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

Matthew LeMay, Ran Libeskind-Hadas, and Yi-Chieh Wu

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.

Index Terms—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* [1], species are treated as populations of individuals, and incongruence is assumed to be caused by *incomplete lineage sorting (ILS)* (Fig. 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 [2].

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 [3, 4]. Thus, given multiple gene trees, it is possible to infer a species tree using either probabilistic or parsimony approaches (see Degnan and Rosenberg [2] 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" [1]. In general, probababilistic approaches tend to be more accurate, whereas 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 (Fig. 1c), in which separate species exchange genetic information, either through introgression or hybrid speciation [5, 6, 7]. Studies have shown that hybridization can play a role in the evolution of eukaryotic species [8, 9, 10, 11].

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 [12, 13, 14, 15, 16, 17, 18, 19] and parsimony approaches [14, 20, 16] 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), considering all mappings of alleles sampled to the leaves of the MUL-tree, and finding the mapping that yields the minimum number of extra lineages. 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 [21, 22]. 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

- *M. LeMay is with the Department of Mathematics, Harvey Mudd College, Claremont, CA, 91711.*
- *R. Libeskind-Hadas and Y.C. Wu are with the Department of Computer Science, Harvey Mudd College, Claremont, CA, 91711.*
- *Address correspondence to Y.C. Wu: yjw@cs.hmc.edu.*

Manuscript received XXX; revised XXX.

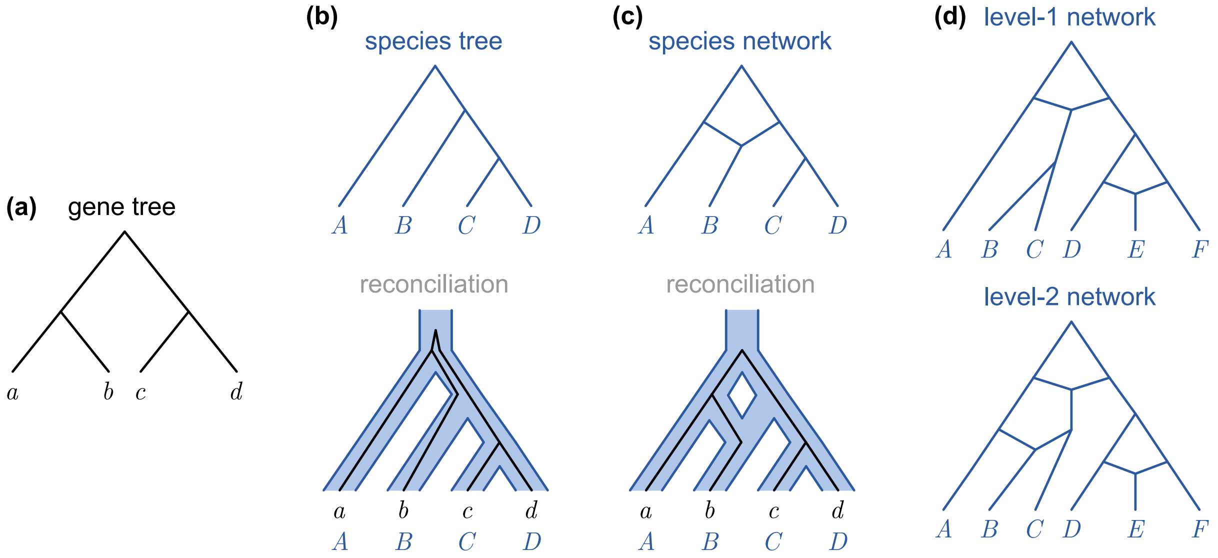

Fig. 1. **Gene trees, species trees, and species networks.** (a) A gene tree. (b) A species tree and reconciliation. Under the multispecies coalescent model, the gene tree evolves within the species tree, and incongruence between the trees is due to ILS. (c) A species network and reconciliation. The 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.

node.

In parallel with these advances in ILS and hybridization, To and Scornavacca [23] developed two 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* [24], intuitively a measure of "how much the network is 'tangled"' [23] or how densely its hybridization nodes are distributed (Fig. 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 a reconciliation between a gene tree G and a species network S that minimizes the deep coalescence cost. We address this challenge with the following contributions:

1) We present a $\mathrm{O}(|G| \cdot |S|^4)$ 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 $\mathrm{O}(|G| \cdot |S|)$ by generalizing the concept of a single lowest common ancestor (LCA) in trees to multiple LCAs in networks.
3) We present a $\mathrm{O}(|G| \cdot |S|)$ 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 $\mathrm{O}(4^k \cdot |G| \cdot |S|)$, 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 [23]. 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; additional internal nodes with either in-degree 1 and out-degree 2, called *branch nodes*, or in-degree 2 and out-degree 1, called *hybridization nodes*; and one or more leaves with in-degree 1 and out-degree 0. Edges leading to hybridization nodes are called *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[1] 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 [25]. 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 *origin nodes*, i.e. nodes with in-degree 0 and out-degree 1. For trees with origin nodes, there exists an edge between the origin node and root node, so the root node has in-degree 1 and out-degree 2.

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 mapping* $Le\colon L(G) \to L(S)$ 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 mapping Le, a *reconciliation*[2] 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 mapping 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 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 subnetwork 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 $e \in E(S)$,

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

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[3]:

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[4]) [26] and can be found in $O(|G| \cdot |S|)$ time [27]. However, when S is a network, the MPR is not necessarily unique.

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

[1] Though we have defined a path as a sequence, we will often use set operators on these sequences when the context is clear.

[2] When explaining topological incongruence through only deep coalescence, a reconciliation is sometimes called a *coalescent history* [6].

[3] The term *most parsimonious reconciliation* is more popularly used in the context of macro-evolutionary gene events, for example, to minimize duplications (D); duplications and losses (DL); or duplications, horizontal transfers, and losses (DTL). In this work, we understand MPRs to refer to reconciliations using the parsimony criterion of minimizing deep coalescences (MDC).

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

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 mapping 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 a 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 mapping Le as well. We do make the dependence on Le explicit in our algorithms.

3.1 Starting with a Simpler Problem

We start with the simpler problem of reconciling a gene tree G and a species network S with one 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 (Fig. 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. We 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 $\mathbb{r} + \mathbb{r} = \mathbb{r}$.
- $\mathbb{l} + \mathbb{r} = \mathbb{r} + \mathbb{l} = \mathbb{b}$.

We define a partial order $<$ over $\mathbb{LR}$ as follows: $\mathbb{n} < \mathbb{l}, \mathbb{r} < \mathbb{b}$.

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 $\mathbf{signature}(E) : \mathcal{P}(E) \to \mathbb{LR}$ that denotes the hybridization edges in the set:

$$\mathbf{signature}(E) = \begin{cases} \mathbb{n}, & e^L, e^R \notin E \\ \mathbb{l}, & e^L \in E, e^R \notin E \\ \mathbb{r}, & e^L \notin E, e^R \in E \\ \mathbb{b}, & e^L, e^R \in E. \end{cases}$$

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

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

Given a reconciliation R, the *signature* of R, denoted $\mathbf{signature}(R)$, is defined to be the signature of $\mathbf{edgeset}(R)$. 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 $\mathbf{signature}(R^1) = \mathbf{signature}(R^2)$, then $\mathbf{edgeset}(R^1) = \mathbf{edgeset}(R^2)$.*

Given a gene tree G, species network S, and reconciliation R between G and S, the *root* of R is defined to be $R_v(r(G))$. Recall that a reconciliation R between G and S is said to be *optimal* if it has the minimum cost among all reconciliations R' 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 $\mathbf{signature}(R') = \mathbf{signature}(R)$.

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 $g \in V(G)$, $R^{*,g}$ is rs-optimal.*

We are now ready to describe our dynamic programming algorithm for reconciling G and S (Algorithm 1). Our algorithm constructs a dynamic programming table $\mathbf{ECrs}$, 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, all rs-optimal reconciliations between G_g and S that have the same root s and signature x share the same edgeset E but not necessarily the same cost. Let $\mathbf{cost}(\mathbf{ECrs}(\cdot, \cdot, \cdot))$ denote the cost component of an entry.

In the base case, if $g \in L(G)$, then, by Definition 2.1, $R_v(g) = Le(g)$ and $R_p(g) = \emptyset$. The reconciliation uses neither of the two hybridization edges and has cost 0. That is, for $g \in L(G)$, our table is initialized with entries $\mathbf{ECrs}(g, Le(g), \mathbb{n}) = (\emptyset, 0)$.

Otherwise, the algorithm considers $g \in I(G)$ in post-order and posits a (not necessarily rs-optimal) reconciliation R between G_g and S. Let g_1 and g_2 denote the children of g. By Lemma 3.2, if R is rs-optimal, it 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 s_1, x_1, E_1, and c_1 denote the root, signature, edgeset, and cost of R^1, respectively, and similarly, let s_2, x_2, E_2, and c_2 denote the respective components of R^2. Note that R must have a root s that is an ancestor of s_1 and s_2, and gene tree edges (g, g_1) and (g, g_2) must be mapped to some path p_1 from s to s_1 and some path p_2 from s to s_2, respectively, in S. The signature x, edgeset E, and cost c of R is computed using the components of R^1 and R^2 and paths p_1 and p_2. To update $\mathbf{ECrs}(g, s, x)$, what remains is to retain only the edgeset E and cost c for some reconciliation that is rs-optimal with respect to a specific root s and signature x.

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 $\mathbf{ECrs}(g, s, x)$.*

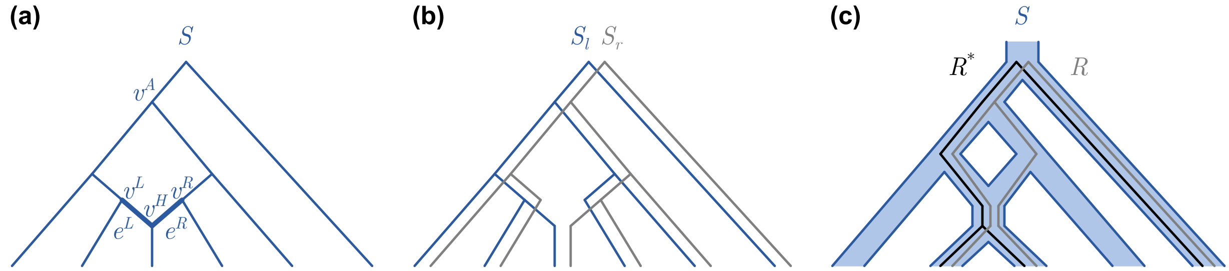

Fig. 2. **Species networks with one hybridization node.** (a) Key nodes are labeled, including the hybridization node v^H, the left and right parents v^L and v^R of v^H, the split node v^A, and the left and right edges e^L and e^R to v^H. (b) Tree S_l constructed from S with e^R removed, and tree S_r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

Algorithm 1

```
function ReconcileSimpleSnetwork($G$, $S$, $Le$)
input gene tree $G$, species network $S$ with one hybridization node, leaf mapping $Le$
output 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 $g \in L(G)$ do
        Set $\mathbf{ECrs}(g, Le(g), \mathbb{n}) = (\emptyset, 0)$.
    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 $(s_1, s_2) \in V(S) \times V(S)$ do
                for each $s \in V(S)$ such that $s_1 \leq_S s$ and $s_2 \leq_S s$ 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 $\mathbf{ECrs}(g, s, x) = (E, c)$.
    return $\mathbf{ECrs}$.
```

Theorem 3.4. *The time complexity of Algorithm 1 is* $\mathrm{O}(|G| \cdot |S|^4)$.

3.2 Reducing the Time Complexity

Next, we present an approach for speeding up the computation of table $\mathbf{ECrs}$ by a factor of $\mathrm{O}(|S|^3)$. Our approach relies on the observation that many entries of $\mathbf{ECrs}$ 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^R removed, and let S_r denote the tree constructed from S with e^L removed (Fig. 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 $\mathbf{DC}_{R^*} \leq \mathbf{DC}_R$ (Fig. 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 and

Algorithm 2

```
function ReconcileBLCASimpleSnetwork($G$, $S$, $Le$)
input gene tree $G$, species network $S$ with one hybridization node, leaf mapping $Le$
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 $\mathbf{ECrs}(g, Le(g), \mathbb{n}) = (\emptyset, 0)$.
        Set $\mathbf{candidates}(g, \mathbb{n}) = \{Le(g)\}$.
    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.
```

a species network S with one hybridization node (Algorithm 2). 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 $g \in V(G)$, $s \in V(S)$, and $x \in \mathbb{LR}$, let $R = (R_v, R_p)$ be an rs-optimal reconciliation between G_g and S such that $R_v(g) = s$ and $\mathbf{signature}(R) = x$. 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 $\mathbf{ECrs}(g, s, x)$, only entries for $s \in \mathbf{candidates}(g, x)$ need be computed. As before, the tables **candidates** and **ECrs** can be completed via post-order traversal of the gene tree. 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* $\mathbf{ECrs}(g, s, x)$.

Lemma 3.7. *In Algorithm 2, each set* $\mathbf{candidates}(g, x)$ *contains at most two elements.*

Theorem 3.8. *The time complexity of Algorithm 2 is* $\mathrm{O}(|G| \cdot |S|)$.

3.3 Extending to Multiple Gene Trees

Next, we extend the previous results towards the ultimate goal of allowing for reconciliations with a level-1 species network.

Given a gene tree G and a species network S with one hybridization node, the root of the gene tree may not be mapped to the species in which the gene family originated, for example, due to gene losses or missing samples. To address this issue, we add an *origin node* $o(G)$ and a root branch $(o(G), r(G))$ to G.

We now consider the problem of reconciling G with an origin node and S. Let $R = (R_v, R_p)$ denote an rs-optimal reconciliation between G and S such that $R_v(o(G)) = r(S)$ and $\mathbf{signature}(R) = x$. Note that R is not restricted to be a BLCA mapping. However, it is straightforward to show that, to minimize the deep coalescence cost between G and S, a reconciliation between $G_{r(G)}$ and S is restricted to be a BLCA mapping. Algorithm 3 describes how to update **ECrs** accordingly via a simple modification of Algorithm 2.

Lemma 3.9. *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 3.10. *The time complexity of Algorithm 3 is* $\mathrm{O}(|G| \cdot |S|)$.

Next, we consider the problem of reconciling a forest $\mathcal{G}$ of gene trees with origin nodes and 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 origin nodes. 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:

Algorithm 3

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

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 $\mathcal{R}_v(g) = R_v^k(g)$ and $\mathcal{R}_p(g) = R_p^k(g)$.

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 $G_k \in \mathcal{G}$, $\mathcal{R}_v(o(G_k)) = R_v^k(o(G_k)) = r(S)$. 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 lineages, edgeset, signature, and cost from (tree) reconciliations to forest reconciliations.

$$\begin{aligned}
\text{for } e \in E(S), \mathbf{L}_{\mathcal{R}}(e) &= |\{g \in V(\mathcal{G}) : e \in \mathcal{R}_p(g)\}| \\
\mathbf{edgeset}(\mathcal{R}) &= \bigcup_{g \in V(\mathcal{G})} \mathcal{R}_p(g) \\
\mathbf{signature}(\mathcal{R}) &= \mathbf{signature}(\mathbf{edgeset}(\mathcal{R})) \\
\mathbf{DC}_{\mathcal{R}} &= \sum_{e \in \mathbf{edgeset}(\mathcal{R})} (\mathbf{L}_{\mathcal{R}}(e) - 1)
\end{aligned}$$

Lemma 3.11 (Equivalent Edgesets for Forest Reconciliations)**.** *Given a forest $\mathcal{G} = \{G_1, \ldots, G_K\}$ of gene trees with origin nodes 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 an origin node 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 $R'_v(o(G)) = R_v(o(G)) = r(S)$ and $\mathbf{signature}(R') = \mathbf{signature}(R)$. Similarly, given a forest $\mathcal{G}$ of gene trees with origin 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 $\mathcal{G}$. Let $\mathcal{R}^k = \{R^1, \ldots, R^k\}$ denote a reconciliation between $\mathcal{G}^k$ and S. It follows that $\mathcal{G}^K = \mathcal{G}$ and $\mathcal{R}^K = \mathcal{R}$.

Lemma 3.12 (Optimal Substructure Property for Forest Reconciliations)**.** *Given a forest $\mathcal{G} = \{G_1, \ldots, G_K\}$ of gene trees with origin nodes 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 $1 \leq k \leq K$, $R^{*,k}$ and $\mathcal{R}^{*,k}$ are s-optimal.*

We are now ready to describe our dynamic programming algorithm for reconciling $\mathcal{G}$ and S (Algorithm 4). Our algorithm constructs another table $\mathbf{ECs}$. 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 3.11, all s-optimal reconciliations between $\mathcal{G}^k$ and S that have the same signature share the same edgeset but not necessarily the same cost. As with $\mathbf{ECrs}$, let $\mathbf{cost}(\mathbf{ECs}(\cdot, \cdot))$ denote the cost component of an entry.

The procedure for completing table $\mathbf{ECs}$ (Algorithm 4) is conceptually similar to the procedure for completing $\mathbf{ECrs}$ (Algorithm 1) but relies on Lemma 3.12 on the substructure for a forest reconciliation rather than Lemma 3.2 on the substructure for a tree reconciliation. Once all entries have been computed, the cost of reconciliation $\mathcal{R}^K$ is returned.

Lemma 3.13. *Algorithm 4 correctly computes the reconciliation cost.*

Lemma 3.14. *The time complexity of Algorithm 4 is $O(\sum_{G_k \in \mathcal{G}} |G_k| \cdot |S|)$.*

3.4 Putting the Pieces Together

In this section, we give an efficient algorithm for solving the most parsimonious reconciliation problem for a gene tree

Algorithm 4

```
function ReconcileForestSimpleSnetwork($\mathcal{G}$, $S$, $Le$)
input forest $\mathcal{G}$ of gene trees $\{G_1, \ldots, G_K\}$ with origin nodes, species network $S$ with one hybridization node, leaf mapping $Le$
output minimum reconciliation cost between $\mathcal{G}$ and $S$ such that each origin node 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 $\mathbf{ECrs} =$ ReconcileOriginSimpleSnetwork$(G_k, S, Le)$.
        if $k = 1$ then
            for each $x \in \mathbb{LR}$ do
                Update $\mathbf{ECs}(G_1, x) = \mathbf{ECrs}(o(G_1), r(S), x)$.
        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 $(E_2, c_2) = \mathbf{ECrs}(o(G_k), r(S), x_2)$.
                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 $\min_{x \in \mathbb{LR}} \mathbf{cost}(\mathbf{ECs}(G_K, x))$.
```

and a level-1 species network. This algorithm has some similarities with Algorithm 1 of To and Scornavacca [23], which finds an optimal switching[5] 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 are given as input a gene tree G, a level-1 species network S, and a leaf mapping Le, and our goal is to compute the minimum deep coalescence cost between G and S. Our algorithm relies on several definitions and lemmas, largely taken verbatim from To and Scornavacca [23] except for minor modifications to notation.

We start by identifying and contracting all biconnected components of the species network (Fig. 3a-b). As presented in To and Scornavacca [23], let B be a biconnected component of a network S.[6] Then B contains exactly one node without ancestors in B; let $r(B)$ denote the root of B. If 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.

Definition 3.2 (Tree $bc(S)$; Fig. 3b; To and Scornavacca [23], Definition 2)**.** Given a network S, the tree $bc(S)$ is obtained from S by contracting all its biconnected components.

Next, we present notation for mapping between a network S, the biconnected components of S, and the contracted tree $bc(S)$.

Definition 3.3 (Mapping $\mathcal{M}$)**.** Given a network S, let $\mathcal{M}$ be a mapping from nodes of S to biconnected components of S. For every $s \in V(S)$, $\mathcal{M}(s)$ is the component to which s contracts.

As in To and Scornavacca [23], let $\mathring{B}$ denote the node in $bc(S)$ that corresponds to a biconnected component B in S. Given two biconnected components 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 $bc(S)$.

Our last step for processing the species network S is to decompose it into disjoint networks based on its biconnected components.

Definition 3.4 (Elementary network; Fig. 3a; To and Scornavacca [23], 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 $u \in V(B)$.[7]

Note that because S is a level-1 network, each elementary network of S is either a binary tree or a network with one hybridization node. While we have presented algorithms for reconciling one or more gene trees with a species network with one hybridization node, it is straightforward to modify each of our previous algorithms to instead reconcile one or more gene trees with a species tree (Supplemental Algorithms S1, S2, S3). The proofs of correctness and complexity

[5] Per Definition 4 of To and Scornavacca [23], a switching chooses, "for each hybridization edge, an incoming edge to switch on and the other to switch off."

[6] For consistency with To and Scornavacca [23], we take some liberties with the formal definition of biconnected components. In particular, we omit biconnected components with only two vertices. Additionally, we consider any single cut vertex not part of another biconnected component to be a biconnected component, and we consider each leaf to be a biconnected component.

[7] To and Scornavacca [23] defines $S(B)$ as "consisting of B and all cut-edges coming out from B".

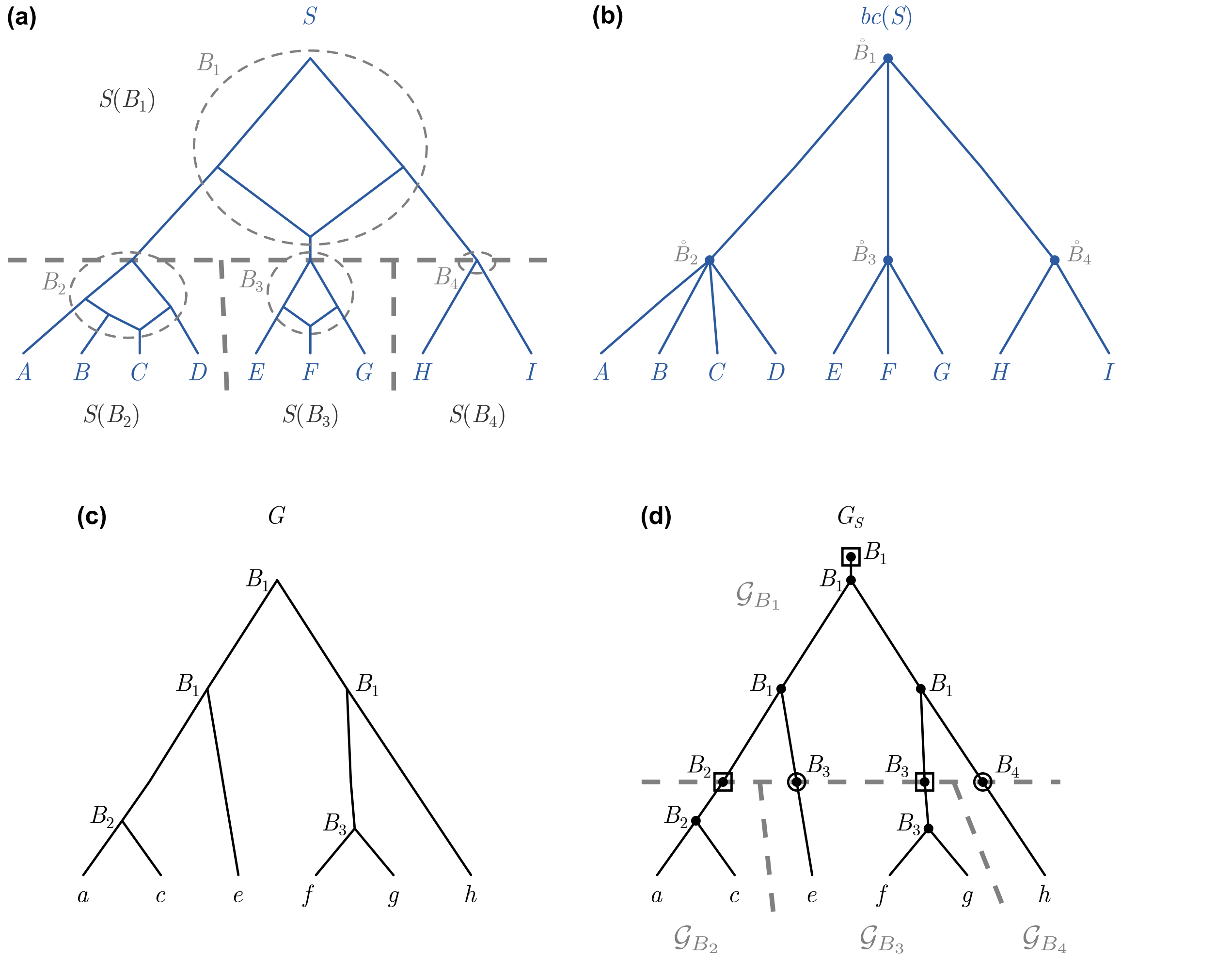

Fig. 3. **Annotated species networks and gene trees.** (a) A level-1 species network S with four biconnected components B_i and four elementary networks $S(B_i)$, where $1 \leq i \leq 4$. (b) The tree $bc(S)$ where, for each i such that $1 \leq i \leq 4$, node $\mathring{B}_i$ in $bc(S)$ corresponds to biconnected component B_i in S. (c) A gene tree G along with its mapping $\mathcal{B}(\cdot)$. (d) The tree G_S along with its mapping $\mathcal{B}(\cdot)$ and subgraphs $\mathcal{G}_{B_i}$, where $1 \leq i \leq 4$. Artificial nodes are circled (if added from the Definition 3.6, normal font) or boxed (if added from Definition 3.6, bold font). Figures and caption adapted from To and Scornavacca [23].

are analogous to those of the corresponding algorithms and are therefore omitted.

Our next step is to similarly decompose the gene tree G into disjoint forests that evolve within each elementary network (Fig. 3c-d). We make some minor modifications to the definitions and lemmas of To and Scornavacca [23] to require origins for each decomposed gene tree (modifications in bold).

Definition 3.5 (Mapping $\mathcal{B}$; Fig. 3c; To and Scornavacca [23], Definition 6)**.** Given a tree G and network S, let $\mathcal{B}$ be a mapping from nodes of G to biconnected components of S. For every $u \in V(G)$, $\mathcal{B}(u)$ is the lowest biconnected component B of S such that $L(S_{r(B)})$ contains $\{Le(v) \mid v \in L(G_u)\}$.[8]

Definition 3.6 (Tree G_S; Fig. 3d; Modified from To and Scornavacca [23], 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. **Then, for each non-root, non-artificial internal node u in G such that $\mathcal{B}(u) \neq \mathcal{B}(p(u))$, we add an artificial node v on the edge $(p(u), u)$, and we fix $\mathcal{B}(v)$ to $\mathcal{B}(u)$. Furthermore, we add an origin node v above $u = r(G)$, and we fix $\mathcal{B}(v)$ to $\mathcal{B}(u)$.**

Definition 3.7 (Subgraph $\mathcal{G}_B$; Fig. 3d; To and Scornavacca [23], 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 3.15 (Modified from To and Scornavacca [23], 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 ***with an origin node*** *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 an origin node****.*

[8] To and Scornavacca [23] 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.

Algorithm 5

```
function RECONCILE($G$, $S$, $Le$)
input gene tree $G$, level-1 species network $S$, leaf mapping $Le$
output minimum reconciliation cost between $G$ and $S$ with $Le$
    Compute tree $bc(S)$ and mapping $\mathcal{M}(\cdot)$.[9]
    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 $S(B_i)$ is a tree then
            Set $c_i =$ RECONCILEFORESTSTREE($\mathcal{G}_{B_i}, S(B_i), Le_{B_i}$).[10]
        else
            Set $c_i =$ RECONCILEFORESTSIMPLE SNETWORK($\mathcal{G}_{B_i}, S(B_i), Le_{B_i}$).
        Update $c = c + c_i$.
    return $c$.
```

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

Next, we extend the definition of subsume, previously defined for reconciliations with a species network with one hybridization node, to reconciliations with a level-1 species network. Let R and R^* be two reconciliations between a gene tree G and a level-1 species network S. R^* is said to *subsume* R if $R^*_v(r(G)) \leq_S R_v(r(G))$, $\mathbf{edgeset}(R^*) \subseteq \mathbf{edgeset}(R)$, and $\mathbf{DC}_{R^*} \leq \mathbf{DC}_R$ (Fig. 4c).

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 $\mathcal{M}$ and a mapping $\mathcal{B}$, 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.*

Given a gene tree G, a level-1 species network S, a mapping $\mathcal{M}$, and a mapping $\mathcal{B}$, a reconciliation R between G and S is said to be *consistent* with $\mathcal{M}$ and $\mathcal{B}$ 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.

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

We are now ready to describe an algorithm for reconciling a binary gene tree G and a level-1 species network S (Algorithm 5, Fig. 4). Let R denote an optimal reconciliation between G and S. By Corollary 3.16.1, we need only consider reconciliations R that are consistent with $\mathcal{M}$ and $\mathcal{B}$. That is, for each g of G, R must satisfy $\mathcal{M}(R_v(g)) = \mathcal{B}(g)$. Our algorithm relies on independently considering each biconnected component B_i of S, reconciling the corresponding gene subgraph $\mathcal{G}_{B_i}$ and species subnetwork $S(B_i)$, and adding together the reconciliation costs of the independent components.

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

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

3.5 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 $\mathrm{O}(4^k \cdot |G| \cdot |S|)$ 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 $\mathrm{O}(4^k \cdot |G| \cdot |S|)$. 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 a most parsimonious reconciliation between a gene tree and a level-1 species networks that explains 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

[9] Note that while $\mathcal{M}$ is not explicitly used in the algorithm, it is implicitly used to determine $\mathcal{B}$, G_S, and, for each biconnected component B_i, $\mathcal{G}_{B_i}$.

[10] See Supplemental Algorithm S3.

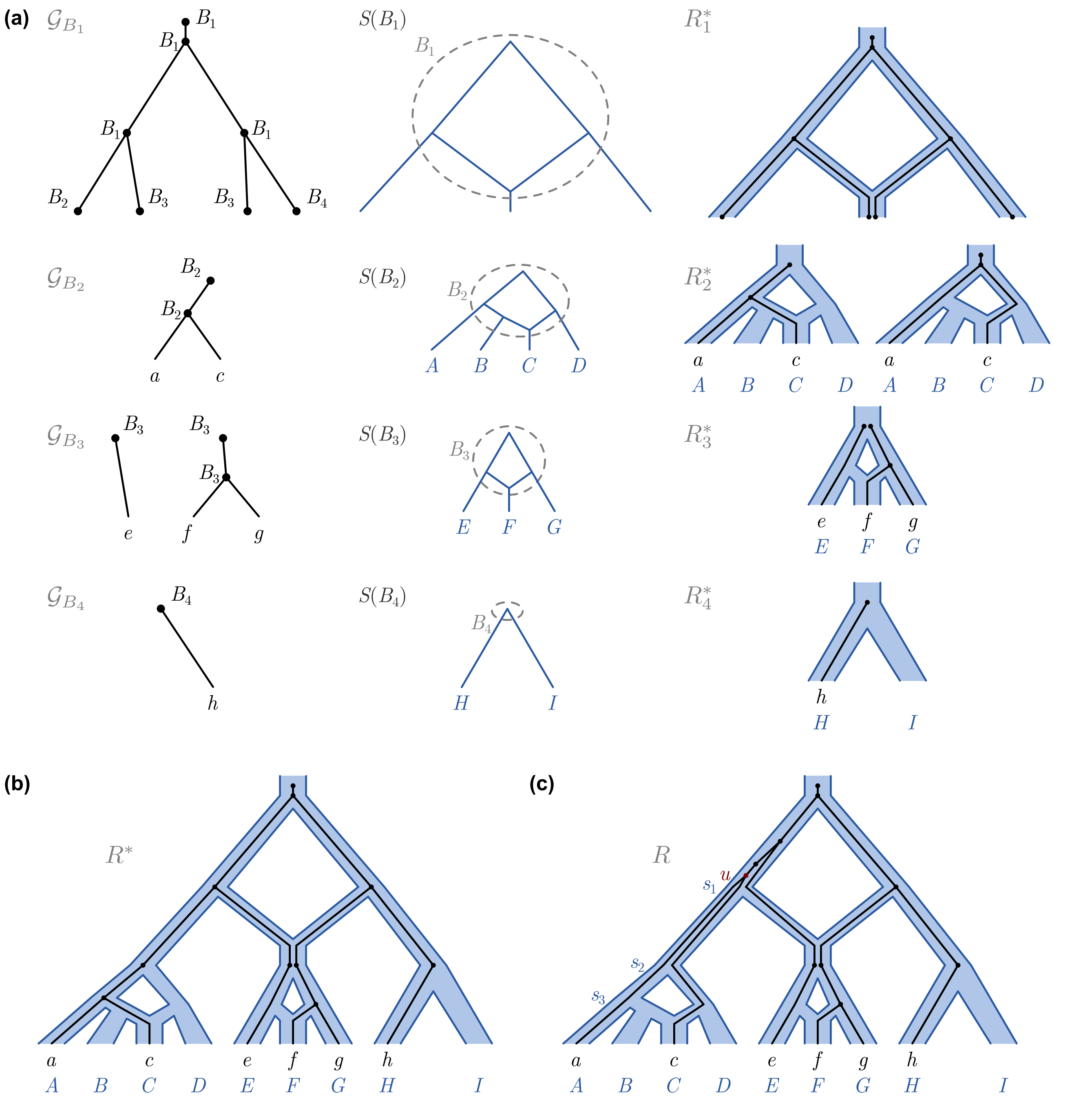

Fig. 4. **Reconciliation algorithm.** (a) Continuing the example from Fig. 3, optimal reconciliations R_i^* between subgraphs $\mathcal{G}_{B_i}$ and $S(B_i)$, where $1 \leq i \leq 4$. (There exist two optimal reconciliations R_2^*.) R_1^* induces 1 extra lineage, and R_2^*, R_3^*, and R_4^* each induce 0 extra lineages. (b) The full optimal reconciliation R^* between G and S. (c) A different reconciliation R between G and S such that R^* subsumes R. Note that for node u, $\mathcal{B}(u) = B_2$. In R^*, u is mapped to $R_v^*(u) = s_3$ so that $\mathcal{M}(R_v^*(u)) = \mathcal{M}(s_3) = B_2$. In contrast, in R, u is mapped to $R_v(u) = s_1$ so that $\mathcal{M}(R_v(u)) = \mathcal{M}(s_1) = B_1$. Furthermore, R^* is consistent with $\mathcal{M}$ and $\mathcal{B}$ whereas R is not consistent with $\mathcal{M}$ and $\mathcal{B}$. Compared to R^*, R induces an additional extra lineage to exit s_2 into s_1.

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, if the species network is considered known but the gene trees must be reconstructed from gene alignments that lack phylogenetic signal, reconciliation can be used to correct errors in gene tree topology [28, 29]. On the other hand, given a set of reconstructed gene trees, there exist several methods for species network inference using a parsimony criterion [14, 20, 16]. However, since more complex networks (with more hybridization) can better fit data (yielding reconciliations with equal or smaller numbers of extra lineages), methods are needed to balance this classic trade-off between complexity and fit. While there exist information criteria such as AIC and BIC for model selection when measuring fit through likelihood, no similar metrics exist when measuring fit through parsimony. Perhaps more troubling, species tree inference by minimizing deep coalescence is inconsistent [30], and similar consistency issues are likely to arise for species network inference using the MDC criterion. But promisingly, parsimony and probabilistic approaches can sometimes reconstruct the same species network [20].

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 [31, 32] and the duplication-loss-coalescence model [33, 34], we believe that similar problems can be explored under a joint hybridization and ILS model. Similarly, several reconciliation algorithms have been extended to handle non-binary gene trees [35, 36] or to incorporate macro-evolutionary events such as gene duplication, loss, and transfer [23, 37]. 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, theoretical analysis has shown that the MPR problem under a hybridization and ILS model is NP-hard in general, e.g. for level-k species networks for arbitrary values of k [38]. Similar analysis might address whether there exist approximation algorithms or fixed-parameter tractable algorithms.

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

[1] W. P. Maddison, “Gene trees in species trees,” *Syst Biol*, vol. 46, no. 3, pp. 523–536, Sep. 1997.

[2] J. H. Degnan and N. A. Rosenberg, “Gene tree discordance, phylogenetic inference and the multispecies coalescent,” *Trends Ecol Evol*, vol. 24, no. 6, pp. 332–340, Jun. 2009.

[3] J. F. C. Kingman, “On the genealogy of large populations,” *J Appl Probab*, vol. 19, pp. 27–43, Aug. 1982.

[4] J. Wakeley, *Coalescent Theory: An Introduction*. W. H. Freeman, 2008.

[5] R. A. Folk, P. S. Soltis, D. E. Soltis, and R. Guralnick, “New prospects in the detection and comparative analysis of hybridization in the tree of life,” *Am J Bot*, vol. 105, no. 3, pp. 364–375, May 2018.

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

[7] A. Runemark, M. Vallejo-Marin, and J. I. Meier, “Eukaryote hybrid genomes,” *PLos Genet*, vol. 15, no. 11, p. e1008404, Nov. 2019.

[8] J. Mavárez, C. A. Salazar, E. Bermingham, C. Salcedo, C. D. Jiggins, and M. Linares, “Speciation by hybridization in heliconius butterflies,” *Nature*, vol. 441, no. 7095, pp. 868–871, Jun. 2006.

[9] M. C. Fontaine, J. B. Pease, A. Steele, R. M. Waterhouse, D. E. Neafsey, I. V. Sharakhov, X. Jiang, A. B. Hall, F. Catteruccia, E. Kakani, S. N. Mitchell, Y.-C. Wu, H. A. Smith, R. R. Love, M. K. Lawniczak, M. A. Slotman, S. J. Emrich, M. W. Hahn, and N. J. Besansky, “Extensive introgression in a malaria vector species complex revealed by phylogenomics,” *Science*, vol. 347, no. 6217, p. 1258524, Jan. 2015.

[10] F. Racimo, S. Sankararaman, R. Nielsen, and E. Huerta-Sánchez, “Evidence for archaic adaptive introgression in humans,” *Nat Rev Genet*, vol. 16, no. 6, pp. 359–371, Jun. 2015.

[11] S. Lamichhaney, F. Han, M. T. Webster, L. Andersson, B. R. Grant, and P. R. Grant, “Rapid hybrid speciation in darwin’s finches,” *Science*, vol. 359, no. 6372, p. 224, Jan. 2018.

[12] C. Meng and L. S. Kubatko, “Detecting hybrid speciation in the presence of incomplete lineage sorting using gene tree incongruence: A model,” *Theor Popul Biol*, vol. 75, no. 1, pp. 35–45, Feb. 2009.

[13] L. S. Kubatko, “Identifying hybridization events in the presence of coalescence via model selection,” *Syst Biol*, vol. 58, no. 5, pp. 478–488, 2009.

[14] Y. Yu, C. Than, J. H. Degnan, and L. Nakhleh, “Coalescent histories on phylogenetic networks and detection of hybridization despite incomplete lineage sorting,” *Syst Biol*, vol. 60, no. 2, pp. 138–149, 2011.

[15] Y. Yu, J. H. Degnan, and L. Nakhleh, “The probability of a gene tree topology within a phylogenetic network with applications to hybridization detection,” *PLos Genet*, vol. 8, no. 4, p. e1002660, Apr. 2012.

[16] Y. Yu, N. Ristic, and L. Nakhleh, “Fast algorithms and heuristics for phylogenomics under ils and hybridization,” *BMC Bioinf*, vol. 14, no. 15, p. S6, Oct. 2013.

[17] Y. Yu, J. Dong, K. J. Liu, and L. Nakhleh, “Maximum

likelihood inference of reticulate evolutionary histories," *Proc Natl Acad Sci USA*, vol. 111, no. 46, p. 16448, Nov. 2014.

[18] D. Wen and L. Nakhleh, "Coestimating reticulate phylogenies and gene trees from multilocus sequence data," *Syst Biol*, vol. 67, no. 3, pp. 439–457, 2017.

[19] C. Zhang, H. A. Ogilvie, A. J. Drummond, and T. Stadler, "Bayesian inference of species networks from multilocus sequence data," *Mol Biol Evol*, vol. 35, no. 2, pp. 504–517, 2017.

[20] Y. Yu, R. M. Barnett, and L. Nakhleh, "Parsimonious inference of hybridization in the presence of incomplete lineage sorting," *Syst Biol*, vol. 62, no. 5, pp. 738–751, 2013.

[21] M. Stolzer, H. Lai, M. Xu, D. Sathaye, B. Vernot, and D. Durand, "Inferring duplications, losses, transfers and incomplete lineage sorting with nonbinary species trees," *Bioinformatics*, vol. 28, no. 18, pp. 409–415, 2012.

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

[23] T.-H. To and C. Scornavacca, "Efficient algorithms for reconciling gene trees and species networks via duplication and loss events," *BMC Genomics*, vol. 16, no. 10, p. S6, Oct. 2015.

[24] C. Choy, J. Jansson, K. Sadakane, and W.-K. Sung, "Computing the maximum agreement of phylogenetic networks," *Theor Comput Sci*, vol. 335, no. 1, pp. 93–107, May 2005.

[25] ——, "Computing the maximum agreement of phylogenetic networks," *Theor Comput Sci*, vol. 335, no. 1, pp. 93–107, May 2005.

[26] T. Wu and L. Zhang, "Structural properties of the reconciliation space and their applications in enumerating nearly-optimal reconciliations between a gene tree and a species tree," *BMC Bioinf*, vol. 12, no. 9, p. S7, Oct. 2011.

[27] C. M. Zmasek and S. R. Eddy, "A simple algorithm to infer gene duplication and speciation events on a gene tree," *Bioinformatics*, vol. 17, no. 9, pp. 821–828, Sep. 2001.

[28] Y.-C. Wu, M. D. Rasmussen, M. S. Bansal, and M. Kellis, "TreeFix: Statistically informed gene tree error correction using species trees," *Syst Biol*, vol. 62, no. 1, pp. 110–120, Jan. 2013.

[29] M. S. Bansal, Y.-C. Wu, E. J. Alm, and M. Kellis, "Improved gene tree error correction in the presence of horizontal gene transfer," *Bioinformatics*, vol. 31, no. 8, pp. 1211–1218, Apr. 2015.

[30] C. V. Than and N. A. Rosenberg, "Consistency properties of species tree inference by minimizing deep coalescences." *J Comput Biol*, vol. 18, no. 1, pp. 1–15, Jan. 2011.

[31] M. S. Bansal, E. J. Alm, and M. Kellis, "Reconciliation revisited: Handling multiple optima when reconciling with duplication, transfer, and loss," *J Comput Biol*, vol. 20, no. 10, pp. 738–754, Sep. 2013.

[32] R. Libeskind-Hadas, Y.-C. Wu, M. S. Bansal, and M. Kellis, "Pareto-optimal phylogenetic tree reconciliation," *Bioinformatics*, vol. 30, no. 12, pp. i87–i95, Jun. 15, 2014.

[33] H. Du, Y. S. Ong, M. Knittel, R. Mawhorter, N. Liu, G. Gross, R. Tojo, R. Libeskind-Hadas, and Y.-C. Wu, "Multiple optimal reconciliations under the duplication-loss-coalescence model," *IEEE/ACM Trans Comput Biol Bioinformatics*, pp. 1–1, 2019.

[34] R. Mawhorter, N. Liu, R. Libeskind-Hadas, and Y.-C. Wu, "Inferring pareto-optimal reconciliations across multiple event costs under the duplication-loss-coalescence model," *BMC Bioinf*, vol. 20, no. 20, p. 639, Dec. 2019.

[35] Y. Yu, T. Warnow, and L. Nakhleh, "Algorithms for MDC-based multi-locus phylogeny inference: Beyond rooted binary gene trees on single alleles," *J Comput Biol*, vol. 18, no. 11, pp. 1543–1559, Nov. 2011.

[36] M. Kordi and M. S. Bansal, "Exact algorithms for duplication-transfer-loss reconciliation with non-binary gene trees," *IEEE/ACM Trans Comput Biol Bioinform*, vol. 16, no. 4, pp. 1077–1090, Jul. 2019.

[37] P. Du, H. A. Ogilvie, and L. Nakhleh, "Unifying gene duplication, loss, and coalescence on phylogenetic networks," in *Bioinformatics Research and Applications*, Z. Cai, P. Skums, and M. Li, Eds. Cham: Springer International Publishing, 2019, pp. 40–51.

[38] M. LeMay, Y.-C. Wu, and R. Libeskind-Hadas, "The most parsimonious reconciliation problem in the presence of incomplete lineage sorting and hybridization is np-hard," in *Workshop on Algorithms in Bioinformatics (WABI 2021)*, Virtual due to COVID-19, Aug. 2–4, 2021.

Matthew LeMay received the BS degree in Mathematics from Harvey Mudd College in 2021.

Ran Libeskind-Hadas received the AB degree in Applied Mathematics from Harvard University in 1987, and the MS and PhD degrees in Computer Science from the University of Illinois at Urbana-Champaign in 1989 and 1993, respectively. He is the R. Michael Shanahan Professor of Computer Science with Harvey Mudd College.

Yi-Chieh Wu received the BSEE degree from Rice University in 2007, and the SM and PhD degrees in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology in 2009 and 2014, respectively. She is an Associate Professor of Computer Science with Harvey Mudd College.