

# Algorithms for MDC-Based Multi-locus Phylogeny Inference

Yun Yu<sup>1</sup>, Tandy Warnow<sup>2</sup>, and Luay Nakhleh<sup>1</sup>

<sup>1</sup> Dept. of Computer Science, Rice University, 6100 Main Street, Houston, TX 77005, USA  
{yy9, nakhleh}@cs.rice.edu

<sup>2</sup> Dept. of Computer Sciences, University of Texas at Austin, Austin, TX 78712, USA  
tandy@cs.utexas.edu

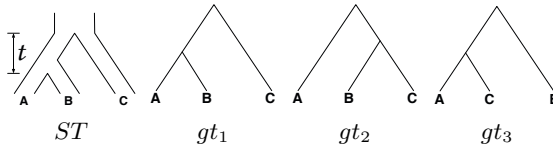
**Abstract.** One of the criteria for inferring a species tree from a collection of gene trees, when gene tree incongruence is assumed to be due to incomplete lineage sorting (ILS), is *minimize deep coalescence*, or MDC. Exact algorithms for inferring the species tree from rooted, binary trees under 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. In this paper, we propose new MDC formulations for the cases where the gene trees are unrooted/binary, rooted/non-binary, and unrooted/non-binary. Further, we prove structural theorems that allow us to extend the algorithms for the rooted/binary gene tree case to these cases in a straightforward manner. Finally, we study the performance of these methods in coalescent-based computer simulations.

## 1 Introduction

Biologists have long acknowledged that the evolutionary history of a set of species—the *species tree*—and that of a genomic region from those species—the *gene tree*—need not be congruent; e.g., [10]. While many processes can cause gene/species tree incongruence, such as horizontal gene transfer and gene duplication/loss, we focus in this paper on *incomplete lineage sorting*, or ILS, which is best understood under the *coalescent model* [13,20,21], as we illustrate in Fig. 1. The coalescent model views gene lineages moving backward in time, eventually coalescing down to one lineage. In each time interval between species divergences (e.g.,  $t$  in Fig. 1), lineages entering the interval from a more recent time period may or may not coalesce—an event whose probability is determined largely by the population size and branch lengths.

Thus, a gene tree is viewed as a random variable conditional on a species tree. For the species tree  $((AB)C)$ , with time  $t$  between species divergences, the three possible outcomes for the gene tree topology random variable, along with their probabilities are shown in Fig. 1. 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 discordance [4,8,14,17,19,25].

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., [2,8]) and democratic vote (e.g., [1,3,26,27]), are fast to compute



**Fig. 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.

and provide a good estimate of the species tree in many cases. However, the accuracy of these methods suffer under certain conditions. Further, these methods do not provide explicit reconciliation scenarios; rather, they provide summaries of the gene trees. Recently, methods that explicitly model ILS were introduced, such as Bayesian inference [5,9], maximum likelihood [7], and the maximum parsimony criterion *Minimize Deep Coalescence*, or MDC [10,11,25]. We introduced the first exact algorithms for inferring species trees under the MDC criterion from a collection of rooted, binary gene trees [22,23]. Nevertheless, in phylogenetic analyses of biological data sets, estimated gene trees may differ from the true gene trees, may be incompletely resolved, and may not be rooted. Requiring gene trees to be fully resolved may result in gene trees with wrong branching patterns (e.g., those branches with low bootstrap support) that masquerade as true gene/species tree incongruence, thus resulting in over-, and possibly under-, estimation of deep coalescences.

Here we propose an approach to estimating species trees from estimated gene trees which avoids these problems. Instead of assuming that all gene trees are correct (and hence fully resolved, rooted trees), we consider the case where 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 which may not be rooted and may 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. We have implemented several of these algorithms and heuristics in our PhyloNet software package [24], which is publicly available at <http://bioinfo.cs.rice.edu/phyloNet>, and we evaluate the performance of these algorithms and heuristics on synthetic data.

## 2 Preliminary Material

*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  $parent(v)$  to be the parent of

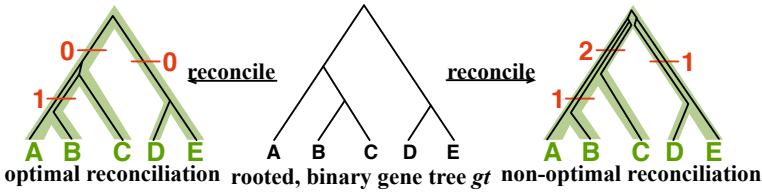
$v$  in  $T$ , and  $Children(v)$  to be the children of  $v$ . A *clade* in a tree  $T$  is a rooted subtree of  $T$ , which can be identified by the node in  $T$  rooting the clade. For a given tree  $T$ , we denote the subtree of  $T$  rooted at  $v$  by  $Clade_T(v)$ , and when the tree  $T$  is understood, by  $Clade(v)$ . The *clade for node  $v$*  is  $Clade(v)$ , and since nodes can have children, the children of a clade  $Clade(v)$  are the clades rooted at the children of  $v$ . The set of all clades of a tree  $T$  is denoted by  $Clades(T)$ . The set of leaves in  $Clade_T(v)$  is called a *cluster* and denoted by  $Cluster_T(v)$  (or more simply by  $Cluster(v)$  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  $MRCAT(A)$  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  $H_T(v) = MRCAT(Cluster_t(v))$ .

We extend the definitions of  $Clades(T)$  and  $Clusters(T)$  to the case where  $T$  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 [18], 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.

*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: (1) if  $w$  is a leaf in  $gt$ , then  $f(w)$  is the leaf in  $ST$  with the same label; and, (2) 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. 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 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. 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  $XL(ST, \mathcal{G}) = \sum_{gt \in \mathcal{G}} XL(ST, gt)$ . 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



**Fig. 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 and 4 extra lineages, respectively

$A \subset A'$  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 [22]) 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.,  $f(v) = MRC_{A_{ST}}(Cluster_{gt}(v))$ ) 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  $O(n^2k)$  time, where  $|\mathcal{X}| = n$ .

The analysis of the running time follows from the following lemma:

**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. We can also compute the set of  $ST$ -maximal clusters in  $gt$  in  $O(n^2)$  time.

### 2.1 The MDC Problem: Rooted, Binary Gene Trees

The MDC problem is the “minimize deep coalescence” problem; as formulated by Wayne Maddison in [10], 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 is from [22]. 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), edges between  $A$  and  $B$  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 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  $MDC(\mathcal{C})$  may not have 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 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

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

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

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. However, Than and Nakhleh showed that using only the clusters of the gene trees would produce almost equally good estimates of the species tree [22,23].

### 3 MDC on Estimated Gene Trees

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.

#### 3.1 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. In this case, the MDC problem can be formulated as follows: given a set  $\mathcal{G} = \{gt_1, gt_2, \dots, gt_k\}$  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 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 3.** *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)  $\mathcal{C}^* \subseteq Clusters(gt')$ , and (2)  $XL(ST, gt') = \sum_{B \in Clusters(ST)} [k_B(gt) - 1]$ , which is the best possible. 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  $\mathcal{C}^* \subseteq Clusters(gt')$ . 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  $\exists A' \in \mathcal{C}^* \text{-} Clusters(gt')$ . 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.

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:

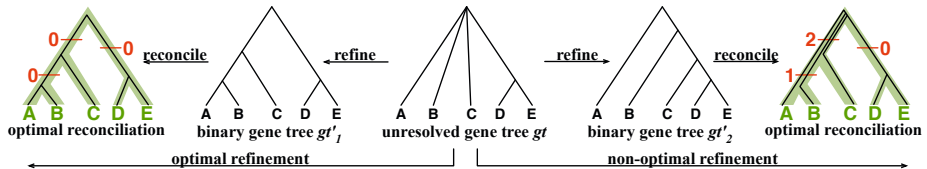
**Corollary 2.** *Let  $ST$  be a species tree and  $\mathcal{G} = \{gt_1, gt_2, \dots, gt_k\}$  be a set of unrooted binary gene trees. Let  $\mathcal{G}' = \{gt'_1, gt'_2, \dots, gt'_k\}$  be a set of binary gene trees such that  $gt'_i$  is a rooted version of  $gt_i$  for each  $i = 1, 2, \dots, k$ , 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.

### 3.2 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. 3. In this case, the MDC problem can be formulated as follows: given a set  $\mathcal{G} = \{gt_1, gt_2, \dots, gt_k\}$  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.

*A Quadratic Algorithm for Optimal Refinement of Gene Trees Under MDC.* We begin with the problem of finding an optimal refinement of a given gene tree  $gt$  with respect to a given species tree  $ST$ , with both trees rooted.



**Fig. 3.** 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



**Definition 1.** (*Optimal tree refinement w.r.t. MDC ( $OTR_{MDC}$ )*)

**Input:** Species tree  $ST$  and gene tree  $gt$ , both rooted and leaf-labelled by set  $\mathcal{X}$  of taxa.

**Output:** Binary rooted tree  $gt^*$  refining  $gt$  that minimizes  $XL(ST, t)$  over all refinements  $t$  of  $gt$ . We denote  $gt^*$  by  $OTR_{MDC}(ST, gt)$ .

We show that  $OTR_{MDC}(ST, gt)$  can be solved in  $O(n^2)$  time, where  $n$  is the number of leaves in either tree. For  $B \in Clusters(ST)$  and gene tree  $gt$ , we define  $F_B(gt)$  to be the number of nodes in  $gt$  that have at least one child whose cluster is  $B$ -maximal. We will show that for a given rooted gene tree  $gt$  and rooted binary species tree  $ST$ , the optimal refinement  $t^*$  of  $gt$  will satisfy  $XL(ST, t^*) = \sum_{B \in Clusters(ST)} [F_B(gt) - 1]$ . Therefore, to compute the score of the optimal refinement of one gene tree  $gt$ , it suffices to compute  $F_B(gt)$  for every  $B \in Clusters(ST)$ .

The algorithm to compute the score of the optimal refinement of  $gt$  first computes the set of  $B$ -maximal clusters, which takes  $O(n)$  time by Lemma 1. It then computes  $F_B(gt)$ , for each  $B$ ; this requires an additional  $O(n)$  time per  $B$ , for a total cost of  $O(n^2)$  time.

*Algorithm for  $OTR_{MDC}(ST, gt)$ :* To compute the optimal refinement, we have a slightly more complicated algorithm.

*Step 1: Preprocessing.* We begin by computing  $H_{ST}(v)$  for every node  $v \in V(gt)$ , as described above; this takes  $O(n)$  time overall.

*Step 2: Refine at every high degree node.* We then visit each internal node  $v$  of  $gt$  that has more than two children, and we modify the tree  $gt$  locally at  $v$  by replacing the rooted star tree at  $v$  by a tree defined by the topology induced in  $ST$  by the images under the mapping  $H_{ST}$  of  $v$  and  $v$ 's children. The order in which we visit the nodes is irrelevant.

We now make precise how this modification of  $gt$  at node  $v$  is performed. We denote by  $Tree(ST, gt, v)$  the tree formed as follows. First, we compute the subtree of  $ST$  induced by the images of  $v$  and its children under the  $H_{ST}$  mapping. If a child  $y$  of  $v$  is mapped to an internal node of the induced subtree, we add a leaf  $l_y$  and make it a child of  $H_{ST}(y)$ ; in this way, the tree we obtain has all the nodes in  $Children(v)$  identified with distinct leaves in  $Tree(ST, gt, v)$ . (Although  $ST$  is assumed to be binary,  $Tree(ST, gt, v)$  may not be binary.) After we compute  $Tree(ST, gt, v)$ , we modify  $gt$  by replacing the subtree of  $gt$  induced by  $v$  and its children with  $Tree(ST, gt, v)$ . The subtree within the refinement that is isomorphic to  $Tree(ST, gt, v)$  is referred to as the *local subtree at  $v$* .

*Step 3: Completely refine if necessary.* Finally, after the refinement at every node is complete, if the tree is not binary, we complete the refinement with an arbitrary refinement at  $v$ .

**Theorem 2.** *Algorithm  $OTR_{MDC}(ST, gt)$  takes  $O(n^2)$  time, where  $ST$  and  $gt$  each have  $n$  leaves.*

It is clear that the algorithm is well-defined, so that the order in which we visit the nodes in  $V(gt)$  does not impact the output.

**Observation 1.** *Let  $gt$  be an arbitrary rooted gene tree,  $gt'$  a refinement of  $gt$ , and  $ST$  an arbitrary rooted binary species tree. Then  $k_B(gt') \geq F_B(gt)$  for all clusters  $B$  of  $ST$ .*

**Theorem 3.** *Let  $gt$  be an arbitrary rooted gene tree,  $ST$  an arbitrary rooted binary species tree,  $t$  the result of the first two steps of  $OTR_{MDC}(ST, gt)$ , and  $t^*$  an arbitrary refinement of  $t$  (thus  $t^* = OTR_{MDC}(ST, gt)$ ). Then for all  $B \in Clusters(ST)$ ,  $F_B(gt) = F_B(t^*)$  and no node in  $t$  or  $t^*$  has more than one  $B$ -maximal child.*

*Proof.* Step 2 of  $OTR_{MDC}(ST, gt)$  can be seen as a sequence of refinements that begins with  $gt$  and ends with  $t$ , in which each refinement is obtained by refining around a particular node in  $gt$ . The tree  $t^* = OTR_{MDC}(ST, gt)$  is then obtained by refining  $t$  arbitrarily into a binary tree, if  $t$  is not fully resolved. Let the internal nodes of  $gt$  with at least three children be  $v_1, v_2, \dots, v_k$ . Thus,  $gt = gt_0 \rightarrow gt_1 \rightarrow gt_2 \rightarrow \dots \rightarrow gt_k = t \rightarrow t^*$ , where  $gt_i \rightarrow gt_{i+1}$  is the act of refining at node  $v_{i+1}$ , and  $t \rightarrow t^*$  is an arbitrary refinement.

We begin by showing that  $F_B(gt_i) = F_B(gt_{i+1})$ , for  $i = 0, 1, 2, \dots, k-1$ . When we refine at node  $v_i$ , we modify the tree  $gt_{i-1}$  by replacing the subtree immediately below node  $v_i$  by  $Tree(ST, gt, v_i)$ , producing the local subtree below  $v_i$ . Fix a cluster  $B \in Clusters(ST)$ . If the cluster for  $v_i$  in  $gt_{i-1}$  does not have any  $B$ -maximal children, then refining at  $v_i$  will not change  $F_B$ , and hence  $F_B(gt_{i-1}) = F_B(gt_i)$ . Otherwise,  $v_i$  has at least one  $B$ -maximal child in  $gt_{i-1}$ . Since  $v_i$  is not  $B$ -maximal within  $gt_{i-1}$ ,  $v_i$  also has at least one child in  $gt_{i-1}$  that is not  $B$ -maximal. Hence, the tree  $gt_i$  produced by refining  $gt_{i-1}$  at  $v_i$  (using  $Tree(ST, gt_i, v_i)$ ) contains a node  $y$  that is an ancestor of all the  $B$ -maximal children of  $v_i$  within  $gt_{i-1}$  and not the ancestor of any other children of  $v_i$  in  $gt_{i-1}$ . Therefore, the cluster for  $y$  is  $B$ -maximal within  $gt_i$ , and no other node that is introduced during this refinement is  $B$ -maximal within  $gt_i$ . Therefore within the local subtree at  $v_i$  in  $gt_i$  there is exactly one node that defines a  $B$ -maximal cluster, and exactly one node that is the parent of at least one  $B$ -maximal cluster. As a result,  $F_B(gt_{i-1}) = F_B(gt_i)$ .

This argument also shows that any node in the local subtree at  $v_i$  that is the parent of at least one  $B$ -maximal cluster is the parent of exactly one  $B$ -maximal cluster. On the other hand, if  $v_i$  does not have any  $B$ -maximal child in  $gt_{i-1}$ , then there is no node in  $v_i$ 's local subtree that has any  $B$ -maximal children. In other words, after refining at node  $v_i$ , any node within the local subtree at  $v_i$  that has one or more  $B$ -maximal children has exactly one such child. As a result, at the end of Step 2 of  $OTR_{MDC}(ST, gt)$ , every node has at most one  $B$ -maximal child, for all  $B \in Clusters(ST)$ .

The last step of the  $OTR_{MDC}$  algorithm produces an arbitrary refinement of  $t = gt_k$ , if it is not fully resolved. But since no node in  $gt_k$  can have more than one  $B$ -maximal child, if  $t^*$  is a refinement of  $t = gt_k$  then  $F_B(t) = F_B(t^*)$ .

**Theorem 4.** *Let  $gt$  be a rooted gene tree,  $ST$  a rooted binary species tree, both on set  $\mathcal{X}$ ,  $t$  the result of the first two steps of  $OTR_{MDC}(ST, gt)$ , and  $t^*$  any refinement of  $t$ . Then  $XL(ST, t^*) = \sum_{B \in Clusters(ST)} [F_B(gt) - 1]$ , and  $t^*$  is a binary refinement of  $gt$  that minimizes  $XL(ST, t')$  over all binary refinements  $t'$  of  $gt$ .*

*Proof.* Let  $B$  be an arbitrary cluster in  $ST$ . By Theorem 3,  $F_B(t^*) = F_B(gt)$ . Also by Theorem 3, no node in  $t$  has more than one  $B$ -maximal child, and so  $k_B(t) =$

$F_B(t)$ . Since  $t^*$  is an arbitrary refinement of  $t$ , it follows that  $k_B(t^*) = F_B(t^*)$ , and so  $k_B(t^*) = F_B(gt)$ . By Observation 1, for all refinements  $t'$  of  $gt$ ,  $k_B(t') \geq F_B(gt)$ . Hence  $k_B(t') \geq k_B(t^*)$  for all refinements  $t'$  of  $gt$ . Since this statement holds for an arbitrary cluster  $B$  in  $ST$ , it follows that  $XL(ST, t') \geq XL(ST, t^*)$  for all refinements  $t'$  of  $gt$ , establishing the optimality of  $t^*$ .

**Corollary 3.** *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.* Corollary 3 allows us to compute the score of any species tree with respect to a set of rooted but unresolved gene trees. We can use this to find optimal species trees from rooted, non-binary gene trees, as we now show. Let  $\mathcal{G}$  be a set of rooted gene trees that are not necessarily binary. By Corollary 3, we can formulate the problem as a minimum-weight clique problem. The graph has one vertex for every subset of  $\mathcal{X}$ , and we set the weight of the vertex corresponding to subset  $B$  to be  $w(B) = \sum_{gt \in \mathcal{G}} [F_B(gt) - 1]$ . We have edges between vertices if the two vertices are compatible (can both be contained in a tree). The solution is therefore a minimum weight clique with  $n - 2$  vertices. And, as before, we can describe this as a maximum weight clique problem by having the weight be  $w'(B) = Q - w(B)$ , for some large enough  $Q$ .

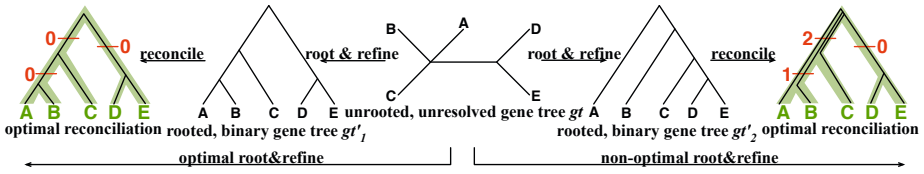
However, we can also address this problem using dynamic programming, as before. Let  $A \subseteq \mathcal{X}$  and  $T \in \mathcal{T}_A$ . Let  $l_T(A, \mathcal{G}) = \sum_{gt \in \mathcal{G}} \sum_B [F_B(gt) - 1]$ , as  $B$  ranges over the clusters of  $T$ . Let  $l^*(A, \mathcal{G}) = \min_{T \in \mathcal{T}_A} \{l_T(A, \mathcal{G})\}$ . Then  $l^*(\mathcal{X}, \mathcal{G})$  is the solution to the problem of inferring a species tree from rooted, non-binary gene trees.

We set base cases  $l^*(\{x\}, \mathcal{G}) = 0$  for all  $x \in \mathcal{X}$ . We order the subproblems by the size of  $A$ , and compute  $l^*(A, \mathcal{G})$  only after every  $l^*(A', \mathcal{G})$  is computed for  $A' \subset A$ . The DP formulation is

$$l^*(A, \mathcal{G}) = \min_{A_1 \subset A} \{l^*(A_1, \mathcal{G}) + l^*(A - A_1, \mathcal{G}) + \sum_{gt \in \mathcal{G}} [F_A(gt) - 1]\}. [-5mm]$$

### 3.3 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. In this case, the MDC 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.



**Fig. 4.** 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

**Observation 2.** 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  $\sum_{B \in Clusters(ST)} [F_B(gt') - 1]$ .

From this, the following theorem follows:

**Theorem 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  $A \in Clusters(gt)$  be a largest  $ST$ -maximal cluster, and  $v$  be the neighbor of the root of the clade for  $A$  that is in  $A$ . If we root  $gt$  at  $v$ , then the resultant tree  $gt'$  minimizes  $\sum_{B \in Clusters(ST)} [F_B(gt') - 1]$  over all rooted versions  $gt'$  of  $t$ .

And, therefore,

**Theorem 6.** Let  $\mathcal{T}$  be a set of gene trees that are unrooted and not necessarily binary. For  $B \subset \mathcal{X}$ , define  $t^B$  to be the rooted version of  $t$  formed by rooting  $t$  at a node  $v$ , as given by Theorem 5. Then, the species tree  $ST$  that minimizes  $\sum_{t \in \mathcal{T}} \sum_{B \in Clusters(ST)} [F_B(t^B) - 1]$  is an optimal solution to the problem.

As a result, we can solve the problem using the clique and DP formulations as in the other versions of the MDC problem.

## 4 Experimental Evaluation

### 4.1 Methods

*Simulated data.* We generated species trees using the “Uniform Speciation” (Yule) module in the program Mesquite [12]. Two sets of species trees were generated: one for 8 taxa plus an outgroup, and one for 16 taxa plus an outgroup. Each data set had 500 species trees. All of them have a total branch length of 800,000 generations excluding the outgroup. Within the branch of each species tree, 1, 2, 4, 8, 16, or 32 gene trees were simulated using the “Coalescence Contained Within Current Tree” module in Mesquite with the effective population size  $N_e$  equal 100,000. We sampled one allele per species. We used the program Seq-gen [15] to simulate the evolution of DNA sequences of length 2000 under the Jukes-Cantor model [6] down each of the gene trees (these settings are similar to those used in [11]).

*Estimated gene trees.* We estimated gene trees from these sequence alignments using default PAUP\* heuristic maximum parsimony (MP) methods, returning the strict consensus of all optimal MP trees. We rooted each estimated tree at the outgroup in order to produce rooted estimated trees.

*Estimated species trees.* The “heuristic” version of our method uses only the clusters of the input gene trees, and the “exact” version uses all possible clusters on the taxon set. For some analyses using the heuristic MDC algorithms, the estimated species tree is not 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. This additional step was limited to 5 minutes of analysis. The only cases where this additional search was not applied were when the polytomy (unresolved node) in the species tree was present in all gene trees; in these cases, any resolution is arbitrary and is as good (under the MDC) criterion as any other resolution.

For the 8-taxon data sets, we used both the exact and heuristic versions of all four algorithms. For the 16-taxon data sets, we used only the heuristic versions.

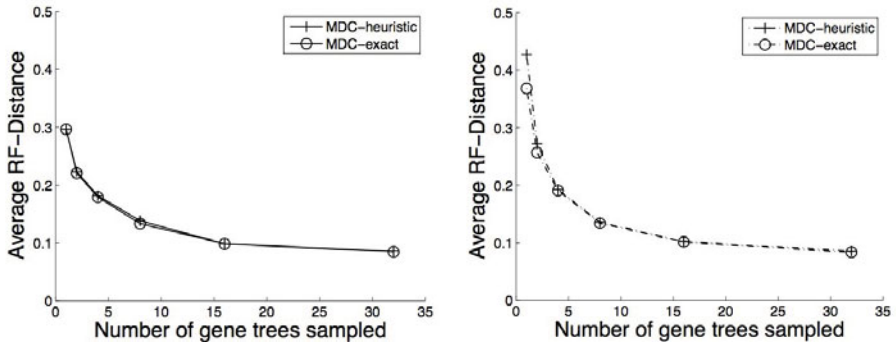
*Measurements.* We report the degree of resolution of each estimated gene tree, which is the number of internal branches in  $t$  divided by  $n - 3$ , where  $t$  has  $n$  leaves. We also report the Robinson-Foulds (RF) error [16] of estimated trees to the true trees, where the RF error is the total number of edges in the two trees that define bipartitions that are not shared by the other tree, divided by  $2n - 6$ . A value 0 of the RF distance indicates the two trees are identical, and a value of 1 indicates the two trees are completely different (they disagree on every branch).

## 4.2 Results

The degree of resolution of the reconstructed gene trees was around 0.6 in the case of 8-taxon gene trees, and around 0.5 in the case of 16-taxon gene trees.

With respect to topological accuracy of the estimated gene trees, we found that for 8 taxa, the RF distance is around 0.21. However, 98% of the estimated gene trees have no false positives; thus, all but 2% of the estimated gene trees can be resolved to match the true gene tree. Similarly, the RF distance for the 16-taxon data sets between true gene trees and reconstructed gene trees is around 0.27, but 96% have 0 false positive values.

We now discuss topological accuracy of the species trees estimated using our algorithms for solving the MDC problem. We show results on running the exact and heuristic versions of the algorithms on 8 taxon estimated gene trees in Figure 5. These results show that increasing the number of gene trees improves the accuracy of the estimated species tree, and that very good accuracy is obtainable from a small number of gene trees. We also see that knowing the true root instead of estimating the root is helpful when the number of gene trees is very small, but that otherwise our algorithm is able to produce comparable results even on unrooted gene trees. The results also show that the heuristic version of our algorithm is as accurate as the exact version once there are four or more gene trees (and almost identical in accuracy for two gene trees).



**Fig. 5.** Performance of MDC methods on estimated gene trees with 8 taxa. Left: MDC on estimated gene trees with correct roots. Right: MDC on unrooted estimated gene trees.

## Acknowledgement

This work was supported in part by NSF grant CCF-0622037, grant R01LM009494 from the National Library of Medicine, an Alfred P. Sloan Research Fellowship to LN, a Guggenheim Fellowship to TW, and by Microsoft Research New England support to TW. The contents are solely the responsibility of the authors and do not necessarily represent the official views of the NSF, National Library of Medicine, the National Institutes of Health, the Alfred P. Sloan Foundation, or the Guggenheim Foundation.

## References

1. Dawkins, R.: *The Ancestor's Tale*. Houghton Mifflin, New York (2004)
2. Degnan, J.H., DeGiorgio, M., Bryant, D., Rosenberg, N.A.: Properties of consensus methods for inferring species trees from gene trees. *Syst. Biol.* 58, 35–54 (2009)
3. Degnan, J.H., Rosenberg, N.A.: Discordance of species trees with their most likely gene trees. *PLoS Genet.* 2, 762–768 (2006)
4. Degnan, J.H., Rosenberg, N.A.: Gene tree discordance, phylogenetic inference and the multispecies coalescent. *Trends Ecol. Evol.* 24, 332–340 (2009)
5. Edwards, S.V., Liu, L., Pearl, D.K.: High-resolution species trees without concatenation. *PNAS* 104, 5936–5941 (2007)
6. Jukes, T.H., Cantor, C.R.: Evolution of protein molecules. In: Munro, H.N. (ed.) *Mammalian Protein Metabolism*, pp. 21–132. Academic Press, New York (1969)
7. Kubatko, L.S., Carstens, B.C., Knowles, L.L.: STEM: species tree estimation using maximum likelihood for gene trees under coalescence. *Bioinformatics* 25(7), 971–973 (2009)
8. Kuo, C.-H., Wares, J.P., Kissinger, J.C.: The Apicomplexan whole-genome phylogeny: An analysis of incongruence among gene trees. *Mol. Biol. Evol.* 25(12), 2689–2698 (2008)
9. Liu, L., Pearl, D.K.: Species trees from gene trees: Reconstructing Bayesian posterior distributions of a species phylogeny using estimated gene tree distributions. *Systematic Biology* 56(3), 504–514 (2007)
10. Maddison, W.P.: Gene trees in species trees. *Syst. Biol.* 46, 523–536 (1997)

11. Maddison, W.P., Knowles, L.L.: Inferring phylogeny despite incomplete lineage sorting. *Systematic Biology* 55(1), 21–30 (2006)
12. Maddison, W.P., Maddison, D.R.: Mesquite: A modular system for evolutionary analysis (2004), version 1.01 <http://mesquiteproject.org>
13. Nei, M.: Stochastic errors in DNA evolution and molecular phylogeny. In: Gershowitz, H., Rucknagel, D.L., Tashian, R.E. (eds.) *Evolutionary Perspectives and the New Genetics*, pp. 133–147. Alan R. Liss, New York (1986)
14. Pollard, D.A., Iyer, V.N., Moses, A.M., Eisen, M.B.: Widespread discordance of gene trees with species tree in *Drosophila*: evidence for incomplete lineage sorting. *PLoS Genet.* 2, 1634–1647 (2006)
15. Rambaut, A., Grassly, N.C.: Seq-gen: An application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Comp. Appl. Biosci.* 13, 235–238 (1997)
16. Robinson, D.R., Foulds, L.R.: Comparison of phylogenetic trees. *Math. Biosci.* 53, 131–147 (1981)
17. Rokas, A., Williams, B.L., King, N., Carroll, S.B.: Genome-scale approaches to resolving incongruence in molecular phylogenies. *Nature* 425, 798–804 (2003)
18. Semple, C., Steel, M.: *Phylogenetics*. Oxford University Press, Oxford (2003)
19. Syring, J., Willyard, A., Cronn, R., Liston, A.: Evolutionary relationships among *Pinus* (Pinaceae) subsections inferred from multiple low-copy nuclear loci. *American Journal of Botany* 92, 2086–2100 (2005)
20. Tajima, F.: Evolutionary relationship of DNA sequences in finite populations. *Genetics* 105, 437–460 (1983)
21. Takahata, N.: Gene genealogy in three related populations: consistency probability between gene and population trees. *Genetics* 122, 957–966 (1989)
22. Than, C., Nakhleh, L.: Species tree inference by minimizing deep coalescences. *PLoS Computational Biology* 5(9), e1000501 (2009)
23. Than, C., Nakhleh, L.: Inference of parsimonious species phylogenies from multi-locus data by minimizing deep coalescences. In: Knowles, L.L., Kubatko, L.S. (eds.) *Estimating Species Trees: Practical and Theoretical Aspects*, pp. 79–98. Wiley-VCH, Chichester (2010)
24. Than, C., Ruths, D., Nakhleh, L.: PhyloNet: a software package for analyzing and reconstructing reticulate evolutionary relationships. *BMC Bioinformatics* 9, 322 (2008)
25. Than, C., Sugino, R., Innan, H., Nakhleh, L.: 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 2008)*
26. Wu, C.-I.: Inferences of species phylogeny in relation to segregation of ancient polymorphisms. *Genetics* 127, 429–435 (1991)
27. Wu, C.-I.: Reply to Richard R. Hudson. *Genetics* 131, 513 (1992)