# Estimating Optimal Species Trees from Incomplete Gene Trees under Deep Coalescence

Md. Shamsuzzoha Bayzid[1] and Tandy Warnow*[1]

[1]Department of Computer Science, University of Texas at Austin, Austin, Texas, USA

Email: shams.bayzid@gmail.com; Tandy Warnow*- tandy@cs.utexas.edu;

*Corresponding author

## Abstract

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

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

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

## Introduction

Phylogenetic tree estimation for individual genes is a challenging problem for many reasons, especially for large datasets where nucleotide alignment estimation can be difficult [1–3] and errors in gene sequence alignment can result in errors in the estimated gene trees. However, new alignment methods, such as MAFFT [4,5], Opal [6], PRANK [7], and SATé [1,8], have been developed that produce improved alignments compared to earlier methods. In addition, while maximum likelihood phylogeny estimation used to be too computationally intensive to be used on datasets with thousands of sequences, newer methods, including RAxML [9], FastTree-2 [10], GARLI [11], and PhyML [12], have made large-scale maximum likelihood analysis much more accessible. Of these methods, RAxML is probably the most popular, but as shown in [13], FastTree-2 is much faster than RAxML and may be as accurate as RAxML for very large datasets. Finally, SATé [1, 8], a method that co-estimates both alignments and trees using an iterative divide-and-conquer approach, has been shown to produce more accurate alignments and trees than standard two-phase methods (first align and then estimate a tree) on large datasets with hundreds to thousands of sequences.

Thus, in the last ten years or so, there has been a dramatic improvement in computational methods for estimating gene trees and sequence alignments. However, species tree estimation represents an additional challenge, because no individual gene tree is necessarily a good estimate of the true species tree. That is, for a number of different biological reasons, including incomplete lineage sorting (ILS), gene duplication and loss, and horizontal gene transfer, gene trees can differ from the species tree, as discussed in [14, 15]. As a result, species tree estimations need to take causes of discord between gene trees and species trees into consideration, in order to produce reasonably accurate estimates of the species tree.

In this paper, we consider the problem of estimating species trees from estimated gene trees when the true gene trees can differ from the true species trees due to incomplete lineage sorting, also known as deep coalescence. Because of the frequency of deep coalescent events in phylogenetic analyses of closely related species, many methods for estimating species trees from gene trees or gene sequence alignments have been developed that explicitly take deep coalescence into account; see [16] for a relatively recent survey of methods, and [15] for a discussion of the importance of these methods for biological data analysis. Studies evaluating these methods have examined performance with respect to tree error and computational requirements on simulated datasets (surveyed in [17]), all restricted to datasets in which all gene trees have at least one individual for each species. Most of these studies have shown that methods that explicitly use statistical models to inform the estimation produce the best results; however, [17] showed that some very simple fast methods (in particular, the greedy consensus) came close to the accuracy of a statistically-based method,

BUCKy [18] on tree distributions estimated using MrBayes [19].

In this paper we consider the problem of estimating species trees from incomplete estimated gene trees, by which we mean the case where the gene trees might not contain any individuals for some species. In this case, methods that require that all the gene trees have the same set of taxa (such as the greedy consensus and BUCKy) cannot be applied. In addition, results from prior studies that evaluated methods on inputs in which all gene trees have at least one individual from each species are not necessarily applicable, since performance on incomplete gene trees could be different.

We begin with a study of the Minimize Deep Coalescence (MDC) problem (find the species tree for which there is a minimum total number of deep coalescences) introduced by Maddison [14]. We show how to extend MDC to the case where the gene trees are incomplete, and we prove that Phylonet-MDC [20] solves this exactly. We then report on a simulation study we performed to evaluate methods for estimating species trees from incomplete gene trees or alignments for datasets with multiple genes and with 11, 17, or 100 taxa. We compare *BEAST [21] (a Bayesian method for estimating species trees from gene sequence alignments when genes can differ from species trees due to ILS) to methods based upon MDC (iGTP-MDC [22] and Phylonet-MDC). We also make comparisons to a heuristic for MRP (matrix representation with parsimony, a standard supertree method) [23, 24] known to be one of the most accurate supertree methods [25, 26] and to heuristics to minimize duplications or duplications+losses in iGTP [22], none of which consider ILS when estimating species trees. We compare these methods on datasets simulated on gene trees that can differ from species trees due to ILS and report the missing branch rates of each species tree that we compute.

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

| Name | Meaning | Comments |
|---|---|---|
| ILS | Incomplete Lineage Sorting | Also called "deep coalescence" |
| MBMC | Minimizing B-maximal clusters | A computational problem for estimating species trees from complete gene trees, shown to be equivalent to MDC in [27] |
| MBMC$_{inc}$ | MBMC for incomplete gene trees | Extension of MBMC to incomplete gene trees, shown here to be equivalent to MDC$_{inc}$. |
| MDC | Minimize Deep Coalescence | Optimization problem for species tree estimation in the presence of ILS, defined only for complete gene trees |
| MDC$_{inc}$ | MDC for incomplete gene trees | MDC$_{inc}$ seeks completions of all gene trees and a species tree, so that the species tree optimizes MDC with respect to the completed gene trees. |
| MRP | Matrix Representation with Parsimony | Standard optimization problem for supertree computation, known to be NP-hard. |

Table 1: Acronyms used in this paper

| Name | Summary | Reference |
|---|---|---|
| *BEAST | Bayesian co-estimation of gene trees and species trees, in the presence of ILS | [21] |
| FastTree-2 (FT) | Fast maximum likelihood phylogeny estimation. FT-75 refers to the tree obtained by running FastTree-2 and then collapsing all branches with support below 75%. | [10] |
| iGTP | Gene Tree Parsimony software, implementing a heuristic search to construct species trees from sets of gene trees, under three criteria: MDC, duplications, and duplications plus losses | [22] |
| PAUP* | Phylogenetic Analysis using Parsimony (*and Other Methods). We use heuristics in PAUP* for parsimony, applied to an MRP matrix we compute. | [28] |
| Phylonet | Software package that performs several functions related to species phylogeny estimation from sets of gene trees. In this paper we use Phylonet to find solutions (exact or heuristic) to the MDC problem. | [20] |

Table 2: Software used in this paper

## Theoretical results for MDC

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

### MDC for complete gene trees.

The MDC problem is as follows:

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

To define the MDC problem, therefore, we need to define $XL(T, t_i)$, i.e., the number of extra lineages of a species tree $T$ with respect to a gene tree $t_i$. Visually, this is defined by embedding the gene tree $t_i$ into the species tree $T$, and then counting how many lineages there on each edge of the species tree; for a given edge, the number of extra lineages is one less than the total number of lineages on the edge [14]. This visual definition of the MDC cost is not necessarily easy to understand.

An alternative definition is given in terms of what are called "$B$-maximal clusters", which we now define. A cluster is a subset of the leaf set of a rooted tree, consisting of all the leaves below some internal node. Thus, given a cluster within a tree defined by the node $v$, we can also define its parent cluster to be the cluster associated with the parent of $v$. Furthermore, the "parent edge" of a cluster $C$ defined by the node $v$ (i.e., $v$ is the root of the subtree whose leafset is $C$) is the edge $e = (v, w)$, where $w$ is between $v$ and the root of $t$. Let $T$ and $t$ be rooted binary trees on $S$, with $T$ denoting the species tree and $t$ denoting the gene tree. Let $B$ be a cluster of $T$. We will say that cluster $A$ of $t$ is $B$-maximal if $A \subseteq B$ and the parent cluster for $A$ is not a subset of $B$.

For a cluster $B$ of $T$, we define $k_B(t)$ to be the number of $B$-maximal clusters of $t$, and we let $w_B(t) = k_B(t) - 1$. It is now known that the embedding of the gene tree $t$ into the species tree $T$ that maps every node in $t$ to MRCA (most recent common ancestor) in $T$ of the leafset below $v$ optimizes the MDC cost. Furthermore, for this embedding, the number of lineages "leaving" the parent edge of the cluster $B$ (i.e., the edge between $B$ and the root of the tree $T$) is $k_B(t)$; therefore, the number of extra lineages on the parent edge is $w_B(t)$ (one less than the number of lineages) (see, for example, [27]). Note that $w_B(t) \geq 0$ since $t$ and $T$ have the same set of taxa, and that $XL(T, t) = \sum_B w_B(t)$, where the sum is taken over all clusters

$B$ in the tree $T$, is the number of extra lineages implied by the pair $t, T$. This is what is meant by the MDC cost for $T$ with respect to gene tree $t$.

The MDC problem can then be restated as follows:

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

Than and Nakhleh noted that this problem could be solved exactly by finding a minimum weight clique of size $n-2$ in a graph in which there is a node for every possible cluster in the species tree (i.e., subset of taxa), an edge between nodes where their clusters are compatible (meaning that they can co-exist in a rooted tree), and where the weight of the node for cluster $B$ is $\sum_i w_B(t_i)$ [29]. This observation yielded the exact version of Phylonet-MDC [20]. By restricting the set of nodes to those clades that appear in the input set of gene trees, they produced the heuristic version of Phylonet-MDC; this method solves the MDC problem exactly when constrained to species trees whose clades are drawn only from the input gene tree clades. Finally, Yu, Warnow and Nakhleh [27] showed how to modify the Phylonet-MDC algorithm so that it could work with unrooted, incompletely resolved gene trees and find optimal rooted refinements and species trees that minimize the MDC score.

**Extension to Incomplete Gene Trees.**

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

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

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

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

Recall that $k_B(t)$ is defined for the case where the gene tree $t$ is rooted and has the same set of taxa as the species tree; in this case, it equals the number of $B$-maximal clusters of $t$. Furthermore, again for the case where the gene trees all have the same set of taxa, we have defined $XL(T, \mathcal{T}) = \sum_B \sum_i w_B(t)$, where $B$ ranges over all clusters of $T$, $i$ ranges from 1 to $k$, and $w_B(t) = k_B(t) - 1$. However, we will modify the definition of $w_B(t)$ to appropriately reflect the possibility that the cluster $B$ may contain taxa that do not appear in $t$. That is, we set

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

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

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

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

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

The main result in this paper is the following:

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

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

**Phylonet-MDC and iGTP-MDC.** The software packages Phylonet [20] and iGTP [22] handle incomplete gene trees differently when attempting to solve MDC, in that they can compute MDC scores differently. In particular, Phylonet defines the MDC score using the $MBMC_{inc}$ cost, as described above (i.e., the cost of a species tree $T$ is $\sum_i \sum_B w_B(t_i)$, where $B$ ranges over the clusters in $T$ and $i$ ranges from 1 to $k$). Theorem 1 thus shows that Phylonet-MDC computes the MDC score correctly. By contrast, there are inputs for which iGTP-MDC does not return this score, indicating that iGTP-MDC defines the MDC score differently for incomplete gene trees. One particular instance in which this occurs is as follows:

- Input gene trees: $T_1$ = (((a, b), c), d), $T_2$ = ((b, c), (d, e)), and $T_3$ = ((a, d), (b, e)).
- Output of Phylonet-MDC exact version: ((d,e),(a,(b,c,))), claiming 3 extra lineages.
- Output of iGTP(MDC): ((d,e),(c,(a,b))), and claims 2 extra lineages.

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

## Establishing the relationship between $MDC_{inc}$ and $MBMC_{inc}$.

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

We begin with the following lemma.

**Lemma 1.** *Let $T$ and $t$ be rooted binary trees with $\mathcal{L}(t) \subset \mathcal{L}(T)$, and let $X$ be a maximal cluster in $T$ with $X \cap \mathcal{L}(t) = \emptyset$. Let $B_0$ be the sibling cluster of $X$ in $T$ (i.e., $X \cup B_0$ is the smallest cluster in $T$ that properly contains $X$), and let $A_0$ be any $B_0$-maximal cluster in $t$. Let $t'$ be the rooted binary tree obtained by modifying $t$ by inserting the clade for $X$ as the the sibling to the clade on $A_0$. Then for all clusters $B$ of $T$, $w_B(t) = w_B(t')$.*

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

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

We take each case in turn.

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

Case 2: $B \subseteq B_0$. First, if $B \cap \mathcal{L}(t) = \emptyset$, then $B \cap \mathcal{L}(t') = \emptyset$ and $w_B(t) = w_B(t') = 0$. Hence, assume that $B \cap \mathcal{L}(t) \neq \emptyset$. We will show that $A$ is a $B$-maximal cluster in $t$ if and only if $A$ is a $B$-maximal cluster in $t'$, and so $w_B(t) = w_B(t')$. Suppose $A$ is $B$-maximal in $t$. Then $A$ is a cluster of $t$ and (since $A \subseteq B \subseteq B_0$) also a cluster of $t'$. Hence $A$ will be $B$-maximal for $t'$ unless the parent cluster in $t'$ of $A$ is a subset of $B$. Since $A$ is $B$-maximal in $t$, the parent cluster of $A$ in $t$ is not a subset of $B$. Note that $A$'s parent cluster in $t'$ is either the same cluster as in $t$, or else the parent cluster in $t'$ contains $A_0 \cup X$; in either case, the parent cluster of $A$ in $t'$ is not a subset of $B$. Therefore, $A$ is also $B$-maximal in $t'$.

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

Case 3: $B_0 \cup X \subseteq B$. Our first observation is that $A_0$ is $B$-maximal in $t$ if and only if $A_0 \cup X$ is $B$-maximal in $t'$. Hence, we need only concern ourselves with the $B$-maximal clusters in $t$ other than $A_0$, and (equally)

with the $B$-maximal clusters in $t'$ other than $A_0 \cup X$. However, when $A \neq A_0$, it is easy to see that $A$ is a $B$-maximal cluster in $t$ if and only if $A$ is a $B$-maximal cluster in $t'$. Hence, $w_B(t) = w_B(t')$.

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

The following lemma is obvious and the proof is omitted:

**Lemma 2.** *Let $t$ be an incomplete gene tree, $T$ a species tree, and $t'$ a completion of $t$ to the taxon set of $T$. Then $w_B(t) \leq w_B(t')$ for all clusters $B$ of $T$.*

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

*Proof.* Let $t \in \mathcal{T}$ be given, and let $T = MBMC_{inc}(\mathcal{T})$. By Lemma 2, for any completion $t'$ of $t$ and any cluster $B$ of $T$, $w_B(t') \geq w_B(t)$. By Lemma 1, there is a completion $t'$ of $t$ that achieves $w_B(t) = w_B(t')$ for all clusters $B$ of $T$. Since $t$ was arbitrary, we can let $\mathcal{T}'$ denote the set of completions of each $t \in \mathcal{T}$ so that $w_B(t) = w_B(t')$ for all clusters $B$ of $T$. Hence, the number of extra lineages in $T$ with respect to $\mathcal{T}'$ is $\sum_B \sum_i w_B(t)$, where $B$ ranges over the clusters $B$ of $T$ and $i$ ranges from 1 to $k$, where $\mathcal{T} = \{t_1, t_2, \ldots, t_k\}$. It follows, by Lemma 2, that $T$ has the minimum number of extra lineages with respect to any set of completions of $\mathcal{T}$, and so $(T, \mathcal{T}')$ is a solution to $MDC_{inc}(\mathcal{T})$.

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

## Materials and Methods

### Overview

The simulation study used gene sequences that evolve down gene trees that can differ from the true species tree due to ILS. To produce these sequence datasets, we used sequences used in previous studies and provided to us by the authors of these studies–the 11-taxon datasets from [30], the 17-taxon datasets from [29], and the 100-taxon datasets from [17]. We summarize the simulation protocols used in these studies here, and direct the reader to the relevant publication for the details of how the data were generated.

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

For each replicate dataset we had the true alignment as well as the true tree. We then deleted taxa randomly, varying the number of taxa removed, from each gene sequence alignment, thus producing incomplete gene sequence alignments. On each resultant gene sequence alignment we estimated trees using FastTree-2 [10]; this produces a tree as well as branch support estimations. We produced a 75%-branch support version of each estimated gene tree by contracting all edges with support below 75%.

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

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

- iGTP v. 1.1. We explore all three optimization criteria (deep coalescence, duplications, duplication-loss) available in iGTP. We ran iGTP on 75% support version of the input binary trees, although it is not guaranteed to give meaningful outputs for non-binary gene trees.

- Phylonet v. 2.4. We explore both heuristic and exact version of Phylonet used to solve the MDC problem on both binary and unresolved gene trees. However, the exact version can only be run on

small datasets, and so we used it only on the 11-taxon datasets.

- Matrix Representation with Parsimony (MRP). We ran MRP heuristics on the FastTree-2 trees (both binary and 75%-support versions), using a Python script to run a parsimony ratchet analysis using PAUP*, with 100 iterations, followed by taking the greedy consensus of the set of trees.

- *BEAST v. 1.6.2. We ran *BEAST on the true alignments for each dataset using its default settings.

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

### Datasets

We ran our experiments on datasets that evolve with ILS. We used 11-taxon datasets, each with 10 genes, obtained from [30]. We also used 17-taxon datasets with 8 genes each, used previously in [29]. Finally, we used 100-taxon datasets with 25 genes each, used in [17].

## Results

### Missing branch rates

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

#### *Results on 11-taxon datasets*

For these datasets, we were able to run the exact version of Phylonet-MDC, and hence solve the MDC problem exactly. As before, we ran the heuristic version of Phylonet-MDC, the three iGTP methods (for the MDC score, duplication score, and duplication plus losses score), and MRP. We explored results with two, three, and five missing taxa; see Figure 1.

The first observation is that *BEAST produced the most accurate species trees, for all percents of missing taxa. The second best method varied depending on the percentage of missing taxa, with MRP on the 75%-support trees best for 20% missing taxa, Phylonet-exact on the 75%-support trees best for 30% missing taxa, and MRP best for 50% missing taxa. Thus, there was no clear second best method. Furthermore, although these three methods generally gave reasonably good results, they were not always among the next most accurate. Between the iGTP methods, iGTP-dup had the worst results, and iGTP-MDC and iGTP-duploss were sometimes reasonably accurate. A noteworthy trend was that Phylonet-heuristic gave the worst results

at all percents of missing taxa, whether applied to the fully resolved trees or the 75%-support trees. Finally, using the 75%-support trees instead of the fully resolved trees improved MRP and Phylonet (both exact and heuristic) for small numbers of missing taxa, but not when the number of missing taxa was large. Also, using the 75%-support trees did not help the other methods.

***Results on 17-taxon datasets***

Performance on 17-taxon datasets with 8 genes showed similar results, see Figure 2. Because of the number of taxa, we did not run Phylonet-exact. However, the results we saw here are similar to what we saw on the 11-taxon datasets. As before, *BEAST was the most accurate, for all percents of missing taxa. The next best methods were MRP and iGTP-MDC (on either binary or 75%-support trees), and sometimes also iGTP-duploss on binary trees, but all had at least 7% higher missing branch rates than *BEAST. The worst results were obtained using Phylonet-heuristic and iGTP-dup on either the binary or 75%-support trees.

***Results on 100-taxon datasets***

We now describe results on the 100-taxon datasets. Because of the number of taxa, we did not run *BEAST (running long enough to reach convergence was infeasible for this experimental study), nor Phylonet-exact. However, these data allow us to compare the other methods, Phylonet-heuristic, the three variants of iGTP, and MRP, on both binary and 75%-support trees; see Figure 3.

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

***Overall results***

For all levels of missing data, certain trends were clearly seen. Results for all methods improved when given more estimated gene trees rather than fewer; these trends are to be expected, and consistent with prior studies (see, for example, [17]). In addition, we saw that for each species tree estimation method, the missing branch rate increased with increased levels of taxon deletion, but the increase in error was particularly large for the heuristic version of Phylonet-MDC.

The relative performance between methods showed clearly that when analyzing estimated gene trees,

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

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

### Computational Issues

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

## Discussion

We begin with some observations about methods that attempt to optimize the MDC criterion. First, it is clear that iGTP-MDC generally gives more accurate trees than Phylonet-MDC run in its heuristic mode; however, when the exact version of Phylonet-MDC can be run, it produces more accurate trees than its heuristic version, and also more accurate trees than iGTP-MDC. The reason for this is likely due to the

improved MDC scores produced by using the exact version of Phylonet-MDC (which are mathematically guaranteed), compared to the other methods. It is worth noting that the substantial reduction in topological accuracy (and MDC scores, results not shown) by using the heuristic version instead of the exact version of Phylonet-MDC is almost certainly a result of the fact that *all* the gene trees are incomplete, with randomly deleted taxa. This greatly impairs the ability of Phylonet-MDC's heuristic to score trees that are topologically similar to the true tree, since all the clades in any estimated tree must be drawn from the input gene tree clades in this case. However, the heuristic used in iGTP-MDC explicitly searches through treespace and so is not impaired in the same way. Given that previous research [17] has shown very good trees resulting from Phylonet-MDC's heuristic version when the input gene trees are all complete, it seems likely that Phylonet-MDC might give better results when the taxon deletion is not random, or when at least some of the gene trees are based upon complete taxon sets. Thus, although this study showed poor accuracy for Phylonet-MDC's heuristic, this trend may not hold under other circumstances, including those that might better represent systematic practice. Future work will investigate this possibility.

We also note that contracting low support branches in estimated gene trees typically (but not always) benefited Phylonet-MDC and MRP, but not the iGTP methods. This difference is likely due to differences in the treatment of unresolved gene trees within the iGTP, Phylonet-MDC, and MRP software. For example, it seems likely that iGTP-MDC and Phylonet-MDC do not score proposed species trees identically when the input gene trees are unresolved (Phylonet-MDC scores species trees with respect to optimal refinements of unresolved gene trees [27], a guarantee that may not be true of iGTP-MDC).

## Conclusions

This study establishes that there is currently no computationally feasible solution for estimating highly accurate species trees from incomplete gene trees for large numbers of taxa. That is, only *BEAST was able to produce highly accurate species trees; all other methods had much higher error rates. Therefore, for small enough numbers of taxa so that *BEAST can be run properly without huge running times, very accurate species trees can be computed. Although this study did not investigate the feasibility of running *BEAST on larger datasets, other studies with Bayesian methods have shown that proper analyses of datasets (even small ones) can take weeks of analysis to reach convergence [17]. Therefore, the poor results of the other methods on larger datasets suggests that highly accurate species tree estimations from incomplete gene trees and alignments may be beyond what current methods can achieve.

This study also suggests some limitations to analyses based upon MDC. Unsurprisingly, we saw that

optimizing MDC generally gave better results than optimizing duplications or duplications and losses. We also observed (as had been noted earlier in [17]) that optimizing the total number of duplications and losses produced more accurate trees than optimizing duplications alone.

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

Thus, this study showed that the standard heuristics (the parsimony ratchet as implemented in PAUP*) for the supertree method MRP produces highly accurate species tree estimations, even though it does not consider ILS, and can do so reasonably quickly, even on large datasets. These observations, combined with the observation that none of the methods we studied (other than *BEAST) that explicitly take into account events such as ILS or duplication and loss produced trees as accurate as MRP, suggest that optimizing MRP *may* be a reasonable approach to species tree estimation for large datasets, when statistical methods (such as *BEAST) cannot be run for computational reasons. Therefore, other supertree methods, such as SuperFine [31–33], a new supertree method that has been shown to produce better MRP scores and more accurate trees than standard MRP heuristics (while also being faster than standard MRP heuristics), should also be investigated.

## Author's contributions

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


## Acknowledgements

This research was supported by two grants from the US NSF (DEB 0733029 and DBI-1062335), the John Simon Guggenheim Memorial Foundation Fellowship to TW, a David Bruton Jr. Centennial Professorship to TW, and a 2010 Fulbright International Science and Technology PhD Award to MSB. The authors thank the anonymous reviewer for helpful suggestions.

## References


1. Liu K, Raghavan S, Nelesen S, Linder CR, Warnow T: **Rapid and Accurate Large-Scale Coestimation of Sequence Alignments and Phylogenetic Trees**. *Science* 2009, **324**(5934):1561–1564.
2. Liu K, Linder CR, Warnow T: **Multiple sequence alignment: a major challenge to large-scale phylogenetics**. *PLoS Currents: Tree of Life* 2010. [PMC2989897. Available from: http://knol.google.com/k/kevin-liu/multiple-sequence-alignment-a-major/%ectabesw3uba/9].
3. Wang LS, Leebens-Mack J, Wall P, Beckmann K, dePamphilis C, Warnow T: **The impact of multiple protein sequence alignment on phylogenetic estimation**. *IEEE Trans Comp Biol Bioinf (TCBB)* 2011, :1108–1119.
4. Katoh K, Kuma K, Miyata T, Toh H: **Improvement in the acccuracy of multiple sequence alignment MAFFT**. *Genome Informatics* 2005, **16**:22–33.
5. Katoh K, Kuma K, Toh H, Miyata T: **MAFFT version 5: improvement in accuracy of multiple sequence alignment**. *Nucleic Acids Res.* 2005, **33**(2):511–518.
6. Wheeler T, Kececioglu J: **Multiple alignment by aligning alignments**. In *Proceedings of the 15th ISCB Conference on Intelligent Systems for Molecular Biology* 2007:559–568.
7. Loytynoja A, Goldman N: **An algorithm for progressive multiple alignment of sequences with insertions**. *Proc. of the National Academy of Sciences* 2005, **102**:10557–10562.
8. Liu K, Warnow T, Holder M, Nelesen S, Yu J, Stamatakis A, Linder C: **SATé-II: Very Fast and Accurate Simultaneous Estimation of Multiple Sequence Alignments and Phylogenetic Trees**. *Syst Biol* 2011, **61**:90–106.
9. Stamatakis A: **RAxML-NI-HPC: Maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models**. *Bioinf* 2006, **22**:2688–2690.
10. Price MN, Dehal PS, Arkin AP: **FastTree 2 – Approximately maximum-likelihood trees for large alignments**. *PLoS ONE* 2010, **5**(3):e9490. doi:10.1371/journal.pone.0009490.
11. Zwickl D: **Genetic algorithm approaches for the phylogenetic analysis of large biological sequence datasets under the maximum likelihood criterion**. *PhD thesis*, The University of Texas at Austin 2006.
12. Guindon S, Dufayard J, Lefort V, Anisimova M, Hordijk W, Gascuel O: **New Algorithms and Methods to Estimate Maximum-Likelihood Phylogenies: Assessing the Performance of PhyML 3.0.** *Syst Biol* 2010, **59**(3):307–21.
13. Liu K, Linder C, Warnow T: **RAxML and FastTree: comparing two methods for large-scale maximum likelihood phylogeny estimation**. *PLoS-ONE* 2011, **6**(11):e27731.
14. Maddison WP: **Gene trees in species trees**. *Syst Biol* 1997, **46**:523–536.
15. Edwards SV: **Is a new and general theory of molecular systematics emerging?** *Evolution* 2009, **63**:1–19.
16. Degnan JH, Rosenberg NA: **Gene tree discordance, phylogenetic inference and the multispecies coalescent**. *Trends Ecology Evolution* 2009, **26**(6).
17. Yang J, Warnow T: **Fast and accurate methods for phylogenomic analyses**. *BMC Bioinformatics* 2011, **12**(4).
18. Larget B, Kotha SK, Dewey CN, Ané C: **BUCKy: Gene tree/species tree reconciliation with the Bayesian concordance analysis**. *Bioinf* 2010, **26**(22):2910–2911.
19. Ronquist F, Huelsenbeck JP: **MrBayes 3: Bayesian phylogenetic inference under mixed models**. *Bioinf* 2003, **19**:1572–1574.
20. Than CV, Ruths D, Nakhleh L: **PhyloNet: A Software Package for Analyzing and Reconstructing Reticulate Evolutionary Relationships**. *BMC Bioinf* 2008, **9**:322.
21. Heled J, Drummond AJ: **Bayesian inference of species trees from multilocus data**. *Mol Biol Evol* 2010, **27**:570–580.
22. Chaudhary R, Bansal MS, Wehe A, Fernández-Baca D, Eulenstein O: **iGTP: A software package for large-scale gene tree parsimony analysis**. *BMC Bioinf* 2010, **11**:574.
23. Baum B: **Combining trees as a way of combining data sets for phylogenetic inference, and the desirability of combining gene trees**. *Taxon* 1992, **41**:3–10.

24. Ragan MA: **Phylogenetic inference based on matrix representation of trees**. *Mol. Phylogenet. Evol.* 1992, **1**:53–58.

25. Swenson M, Barbançon F, Linder C, Warnow T: **A simulation study comparing supertree and combined analysis methods using SMIDGen**. *Algorithms for Molecular Biology* 2010, **5**:8. [PMID: 20047664].

26. Swenson M, Suri R, Linder C, Warnow T: **An experimental study of Quartets MaxCut and other supertree methods**. *Algorithms for Molecular Biology* 2011, **6**:7. [PMID: 21504600].

27. Yu Y, Warnow T, Nakhleh L: **Algorithms for MDC-based Multi-locus Phylogeny Inference: Beyond rooted binary gene trees on single alleles**. *J Comp Biol* 2011, **18**:1543–1559.

28. Swofford DL: *PAUP*: Phylogenetic Analysis Using Parsimony (and Other Methods), version 4.0*. Sinauer Associates 1996.

29. Than CV, Nakhleh L: **Species Tree Inference by Minimizing Deep Coalescences**. *PLoS Comp Biol* 2009, **5**(9).

30. Chung Y, Ané C: **Comparing two Bayesian methods for gene tree/species tree reconstruction: A simulation with incomplete lineage sorting and horizontal gene transfer**. *Syst Biol* 2011, **60**(3):261–275.

31. Swenson M, Suri R, Linder C, Warnow T: **SuperFine: fast and accurate supertree estimation**. *Syst Biol* 2011. [Syr092v1-syr092].

32. Neves D, Warnow T, Sobral J, Pingali K: **Parallelizing SuperFine**. In *27th Symposium on Applied Computing (ACM-SAC)* 2012.

33. Nguyen N, Mirarab S, Warnow T: **MRL and SuperFine+MRL: new supertree methods**. *Algorithms for Molecular Biology* 2012. in press.

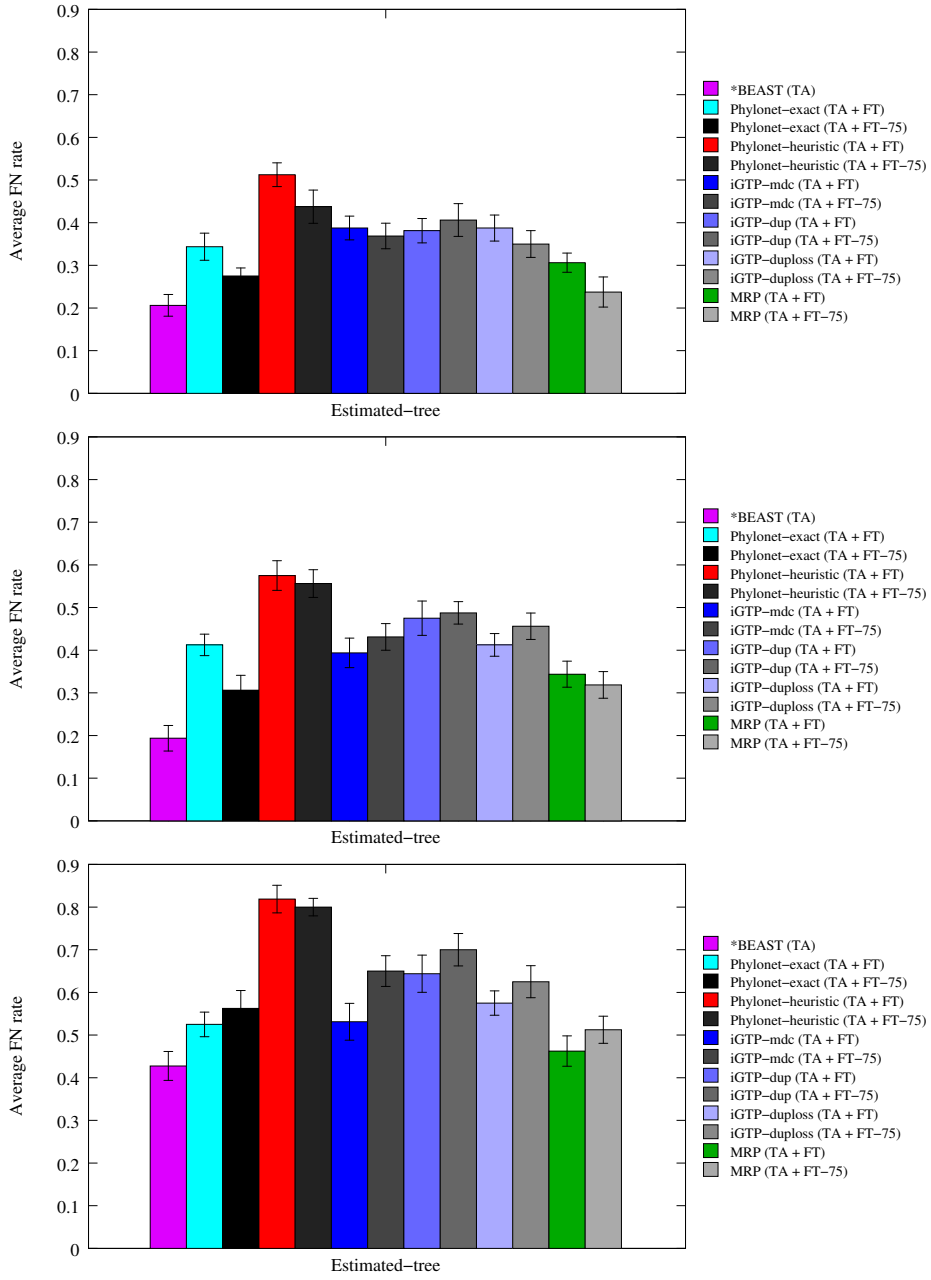


Figure 1: Average missing branch rates of methods on twenty (20) 11-taxon 10-gene datasets on true alignments (TA). Gene trees are estimated using FastTree-2 (FT), and in some cases the branches with support less than 75% are contracted (FT-75). From top to bottom, the number of missing taxa is 2, 3, and 5.

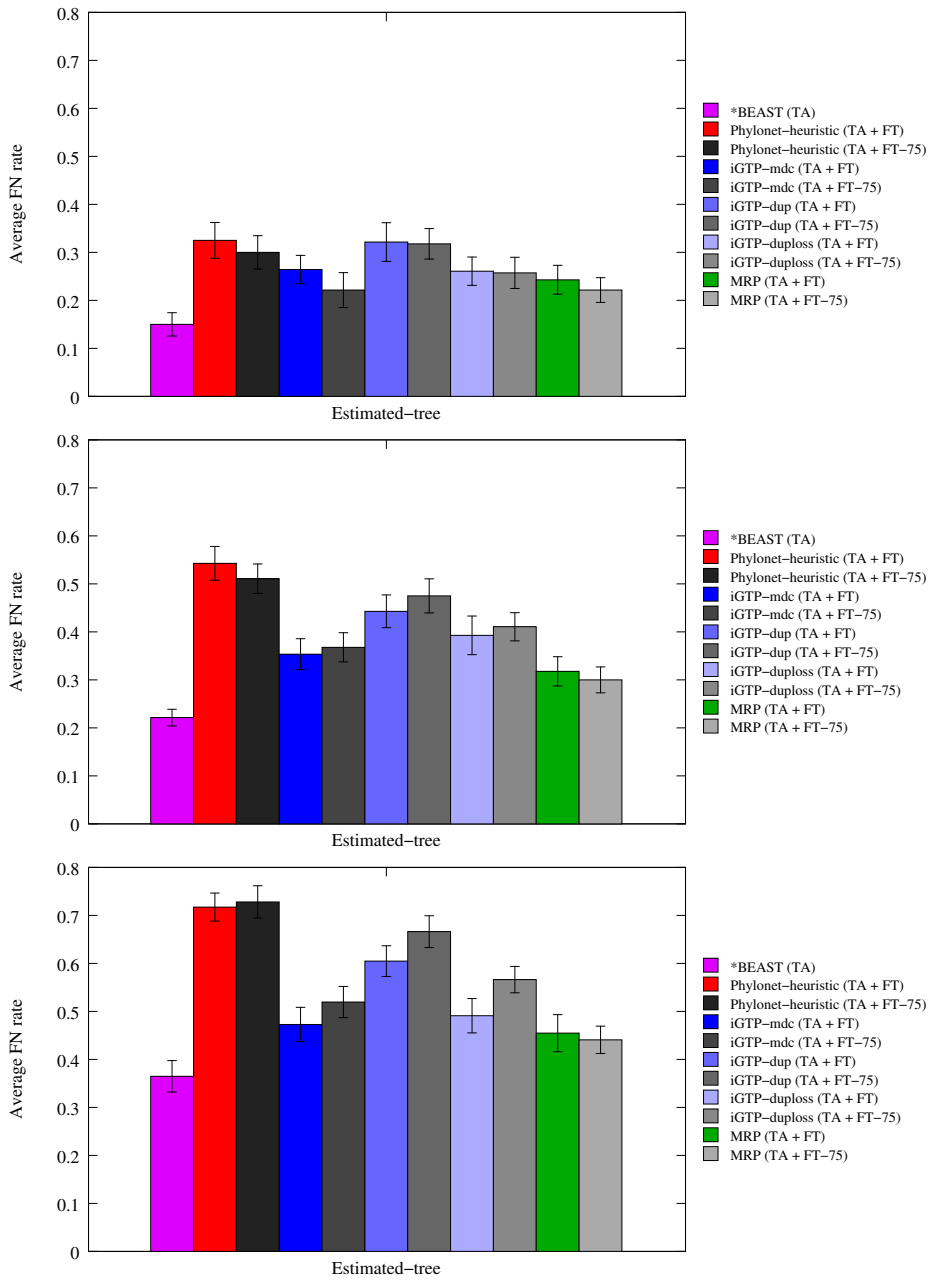


Figure 2: Average missing branch rates of methods on twenty (20) 17-taxon 8-gene datasets on true alignments (TA). Gene trees are estimated using FastTree-2 (FT), and in some cases the branches with support less than 75% are contracted (FT-75). From top to bottom, the number of missing taxa is 1, 5,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

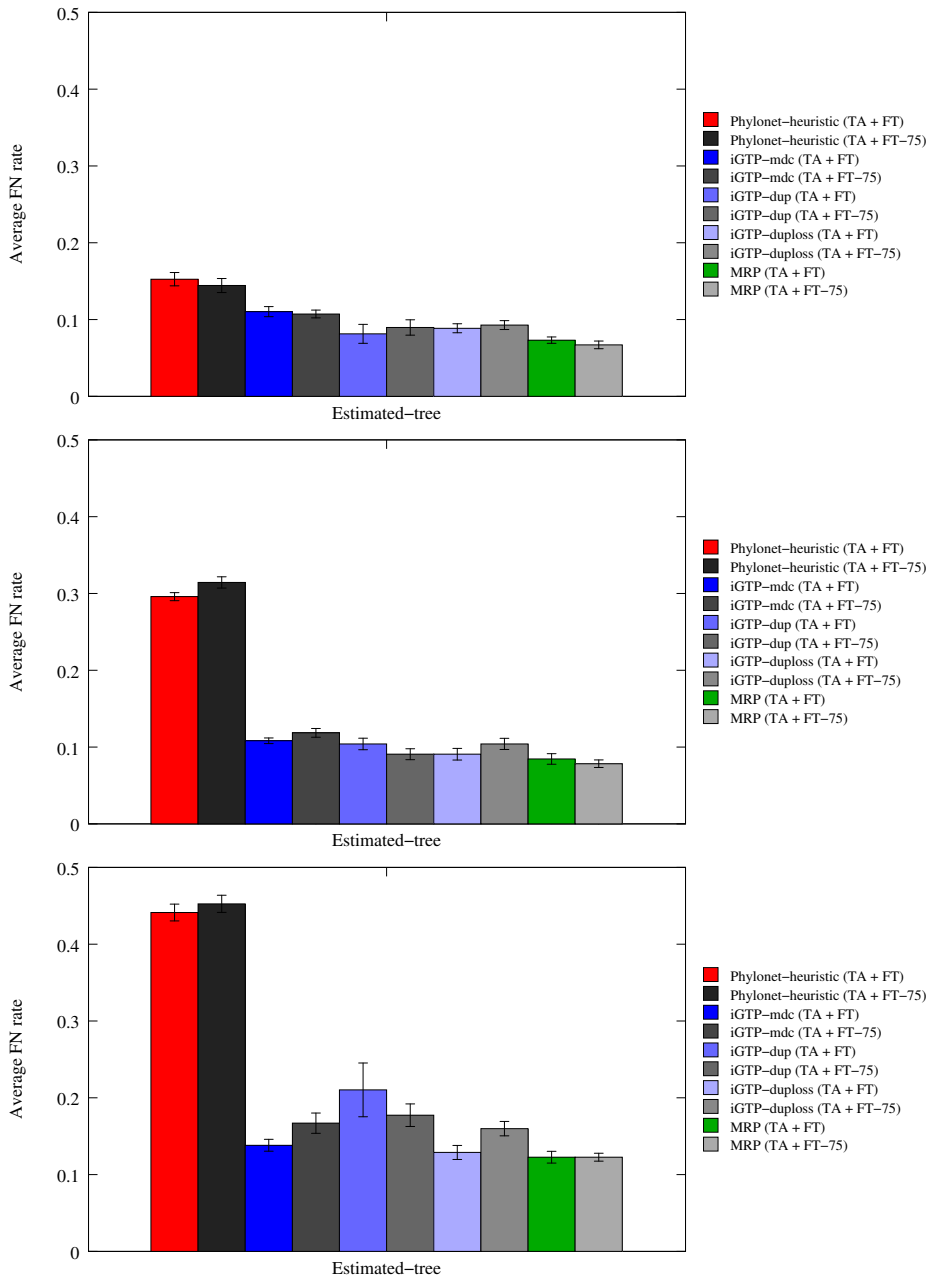


Figure 3: Average missing branch rates of methods on ten (10) 100-taxon 25-gene datasets on true alignments (TA). Gene trees are estimated using FastTree-2 (FT), and in some cases the branches with support less than 75% are contracted (FT-75). From top to bottom, the number of missing taxa is 10, 30, and 50.