INFERENCE OF PARSIMONIOUS SPECIES TREES FROM MULTI-LOCUS DATA BY MINIMIZING DEEP COALESCENCES

CUONG THAN AND LUAY NAKHLEH

Abstract. One approach for inferring a species tree from a given multi-locus data set entails computing a tree that optimizes a certain criterion. In 1997, W. Maddison proposed “minimizing deep coalescences”, or MDC, as one such criterion. This is a parsimonious criterion that, roughly speaking, seeks the tree that minimizes a quantity called *extra lineages* when all gene trees are reconciled within its branches. Recently, we developed the first exact algorithms for finding the optimal tree under the MDC criterion; these algorithms are guaranteed to find the tree (or trees) that minimizes the number of extra lineages when all input gene trees are reconciled within its branches. These exact algorithms, while taking time that is asymptotically exponential in the number of species under consideration, are capable of analyzing data sets of tens of species and thousands of loci in seconds or minutes.

In this chapter, we report on those algorithms, highlighting the computational techniques underlying their development, which we believe may be helpful in achieving algorithmic improvements in more compute-intensive criteria, such as maximum likelihood. Further, we show how these algorithmic solutions deal with gene trees that are not necessarily binary, and how to use them on data sets with multiple individuals. While guaranteeing the optimality of the solution entails considering all possible clusters of the taxa under study, we investigate the performance of those solutions when restricted to the set of clusters displayed by the gene trees. In practice, this latter set is much smaller than the set of all clusters, and hence it leads to drastic improvement in the speed of the methods, particularly for very large (in terms of the number of species) data sets.

We have implemented our algorithms in the PhyloNet software package, which is freely available at `http://bioinfo.cs.rice.edu/phylonet`. Using this implementation, we demonstrate the accuracy of trees inferred as well as the speed, on a large number of simulated data sets.

1. Introduction

The task of inferring phylogenetic relationships of species is commonplace in evolutionary biology, and its significance goes beyond evolutionary biology. The increasing availability of whole-genome sequences in general, and multi-locus data sets in particular, presents both an opportunity and a challenge for addressing this important task. On the one hand, using multiple loci generally gives a better signal for phylogenetic relationships. On the other hand, evidence of massive incongruence among the evolutionary histories of loci has given rise to a new level of complexity: while traditional phylogenetic approaches had to consider mutations at the nucleotide level, new approaches have to explicitly incorporate incongruence among the loci’s evolutionary histories as well.

Maddison proposed a parsimony-based criterion for inferring species trees from gene trees by minimizing the number of extra lineages, or minimizing deep coalesces (MDC) [5]. Maddison and Knowles implemented a heuristic approach based on this criterion [6]. However, no exact algorithms for computing the MDC criterion exist. We recently provided a two-stage heuristic for inferring the species tree under a similar criterion [15]. More recently, we provided a formal definition of the notion of extra lineages and the first exact algorithms—an integer linear programming (ILP) algorithm and a dynamic programming (DP) algorithm—for finding the optimal species tree topology from a set of gene tree topologies, under the MDC criterion [12]. We also demonstrated the accuracy and efficiency with which species trees were inferred using our exact algorithms on the yeast data set of [10], the *Apicomplexan* data set of [4], and a large number of data sets simulated under the coalescent model. However, all these analyses were conducted on data sets that contained only binary trees, and with exactly one individual sampled per species for each of the loci.

In this chapter, we review the concepts of compatibility graphs of clusters induced by a set of gene trees, as well as by the set of all clusters (i.e., all subsets) of a given set of taxa (Section 2). Further, we revisit the

Department of Computer Science, Rice University, Houston, Texas, USA. Email: {cvthan,nakhleh}@cs.rice.edu.

concept of *valid coalescent histories* [3], as it is the basis for defining the notion of *extra lineages* (Section 3). These two structures form the basis for our exact algorithms [12], which we review in Section 4. As the algorithms of [12] and the analyses therein assumed binary trees, with exactly a single individual per species, we describe how to extend the algorithms to non-binary gene trees and data sets in which multiple individuals may be sampled (Section 5).

We have implemented both exact algorithms in the PhyloNet software package [14]. We show the performance of the MDC criterion in Section 6. Using the simulation protocol and parameters of [6], we study the performance of the MDC criterion on a large number of 8-taxon simulated data sets, with numbers of loci and alleles taken from the set $\{1, 3, 9, 27\}$. The results of our study show that the criterion is accurate for inferring species trees, particularly as the numbers of loci used and individuals sampled increase, indicating patterns of statistical consistency under the simulation conditions used. Further, we consider in Section 7 the question of whether it is sufficient to consider only the clusters of taxa supported by the input gene trees for inferring an optimal tree under MDC. While in theory one can devise examples to show this is not necessarily the case, our simulation results indicate that under realistic conditions, the answer to this question is positive. Finally, we illustrate in Section 8 how to use PhyloNet to run the methods described in this chapter.

In summary, the chapter is organized into two main parts: the first part covers the technical details behind the methods (Sections 2–5), and the second part covers the performance, as well as illustrations of how to use the PhyloNet software package to run the methods described here (Section 6–8). The reader who may be interested mainly in the performance of the methods and how to run them, can skip the first part and go directly to the second.

The computational efficiency of our methods, coupled with the promising properties of the MDC criterion, makes our methods particularly applicable to large, genome-scale data sets.

2. Trees, Clusters, and the Compatibility Graph

Let T be a rooted phylogenetic tree on a set $\mathscr{X}$ of species taxa (species names); i.e., each leaf of T is uniquely labeled by exactly one element of $\mathscr{X}$. For example, trees T_1, T_2, and T_3 in Figure 1 are rooted phylogenetic trees on $\mathscr{X} = \{a, b, c, d, e\}$. We denote by $L(T)$ the set of leaves of T, and by $E(T)$ the set of internal edges of T, plus one edge, called re, incident into the root of T. If T is binary (i.e., every internal node has exactly two children), then $E(T)$ has $|L(T)| - 1$ edges[1]. For example, for tree T_1 in Figure 1, using the labelings of the leaves and edges, we have $L(T_1) = \{a, b, c, d, e\}$ and $E(T_1) = \{1, 2, 3, 4\}$.

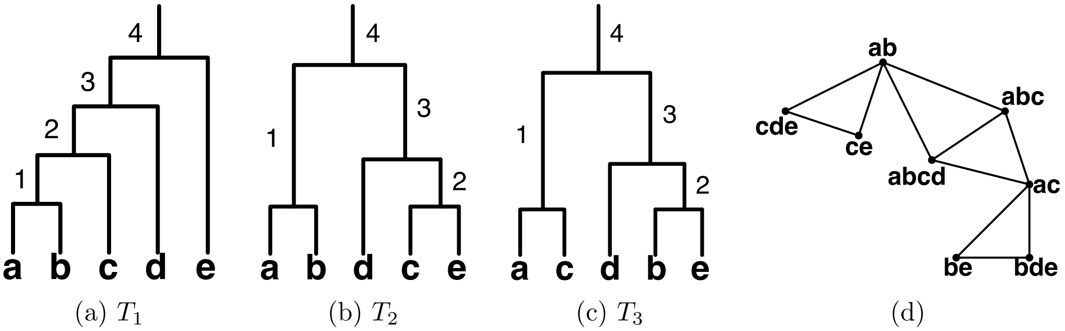
We assume in this chapter that the edges in $E(T)$ are labeled via a post-order numbering, which means that an edge e is labeled by a number that is larger than the numbers labeling the edges "under" (or, descendants of) e. Further, we assume that the edges emanating from the same node are labeled in increasing order from "left" to "right." See the labelings of T_1, T_2, and T_3 in Figure 1 for an illustration. Let us denote such a labeling by a bijection $h^T \colon E(T) \to \{1, \ldots, |E(T)|\}$ that satisfies the following two conditions: (1) for all $e_1, e_2 \in E(T)$, e_1 being a child edge of e_2, then $h^T(e_1) < h^T(e_2)$; and (2) for all $e_1, e_2 \in E(T)$, e_1 being a left child edge and e_2 being a right child edge of the same node, then $h^T(e_1) < h^T(e_2)$.

A cluster is defined as a nonempty subset of $\mathscr{X}$. A tree T defines a set $C_T = \{c_e^T : c_e^T \subseteq \mathscr{X},\ e \in E(T)\}$ of *induced clusters*, where c_e^T is the set of all leaf labels in $\mathscr{X}$ which are "under" edge e in T. Since each edge $e = (u, v)$ in a rooted tree is uniquely defined by its head v, c_e^T is equivalent to $c^T(v)$, which is the set of all leaves under node v. For example, we have $c_1^{T_1} = ab$[2] and $c_3^{T_3} = bde$, and the sets of all induced clusters for T_1, T_2, and T_3 in Figure 1 are $C_{T_1} = \{ab, abc, abcd, abcde\}$, $C_{T_2} = \{ab, ce, cde, abcde\}$, and $C_{T_3} = \{ac, be, bde, abcde\}$, respectively.

For a cluster A (i.e., a subset) of $\mathscr{X}$, we denote by $\mathrm{MRCA}_T(A)$ the *most recent common ancestor*, or MRCA (also known as the *least common ancestor*, or lca) of taxa in A in tree T. For two clusters A and B (each of which is a subset of taxa from $\mathscr{X}$), we say that they are *compatible* if there exists a tree T, leaf-labeled by $\mathscr{X}$, with two nodes u and v in T such that $c^T(u) = A$ and $c^T(v) = B$. In other words, A and B are compatible if there exists a tree that induces both of them. Mathematically, this is equivalent to

[1] $|A|$ denotes the cardinality of set A. For example, $|\{a, b, c\}| = 3$.

[2] In this context, a string $x_1 x_2 \cdots x_k$ denotes the set $\{x_1, x_2, \ldots, x_k\}$ of leaf labels. For example, the string ab corresponds to the cluster $\{a, b\}$.



(a) T_1 (b) T_2 (c) T_3 (d)

FIGURE 1. Three rooted phylogenetic trees T_1, T_2, and T_3, over the taxon-set $\{a, b, c, d, e\}$. The compatibility graph (d) that is built from clusters induced by the three trees. Each vertex of the graph corresponds to a cluster (a string next to it), and two vertices are adjacent (i.e., connected by an edge) if the two clusters they represent are compatible.

the condition that either $A \subseteq B$, $B \subseteq A$, or $A \cap B = \emptyset$. For example, the clusters $A = ab$ and $B = abcd$ are compatible, whereas the clusters $C = abc$ and $D = bcd$ are incompatible. A nonempty collection of pairwise-compatible clusters uniquely defines a rooted tree [8].

Given a collection $\mathscr{G}$ of trees, each of which is leaf-labeled by $\mathscr{X}$, we define the *compatibility graph* of $\mathscr{G}$, denoted by $CG(\mathscr{G}) = (V, E)$, where

- $V = \left(\bigcup_{g \in \mathscr{G}} C_g\right) \setminus \mathscr{X}$ and
- $E = \{(c_1, c_2)\colon c_1, c_2 \in V,\ c_1 \text{ and } c_2 \text{ are compatible.}\}$

In other words, the compatibility graph $CG(\mathscr{G})$ contains one node per cluster that is induced by at least one of the gene trees (each cluster appears only once, even if it is induced my multiple gene trees), and an edge connects two nodes in the graph if their corresponding clusters are compatible. For example, the graph in Figure 1(d) is the compatibility graph of the three given trees. Since the cluster $\mathscr{X}$ is compatible with every other cluster, we omit it from the compatibility graph.

It is worth mentioning that various properties of the compatibility graph may reflect certain characteristics of the tree data set $\mathscr{G}$. For example, if the number of nodes in $CG(\mathscr{G})$ is close to $|\mathscr{X}| - 2$, which is the number of internal edges of a binary tree on $\mathscr{X}$, then the amount of incongruence among the trees in $\mathscr{G}$ is very small—i.e., it may be an "easy" set of trees to reconcile. Further, each node in $CG(\mathscr{G})$ can be weighted by the number of trees in which the cluster appears. Then, a node with high weight and high connectivity indicates a cluster that is compatible with a large number of the trees in $\mathscr{G}$; from an inference perspective, such a cluster may be a good candidate to include in the species tree estimate that reconciles the trees in $\mathscr{G}$. Notice that each of the trees in $\mathscr{G}$ is a clique[3] in $CG(\mathscr{G})$.

The notion of a compatibility graph can be extended to all clusters of taxa in $\mathscr{X}$. More formally, the compatibility graph of $\mathscr{X}$, denoted by $CG(\mathscr{X})$, is the graph (V, E), where

- $V = \{c : c \subseteq \mathscr{X},\ c \neq \emptyset\}$ and
- $E = \{(c_1, c_2) : c_1, c_2 \in V,\ c_1 \text{ and } c_2 \text{ are compatible.}\}$

Given a set $\mathscr{X}$ of taxa, the compatibility graph $CG(\mathscr{X})$ contains $2^n - 1$ nodes, where $|\mathscr{X}| = n$. The set of all maximal cliques in $CG(\mathscr{X})$ corresponds to the set of all binary trees on $\mathscr{X}$. In other words, $CG(\mathscr{X})$ provides a compact representation of the much larger set of all trees leaf-labeled by $\mathscr{X}$ (there are $(2n-3)!!$ rooted binary phylogenetic trees on $\mathscr{X}$).

In the algorithms we describe below, $CG(\mathscr{X})$ needs to be used in order to guarantee optimality. However, we show empirically that, under the conditions we consider, $CG(\mathscr{G})$ is sufficient. This is a crucial observation for the feasibility of algorithms since in practice, for given sets of gene trees from biological data, the graph $CG(\mathscr{G})$ is much smaller than $CG(\mathscr{X})$. For example,

- $CG(\mathscr{G})$ has 17 vertices for the 106 gene trees in the 8-species yeast data set of [10]. For this data set, $CG(\mathscr{X})$ has 255 vertices;

[3] A clique in a graph $G = (V, E)$ is a subgraph (V', E') of G such that every two vertices in V' are adjacent.

- $CG(\mathscr{G})$ has 37 vertices for the 268 gene trees in the 8-species *Apicomplexan* data set of [4]. For this data set, $CG(\mathscr{X})$ has = 255 vertices; and
- $CG(\mathscr{G})$ has 36 vertices for the 1898 gene trees in the 9-strain *Staphylococcus aureus* data set of [15]. For this data set, $CG(\mathscr{X})$ has 512 vertices.

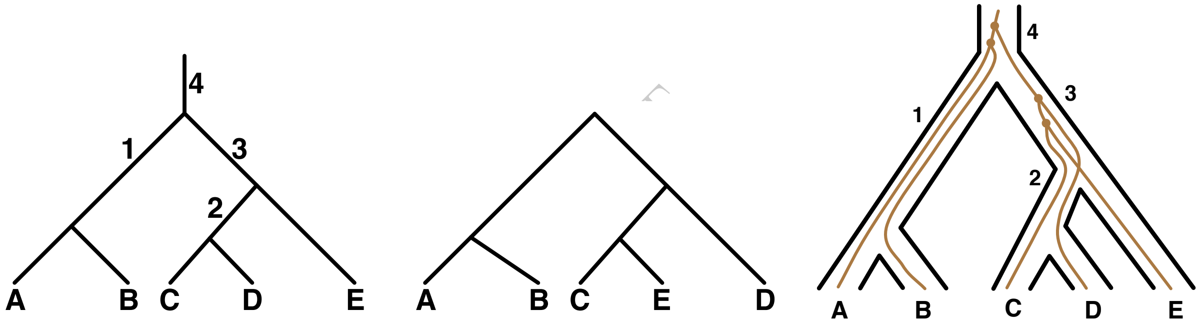
3. Valid Coalescent Histories, Extra Lineages, and the MDC Criterion

Under the coalescent model, the gene tree is viewed as a random variable conditional on the species tree that contains it. In this respect, given a gene tree topology g and a species tree topology Ψ, there is a finite, well-defined set of coalescent scenarios within the branches of Ψ, each of which can give rise to g—each such scenario is called a *valid coalescent history* [3]. The clades of leaves in g can potentially coalesce on any of the branches in Ψ provided that two conditions are satisfied:

C1: clade c of g coalesces on a branch that is on the path between the MRCA of c and the root of Ψ, or on the edge incident into the root of Ψ.

C2: the order in which the clades of g coalesce on the branches of Ψ is consistent with the topology of g.

For example, in Figure 2, clade $\{C, E\}$ in the gene tree g can coalesce on either branch 3 or branch 4 in the species tree Ψ, but cannot coalesce on either of the two branches 1 and 2. If both conditions are satisfied, then the coalescent scenario is considered a valid coalescent history. The set of all valid coalescent histories of a gene tree g, given a species tree Ψ, is denoted by $H_\Psi(g)$.



(a) Species tree Ψ (b) Gene tree g (c) $h \in H_\Psi(g)$

Figure 2. A species tree Ψ (a) and a gene tree g (b), with the branches of Ψ labeled in a post-order manner. (c) A valid coalescent history of g within the branches of Ψ, which corresponds to line 4 in Table 1. The solid circles indicate the coalescent events.

Let us consider the pair of species and gene trees, Ψ and g, respectively, in Figure 2. The tree g contains four clades (including the one that contains all leaves, and excluding all the ones that include only a single leaf): (A,B), (C,E), ((C,E),D), and ((A,B),((C,E),D)). The set of all valid coalescent histories of g within the branches of Ψ is given in Table 1. Each valid coalescent history is given by the set of branch labels in Ψ on which the clades of g coalesce.

We now define the concept of valid coalescent history formally.

Definition 1. *Given a species tree Ψ and a gene tree g, both leaf-labeled with set $\mathscr{X}$ of species taxa, a (valid) coalescent history is a mapping $\alpha : C_g \to E(\Psi)$, such that:*

1. *For every $X \in C_g$, $X \subseteq c^{\Psi}_{\alpha(X)}$, and*
2. *For every two edges $e_1, e_2 \in E(g)$, if e_1 is a child edge of e_2 then $h^{\Psi}(\alpha(e_1)) \leq h^{\Psi}(\alpha(e_2))$.*

Conditions (1) and (2) of Definition 1 correspond to conditions **C1** and **C2** above. We have recently presented an algorithm for enumerating the set $H_\Psi(g)$ for a given pair of trees, Ψ and g [13]. Further, Rosenberg [11] presented a different techniques for enumerating this set.

Table 1. The set $H_\Psi(g)$ of all valid coalescent histories of the gene tree g given the species tree Ψ, both of which are shown in Figure 2. Coalescent history (4) is illustrated in Figure 2(c). The rightmost column provides the number of extra lineages arising from each coalescent history.

Coalescent history	Clades of the gene tree				Number of extra lineages
	(A,B)	(C,E)	((C,E),D)	((A,B),((C,E),D))	
1	1	3	3	4	1
2	1	3	4	4	2
3	1	4	4	4	3
4	4	3	3	4	2
5	4	3	4	4	3
6	4	4	4	4	4

Given a valid coalescent history of a gene tree g within the branches of species tree Ψ, we can count the number of extra lineages[4] resulting from this coalescence history. This is the sum, over all branches in Ψ, of the number of lineages in a branch minus 1. For example, let us consider coalescent history (4) from Table 1, which is illustrated in Figure 2(c). This history contributes one extra lineage on branch 1, one extra lineage on branch 2, no extra lineages on branch 3, and no extra lineages on branch 4—a total of two extra lineages. We denote by $XL(\Psi, g)$ the minimum number of extra lineages contributed by a valid coalescent history for a gene tree g, where the minimum is taken over all histories in $H_\Psi(g)$. The rightmost column in Table 1 gives the number of extra lineages contributed by each of the six valid coalescent histories. Valid coalescent history 1 contributes the smallest number of extra lineages; hence, $XL(\Psi, g) = 1$, for the trees Ψ and g shown in Figure 2. We can generalize this notion to a set $\mathscr{G}$ of gene trees as

$$XL(\Psi, \mathscr{G}) = \sum_{g \in \mathscr{G}} XL(\Psi, g). \tag{1}$$

The cardinality of $H_\Psi(g)$ can be exponential in the size of the trees, which implies that a brute-force approach to computing $XL(\Psi, \mathscr{G})$, by which all valid coalescent histories are enumerated explicitly and the one resulting in the minimum number of extra lineages is selected, is infeasible. We review below an efficient algorithm that we [12] have developed recently for computing this quantity.

Finally, we are in position to define the MDC-T problem—the problem of inferring the species tree from a set of gene trees under the *minimize deep coalescence*, or MDC, criterion.

Definition 2. *(The MDC-T Problem)*

Input: *Set $\mathscr{G}$ of gene trees.*
Output: $\Psi^* = \text{argmin}_\Psi XL(\Psi, \mathscr{G})$.

Maddison and Knowles [6] developed a heuristic for solving the MDC-T problem. However, the heuristic is not guaranteed to compute the optimal solution. We have recently developed the first exact algorithms to the problem and demonstrated their efficiency, showing that they runs in seconds on data sets with tens of taxa and thousands of loci [12]. Further, both in [6] and [12], the performance of MDC as a criterion for inferring the species tree was shown to be very good on both biological and synthetic data sets. In the next two sections, we give the details of how to solve the MDC-T problem exactly.

4. Exact Algorithms for the MDC Problem

In this section we review two algorithms that we [12] developed recently for solving the MDC-T problem exactly; i.e., given a set $\mathscr{G}$ of gene trees, both algorithms find a tree Ψ^* such that $XL(\Psi^*, \mathscr{G})$ is minimum over all possible trees Ψ. These two algorithms make use of the central observation we made in [12] that the number of extra lineages on each branch of a given species tree topology can be computed independent of the other branches (Lemma 1 in [12]). Further, we have the following result.

[4] Intuitively, an extra lineage results from the failure of two lineages to coalesce on a branch in the species tree, resulting in two lineages "exiting" that branch, instead of a single lineage, which would have been the case had they coalesced.

Theorem 1. *(From [12]) Let (u, v) be a branch in the species tree Ψ, and let $t_1, \ldots, t_k$ be all the maximal clades of gene tree g such that $L(t_i) \subseteq c_\Psi(v)$ for $1 \leq i \leq k$. Then, the number of extra lineages on branch (u, v) is $\beta(c_\Psi(v), g) = k - 1$.*

For example, let $((A, B), C)$ be a clade in the species tree Ψ and let $c_1 = (A, (B, C))$ be a clade in gene tree g_1. There is only one maximal clade of g_1 that satisfies the condition of Theorem 1; hence, the number of extra lineages is 0. Indeed, in a parsimonious reconciliation, all three alleles from A, B, and C would coalesce on the branch above the clade $((A, B), C)$, and that branch would not have any extra lineages.

Now, consider clade $((A, B), (C, D))$ in gene trees g_2. Based on Theorem 1, there are two maximal clades that satisfy the theorem's condition: clade (A, B) and clade (C). Therefore, the number of extra lineages is 1. Indeed, since C coalesces with D before their MRCA coalesces with the MRCA of A and B, the branch over the clade $((A, B), C)$ in the species tree would have an extra lineage when reconciling g_2 within the branches of Ψ using MDC.

As we will show below, using Theorem 1, it is possible to compute the minimum number of extra lineages without explicitly searching the species tree space and computing the number of extra lineage for each tree in the space. In other words, Theorem 1 allows us to avoid the search that is implemented in the heuristic of Maddison and Knowles [6]. We are now in position to describe both exact algorithms for the MDC-T problem.

4.1. **An Integer Linear Programming Algorithm.** *Linear programming* (LP) is an algorithmic technique for maximizing or minimizing a *linear* objective function, cx, where c is a vector of coefficients and x is a vector of variables, subject to a set of *linear* constraints $Ax \leq b$, where A is a matrix of coefficients and b is a vector of coefficients. This is usually written in the form

$$\text{max/min: } cx,$$
$$\text{subject to: } Ax \leq b.$$

When the variables x are required to be integers, the problem becomes an *integer linear programming* (ILP). Solving an ILP problem is NP-hard in general, yet powerful solvers for (I)LP problems exist, both in open source and commercially. We now show how to use ILP to optimize the MDC criterion. Using the compatibility graph $CG(\mathscr{X})$ of a set of taxa $\mathscr{X}$, we want to find the maximal clique of $CG(\mathscr{X})$ that solves the MDC-T problem.

Constructing the weighted compatibility graph. Since we seek a clique in $CG(\mathscr{X})$ that is maximal in terms of size (so as to obtain as resolved a tree as possible) and minimal in the number of extra lineages resulting from reconciling the trees of $\mathscr{G}$ within its branches, we assign weights to the vertices of $CG(\mathscr{X})$ in a special way. Let v be a vertex in the graph $CG(\mathscr{X})$ (i.e., v corresponds to a subset of $\mathscr{X}$) and let A be the cluster it represents. For each gene tree $g \in \mathscr{G}$, we count the number of extra lineages contributed by A as described in Theorem 1. In total, having cluster A in the species tree Ψ results in $\sum_{g \in \mathscr{G}} \beta(A, g)$ extra lineages. Let m be the maximum value of $\sum_{g \in \mathscr{G}} \beta(A, T)$ over all possible clusters A. We assign to vertex v the weight

$$w(v) = m + 1 - \sum_{g \in \mathscr{G}} \beta(A, g). \tag{2}$$

Roughly speaking, the weight of a vertex v corresponds to a quantity that reflects the number of extra lineages arising from having the cluster associated with v in the species tree. In the next paragraph, we further elaborate on this weighting scheme.

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

We seek a maximum vertex-weighted clique in the compatibility graph $CG(\mathscr{X})$. This clique is clearly a maximal one, because each vertex is assigned a positive weight by function $w(v)$, which guarantees having the maximum number possible of compatible clusters in the species tree. Moreover, because we maximize the clique weight, by the definition of function $w(v)$, we in fact minimize the total number of extra lineages (among all cliques of the same size).

Let V and E be the sets of nodes and edges, respectively, in $CG(\mathscr{X})$. Then, finding a maximal vertex-weighted clique in a graph can be converted to a linear programming formulation as follows [12]:

$$\begin{aligned} \text{max: } & \sum_{v\in V} w(v)x_v, \\ \text{subject to: } & x_u + x_v \leq 1, \forall (u,v) \notin E, \\ & x_v \in \{0,1\}, \forall v \in V. \end{aligned} \tag{3}$$

Using an ILP solver on this formulation results in an exact algorithm to the MDC-T problem.

As discussed in Section 2, the size of $CG(\mathscr{G})$ is often much smaller than that of $CG(\mathscr{X})$, which results in a drastic improvement in the size of the ILP formulation, and, in turn, the actual computing time when solving the problem. We showed in [12] how to improve the ILP formulation when $CG(\mathscr{G})$, rather than $CG(\mathscr{X})$, is used, given that $CG(\mathscr{G})$ is very sparse in practice. Notice that, in theory, the exact algorithm to the MDC-T problem is guaranteed only if $CG(\mathscr{X})$ is used. In practice, $CG(\mathscr{G})$ may be sufficient, as we show below.

4.2. **A Dynamic Programming Algorithm.** Dynamic programming (DP) is a divide-and-conquer algorithmic technique that breaks a problem into sub-problems, solves the sub-problems, and then uses those solutions in an efficient way to form the solution to the main problem. For a problem to be amenable to a DP solution, it has to exhibit certain properties. For more details, the reader is referred to any standard textbook on algorithms; e.g., [1]. We now describe how to solve the MDC-T problem using a DP algorithm.

Let t be a rooted binary tree, with $L(t) = A \subseteq \mathscr{X}$. Given a collection $\mathscr{G}$ of gene trees, we denote by $\ell(t,\mathscr{G})$ the sum of $\sum_{g\in\mathscr{G}} \beta(B,g)$ for all clusters B in t, including A. Further, we denote by $\ell^*(A,\mathscr{G})$ the minimum value of $\ell(t,\mathscr{G})$ over all possible binary trees t on A. If t_1 and t_2 are the two subtrees whose roots are the children of t, then clearly we have

$$\ell(t,\mathscr{G}) = \ell(t_1,\mathscr{G}) + \ell(t_2,\mathscr{G}) + \sum_{g\in\mathscr{G}} \beta(A,g).$$

The quantity $\sum_{g\in\mathscr{G}} \beta(A,g)$ is fixed for each A, and therefore, if t is an optimal tree on A such that $\ell(t,\mathscr{G})$ is minimum, then $\ell(t_1,\mathscr{G})$ and $\ell(t_2,\mathscr{G})$ must also be minimum. This allows us to compute $\ell^*(A,\mathscr{G})$ recursively as follows.

(1) Let $\mathscr{C}$ be a collection of all non-empty subsets of $\mathscr{X}$. We partition $\mathscr{C}$ into subsets $\mathscr{C}_1, \ldots, \mathscr{C}_{|X|}$, where $\mathscr{C}_i$, $1 \leq i \leq |\mathscr{X}|$, is the collection of all clusters of size i in $\mathscr{C}$.
(2) For every $A \in \mathscr{C}_1$, $\ell^*(A,\mathscr{G}) = 0$, and for $A \in \mathscr{C}_2$, $\ell^*(A,\mathscr{G}) = \sum_{g\in\mathscr{G}} \beta(A,g)$.
(3) For $A \in \mathscr{C}_i$, $3 \leq i \leq |\mathscr{X}|$,

$$\ell^*(A,\mathscr{G}) = \min\{\ell^*(A_1,\mathscr{G}) + \ell^*(A_2,\mathscr{G})\colon A_1 \cap A_2 = \emptyset \text{ and } A = A_1 \cup A_2\} + \sum_{g\in\mathscr{G}} \beta(A,g).$$

(4) Return $\ell^*(\mathscr{X},\mathscr{G})$.

Although the algorithm described above only returns the number of extra lineages, we can easily modify it so that we can actually reconstruct the optimal species tree. For each i, $3 \leq i \leq |\mathscr{X}|$, in Step 3, we also record two pointers to optimal subclusters A_1 and A_2. By backtracking those pointers starting with cluster $\mathscr{X}$, we can obtain the optimal set of compatible clusters.

The running time of the algorithm is bounded by $\sum_{i=0}^{|\mathscr{X}|} \binom{|\mathscr{X}|}{i} 2^i = 3^{|\mathscr{X}|}$. Although this is exponential, it is significantly better than a brute-force approach that examines all $(2|\mathscr{X}|-3)!!$ binary rooted phylogenetic trees on $\mathscr{X}$. However, it is important to note that this bound drops significantly when only the clusters of the input gene trees are considered; i.e., when the nodes of the compatibility graph $CG(\mathscr{G})$ are considered. Further, we expect that as the problem size increases, in terms of the numbers of taxa, loci, and alleles, the ILP algorithm may have an advantage over the DP algorithm, owing to the power of the commercial (and some non-commercial) ILP solvers.

5. Handling Special Cases

Thus far, we have discussed the MDC criterion and presented algorithms for the MDC-T problem only for the case when the trees in $\mathscr{G}$ are all binary, and have exactly a single individual (or, allele) per locus per species. We now discuss how the criterion, and algorithms, extend to the cases where multiple individuals per species may be sampled, and when the trees are not necessarily binary.

5.1. **Multiple Individuals Per Species.** Suppose that we sample more than one individual per species when reconstructing a gene tree. We can extend the MDC criterion as follows. All taxa in the gene trees are considered distinct, even if they are from the same species. When fitting the gene tree into the species tree, we simply draw as many lineages originated backwards from a species as the number of individuals sampled for that species, and the remaining process is carried out in a similar manner as in [5]. For instance, consider the species tree and gene tree in Figure 3. There are three species A, B and C, and for species A, we sample two individuals, represented as A_1 and A_2 in the gene tree. Because we sample two individuals for A, there are two lineages within the branch incident with the leaf A. As we trace the evolution backwards in time, we find that A_1 coalesces first with C, then with A_2, and finally with B. All of those coalescence events occur on the branch incident into the root of the species tree. For this example, there is one extra lineage on the branch incident with the leaf A, and two extra lineages on the branch (u, v), accounting for a total of three extra lineages.

In [12], we showed how to count the number of extra lineages for each cluster. The counting method applies to the case of multiple individuals in a straightforward manner, with the convention that individuals of the same species are mapped to the same species. For the trees in Figure 3, the number of extra lineages for cluster A is $2 - 1 = 1$ because there are two subtrees of the gene tree, (A_1) and (A_2), such that their leaf sets are subsets of $\{A\}$ (note here that both A_1 and A_2 are mapped onto A). Similarly, the number of extra lineages for cluster AB is $3 - 1 = 2$.

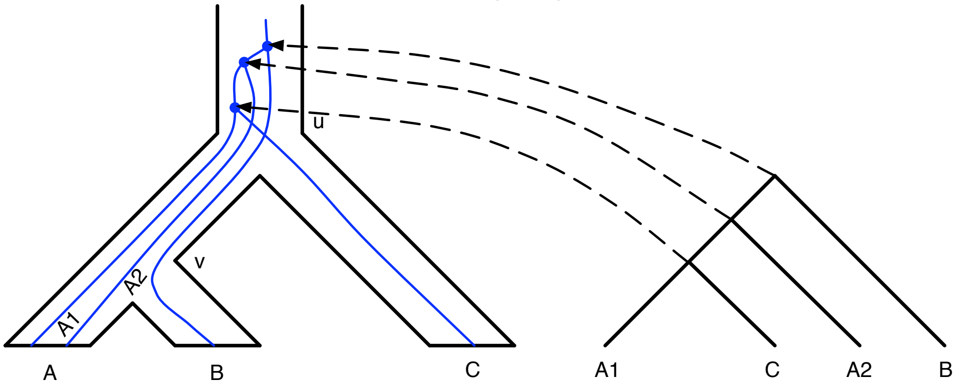


Figure 3. MDC for gene trees with multiple alleles/individuals. On the left, the species tree is shown in tubes, while the thin lines show how the gene tree, on the right, is fitted within the branches of the species tree. On the right, a gene tree with four leaves, two of which correspond to two individuals of species A.

5.2. **Non-binary Trees.** The extension of the MDC criterion for non-binary trees is quite straightforward. A non-binary node (a node with out-degree higher than 2) in the gene tree indicates that the lineages in the subtree rooted at that node all coalesce together. Fitting a gene tree into a species tree can be carried out in exactly the same way as in [5]. Figure 4 provides an illustration. Here, lineages from A, B, and D fail to coalesce along the branch (u, v), resulting in $3 - 1 = 2$ extra lineages on that branch. We note here that a non-binary node in the species tree does not affect the way we count the number of extra lineages on the branch incident into it. In this example, we have a node with out-degree 3 in the species tree corresponding to the cluster $\{A, B, D\}$. In the gene tree, we have exactly three subtrees (A), (B) and (D) such that their leaf sets are subsets of $\{A, B, D\}$.

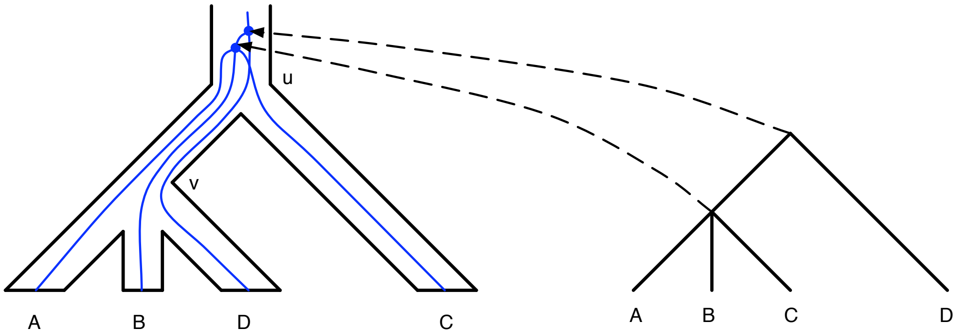


FIGURE 4. MDC for non-binary trees. On the left, the species tree is shown in tubes, while the thin lines show how the non-binary gene tree, on the right, is fitted within the branches of the species tree.

6. Performance of MDC

We have implemented our methods in the PhyloNet software package [14] and analyzed two biological data sets [4, 10] and a large set of simulated data sets. Analyses showed very good performance on all these data sets, as we reported in [12]. However, all these analyses involved a single individual per species. In this section, we study the performance of MDC on simulated data sets, when multiple alleles per species are sampled. To generate the data sets, we used the Mesquite tool [7] and the same procedure and parameters that Maddison and Knowles used in [6].

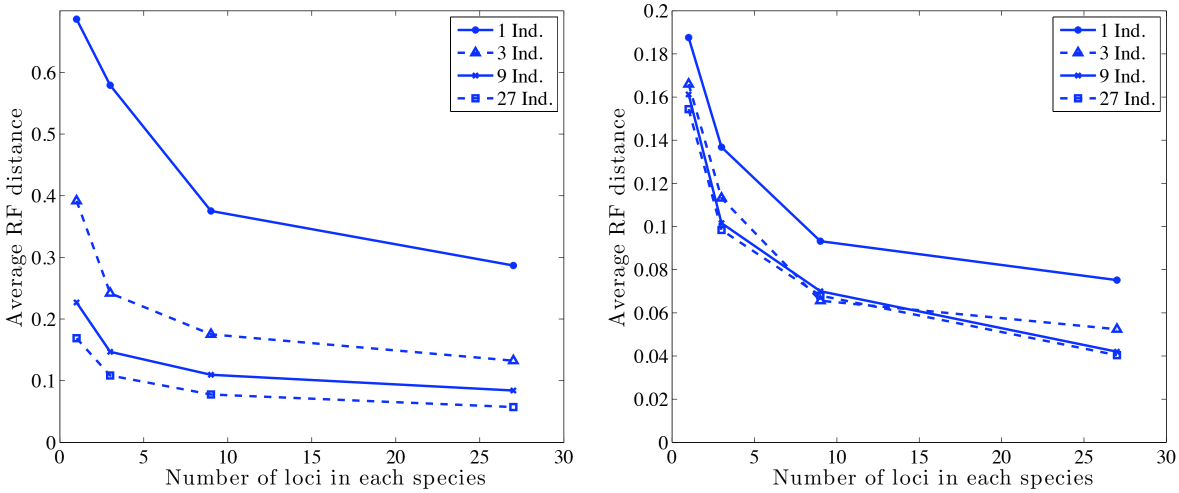
Species trees were simulated by using the "Uniform Speciation" (Yule) module in Mesquite. Two sets species trees were generated: one for those with a total branch length of 100,000 generations, and one for 1,000,000 generations. Each data set has 500 species trees. Within the branches of each species 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.

Since the species tree is known for simulated data, we studied the performance of MDC by comparing the inferred species tree against the true species tree. For this comparison, we used the normalized Robinson-Foulds (RF) measure [9], which quantifies the average proportion of branches present in one, but not both, of the trees. A value 0 of the RF distance indicates the two trees are identical, and a value of 1 indicates the two trees and completely different (they disagree on every branch).

We inferred the species trees using the exact algorithms to the MDC-T problem, that we described above, while using *all* clusters of taxa, to guarantee optimality (we will show below the results when only clusters exhibited by the input gene trees were used). Since the study includes eight taxa, there were $2^8 - 1 = 255$ clusters, which were used as the basis for inference of the species trees.

Figure 5 shows the normalized RF distance between the inferred species tree and the true one. Clearly, for a given number of loci and alleles, the performance of MDC is better for the case of deep divergence (total branch length of $10N_e$ than the case of recent divergence (total branch length of $1N_e$). However, the difference in performance shrinks as the number of individuals sampled increases. For example, when only a single individual is sampled per species and a single locus is used, MDC has an error rate of about 19% in the case of deep divergence, whereas it has an error rate of about 70% in the case of recent divergence. However, this gap closes as the number of individuals and number of loci increase.

In general, we observe the MDC's performance improves as the number of loci and individuals increases, regardless of the level of divergence. However, in the case of recent divergence, we observe that increasing the number of individuals yields a higher gain in performance than an increase in the number of loci (see also [6]). Further, under this divergence, the gain from increasing the number of loci becomes much smaller as the number of individuals sampled is larger. For example, for the case of 27 individuals, there is hardly any gain from increasing the number of loci from 9 to 27. Nonetheless, it is expected, based on mathematical properties of gene genealogies under the coalescent model (e.g., see [16]) that an increase in the number of individuals beyond a certain number ℓ would seize to have an effect on the accuracy of the inferred species



(a)

(b)

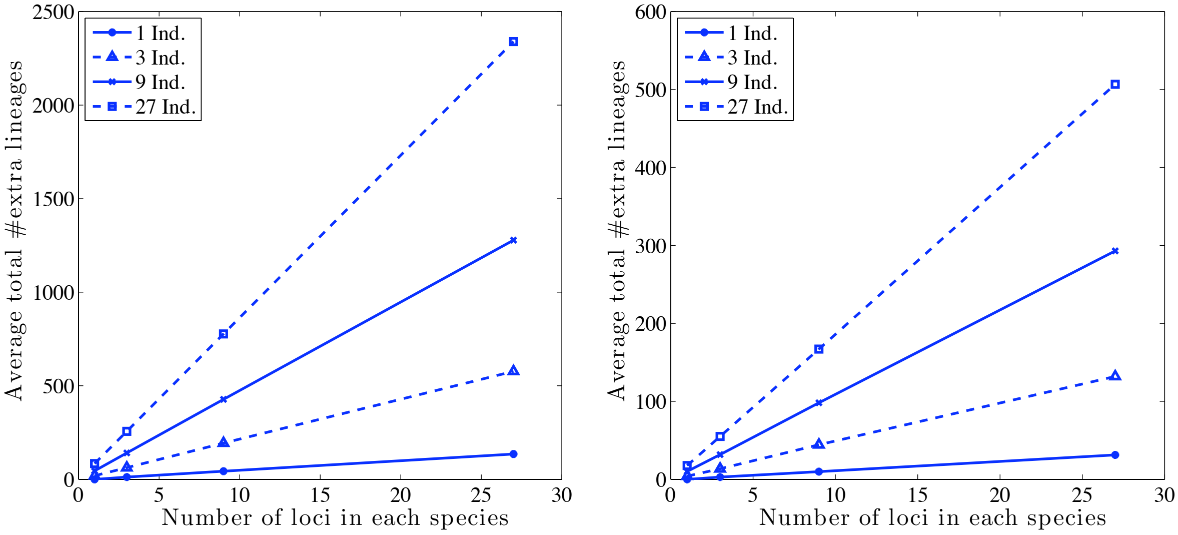
FIGURE 5. Accuracy of the inferred species tree as measured by the Robinson-Foulds distance when all clusters (there are $2^8 - 1 = 255$ of them) are used. (a)Recent divergence (total branch length is $1N_e$); (b) Deep divergence (total branch length is $10N_e$). We note that the y-axes in (a) and (b) are on different scales to make the difference between the curves more visible.

tree. For example, observe in Figure 5(b) that increasing the number of individuals beyond three does not have much of an impact on the accuracy of the species tree estimate. We expect that the value of ℓ would be smaller for larger total branch lengths. However, a thorough study of this observation is imperative, and is beyond the scope of this chapter.

It is important to note that when a single gene tree is used as the input to MDC, the method returns a species tree that is identical to the gene tree, since that is the tree with the minimum (zero, in this case) number of extra lineages. We observe that the performance, in the case of a single locus and single individual, is much better in the deep divergence case—this is simply because the gene tree in this case has a smaller degree of incongruence with the species tree. However, even in the case of recent divergence, using only one locus but with increasing the number of alleles from 1 to 27, results in a drastic improvement in performance. Last but not least, Figure 5 indicates statistical consistency of MDC under the simulation conditions.

The amount of incongruence in a data set may be reflected in the optimal number of extra lineages required to reconciled all the gene trees within the branches of a species tree, over all possible species tree. Figure 6 shows the average number of extra lineages required to reconcile all gene trees in the input within the branches of the optimal (under MDC) tree. Clearly, the average number of extra lineages is much smaller in the case of deep divergence—we would expect much less incongruence in this case than in the case of recent divergence. Further, we observe that for small numbers of individuals, the increase in the number of extra lineages is much slower than for the case of large numbers of individuals. This indicates that a large extent of the incongruence is caused by the multiplicity of individuals, rather than from the size of the set $\mathscr{G}$ of gene trees. This has a practical implication on the running time of inference methods: when analyzing genome-scale data sets, the number of loci, particularly for small numbers of individuals, may not be the crucial factor affecting the performance (in terms of time and memory requirements) of the inference method.

Finally, it is important to note that our method took about a second to infer the optimal tree on the largest data set (27 loci and 27 individuals), and shorter than that for the smaller data sets. Our method took seconds on the 106-locus yeast data set of [10] as well as the 268-locus *Apicomplexan* data set of [4]. Further, in the simulation study we conducted in [12], our method took seconds on data sets including 2000 loci.



(a) (b)

Figure 6. Average numbers of extra lineages required to reconcile the inferred species tree and gene trees when all clusters (there are $2^8 - 1 = 255$ of them) are used for the inference. (a) Recent divergence (total branch length is $1N_e$); (b) Deep divergence (total branch length is $10N_e$). We note that the y-axes in (a) and (b) are on different scales to make the difference between the curves more visible.

7. Inference From the Clusters of the Gene Trees

As the algorithms presented above require consideration of all clusters (subsets) of taxa in set $\mathscr{X}$ to guarantee optimality of the solution, the actual running time of the methods increases exponentially with the number of taxa. While these algorithms take seconds on data sets with 10 taxa, for example, this time would increase significantly when analyzing 20 taxa, and it becomes prohibitive for larger data sets.

A central question that we considered in [12] and revisit here is: do we need to consider all subsets of $\mathscr{X}$ or is it sufficient to consider only the subsets of $\mathscr{X}$ that appear as clusters in the gene trees in input set $\mathscr{G}$. The motivation behind considering this question is two-fold. First, given that, under the coalescent model, the gene tree is a random variable conditional on the species tree, gene trees are expected to contain the signal for the phylogenetic relationship of the species. Notice that, under certain conditions, this premise may not hold; e.g., [2]. We begin by showing that, in theory, using only the gene tree clusters is not sufficient. For example, consider the three gene trees in Figure 1. For these three trees, and using their compatibility graph $CG(\mathscr{G})$, shown in Figure 1(d), the optimal tree is one whose topology is identical to the tree in Figure 1(a). Such a species tree estimate requires seven extra lineages when reconciling all three gene trees within its branches. However, the tree in Figure 7 requires only six extra lineages to reconcile those three trees. We

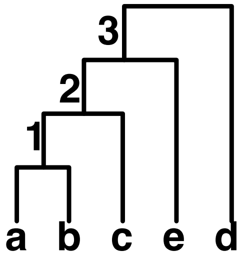
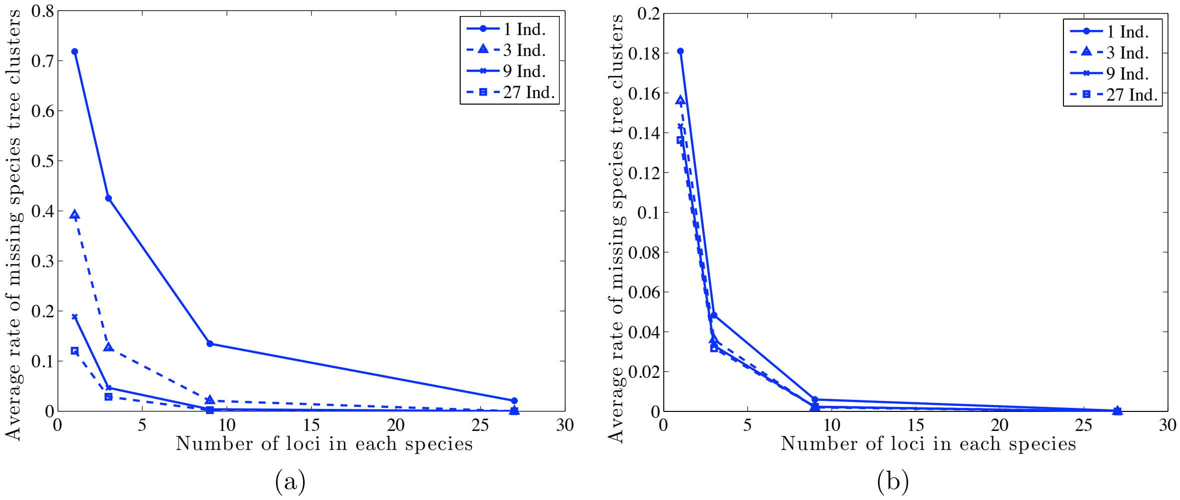


Figure 7. A tree that requires six extra lineages to reconcile the three gene trees in Figure 1.



(a)

(b)

FIGURE 8. Average rates of species tree clusters that do not appear in any gene trees. (a) for data with recent divergence (total branch length is $1N_e$); (b) for data with deep divergence (total branch length is $10N_e$). We note that the y-axes in (a) and (b) are on different scales to make the difference between the curves more visible.

note that it induces cluster $\{a, b, c, e\}$ that does not appear as a node in $CG(\mathscr{G})$. This illustrates that, in general, to obtain an optimal solution to the MDC-T problem, it may not be sufficient to consider only $CG(\mathscr{G})$. It is trivial to state that an optimal solution is guaranteed when using $CG(\mathscr{X})$, since this graph contains *all* possible binary trees as maximal cliques. However, despite these boundary cases, we observe that considering only the set of clusters in $\mathscr{G}$ may be sufficient in practice.

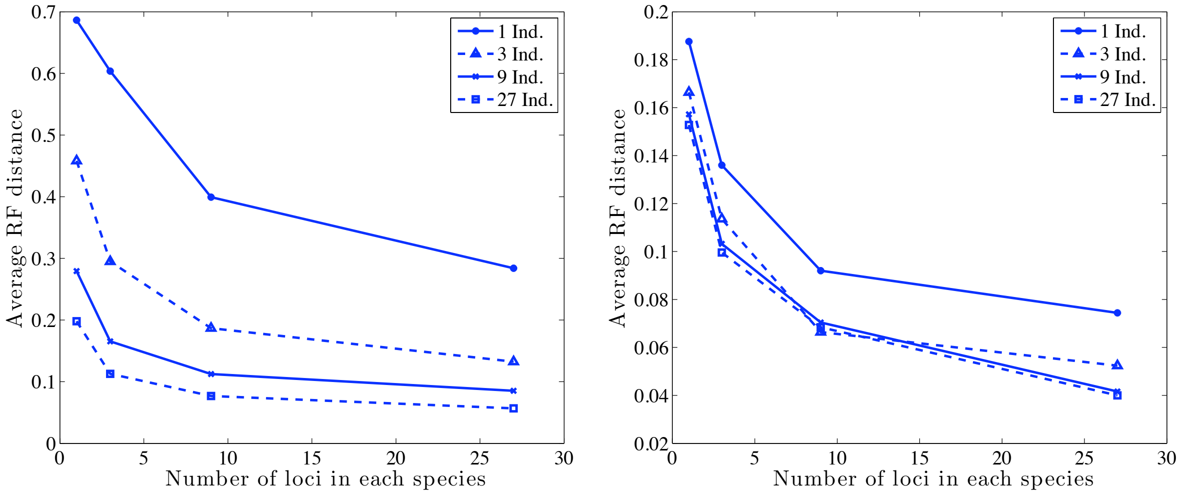
Figure 8 plots the average number of branches (or, clusters) that would be missing from the true species tree if only $CG(\mathscr{G})$ is considered. Clearly, the number of missing branches decreases as the numbers of loci/individuals increase, and no branches are missing when 9 loci are used and at least 3 individuals are sampled. When a single individual is sampled, using all 27 loci guarantees that almost all branches of the species tree would be recoverable from $CG(\mathscr{G})$.

Figure 9 shows the normalized RF distance between the inferred tree and the true species tree, when only $CG(\mathscr{G})$ is considered. When comparing the results in this figure with those in Figure 5, we observe almost no loss in performance of MDC by restricting the set of clusters to those in $CG(\mathscr{G})$. It is important to note that while all branches of the true species tree are present in $CG(\mathscr{G})$ for the case of 27 loci and 27 individuals, the MDC criterion may still not identify the true species tree, since the true species tree may not necessarily be the one that minimizes the amount of deep coalescences. This indicates the number of extra lineages required by the true species tree is larger than the optimal one—a phenomenon encountered by all parsimony-based criteria.

Further, the average number of extra lineages required to reconcile all gene trees within the branches of the optimal species tree estimate was not affected when using only $CG(\mathscr{G})$, as is evident from comparing the results in Figure 10 to those in Figure 6.

It is worth mentioning as well that the accuracy of our inference of the species tree of the yeast data set [10] and *Apicomplexan* data set [4] was not affected by considering only $CG(\mathscr{G})$.

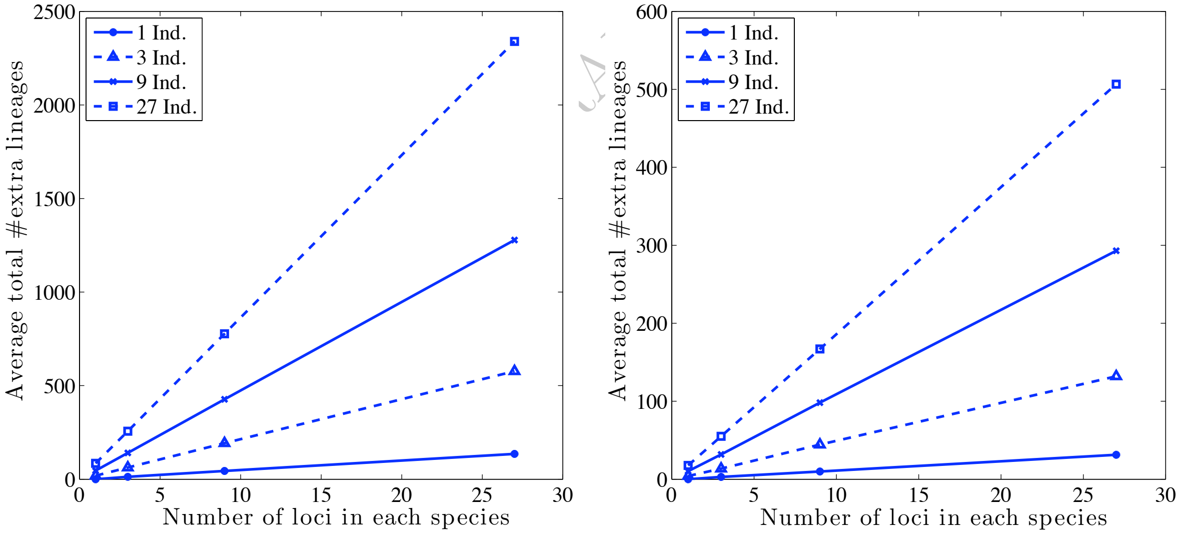
The second motivation behind considering only the clusters of the gene trees is computation. The number of clusters of the gene trees tends to be much smaller than the number of all subsets of $\mathscr{X}$, as illustrated in Section 2. This has a significant impact on the actual running time of the method. Figure 11 shows the average number of clusters induced by the gene trees in each of the simulated data sets. It clearly shows that, while the number of clusters induced by the gene trees increases with the number of loci (and slightly with the number of individuals), this number is still much smaller than the total number of all possible clusters, which is $2^8 - 10 = 246$ (we exclude the clusters including zero, one, or all eight taxa).



(a)

(b)

Figure 9. Accuracy of the inferred species tree as measured by the Robinson-Foulds distance when only clusters induced by gene trees are used. (a) for data with recent divergence (total branch length is $1N_e$); (b) for data with deep divergence (total branch length is $10N_e$). We note that the y-axes in (a) and (b) are on different scales to make the difference between the curves more visible.

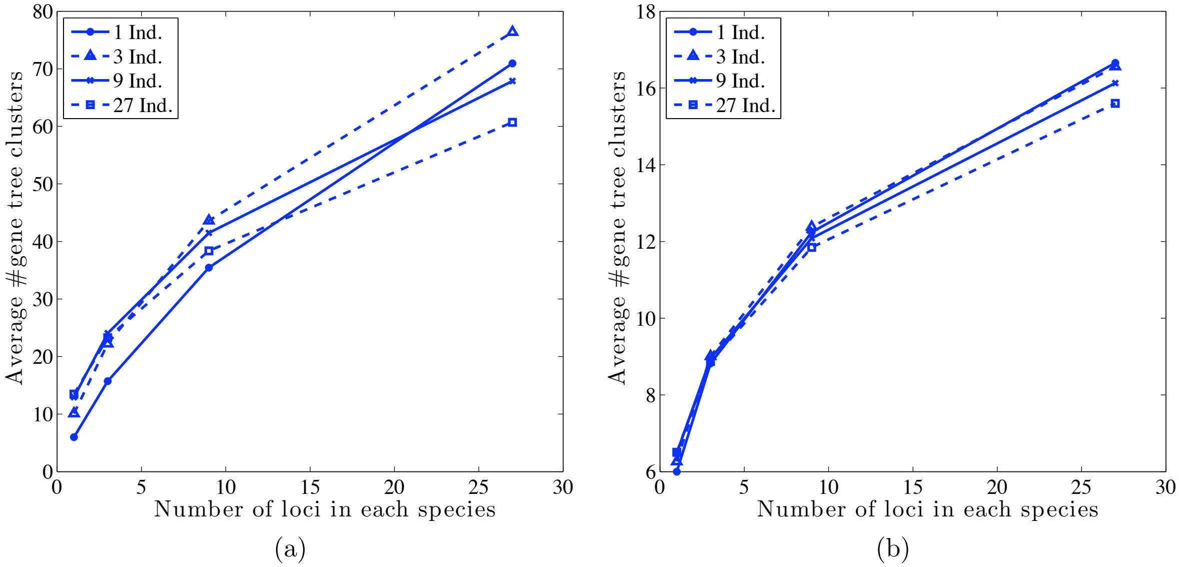


(a)

(b)

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

In summary, while some boundary cases may result in sub-optimality of the tree inferred by MDC when considering only the clusters induced by the input gene trees, empirical results indicate that this may not necessarily be the case in general. Further, as the gap between the number of vertices in $CG(\mathscr{G})$ and the



(a)

(b)

Figure 11. Average numbers of clusters induced by gene trees, excluding single-element and all-element clusters. (a) Recent divergence (total branch length is $1N_e$); (b) Deep divergence (total branch length is $10N_e$). We note that the y-axes in (a) and (b) are on different scales to make the difference between the curves more visible.

number of vertices in $CG(\mathscr{X})$ is very large in practice, it is unclear whether there exists a set $C \subset 2^{\mathscr{X}}$[5] such that (1) C is much smaller than $2^{\mathscr{X}}$ and (2) an optimal solution to the MDC-T problem is guaranteed to be found by considering $CG(C)$. We identify this as a direction for future research.

8. Using PhyloNet

We have implemented the algorithms presented in this chapter in the PhyloNet software package [14]. In this section, we show how to use PhyloNet to (1) count the number of valid coalescent histories, given a species tree and a gene tree, and (2) infer the species tree from multiple gene trees. Once PhyloNet is downloaded from the website (`http://bioinfo.cs.rice.edu/phylonet`), a directory with the file named `phylonet.jar` will be available; this file is the executable program for running all features in PhyloNet.

8.1. Using PhyloNet to Count Valid Coalescent Histories.

PhyloNet implements our algorithm [13] for counting the number of valid coalescent histories; the tool is called `countcoal`. To use this tool, the user invokes the tool on the command line as:

```
java -jar phylonet.jar countcoal -f input
```

Here, `input` is the name of the file that contains a species tree on the first line and a gene tree on the second line. For example, suppose we have the following file, named input-fig2, for the trees in Figure 2:

```
S = ((A, B), ((C, D), E));
G = ((A, B), ((C, E), D));
```

Then, invoking the command:

[5] 2^A denotes the power set of A; that is, the set of all subsets of A.

```
java -jar phylonet.jar countcoal -f input-fig2
```

returns 6, which is the number of valid coalescent histories for reconciling the gene tree in Figure 2(b) within the branches of the species tree in Figure 2(a).

8.2. **Using PhyloNet to Infer Species Trees Under MDC.** There are two tools for inferring the species tree using the MDC criterion: `coal_infer_st`, which is an implementation of the ILP algorithm, and `dpcoal_infer_st`, which is an implementation of the DP algorithm. To use the first tool, the user invokes PhyloNet as follows:

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

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

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

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

As an example, suppose we have a file named `input-fig1` that contains the gene trees in Figure 1:

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

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

```
java -jar phylonet.jar dpcoal_infer_st input-fig1
```

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

In the case where multiple individuals per species may be sampled, the user needs to supply a mapping between gene tree taxa and species tree taxa in a separate file. If a total of k individuals are sampled from all species, then this mapping file contains k lines, each line containing two entries:

```
ind    sp
```

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

```
T1 = ((a1, a2), ((b1, c1), (b2, c2)));
T2 = (((a1, b1), (c1, b2)), (a2, c2));
```

where $a1$, $a2$ are two sampled individuals of species a; $b1$, $b2$ are two sampled individuals of species b; and $c1$, $c2$ are two sampled individuals of species c. Then, in order to reconstruct the species tree for the three species a, b, an c, using the DP algorithm for solving MDC, the user invokes the command:

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

where file `map` contains the following lines

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

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

Finally, it is worth mentioning that PhyloNet has several other features; the user may consult the tool's user manual (available in the download), as well as [14].

9. Conclusions

In this chapter, we reviewed the *minimize deep coalescences*, or MDC, criterion for inferring species trees, which was first proposed by Maddison [5]. Further, we reviewed two exact algorithms that we [12] developed recently for inferring the optimal species tree under the MDC criterion, and demonstrated their accuracy on additional simulated data sets. Finally, we discussed the issue of solving the MDC problem by considering only the clusters induced by the input gene trees. This last point can have implications on the actual running time of the methods, as well as on developing efficient species tree inference methods under other criteria, such as maximum likelihood.

Acknowledgments

We would like to thank Lacey Knowles and Laura Kubatko for inviting us to contribute this chapter, and for their extensive comments on earlier drafts that helped improve the manuscript significantly. This work was supported in part by DOE grant DE-FG02-06ER25734, NSF grant CCF-0622037, and grant R01LM009494 from the National Library of Medicine. The contents are solely the responsibility of the author and do not necessarily represent the official views of the DOE, NSF, National Library of Medicine or the National Institutes of Health.

References

[1] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 2 edition, 2001.

[2] J.H. Degnan and N.A. Rosenberg. Discordance of species trees with their most likely gene trees. *PLoS Genetics*, 2(5):e68, 2006.

[3] J.H. Degnan and L.A. Salter. Gene tree distributions under the coalescent process. *Evolution*, 59:24–37, 2005.

[4] Chih-Horng Kuo, John P. Wares, and Jessica C. Kissinger. The Apicomplexan whole-genome phylogeny: An analysis of incongurence among gene trees. *Mol. Biol. Evol.*, 25(12):2689–2698, 2008.

[5] W.P. Maddison. Gene trees in species trees. *Systematic Biology*, 46(3):523–536, 1997.

[6] W.P. Maddison and L.L. Knowles. Inferring phylogeny despite incomplete lineage sorting. *Systematic Biology*, 55(1):21–30, 2006.

[7] W.P. Maddison and D.R. Maddison. Mesquite: A modular system for evolutionary analysis. Version 1.01. http://mesquiteproject.org, 2004.

[8] Buneman P. The recovery of trees from measures of dissimilarity. In *Mathematics in the Archaeological and Historical Sciences*, pp. 387–395, 1971.

[9] D.R. Robinson and L.R. Foulds. Comparison of phylogenetic trees. *Math. Biosciences*, 53:131–147, 1981.

[10] A. Rokas, B.L. Williams, N. King, and S.B. Carroll. Genome-scale approaches to resolving incongruence in molecular phylogenies. *Nature*, 425:798–804, 2003.

[11] N.A. Rosenberg. Counting coalescent histories. *Journal of Computational Biology*, 14:360–377, 2007.

[12] C. Than and L. Nakhleh. Species tree inference by minimizing deep coalescences. *PLoS Computational Biology*, 5(9):e1000501, 2009.

[13] C. Than, D. Ruths, H. Innan, and L. Nakhleh. Confounding factors in HGT detection: Statistical error, coalescent effects, and multiple solutions. *Journal of Computational Biology*, 14(4):517–535, 2007.

[14] C. Than, D. Ruths, and L. Nakhleh. PhyloNet: a software package for analyzing and reconstructing reticulate evolutionary relationships. *BMC Bioinformatics*, 9(1):322, 2008.

[15] C. Than, R. Sugino, H. Innan, and L. Nakhleh. Efficient inference of bacterial strain trees from genome-scale multi-locus data. *Bioinformatics*, 24:i123–i131, 2008. Proceedings of the 16th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB '08).

[16] J. Wakeley. *Coalescent Theory: An Introduction*. Robserts & Company Publishers, 2007.