Research Article:

A Polynomial-Time Algorithm for Minimizing the Deep Coalescence Cost for Level-1 Species Networks

Matthew LeMay[1],
Ran Libeskind-Hadas[2],
Yi-Chieh Wu[2],

1. Department of Mathematics, Harvey Mudd College, Claremont, California 91711, USA
2. Department of Computer Science, Harvey Mudd College, Claremont, California 91711, USA

[*] **Corresponding author:**
Yi-Chieh Wu
301 Platt Blvd
Claremont, California 91711
Phone: 909-621-8127
E-mail: yjw@cs.hmc.edu

Abstract: Phylogenetic analyses commonly assume that the species history can be represented as a tree. However, in the presence of hybridization, the species history is more accurately captured as a network. Despite several advances in modeling phylogenetic networks, there is no known polynomial-time algorithm for parsimoniously reconciling gene trees with species networks while accounting for incomplete lineage sorting. To address this issue, we present a polynomial-time algorithm for the case of level-1 networks, in which no hybrid species is the direct ancestor of another hybrid species. This work enables more efficient reconciliation of gene trees with species networks, which in turn, enables more efficient reconstruction of species networks.

Keywords: phylogenetics, reconciliation, deep coalescence, hybridization

1 Introduction

Reconstructing the evolutionary histories of a group of species is a fundamental step in phylogenetic analysis. While it is possible to infer trees from whole-genome alignments or from concatenated alignments, a common approach relies on first reconstructing individual *gene trees*, then reconstructing a *species tree* from the gene trees. However, gene trees and species trees may be incongruent due to various evolutionary processes, thus requiring *reconciliation* methods that map a gene tree "within" a species tree and explain topological differences by postulating a sequence of evolutionary events, with different models allowing for different types of events.

In the popular *multispecies coalescent (MSC) model* (Maddison 1997), species are treated as populations of individuals, and incongruence is assumed to be caused by *incomplete lineage sorting (ILS)* (Figure 1a,b). Formally, two lineages may fail to coalescence at their most recent opportunity, a phenomenon known as *deep coalescence*. ILS occurs when one lineage then coalesces with a lineage from a less closely-related population (Degnan and Rosenberg 2009).

Coalescent theory allows for computing the probability of a gene tree topology given a species tree topology and parameters such as population size and divergence time (Kingman 1982, Wakeley 2008). Thus, given multiple gene trees, it is possible to infer a species tree using either probabilistic or parsimony approaches (see Degnan and Rosenberg (2009) for a review of such methods). Probabilistic approaches rely on maximum likelihood or Bayesian estimation, whereas a parsimony approach chooses a species tree by minimizing deep coalescences (MDC), which "minimizes the number of extra lineages that had to coexist along species lineages" (Maddison 1997). In general, parsimony approaches require only topologies and are more efficient than probabilistic approaches, and thus are more broadly applicable.

However, the MSC model commonly assumes that species histories can be represented as a tree and therefore cannot account for hybridization (Figure 1c), in which separate species exchange genetic information, either through introgression, in-breeding between separate populations, or hybrid speciation (Elworth et al. 2019, Runemark et al. 2019). Studies have shown that hybridization can play a role in the evolution of eukaryotic species (Mavárez et al. 2006, Fontaine et al. 2015, Racimo et al. 2015, Lamichhaney et al. 2018).

In the last decade, several algorithms have been developed to infer species networks by simultaneously modelling ILS and hybridization. In a species network, species branches can join together at *hybridization nodes* (also known as *reticulation nodes*). As with the simpler MSC model, there exist both probabilistic (Meng and Kubatko 2009, Kubatko 2009, Yu et al. 2011a; 2012; 2013b; 2014, Wen and Nakhleh 2017, Zhang et al. 2017) and parsimony approaches (Yu et al. 2011a; 2013a;b) for inferring species networks under these models. Many of the parsimony approaches rely on converting a species network to a multi-labeled tree (MUL-tree), then considering all mappings of alleles to their sampled species. Because there can exist an exponential number of allele mappings, such approaches may not scale to large numbers of species or hybridizations.

Rather than model ILS and hybridization, some models instead allow for ILS and horizontal gene transfer, often with gene duplication and loss (Stolzer et al. 2012, Chan et al. 2017). However, such models also assume the species history can be represented as a tree and that gene transfers result in gene trees that are incongruent with the species tree. In contrast, by relying on a species network rather than a tree, hybridization allows different segments of the gene tree to have different histories naturally by using different edges leading to a hybridization node.

In parallel with these advances in ILS and hybridization, To and Scornavacca (2015) developed two

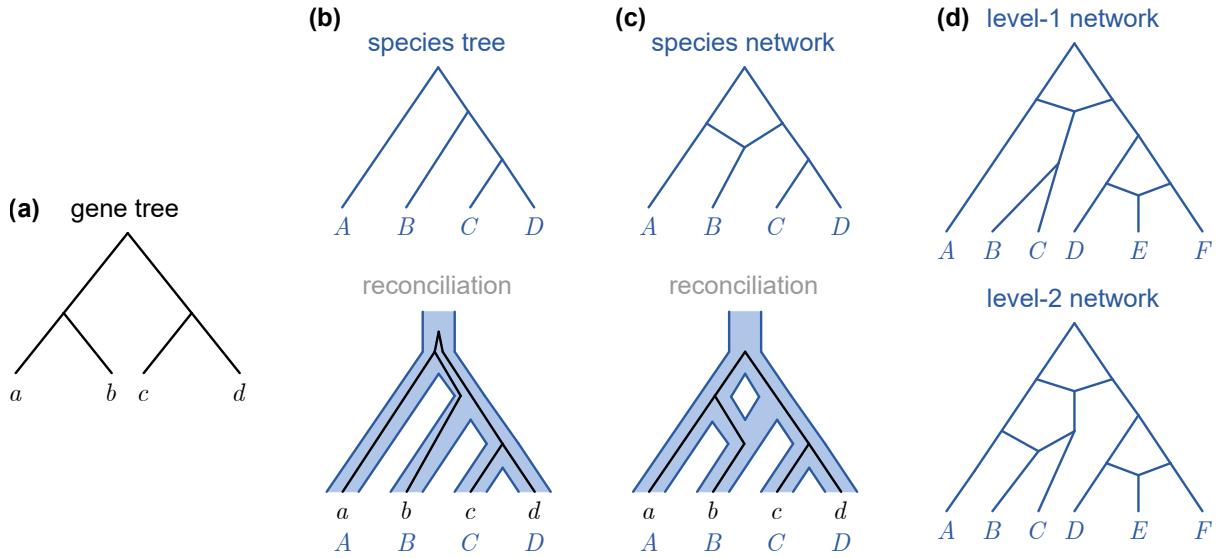

Figure 1. Gene trees, species trees, and species networks. (a) A gene tree. (b) A species tree and reconciliation. Under the multispecies coalescent model, the gene tree evolves within the species tree, and incongruence between the trees is due to ILS. (c) A species network and reconciliation. The same gene tree evolves within the species network, and no ILS is necessary. (d) A level-1 species network and a level-2 species network.

algorithms for reconciling gene trees and species networks that take into account duplication and loss events. They studied two variations: first, finding an optimal tree in a network such that the reconciliation of the gene tree and the "displayed" species tree has minimum cost, and second, finding a minimum cost reconciliation between the gene tree and the full species network. Interestingly, the time complexity of their first algorithm depends not on the number of hybridization events but on a parameter of the network called its *level* (Choy et al. 2005a), intuitively a measure of "how much the network is 'tangled"' (To and Scornavacca 2015) or how densely its hybridization nodes are distributed (Figure 1d). This algorithm is fixed-parameter tractable when parameterized by the level of the network and the number of biconnected components in the network. Their second algorithm is polynomial in the number of hybridization nodes, the size of the gene tree, and the size of the species network.

Despite these advances, there is currently no known polynomial-time algorithm for inferring an MDC reconciliation between a gene tree G and a species network S. We address this challenge with the following contributions:

1. We present a $O(|G|\cdot|S|^5)$ algorithm for reconciling a gene tree G and species network S when S has one hybridization node. Like many parsimony approaches, our algorithm relies on dynamic programming. Our key insight is to introduce a new parameter of the reconciliation, the *signature*, which specifies which hybridization edges are used by different parts of the reconciliation.

2. We reduce the time complexity of the previous algorithm to $O(|G|\cdot|S|^2)$ by generalizing the concept of a single lowest common ancestor (LCA) in trees to multiple LCAs in networks.

3. We present a $O(|G|\cdot|S|^2)$ algorithm for reconciling G and S when S is a level-1 network. Intuitively, in a level-1 network, no hybrid species is the direct ancestor of another hybrid species. For a general level-k network, the time complexity increases to $O(4^k\cdot|G|\cdot|S|^2)$, which, while exponential, is still smaller than existing algorithms that are exponential in the number of species and hybridization nodes.

2 Background

2.1 Preliminaries

We start by giving some basic definitions using notation largely verbatim from To and Scornavacca (2015). A summary of notation can be found in Supplemental Table S1.1.

A *rooted phylogenetic network* refers to a rooted directed acyclic graph with a single root with in-degree 0 and out-degree 2, multiple leaves with in-degree 1 out-degree 0, and internal nodes with either in-degree 1 and out-degree 2, or in-degree 1 and out-degree 2. Nodes in this last set are *hybridization nodes*, and edges leading to hybridization nodes are *hybridization edges*. Given a network N, let $V(N)$ denote its node set and $E(N)$ denote its edge set. Let $L(N) \subset V(N)$ denote its leaf set, $I(N) = V(N) \setminus L(N)$ denote its set of internal nodes, and $r(N) \in I(N)$ denote its root node. For node $v \in V(N)$, let $c(v)$ denote its set of children, $p(v)$ denote its parent (either a single node or a set of two nodes), and, if v has a single parent, $e(v)$ denote the edge $(p(v), v)$. The size of N, denoted by $|N|$, is equal to $|V(N)| + |E(N)|$. Given $v \in V(N)$, let N_v denote the subnetwork of N rooted at v, i.e. the subgraph of N consisting of all nodes and edges reachable from v.

Define $\leq_N$ ($<_N$) to be the partial order on $V(N)$, where given two nodes u and v of N, $u \leq_N v$ ($u <_N v$) if and only if there exists a path in N from v to u (and $u \neq v$). The partial order $\geq_N$ ($>_N$) is defined analogously. In such a case, u is said to be *lower or equal to* (lower than) v, and u a (strict) *descendant* of v, and v a (strict) *ancestor* of u.

Given two nodes u and v of N such that $u \leq_N v$, a path from v to u in N is a sequence of contiguous edges from v to u in N. Note that if $v = u$, the path from v to u is empty. As there can be multiple paths between pairs of vertices in a network, let $paths_N(v, u)$ denote the set of all paths from v to u. Let $paths(N)$ denote the set of all paths in network N.

Let $\hat{N}$ be the underlying undirected graph corresponding to N. An undirected graph is said to be biconnected if it remains connected when any single node is removed. A subgraph of a graph $\hat{N}$ is said to be a *biconnected component* if it is a maximal biconnected subgraph of $\hat{N}$. If every biconnected component of $\hat{N}$ has at most k hybridization nodes, we say that N is of level-k (Choy et al. 2005b). A *rooted phylogenetic tree* is a rooted phylogenetic network with no hybridization nodes, i.e. a level-0 network. In the remainder of this paper, we refer to rooted phylogenetic networks and rooted phylogenetic trees simply as *networks* and *trees*, respectively. We allow trees to contain *artificial nodes*, i.e. nodes with in-degree and out-degree 1, and *handle nodes*, i.e. nodes with in-degree 0 and out-degree 1.

Let a *species network* S depict the evolutionary history of a set of species, and let a *gene tree* G depict the evolutionary history of a set of genes sampled from these species. To compare a gene tree with a species network, let a *leaf map* $Le\colon L(G) \to L(S)$ label each leaf of the gene tree with the leaf of the species network from which the gene was sampled. The mapping need not be one-to-one nor onto.

2.2 Reconciliations

Definition 2.1 (Reconciliation)**.** Given a gene tree G, a species network S, and a leaf map Le, a *reconciliation*[1] R for (G, S, Le) is a pair of mappings (R_v, R_p) where $R_v : V(G) \to V(S)$ is a *vertex mapping* and $R_p : V(G) \to paths(S)$ is a *path mapping* subject to the following constraints:

1. If $g \in L(G)$, then $R_v(g) = Le(g)$.
2. If $g \in I(G)$, then for each $g' \in c(g)$, $R_v(g') \leq_S R_v(g)$.
3. If $g \neq r(G)$, then $R_p(g) \in paths_S(R_v(p(g)), R_v(g))$. Otherwise, $R_p(g) = \emptyset$.

Constraint 1 asserts that R_v extends the leaf map Le. Constraint 2 asserts that R_v satisfies the temporal constraints implied by S. Constraint 3 asserts that the vertex mapping and path mapping are consistent.

The vertex mapping specifies which node of S a node of G is mapped to, and the path mapping specifies a path in S between a node of G and its parent. Note that if S is a tree, a reconciliation can be specified

[1] When explaining topological incongruence through only deep coalescence, a reconciliation is sometimes called a *coalescent history* (Elworth et al. 2019).

by the vertex mapping alone, and the paths between nodes in the species tree would be implied. However, when hybridization is allowed, there can exist multiple paths between nodes in the species network, thus requiring the path mapping.

It will be convenient to consider several variants of a reconciliation. In the first, given $g \in V(G)$, a reconciliation R^g denotes the reconciliation R restricted to subtree G_g. In the second, a reconciliation is restricted to a subnetwork of the species network (that consists of a subset of nodes and all edges between those nodes), and only the parts of the gene tree that evolve within the sub-network are considered. In the third, a reconciliation is extended to a forest of multiple gene trees, all of which evolve within the same species network. Henceforth, the term reconciliation encompasses these variants.

As typical in a multispecies coalescent process, evolution in the species network is viewed backward in time, from the leaves toward the root. Then, given a reconciliation R, one can directly count the number of gene lineages "exiting" each edge e of the species network. Specifically, given edge $e \in E(S)$,

$$\mathbf{L}_R(e) = |\{g \in V(G) : e \in R_p(g)\}|,$$

and the number of extra lineages is

$$\mathbf{XL}_R(e) = \max(0, \mathbf{L}_R(e) - 1).$$

Finally, the *deep coalescence cost* of a reconciliation is the sum of extra lineages across all edges of the species network:

$$\mathbf{DC}_R = \sum_{e \in E(S)} \mathbf{XL}_R(e).$$

This value is also known as the *reconciliation cost*. Given a reconciliation R, the *edgeset* of R is the set of species edges used in the path mapping:

$$\mathbf{edgeset}(R) = \bigcup_{g \in V(G)} R_p(g).$$

Clearly, for $e \in \mathbf{edgeset}(R)$, $\mathbf{XL}_R(e) = \mathbf{L}_R(e) - 1$, and thus, the following is an equivalent definition for the deep coalescence cost:

$$\mathbf{DC}_R = \sum_{e \in \mathbf{edgeset}(R)} (\mathbf{L}_R(e) - 1).$$

Finally, we define the Most Parsimonious Reconciliation Problem:

Problem 2.1 (Most Parsimonious Reconciliation (MPR))**.** Given G, S, and Le, the *MPR problem* is to find a reconciliation with minimum cost.

When S is a tree, the MPR is unique (the LCA reconciliation[2]) (Wu and Zhang 2011) and can be found in $O(|G| \cdot |S|)$ time (Zmasek and Eddy 2001). However, when S is a network, the MPR is not necessarily unique.

In this work, we consider the MPR Problem for special case of a binary gene tree and a level-1 species network.

3 Methods

In this section, we provide a polynomial-time algorithm for inferring an MPR between a binary gene tree G and a level-1 species network S with leaf map Le. Like many parsimony approaches, our algorithm relies on dynamic programming. For the sake of simplicity, we present only the algorithm for minimizing the cost of a reconciliation. By using standard annotation of the dynamic programming table, we can subsequently perform a traceback and reconstruct an actual reconciliation. A summary of notation can be found in Supplemental Table S1.2. For brevity, proofs appear in Supplemental Section S2.

In the remainder of this section, also for the sake of simplicity, we often omit Le in our exposition and theorems, with the understanding that given a gene tree G and species network S, we are given a leaf map Le as well. We do make the dependence on Le explicit in our algorithms.

[2]Specifically, R_v is the LCA reconciliation, and R_p can be inferred from R_v.

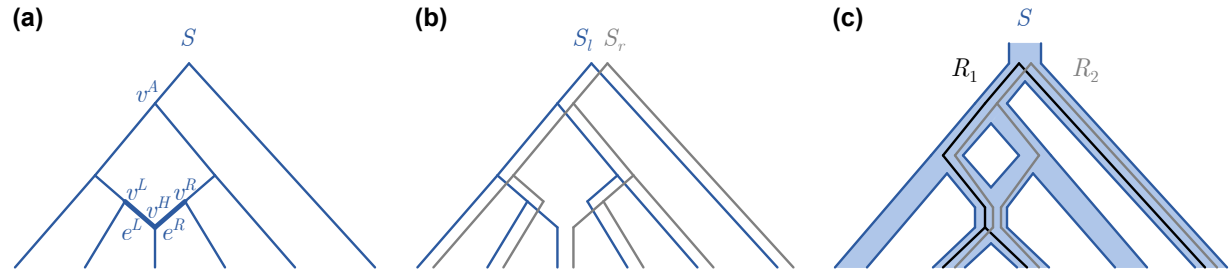

Figure 2. Species networks with one hybridization node. (a) Important nodes are labeled, including the hybridization node v^H, the left and right parents v^L and v^R of v^H, the split node v^A, and the left and right edges e^L and e^R to v^H. (b) Tree S_l constructed from S with e^L removed, and tree S_r constructed from S with e^R removed. (c) Two reconciliations R_1 and R_2 between the same gene tree and species tree. R_1 subsumes R_2.

3.1 Starting with a Simpler Problem

We start with the simpler problem of reconciling a gene tree G to a species network S with a single hybridization node v^H. For two nodes u and v of S, it can be easily shown that there exists a single node, called the *lowest common ancestor (LCA)* and denoted $lca_S(u, v)$, that is the lowest element of S that is an ancestor of both u and v. Let v^L and v^R denote the left and right parents of v^H, and let $e^L = (v^L, v^H)$ and $e^R = (v^R, v^H)$ denote the left and right edges to v^H. Let $v^A = lca_S(v^L, v^R)$, called the *split node*, denote the lowest common ancestor of v^L and v^R (Figure 2a).

Let $\mathbb{LR} = \{\mathbb{n}, \mathbb{l}, \mathbb{r}, \mathbb{b}\}$ be a set of four symbols which will be used to denote the hybridization edges in a set. Because the species network has a single hybridization node, there are four options: none, left edge, right edge, both edges. For a set of elements X, let $\mathcal{P}(X)$ denote the power set over the elements. Then, for a set of edges $E \in E(S)$, define a function $\textbf{signature}(E) : \mathcal{P}(E) \to \mathbb{LR}$ that denotes the hybridization edges in the set:

$$\textbf{signature}(E) = \begin{cases} \mathbb{n}, & e_l, e_r \notin E \\ \mathbb{l}, & e_l \in E, e_r \notin E \\ \mathbb{r}, & e_l \notin E, e_r \in E \\ \mathbb{b}, & e_l, e_r \in E. \end{cases}$$

Define the binary operator $+$ over $\mathbb{LR}$ as follows:

- For each $x \in \mathbb{LR}$, $\mathbb{n} + x = x + \mathbb{n} = x$.
- For each $x \in \mathbb{LR}$, $\mathbb{b} + x = x + \mathbb{b} = \mathbb{b}$.
- $\mathbb{l} + \mathbb{l} = \mathbb{l}$ and $\mathbb{r} + \mathbb{r} = \mathbb{r}$.
- $\mathbb{l} + \mathbb{r} = \mathbb{r} + \mathbb{l} = \mathbb{b}$.

It is easily verified that given two subsets E_1 and E_2 of $E(S)$,

$$\textbf{signature}(E_1) + \textbf{signature}(E_2) = \textbf{signature}(E_1 \cup E_2).$$

Given a reconciliation R, the signature of R, denoted $\textbf{signature}(R)$, is defined to be the signature of $\textbf{edgeset}(R)$. Conceptually, the signature of a reconciliation R denotes whether R uses neither edge, only the left edge, only the right edge, or both edges leading to the hybridization node.

Lemma 3.1 (Equivalent Edgesets)**.** *Given a gene tree G and a species network S with one hybridization node, let $R^1 = (R^1_v, R^1_p)$ and $R^2 = (R^2_v, R^2_p)$ denote two reconciliations between G and S. If $R^1_v(r(G)) = R^2_v(r(G))$ and $\textbf{signature}(R^1) = \textbf{signature}(R^2)$, then $\textbf{edgeset}(R^1) = \textbf{edgeset}(R^2)$.*

Given a gene tree G and species network S, recall that a reconciliation is said to be *optimal* if it has the minimum cost among all reconciliations between G and S. A reconciliation R between G and S is said to be *root-signature-optimal (rs-optimal)* if it has the minimum cost among all reconciliations R' between G and S such that $R'_v(r(G)) = R_v(r(G))$ and $\textbf{signature}(R') = \textbf{signature}(R)$.

Algorithm 1

```
function ReconcileSimpleSnetwork(G, S, Le)
input gene tree G, species network S with one hybridization node, leaf map Le
output mapping ECrs(g, s, x)
    for each g ∈ V(G) and each s ∈ V(S) and each x ∈ 𝕃ℝ do
        Initialize ECrs(g, s, x) = (∅, ∞).
    for each g ∈ L(G) do
        Set ECrs(g, Le(g), n) = (∅, 0).
    for each g ∈ I(G) in post-order do
        Set (g1, g2) = c(g).
        for each (x1, x2) ∈ 𝕃ℝ × 𝕃ℝ do
            for each (s1, s2) ∈ V(S) × V(S) do
                for each s ∈ V(S) such that s1 ≤_S s and s2 ≤_S s do
                    for each (p1, p2) ∈ paths_S(s, s1) × paths_S(s, s2) do
                        Set (E1, c1) = ECrs(g1, s1, x1).
                        Set (E2, c2) = ECrs(g2, s2, x2).
                        Set E = E1 ∪ E2 ∪ p1 ∪ p2.
                        Set c = c1 + c2 + |E1 ∩ E2| + |E1 ∩ p2| + |E2 ∩ p1| + |p1 ∩ p2|.
                        Set x = x1 + x2 + signature(p1) + signature(p2).
                        if c < cost(ECrs(g, s, x)) then
                            Update ECrs(g, s, x) = (E, c).
    return ECrs.
```

Lemma 3.2 (Optimal Substructure Property)**.** *Given a gene tree G and a species network S with one hybridization node, let $R^* = (R^*_v, R^*_p)$ be an rs-optimal reconciliation between G and S. Then for each $g \in V(G)$, $R^{*,g}$, the reconciliation R^* restricted to G_g, is rs-optimal.*

We are now ready to describe our dynamic programming algorithm for reconciling G and S. Our algorithm constructs a dynamic programming table **ECrs**, where given any $g \in V(G)$, $s \in V(S)$, and $x \in \mathbb{LR}$, entry $\mathbf{ECrs}(g, s, x)$ is an ordered pair (E, c) for an rs-optimal reconciliation $R = (R_v, R_p)$ between G_g and S such that $R_v(g) = s$ and $\mathbf{signature}(R) = x$. $E \in \mathcal{P}(E(S))$ denotes the edgeset of R, and non-negative integer c denotes the cost of R. Note that by Lemma 3.1 and the definition of rs-optimality, the specific vertex mapping R_v and path mapping R_p do not matter: all rs-optimal reconciliations such that $R_v(g) = s$ and $\mathbf{signature}(R) = x$ share the same edgeset and cost. By Definition 2.1, if $g \in L(G)$, then $R_v(g) = Le(g)$ and $R_p(g) = \emptyset$. In this base case, the reconciliation uses neither of the two hybridization edges and has cost 0. Otherwise, the algorithm considers $g \in I(G)$ in post-order. Let g_1 and g_2 denote the children of g. By Lemma 3.2, an rs-optimal reconciliation R between G_g and S must extend some rs-optimal reconciliation R^1 between G_{g_1} and S and some rs-optimal reconciliation R^2 between G_{g_2} and S. Let $\mathbf{cost}(\mathbf{ECrs}(\cdot,\cdot,\cdot))$ denote the cost component of an entry. Algorithm 1 describes how to complete table **ECrs**. Note that once all entries $\mathbf{ECrs}(\cdot,\cdot,\cdot)$ have been computed, the optimal cost between G and S is simply $\min_{s \in V(S), x \in \mathbb{LR}} \mathbf{cost}(\mathbf{ECrs}(r(G), s, x))$.

Theorem 3.3. *For each $g \in V(G)$, $s \in V(S)$, and $x \in \mathbb{LR}$, Algorithm 1 correctly computes $\mathbf{ECrs}(g, s, x)$.*

Theorem 3.4. *The time complexity of Algorithm 1 is $\mathrm{O}(|G| \cdot |S|^5)$.*

3.2 Reducing the Time Complexity

Next, we present an approach for speeding up the computation of table **ECrs** by a factor of $\mathrm{O}(|S|^3)$. Our approach relies on the observation that many entries of **ECrs** will never correspond to an rs-optimal reconciliation and thus need not be considered in the dynamic program. Specifically, we show that for $g \in V(G)$ and $x \in \mathbb{LR}$, the set of species $s \in V(S)$ that must be considered for entry $\mathbf{ECrs}(g, s, x)$ can be restricted to a set of constant size that corresponds to a generalization of the LCA.

Let S_l denote the tree constructed from S with e^L removed, and let S_r denote the tree constructed from S with e^R removed (Figure 2b). We extend the definition of the LCA to the *left lowest common ancestor*, denoted by $llca_S(u, v)$, and *right lowest common ancestor*, denoted by $rlca_S(u, v)$, defined as the lowest common ancestor of u and v in trees S_l and S_r. Let $BLCA_S(u, v)$ denote the set containing both the left and right lowest common ancestors.

Given a gene tree G and a species network S with one hybridization node, a reconciliation R between G and S is said to be a *BLCA mapping* if, for each internal node g of G with children g_1 and g_2, $R_v(g) \in BLCA_S(R_v(g_1), R_v(g_2))$. Note that if G has no internal nodes, then any reconciliation between G and S is trivially a BLCA mapping.

Let R and R^* be two reconciliations between G and S. R^* is said to *subsume* R if $R^*_v(r(G)) \leq_S R_v(r(G))$, $\mathbf{signature}(R^*) \leq \mathbf{signature}(R)$, $\mathbf{edgeset}(R^*) \subseteq \mathbf{edgeset}(R)$, and $\mathbf{DC}_{R^*} \leq \mathbf{DC}_R$ (Figure 2c).

Lemma 3.5. *Given a gene tree G and a species network S with one hybridization node, let $R = (R_v, R_p)$ be a reconciliation between G and S. If there exists an internal node $g \in I(G)$ with children g_1 and g_2 such that $R_v(g) \notin BLCA_S(R_v(g_1), R_v(g_2))$, then there exists some other reconciliation $R^* = (R^*_v, R^*_p)$ between G and S such that for each $u \in V(G)$ where $g \leq_G u$, $R^{*,u}$ subsumes R^u.*

Corollary 3.5.1. *Given a gene tree G and species network S with one hybridization node, then for any reconciliation $R = (R_v, R_p)$ between G and S that is not a BLCA mapping, there exists some other reconciliation R^* that is a BLCA mapping and subsumes R.*

We are now ready to describe our revised dynamic programming algorithm for reconciling a gene tree G to a species network S with one hybridization node. In addition to **ECrs**, we construct a second table **candidates** that limits the set of species that need to be considered in completing **ECrs**. As in Algorithm 1, given any $g \in V(G)$ and $x \in \mathbb{LR}$, let $R = (R_v, R_p)$ be an rs-optimal reconciliation between G_g and S such that $\mathbf{signature}(R) = x$. By Corollary 3.5.1, the algorithm need only consider R that are BLCA mappings; that is, for each internal node g with children g_1 and g_2, R must satisfy $R_v(g) \in BLCA_S(R_v(g_1), R_v(g_2))$. Let entry $\mathbf{candidates}(g, x)$ denote the set of possible values for $R_v(g)$, that is, the set of species nodes to which g can be mapped as part of some R. Then, for entry $\mathbf{ECrs}(g, s, x)$, it is clear that only entries for $s \in \mathbf{candidates}(g, x)$ need be computed. As before, entry $\mathbf{ECrs}(g, s, x)$ is an ordered pair (E, c), where E is the edgeset and c the cost of a reconciliation R as defined above. Algorithm 2 describes how to complete tables **candidates** and **ECrs**. Note that once all entries $\mathbf{candidates}(\cdot, \cdot)$ and $\mathbf{ECrs}(\cdot, \cdot)$ have been computed, the optimal cost between G and S is simply $\min_{x \in \mathbb{LR}, s \in \mathbf{candidates}(r(G), x)} \mathbf{cost}(\mathbf{ECrs}(r(G), s, x))$.

Theorem 3.6. *For each $g \in V(G)$ and $x \in \mathbb{LR}$, Algorithm 2 correctly computes $\mathbf{candidates}(g, x)$. Furthermore, for each $s \in \mathbf{candidates}(g, x)$, Algorithm 2 correctly computes $\mathbf{ECrs}(g, s, x)$.*

Theorem 3.7. *The time complexity of Algorithm 2 is $\mathrm{O}(|G| \cdot |S|^2)$.*

3.3 Allowing Gene Tree Handles

Next, we extend the previous results towards the ultimate goal of allowing for reconciliations with a level-1 species network. For reasons that will be clear later, for a gene tree G' with root node $g_1 = r(G')$, we consider the problem of reconciling a gene tree G with "handle" (gh, g_1) to a species network S with one hybridization node. gh is a *handle node* that has no parents and a single child g_1. Let $R = (R_v, R_p)$ denote an rs-optimal reconciliation between G and S such that $R_v(gh) = r(S)$ and $\mathbf{signature}(R) = x$. While R is not restricted to be a BLCA mapping, a reconciliation R' between G' and S is restricted to be a BLCA mapping. Algorithm 3 describes how to update **ECrs** accordingly via a straightforward modification of Algorithm 2.

Lemma 3.8. *For each $g \in V(G)$ and $x \in \mathbb{LR}$, Algorithm 3 correctly computes $\mathbf{candidates}(g, x)$. Furthermore, for each $s \in \mathbf{candidates}(g, x)$, Algorithm 3 correctly computes $\mathbf{ECrs}(g, s, x)$.*

Lemma 3.9. *The time complexity of Algorithm 3 is $\mathrm{O}(|G| \cdot |S|^2)$.*

Algorithm 2

function ReconcileBLCASimpleSnetwork(G, S, Le)
input gene tree G, species network S with one hybridization node, leaf map Le
output mapping $\mathbf{candidates}(g, x)$, mapping $\mathbf{ECrs}(g, s, x)$
  for each $g \in V(G)$ and each $s \in V(S)$ and each $x \in \mathbb{LR}$ **do**
    Initialize $\mathbf{ECrs}(g, s, x) = (\emptyset, \infty)$.
  for each $g \in V(G)$ and each $x \in \mathbb{LR}$ **do**
    Initialize $\mathbf{candidates}(g, x) = \emptyset$.
  for each $g \in L(G)$ **do**
    Set $\mathbf{ECrs}(g, Le(g), \mathrm{n}) = (\emptyset, 0)$.
    Set $\mathbf{candidates}(g, \mathrm{n}) = \{Le(g)\}$.
  for each $g \in I(G)$ in post-order **do**
    Set $(g_1, g_2) = c(g)$.
    for each $(x_1, x_2) \in \mathbb{LR} \times \mathbb{LR}$ **do**
      for each $(s_1, s_2) \in \mathbf{candidates}(g_1, x_1) \times \mathbf{candidates}(g_2, x_2)$ **do**
        for each $s \in BLCA_S(s_1, s_2)$ **do**
          for each $(p_1, p_2) \in paths_S(s, s_1) \times paths_S(s, s_2)$ **do**
            Set $(E_1, c_1) = \mathbf{ECrs}(g_1, s_1, x_1)$.
            Set $(E_2, c_2) = \mathbf{ECrs}(g_2, s_2, x_2)$.
            Set $E = E_1 \cup E_2 \cup p_1 \cup p_2$.
            Set $c = c_1 + c_2 + |E_1 \cap E_2| + |E_1 \cap p_2| + |E_2 \cap p_1| + |p_1 \cap p_2|$.
            Set $x = x_1 + x_2 + \mathbf{signature}(p_1) + \mathbf{signature}(p_2)$.
            if $c < \mathbf{cost}(\mathbf{ECrs}(g, s, x))$ **then**
              Update $\mathbf{candidates}(g, x) = \mathbf{candidates}(g, x) \cup \{s\}$.
              Update $\mathbf{ECrs}(g, s, x) = (E, c)$.
  return candidates, **ECrs**.

Algorithm 3

function ReconcileWithHandleSimpleSnetwork(G, S, Le)
input gene tree G with handle (gh, g_1), species network S with one hybridization node, leaf map Le
output mapping $\mathbf{ECrs}(g, s, x)$
  Set $\mathbf{candidates}, \mathbf{ECrs} = \text{ReconcileBLCASimpleSnetwork}(G_{g_1}, S, Le)$.
  for each $x \in \mathbb{LR}$ **do**
    Initialize $\mathbf{ECrs}(gh, r(S), x) = (\emptyset, \infty)$
  for each $x_1 \in \mathbb{LR}$ **do**
    for each $s_1 \in \mathbf{candidates}(g_1, x_1)$ **do**
      for each $p_1 \in paths_S(r(S), s_1)$ **do**
        Set $(E_1, c_1) = \mathbf{ECrs}(g_1, s_1, x_1)$.
        Set $E = E_1 \cup p_1$.
        Set $c = c_1$.
        Set $x = x_1 + \mathbf{signature}(p_1)$.
        if $c < \mathbf{cost}(\mathbf{ECrs}(gh, r(S), x))$ **then**
          Update $\mathbf{ECrs}(gh, r(S), x) = (E, c)$.
  return ECrs.

3.4 Considering Multiple Gene Trees

Next, we consider the problem of reconciling a forest of gene trees with handle nodes to a species network S with one hybridization node. We start by extending the definition of a reconciliation to a forest of gene trees.

Definition 3.1 (Forest Reconciliation)**.** Let $\mathcal{G} = \{G_1, \ldots, G_K\}$ denote a forest of gene trees with handle nodes $\mathcal{GH} = \{gh_1, \ldots, gh_K\}$. A *forest reconciliation* for $\mathcal{G}$ and S is a pair of mappings $(\mathcal{R}_v, \mathcal{R}_p)$ and a set of subreconciliations $\{R^1, \ldots, R^K\}$ subject to the following constraints:

1. For each k such that $1 \leq k \leq K$, R^k is a reconciliation between G_k and S.
2. For each $g \in V(\mathcal{G})$, if $g \in V(G_k)$, then $\mathcal{R}_v(g) = R^k_v(g)$ and $\mathcal{R}_p(g) = R^k_p(g)$.

Constraint 1 asserts that each R^k is associated with G_k, and Constraint 2 asserts that $\mathcal{R}$ extends each R^k.

For convenience, we often refer to a forest reconciliation as simply a reconciliation. To distinguish the two, we denote forest reconciliations using calligraphic font $\mathcal{R}$ and (tree) reconciliations using standard math font R. In the remainder of this work, we include one additional constraint on all $\mathcal{R}$: For each $gh_k \in \mathcal{GH}$, $\mathcal{R}_v(gh_k) = R^k_v(gh_k) = r(S)$. This constraint will be necessary later when we combine the forest of gene trees into a single tree. It is straightforward to extend definitions of edgeset, signature, lineages, and cost from (tree) reconciliations to forest reconciliations.

Lemma 3.10. *Given a forest $\mathcal{G} = \{G_1, \ldots, G_K\}$ of gene trees with handle nodes $\mathcal{GH} = \{gh_1, \ldots, gh_K\}$ and a species network S with one hybridization node, let $\mathcal{Q}$ and $\mathcal{R}$ denote two reconciliations between $\mathcal{G}$ and S. If* $\mathbf{signature}(\mathcal{Q}) = \mathbf{signature}(\mathcal{R})$*, then* $\mathbf{edgeset}(\mathcal{Q}) = \mathbf{edgeset}(\mathcal{R})$*.*

Given a gene tree G with handle node gh and a species network S with one hybridization node, a reconciliation $R = (R_v, R_p)$ between G and S is said to be *signature-optimal (s-optimal)* if it has the minimum cost among all reconciliations R' between G and S such that $R'_v(gh) = R_v(gh) = r(S)$ and $\mathbf{signature}(R') = \mathbf{signature}(R)$. Similarly, given a forest $\mathcal{G}$ of gene trees with handle nodes and a species network S, a reconciliation $\mathcal{R}$ is said to be *signature-optimal (s-optimal)* if it has the minimum cost among all reconciliations $\mathcal{R}'$ between $\mathcal{G}$ and S such that $\mathbf{signature}(\mathcal{R}') = \mathbf{signature}(\mathcal{R})$.

Now, define some (arbitrary) order on the trees in a forest $\mathcal{G}$. For each k such that $1 \leq k \leq K$, let $\mathcal{G}^k = \{G_1, \ldots, G_k\}$ denote the first k gene trees of $\mathcal{G}$ with handle nodes $\mathcal{GH}^k = \{gh_1, \ldots, gh_k\}$. Let $\mathcal{R}^k = \{R^1, \ldots, R^k\}$ denote a reconciliation between $\mathcal{G}^k$ and S. It follows that $\mathcal{G}^K = \mathcal{G}$, $\mathcal{GH}^K = \mathcal{GH}$, and $\mathcal{R}^K = \mathcal{R}$.

Lemma 3.11 (Optimal Substructure Property for Forest of Trees)**.** *Given a forest $\mathcal{G} = \{G_1, \ldots, G_K\}$ of gene trees and a species network S with one hybridization node, let $\mathcal{R}^* = \{R^{*,1}, \ldots, R^{*,K}\}$ denote an s-optimal reconciliation between $\mathcal{G}$ and S. Then, for each k such that $1 \leq k \leq K$, $\mathcal{R}^{*,k-1}$ and $R^{*,k}$ are s-optimal.*

We are now ready to describe our dynamic programming algorithm for reconciling $\mathcal{G}$ and S. Our algorithm constructs another table **ECs**. Given any $G_k \in \mathcal{G}$ and $x \in \mathbb{LR}$, entry $\mathbf{ECs}(G_k, x)$ is an ordered pair (E, c) for an s-optimal reconciliation $\mathcal{R}^k$ between $\mathcal{G}^k$ and S such that $\mathbf{signature}(\mathcal{R}^k) = x$. $E \in \mathcal{P}(E(S))$ denotes the edgeset of $\mathcal{R}^k$, and non-negative integer c denotes the cost of $\mathcal{R}^k$. Note that by Lemma 3.10 and the definition of s-optimality, the specific reconciliation $\mathcal{R}^k$ does not matter. As with **ECrs**, let $\mathbf{cost}(\mathbf{ECs}(\cdot, \cdot))$ denote the cost component of an entry. Algorithm 4 describes how to complete table **ECs**. It is conceptually similar to Algorithm 1 but relies on Lemma 3.11 for multiple gene trees rather than Lemma 3.2 for a single gene tree. Once all entries have been computed, the algorithm returns the cost of reconciliation $\mathcal{R}^K$.

Lemma 3.12. *Algorithm 4 correctly computes the reconciliation cost.*

Lemma 3.13. *The time complexity of Algorithm 4 is* $\mathrm{O}(\sum_{G_k \in \mathcal{G}} |G_k| \cdot |S|^2)$*.*

Algorithm 4

function ReconcileMultipleSimpleSnetwork($\mathcal{G}$, S, Le)
input forest $\mathcal{G}$ of gene trees $\{G_1, \ldots, G_K\}$ with handle nodes $\{gh_1, \ldots, gh_K\}$, species network S with one hybridization node, leaf map Le
output minimum reconciliation cost between $\{G_1, \ldots, G_K\}$ and S such that each node of $\{gh_1, \ldots, gh_K\}$ is mapped to $r(S)$
  for each k from 1 to K and each $x \in \mathbb{LR}$ **do**
    Initialize $\mathbf{ECs}(G_k, x) = (\emptyset, \infty)$.
  for each k from 1 to K **do**
    Set $\mathbf{ECrs} =$ ReconcileWithHandleSimpleSnetwork(G_k, S, Le).
    if $k = 1$ **then**
      for each $x \in \mathbb{LR}$ **do**
        Update $\mathbf{ECs}(G_1, x) = \mathbf{ECrs}(gh_1, r(S), x)$.
    else
      for each $(x_1, x_2) \in \mathbb{LR} \times \mathbb{LR}$ **do**
        Set $(E_1, c_1) = \mathbf{ECs}(G_{k-1}, x_1)$.
        Set $(E_2, c_2) = \mathbf{ECrs}(gh_k, r(S), x_2)$.
        Set $E = E_1 \cup E_2$.
        Set $c = c_1 + c_2 + |E_1 \cap E_2|$.
        Set $x = x_1 + x_2$.
        if $c < \mathbf{cost}(\mathbf{ECs}(G_k, x)$ **then**
          Update $\mathbf{ECs}(G_k, x) = (E, c)$.
  return $\min_{x \in \mathbb{LR}} \mathbf{cost}(\mathbf{ECs}(G_K, x))$.

3.5 Putting the Pieces Together

In this section, we give an efficient algorithm for solving the most parsimonious reconciliation problem for a gene tree and a level-1 species network. This algorithm has some similarities with Algorithm 1 of To and Scornavacca (2015), which finds an optimal switching[3] of a level-k species network that minimizes the duplication-loss cost between a gene tree and the resulting species tree. We demonstrate that their general approach of decomposing the gene tree and species network can be applied to our problem of minimizing the deep coalescence cost, where we reconcile each component of the decomposition using our previously presented algorithms.

We first give some definitions and lemmas, largely taken verbatim from To and Scornavacca (2015) except for minor modifications to notation.

Let B be a biconnected component of a network S. Then B contains exactly one node without ancestors in B; let $r(B)$ denote the root of B. If B is not trivial, i.e. B consists of more than one node, we can contract it by removing all nodes of B other than $r(B)$, then connect $r(B)$ to every node with in-degree 0 created by this removal. Then the following definitions are well-posed (Figure 3).

Definition 3.2 (Tree $bc(S)$; To and Scornavacca (2015), Definition 2)**.** Given a network S, the tree $bc(S)$ is obtained from S by contracting all its biconnected components.

Let $\mathring{B}$ denote the node in $bc(S)$ that corresponds to a biconnected component B in S. Given two biconnected component B_i and B_j, we say that $B_i \leq_S B_j$ (resp. $B_i <_S B_j$) if and only if $\mathring{B}_i \leq_{bc(S)} \mathring{B}_j$ (resp. $\mathring{B}_i <_{bc(S)} \mathring{B}_j$). In such a case, B_i is said to be *lower than or equal to* (resp. lower than) B_j. We say that B_i is the parent (resp. a child) of B_j if $\mathring{B}_i$ is the parent (resp. a child) of $\mathring{B}_j$ in $bc(S)$.

Definition 3.3 (Elementary network; To and Scornavacca (2015), Definition 3)**.** Given a network S, each biconnected component B that is not a leaf of S defines an elementary network, denoted by $S(B)$, consisting of B and all edges (u, v) such that $u \in V(B)$.[4]

[3] Per Definition 4 of To and Scornavacca (2015), a switching chooses, "for each hybridization edge, an incoming edge to switch on and the other to switch off."

[4] To and Scornavacca (2015) defines $S(B)$ as "consisting of B and all cut-edges coming out from B".

Definition 3.4 (Mapping $\mathcal{B}$; To and Scornavacca (2015), Definition 6)**.** For every $u \in V(G)$, $\mathcal{B}(u)$ is defined as the lowest biconnected component B of S such that $L(S_{r(B)})$ contains $\{Le(v) \mid v \in L(G_u)\}$.[5]

Definition 3.5 (Tree G_S; To and Scornavacca (2015), Definition 7)**.** The tree G_S is obtained from G as follows: For each internal node u in G with child nodes u_1 and u_2 such that there exist k biconnected components $B_{i_1} >_S \ldots >_S B_{i_k}$ strictly below $\mathcal{B}(u)$ and strictly above $\mathcal{B}(u_1)$, we add k artificial nodes $v_1 > \ldots > v_k$ on the edge (u, u_1), and we fix $\mathcal{B}(v_j)$ to B_{i_j}. We do the same for u_2.

Definition 3.6 (Subgraph $\mathcal{G}_B$; To and Scornavacca (2015), Definition 8)**.** Let B be a biconnected component of S that is not a leaf. Then $\mathcal{G}_B$ is the set of all maximal connected subgraphs H of G_S such that $\mathcal{B}(u) = B$ for every $u \in I(H)$.

Lemma 3.14 (To and Scornavacca (2015), Lemma 2)**.** *Let B be a biconnected component of S that is not a leaf. Then we have the following:*

(i) for every $H \in \mathcal{G}_B$, H is either a binary tree or an edge whose upper extremity is an artificial node. Moreover, for every leaf u of H, $\mathcal{B}(u)$ is a child of B.

(ii) if $B = \mathcal{B}(r(G))$, then $\mathcal{G}_B$ consists of one binary tree.

We make the following minor modifications to our above definitions and lemmas to require handles for each gene tree:

Definition 3.7 (Modified from Definitions 3.5 and 3.6)**.** Obtain G_S from G as in Definition 3.5. For each biconnected component B of S that is not a leaf, obtain $\mathcal{G}_B$ as in Definition 3.6. Then, consider each node u in G such that u is the root of some $H \in \mathcal{G}_B$ and u is not an artificial node. If $B = \mathcal{B}(r(G))$, add a handle node v above u. Otherwise, add an artificial node v above u. Then, $\mathcal{G}_B$ has a handle (v, u). Furthermore, fix $\mathcal{B}(v)$ to $\mathcal{B}(u)$.

Lemma 3.15 (Modified from Lemma 3.14)**.** *Let B be as defined in Definition 3.14. Then we have the following:*

(i) for every $H \in \mathcal{G}_B$, H is either a binary tree ***with a handle*** *or an edge whose upper extremity is an artificial node. Moreover, for every leaf u of H, $\mathcal{B}(u)$ is a child of B.*

(ii) if $B = \mathcal{B}(r(G))$, then $\mathcal{G}_B$ consists of one binary tree ***with a handle****.*

It can be easily shown that adding artificial nodes and handle nodes to G does not change the minimum reconciliation cost between G and S.

Given a level-1 species networks S, a biconnected component of S is either a binary tree or a network with a single hybridization node. It is straightforward to modify each of our previous algorithms that reconcile one or more gene trees to a species network with one hybridization node, to reconcile one or more gene trees to a species tree (Supplemental Algorithms S1, S2, S3). The proofs of correctness and complexity are analogous to those of the corresponding algorithms and are therefore omitted.

Let $\mathcal{M}$ denote a mapping from nodes of S to biconnected components of S such that for each node s of S, $\mathcal{M}(s)$ is the component to which s contracts. Given a gene tree G, a level-1 species network S, a mapping $\mathcal{B}$, and a mapping $\mathcal{M}$, a reconciliation R between G and S is said to be *consistent* with $\mathcal{B}$ and $\mathcal{M}$ if, for each internal node g of G with children g_1 and g_2, $\mathcal{M}(R_v(g)) = \mathcal{B}(g)$. Note that if G has no internal nodes, then any reconciliation between G and S is trivially consistent.

Lemma 3.16. *Given a gene tree G and a level-1 species network S, let $R = (R_v, R_p)$ be a reconciliation between G and S. Given a mapping $\mathcal{B}$ and a mapping $\mathcal{M}$, if there exists an internal node $g \in I(G)$ with children g_1 and g_2 such that $\mathcal{M}(R_v(g)) \neq \mathcal{B}(g)$, $\mathcal{M}(R_v(g_1)) = \mathcal{B}(g_1)$, and $\mathcal{M}(R_v(g_2)) = \mathcal{B}(g_2)$, then there exists some other reconciliation $R^* = (R^*_v, R^*_p)$ between G and S such that R^* subsumes R.*

Corollary 3.16.1. *Given a gene tree G, a level-1 species network S, a mapping $\mathcal{B}$, and a mapping $\mathcal{M}$, then for any reconciliation $R = (R_v, R_p)$ between G and S that is not consistent with $\mathcal{B}$ and $\mathcal{M}$, there exists some other reconciliation R^* that is consistent with $\mathcal{B}$ and $\mathcal{M}$ and subsumes R.*

[5] To and Scornavacca (2015) denoted this mapping as B and phrased the definition in terms of leaf labels. We use $\mathcal{B}(\cdot)$ to distinguish the mapping from a biconnected component B.

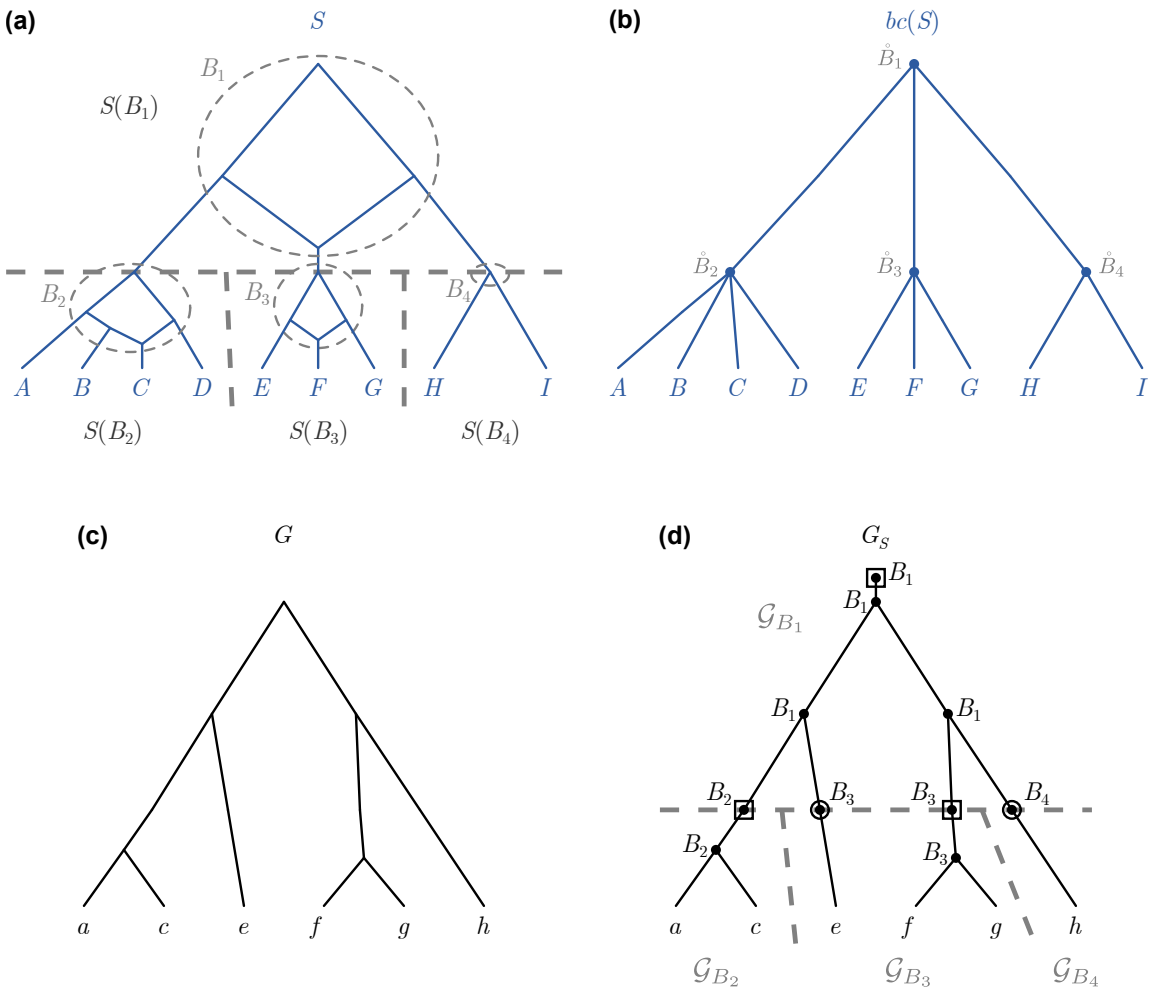

Figure 3. Annotated species networks and gene trees. (a) A level-1 species network S with four biconnected components B_i and four elementary networks $S(B_i)$, where $1 \leq i \leq 4$. (b) The tree $bc(S)$ where, for each i such that $1 \leq i \leq 4$, node $\mathring{B}_i$ in $bc(S)$ corresponds to biconnected component B_i in S. (c) A gene tree G. (d) The tree G_S along with its labeling $\mathcal{B}(\cdot)$ of nodes and subgraphs $\mathcal{G}_{B_i}$, where $1 \leq i \leq 4$. Circled nodes indicate artificial nodes (from Definition 3.5), and boxed nodes indicate handle nodes (from Definition 3.7). Figures and caption adapted from To and Scornavacca (2015).

Algorithm 5

function RECONCILE(G, S, Le)
input gene tree G, level-1 species network S, leaf map Le
output minimum reconciliation cost between G and S with Le
  Compute tree $bc(S)$ and mapping $\mathcal{M}(\cdot)$.
  Compute tree G_S and mapping $\mathcal{B}(\cdot)$.
  Compute subgraph $\mathcal{G}_{B_i}$ for each biconnected component B_i of S that is not a leaf.
  Initialize $c = 0$.
  for each biconnected component B_i of S that is not a leaf **do**
    for each leaf $g \in L(\mathcal{G}_{B_i})$ **do**
      Set $Le_{B_i}(g) = r(\mathcal{B}(g))$.
    if $|\mathcal{M}^{-1}(B_i)| = 1$ **then**
      Set $c_i = \text{RECONCILEMULTIPLESTREE}(\mathcal{G}_{B_i}, S(B_i), Le_{B_i})$.
    else
      Set $c_i = \text{RECONCILEMULTIPLESNETWORK}(\mathcal{G}_{B_i}, S(B_i), Le_{B_i})$.
    Update $c = c + c_i$.
  return c.

We are now ready to describe an algorithm for reconciling a binary gene tree G and a level-1 species network S. Algorithm 5 describes how to compute the minimum reconciliation cost between a G and S by analyzing each biconnected component of S independently (Figure 4).

Theorem 3.17. *Algorithm 5 correctly computes the minimum reconciliation cost between G and S with leaf map Le.*

Theorem 3.18. *The time complexity of Algorithm 5 is* $\mathrm{O}(|G| \cdot |S|^2)$.

3.6 Beyond Level-1 Networks

Algorithm 5 can be extended for general species networks S of level-k. To do so, Algorithms 2, 3, and 4 are easily extended to take species networks with up to k hybridization nodes. Such a modification requires tracking k separate signatures, one for each hybridization node. As there are four possible values for each signature, the time complexity of each extended algorithm would gain an additional factor of 4^k, resulting in an overall time complexity of $\mathrm{O}(4^k \cdot |G| \cdot |S|^2)$ for Algorithm 4. Since the complexity of Algorithm 4 dominates the complexity of Algorithm 5, the extended version of Algorithm 5 would then also have a time complexity of $\mathrm{O}(4^k \cdot |G| \cdot |S|^2)$. Although this time complexity is exponential in the level of the network, we might expect k to be small for most phylogenies. Thus, the algorithm could still be practical in most cases.

4 Discussion

In this work, we have presented a polynomial-time algorithm for inferring most parsimonious reconciliations between gene trees and level-1 species networks that explain topological incongruence through hybridization and ILS. Our dynamic program required several developments. First, we introduced the concept of a reconciliation signature, which specifies which hybridization edges are used by different parts of the reconciliation. Next, we showed that the number of candidate species to consider in the dynamic program can be restricted to a set of constant size that corresponds to a generalization of the LCA. Finally, we decomposed the gene tree and species network using biconnected components and reconciled each component independently. While we have focused on level-1 networks, our algorithm can be extended to level-k species networks, though the time complexity is exponential in k.

We believe that our algorithm can be applied in several contexts. When the gene tree and species network are fixed, the algorithm can be used directly to infer reconciliations. Perhaps more interesting applications include incorporating the algorithm as part of a larger pipeline. For example, given a set of reconstructed gene trees, there exist several methods for species network inference using a parsimony criterion (Yu et al.

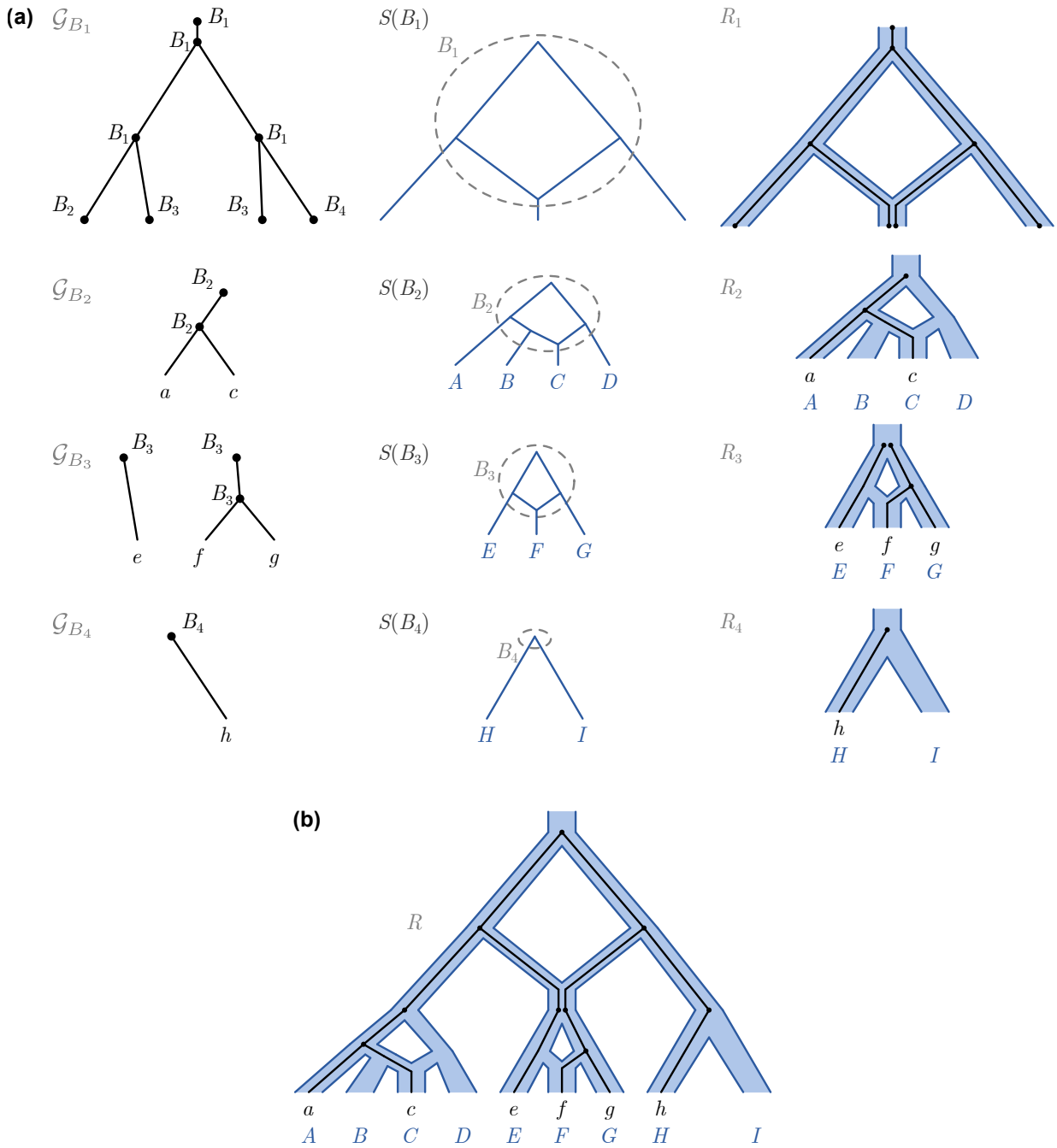

Figure 4. Reconciliation algorithm. (a) Continuing the example from Figure 3, reconciliations R_i between subgraphs $\mathcal{G}_{B_i}$ and $S(B_i)$, where $1 \leq i \leq 4$. R_1 induces 1 extra lineage, and R_2, R_3, and R_4 each induce 0 extra lineages. (b) The full reconciliation between G and S.

2011a; 2013a;b). However, since more complex networks (with more hybridization) can better fit the data (yielding reconciliations with equal or smaller numbers of extra lineages), methods are needed to balance this classic trade-off between complexity and fit. Lastly, if the species network is considered known but the gene alignments lack phylogenetic signal, reconciliation can be used to correct errors in gene tree topology (Wu et al. 2013, Bansal et al. 2015).

There are numerous directions for future work. There can exist multiple MPRs for a fixed gene tree and species network, and reconciliations are sensitive to user-defined event costs. While several papers have investigated the space of MPRs under the duplication-transfer-loss model (Bansal et al. 2013, Libeskind-Hadas et al. 2014) and the duplication-transfer-coalescence model (Du et al. 2019a, Mawhorter et al. 2019), we believe that similar problems can be explored in a joint hybridization and ILS model. Similarly, several reconciliation algorithms have been extended to handle non-binary gene trees (Yu et al. 2011b, Kordi and Bansal 2019) or species networks, or to incorporate macro-evolutionary events such as gene duplication, loss, and transfer (To and Scornavacca 2015, Du et al. 2019b). While hybridization and gene transfer may result in similar types of incongruence, more investigation is needed to see how we might disentangle the two signals. Finally, a theoretical analysis might address whether the MPR problem under a hybridization and ILS model is NP-hard for level-k species networks for arbitrary values of k.

Acknowledgements

This work was supported by the Department of Computer Science and the Dean of Faculty of Harvey Mudd College. This material is based upon work supported by the National Science Foundation under Grant No. IIS-1751399 to YW and IIS-1905885 to RLH.

References

Bansal M. S, Alm E. J and Kellis M. 2013. Reconciliation revisited: Handling multiple optima when reconciling with duplication, transfer, and loss. Journal of Computational Biology **20**:738–754.

Bansal M. S, Wu Y.-C, Alm E. J and Kellis M. 2015. Improved gene tree error correction in the presence of horizontal gene transfer. Bioinformatics **31**:1211–1218.

Chan Y.-b, Ranwez V and Scornavacca C. 2017. Inferring incomplete lineage sorting, duplications, transfers and losses with reconciliations. J Theor Biol **432**:1–13.

Choy C, Jansson J, Sadakane K and Sung W.-K. 2005a. Computing the maximum agreement of phylogenetic networks. Theor Comput Sci **335**:93–107.

Choy C, Jansson J, Sadakane K and Sung W.-K. 2005b. Computing the maximum agreement of phylogenetic networks. Theor Comput Sci **335**:93–107.

Degnan J. H and Rosenberg N. A. 2009. Gene tree discordance, phylogenetic inference and the multispecies coalescent. Trends in Ecology & Evolution **24**:332–340.

Du H, Ong Y. S, Knittel M, Mawhorter R, Liu N, Gross G, Tojo R, Libeskind-Hadas R and Wu Y.-C. 2019a. Multiple optimal reconciliations under the duplication-loss-coalescence model. IEEE/ACM Transactions on Computational Biology and Bioinformatics 1–1.

Du P, Ogilvie H. A and Nakhleh L. 2019b. Unifying gene duplication, loss, and coalescence on phylogenetic networks. In Z. Cai, P. Skums and M. Li, eds., Bioinformatics Research and Applications, 40–51. Springer International Publishing, Cham.

Elworth R. A. L, Ogilvie H. A, Zhu J and Nakhleh L. 2019. Advances in computational methods for phylogenetic networks in the presence of hybridization. In T. Warnow, ed., Bioinformatics and Phylogenetics: Seminal Contributions of Bernard Moret, 317–360. Springer International Publishing, Cham.

Fontaine M. C, Pease J. B, Steele A, Waterhouse R. M, et al. (19 co-authors). 2015. Extensive introgression in a malaria vector species complex revealed by phylogenomics. Science **347**:1258524.

Kingman J. F. C. 1982. On the genealogy of large populations. J Appl Probab **19**:27–43.

Kordi M and Bansal M. S. 2019. Exact algorithms for duplication-transfer-loss reconciliation with non-binary gene trees. IEEE/ACM Transactions on Computational Biology and Bioinformatics **16**:1077–1090.

Kubatko L. S. 2009. Identifying hybridization events in the presence of coalescence via model selection. Syst Biol **58**:478–488.

Lamichhaney S, Han F, Webster M. T, Andersson L, Grant B. R and Grant P. R. 2018. Rapid hybrid speciation in darwin's finches. Science **359**:224.

Libeskind-Hadas R, Wu Y.-C, Bansal M. S and Kellis M. 2014. Pareto-optimal phylogenetic tree reconciliation. Bioinformatics **30**:i87–i95.

Maddison W. P. 1997. Gene trees in species trees. Syst Biol **46**:523–536.

Mavárez J, Salazar C. A, Bermingham E, Salcedo C, Jiggins C. D and Linares M. 2006. Speciation by hybridization in heliconius butterflies. Nature **441**:868–871.

Mawhorter R, Liu N, Libeskind-Hadas R and Wu Y.-C. 2019. Inferring pareto-optimal reconciliations across multiple event costs under the duplication-loss-coalescence model. BMC Bioinformatics **20**:639.

Meng C and Kubatko L. S. 2009. Detecting hybrid speciation in the presence of incomplete lineage sorting using gene tree incongruence: A model. Theor Popul Biol **75**:35–45.

Racimo F, Sankararaman S, Nielsen R and Huerta-Sánchez E. 2015. Evidence for archaic adaptive introgression in humans. Nat Rev Genet **16**:359–371.

Runemark A, Vallejo-Marin M and Meier J. I. 2019. Eukaryote hybrid genomes. PLoS Genet **15**:e1008404.

Stolzer M, Lai H, Xu M, Sathaye D, Vernot B and Durand D. 2012. Inferring duplications, losses, transfers and incomplete lineage sorting with nonbinary species trees. Bioinformatics **28**:409–415.

To T.-H and Scornavacca C. 2015. Efficient algorithms for reconciling gene trees and species networks via duplication and loss events. BMC Genomics **16**:S6.

Wakeley J. 2008. Coalescent Theory: An Introduction. W. H. Freeman.

Wen D and Nakhleh L. 2017. Coestimating reticulate phylogenies and gene trees from multilocus sequence data. Syst Biol **67**:439–457.

Wu T and Zhang L. 2011. Structural properties of the reconciliation space and their applications in enumerating nearly-optimal reconciliations between a gene tree and a species tree. BMC Bioinformatics **12**:S7.

Wu Y.-C, Rasmussen M. D, Bansal M. S and Kellis M. 2013. TreeFix: Statistically informed gene tree error correction using species trees. Systematic Biology **62**:110–120.

Yu Y, Barnett R. M and Nakhleh L. 2013a. Parsimonious inference of hybridization in the presence of incomplete lineage sorting. Syst Biol **62**:738–751.

Yu Y, Degnan J. H and Nakhleh L. 2012. The probability of a gene tree topology within a phylogenetic network with applications to hybridization detection. PLoS Genet **8**:e1002660.

Yu Y, Dong J, Liu K. J and Nakhleh L. 2014. Maximum likelihood inference of reticulate evolutionary histories. Proc Natl Acad Sci USA **111**:16448.

Yu Y, Ristic N and Nakhleh L. 2013b. Fast algorithms and heuristics for phylogenomics under ils and hybridization. BMC Bioinformatics **14**:S6.

Yu Y, Than C, Degnan J. H and Nakhleh L. 2011a. Coalescent histories on phylogenetic networks and detection of hybridization despite incomplete lineage sorting. Syst Biol **60**:138–149.

Yu Y, Warnow T and Nakhleh L. 2011b. Algorithms for MDC-based multi-locus phylogeny inference: Beyond rooted binary gene trees on single alleles. Journal of Computational Biology **18**:1543–1559.

Zhang C, Ogilvie H. A, Drummond A. J and Stadler T. 2017. Bayesian inference of species networks from multilocus sequence data. Mol Biol Evol **35**:504–517.

Zmasek C. M and Eddy S. R. 2001. A simple algorithm to infer gene duplication and speciation events on a gene tree. Bioinformatics **17**:821–828.