Optimal Completion of Incomplete Gene Trees in Polynomial Time Using OCTAL*

Sarah Christensen[1], Erin K. Molloy[2], Pranjal Vachaspati[3], and Tandy Warnow[4]

1 **University of Illinois at Urbana-Champaign, Urbana, IL, USA**
sac2@illinois.edu
2 **University of Illinois at Urbana-Champaign, Urbana, IL, USA**
emolloy2@illinois.edu
3 **University of Illinois at Urbana-Champaign, Urbana, IL, USA**
vachasp2@illinois.edu
4 **University of Illinois at Urbana-Champaign, Urbana, IL, USA**
warnow@illinois.edu

Abstract

Here we introduce the *Optimal Tree Completion Problem*, a general optimization problem that involves completing an unrooted binary tree (i.e., adding missing leaves) so as to minimize its distance from a reference tree on a superset of the leaves. More formally, given a pair of unrooted binary trees (T, t) where T has leaf set S and t has leaf set $R \subseteq S$, we wish to add all the leaves from $S \setminus R$ to t so as to produce a new tree t' on leaf set S that has the minimum distance to T. We show that when the distance is defined by the Robinson-Foulds (RF) distance, an optimal solution can be found in polynomial time. We also present *OCTAL*, an algorithm that solves this RF Optimal Tree Completion Problem exactly in $O(|S|^2)$ time. We report on a simulation study where we complete estimated gene trees using a reference tree that is based on a species tree estimated from a multi-locus dataset. OCTAL produces completed gene trees that are closer to the true gene trees than an existing heuristic approach, but the accuracy of the completed gene trees computed by OCTAL depends on how topologically similar the estimated species tree is to the true gene tree. Hence, under conditions with relatively low gene tree heterogeneity, OCTAL can be used to provide highly accurate completions of estimated gene trees. We close with a discussion of future research.

1998 ACM Subject Classification G.2.1 Combinatorics, G.2.2 Graph Theory, J.3 Life and Medical Sciences

Keywords and phrases phylogenomics, missing data, coalescent-based species tree estimation, gene trees

Digital Object Identifier 10.4230/LIPIcs.WABI.2017.27

1 Introduction

Species tree estimation from multi-gene datasets is now increasingly common. One challenge is that the evolutionary history for a single locus (called a "gene tree") may differ from the species phylogeny due to a variety of different biological processes. Some of these processes,

* This work was partially supported by US National Science Foundation Graduate Research Fellowship Program under Grant Number DGE-1144245 to PV and EKM, US National Science Foundation grant CCF-1535977 to TW, and by the University of Illinois.

such as hybridization and horizontal gene transfer, result in non-treelike evolution and so require phylogenetic networks for proper analysis. However, other biological processes, such as gene duplication and loss, incomplete lineage sorting (ILS), and gene flow, produce heterogeneity across the genome but are still properly modeled by a single species tree [8]. In the latter case, species tree estimation methods must account for discordance or heterogeneity across the genome.

Much of the recent focus in the mathematical and statistical phylogenetics literature has been on developing methods for species tree estimation in the presence of incomplete lineage sorting (ILS), which is modelled by the multi-species coalescent (MSC) model [15]. One popular approach for estimating species trees under the MSC model is to estimate trees on individual loci and then combine these gene trees into a species tree. Some of these "summary methods", such as ASTRAL-II and ASTRID, have been shown to scale well to datasets with many taxa (i.e., >100 species) and provide accurate species tree estimates [12, 20].

A common challenge to species tree estimation methods is that sequence data may not be available for all genes and species of interest, creating conditions with missing data (see discussion in [6, 18]). For example, gene trees can be missing species simply because some species do not contain a copy of a particular gene; in some cases, no common gene will be shared by every species in the set of taxa [7]. Additionally, not all genomes may be fully sequenced and assembled, as this can be operationally difficult and expensive [3, 18].

Although summary methods are statistically consistent under the MSC model [1], the proofs of statistical consistency assume that all gene trees are complete, and so may not apply when the gene trees are missing taxa. Furthermore, multi-gene datasets with missing data can be "phylogenetically indecisive", meaning more than one tree topology can be optimal [16]. Because of concerns that missing data may reduce accuracy in multi-locus species tree estimation, many phylogenomic studies have restricted their analyses to only include genes with most of the species (see discussion by [6, 13, 18]).

Another way in which missing data impacts phylogenetics is that it is not that obvious how to evaluate the topological similarity between two trees when they are on different sets of species. A common approach is to constrain the two trees to the shared species and then compute the topological distance between the induced subtrees. However, it may be more interesting to ask how close the two trees could be if they were both completed (via the addition of the missing species) so that they are on the same species set.

Therefore, we formulate a class of optimization problems we refer to as Optimal Tree Completion problems, where we seek to add missing species to a tree to minimize the distance (defined in some way) to another tree. A natural version of this problem uses the Robinson-Foulds (RF) [14] distance between two trees, where the RF distance is the total number of unique bipartitions in the two trees. In Section 2, we formalize the RF Optimal Tree Completion problem. The Optimal Completion of incomplete gene Tree ALgorithm (or OCTAL) is a simple algorithm that incrementally adds the missing species one at a time into the tree, and which we prove solves the RF Optimal Tree Completion problem exactly. In Section 3, we present results from a experimental study on simulated datasets comparing OCTAL to a heuristic for tree completion within ASTRAL-II. Finally, we conclude with a discussion of results and future research in Section 6.

2 Optimal Tree Completion

2.1 Terminology

Each edge e in an unrooted phylogenetic tree defines a bipartition π_e on the leaves of the tree induced by the deletion of e (but not its endpoints). Each bipartition is thus a split $A|B$ of the leaf set into two non-empty parts, A and B. The set of bipartitions of a tree T is given by $C(T) = \{\pi_e : e \in E(T)\}$, where $E(T)$ is the set of edges for tree T. When two trees T and T' have the same leaf set, then the *Robinson-Foulds* (RF) distance [14] between T and T', denoted by $\mathrm{RF}(T, T')$, is $|C(T)\Delta C(T')|$. Thus, every bipartition in T or T' is either shared between the two trees or is unique to one tree, and the RF distance counts just the unique bipartitions. When two trees are binary and on the same leaf set, as is the case in this study, the numbers of bipartitions that are unique in each tree are equal, and each is half the RF distance.

Given tree T on leaf set S, T *restricted* to $R \subseteq S$, denoted by $T|_R$, is the minimal subgraph of T that connects all elements of R, suppressing nodes of degree two. Note that if T contains the bipartition $A|B$, $T|_R$ contains the restricted bipartition $(A \cap R)|(B \cap R)$. If T and T' are two trees with R as the intersection of their leaf sets, their *shared edges* are edges whose bipartitions restricted to R are in the set $C(T|_R) \cap C(T'|_R)$. Correspondingly, their *unique edges* are edges whose bipartitions restricted to R are not in the set $C(T|_R) \cap C(T'|_R)$.

2.2 The RF Optimal Tree Completion problem

The problem we address in this paper is the **RF Optimal Tree Completion Problem**, where the distance between trees is defined by the Robinson-Foulds distance, as follows:

- Input: An unrooted binary tree T on the full taxon set S and an unrooted binary tree t on a subset of taxa $R \subseteq S$
- Output: An unrooted binary tree T' on the full taxon set S with two key properties:
  1. T' is a *S-completion* of t (i.e., T' contains all the leaves of S and $T'|_R = t$) and
  2. T' minimizes the RF distance to T among all *S-completions* of t

Note that t and $T|_R$ are both on taxon set R, but need not be identical. In fact, the RF distance between these two trees is a lower bound on the RF distance between T and T'.

2.3 OCTAL: Optimal Completion of incomplete gene trees ALgorithm

The algorithm begins with input tree t and adds leaves one at a time from the set $S \setminus R$ until a tree on the full set of taxa S is computed. To add the first leaf, we choose an arbitrary taxon x to add from the set $S \setminus R$. We root the tree $T|_{R\cup\{x\}}$ (i.e., T restricted to the leaf set of t plus the new leaf being added) at x, and then remove x and the incident edge; this produces a rooted binary tree we will refer to as $T^{(x)}$ that has leaf set R.

We perform a depth-first traversal down $T^{(x)}$ until a shared edge e (i.e., an edge where the clade below it appears in tree t) is found. Since every edge incident with a leaf in $T^{(x)}$ is a shared edge, every path from the root of $T^{(x)}$ to a leaf has a distinct first edge e that is a shared edge. Hence, the other edges on the path from the root to e are unique edges.

After we identify the shared edge e in $T^{(x)}$, we identify the edge e' in t defining the same bipartition, and we add a new node $v(e')$ into t so that we subdivide e'. We then make x adjacent to $v(e')$. Note that since t is binary, the modification t' of t that is produced by adding x is also binary and that $t'|_R = t$. These steps are then repeated until all leaves from $S \setminus R$ are added to t. This process is shown in Fig. 1 and given in pseudocode below.

Algorithm 1: RF Optimal Tree Completion Algorithm (OCTAL)

procedure AddLeaf(Taxon x, binary tree T_1 on taxon set K, binary tree T_2 on taxon set $K \cup \{x\}$, set E of shared edges between T_1 and $T_2|_K$)
    Root T_2 at v, the neighbor of x, and delete x to produce a rooted version of $T_2|_K$
    Pick an arbitrary leaf y in $T_2|_K$ and find first edge $e \in E$ on path from v to y
    Find e' in T_1 defining the same bipartition as e
    Attach x to e' in T_1 by subdividing e' and making x adjacent to the newly created node; call the resulting tree T_1'
    return T_1'
end procedure

procedure OCTAL(Binary tree t on taxon set $R \subseteq S$, Binary tree T on taxon set S)
    if R=S **then**
        return t
    else
        $E \leftarrow$ Preprocess and initialize set of shared edges between t and $T|_R$
        $R' \leftarrow R$ ▷ Initialize R' by setting it equal to input R
        $t' \leftarrow t$ ▷ Initialize t' by setting it equal to input t
        for $x \in S \setminus R$ **do**
            $R' \leftarrow R' \cup \{x\}$
            $T' \leftarrow T|_{R'}$
            $t' \leftarrow$ AddLeaf(x, t', T', E)
            $E \leftarrow$ Update shared edges between t' and T'
        end for
        return t'
    end if
end procedure

We begin by preprocessing the two trees to identify shared edges; this takes $O(|S|^2)$ time. After this preprocessing is done, it is easy to see that *AddLeaf* takes $O(|S|)$ time to add a single taxon to t. Hence, OCTAL runs in $O(|S|^2)$ time, since there are $O(|S|)$ leaves to add.

2.4 Proof of Correctness

In what follows, let T be an arbitrary binary tree on taxon set S and t be an arbitrary binary tree on taxon set $R \subseteq S$. Let T' denote the tree returned by OCTAL given T and t. We set $r = RF(T|_R, t)$. As we have noted, OCTAL returns a binary tree T' that is an S-completion of t. Hence, to prove that OCTAL solves the RF Optimal Tree Completion problem exactly, we only need to establish that $RF(T, T')$ is the smallest possible of all binary trees on leaf set S that are S-completions of t. While the algorithm works by adding a single leaf at a time, we use two types of subtrees, denoted as *superleaves*, to aid in the proof of correctness.

▶ **Definition 1.** We define the superleaves of T with respect to t as follow (see Figure 2). The set of edges in T that are on a path between two leaves in R define the *backbone*; if this backbone is removed, the remainder of T breaks into pieces. The components of this graph that contain vertices from $S \setminus R$ are the **superleaves**. Each superleaf X is rooted at the node that was incident to one of the edges in the backbone, and is one of two types:

- Type I superleaves: the edge e in the backbone to which the superleaf was attached is a shared edge in $T|_R$ and t.
- Type II superleaves: the edge e in the backbone to which the superleaf was attached is a unique edge in $T|_R$ and t

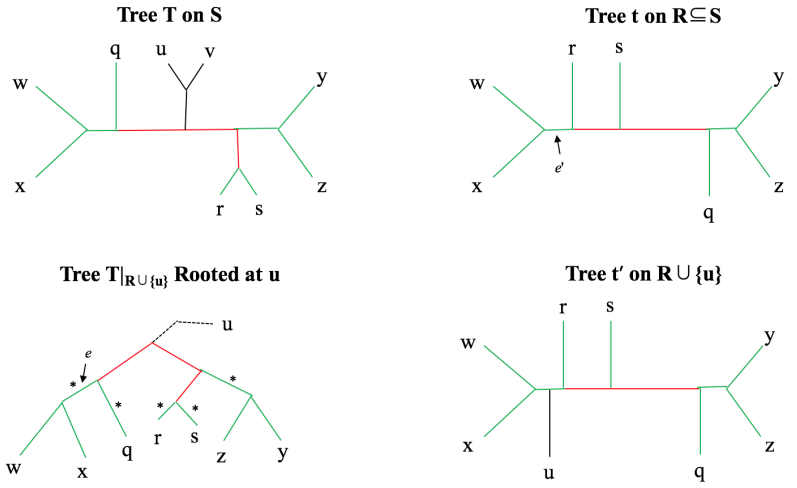

Figure 1 Trees T and t with edges in the backbone (defined to be the edges on paths between nodes in the common leaf set) colored green for shared, and red for unique; all other edges are colored black. After rooting $T|_R$ with respect to u, the edges in $T|_R$ that could be identified by the algorithm for "placement" are indicated with an asterisk (*). Note that any path in $T|_R$ from the root to a leaf will encounter a shared edge, since the edges incident with leaves are always shared. In this scenario, the edge e above the least common ancestor of leaves w and x is selected; this edge defines the same bipartition as edge e' in t. Hence, *AddLeaf* will insert leaf u into t by subdividing edge e', and making u adjacent to the newly added node.

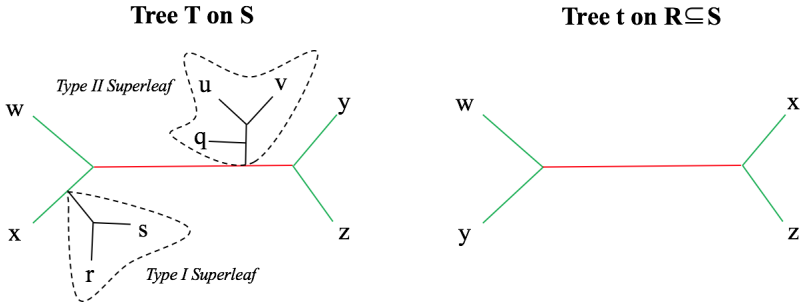

Figure 2 Trees T and t with edges in the backbone (defined to be the edges on paths between nodes in the common leaf set) colored green for shared, and red for unique; all other edges are colored black. The deletion of the backbone edges in T defines the superleaves; one is a Type I superleaf because it is attached to a shared (green) edge and the other is a Type II superleaf because it is attached to a unique (red) edge. The RF distance between t and $T|_R$ is equal to 2, the number of red edges. The Type I superleaf containing leaves r and s can be added to the shared edge incident to leaf x in tree t without increasing the RF distance. However, notice that adding the Type II superleaf to any edge in t creates at least one new unique edge in each tree and therefore increases the RF distance by at least 2, independent of how r and s are placed in t. This example motivates a more general result about elements of Type I and Type II superleaves proved in Section 2.4.

Observe that these intuitive definitions are equivalent to the following more formal definitions we sometimes invoke below. A superleaf X is a Type I superleaf if and only if there exists a bipartition $A|B$ in $C(t) \cap C(T|_R)$ where $A|(B \cup X)$ and $(A \cup X)|B$ are both in $C(T|_{R \cup X})$. Furthermore, a superleaf X is a Type II superleaf if and only if there does not exist such a bipartition in $C(t) \cap C(T|_R)$.

Now we begin our proof by establishing a lower bound on the RF distance to T for all binary, S-completions of t.

▶ **Lemma 2.** *Let Y be a Type II superleaf for the pair (T, t), and let $x \in S \setminus R$. Let t^* be the result of adding x into t arbitrarily (i.e., we do not attempt to minimize the resulting RF distance). If $x \notin Y$, then Y is a Type II superleaf for the pair (T, t^*). Furthermore, if $x \in Y$, then $RF(T|_{R \cup \{x\}}, t^*) \geq RF(T|_R, t) + 2$.*

Proof. It is easy to see that if $x \notin Y$, then Y remains a Type II superleaf after x is added to t. Now suppose $x \in Y$. We will show that we cannot add x into t without increasing the RF distance by at least 2. Since Y is a Type II superleaf, it is attached to a unique edge in $T|_{R \cup Y}$, and this is the same edge that x is attached to in $T|_{R \cup \{x\}}$. So suppose that x is added to t by subdividing an arbitrary edge e' in t with bipartition C|D; note that we do not require that x is added to a shared edge in t. After adding x to t we obtain tree t^* whose bipartition set includes $C|(D \cup \{x\})$ and $(C \cup \{x\})|D$. If C|D corresponds to a unique edge relative to t and $T|_R$, then both of these bipartitions correspond to unique edges relative to t^* and $T|_{R \cup \{x\}}$. If C|D corresponds to a shared edge, then at most one of the two new bipartitions can correspond to a shared edge, as otherwise we can derive that Y is a Type I superleaf. Hence, the number of unique edges in t must increase by at least one no matter how we add x to t, where x belongs to a Type II superleaf. Since t is binary, the tree that is created by adding x is binary, so that $RF(T|_{R \cup \{x\}}, t^*) \geq RF(T|_R, t) + 2$. ◀

▶ **Lemma 3.** *Let T^* be an unrooted binary tree that is a S-completion of t. Then $RF(T^*, T) \geq r + 2m$, where $r = RF(T|_R, t)$ and m is the number of Type II superleaves for the pair (T, t).*

Proof. We note that adding a leaf can never reduce the total RF distance. The proof follows from Lemma 2 by induction. ◀

Now that we have established a lower bound on the best achievable RF distance (i.e., the optimality criterion for the RF Optimal Tree Completion problem), we show OCTAL outputs a tree T' that is guaranteed to achieve this lower bound. We begin by noting that when we add x to t by subdividing some edge e', creating a new tree t', all the edges other than e' in t continue to "exist" in t' although they define new bipartitions. In addition, e' is split into two edges, which can be considered new. Thus, we can consider whether edges that are shared between t and T *remain* shared after x is added to t.

▶ **Lemma 4.** *Let t' be the tree created by AddLeaf given input tree t on leaf set R and tree T on leaf set $R \cup \{x\}$. If x is added to tree t by subdividing edge e' (thus creating tree t'), then all edges in t other than e' that are shared between t and T remain shared between t' and T.*

Proof. Let $T^{(x)}$ be the rooted tree obtained by rooting T at x and then deleting x. Let e be the edge in $T^{(x)}$ corresponding to e', and let $\pi_e = A|B$; without loss of generality assume A is a clade in $T^{(x)}$. Note that $C(T)$ contains bipartition $A|(B \cup \{x\})$ (however, $C(T)$ may not contain $(A \cup \{x\})|B$, unless e is incident with the root of $T^{(x)}$). Furthermore, for subclade $A' \subseteq A$, $A'|(R \setminus A') \in C(T|_R)$ and $A'|(R \setminus A' \cup \{x\}) \in C(T)$. Now suppose e^* in t is a shared edge between t and $T|_R$ that defines bipartition $C|D \neq A|B$. Since $A|B$ and $C|D$

are both bipartitions of t, without loss of generality either $C \subset A$ or $A \subset C$. If $C \subset A$, then C is a clade in $T^{(x)}$, and so e^* defines bipartition $C|(D \cup \{x\})$ within t'. But since $C \subset A$, the previous analysis shows that $C|(D \cup \{x\})$ is also a bipartition of T, and so e^* is shared between T and t'. Alternatively, suppose $A \subset C$. Then within t', e^* defines bipartition $(C \cup \{x\})|D$, which also appears as a bipartition in T. Hence, e^* is also shared between T and t'. Therefore, any edge e^* other than e' that is shared between t and T remains shared between t' and T, for all leaves x added by *AddLeaf*. ◀

▶ **Lemma 5.** *OCTAL(T, t) preserves the topology of superleaves in T.*

Proof. We will show this by induction on the number of leaves added. The lemma is trivially true for the base case when just one leaf is added to t. Let the inductive hypothesis be that the lemma holds for adding up to n leaves to t for some arbitrary $n \in \mathbb{N}^+$. Now consider adding $n+1$ leaves, and choose an arbitrary subset of n leaves to add to t, creating an intermediate tree t' on leaf set K using the algorithm OCTAL. Let x be the next additional leaf to be added by OCTAL.

If x is the first element of a new superleaf to be added, it is trivially true that the topology of its superleaf is preserved, but we need to show that x will not break the monophyly of an existing superleaf in t'. By the inductive hypothesis, the topology of each superleaf already placed in t' has been preserved. Thus, each superleaf placed in t' has some shared edge in t' and $T|_K$ incident to that superleaf. If x were placed onto an edge contained in some existing superleaf, that edge would change its status from being shared to being unique, which contradicts Lemma 4.

The last case is where x is part of a superleaf for the pair (T, t) that already has been added in part to t. *AddLeaf* roots $T|_{K\cup\{x\}}$ at x and removes the edge incident to x, creating rooted tree $T^{(x)}$. The edge incident to the root in $T^{(x)}$ must be a shared edge by the inductive hypothesis. Thus, OCTAL will add x to this shared edge and preserve the topology of the superleaf. ◀

▶ **Lemma 6.** *OCTAL(T, t) returns binary tree T' such that $RF(T, T') = r + 2m$, where m is the number of Type II superleaves for the pair (T, t) and $r = RF(T|_R, t)$.*

Proof. We will show this by induction on the number of leaves added.

Base Case: Assume $|S \setminus R| = 1$. Let x be the leaf in S$\backslash R$. *AddLeaf* adds x to a shared edge of t corresponding to some bipartition A|B, which also exists in $T^{(x)}$.

1. First we consider what happens to the RF distance on the edge x is attached to.

   If x is a Type I superleaf, the edge incident to the root in $T^{(x)}$ will be a shared edge by the definition of Type I superleaf, so *AddLeaf* adds x to the corresponding edge e' in t. The two new bipartitions that are created when subdividing e' will both exist in T by the definition of Type I superleaf so the RF distance does not change.

   If x is a Type II superleaf, either $(A \cup \{x\})|$B or $A|(B \cup \{x\})$ must not exist in $C(T)$. Since *AddLeaf* adds x to a shared edge, exactly one of those new bipartitions must exist in $C(T)$.
2. Now we consider what happens to the RF distance on the edges x is *not* attached to.

   Lemma 4 shows that *AddLeaf* (and therefore OCTAL) preserves existing shared edges between t and $T|_R$, possibly excluding the edge where x is added.

Thus, the RF distance will only increase by 2 if x is a Type II superleaf, as claimed.

Inductive Step: Let the inductive hypothesis be that the lemma holds for up to n leaves for some arbitrary $n \in \mathbb{N}^+$. Assume $|S \setminus R| = n + 1$. Now choose an arbitrary subset of leaves $Q \subseteq S \setminus R$, where $|Q| = n$, to add to t, creating an intermediate tree t' using the algorithm OCTAL. By the inductive hypothesis, assume t' is a binary tree with the RF distance between $T|_{Q \cup R}$ and t' equal to $r + 2m$, where m is the number of Type II superleaves in Q. *AddLeaf* adds the remaining leaf $x \in S \setminus R$ to a shared edge of t' and $T|_{Q \cup R}$.

1. Lemma 4 shows that *AddLeaf* (and therefore OCTAL) preserves existing shared edges between t' and $T|_{Q \cup R}$, possibly excluding the edge where x is added.
2. Now we consider what happens to the RF distance on the edge x is attached to. There are three cases: (i) x is not the first element of a superleaf (ii) x is the first element of a Type I superleaf or (iii) x is the first element of a Type II superleaf.

   Case (i): If x is not the first element of a superleaf to be added to t, it directly follows from Lemma 5 that OCTAL will not change the RF distance when adding x.

   Case (ii): If x is the first element of a Type I superleaf to be added, then x is attached to a shared edge in the backbone corresponding to some bipartition $A|B$ existing in both $C(t)$ and $C(T|_R)$. Let e' be the edge in t s.t. $\pi_{e'} = A|B$. Note there must exist an edge e in $T|_{Q \cup R}$ producing $A|B$ when restricted to just R. Hence, the bipartition π_e has the form $M|N$ where $(M \cap R) = A$ and $(N \cap R) = B$. We need to show that $M|N \in C(t')$.
   - By Lemma 4, any leaves from Q not attached to e' by OCTAL will preserve this shared edge in t'.
   - Now consider when leaves from Q are added to e' by OCTAL. We decompose M and N into the subsets of leaves existing in either R or Q: let $M = A \cup W$ and $N = B \cup Z$. OCTAL will not cross a leaf from W with a leaf from Z along e' because this would require crossing the shared edge dividing these two groups: any leaf $w \in W$ has the property that $(A \cup \{w\})|B$ is a shared edge and any leaf $z \in Z$ has the property that $A|(B \cup \{z\})$ is a shared edge. Hence, any leaves added from Q that subdivide e' will always preserve an edge between leaves contained in W and Z on e'.

   Thus, $M|N \in C(t')$. Moreover, $(M \cup \{x\})|N$ and $M|(N \cup \{x\})$ are bipartitions in $C(T)$. *AddLeaf* roots T at x and removes the edge incident to x, creating rooted tree $T^{(x)}$. We have shown that the edge incident to the root in $T^{(x)}$ must be a shared edge, so adding x does not change the RF distance.

   Case (iii): If x is the first element of a Type II superleaf to be added, we have shown in Lemma 2 that the RF distance must increase by at least two. Since *AddLeaf* always attaches x to some shared edge e', the RF distance increases by exactly 2 when subdividing e'.

Thus, OCTAL will only increase the RF distance by 2 if x is a new Type II superleaf. ◀

Combining the above results, we establish our main theorem:

▶ **Theorem 7.** *Given unrooted binary trees t and T with the leaf set of t a subset of the leaf set of T, OCTAL(T, t) returns an unrooted binary tree T' that is a completion of t and that has the smallest possible Robinson-Foulds distance to T. Hence, OCTAL finds an optimal solution to the RF Optimal Tree Completion problem. Furthermore, OCTAL runs in $O(n^2)$ time, where T has n leaves.*

Proof. The running time bound was described above in Section 2.3. To prove that OCTAL solves the RF Optimal Tree Completion problem optimally, we need to establish that OCTAL returns an S-completion of the tree t, and that the RF distance between the output tree T' and the reference tree T is the minimum among all S-completions. Since OCTAL always returns a binary tree and only adds leaves into t, by design it produces a completion of t and

so satisfies the first property. By Lemma 6, the tree T' output by OCTAL has an RF score that matches the lower bound established in Lemma 3. Hence, OCTAL returns a tree with the best possible score among all S-completions. ◀

3 Methods

We compare OCTAL to the heuristic used in ASTRAL-II [12] for completing incomplete gene trees, as described in [10], noting however that the ASTRAL-II technique is used to expand the search space explored by ASTRAL-II and does not explicitly attempt to minimize the distance to a reference tree.

Datasets used in this simulation study have 26 species (one outgroup) and 200 genes. The simulation protocol is as follows.

(1) SimPhy [9] was used to simulate a model species tree and a collection of gene trees (with branch lengths deviating from a molecular clock) under the MSC model. Note that we refer to these simulated trees as the "true" gene and species trees. Under this process, the true gene trees differ topologically from the true species tree due only to ILS and not to any other processes.

(2) For each individual true gene tree, INDELible [5] was used to simulate DNA sequences under the GTR+Γ model of evolution without insertions or deletions. Model parameters varied across the gene trees and were determined by drawing from a distribution.

(3) Of the 200 genes, 150 genes were randomly selected to be missing data, created by deleting the sequences corresponding to randomly selected species. The number of species missing varies across gene trees from 2 to 20, and on average the estimated gene trees were missing approximately 60% of the species.

(4) RAxML [17] was then used to estimate gene trees from each (often incomplete) gene alignment under the GTRGAMMA model.

(5) ASTRID [20] was run on the 200 estimated gene trees to get a fast estimate of the species tree to be used as a reference tree by OCTAL.

(6) OCTAL (using the ASTRID tree as a reference) and ASTRAL-II were used to complete the estimated gene trees.

(7) The completed gene trees computed by OCTAL and ASTRAL-II were compared to the true gene trees, and the normalized RF error distance between the 150 completed gene trees and the true gene trees was recorded.

Overall we completed 6000 gene trees using OCTAL and ASTRAL-II for this study (20 replicates for 2 model conditions, and each replicate has 150 genes that are incomplete).

These datasets were originally generated for the ASTRAL-II study [12], and full details of this protocol (Steps 1 and 2) are provided in [12]. Data can be downloaded at [4]. OCTAL was scripted using the Python library DendroPy [19] and can be found at https://github.com/pranjalv123/OCTAL-2.

We explored two model conditions which vary in the degree of gene tree heterogeneity due to ILS. The two different levels of ILS can be characterized by the average normalized topological distance (AD) between true gene trees and the true species tree. The moderate ILS condition has AD of 10%, and the high ILS condition has AD of 35%. There were 20 replicate datasets for each of the two model conditions. We used one-sided paired Wilcoxon Signed-Rank tests to determine whether using OCTAL was significantly better than ASTRAL-II on each replicate dataset (200 genes). As 20 replicate datasets were tested per model condition, a Bonferroni multiple comparison correction was applied (i.e., p-values indicating significance are less than 0.0025).

4 Results

Moderate ILS (10% AD): OCTAL frequently produced more accurate gene trees than ASTRAL-II: the average RF error rate for ASTRAL-II was 0.18 ± 0.10 and the average RF error rate for OCTAL was 0.16 ± 0.09. OCTAL had better accuracy than ASTRAL-II on 1,247 genes, ASTRAL-II had better accuracy on 319, and the methods were tied on the remaining 1,434 genes. The degree of improvement in RF rate varied, but was as great as 20% on some replicates. The improvement obtained by using OCTAL over ASTRAL-II was statistically significant in 18 out of 20 of the replicates (Fig. 3). The degrees of missing data and gene tree error did not impact whether OCTAL improved over ASTRAL-II (Fig. 4).

High ILS (35% AD): OCTAL and ASTRAL-II achieved similar levels of accuracy on high ILS condition: the average RF error rate for ASTRAL-II was 0.39 ± 0.11 and the average RF error rate for OCTAL was 0.38 ± 0.11. OCTAL was more accurate than ASTRAL-II on 945 genes, ASTRAL-II was more accurate on 568 genes, and the methods were tied on the remaining 1,487 genes. OCTAL provided a statistically significant advantage over ASTRAL-II in 8 of the 20 replicates, and the differences between the two methods was not statistically significant on the remaining 12 replicates (Fig. 5). Similar to the moderate ILS condition, whether OCTAL or ASTRAL-II performed best appears to be unrelated to the degree of missing data or gene tree error (Fig. 6).

5 Discussion

These results show that OCTAL can be more accurate than ASTRAL-II at completing gene trees and that the degree and frequency of improvement depends on the level of ILS. Under low to moderate ILS, a reasonably accurate estimate of the species tree will be close to the true gene trees, and hence be useful as a reference tree for OCTAL. However, under higher ILS, estimated species trees will be further from the true gene trees, impairing their utility as reference trees for OCTAL.

Therefore, it may make sense to only use OCTAL in conditions with sufficiently low gene tree heterogeneity (i.e., when ILS levels are at most moderate), so that the computed reference tree is topologically close to the gene trees. However, another strategy is to define a set of reference trees rather than a single reference tree, and complete each gene tree based on an appropriately selected reference tree. Statistical binning [11] and its variant weighted statistical binning [2] are examples of this kind of technique. In statistical binning, the genes are clustered into sets (called "bins") on the basis of gene tree similarity taking bootstrap support into account. Then, for each bin, a concatenated maximum likelihood tree is computed and used as the new gene tree for the genes in the bin. As shown in [11], statistical binning improves gene tree estimation and leads to improved species tree estimation in downstream analyses using summary methods. The same basic approach could be combined with OCTAL, so that after clustering the genes, a reference tree for the genes in each bin could be computed based only on the genes in the bin, and then the gene trees could be completed using the reference tree for their bin. Given the high accuracy obtained by OCTAL when the gene tree heterogeneity is low, this may well produce improved accuracy when the overall heterogeneity is high.

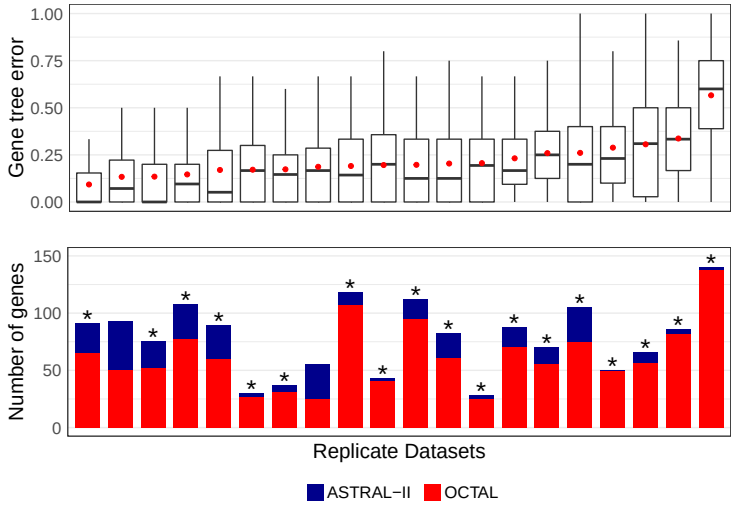

Figure 3 The relative performance of OCTAL and ASTRAL-II under the moderate ILS condition (10% AD) is shown. The top subfigure shows the amount of gene tree estimation error in each of the 150 completed genes for each of the 20 replicates. The bottom subfigure shows the relative performance of OCTAL and ASTRAL-II. The number of gene trees for which OCTAL is better than ASTRAL-II is shown in red, the number of gene trees for which ASTRAL-II is better is shown in blue, and ties are indicated by empty space. OCTAL has a statistically significant improvement over ASTRAL-II on replicates indicated with an asterisk (*).

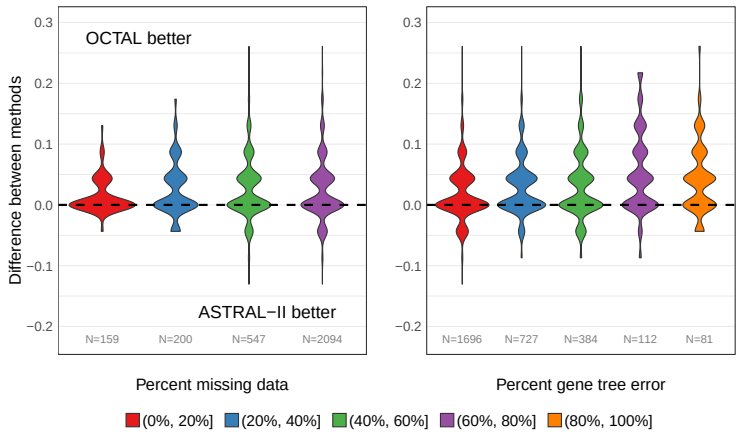

Figure 4 The relative performance of OCTAL and ASTRAL-II under the moderate ILS condition (10% AD) is shown. The y-axis shows the difference in the RF error rate between trees completed using OCTAL and ASTRAL-II. Positive values indicate that OCTAL is better than ASTRAL-II, and negative values indicate that ASTRAL-II is better. The violin plots show that for many genes (indicated by thickness of the violin plot), there is no difference in accuracy between OCTAL and ASTRAL-II. However, when there is a difference between the two methods, OCTAL frequently outperforms ASTRAL-II. This finding holds regardless of the degree of missing data or amount of gene tree estimation error. In the left subfigure, each violin plot includes genes with a certain percent of missing data, e.g., red indicates genes are missing 0-20% of the species. In the right subfigure, each violin plot includes genes with a certain percent gene tree estimation error. The number of genes in each violin plot is provided on the x-axis.

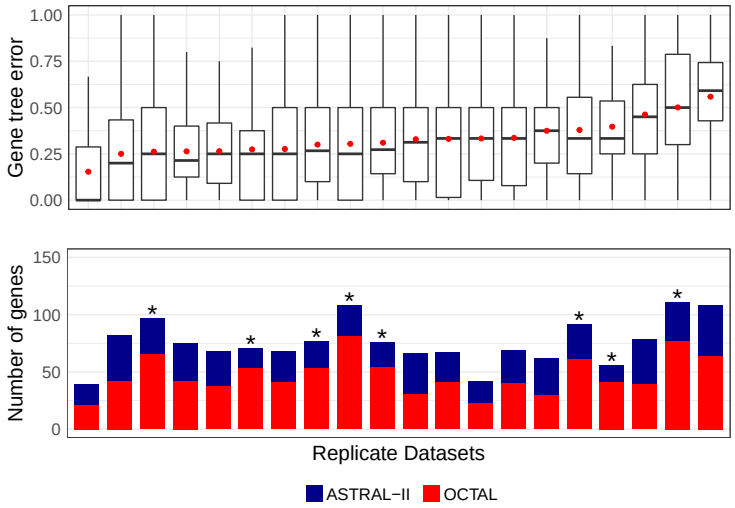

Figure 5 The relative performance of OCTAL and ASTRAL-II under the high ILS condition (35% AD) is shown. The top subfigure shows the amount of gene tree estimation error in each of the 150 completed genes for each of the 20 replicates. The bottom subfigure shows the relative performance of OCTAL and ASTRAL-II. The number of gene trees for which OCTAL is better than ASTRAL-II is shown in red, the number of gene trees for which ASTRAL-II is better is shown in blue, and ties are indicated by empty space. OCTAL has a statistically significant improvement over ASTRAL-II on replicates indicated with an asterisk (*).

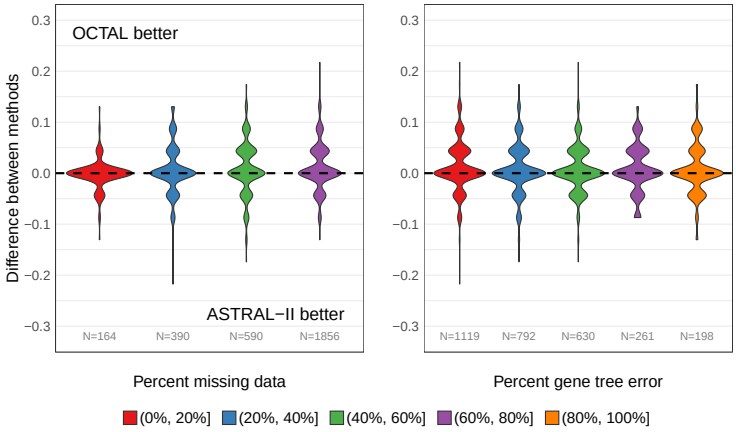

Figure 6 The relative performance of OCTAL and ASTRAL-II under the high ILS condition (35% AD) is shown. The y-axis shows the difference in the RF error rate between the 150 gene trees computed using OCTAL and ASTRAL-II. Positive values indicate that OCTAL is better than ASTRAL-II, and negative values indicate that ASTRAL-II is better. In the left subfigure, each violin plot includes genes with a certain percent of missing data, e.g., red indicates genes are missing 0-20% of the species. In the right subfigure, each violin plot includes genes with a certain percent gene tree estimation error. The number of genes in each violin plot is provided on the x-axis.

6 Conclusions

OCTAL is a simple polynomial time algorithm that can add species into an estimated gene tree and minimize the RF distance with respect to a reference tree. As we saw, OCTAL frequently produces more accurate completed gene trees than ASTRAL-II under both moderate and high ILS conditions; however, the improvement under high ILS conditions is much lower and less frequent than under the low to moderate ILS condition. The results shown here suggest that OCTAL (or some modification of OCTAL) might be useful for coalescent-based species tree estimation using summary methods, especially for those summary methods that are impacted by missing data. As OCTAL only adds missing species and does not provide statistical support for the placements, future work should address this issue. In addition, the current approach assumes that the gene tree is accurate, but typically gene trees have some estimation error; hence, another approach would allow the low support branches in gene trees to be collapsed and then seek a complete gene tree that refines the collapsed gene tree. Finally, this paper addresses tree completion when the distance to be minimized was the Robinson-Foulds distance; yet, many other tree distances have been considered. For example, the Minimize Deep Coalescence (MDC) distance [8] between two trees is a measure of the amount of incomplete lineage sorting, and adding missing species to produce the minimum MDC distance may be more suitable than minimizing the RF distance when datasets have high ILS.

Acknowledgements The authors would like to thank Michael Nute for helpful discussions regarding statistical testing. SC and TW are supported by National Science Foundation Grant Number CCF-1535977. SC is also supported by the Chirag Foundation Graduate Fellowship in Computer Science. EKM and PV are supported by the National Science Foundation Research Fellowship Program under Grant Number DGE-1144245. This research made use of the Illinois Campus Cluster, a computing resource that is operated by the Illinois Campus Cluster Program in conjunction with the National Center for Supercomputing Applications and which is supported by funds from the University of Illinois at Urbana-Champaign.

References

1 Elizabeth S. Allman, James H. Degnan, and John A. Rhodes. Split Probabilities and Species Tree Inference under the Multispecies Coalescent Model. *arXiv:1704.04268*, 2017.

2 Md. Shamsuzzoha Bayzid, Siavash Mirarab, Bastien Boussau, and Tandy Warnow. Weighted statistical binning: enabling statistically consistent genome-scale phylogenetic analyses. *PLOS One*, 10(6):30129183, 2015. doi:10.1371/journal.pone.0129183.

3 J. Gordon Burleigh, Khidir W. Hilu, and Douglas E. Soltis. Inferring Phylogenies with Incomplete Data Sets: A 5-gene, 567-taxon analysis of angiosperms. *BMC Evolutionary Biology*, 9(1):61, 2009. doi:10.1186/1471-2148-9-61.

4 Sarah Christensen, Erin Molloy, Pranjal Vachaspati, and Tandy Warnow. Datasets from the study: Optimal completion of incomplete gene trees in polynomial time using OCTAL, 2017. doi:10.13012/B2IDB-8402610_V1.

5 William Fletcher and Ziheng Yang. INDELible: A Flexible Simulator of Biological Sequence Evolution. *Molecular Biology and Evolution*, 26(8):1879–1888, 2009. doi:10.1093/molbev/msp098.

6 Peter A. Hosner, Brant C. Faircloth, Travis C. Glenn, Edward L. Braun, and Rebecca T. Kimball. Avoiding Missing Data Biases in Phylogenomic Inference: An Empirical Study in the Landfowl (Aves: Galliformes). *Molecular Biology and Evolution*, 33(4):1110–1125, 2016. doi:10.1093/molbev/msv347.

7 Martyn Kennedy and Roderic D. M. Page. Seabird Supertrees: Combining Partial Estimates of Procellariiform Phylogeny. *The Auk*, 119(1):88–108, 2002. doi:10.1642/0004-8038(2002)119[0088:SSCPEO]2.0.CO;2.

8 Wayne Maddison. Gene Trees in Species Trees. *Systematic Biology*, 46(3):523–536, 1997. doi:10.1093/sysbio/46.3.523.

9 Diego Mallo, Leonardo De Oliveira Martins, and David Posada. SimPhy: phylogenomic simulation of gene, locus, and species trees. *Systematic biology*, 65(2):334–344, 2016. doi:doi:10.1093/sysbio/syv082.

10 Siavash Mir arabbaygi (Mirarab). *Novel Scalable Approaches for Multiple Sequence Alignment and Phylogenomic Reconstruction*. PhD thesis, The University of Texas at Austin, 2015. URL: http://hdl.handle.net/2152/31377.

11 Siavash Mirarab, Md. Shamsuzzoha Bayzid, Bastien Boussau, and Tandy Warnow. Statistical binning enables an accurate coalescent-based estimation of the avian tree. *Science*, 346(6215), 2014. doi:10.1126/science.1250463.

12 Siavash Mirarab and Tandy Warnow. ASTRAL-II: Coalescent-based Species Tree Estimation with Many Hundreds of Taxa and Thousands of Genes. *Bioinformatics*, 31(12):i44, 2015. doi:10.1093/bioinformatics/btv234.

13 Erin Molloy and Tandy Warnow. To include or not to include: The impact of gene filtering on species tree estimation methods. *bioRxiv*, 2017. doi:10.1101/149120.

14 David F. Robinson and Leslie R. Foulds. Comparison of Phylogenetic Trees. *Mathematical Biosciences*, 53(1-2):131–147, 1981. doi:10.1016/0025-5564(81)90043-2.

15 Sébastien Roch and Mike Steel. Likelihood-based Tree Reconstruction on a Concatenation of Alignments can be Positively Misleading. *arXiv:1409.2051*, 2014.

16 Michael J. Sanderson, Michelle M. McMahon, and Mike Steel. Phylogenomics with incomplete taxon coverage: the limits to inference. *BMC Evolutionary Biology*, 10, 2010. doi:10.1186/1471-2148-10-155.

17 Alexandros Stamatakis. RAxML Version 8: A tool for Phylogenetic Analysis and Post-Analysis of Large Phylogenies. *Bioinformatics*, 30(9), 2014. doi:10.1093/bioinformatics/btu033.

18 Jeffrey W. Streicher, James A. Schulte, II, and John J. Wiens. How Should Genes and Taxa be Sampled for Phylogenomic Analyses with Missing Data? An Empirical Study in Iguanian Lizards. *Systematic Biology*, 65(1):128, 2016. doi:10.1093/sysbio/syv058.

19 Jeet Sukumaran and Mark T. Holder. Dendropy: a Python library for phylogenetic computing. *Bioinformatics*, 26(12):1569–1571, 2010. doi:10.1093/bioinformatics/btq228.

20 Pranjal Vachaspati and Tandy Warnow. ASTRID: Accurate Species Trees from Internode Distances. *BMC Genomics*, 16(10):S3, 2015. doi:10.1186/1471-2164-16-S10-S3.