Most Parsimonious Reconciliation in the Presence of Gene Duplication, Loss, and Deep Coalescence using Labeled Coalescent Trees – Supplementary Material

Yi-Chieh Wu, Matthew D. Rasmussen, Mukul S. Bansal, Manolis Kellis

Supplemental Methods

Within this supplement, all proofs are provided in Supplemental Section S3.

S1 The Labeled Coalescent Tree

Throughout this work, the term tree refers to a rooted binary tree. Let $T = (V(T), E(T))$ be a tree with a set $V(T)$ of nodes and a set $E(T)$ of directed branches (u, v). Let $L(T) \subset V(T)$ denote the set of leaves of T, $I(T) = V(T) \setminus L(T)$ denote the set of internal nodes, and $r(T) \in I(T)$ denote the root node. For node v, let $c(v)$ denote its set of children, $p(v)$ denote its parent, $e(v)$ denote the branch $(p(v), v)$, and $T(v)$ denote the (maximal) subtree of T rooted at v. Define $\leq_T$ to be the partial order on $V(T)$, where $u \leq_T v$ if u is a node on the path between $r(T)$ and v (inclusive), and define $\geq_T$ analogously, that is, $u \geq_T v$ if v is a node on the path between $r(T)$ and u (inclusive). For $u \neq v$, we say that u is an ancestor of v, or v is a descendant of u, if $u \leq_T v$. Given a non-empty set of nodes $\mathfrak{V} \subset V(T)$, let $lca_T(\mathfrak{V})$ denote the shared ancestor of $\mathfrak{V}$ that is located farthest from the root. Finally, for $u, v \in V(T)$, let $d_T(u, v)$ denote the number of edges on the unique path from u to v in T.

Assume that we are given a gene tree (topology) G, a species tree (topology) S, and a leaf mapping $Le\colon L(G) \to L(S)$.

Definition S1.1. The *LCT* for G, S, Le is a tuple $\langle \mathcal{M}, \mathbb{L}, \mathcal{L}, \mathcal{O} \rangle$, where

- $\mathcal{M}\colon V(G) \to V(S)$ defines a **species map** that maps each node of G to a node of S,
- $\mathbb{L}$ defines a **locus set**, that is, a set of loci that have evolved within the gene family,
- $\mathcal{L}\colon V(G) \to \mathbb{L}$ defines a **locus map** that maps each node of G to a locus in $\mathbb{L}$, and
- $\mathcal{O}$ defines a **partial order** on $V(G)$. For $s \in V(S)$, let $parent\_loci(s) \subset \mathbb{L}$ denote the set of loci that yield a new locus in species s. Then for $s \in V(S)$ and $l \in parent\_loci(s)$, $\mathcal{O}$ enforces a total order on the set of nodes $O(s, l) = \{\, g : g \in V(G);\ g \neq r(G);\ \mathcal{M}(g) = s;\ \mathcal{L}(p(g)) = l \,\}$. For $g, g' \in O(s, l)\colon g \neq g'$, define $g <_{\mathcal{O}} g'$ if g precedes g',

subject to the following constraints:

1. If $g \in L(G)$, then $\mathcal{M}(g) = Le(g)$.
2. If $g \in I(G)$, then for $g' \in c(g)$, $\mathcal{M}(g) \leq_S \mathcal{M}(g')$.
3. For $g, g' \in L(G)\colon g \neq g'$, if $\mathcal{M}(g) = \mathcal{M}(g')$, then $\mathcal{L}(g) \neq \mathcal{L}(g')$.
4. For $l \in \mathbb{L}, \exists g \in V(G)\colon \mathcal{L}(g) = l$.
5. For $l \in \mathbb{L}$, let $N(l) = \{\, g : g \in V(G);\ g \neq r(G);\ \mathcal{L}(g) = l;\ \mathcal{L}(p(g)) \neq l \,\}$. Then $|N(l)| \leq 1$, where equality holds everywhere except for $l = \mathcal{L}(r(G))$.

6. For $s \in V(S)$, $l \in parent\_loci(s)$, $g, g' \in O(s,l)$: $g \neq g'$, if $g <_{\mathcal{O}} g'$, then $g \not\geq_G g'$.

Constraint 1 above ensures that $\mathcal{M}$ is consistent with the known leaf mapping, and constraint 2 imposes on $\mathcal{M}$ the temporal constraints implied by S. Constraint 3 imposes on $\mathcal{L}$ that genes within the same extant species must belong to different loci, constraint 4 restricts $\mathbb{L}$ to the range of $\mathcal{L}$, and constraint 5 imposes that for every locus, the nodes that map to that locus must belong to a single component so that each locus is created only once. Finally, constraint 6 imposes on $\mathcal{O}$ the temporal constraints implied by G.

Before we develop the LCT further, note that from constraint 4, the locus set $\mathbb{L}$ is defined by the locus map $\mathcal{L}$; therefore, in the main manuscript and in the remainder of this work, we reduce the LCT notation to a tuple $\langle \mathcal{M}, \mathcal{L}, \mathcal{O} \rangle$. In addition, as mentioned in the main manuscript, the LCT also includes implied speciation nodes (Supplemental Section S1.1).

S1.1 Implied Speciation Nodes

For $g \in I(G)$, let $spec(g)$ be a boolean variable that denotes whether g is a speciation node. That is $spec(g)$ is true if $\forall g' \in c(g), \mathcal{M}(g) \neq \mathcal{M}(g')$ and false otherwise. In this section, we discuss how to add speciation nodes that are hidden in the gene tree; such implied speciation nodes are added to G and included in $\mathcal{M}$, $\mathcal{L}$, and $\mathcal{O}$.

Implied speciation nodes are necessary when branches of the gene tree span multiple branches of the species tree. To break up gene tree branches, locate all $g \in V(G) \setminus \{r(G)\}$ such that (a) $p(\mathcal{M}(g)) \neq \mathcal{M}(p(g))$ or (b) $\neg spec(p(g))$ and $\mathcal{M}(g) \neq \mathcal{M}(p(g))$. Replace edge $(p(g), g)$ with a new node v and two new edges $(p(g), v)$ and (v, g), and set $\mathcal{M}(v) = p(\mathcal{M}(g))$. Repeat until there exist no g exist that satisfy the above conditions.

In addition, by definition, if a speciation node has multiple children, then these children must map to different species. To account for this, locate all $g \in I(G)$ such that $spec(g)$ and $\mathcal{M}(g') = \mathcal{M}(g'')$, where g' and g'' denote the children of g. (Note that no such g exist if $\mathcal{M}$ is the LCA reconciliation. This is because in the LCA reconciliation, if $\mathcal{M}(g') = \mathcal{M}(g'') = s$, then $\mathcal{M}(g) = s$ so that $spec(g)$ is false; that is, the two conditions are mutually exclusive.) For $g' \in c(g)$, replace edge (g, g') with a new node v and two new edges (g, v) and (v, g'), and set $\mathcal{M}(v) = \mathcal{M}(g)$.

S1.1.1 Delayed Speciation Nodes

A delayed speciation node is a special type of implied speciation node. Unlike previous implied speciation nodes, delayed speciation nodes are not hidden as a result of gene loss or deep coalescence. Rather, these are necessary to model delays between speciation and coalescence. That is, while gene tree nodes are found within a species tree branch, $\mathcal{M}$ maps $V(G)$ to $V(S)$ rather than $E(S)$. In effect, the LCT rounds speciation nodes to the breakpoint between species (Supplemental Figure S2A). To correct for this, the LCT includes delayed speciation nodes (Supplemental Figure S2B). To add these, locate all $g \in I(G)$ such that $spec(g)$ and $|c(g)| > 1$. For $g' \in c(g)$, replace edge (g, g') with a new node v and two new edges (g, v) and (v, g'), and set $\mathcal{M}(v) = \mathcal{M}(g)$. By default, delayed speciation nodes are *not* included in the LCT when inferring a MP LCT using DLCpar. In the remainder of this section, we discuss how this affects evolutionary events and our reasoning behind this choice.

Definition S1.2. Within each species, consider two lineages that exist to the speciation event such that, looking backwards in time, the two lineages have coalesced in a locus in this species. A *post-coalescence duplication* is a duplication has occurred in the locus in the time between the speciation event and the coalescence event. Formally, for $g \in I(G), g', g'' \in c(g)$: $g' \neq g''$; $\mathcal{M}(g) = \mathcal{M}(g') = \mathcal{M}(g'')$; $spec(g') \wedge spec(g'')$; $|c(g')| = 1 \wedge |c(g'')| = 1$, then (g, g') and (g, g'') are two lineages that have existed to the speciation event and coalesced at g in locus $l = \mathcal{L}(g)$ in species $s = \mathcal{M}(g)$. A post-coalescence duplication is one that has occurred along (g, g') so that $\mathcal{L}(g') \neq l$ or along (g, g'') so that $\mathcal{L}(g'') \neq l$.

Remark. To understand the last requirement $|c(g')| = 1 \wedge |c(g'')| = 1$, note that if $|c(g')| > 1$ or $|c(g'')| > 1$, then it is the lineages in s immediately following g' or g', respectively, that exist to the speciation event.

We now consider the relationship between delayed speciation nodes and post-coalescence duplications (Supplemental Figure S2C).

Proposition S1.1. *Delayed speciation nodes are required if and only if post-coalescence duplications exist.*

Finally, the concept of a post-coalescence duplication is not intuitive; therefore, we relate it to LCA reconciliation between the locus tree and species tree.

Proposition S1.2. *Assume that losses incur a positive cost. If the reconciliation between the locus tree and species tree is the LCA mapping, then there are no post-coalescence duplications.*

In simulation, we found LCA reconciliation between the locus tree and the species tree occurred in at least 99.6% (94.2%) of gene families when duplications and losses are simulated at 1× (4×) the rate estimated from real data. Using Proposition S1.2, this is a lower bound for the number of gene families with no post-coalescence duplications.

S1.2 Inferring Evolutionary Events in the LCT

For reasons that will become clear later, we consider evolutionary events separately for each species tree branch, but first, we define some useful functions for relating how gene tree nodes map within species tree branches. (To reduce notational complexity, in the remainder of this work, unless otherwise noted, the dependency on G, S, $\mathcal{M}$, $\mathcal{L}$, and $\mathcal{O}$ are implicit in our function definitions.) For $s \in V(S)$, let $nodes(s)$ be the set of gene tree nodes mapped to s and $leaves(s)$ and $roots(s)$ be the set of gene tree nodes mapped to the bottom and top of species tree branch $e(s)$, respectively. That is, $nodes(s) = \{\, g : g \in V(G);\ \mathcal{M}(g) = s \,\}$, $leaves(s) = \{\, g : g \in nodes(s);\ (g \in L(G) \vee \forall g' \in c(g), g' \notin nodes(s)) \,\}$, and $roots(s) = leaves(p(s))$ if $s \neq r(S)$ and $roots(s) = \emptyset$ otherwise. (Note that under these definitions, $nodes(s)$ does not include $roots(s)$.) Finally, for a set of gene tree nodes $\mathfrak{G} \subset V(G)$, let $loci(\mathfrak{G}) = \{\, l : l = \mathcal{L}(g);\ g \in \mathfrak{G} \,\}$ be the set of associated loci.

Definition S1.3. For G, S, Le and LCT $\alpha = \langle \mathcal{M}, \mathcal{L}, \mathcal{O} \rangle$, the following *evolutionary events* are defined:

- **Duplication** For $s \in V(S)$, let $D(s) = \{\, g : g \in nodes(s);\ g \neq r(G);\ \mathcal{L}(g) \neq \mathcal{L}(p(g)) \,\}$ be the set of gene tree nodes within $e(s)$ that are the result of a duplication. Then $nD(s) = |D(s)|$ and $nD_\alpha = \sum_{s \in V(S)} nD(s)$ are the number of duplications in a species and in the LCT, respectively.
- **Loss** For $s \in V(S)$, let $lost\_loci(s) = \{\, l : l \in loci(roots(s) \cup nodes(s)) \setminus loci(leaves(s)) \,\}$ be the set of loci lost within $e(s)$. Then $nL(s) = |lost\_loci(s)|$ and $nL_\alpha = \sum_{s \in V(S)} nL(s)$ are the number of losses in a species and in the LCT, respectively.
- **Deep coalescence** In the following, for a set $\mathcal{X}$ of lineages, define the number of extra lineages as $extra(\mathcal{X}) = \max(0, |\mathcal{X}| - 1)$.
  - ILS due to speciations: At the top of species tree branch $e(s)$ for $s \in V(S)$, let the lineages in locus $l \in loci(roots(s))$ be given by $\mathcal{CS}(s, l) = \{\, (g, g') : (g, g') \in E(G);\ g \in roots(s);\ \mathcal{L}(g) = l \,\}$. Then $nCS(s) = \sum_{l \in loci(roots(s))} extra(\mathcal{CS}(s, l))$ and $nCS_\alpha = \sum_{s \in V(S)} nCS(s)$ are the number of extra lineages induced by speciations in a species and in the LCT, respectively.
  - ILS due to duplications: For $s \in V(S)$, let $parent\_loci(s) = \{\, l : l = \mathcal{L}(p(g));\ g \in D(s) \,\}$. Then, for $s \in V(S)$, $l \in parent\_loci(s)$, the number $nCD(s, l)$ of contemporary lineages with locus l at the time of a duplication, summed over all duplications in s and l, can be determined by Algorithm S1, and $nCD(s) = \sum_{l \in parent\_loci(s)} nCD(s, l)$ and $nCD_\alpha = \sum_{s \in V(S)} nCD(s)$ are the number of extra lineages induced by duplications in a species and in the LCT, respectively.

  Finally, $nC(s) = nCS(s) + nCD(s)$ and $nC_\alpha = nCS_\alpha + nCD_\alpha = \sum_{s \in V(S)} nCS(s) + nCD(s)$ are the number of extra lineages in a species and in the LCT, respectively.

S2 The DLCpar Algorithm

S2.1 Problem Statement

Definition S2.1. Given G, S, Le, D, L, C, the *reconciliation cost* for LCT $\alpha = \langle \mathcal{M}, \mathcal{L}, \mathcal{O} \rangle$ is $\mathcal{R}_\alpha = D \cdot nD_\alpha + L \cdot nL_\alpha + C \cdot nC_\alpha$.

Algorithm S1 Counting extra lineages due to duplications

Input: G, $s \in V(S)$, $l \in parent\_loci(s)$, $\langle \mathcal{M}, \mathcal{L}, \mathcal{O} \rangle$
Output: $nCD(s, l)$

Let $\hat{O}(s, l)$ be the *ordered* set of nodes such that the set of nodes in $\hat{O}(s, l)$ is equal to $O(s, l)$ and the order of nodes in $\hat{O}(s, l)$ satisfies $\mathcal{O}$. That is, for $g, g' \in \hat{O}(s, l)$: $g \neq g'$, g precedes g' if and only if $g <_{\mathcal{O}} g'$.
Let $start\_lineages(s, l) = \{ (g, g') : (g, g') \in E(G);\ (g \in D(s) \vee g = r(G) \vee g \in roots(s));\ g' \in nodes(s);\ \mathcal{L}(g) = l \}$ be the set of starting lineages.
Initialize $\mathcal{CD}(s, l) = start\_lineages(s, l)$ and $nCD(s, l) = 0$.
for $g' \in \hat{O}(s, l)$ **do** {traverse nodes using the partial order $\mathcal{O}$}
    Set $l' = \mathcal{L}(g')$.
    if not $g' \in L(G) \wedge l' = l$ **then** {node is not an extant gene in the same locus}
        Remove $(p(g'), g')$ from $\mathcal{CD}(s, l)$.
    if $l' = l$ **then** {node is mapped to the same locus}
        $\forall g'' \in c(g')$, add (g', g'') to $\mathcal{CD}(s, l)$.
    else {node is mapped to a new locus, that is, a duplication has occurred}
        Add $extra(\mathcal{CD}(s, l))$ to $nCD(s, l)$.

Problem S2.1. Given G, S, Le, D, L, C, the *MP LCT* is $\alpha^* = \langle \mathcal{M}^*, \mathcal{L}^*, \mathcal{O}^* \rangle = \arg\min_\alpha \mathcal{R}_\alpha$.

Problem S2.2. Given G, S, Le, D, L, C, the *restricted MP LCT* is $\alpha^* = \langle \mathcal{M}^*, \mathcal{L}^*, \mathcal{O}^* \rangle = \arg\min_\alpha \mathcal{R}_\alpha$ subject to the condition that the reconciliation between the locus tree and species tree is the LCA mapping.

For Problems S2.1 and S2.2, α^* is not necessarily unique. Furthermore, Problem S2.2 enforces an additional constraint on the MP LCT, in particular, that the reconciliation between the locus tree and the species tree is the LCA mapping. To understand this restriction, recall that the LCT is a simplified representation of the three-tree DLCoal model. Thus, inferring a LCT is equivalent to inferring three components: a locus tree (with annotated daughter lineages), a reconciliation between the gene tree and locus tree, and a reconciliation between the locus tree and species tree. Naively, in a MP LCT, we could enforce that the two reconciliations must be the LCA mapping, as we know that the LCA mapping between the gene tree and locus tree minimizes the number of extra lineages, and the LCA mapping between the locus tree and species tree minimizes the number of duplications and losses. However, these two reconciliations are *not* independent; in particular, there exist reconciliations that violate at least one of the LCT constraints. Therefore, to make our algorithm more efficient, we constrain the reconciliation between the locus tree and species tree so that rather than inferring a global MP LCT (Problem S2.1), we infer a restricted MP LCT (Problem S2.2). Determining the gap between the global and restricted MP solutions is non-trivial, so instead, we used simulations to ask how often the true LCT satisfies the additional constraint: we found that the reconciliation between the locus tree and species tree is the LCA mapping in at least 99.6% (94.2%) of gene families when duplications and losses are simulated at $1\times$ ($4\times$) the rate estimated from real data. For simplicity, in the main manuscript and in the remainder of this work, we have omitted the term restricted and understand that a MP LCT always refers to a restricted MP LCT.

S2.2 Main Algorithm

In the remainder of this section, to keep track of the current LCT $\alpha = \langle \mathcal{M}, \mathcal{L}, \mathcal{O} \rangle$, dependencies on α (or parts of α) are explicit in our function definitions and will be denoted using subscripts. The main algorithm is shown in Algorithm S2. Throughout, if multiple optima exist, one is chosen randomly.

S2.3 Inferring an Optimal Species Map

Without a species map, we cannot determine how the gene tree fits within the species tree. Thus, in this step of the algorithm, all internal nodes $g \in I(G)$ of the gene tree have more than one child; that is, the LCT does not contain implied speciation nodes. As the leaf mapping Le of extant genes to extant species is known, our goal is to map each internal gene tree node $g \in I(G)$ to a species tree node $s \in V(S)$.

Algorithm S2 DLCpar

Input: G, S, Le, D, L, C
Output: $\mathcal{M}^*$, $\mathcal{L}^*$, $\mathcal{O}^*$

1: Use LCA to find $\mathcal{M}^*$.
2: Prune the species tree to $\hat{S} = S(\mathcal{M}^*(r(G)))$.
3: Add implied speciation nodes and factor G (Figure 3A).
4: **for** $s \in preorder(\hat{S})$ **do**
5: Use EnumerateLoci to find the set $\mathfrak{L}(s)$ of locus maps for nodes in this species tree branch. That is, each item of $\mathfrak{L}$ is one choice for $\mathcal{L}_s : roots_{\mathcal{M}^*}(s) \cup nodes_{\mathcal{M}^*}(s) \to \mathbb{L}$ (Figure 3B, top).
6: **for** $\mathcal{L}_s \in \mathfrak{L}(s)$ **do**
7: Use FindOrder to find an optimal order (associated with this locus map) $\mathcal{O}^*_{\mathcal{L}_s}$.
8: Compute the reconciliation cost for this species. For $\alpha = \langle \mathcal{M}^*, \mathcal{L}_s, \mathcal{O}^*_{\mathcal{L}_s} \rangle$, $\mathcal{R}_\alpha(s) = D \cdot nD_\alpha(s) + L \cdot nL_\alpha(s) + C \cdot (nCS_\alpha(s) + nCD_\alpha(s))$ (Figure 3B, bottom).
9: Use RemapLoci to map the set $\mathfrak{L}(s)$ of *absolute* locus maps to a set $\mathfrak{RL}(s)$ of *relative* locus maps. Each item of $\mathfrak{RL}(s)$ is one choice for $\mathcal{RL}_s : leaves_{\mathcal{M}^*}(s) \to \hat{\mathbb{L}}$. This also returns $\mathcal{RA}_s : \mathfrak{RL}(s) \to \mathfrak{L}(s)$, which maps each relative locus map to an optimum underlying absolute locus map (Figure 3C), and a cost function $\mathcal{C}_s(l_t, l_b)$, which denotes the minimum cost of assigning l_t and l_b to $roots_{\mathcal{M}^*}(s)$ and $leaves_{\mathcal{M}^*}(s)$.
10: Use FindOptimum to find optimal relative locus maps $\mathcal{RL}^*_s \, \forall s \in V(\hat{S})$ (Figure 3E).
11: For $s \in V(\hat{S})$, find $\mathcal{L}^*_s$ using $\mathcal{RL}^*_s$ and $\mathcal{RA}_s$. Then lookup $\mathcal{O}^*_{\mathcal{L}_s}$.
12: Put together $\mathcal{L}^*_s$ and $\mathcal{O}^*_{\mathcal{L}_s}$ $\forall s \in V(\hat{S})$ to determine $\mathcal{L}^*$ and $\mathcal{O}^*$.

Theorem S2.1. *If $\alpha^* = \langle \mathcal{M}^*, \mathcal{L}^*, \mathcal{O}^* \rangle$ is a MP LCT, then an optimal species map $\mathcal{M}^*$ is the LCA mapping.*

Theorem S2.1 allows us to infer an optimal species map $\mathcal{M}^*$ using the LCA mapping; for completeness, we reproduce the LCA algorithm (Algorithm S3). Once this is done, implied speciation nodes are added, and the speciation nodes are used to factor the gene tree. That is, for $s \in V(S)$, we decompose G into disjoint subtrees $subtrees(s)$ that evolve within the species tree branch $e(s)$; these subtrees can be partitioned from G using $roots(s)$ and $leaves(s)$. In addition, for the remainder of the algorithm, we work with the pruned species tree $\hat{S} = S(\mathcal{M}^*(r(G)))$ as the gene tree evolves entirely within $\hat{S}$.

Algorithm S3 LCA

Input: G, S, Le
Output: $\mathcal{M}^*$

1: **for** $g \in postorder(G)$ **do**
2: **if** $g \in L(G)$ **then**
3: Set $\mathcal{M}^*(g) = Le(g)$.
4: **else**
5: Set $\mathcal{M}^*(g) = lca_S(\{\mathcal{M}^*(g'), \mathcal{M}^*(g'')\})$, where g' and g'' denote the children of g.

S2.4 Enumerating Locus Maps

Let us start by considering the space of locus maps that can yield MP LCTs:

Proposition S2.2. *There always exists a MP LCT that satisfies the following properties:*

- *The locus changes at most once along any branch of the gene tree. (This constraint is implicit in the LCT definition.)*
- *The locus changes in at most one child for nodes internal to a species tree branch.*
- *The number of loci is at most one more than the minimum number of inferred duplications under the duplication-loss model.*

For each gene tree branch $e \in E(G)$, let $changed(e)$ denote whether the locus changed along e; that is, if $e = (g, g')$, then $changed(e)$ is true if $\mathcal{L}(g) \neq \mathcal{L}(g')$ and false otherwise. Then one option for enumerating locus maps is to consider all possible labels $changed(e)$ for each gene tree branch $e \in E(G)$, where these labels must satisfy Proposition S2.2. We can then assign $r(G)$ to an arbitrary locus and recur down G to determine the locus assignments for all $g \in V(G)$. However, this brute-force approach neglects to reuse information across locus maps. In particular, knowledge of the loci at the speciation nodes would allow us to infer the loci within each species tree branch independently of one another; thus, rather than considering the entire locus map at once, we focus separately on each species tree branch.

For $s \in V(S)$, let $\mathfrak{L}(s)$ denote the set of locus maps for the nodes in species tree branch $e(s)$; that is, each item of $\mathfrak{L}(s)$ is a species-specific locus map $\mathcal{L}_s \colon roots(s) \cup nodes(s) \to \mathbb{L}$ defined only on the set of gene tree nodes within $e(s)$ (inclusive of the root nodes). We can determine $\mathfrak{L}(s)$ using $subtrees(s)$ and $changed(e)$. In particular, we start by assigning $r(G)$ to an arbitrary locus. Then for each species tree node s in pre-order traversal, we consider all labels $changed(e)$ for $e \in E(T), T \in subtrees(s)$, where these labels must satisfy Proposition S2.2. Next, if $s = r(S)$, we combine $changed$ with knowledge of the locus at $r(G)$ to find $\mathfrak{L}(s)$. If $s \neq r(S)$, we iterate over the locus maps $\mathfrak{L}(p(s))$ for the parent species. Note that each locus map $\mathcal{L}' \in \mathfrak{L}(p(s))$ defines a locus assignment for $leaves(p(s))$, which equals the locus assignment for $roots(s)$. Thus, combining $changed$ with $\mathfrak{L}(p(s))$ determines $\mathfrak{L}(s)$. While this implementation is currently equivalent to the brute-force approach, we shall see later how $\mathfrak{L}(p(s))$ can be replaced by a set of *relative* locus maps $\mathfrak{RL}(p(s))$ to reduce redundancy.

The pseudo-code for this algorithm differs slightly from our description here as we have used $\mathfrak{RL}(p(s))$ in place of $\mathfrak{L}(p(s))$ for the input (Algorithm S4). Furthermore, for clarity, we do not include the third locus map constraint (on the maximum number of loci in a MP LCT) in our algorithm; to include this constraint, we must keep track of the cumulative (minimum) number of duplications for each locus map as we traverse down the species tree.

Algorithm S4 EnumerateLoci

Input: G, $\mathcal{M}^*$, $s \in V(\hat{S})$, $\mathfrak{RL}(p(s))$ if $s \neq r(\hat{S})$
Output: $\mathfrak{L}(s)$

Initialize $\mathfrak{L}(s) = \emptyset$.
for $T \in subtrees_{\mathcal{M}^*}(s)$ **do**
  Assign all combinations $\mathcal{C}(T)$ of $changed(e)\,\forall e \in E(T)$ subject to the constraints of Proposition S2.2.
if $s = r(\hat{S})$ **then**
  Initialize $\mathcal{RL}'(r(G)) = l_0$. Set $\mathfrak{RL}_p$ to be the set (of size 1) that includes $\mathcal{RL}'$.
else
  Set $\mathfrak{RL}_p = \mathfrak{RL}(p(s))$. {each item of $\mathfrak{RL}_p$ is a choice for $\mathcal{RL}' \colon leaves_{\mathcal{M}^*}(p(s)) \to \hat{\mathbb{L}}$}
for $\mathcal{RL}' \in \mathfrak{RL}_p$ **do**
  for all combinations of $\mathcal{C}(T)\,\forall T \in subtrees_{\mathcal{M}^*}(s)$ **do** {this defines $changed(e)\,\forall T \in subtrees_{\mathcal{M}^*}(s), e \in E(T)$}
    If $s \neq r(\hat{S})$, set $\mathcal{L}_s$ defined on $roots_{\mathcal{M}^*}(s)$ using $\mathcal{RL}'$. Else set $\mathcal{L}_s$ defined on $r(G)$ using $\mathcal{RL}'$.
    Find $\mathcal{L}_s$ defined on $nodes_{\mathcal{M}^*}(s)$ using $changed(e)\,\forall T \in subtrees_{\mathcal{M}^*}(s), e \in E(T)$. If $changed(e)$ is **true**, the new locus must not currently exist in order to satisfy Constraint 5 of the LCT definition.
    if not $s \in L(\hat{S}) \wedge \exists g, g' \in leaves_{\mathcal{M}^*}(s) \colon g \neq g';\ \mathcal{L}_s(g) = \mathcal{L}_s(g')$ **then** {check Constraint 3 of the LCT definition}
      Add $\mathcal{L}_s$ to $\mathfrak{L}(s)$.

S2.5 Inferring an Optimal Order for Each Locus Map

At this point, an optimal species map $\mathcal{M}^*$ and the set $\mathfrak{L}(s)$ of species-specific locus maps for each $s \in V(S)$ is known. What remains to be determined is an optimal order $\mathcal{O}^*_{\mathcal{L}_s}$ for each locus map $\mathcal{L}_s \in \mathfrak{L}(s)$.

Proposition S2.3. *Let $\alpha^* = \langle \mathcal{M}^*, \mathcal{L}^*, \mathcal{O}^* \rangle$ be a MP LCT. If $\mathcal{M}^*$ and $\mathcal{L}^*$ are known, then an optimal associated order $\mathcal{O}^* = \arg\min_{\mathcal{O}} R_{\langle \mathcal{M}^*, \mathcal{L}^*, \mathcal{O} \rangle}$ is one such that for $s \in V(S), l \in parent\_loci(s)$, $\mathcal{O}^*$ enforces a total order on $O(s, l)$ such that duplications are as early in the species tree branch as possible.*

Thus, for each locus map $\mathcal{L}_s \in \mathfrak{L}(s)$, rather than using brute-force to enumerate all orders then choosing an optimal $\mathcal{O}^*_{\mathcal{L}_s}$, Proposition S2.3 allows us to intelligently infer $\mathcal{O}^*_{\mathcal{L}_s}$ (Algorithm S5).

Algorithm S5 FINDORDER

Input: G, $\mathcal{M}^*$, $s \in V(\hat{S})$, $\mathcal{L}_s$
Output: $\mathcal{O}^*_{\mathcal{L}_s}$

for $l \in parent\_loci_{\mathcal{M}^*,\mathcal{L}_s}(s)$ **do**
  Initialize $\hat{O}^*_{\mathcal{L}_s}(s,l)$ (to an empty ordered set).
  Find $start\_lineages_{\mathcal{M}^*,\mathcal{L}_s}(s,l)$.
  Initialize the set of nodes that require ordering $X = O_{\mathcal{M}^*,\mathcal{L}_s}(s,l)$.
  Initialize the set of current lineages $\mathcal{X} = start\_lineages_{\mathcal{M}^*,\mathcal{L}_s}(s,l)$.
  For $g \in X$, let $d(g) = d_G(r,g)$, where $e(r) \in \mathcal{X}$ and $r \leq_G g$, be the length of the path to a current lineage.
  while $\exists g \in X \colon \mathcal{L}_s(g) \neq l$ **do** {extra lineages can be incurred due to duplications}
    Select g from among the set $\{\, g : g = \arg\min_{g' \in X \colon \mathcal{L}_s(g') \neq l} d(g') \,\}$ of nodes mapped to a different locus with shortest path to a current lineage.
    if $d_G(r,g) > 1$ **then** {g cannot be immediately added}
      for g' along the path from r to g (non-inclusive) **do**
        Add g' to $\hat{O}^*_{\mathcal{L}_s}(s,l)$. Remove g' from X. Remove $(p(g'),g')$ from $\mathcal{X}$, and $\forall g'' \in c(g')$, add (g',g'') to $\mathcal{X}$.
    Add g to $\hat{O}^*_{\mathcal{L}_s}(s,l)$. Remove g from X. Remove $(p(g),g)$ from $\mathcal{X}$. (g maps to a different locus so $c(g)$ do not affect $\mathcal{X}$.)
  while $|X| > 0$ **do** {there remain nodes to be ordered}
    Select g from among the set $\{\, g : g \in X;\ d(g) = 1 \,\}$ of immediate descendants of current lineages.
    Add g to $\hat{O}^*_{\mathcal{L}_s}(s,l)$. Remove g from X. Remove $(p(g),g)$ from $\mathcal{X}$, and $\forall g' \in c(g)$, add (g,g') to $\mathcal{X}$.
Combine $\hat{O}^*_{\mathcal{L}_s}(s,l)\, \forall l \in parent\_loci_{\mathcal{M}^*,\mathcal{L}_s}(s)$ to find an optimal order $\mathcal{O}^*_{\mathcal{L}_s}$.

S2.6 Computing Relative Locus Maps

For $s \in V(S)$ and $\mathcal{L}_s \in \mathfrak{L}(s)$, the reconciliation cost of $\langle \mathcal{M}^*, \mathcal{L}_s, \mathcal{O}^*_{\mathcal{L}_s} \rangle$ can finally be computed. For the last step within the species tree branch, let $\mathcal{RL}_s \colon leaves(s) \to \hat{\mathbb{L}}$ denote a species-specific *relative* locus map defined on the leaf nodes of species s. For $s \in V(S)$, we remap the set $\mathfrak{L}(s)$ of absolute locus maps to a set $\mathfrak{RL}(s)$ of relative locus maps. Additionally, we keep a mapping $\mathcal{RA}_s \colon \mathfrak{RL}(s) \to \mathfrak{L}(s)$ of each relative locus map to its optimal underlying (absolute) locus map and a lookup table $\mathcal{C}_s(l_t, l_b)$ that denotes the minimum cost of assigning l_t and l_b to $roots_{\mathcal{M}^*}(s)$ and $leaves_{\mathcal{M}^*}(s)$, respectively (Algorithm S6).

S2.7 Inferring an Optimal Locus Map and Order

At this point, the set of relative locus maps are known for all speciation nodes, allowing us to use dynamic programming to determine an optimal relative locus map for the speciation nodes (Algorithm S7). For $s \in V(S)$, let $F(s,l,p)$ denote the cost-to-go of assigning relative loci l at the top ($p = t$) or bottom ($p = b$) of species tree branch $e(s)$, where the cost-to-go is defined as the minimum total cost along all descendant species tree branches. In the forward phase, we perform a post-order traversal of the species tree, where at each node $s \in V(S)$, we first determine the cost-to-go of assigning l_b at $leaves(s)$, then determine the cost-to-go of assigning l_t at $roots(s)$, for all possible choices of relative loci l_t and l_b. At the root of the species tree, the minimum cost solution is selected. Then, in the backward phase, we perform a pre-order traversal to traceback down the species tree, where at each node $s \in V(S)$, we assign an optimal relative locus map $\mathcal{RL}^*_s$.

Finally, as stated in the main text, for each $s \in V(\hat{S})$, we can use $\mathcal{RL}^*_s$ in conjunction with $\mathcal{RA}_s$ to look up $\mathcal{L}^*_s$, which in turn can be used to look up $\mathcal{O}^*_{\mathcal{L}^*_s}$. These are the subparts of $\mathcal{L}^*$ and $\mathcal{O}^*$, and together with our previously inferred optimal species map $\mathcal{M}^*$, constitutes the MP LCT.

Algorithm S6 REMAPLOCI

Input: G, $\mathcal{M}^*$, $s \in V(\hat{S})$, $\mathfrak{L}(s)$, $\mathcal{R}_{\langle \mathcal{M}^*, \mathcal{L}_s, \mathcal{O}^*_{\mathcal{L}_s} \rangle}(s) \, \forall \mathcal{L}_s \in \mathfrak{L}(s)$
Output: $\mathfrak{RL}(s)$, $\mathcal{RA}_s$, $\mathcal{C}_s$

Initialize $\mathfrak{RL}(s) = \emptyset$.
Initialize the locus count $n = 0$.
Arbitrarily order $leaves_{\mathcal{M}^*}(s)$, for example, each $g \in leaves_{\mathcal{M}^*}(s)$ is ordered by its index in a preorder traversal of G.
for $\mathcal{L}_s \in \mathfrak{L}(s)$ **do**
    Set g to be the first node in $leaves_{\mathcal{M}^*}(s)$.
    Set $\mathcal{RL}_s(g) = l_n$. Set $map(\mathcal{L}_s(g)) = l_n$. Increment n.
    for each successive $g \in leaves_{\mathcal{M}^*}(s)$ **do**
        if $\mathcal{L}_s(g)$ is not already defined in map **then**
            Set $map(\mathcal{L}_s(g)) = l_n$. Increment n.
        Set $\mathcal{RL}_s(g) = map(\mathcal{L}_s(g))$.
    Add $\mathcal{RL}_s$ to $\mathfrak{RL}(s)$.
    Set $cost(\mathcal{L}_s, \mathcal{RL}_s) = \mathcal{R}_{\langle \mathcal{M}^*, \mathcal{L}_s, \mathcal{O}^*_{\mathcal{L}_s} \rangle}(s)$.
for $\mathcal{RL}_s \in \mathfrak{RL}(s)$ **do**
    Find $\hat{\mathcal{L}}_s = \arg\min_{\mathcal{L}_s} cost(\mathcal{L}_s, \mathcal{RL}_s)$. Set $\mathcal{RA}_s(\mathcal{RL}_s) = \hat{\mathcal{L}}_s$.
    Find a relative locus map $\mathcal{RL}\colon roots_{\mathcal{M}^*}(s) \to \hat{\mathbb{L}}$ from $\hat{\mathcal{L}}_s(g) \, \forall g \in roots_{\mathcal{M}^*}(s)$.
    Set $l_t = \mathcal{RL}(g) \, \forall g \in roots_{\mathcal{M}^*}(s)$, $l_b = \mathcal{RL}_s(g) \, \forall g \in leaves_{\mathcal{M}^*}(s)$, $\mathcal{C}_s(l_t, l_b) = cost(\hat{\mathcal{L}}_s, \mathcal{RL}_s)$.

Algorithm S7 FINDOPTIMUM

Input: G, $\hat{S}$, $\mathfrak{RL}(s) \, \forall s \in V(\hat{S})$, $\mathcal{C}_s \, \forall s \in V(\hat{S})$
Output: $\mathcal{RL}^*_s \, \forall s \in V(\hat{S})$

for $s \in postorder(\hat{S})$ **do**
    for $\mathcal{RL}_s \in \mathfrak{RL}(s)$ **do**
        Set $l = \mathcal{RL}_s(g) \, \forall g \in leaves_{\mathcal{M}^*}(s)$.
        if $s \in L(\hat{S})$ **then**
            Set $F(s, l, b) = 0$.
        else
            Set $F(s, l, b) = F(s', l, t) + F(s'', l, t)$, where s' and s'' denote the children of s.
        Set $\mathfrak{RL}_p = \mathfrak{RL}(p(s))$ if $s \neq r(\hat{S})$. Else set $\mathfrak{RL}_p$ to be the set (of size 1) with element l_t such that l_t is empty (that is, at the root of the species tree, $roots_{\mathcal{M}^*}(s)$ is empty so no loci are assigned).
        for $\mathcal{RL}' \in \mathfrak{RL}_p$ **do**
            Set $l_t = \mathcal{RL}'(g) \, \forall g \in leaves_{\mathcal{M}^*}(p(s))$ if $s \neq r(\hat{S})$. Else set l_t to be empty.
            Find $l_b = \arg\min_{l'_b \in \mathfrak{RL}(s)} F(s, l'_b, b) + \mathcal{C}_s(l_t, l'_b)$.
            Set $F(s, l_t, t) = F(s, l_b, b) + \mathcal{C}_s(l_t, l_b)$.
            Set $TB(s, l_t) = l_b$. {traceback pointer}
for $s \in preorder(\hat{S})$ **do**
    if $s = r(\hat{S})$ **then**
        Set l_t to be empty. {locus map at top of species tree root is known}
    else
        Set l_t using $\mathcal{RL}^*_{p(s)}$. {locus map at top of species is equal to locus map at bottom of parent species}
    Set $\mathcal{RL}^*_s$ using $l_b = TB(s, l_t)$. {use traceback to assign locus map at bottom of species}

S3 Proofs

Proof of Proposition S1.1

Proof. We first prove that if post-coalescence duplications exist, then delayed speciation nodes are required. This follows directly from Definition S1.2 by observing that g' and g'' must be delayed speciation nodes.

Next we prove that if delayed speciation nodes are required, then post-coalescence duplications exist. This is equivalent to stating that if post-coalescence duplications do not exist, then delayed speciation nodes are not required. Intuitively, this should be true because an absence of post-coalescence duplications means that nothing has occurred in the locus between the speciation event and the coalescence event; thus, we do not need to model the delay between speciation and coalescence. Formally, since g' and g'' are delayed speciation nodes, each has a single child cg' and cg'', respectively. Furthermore, if there are no post-coalescence duplications, then $\mathcal{L}(g) = \mathcal{L}(g') = \mathcal{L}(g'')$. These two observations, in conjunction with $\mathcal{M}(g) = \mathcal{M}(g') = \mathcal{M}(g'')$, allows us to replace node g' and edges (g, g'), (g', cg') with a single edge (g, cg') and replace node g'' and edges (g, g''), (g'', cg'') with a single edge (g, cg''), thus removing the delayed speciation nodes g', g''. □

Remark. Note that while node g has been rounded to the point of speciation, the actual coalescence at g may still occur prior to the speciation. This is an important point as the time of coalescence may affect the number of extra lineages. The discussion here simply shows that, assuming that post-coalescence duplications are not allowed, we do not need to add delayed speciation nodes to explicitly model the delay between speciation and coalescence.

Proof of Proposition S1.2

Proof. Under the assumption that losses incur a positive cost, for a given locus tree and species tree, LCA reconciliation is the *unique* solution that minimizes the total duplication and loss cost (Górecki and Tiuryn 2006). Thus, the proposition is equivalent to the following: for a LCT, consider the associated locus tree and reconciliation between the locus tree and the species tree (both of these are implicit in a LCT); if the reconciliation has minimum duplication and loss cost, then there are no post-coalescence duplications. We prove by contrapositive: if there are post-coalescence duplications, then the reconciliation between the locus tree and species tree (implied by the LCT) does not have minimum duplication and loss cost. Note that for the LCT, we must fix the implied locus tree topology T^L; otherwise, there could exist a different locus tree topology with lower minimum duplication and loss cost, but this has no bearing on the whether the reconciliation between locus tree T^L and species tree S is the LCA mapping.

For g, g', g'', cg', cg'' as previously defined, assume that a post-coalescence duplication has occurred along (g, g') so that $\mathcal{L}(g) \neq \mathcal{L}(g')$. Let $l = \mathcal{L}(g)$ denote the original locus, $l' = \mathcal{L}(g')$ denote the new locus, and let $s' = \mathcal{M}(cg')$ and s'' be the child of s that is not s'. Because $|c(g')| = 1$ and a locus can be created only once in the LCT, l' cannot exist in s'', that is, l' is lost in s''. Now consider alternative scenarios in which we remove the duplication along (g, g'). Such an operation has no effect on the LCT beyond g, g', cg', in particular, the resulting LCT is valid. (1) If $\mathcal{L}(cg') \neq l'$, then the duplication along (g, g') was followed by a duplication along (g', cg'), and removing the duplication along (g, g') would not affect the locus tree topology but would reduce the number of duplications by one and remove the loss of l' in s'', thus reducing the number of losses by one. (2) If $\mathcal{L}(cg') = l'$ then "pushing" the duplication from (g, g') to (g', cg') would not affect the locus tree topology nor the number of duplications but would remove the loss of l' in s'', thus reducing the number of losses by one. In either alternative scenario, the number of extra lineages may also be affected. However, what is important is that the original scenario, with the post-coalescence duplication, has the same locus tree topology but a locus tree-species tree reconciliation that does not minimize the number of duplications and losses, thus completing our proof. □

Proof of Theorem S2.1

Strictly speaking, we do not require LCA reconciliation between the locus tree and species tree. Rather, we require the following weaker condition: post-coalescence duplications have not occurred. We have chosen to

phrase Theorem S2.1 in terms of the LCA reconciliation between the locus tree and species tree since the concept of post-coalescence duplications is not intuitive and the former implies the latter (Proposition S1.2).

Consider the problem of mapping explicit nodes in the coalescent tree to a node in the species tree. The outline of our proof is as follows:

1. Let $\alpha^* = \langle \mathcal{M}^*, \mathcal{L}^*, \mathcal{O}^* \rangle$ be a MP LCT in which $\mathcal{M}^*$ is not the LCA reconciliation.
2. There exists an explicit node $g \in I(G)$ such that $\mathcal{M}^*(g)$ is not the LCA mapping.
3. There exists an alternative LCT $\bar{\alpha} = \langle \bar{\mathcal{M}}, \bar{\mathcal{L}}, \bar{\mathcal{O}} \rangle$ in which g is mapped "down" the species tree and $\bar{\alpha}$ implies the same number of duplications and losses as α but implies one fewer extra lineage.
4. The reconciliation cost of the alternative LCT is lower than than of the original LCT, that is, $\mathcal{R}_{\bar{\alpha}} < \mathcal{R}_{\alpha^*}$.
5. α^*, in which $\mathcal{M}^*$ is not the LCA reconciliation, cannot be a MP LCT.

Obviously, 1 implies 2, 3 implies 4, and 4 implies 5; thus we only need to prove that 2 implies 3. For notational simplicity, we remove the asterisks and consider $\alpha = \langle \mathcal{M}, \mathcal{L}, \mathcal{O} \rangle$. Supplemental Figure S3 may also be useful in following this proof. Finally, note that this proof is conceptually simple but requires a large amount of setup and mathematical notation to be precise.

If $\mathcal{M}$ is not a MP LCT, let $g \in I(G)$ denote an explicit node that is not mapped to the LCA, and let g' and g'' denote the children of g. By definition, g can be mapped down the species tree, meaning that g' and g'' must evolve down the same child species. That is, g' and g'' must map to the same child species or to the descendants of the same child species. Without loss of generality, we assume the former; else, nodes can be added along (g, g') and (g, g'') until the new children of g map to the same child species. Formally, let $s \in I(S)$ be the species such that $\mathcal{M}(g) = s$, let $s' \in c(s)$ be the child species such that $\mathcal{M}(g') = \mathcal{M}(g'') = s'$, and let s'' be the child species that is not s'. Also, let $l = \mathcal{L}(g)$.

For our proof, we must consider how the implied evolutionary events change between this original scenario $\alpha = \langle \mathcal{M}, \mathcal{L}, \mathcal{O} \rangle$ and an alternative scenario $\bar{\alpha} = \langle \bar{\mathcal{M}}, \bar{\mathcal{L}}, \bar{\mathcal{O}} \rangle$ in which g is mapped down the species tree to s'. Recall that we have assumed α is a MP LCT; that is, $\mathcal{L}$ and $\mathcal{O}$ are an optimal locus map and order for the species map $\mathcal{M}$. However, in the alternative scenario, we have full control over $\bar{\mathcal{L}}$ and $\bar{\mathcal{O}}$. In particular, these do not have to be an optimal locus map and order for the species map $\bar{\mathcal{M}}$; we must only show that there exists a $\bar{\mathcal{L}}$ and $\bar{\mathcal{O}}$ for $\bar{\mathcal{M}}$ such that $\bar{\alpha}$ implies the same number of duplications and losses as α but fewer extra lineages.

Now let us understand how the gene tree maps within the species tree as implied by $\mathcal{M}$ and $\bar{\mathcal{M}}$. Because we have assumed no post-coalescence duplications, delayed speciation nodes are unnecessary (Proposition S1.1). Therefore, for α, g is mapped to the bottom of species tree branch $e(s)$ and lineages (g, g') and (g, g'') are found at the top of species tree branch $e(s')$. However, for $\bar{\alpha}$, an implied speciation node is necessary so as not to force the original parent node of g to map to the bottom of species tree branch $e(s)$. That is, let $pg = p(g)$ denote the original parent node of g; we break up (pg, g) by adding implied speciation node sg (and setting $\mathcal{M}(sg) = s$); this creates branches $(pg, sg), (sg, g)$ with lineage (sg, g) at the top of species tree branch $e(s')$.

Finally, consider possible cases for α. Recall that we have assumed no post-coalescence duplications, that is, duplications are not allowed along either child branch of the explicit node so $\mathcal{L}(g') = \mathcal{L}(g'') = l$. Therefore, we must consider two cases:

1. No duplication has occurred along the branch leading to the explicit node g (Supplemental Figure S3A, top): $\mathcal{L}(p(g)) = l$. For $\bar{\alpha}$, choose $\bar{\mathcal{L}}$ to "match" $\mathcal{L}$ (Supplemental Figure S3A, bottom): $\bar{\mathcal{L}}(sg) = l$, and for all other nodes, $\bar{\mathcal{L}}(v) = \mathcal{L}(v)$ (where we have used v so as not to confuse a node with g). Also, choose $\bar{\mathcal{O}}$ to "match" $\mathcal{O}$: all v remain in the same position except g is replaced by sg for $\bar{O}(s, l)$ and g is moved to the first position in $\bar{O}(s', l)$, assuming that l is a parent locus of a duplication in s and s', respectively. Then, comparing α and $\bar{\alpha}$, the only difference is in the number of lineages that exist at the species breakpoint between s and s'; thus, the numbers of duplications, losses, and extra lineages due to duplications remain unchanged. To consider how the number of extra lineages due to speciations is affected, we determine the set $\mathfrak{X}$ of lineages in locus l at the top of $e(s')$ as all other species and loci are unaffected. For α, $\mathfrak{X}$ includes lineages (g, g'), (g, g''), and possibly k other lineages, thus incurring $k+1$ extra lineages. In comparison, for $\bar{\alpha}$, $\mathfrak{X}$ includes lineage (sg, g) and k other lineages, thus incurring k extra lineages. That is, $\bar{\alpha}$ implies one fewer extra lineage (due to speciation) than α.
2. A duplication has occurred along the branch leading to the the explicit node g (Supplemental Figure S3B, top): $\mathcal{L}(p(g)) \neq l$. For $\bar{\alpha}$, as with the previous case, we choose $\bar{\mathcal{L}}$ and $\bar{\mathcal{O}}$ to "match" $\mathcal{L}$

and $\mathcal{O}$, respectively, with the difference that the duplication along $(p(g), g)$ in α remains in species s but occurs along (pg, sg) in $\bar{\alpha}$ (Supplemental Figure S3B, bottom). Then, a comparison of implied evolutionary events for α and $\bar{\alpha}$ is identical to the previous case, that is, the numbers of duplications, losses, and extra lineages due to duplications remain unchanged but the number of extra lineages due to speciations decreases by one.

Remark. We now consider the implications of allowing post-coalescence duplications. If a post-coalescence duplication has occurred, delayed speciation nodes are necessary (Proposition S1.1). That is, for α, we break up (g, g') and (g, g'') by adding implied speciation nodes sg' and sg'', respectively, thus creating branches $(g, sg'), (sg', g'), (g, sg''), (sg'', g'')$ with $\mathcal{M}(sg') = \mathcal{M}(sg'') = s$ and with lineages $(sg', g'), (sg'', g'')$ at the top of species tree branch $e(s')$. Furthermore, the locus of at least one node sg' or sg'' is different from the locus of g; assume that $\mathcal{L}(sg'') \neq l$ (Supplemental Figure S3C). Then shifting g in the alternative scenario $\bar{\alpha}$ may induce *more* events. In particular, sg' and sg'' do not exist in $\bar{\alpha}$, but setting $\bar{\mathcal{M}}(g) = s'$ would require us to determine where to place the duplication originally found along (g, sg''). If the duplication remains in species s (that is, $\bar{\mathcal{L}}(pg) \neq \bar{\mathcal{L}}(sg)$), then $\bar{\mathcal{L}}(g') = \bar{\mathcal{L}}(g'')$ so that $\bar{\alpha}$ may no longer be a valid LCT. Adding another duplication along (g, g'') to force $\bar{\mathcal{L}}(g') \neq \bar{\mathcal{L}}(g'')$ would, of course, go against our efforts by incurring an additional duplication. Otherwise, if the duplication is "pushed" to occur along (g, g'') in species s' (that is, $\bar{\mathcal{L}}(g) \neq \bar{\mathcal{L}}(g'')$), then $\bar{\alpha}$ may incur more extra lineages due to duplication compared to α. This is because the duplication now occurs later in the gene evolutionary history; therefore, bifurcations in the gene tree in locus l between the time of the old duplication event in α and the new duplication event in $\bar{\alpha}$ could result in additional lineages in locus l that are contemporaneous to the duplication along (g, g'') in $\bar{\alpha}$.

Proof of Proposition S2.2

We prove each component separately:

1. Allowing multiple changes would incur additional duplications and losses, and possibly additional extra lineages, and therefore be strictly less parsimonious.

2. Consider a gene tree node that is internal to a species tree branch and has two children that map to distinct loci that are both different from the locus of their shared parental node (ex. Figure 3B, map 6); such a scenario would correspond to a locus duplicating twice to create two daughter loci, possibly followed, as in the figure, by the original mother locus becoming lost. An alternative explanation is for the original locus to duplicate, then for the new locus to duplicate, again possibly followed by the original locus becoming lost (ex. Figure 3B, map 5). These two evolutionary histories are identical in terms of locus assignments after the double round of duplications, and the latter is always at least as parsimonious as the former. (In particular, extra lineages can be incurred in the former if there exist additional lineages in the same locus. In the figure, an additional red lineage in maps 5 and 6 would result in one extra lineage at the root of the species tree branch in both maps, but map 6 would also incur an extra lineage at the first duplication.)

3. Incorporating ILS allows a decrease in the number of inferred duplications. If extra lineages have a low cost, very few duplications can be inferred, in particular fewer duplications than in a duplication-loss model. As this cost increases, reconciliation reverts to a duplication-loss model, thus giving an upper bound on the maximum number of duplications, and consequently, the maximum number of loci, in the MP LCT.

Proof of Proposition S2.3

Recall that $\mathcal{O}$ defines ordered sets $\hat{O}(s, l) \, \forall s \in V(S), l \in \mathit{parent\_loci}(s)$. Furthermore, $\hat{O}(s, l)$ only affects the number of extra lineages due to duplications, and each $\hat{O}(s, l)$ affects this number independently. This observation allows to infer an optimal ordered set $\hat{O}^*(s, l)$ independently for each $s \in V(S), l \in \mathit{parent\_loci}(s)$. In particular, to find $\hat{O}^*(s, l)$, we begin with $\mathit{start\_lineages}(s, l)$ and intelligently choose the next node $g \in O(s, l)$ until all nodes have been exhausted. To optimally select g, realize that the lineage count in l decreases by one every time $\mathcal{L}^*(g) \neq l$ (due to a duplication). Otherwise, assuming $\mathcal{L}^*(g) = l$, if $g \in \mathit{leaves}(s)$, then the lineage count either is unchanged (if $|c(g)| \leq 1$; note that if $|c(g)| = 0$, then $s = L(S)$ but extant

genes exist to present day and are not removed from the set of lineages) or increases by one (if $|c(g)| = 2$), and if $g \in nodes(s) \setminus leaves(s)$, then the lineage count increases by one (since $|c(g)| = 2$). That is, choosing the next node g: $\mathcal{L}^*(g) \neq l$ decreases the lineage count, and choosing g: $\mathcal{L}^*(g) = l$ either has no effect or increases the lineage count. Thus, $\hat{O}^*(s, l)$ should have all duplications as early in the species tree branch as possible.

Remark. Note that we have *not* shown that all $\mathcal{O}^* = \arg\min_{\mathcal{O}} R_{\langle \mathcal{M}^*, \mathcal{L}^*, \mathcal{O} \rangle}$ satisfy this property. This theorem simply states that this particular $\mathcal{O}^*$ is an optimal solution.

S4 Comparison with the Three-Tree Model

The idea for separating the locus history and coalescent history of a gene family was initially proposed in the three-tree model of DLCoal (Rasmussen and Kellis 2012). There, the locus tree was introduced to trace the history of a pool or set of sequences, namely all of the sequences in a population that belong to the same species and the same locus. This is important as coalescence is restricted to sequences within the same pool. In addition, each daughter edge (defined in the locus tree as the child lineage of a duplication that has evolved in the new locus) has complete coalescence, that is, only one gene lineage is present at the top of each daughter edge.

For a given gene tree and species tree, provided that the underlying evolutionary model used in phylogenetic reconstruction is compatible with the three-tree model, the reconciliation problem is to infer both the "hidden" locus tree and the reconciliations between the three trees. DLCoal (Rasmussen and Kellis 2012) separates the locus history and coalescent history of a gene family by explicitly representing the locus tree and two-step reconciliations. Therefore, DLCoalRecon should ideally search over the space of locus trees and reconciliations. However, the number of distinct locus tree topologies for a given set of extant sequences is exponential in the number of extant sequences, making a full search infeasible. Furthermore, for a gene tree, locus tree (with annotated daughter nodes), and species tree, only certain reconciliations are valid; in particular, despite the LCA reconciliation being the optimal solution for minimizing deep coalescence and minimizing duplications and losses, LCA reconciliation between the gene tree and locus tree and between the locus tree and species tree may violate the requirement of complete coalescence in a daughter edge.

The LCT overcomes these challenges by collapsing the locus tree and reconciliations into the gene tree. We accomplish this by tracking the locus directly within the gene tree. In addition, the LCT is a simplified representation of the three-tree model. In particular, inferring the duplication-loss history of a gene family requires a locus tree *topology* and its reconciliation to the species tree. That is, the dates of the ancestral nodes in the locus tree are unknown. To overcome this problem, DLCoal models fully dated locus trees, and DLCoalRecon integrates over many samples of duplication times (as speciation times are fixed by the species tree) to yield the *max a posteriori* locus tree topology. In contrast, the LCT models only the locus tree topology. In addition, though loci change along gene tree branches, the LCT locus map labels gene tree nodes. In effect, the LCT rounds locus changes to the nearest node: for $g \in V(G)$: $g \neq r(G)$, if $\mathcal{L}(g) \neq \mathcal{L}(p(g))$, then the locus changes immediately before g. Despite these simplifications, the LCT can model the same evolutionary histories captured by the three-tree model; this includes the popular and simple duplication-loss only or coalescent-only models for gene evolution (Supplemental Figure S1B,C) as well as more complicated histories involving duplications, losses, and ILS.

For the reconciliation problem, the LCT and DLCpar provide further advantages. Motivated by the need for simple and efficient ILS-aware reconciliation algorithms, we chose to adopt a maximum parsimony approach for DLCpar. This means fewer assumptions on the evolutionary model. As an example, unlike DLCoalRecon, we do not assume that the locus tree evolves within the species tree according to a birth-death process (Arvestad et al. 2004, Dubb 2005, Åkerborg et al. 2009, Rasmussen and Kellis 2011). Furthermore, we can directly take into account the restrictions placed on a MP LCT by the gene tree and species tree. This allows us to infer how the gene tree maps to the species tree (bypassing the locus tree) as well as enumerate over the space of valid reconciliations. That is, we have *a priori* restricted the reconciliation search space. In comparison, since it is impractical to search through the space of all possible reconciliations using the three-tree model, DLCoalRecon uses a search strategy in which reconciliations are proposed then checked for validity.

Finally, due to the small number of trees that have post-coalescence duplications, we have chosen to

make DLCpar more efficient by disallowing post-coalescence duplications; thus, delayed speciation nodes are not required (Proposition S1.1. However, note that no post-coalescence duplications does *not* imply LCA reconciliation between the locus tree and species tree. This is an additional advantage of DLCpar over DLCoalRecon; namely, while DLCoalRecon only searches over the space of locus trees and reconciliations such that the reconciliation between the locus tree and the species tree is the LCA mapping, DLCpar actually allows non-LCA reconciliation between the locus tree and species tree.

S5 Relating Duplication-Loss Events and Deep Coalescence Events in a Parsimony Framework

Here, we discuss the implications of Zhang (2011) on our work. In particular, for gene tree G and species tree S, Zhang (2011) presented a simple formula relating the number $n_C(G,S)$ of extra lineages (induced by deep coalescence) to the numbers $n_D(G,S)$ and $n_L(G,S)$ of duplications and losses; however, his work applied the duplication-loss and deep coalescence parsimony scores *separately* to the reconciliation between gene tree and species tree. We highlight that our work *jointly* models duplication-loss and ILS. This is an important distinction as it means that, for many gene families (in particular, those affected by duplication-loss and ILS), the results of Zhang (2011) do not necessarily apply.

What about distinguishing between gene trees that are affected only by duplication-loss (DL) or deep coalescence (DC)? Recall that we have assumed that genes that map to the same species must be paralogous (as they belong to different loci), and contrast this with coalescent theory that traces orthologous genes. This means that if multiple gene copies exist within any species, the gene tree must include duplications, and distinguishing between DL and DC is moot. So let us restrict our analysis to uniquely-labeled gene trees. For simplicity, we first consider equal costs for duplication, loss, and deep coalescence. In this case, applying the results of Zhang (2011), the DC cost is always less than the DL cost:

$$\text{DC cost} = n_C = n_L - 2 \cdot n_D < n_D + n_L = \text{DL cost}$$

where we have hidden the dependencies on G and S for notational simplicity. This suggests that, if the gene tree is incongruent to the species tree, we would always infer deep coalescence over duplication-loss. We argue that this is acceptable as one-to-one gene families often assume that all genes are orthologous to one another, so inferring deep coalescence is the "correct" solution. If the costs for duplication, loss, and deep coalescence are unequal, then raising the cost C of deep coalescence (or lowering the costs D and L of duplications and losses) could result in the DC cost being greater than the DL cost:

$$\text{DC cost} = C \cdot n_C = C \cdot (n_L - 2 \cdot n_D) > D \cdot n_D + L \cdot n_L = \text{DL cost}$$

This is, of course, exactly what we would want, since, choosing such costs implies that we prefer inferring DL over DC.

S6 Duplication, Loss, and Coalescence Costs

As the locus tree is used to infer duplications, losses, orthologs, and paralogs, we assessed the robustness of DLCpar to different event costs by analyzing the similarity of inferred locus trees. Using the parameter setting $D = L = 1$, $C = 0.5$ as a reference, the Robinson-Foulds (RF) distances (Robinson and Foulds 1981) of locus trees reconstructed under varying costs were < 0.040 for fly population sizes ≤ 100 million (max RF $= 0.125$ for $N = 500$ million, $4\times$ rate) and < 0.022 for all primate datasets. We then analyzed whether this similarity in locus tree topology translates to accurate event inference (Supplemental Figure S6). While we have simulated a wide range of population sizes in order to analyze reconciliation performance for datasets with a large amount of ILS, a biologically realistic range of population sizes is on the order of 1–50 million for *Drosophila* and 10–50 thousand for primates (Charlesworth 2009). In this population range, with a duplication-loss rate equal to the estimated real rate, DLCpar performance as measured by topological accuracy varies by 0.2–4.2% to a low of 94.8%, duplication precision by 0.2–9.6% to a low of 87.6%, and loss precision by 0.4–17.2% to a low of 82.8%, with other metrics varying up to 9.6% to a low of 87.6%.

If we consider only the primates datasets for which biologically reasonable population sizes change by 5×, compared to the 50 million × difference in *Drosophila*, performance variation across all metrics is $< 4.0\%$, with the lowest metric at 94.6%. For larger population sizes or higher duplication-loss rates, more care must be taken in choosing event costs, but as expected, as population size, and consequently ILS rate, increases, a smaller coalescence cost (relative to duplication and loss costs) yields better performance.

S7 Search Heuristics

Some gene trees may be too complex for DLCpar to reconcile. In particular, large gene trees or gene trees highly incongruent to the species tree contain many implied gene tree nodes, thus increasing the LCT search space. To address this problem, we introduce two heuristics to the DLCpar algorithm.

The first heuristic bounds the LCT search space by limiting the maximum number of duplications, losses, or loci in each species tree branch and by prescreening proposed locus maps. For the prescreen, when enumerating locus maps, we keep track of the cost-to-go of varying loci assignments at the bottom of each species tree branch, where the cost-to-go is the minimum total cost along all ancestor branches. If the minimum cost-to-go c_{min} across all loci assignments exceeds a threshold p_{min}, then loci assignments with cost-to-go exceeding $p_{factor} * c_{min}$ are pruned. Based on analysis of the simulated datasets, we chose default parameters of 4 duplications and 4 losses per species tree branch (no restriction on the maximum number of loci) and $p_{min} = 5$, $p_{factor} = 2$. We find this heuristic to have limited effect on algorithm performance (Supplemental Figure S7). Compared to DLCpar, RF distances for DLCpar-bound are < 0.008 for fly population sizes ≤ 100 million (max RF $= 0.032$ for $N = 500$ million, 4× rate) and < 0.004 for all primate datasets.

The second heuristic further limits the LCT search space by using the three-tree model and the hill-climbing search strategy of DLCoalRecon (Rasmussen and Kellis 2012). We chose default search parameters of 1000 iterations and 20 prescreens. Compared to DLCpar, RF distances for DLCpar-search are < 0.057 for fly population sizes ≤ 100 million (max RF $= 0.142$ for $N = 500$ million, 4× rate) and < 0.031 for all primate datasets.

S8 Simulated Species Trees

We also assessed how DLCpar performance changes with varying speciation rates and species tree sizes. Here, we used the simulated species trees of Wu et al. (2013), which were generated by TreeSample (Hartmann et al. 2010) with a constant-rate birth-death model, a variety of settings for the speciation rate (0.05–1 events/species/myr) and tree size (5–100 extant species), and a fixed speciation rate-extinction rate ratio ($\mu = 0.9\lambda$). For each species tree, we used the DLCoal model, with parameters from the *Drosophila* clade, to simulate gene trees, then used DLCpar, DLCoalRecon, and MPR to reconcile these gene trees with the species tree. As before, DLCpar used the same event costs across all settings whereas DLCoalRecon was tuned to each simulation setting. As with the simulated gene trees reconciled to real species trees, almost universally, DLCpar shows dramatic improvement over other reconciliation programs (Supplemental Figure S9).

As the speciation rate increases, resulting in shorter branches in the species tree and therefore fewer duplications and losses and more deep coalescence, a number of trends emerge. MPR performs consistently low, and across all metrics, performance decreases with increasing speciation rate. The latter observation suggests that the reconciliation problem becomes more difficult as the speciation rate increases and that the increased ILS rate overcomes the lower number of implanted duplications and losses to result in more gene tree-species tree incongruence. Of particular note is the dramatic drop in branch accuracy of the locus tree, which was shown to be a robust metric in the gene tree reconstruction problem (Rasmussen and Kellis 2011, Wu et al. 2013). In contrast, with increasing speciation rate, DLCpar and DLCoalRecon branch accuracies remain consistently high, but more interestingly, topological accuracies increase. This is the reverse of our observation in the previous simulation study, in which we found locus tree topological accuracy to be stable or slightly decrease with increasing deep coalescence. The difference is that previously, increased population sizes resulted in increased ILS rates without affecting the number of implanted duplications and losses, whereas here, decreased speciation times result in increased ILS rates and fewer implanted duplications and losses. Finally, for the last metrics, duplication and loss precision of DLCpar and DLCoalRecon decrease

with increasing speciation rate, with DLCpar slightly outperforming DLCoalRecon in duplication precision but DLCoalRecon holding a slight (but insignificant) edge in loss precision (where significance is measured by overlap in the 95% confidence intervals). Overall, the robust performance of DLCpar, and to a lesser extent, DLCoalRecon, suggests that varying speciation rates have limited effect on the accuracy of inferred reconciliations when an ILS-aware method is used.

As the species tree size increases, larger numbers of duplication, loss, and coalescent events lead to more gene tree-species tree incongruence and thus a more difficult inference problem. This is evident as all three reconciliation programs show decreased locus tree topological accuracy with increasing tree size. Across the other metrics, MPR performance is consistently low regardless of tree size, and except for the relatively robust metric of branch accuracy, DLCoalRecon performance decreases rapidly with increasing tree size. In contrast, DLCpar performance is highly robust to tree size. The noticeable exception is that DLCpar loss precision is lower than that of DLCoalRecon for small species trees, but at these sizes, very few (single digit) duplications and losses are simulated, meaning that small counts are likely the source of low precision and explaining the larger variance in precision at small tree sizes. For larger tree sizes, DLCpar performs substantially better than DLCoalRecon despite the average run time of DLCoalRecon being 16.8 (9.8)$\times$ that of DLCpar for 50 (100) extant species. This is again likely attributable to the heuristic search strategy and limited search space of DLCoalRecon. In evaluation, DLCoalRecon searched over the same number of reconciliations regardless of species tree size, whereas DLCpar, by virtue of its algorithm, searched over increasingly larger spaces as the species tree size increased.

Supplemental Figures and Tables

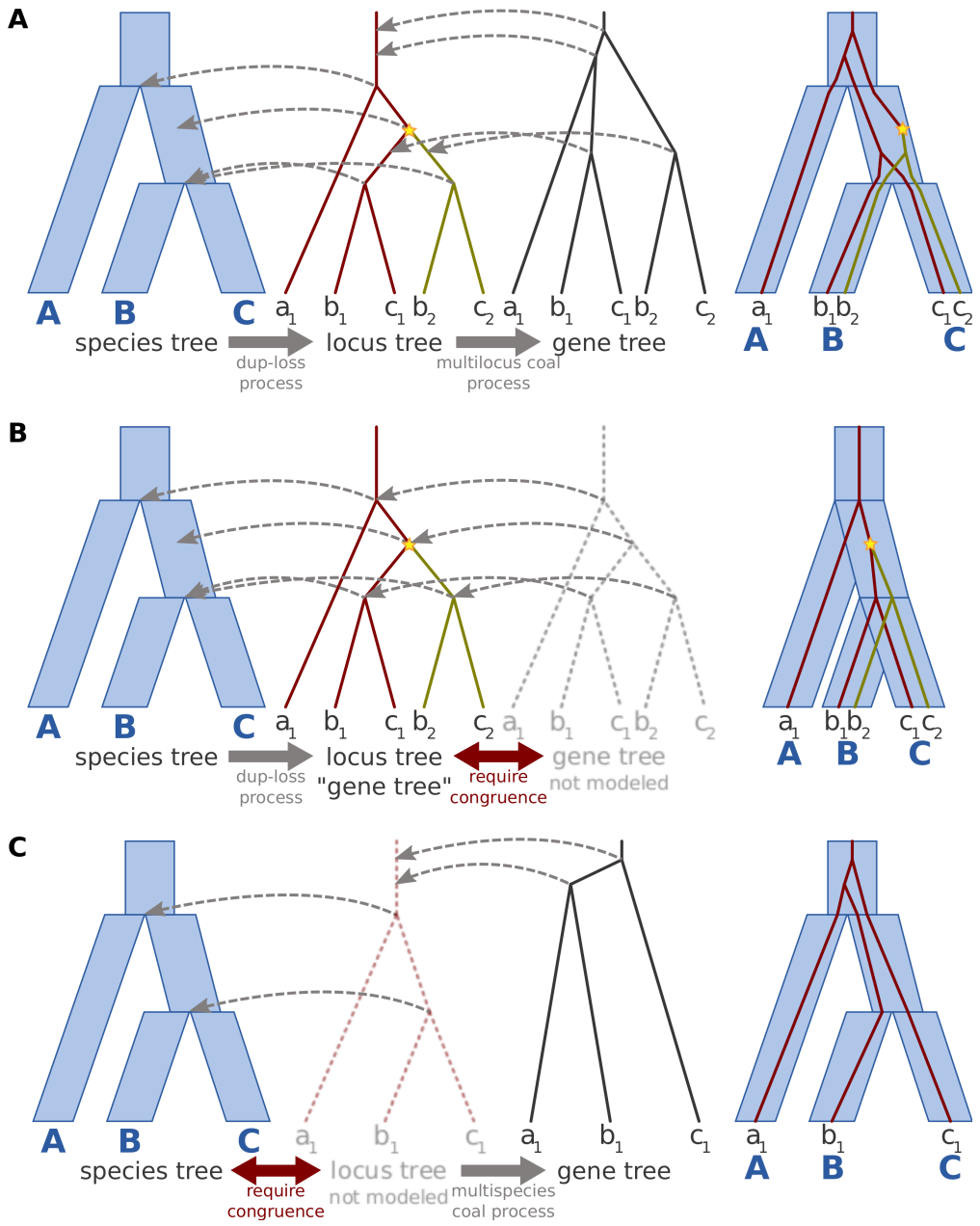

Figure S1. Models for gene family evolution. Evolution of a gene family is shown using the three-tree model (left) and the labeled coalescent tree (right). (*A*) In the three-tree model, the (coalescent) gene tree reconciles to the locus tree, allowing inference of ILS, and the locus tree reconciles to the species tree, allowing inference of duplications and losses. (*B*) In a duplication-loss model, ILS is assumed not to occur; thus, the gene tree and locus tree are congruent. Therefore, (1) effectively, the daughter lineage of a duplication coalesces immediately with its parent lineage and (2) within each species, at most one lineage exists per locus. In the LCT, this translates to (1) loci changes occur at nodes and (2) all coexisting lineages within the same species tree branch belong to different loci. (*C*) In a multispecies coalescent model, duplications and losses are assumed not to occur; thus, the locus tree and species tree are congruent. Therefore, there exists a single gene per extant species, and all lineages evolve within the same locus. [Parts of this figure have been adapted with permission from (Rasmussen and Kellis 2012).]

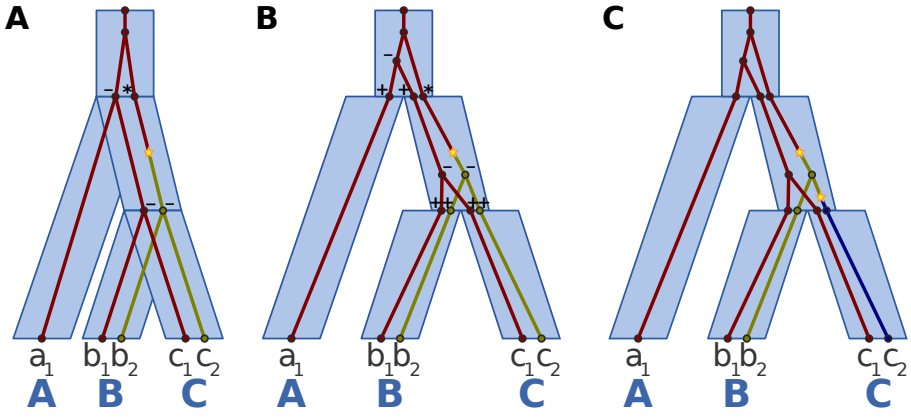

Figure S2. Delay between speciation and coalescence. (*A*) Explicit (−) and implicit (∗) speciation nodes separate the gene tree into disjoint subtrees that evolve within the species tree branches. Note that speciation nodes are rounded to the breakpoint between species. (*B*) Delay between speciation and coalescence causes explicit speciation nodes to be found earlier in the tree and adds delayed speciation nodes (+). It is now the delayed and implicit speciation nodes that separate the gene tree. (In particular, previously defined explicit speciation nodes are no longer speciation nodes.) Note that this models the same three trees as before (albeit with different timings), so delayed speciation nodes are unnecessary. (*C*) Delayed speciation nodes are necessary when post-coalescence duplications have occurred (new locus in dark blue). These duplications could instead be inferred in the child species, which would reduce the number of losses without impacting the locus tree topology. Thus, such duplications are not possible if we assume LCA reconciliation between the *locus tree* and species tree, as the LCA in this instance minimizes the number of duplications + losses. In this example, duplication in the ancestral species of species B and species C is followed by loss of the dark blue locus in species B and loss of the yellow locus in species C. Inferring the duplication in species C instead would reduce the number of losses by one since the dark blue locus is no longer lost in species B.

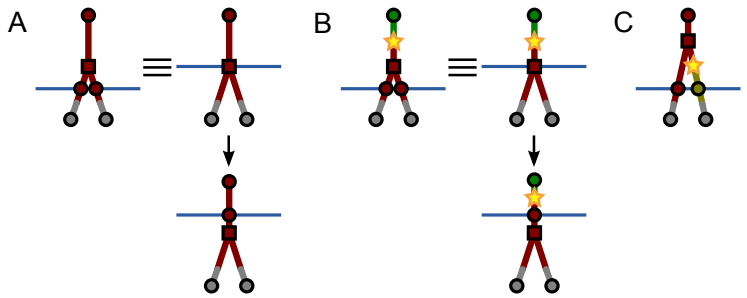

Figure S3. Reducing the number of inferred evolutionary events by shifting gene tree nodes down the species tree. Explicit nodes in the gene tree are mapped to a species (break between species tree branches shown as blue line). Assume that there exists an explicit node (square) that is not mapped to the LCA; thus, it can be mapped "down" the species tree. Such a shift is only possible if the children of the explicit node evolve down the same child species; thus the other child branch in the species tree is not shown. Lineages with an unknown locus mapping are depicted in gray. Note that other lineages may exist; we have restricted the figure to a single triplet (parent with two children) for clarity. (*A*) No duplication has occurred along any branch incident to the explicit node (top left), so delayed speciation nodes are unnecessary (top right). Shifting the explicit node incurs one fewer extra lineage due to speciation (bottom). In this example, in a non-LCA mapping, two red lineages at the species break implies one extra lineage (top), whereas, by shifting the explicit node, one red lineage at the species break implies no extra lineages (bottom). The numbers of implied duplications, losses, and extra lineages due to duplications are unchanged. (*B*) A duplication has occurred along the parent branch of the explicit node (top); if the explicit node is shifted to the child species but the duplication remains in the original species (bottom), this is identical to case A. (*C*) A duplication has occurred along a child branch of the explicit node, between the time of coalescence and the speciation event; such a duplication is a post-coalescence duplication and is not allowed if we assume LCA reconciliation between the locus tree and species tree. That is, the lineage with the new locus (yellow) evolves down a single child species, so the duplication can be shifted to occur in the child species, immediately after the speciation, which would incur one fewer loss without impacting the locus tree topology. Note that if the lineage with the new locus evolved down multiple children species (not shown), the duplication could *not* be shifted because the speciation node with the new locus would have two children, making it impossible to shift the duplication down a single lineage.

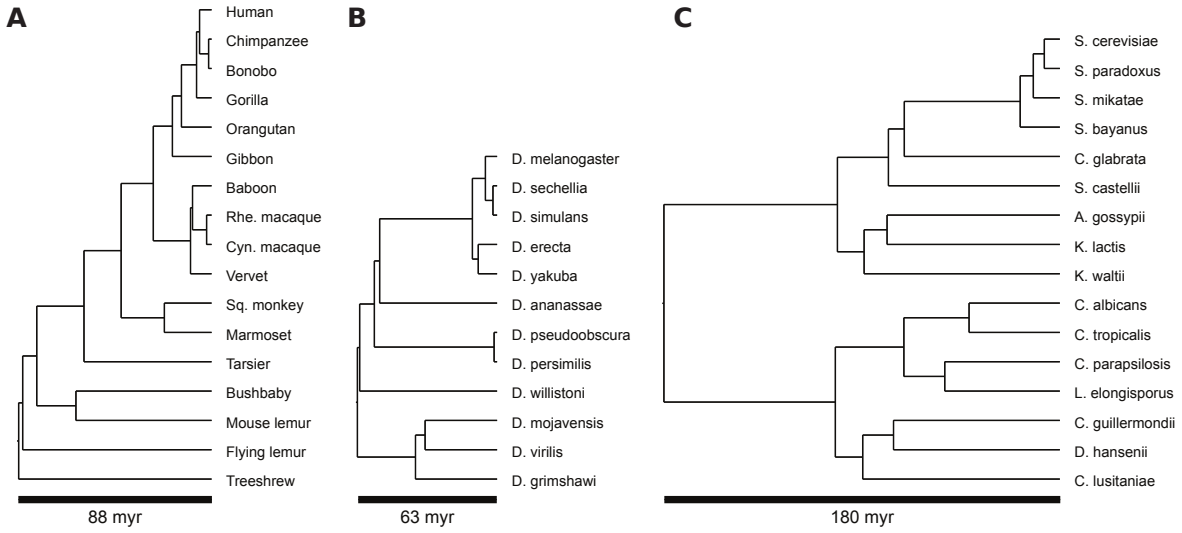

Figure S4. Species and phylogenies used in evaluation. For our evaluation on simulated data, we used (*A*) 15 primates (plus two outgroup species), (*B*) 12 *Drosophila* species, as well as simulated phylogenies. (*C*) For our evaluation on real data, we used 16 fungal species.

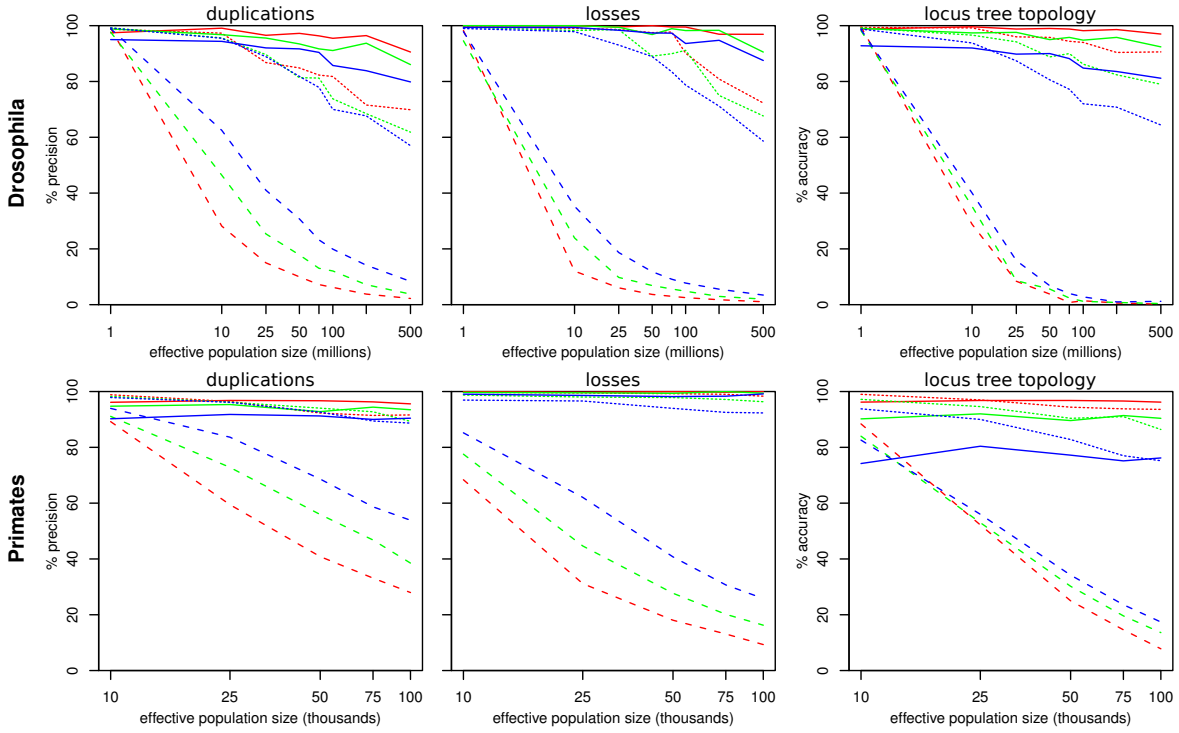

Figure S5. Phylogenetic accuracy of DLCpar on simulated fly and primate gene trees. DLCpar (solid), DLCoalRecon (dot), and MPR (dash) were used to reconcile simulated gene trees. Duplication and losses were simulated at rates that were the same as (1×, red), twice (2×, green), and four times (4×, blue) the rate estimated in real data. See also Figure 4, which corresponds to simulation at 1× rate.

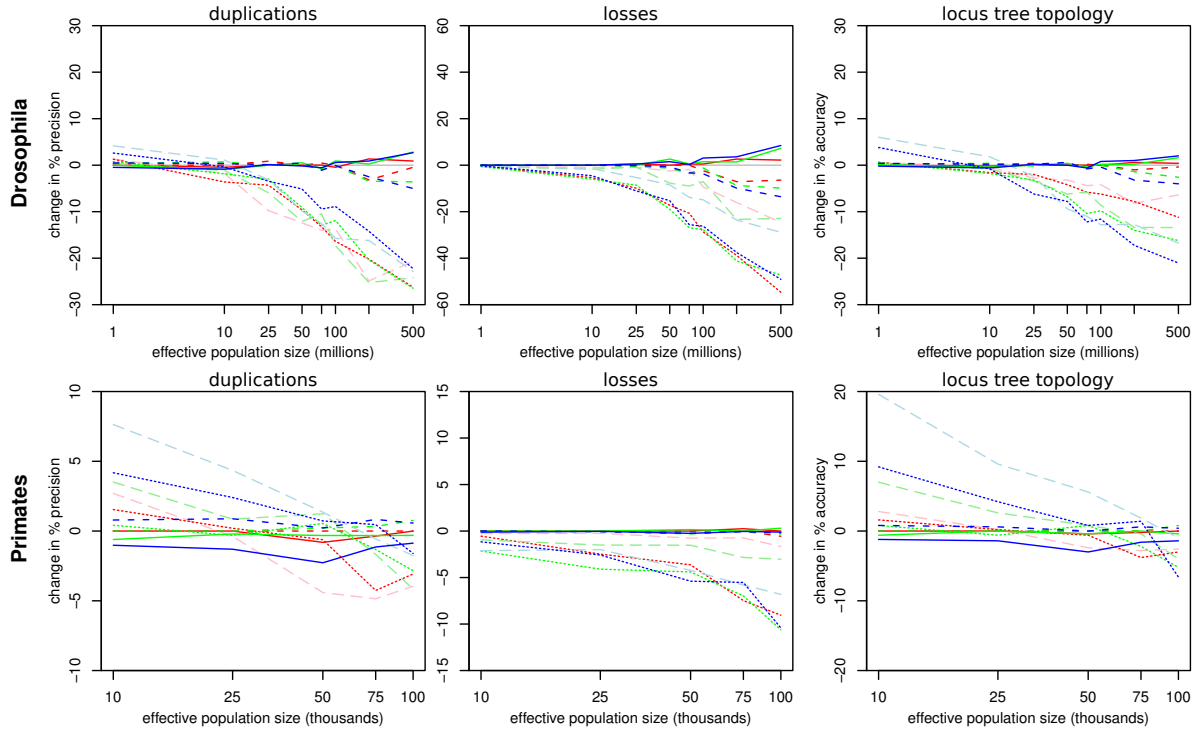

Figure S6. Phylogenetic accuracy of DLCpar on simulated gene trees using various event costs. DLCpar was run on simulated fly and primate datasets using $D = L = 1$, and a range of costs C for extra lineages. The difference in precision or accuracy compared to DLCpar with $C = 0.5$ is shown for $C = 0.25$ (solid), $C = 0.75$ (dash), $C = 1$ (dot). A comparison to DLCoalRecon is also shown (lighter colors, long dash), and the gray line indicates no change. For details on the simulation procedure, see Figure 4 and Supplemental Figure S5.

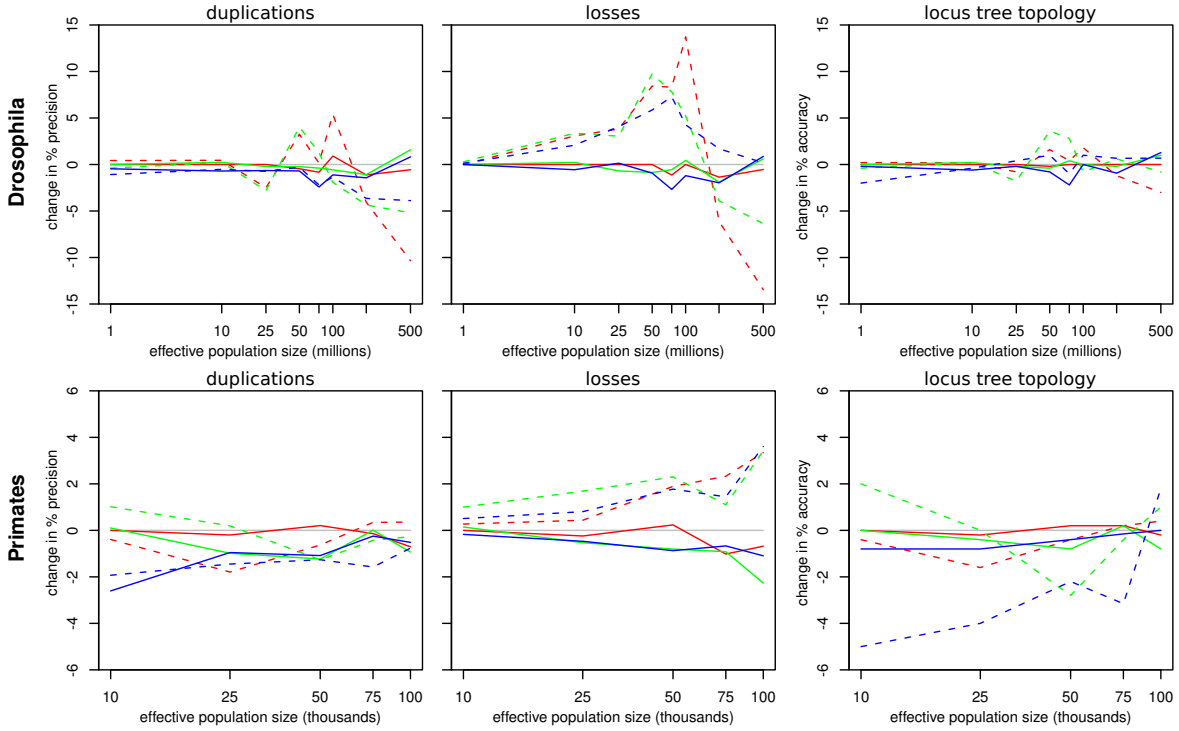

Figure S7. Phylogenetic accuracy of DLCpar on simulated gene trees using search heuristics. DLCpar was run on simulated fly and primate datasets using $D = L = C = 1$ and two search heuristics. The difference in precision or accuracy compared to DLCpar (without search heuristics) is shown, with the gray line indicating no change. In DLCpar-bound (solid), the search space was limited by pre-screening locus maps and enforcing upper bounds on the maximum number of duplications and losses per species. In DLCpar-search (dash), the three-tree model rather than the LCT model was used, and a local hill-climbing strategy was employed to search the space of locus trees and reconciliations (as in DLCoal). For details on the simulation procedure, see Figure 4 and Supplemental Figure S5.

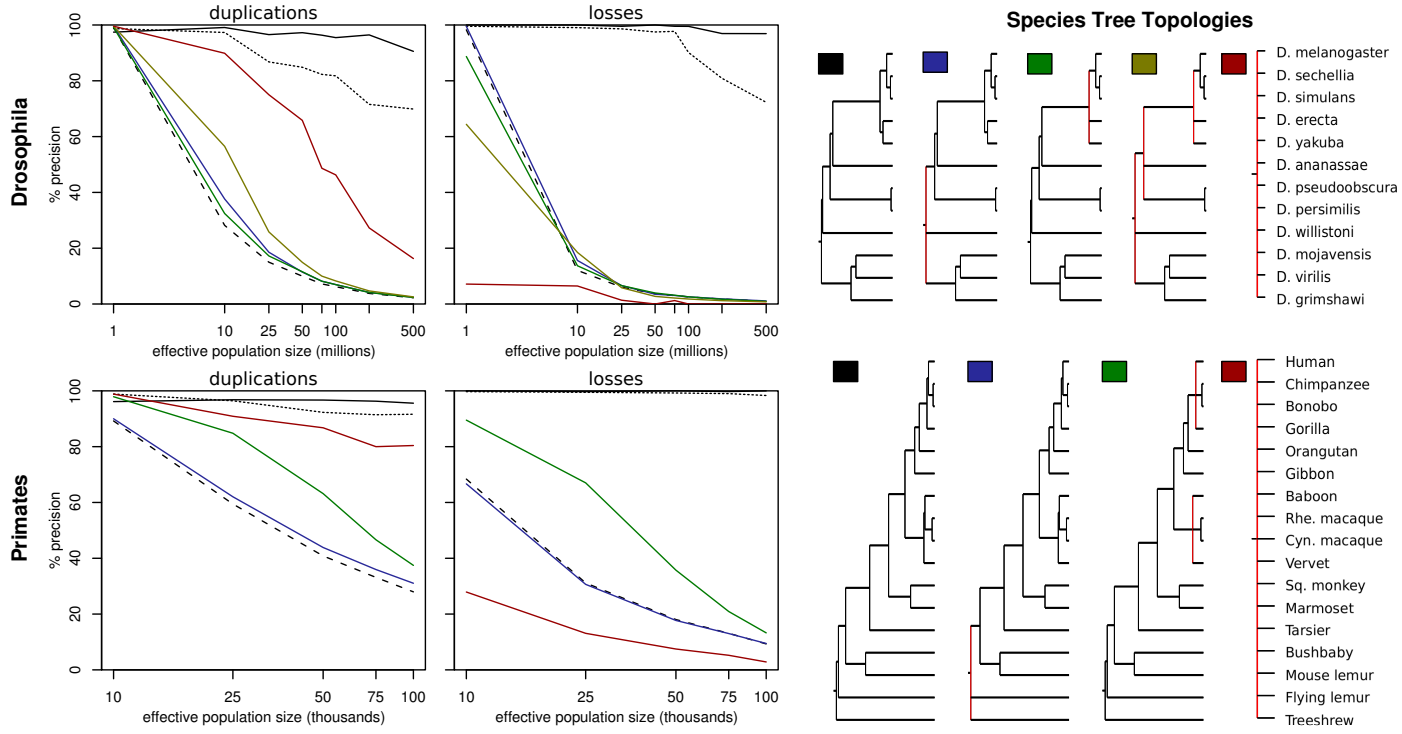

Figure S8. Phylogenetic accuracy of NOTUNG on simulated gene trees. In addition to DLCpar (black solid), DLCoalRecon (black dot), and MPR (black dash), NOTUNG (colored) was used to reconcile simulated gene trees. To infer ILS, NOTUNG reconciles binary gene trees to non-binary species trees; the species trees used in reconciliation are shown at right, with multifurcated branches highlighted in red. For the fly dataset, we collapsed the shortest branch (blue), which separates the *Sophophora* and *Drosophila* subgenera; the third shortest branch (green), which groups *D. erecta* and *D. yakuba* as sister species to *D. melanogaster* though large portions of their genomes support alternative phylogenies (Pollard et al. 2006); the three shortest branches (yellow); and all branches (red). For the primate dataset, we collapsed the third shortest branch (blue), which implies that the relationships between the outgroup species and primates are poorly resolved; the two shortest branches (green), which implies co-divergence of baboon, macaque, and vervet and of human, chimp and bonobo, and gorilla; and all branches (red). For details on the simulation procedure, see Figure 4.

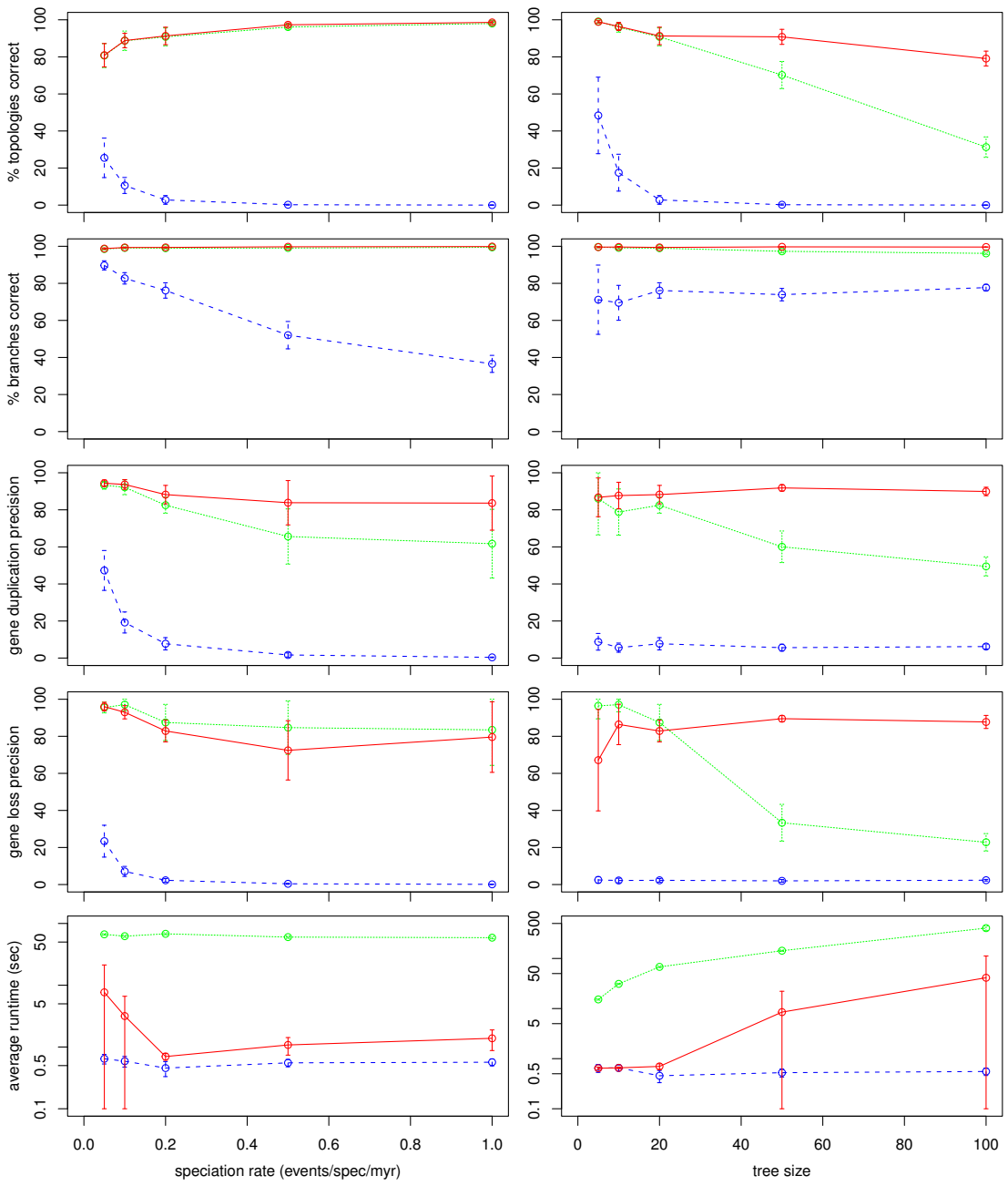

Figure S9. Phylogenetic accuracy and runtime of various reconciliation methods on simulated species trees. Simulated species trees were obtained from Wu et al. (2013), which used TreeSample (Hartmann et al. 2010) with a constant-rate birth-death model, varying speciation rates λ (events/spec/myr) and tree sizes n (number of extant taxa), and extinction rates set to $\mu = 0.9\lambda$, with 10 species trees simulated per setting. For each species tree, 100 locus trees and coalescent trees were simulated using the DLCoal model (Rasmussen and Kellis 2012), with parameters from the fly clade ($\lambda = \mu = 0.0012$ events/gene/myr, $N = 10$ million, $g = 0.1$ yr). Finally, DLCpar ($D = L = C = 1$, red solid), DLCoalRecon (green dot), and MPR (blue dash) were used to reconcile the simulated (coalescent) gene trees. Performance is measured in terms of mean accuracy/precision and runtime, with bars indicating the 95% confidence interval (mean $\pm$ 1.96 $\times$ standard error) across all species trees with the same setting. (*left*) Performance for $\lambda = 0.05, 0.1, 0.2, 0.5, 1.0$ and $n = 20$. (*right*) Performance for $\lambda = 0.2$ and $n = 5, 10, 20, 50, 100$.

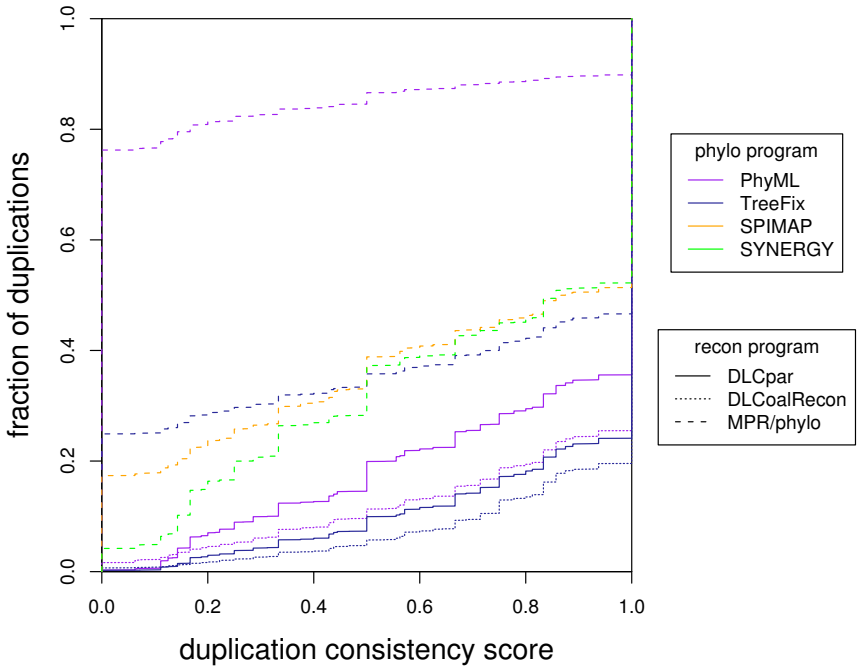

Figure S10. Cumulative distribution of duplication consistency scores for all duplications inferred on the real fungal dataset using several phylogenetic methods. Each program was used genome-wide to infer the duplications present in 16 fungal species. For each duplication, we computed the duplication consistency score, defined as $|L \cap R|/|L \cup R|$, where L and R are the sets of species present in descendants left and right of the duplication node, respectively.

References

Åkerborg Ö, Sennblad B, Arvestad L and Lagergren J. 2009. Simultaneous bayesian gene tree reconstruction and reconciliation analysis. Proceedings of the National Academy of Sciences **106**:5714–5719.

Arvestad L, Berglund A.-C, Lagergren J and Sennblad B. 2004. Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution. In Proceedings of the eighth annual international conference on research in computational molecular biology, RECOMB '04, 326–335. ACM, New York, NY, USA.

Charlesworth B. 2009. Fundamental concepts in genetics: Effective ppulation size and patterns of molecular evolution and variation. Nature Review Genetics **10**:195–205.

Dubb L. 2005. A likelihood model of gene family evolution. Ph.D. thesis, University of Washington, Seattle.

Górecki P and Tiuryn J. 2006. Dls-trees: A model of evolutionary scenarios. Theoretical Computer Science **359**:378–399.

Hartmann K, Wong D and Stadler T. 2010. Sampling trees from evolutionary models. Systematic Biology **59**:465–476.

Pollard D. A, Iyer V. N, Moses A. M and Eisen M. B. 2006. Widespread discordance of gene trees with species tree in ¡italic¿drosophila:¡/italic¿ evidence for incomplete lineage sorting. PLoS Genet. **2**:e173–.

Rasmussen M. D and Kellis M. 2011. A Bayesian approach for fast and accurate gene tree reconstruction. Mol. Biol. Evol. **28**:273–290.

Rasmussen M. D and Kellis M. 2012. Unified modeling of gene duplication, loss, and coalescence using a locus tree. Genome Res. **22**:755–765.

Robinson D and Foulds L. 1981. Comparison of phylogenetic trees. Math. Biosci. **53**:131–147.

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.

Zhang L. 2011. From gene trees to species trees ii: Species tree inference by minimizing deep coalescence events. IEEE/ACM Trans. Comput. Biol. Bioinformatics **8**:1685–1691.