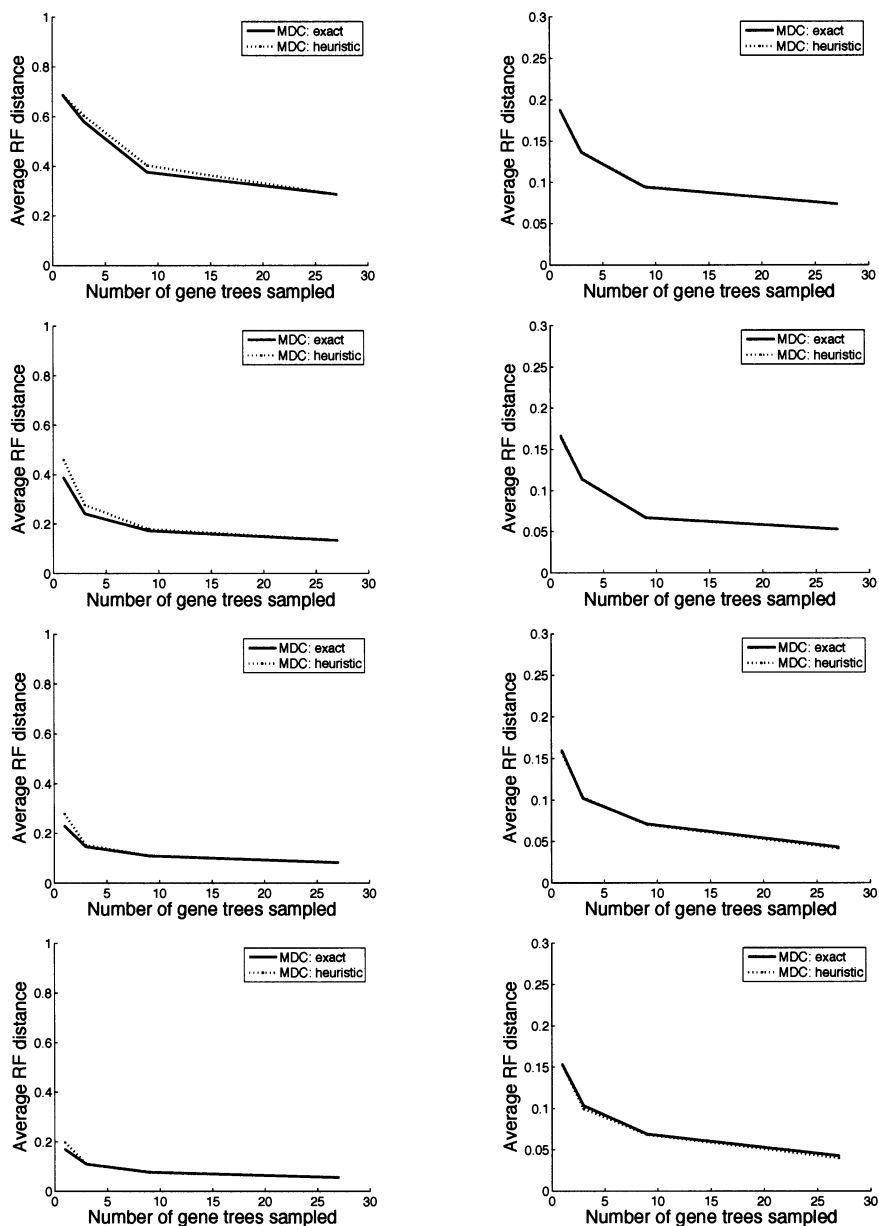


Figure 5.3: Results of the exact and heuristic solutions of MDC on 1Ne (left column) and 10Ne (right column) data sets, using the true gene trees. Rows from top to bottom correspond to 1, 3, 9, and 27 alleles, respectively.

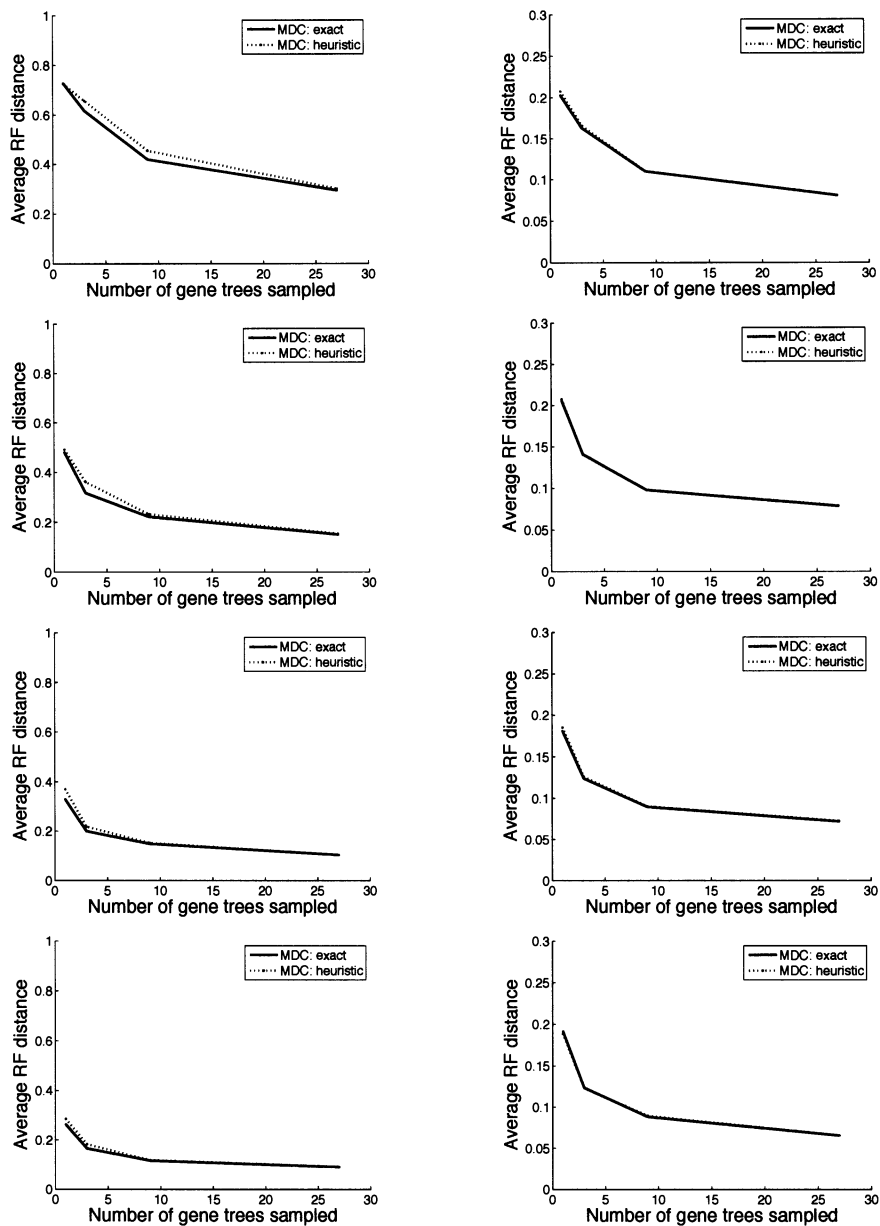


Figure 5.4: Results of the exact and heuristic solutions of MDC on 1Ne (left column) and 10Ne (right column) data sets, using the gene tree reconstructed by MP. Rows from top to bottom correspond to 1, 3, 9, and 27 alleles, respectively.

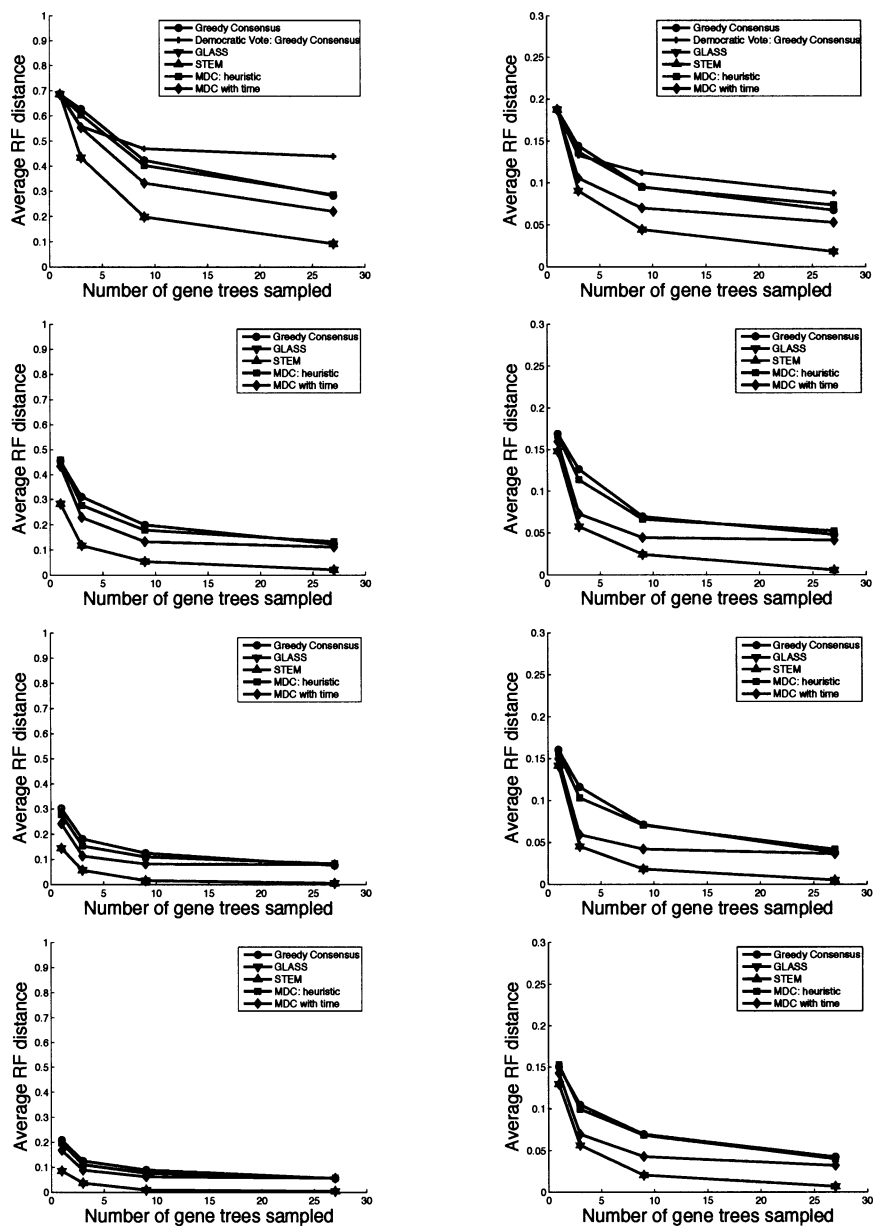


Figure 5.5: Performance of all methods on 1Ne (left column) and 10Ne (right column) data sets, using the true gene trees. Rows from top to bottom correspond to 1, 3, 9, and 27 alleles, respectively.

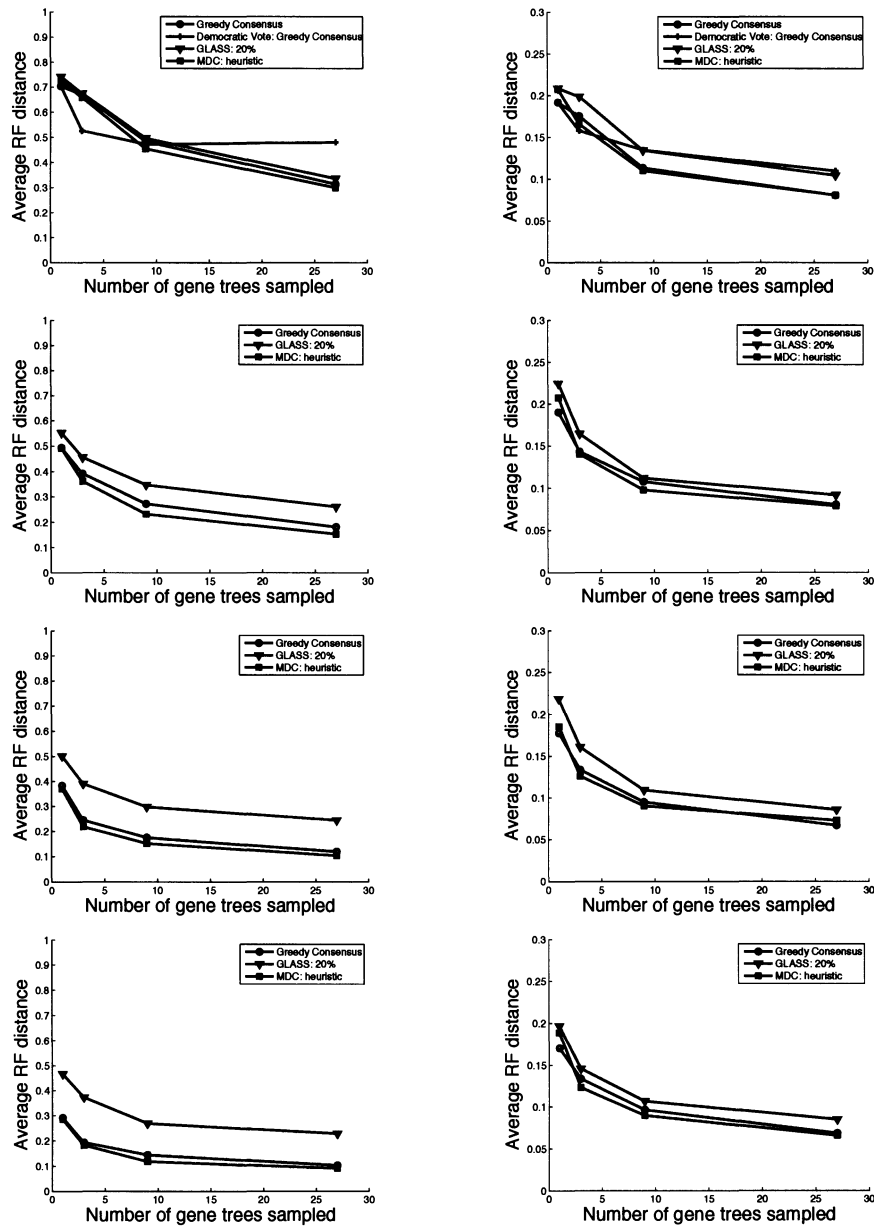


Figure 5.6: Performance of all methods on 1Ne (left column) and 10Ne (right column) data sets, using the gene trees reconstructed by MP. Rows from top to bottom correspond to 1, 3, 9, and 27 alleles, respectively.

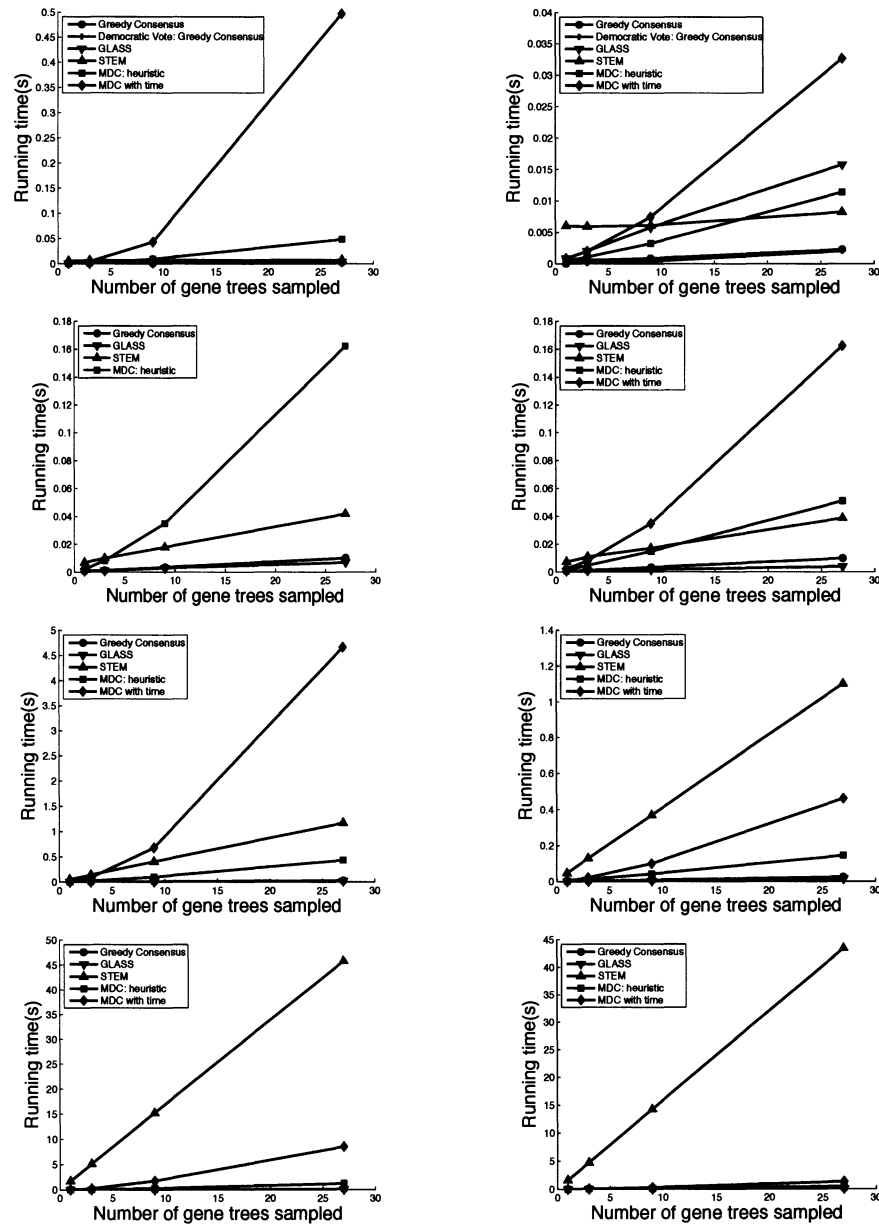


Figure 5.7: Running time of methods on 1Ne (left column) and 10Ne (right column) data sets, using the true gene trees. Rows from top to bottom correspond to 1, 3, 9, and 27 alleles, respectively.

Chapter 6

PhyloNet

PhyloNet [TRN08] (<http://bioinfo.cs.rice.edu/phylonet>) is a software package developed and maintained by the Bioinformatics Group at the Department of Computer Science at Rice University. As a toolkit of phylogenetics, it provides a suite of tools for efficient and accurate analysis of evolutionary phylogenies, such as detecting horizontal gene transfer from a pair of species and gene trees, detecting interspecific recombination breakpoints in a sequence alignment, and so on.

As stated above, many methods in this work have been implemented in PhyloNet for inferring species tree from a set of gene trees, including MDC, MDC with time, GLASS, democratic vote, and greedy consensus. The usage of MDC and MDC with time is described next.

MDC on rooted gene trees

Given a set of rooted gene trees, to infer species tree using MDC, we can use the following command in PhyloNet.

```
java -jar phylonet.jar infer_st -m MDC -i input [-e proportion]
[-x] [-b threshold] [-a mapping] [-ur] [-t time] [-o output]
```

The parameters in this tool include

- *-i input*: Specify the file that contains the input gene trees, which are rooted and not necessarily be binary. Also, gene losses are allowed.
- *-e proportion*: By default, the method returns the optimal species tree. But this option allows the users to get the optimal species tree and a set of sub-optimal ones. More precisely, if the optimal species tree has n extra lineages, all the sub-optimal species trees that have extra lineages less than $(1+proportion/100)*n$ will also be returned with the optimal one.
- *-x*: By default, the method uses clusters induced from gene trees to infer species tree. However, this option allows users to use all possible clusters instead.
- *-b threshold*: If the gene trees have bootstrap values, users can use this option to set the threshold. Then all the branches in the gene trees that have bootstrap values lower than this threshold will be contracted.
- *-a mapping*: If multiple alleles are sampled per species, this option can be used to include a file that contains the associations between gene tree taxa and

species tree taxa. For example, if alleles a_1 and a_2 in gene trees are sampled from species A , and alleles b_1 and b_2 in gene trees are sampled from species B , the following associations should be contained in the *mapping* file:

$$A : a_1, a_2;$$

$$B : b_1, b_2;$$

- *-ur*: If the gene trees are not binary and only clusters from gene trees are used to do the inference, the inferred species tree might also be non-binary, which triggers an exhaustive search for an optimal binary resolution of the species tree. If the inferred species tree has low degree of resolution, the search might take a large amount of time. So if non-binary species tree is tolerated, users can use this option, which indicates that non-binary species tree is allowed, to avoid the search.
- *-t time*: As mentioned above, if the inferred species tree is not fully resolved, searching the optimal binary resolution may take a large amount of time. Users can use this option to limit the search time to *time* minutes. In that case, when the time is reached, the optimal resolution obtained by that time will be returned.
- *-o output*: Users can use this option to save the result to a specified file.

MDC on unrooted gene trees

Given a set of unrooted gene trees, to infer the species tree using MDC, we can use the following command in PhyloNet.

```
java -jar phylonet.jar infer_st -m MDC_UR -i input [-e proportion]
      [-x] [-a mapping] [-b threshold] [-ur] [-t time] [-o output]
```

The parameters for this tool in PhyloNet are the same as those for MDC on rooted trees, which have been introduced in the previous section.

MDC with time

Given a set of rooted gene trees with branch lengths, to infer the species tree using MDC with time, we can use the following command in PhyloNet.

```
java -jar phylonet.jar infer_st -m MDC_TIME -i input [-a mapping]
      [-b bootstrap] [-o output]
```

The parameters for this tool in PhyloNet have also been introduced in the previous section.

Example

If we want to use MDC to infer the species tree from a set of rooted gene trees stored in file *gt.in*, which are listed below,

$$(((a1,b1),(a2,c2)),(b2,c1));$$

$$(((a1,a2),(b1,b2)),(c1,c2));$$

$$((((a1,b1),a2),b2),(c1,c2));$$

with a file *map.in* contains the following associations between alleles and species,

$$A:a1,a2;$$

$$B:b1,b2;$$

$$C:c1,c2;$$

we can use the following command in PhyloNet,

```
java -jar phylonet.jar infer_st -m mdc -i gt.in -a map.in
```

and PhyloNet will return the following inferred species tree

$$(C:1,(B:2,A:2):2):0; \text{ 7 extra lineages in total}$$

with the number of extra lineages on every branch shown after colons.

Chapter 7

Conclusions

In this work, we extended the MDC method [TN09] so that it applies to the cases where the gene trees are unrooted/binary, rooted/non-binary and unrooted/non-binary. Further, we proposed a new MDC method that takes into account not only the topology of the gene trees but also the coalescence times when they are available. In addition, we devise all MDC-based algorithms so that they can work on cases where multiple alleles per species are sampled.

We studied the performance of five methods for inferring species trees from multi-locus data sets, including democratic vote, greedy consensus, STEM, GLASS, and MDC. When the true gene trees and coalescence times were used, we found that the distance-based GLASS and likelihood-based STEM performed best in terms of accuracy. And when incorporating times into the MDC framework, the method performed only second to GLASS and STEM. However, when coalescence time estimates were used, this trend was reversed almost completely for GLASS compared to other methods. As for STEM, since the reconstructed gene trees were not rooted and violated

the molecular clock assumption, the tool could not be applied to the data. When gene tree estimates were used, the MDC method performed best, followed closely by the greedy consensus. Possibly not surprising was the fact that the performance of all methods was better on the 10Ne data sets than the 1Ne data sets; this is a reflection of the more extensive level of incongruence in the former compared to the latter. Nonetheless, all methods exhibited trends of statistical consistency under almost all conditions, in that their error rates decreased as more alleles and/or more loci were used in the analyses. The only exception to this observation is the democratic vote, when run on reconstructed gene trees with single allele per species; in this case, the error rate seems to plateau beyond nine loci. In terms of speed, MDC with time was the slowest, followed by STEM, with all the other methods being faster. However, the differences observed among the methods indicate that they can comfortably apply to larger data sets without the need for extensive computational resources. This is one advantage of all these tools over the Bayesian ones.

Finally, the PhyloNet software package [TRN08] now has extensive functionalities for species tree inference from multi-locus data, including greedy consensus, the democratic vote, GLASS, and all variants of MDC described above.

Bibliography

- [BFC00] M.A. Bender and M. Farach-Colton, *The LCA problem revisited*, Latin American Theoretical Informatics, 2000, pp. 88–94.
- [DAB⁺04] A. C. Driskell, C. Ané, J. G. Burleigh, M. M. McMahon, B. C. O’Meara, and M. J. Sanderson, *Prospects for building the tree of life from large sequence databases*, Science **306** (2004), 1172–1174.
- [Daw04] R. Dawkins, *The ancestor’s tale*, Houghton Mifflin, New York, 2004.
- [DDBR09] J. H. Degnan, M. DeGiorgio, D. Bryant, and N. A. Rosenberg, *Properties of consensus methods for inferring species trees from gene trees*, Syst. Biol. **58** (2009), 35–54.
- [DR06] J. H. Degnan and N. A. Rosenberg, *Discordance of species trees with their most likely gene trees*, PLoS Genet. **2** (2006), 762–768.
- [DR09a] ———, *Gene tree discordance, phylogenetic inference and the multi-species coalescent*, Trends Ecol. Evol. **24** (2009), 332–340.

- [DR09b] J.H. Degnan and N.A. Rosenberg, *Gene tree discordance, phylogenetic inference and the multispecies coalescent*, Trends in Ecology and Evolution **24** (2009), no. 6, 332–340.
- [DS05] J. H. Degnan and L. A. Salter, *Gene tree distributions under the coalescent process*, Evolution **59** (2005), 24–37.
- [Edw09] S. Edwards, *Is a new and general theory of molecular systematics emerging?*, Evolution **63** (2009), 1–19.
- [ELP07] S. V. Edwards, L. Liu, and D. K. Pearl, *High-resolution species trees without concatenation*, PNAS **104** (2007), 5936–5941.
- [HT84] D. Harel and R.E. Tarjan, *Fast algorithms for finding nearest common ancestors*, SIAM Journal on Computing **13** (1984), no. 2, 338–355.
- [Hud83] R. R. Hudson, *Testing the constant-rate neutral allele model with protein sequence data*, Evolution **37** (1983), 203–217.
- [JC69] T. H. Jukes and C. R. Cantor, *Evolution of protein molecules*, Mammalian Protein Metabolism (H. N. Munro, ed.), Academic Press, New York, 1969, pp. 21–132.
- [KCK09] L.S. Kubatko, B.C. Carstens, and L.L. Knowles, *STEM: species tree estimation using maximum likelihood for gene trees under coalescence*, Bioinformatics **25** (2009), no. 7, 971–973.

- [KD07] L. S. Kubatko and J. H. Degnan, *Inconsistency of phylogenetic estimates from concatenated data under coalescence*, Syst. Biol. **56** (2007), 17–24.
- [Kin82] J. F. C. Kingman, *The coalescent*, Stochastic Processes and Their Applications **13** (1982), 235–248.
- [KWK08] C-H. Kuo, J. P. Wares, and J. C. Kissinger, *The Apicomplexan whole-genome phylogeny: An analysis of incongruence among gene trees*, Mol. Biol. Evol. **25** (2008), no. 12, 2689–2698.
- [LP07] 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 **56** (2007), no. 3, 504–514.
- [LYK⁺09] L. Liu, L. L. Yu, L. Kubatko, D. K. Pearl, and S. V. Edwards, *Coalescent methods for estimating phylogenetic trees*, Mol. Phylogenet. Evol. **53** (2009), 320–328.
- [Mad97] W. P. Maddison, *Gene trees in species trees*, Syst. Biol. **46** (1997), 523–536.
- [MK06] W.P. Maddison and L.L. Knowles, *Inferring phylogeny despite incomplete lineage sorting*, Systematic Biology **55** (2006), no. 1, 21–30.
- [MM04] W.P. Maddison and D.R. Maddison, *Mesquite: A modular system for evolutionary analysis. Version 1.01*. <http://mesquiteproject.org>, 2004.

- [MR10] E. Mossel and S. Roch, *Incomplete lineage sorting: Consistent phylogeny estimation from multiple loci*, IEEE/ACM Transactions on Computational Biology and Bioinformatics **7** (2010), 166–171.
- [Nei86] M. Nei, *Stochastic errors in DNA evolution and molecular phylogeny*, Evolutionary Perspectives and the New Genetics (H. Gershowitz, D. L. Rucknagel, and R. E. Tashian, eds.), Alan R. Liss, New York, 1986, pp. 133–147.
- [Nei87] ———, *Molecular evolutionary genetics*, Columbia University Press, New York, 1987.
- [PIME06] 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 Genet. **2** (2006), 1634–1647.
- [PN88] P. Pamilo and M. Nei, *Relationships between gene trees and species trees*, Mol. Biol. Evol. **5** (1988), 568–583.
- [RF81] D.R. Robinson and L.R. Foulds, *Comparison of phylogenetic trees*, Math. Biosci. **53** (1981), 131–147.
- [Ros02] N. A. Rosenberg, *The probability of topological concordance of gene trees and species trees*, Theor. Pop. Biol. **61** (2002), 225–247.
- [RWKC03] A. Rokas, B.L. Williams, N. King, and S.B. Carroll, *Genome-scale approaches to resolving incongruence in molecular phylogenies*, Nature **425** (2003), 798–804.

- [RY03] B. Rannala and Z. Yang, *Bayes estimation of species divergence times and ancestral population sizes using DNA sequences from multiple loci*, Genetics **164** (2003), 1645–1656.
- [SS03] C. Semple and M. Steel, *Phylogenetics*, Oxford University Press, Oxford, 2003.
- [SWCL05] 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 **92** (2005), 2086–2100.
- [Swo95] D. L. Swofford, *PAUP*: Phylogenetic analysis using parsimony (and other methods)*, Sinauer Associates, Underland, Massachusetts, Version 4.0.
- [Taj83] F. Tajima, *Evolutionary relationship of DNA sequences in finite populations*, Genetics **105** (1983), 437–460.
- [Tak89] N. Takahata, *Gene genealogy in three related populations: consistency probability between gene and population trees*, Genetics **122** (1989), 957–966.
- [TN09] C. Than and L. Nakhleh, *Species tree inference by minimizing deep coalescences*, PLoS Computational Biology **5** (2009), no. 9, e1000501.
- [TN10] ———, *Inference of parsimonious species phylogenies from multi-locus data by minimizing deep coalescences*, Estimating Species Trees: Practical

and Theoretical Aspects (L.L. Knowles and L.S. Kubatko, eds.), Wiley-VCH, 2010, pp. 79–98.

- [TRN08] C. Than, D. Ruths, and L. Nakhleh, *PhyloNet: a software package for analyzing and reconstructing reticulate evolutionary relationships*, BMC Bioinformatics **9** (2008), 322.
- [TSIN08] C. Than, R. Sugino, H. Innan, and L. Nakhleh, *Efficient inference of bacterial strain trees from genome-scale multi-locus data*, Bioinformatics **24** (2008), i123–i131, Proceedings of the 16th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB ‘08).
- [Wu91] C.-I. Wu, *Inferences of species phylogeny in relation to segregation of ancient polymorphisms*, Genetics **127** (1991), 429–435.
- [Wu92] ———, *Reply to Richard R. Hudson*, Genetics **131** (1992), 513.
- [YWN11a] Y. Yu, T. Warnow, and L. Nakhleh, *Algorithms for MDC-based multi-locus phylogeny inference*, The 15th Annual International Conference on Research in Computational Molecular Biology (RECOMB) LNBI **6577** (2011), 531–545.
- [YWN11b] ———, *Algorithms for MDC-based multi-locus phylogeny inference: Beyond rooted binary gene trees on single alleles*, Journal of Computational Biology **18** (2011), 1543–1559.