Multiple Optimal Reconciliations Under the Duplication-Loss-Coalescence Model

Haoxing Du, Yi Sheng Ong, Marina Knittel, Ross Mawhorter, Nuo Liu, Gianluca Gross, Reiko Tojo, Ran Libeskind-Hadas, and Yi-Chieh Wu

Abstract—Gene trees can differ from species trees due to a variety of biological phenomena, the most prevalent being gene duplication, horizontal gene transfer, gene loss, and coalescence. To explain topological incongruence between the two trees, researchers apply reconciliation methods, often relying on a maximum parsimony framework. However, while several studies have investigated the space of maximum parsimony reconciliations (MPRs) under the duplication-loss and duplication-transfer-loss models, the space of MPRs under the duplication-loss-coalescence (DLC) model remains poorly understood. To address this problem, we present new algorithms for computing the size of MPR space under the DLC model and sampling from this space uniformly at random. Our algorithms are efficient in practice, with runtime polynomial in the size of the species and gene tree when the number of genes that map to any given species is fixed, thus proving that the MPR problem is fixed-parameter tractable. We have applied our methods to a biological data set of 16 fungal species to provide the first key insights in the space of MPRs under the DLC model. Our results show that a plurality reconciliation, and underlying events, are likely to be representative of MPR space.

Index Terms—Phylogenetics, reconciliation, coalescence, incomplete lineage sorting, gene duplication and loss

1 Introduction

Understanding the evolutionary history of genes can offer insight into how new genes and functions arise in species [1], [2], [3], [4] and how gene losses shape gene families [5]. In phylogenetics, these histories are often understood by comparing two kinds of phylogenetic trees: the *species tree* that depicts the evolutionary relationship of a set a species, and the *gene tree* that depicts how a set of genes within these species have evolved. The gene tree can be thought of as evolving "inside" the species tree, and the goal of *reconciliation* methods is to infer this nesting.

Reconciliation methods rely on underlying evolutionary models in that topological incongruence between the gene and species tree must be accounted for using only the biological events allowed by the model. Among the most well-studied models are the *duplication-loss* (DL) model [6], [7], [8], [9], [10], [11], [12], [13], which allows for gene duplication and gene loss; the *duplication-transfer-loss* (DTL) model [14], [15], [16], [17], [18], [19], which considers horizontal gene transfers as well; and the *multispecies coalescent* (MSC) model [20], [21], [22], [23], which allows for incomplete lineage sorting (ILS) through deep coalescence.

For eukaryotic species, when a gene family evolves over sufficiently large evolutionary distances, its history can often be explained through the DL model alone. However, for smaller evolutionary distances or large population sizes, the MSC model must be taken into account. Several recent methods have considered reconciliations under a combined *duplication-loss-coalescence (DLC) model*, which allows for duplication, loss, *and* coalescence. For example, Rasmussen and Kellis [24] introduced a generative DLCoal model and associated algorithm DLCoalRecon for inferring the maximum *a posteriori* reconciliation. While DLCoalRecon was shown to improve over the duplication-loss model alone, it relies on a heuristic search and is highly parameterized, making it difficult to use in practice. Building on the DLCoal model, we previously introduced a new structure for representing reconciliations and an algorithm DLCpar for inferring a maximum parsimonious reconciliation (MPR) [25]. DLCpar achieves accuracy comparable to DLCoalRecon at reduced run time and with fewer parameters, making it more applicable to a broad range of species and large data sets.

However, adopting a parsimony approach presents its own set of challenges. For the DL model, assuming that loss events have a positive cost, the MPR is always unique [10], but for more general models, there may exist multiple MPRs for a given gene tree and species tree for a fixed assignment of event costs. For some insight, we can look to reconciliation under the DTL model. While probabilistic methods exist for DTL reconciliation [26], most formulations rely on a maximum parsimony framework [14], [15], [16], [17], [18], [19], [27], [28]. Under this model, the number of MPRs can grow exponentially with the size of the gene tree and the species tree [29], and consequently, efficient algorithms have been developed to summarize this space [29], [30], [31], [32], [33], [34].

- *H. Du, Y.S. Ong, M. Knittel, R. Mawhorter, N. Liu, R. Tojo, R. Libeskind-Hadas, and Y.-C. Wu are with the Department of Computer Science, Harvey Mudd College, Claremont, CA 91711, USA. E-mail: {hdu, yiong, mknittel, rmawhorter, ivliu, rtojo, hadas, yjw}@cs.hmc.edu.*
- *G. Gross is with the Department of Computer Science, University of Pennsylvania, Philadelphia, PA 19104, USA. E-mail: ggross@seas.upenn.edu.*

Manuscript received 12 Mar. 2019; accepted 12 May 2019. Date of publication 12 June 2019; date of current version 8 Dec. 2021.
(Corresponding author: Yi-Chieh Wu.)
Digital Object Identifier no. 10.1109/TCBB.2019.2922337

But the space of MPRs under the DLC model remains poorly understood. DLCpar returns only a single random MPR. That is, we lack information about the size of the MPR space, and furthermore, we do not know whether an inferred MPR is representative of this space, hindering downstream analyses. To address these shortcomings, we investigate the solution space of MPRs under the DLC model. Specifically, we have extended the DLCpar algorithm to (1) count the number of different, equally optimal reconciliations, and (2) sample the space of optimal reconciliations uniformly at random. Additionally, we show how to use these multiple samples to analyze the robustness of reconciliations and underlying events. These updates are part of the DLCpar software, which is freely available for download at https://www.cs.hmc.edu/~yjw/software/dlcpar.

We previously showed that the MPR problem for the DLC model is NP-complete and even hard to approximate (APX-complete), and it is therefore unlikely that polynomial-time algorithms or approximation schemes exist for this problem [35]. Thus, unsurprisingly, the DLCpar algorithm has worst-case exponential runtime. However, we prove that the reconciliation problem is fixed-parameter tractable by showing that the runtime of DLCpar (including the augmentations described above) is polynomial in the size of the species and gene tree when the number of genes that map to any given species is fixed.

To demonstrate the utility of our approach, we have applied our algorithm to a biological data set of 16 fungal species [36]. We show that while the majority of gene families have a unique optimal reconciliation, there exist families with millions of optimal reconciliations. But even in the presence of multiple optima, the underlying events are often well-supported, with these results holding across a variety of event cost settings.

To summarize, the contributions of this paper are significant extensions to the DLCpar algorithm and analysis that demonstrates DLCpar is efficient except when the two trees are extremely incongruent. By applying these extensions to a biological data set, we present new insights into both the size of MPR space and the support for underlying events in this space.

2 Background

We start by reviewing prior work on DLC reconciliations.

2.1 A Unified Model of Gene Family Evolution

While several DLC models exist, in this work, we rely on the DLCoal model developed by Rasmussen et al. [24].

To understand the interactions of duplications, losses, and coalescence in this model, we consider the gene family illustrated in Fig. 1A. In this example, a duplication occurs in one chromosome along the branch ancestral to species B and C, creating a new locus ("locus 2") in the genome distinct from the original locus ("locus 1"). At the new locus, this duplicate evolves within the population according to the Wright-Fisher process [21], [37], [38], [39], [40] until it eventually fixates. Thus, the sampled genomes of A, B, and C contain genes a_1, b_1, b_2, c_1, and c_2, and their phylogenetic tree is a "traceback" in the combined Wright-Fisher processes of loci 1 and 2. Note that all gene lineages for the duplicate (daughter) locus are forced to completely coalesce at the root of the locus 2 tree, allowing only one lineage to traceback into the locus 1 tree. Furthermore, the duplication creates an additional lineage within the locus 1 tree that must coalesce, creating another opportunity for deep coalescence. A similar process allows for gene loss (not shown). When a loss occurs, a single gene is deleted from one chromosome of the population, and this deletion drifts until it either fixes or goes extinct.

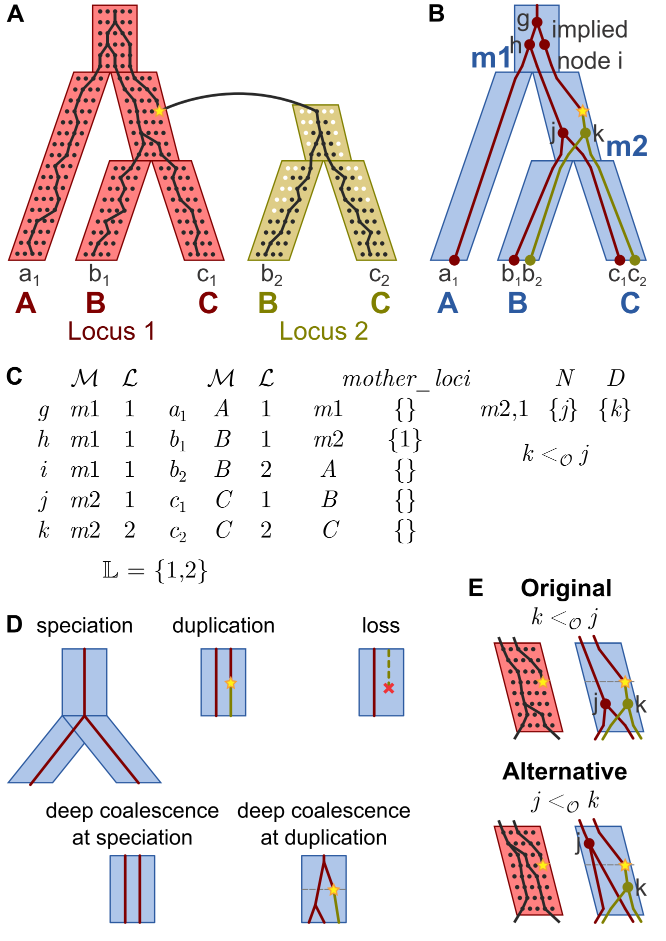

Fig. 1. *Gene family evolution and the labeled coalescent tree.* (*A*) The unified model DLCoal model combines the duplication-loss and multispecies coalescent models. In this example, a duplication occurs in one chromosome and creates a new locus, "locus 2," in the genome. At locus 2, the daughter duplicate (black dots) competes with the null allele (white dots) until it eventually fixates. A gene tree is a "traceback" in this combined process. Additionally, the red and yellow trees form an intermediate locus tree (distinct from the gene tree and species tree) that describes how loci are created and destroyed. (*B*) Evolution under the DLCoal model is represented using the labeled coalescent tree (LCT). (*C*) The LCT consists of four components: Species map $\mathcal{M}$, locus set $\mathbb{L}$, locus map $\mathcal{L}$, and partial order $\mathcal{O}$. Sets $mother\_loci(\cdot)$ of loci and $N(\cdot,\cdot)$ and $D(\cdot,\cdot)$ of nodes necessary for the partial order are also shown. (*D*) Evolutionary events are depicted in the LCT. Except for speciation, evolution within a single species tree branch is shown. (*E*) An alternative scenario is presented for evolution in species $m2$. The new partial order induces an extra lineage at the time of the duplication. [Figure and caption adapted with permission from Wu et al. [25] and Rasmussen and Kellis [24].]

For the example, notice that the red and yellow trees representing loci 1 and 2 form an intermediate *locus tree* that is distinct from the gene tree and species tree and describes how loci are created and destroyed. To disentangle the effects of duplication-loss and coalescence, we can think of the gene tree as evolving "inside" the locus tree, with multispecies coalescent processes within each locus, and we can think of the locus tree as evolving "inside" the species tree according to a duplication-loss process. As the gene tree of this model represents the history of gene sequences as they coalesce within the locus tree, we will use the term coalescent tree and gene tree interchangeably throughout the remainder of this manuscript.

2.2 DLC Reconciliation

Next, we review our previous work that formalized the concept of reconciliations and maximum parsimony reconciliations under the DLC model [25].

Throughout this work, the term *tree* refers to a rooted binary tree. Given a tree T, let $V(T)$ denote its node set and $E(T)$ denote its branch set. Let $L(T) \subset V(T)$ denote its leaf set, $I(T) = V(T) \setminus L(T)$ denote its set of internal nodes, and $r(T) \in I(T)$ denote its root node. For node $v \in V(T)$, let $c(v)$ denote its set of children, $p(v)$ denote its parent, and $e(v)$ denote the branch $(p(v), v)$. Define $\leq_T$ ($<_T$) to be the partial order on $V(T)$, where given two nodes u and v of T, $u \leq_T v$ ($u <_T v$) if and only if u is on the unique path between $r(T)$ and v (and $u \neq v$). The partial order $\geq_T$ ($>_T$) is defined analogously. In such a case, u is said to be a (strict) *ancestor* of v and v a (strict) *descendant* of u.

Let a *species tree* 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 tree, let a *leaf map* $Le : L(G) \rightarrow L(S)$ label each leaf of the gene tree with the leaf of the species tree from which the gene was sampled.

The *labeled coalescent tree* (Figs. 1B and 1C) formalizes the notion of a reconciliation in the DLC model.

Definition 2.1 (Labeled Coalescent Tree). *Given G, S, and Le, a* labeled coalescent tree (LCT) *for $\langle G, S, Le\rangle$ is a tuple $\langle \mathcal{M}, \mathbb{L}, \mathcal{L}, \mathcal{O}\rangle$, where*

- *$\mathcal{M} : V(G) \rightarrow V(S)$ is a* species map *that maps each node of G to a node of S.*
- *$\mathbb{L} \subset \mathbb{N}$ is a* locus set, *a finite set of natural numbers, each representing a locus that has evolved within the gene family.*
- *$\mathcal{L} : V(G) \rightarrow \mathbb{L}$ is a* locus map *that maps each node of G to a locus in $\mathbb{L}$.*
- *$\mathcal{O}$ is a* partial order *on $V(G)$ that represents the relative times of nodes. For each species node $s \in V(S)$, let $mother\_loci(s) \subset \mathbb{L}$ be the set of loci that yield a new locus in species s*

$$mother\_loci(s) = \{\, \mathcal{L}(g) \,|\, g \in I(G); \exists g' \in c(g), \mathcal{M}(g') = s, \mathcal{L}(g') \neq \mathcal{L}(g) \,\}.$$

Then for each species node $s \in V(S)$ and each locus $l \in mother\_loci(s)$, consider the set of gene nodes $O(s,l) = N(s,l) \cup D(s,l)$, where $N(s,l)$ contains "original" gene nodes that map to species s and locus l, descend from locus l, and have multiple children

$$N(s,l) = \{\, g \mid\ g \in V(G) \setminus \{r(G)\};\ \mathcal{M}(g) = s;\ \mathcal{L}(g) = l;\ \mathcal{L}(p(g)) = l;\ |c(g)| > 1 \,\},$$

and $D(s,l)$ contains "duplication" gene nodes that map to species s and not locus l but immediately descend from locus l

$$D(s,l) = \{\, g \mid\ g \in V(G) \setminus \{r(G)\};\ \mathcal{M}(g) = s;\ \mathcal{L}(g) \neq l;\ \mathcal{L}(p(g)) = l \,\}.$$

Note that the sets $N(s,l)$ and $D(s,l)$ are disjoint. Now consider a total order on $D(s,l)$; this order introduces $|D(s,l)| + 1$ bins in which each node in $N(s,l)$ may occur. The total order on $D(s,l)$ and the partition of $N(s,l)$ represent the relative times of duplication nodes as well as the relative times of original nodes with respect to duplication nodes. Define $<_{\mathcal{O}}$ to be the partial order on $O(s,l)$, where given two nodes $g, g' \in O(s,l)$, $g \neq g'$, then $g <_{\mathcal{O}} g'$ if and only if g precedes g' in time. Note that no order is induced on nodes of $N(s,l)$ in the same bin.

The LCT is 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 each $g' \in c(g)$, $\mathcal{M}(g) \leq_S \mathcal{M}(g')$.*
3) *For each $g, g' \in L(G)$, $g \neq g'$, if $\mathcal{M}(g) = \mathcal{M}(g')$, then $\mathcal{L}(g) \neq \mathcal{L}(g')$.*
4) *For each $l \in \mathbb{L}$, there exists a $g \in V(G)$ such that $\mathcal{L}(g) = l$.*
5) *For each $l \in \mathbb{L}$, there exists exactly one $g \in V(G)$ such that $L(g) = l$ and either $g = r(G)$ or $\mathcal{L}(p(g)) \neq l$.*
6) *For each $s \in V(S)$, each $l \in mother\_loci(s)$, and each $g, g' \in O(s,l)$, $g \neq g'$, if $g <_{\mathcal{O}} g'$, then $g \not\geq_G g'$.*

Constraint 1 asserts that $\mathcal{M}$ extends the leaf map Le. Constraint 2 asserts that $\mathcal{M}$ satisfies the temporal constraints implied by S. Constraint 3 asserts that extant genes (leaves) mapped to the same extant species (leaves) belong to different loci. Constraint 4 asserts that $\mathbb{L}$ includes only loci used by at least one gene. Constraint 5 asserts that every locus is created only once. Constraint 6 asserts that $\mathcal{O}$ satisfies the temporal constraints implied by G.

Because the locus set $\mathbb{L}$ is defined by the locus map $\mathcal{L}$, we often represent an LCT using the reduced tuple $\langle \mathcal{M}, \mathcal{L}, \mathcal{O}\rangle$.

An internal gene node $g \in I(G)$ is said to be a *speciation node* with respect to species map $\mathcal{M}$ if for each child $g' \in c(g), \mathcal{M}(g) \neq \mathcal{M}(g')$. Given a map $\mathcal{M}$, some nodes may initially be hidden in a gene tree due to losses and deep coalescence. Such "implied speciation nodes" are added to each gene branch that spans multiple branches of the species tree (Supplemental Section S1.1, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TCBB.2019.2922337). Note that the species map $\mathcal{M}$ is defined first, then implied speciation nodes are added as required, and finally the locus map $\mathcal{L}$ and partial order $\mathcal{O}$ are defined on the nodes of the gene tree, which now includes any implied speciation nodes.

Next, we define some useful sets. Given a species node $s \in V(S)$ and a species map $\mathcal{M}$, let $nodes(s)$ denote the set of gene nodes mapped to s; $bottoms(s)$ denote the set of speciation nodes mapped to s; and $tops(s) = bottoms(p(s))$ if $s \neq r(S)$ and $tops(s) = \{r(G)\}$ otherwise. (We can think of $bottoms(s)$ and $tops(s)$ as the set of gene nodes at the "bottom" or "top" of species branch $e(s)$, respectively.)

The LCT allows for several evolutionary events (Fig. 1D). A *speciation* event corresponds to a locus present at the bottom of a species branch continuing at the same locus in at least one child species. As a speciation in the LCT reflects a speciation in the species tree, it is considered a null event. A *duplication* event corresponds to the creation of a new locus along a gene branch; such a gene branch is said to have a duplication. A *loss* event corresponds to a locus present at either the top of a species branch, or created via a duplication within the species branch, being no longer present at the bottom of the species branch. A *coalescence* event is, in fact, a *deep coalescence* or *failure to coalesce*, which results in "extra" branches (lineages) in a species and locus. Two gene lineages may fail to coalesce at speciations or duplications, resulting in extra lineages at the speciation or duplication, respectively. Note that the speciation, duplication, loss, and coalescence at speciation events depend only on $\mathcal{M}$ and $\mathcal{L}$ while coalescence at duplication events also depend on $\mathcal{O}$ (Fig. 1E). Formal definitions are provided in Supplemental Section S1.2, available online.

Let C_D, C_L, C_C, and C_K denote the positive real-number costs associated with duplication, loss, and coalescence at speciations and duplications, respectively. The cost of reconciling G and S according to LCT $\langle \mathcal{M}, \mathcal{L}, \mathcal{O} \rangle$ is defined as follows:

Definition 2.2 (Reconciliation Cost). *Given G, S, Le, C_D, C_L, C_C, and C_K, the* reconciliation cost *of an LCT $\langle \mathcal{M}, \mathcal{L}, \mathcal{O} \rangle$ for $\langle G, S, Le \rangle$ with d duplication events, ℓ loss events, c coalescence at speciation events, and k coalescence at duplication events is $\mathcal{R}_{\langle \mathcal{M}, \mathcal{L}, \mathcal{O} \rangle} = d \cdot C_D + \ell \cdot C_L + c \cdot C_C + k \cdot C_K$.*

Our goal is to find a most parsimonious reconciliation. Formally:

Problem 2.1 (Most Parsimonious Reconciliation (MPR) Problem). *Given G, S, Le, C_D, C_L, C_C, and C_K, find an LCT for $\langle G, S, Le \rangle$ with minimum reconciliation cost.*

Note that the solution to Problem 2.1 is not necessarily unique.

Next, we define optimality of LCT components.

Definition 2.3 (Optimal LCT Components). *A species map $\mathcal{M}^*$ is said to be optimal if there exists a locus map $\mathcal{L}$ and a partial order $\mathcal{O}$ such that $\langle \mathcal{M}^*, \mathcal{L}, \mathcal{O} \rangle$ solves the MPR problem. Given a species map $\mathcal{M}$, a locus map $\mathcal{L}^*$ is said to be optimal if there exists a partial order $\mathcal{O}$ such that $\langle \mathcal{M}, \mathcal{L}^*, \mathcal{O} \rangle$ solves the MPR problem. Given a species map $\mathcal{M}$ and locus map $\mathcal{L}$, a partial order $\mathcal{O}^*$ is said to be optimal if $\langle \mathcal{M}, \mathcal{L}, \mathcal{O}^* \rangle$ solves the MPR problem.*

Note that neither the given species map nor locus map need be optimal. Henceforth, an MPR refers to an LCT that solves the MPR problem. An MPR must satisfy certain properties.

Theorem 2.1 (Optimal Species Maps). *The species map $\mathcal{M}^*$ is optimal if and only if $\mathcal{M}^*$ is the lowest common ancestor (LCA) map.*

Theorem 2.2 (Optimal Locus Maps). *Given a species map $\mathcal{M}$, if the locus map $\mathcal{L}^*$ is optimal, then*[1]*:*

- *Each gene branch $e(g) \in E(G)$ has at most one duplication.*[2]
- *For each species node $s \in V(S)$ and each gene node $g \in nodes(s) \setminus bottoms(s)$ internal to the species branch, if g' and g'' denote the children of g, then at most one of the two children branches $e(g')$ or $e(g'')$ has a duplication.*

Theorem 2.3 (Optimal Partial Orders). *Given a species map $\mathcal{M}$ and locus map $\mathcal{L}$, if the partial order $\mathcal{O}^*$ is optimal, then for each species $s \in V(S)$ and each locus $l \in mother\_loci(s)$, duplications are placed as early in the species branch as possible. That is, for each original node $g \in N(s,l)$ and each duplication node $d \in D(s,l)$, $g <_{\mathcal{O}^*} d$ if and only if $g \leq_G d$.*

Proofs are provided in Supplemental Section S2, available online.[3]

2.3 DLCpar Algorithm

We now outline the basic steps of the DLCpar algorithm (Fig. 2, [25]) for solving the MPR problem. The formal pseudo-code is provided in Supplemental Section S3, available online.

From Theorem 2.1, DLCpar sets $\mathcal{M}^*$ to be the LCA map, then uses this map to decompose the gene tree into disjoint subtrees that evolve within each species branch (Fig. 2A).

For each species node $s \in V(S)$, let a *sub-locus map* and *sub-partial order* be a locus map and partial order defined over gene nodes in the species branch $e(s)$, that is, over $g \in tops(s) \cup nodes(s)$, and let a *tile* consist of a particular sub-locus map and sub-partial order with associated reconciliation cost. To determine an optimal locus map and partial order, DLCpar constructs a set of tiles for each species, then uses dynamic programming to combine tiles so that loci of nodes shared across species match. In the remainder of this section, we provide more details on this process.

For each species via pre-order traversal of the species tree, DLCpar constructs a set of tiles by first considering all valid sub-locus maps that satisfy Theorem 2.2. As an example, in the root species, which contains a (single) subtree of the gene tree, DLCpar assigns the root of the subtree to an arbitrary locus, then considers all possible placements of duplications along branches of the subtree, subject to the aforementioned constraints (Fig. 2B). Each combination of duplication placements yields a sub-locus map. For each sub-locus map, DLCpar considers all valid sub-partial orders that satisfy Theorem 2.3, then chooses one with minimum number of coalescence at duplication events (as other event types do not depend on the sub-partial order). For the tile consisting of the sub-locus map and chosen sub-partial order, DLCpar finds the set of events within the tile and computes the reconciliation cost.

[1] Previously, this theorem also stated that "The number of loci is at most one more than the minimum number of inferred duplications under the duplication-loss model using the same duplication and loss cost." The accompanying proof was flawed, so this property is no longer included.

[2] This constraint follows from the definition of the LCT and duplications in the LCT.

[3] Previously, these theorems were poorly worded in that it was unclear if the conditions were necessary or sufficient. Furthermore, the proofs showed that *there exists an* optimal component that satisfies these properties. In this work, the proofs have been extended to show that *every* optimal component satisfies these properties. This modification guarantees that the DLCpar algorithm does not ignore any potentially optimal species maps, locus maps, or partial orders, and thus, is not under-counting the number of optimal reconciliations.

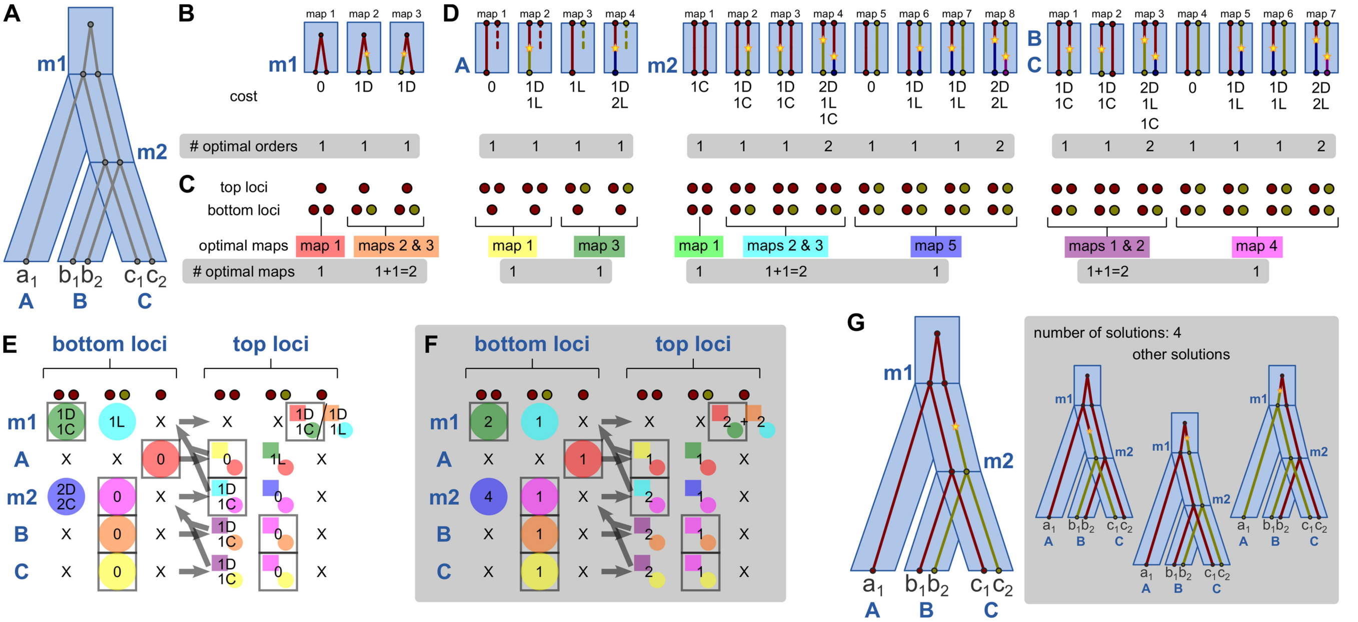

Fig. 2. *The DLCpar algorithm.* See text for an explanation of the algorithm. (*A*) The optimal species map. (*B*) Tiles for the root species, each consisting of a sub-locus map with an optimal sub-partial order and reconciliation cost. (In this example, event counts rather than the reconciliation cost is shown, and each event has equal cost.) Each sub-locus map may have multiple optimal sub-partial orders. (*C*) Top and bottom loci for each sub-locus map, and for each unique pair of top and bottom loci, the optimal underlying sub-locus maps. (*D*) Tiles for the remaining species via pre-order traversal of the species tree. (*E*) The dynamic programming table for assigning optimal top and bottom loci for each species. The table is filled via post-order traversal of the species tree (arrows), and each cell contains the minimum total cost along all descendant species branches. For top loci, colors indicate which bottom loci (circles) and which sub-locus map (squares with colors corresponding to parts *C* and *D*) are used. At the species root, there is only only possible assignment of top loci, and traceback allows assignment of top and bottom loci for all species (boxed). These loci assignments are used to determine optimal underlying sub-locus maps and sub-partial orders. (*F*) The number of equally optimal reconciliations for each assignment of top and bottom loci. (*G*) A most parsimonious reconciliation (sampled uniformly at random), along with the number of equally optimal reconciliations. [Figure and caption adapted with permission from Wu et al. [25]. Gray boxes indicate new content.]

Next, DLCpar considers the problem of propagating locus assignments across species. For each sub-locus map, DLCpar computes *top loci* and *bottom loci*, which are compact representations of the locus assignments at $tops(s)$ and $bottoms(s)$. To construct these representations, the algorithm arbitrarily (but consistently) orders $tops(s)$ (or $bottoms(s)$), assigns the first node to an arbitrary "locus 1", then assigns each subsequent node either to one of the previous loci, if the node is mapped to the same locus as a previous node, or to the next available locus. The *relative locus pair* for a sub-locus map is a tuple (l_t, l_b) with top loci l_t and bottom loci l_b. DLCpar computes the relative locus pair for each sub-locus map, then, for each (l_t, l_b), records an underlying sub-locus map (that induces (l_t, l_b)) with minimum reconciliation cost, denoted as $C^s(l_t, l_b)$ (Fig. 2C). Note that by traversing the species tree in pre-order, DLCpar ensures that the set of top loci for any non-root species is determined by the set of bottom loci of its parent species, and the set of bottom loci for any species is in turn determined by the sets of top loci and enumerated sub-locus maps for the species (Fig. 2D).

Once all tiles are constructed for all species, DLCpar uses dynamic programming to determine an optimal assignment of top loci and bottom loci for each species (Fig. 2E). The algorithm constructs two tables, F^b and F^t, where the entries $F^b(s, l)$ and $F^t(s, l)$ are the minimum costs for assigning bottom loci l to $bottoms(s)$ or top loci l to $tops(s)$, respectively, and these costs include events along all descendant species branches. These tables are completed via post-order traversal of the species tree, and for each species, F^b then F^t is filled. To compute $F^b(s, l)$, there are two cases to consider. If $s \in L(S)$, then when constructing tiles, DLCpar has already required that bottom loci for extant species be distinct; therefore, the only possible assignment of bottom loci is valid. Otherwise, assigning bottom loci to s requires assigning top loci to children species s' and s''

$$F^b(s, l) = \begin{cases} 0, & \text{if } s \in L(S) \\ F^t(s', l) + F^t(s'', l), & \text{otherwise} \end{cases}.$$

To compute $F^t(s, l)$, DLCpar must combine a bottom loci with a relative locus pair that has the same bottom loci, then choose a bottom loci with minimum cost

$$F^t(s, l) = \min_{l_b : (l, l_b) \in \mathbf{RLP}(s)} \{F^b(s, l_b) + C^s(l, l_b)\},$$

where $\mathbf{RLP}(s)$ denotes the set of relative locus pairs for species s. Once the species root is reached, since $tops(r(S)) = \{r(G)\}$, there is only one possible assignment of top loci. By using standard dynamic programming "bookkeeping", DLCpar then traces back through these tables via a pre-order traversal of the species tree to assign optimal top and bottom loci for each species.

Finally, for each species, DLCpar looks up the optimal sub-locus map for the chosen relative locus pair, and looks up the optimal sub-partial order for the chosen sub-locus map. These components, together with the the optimal species map, constitute a most parsimonious reconciliation (Fig. 2G).

3 Multiple Optimal Reconciliations

In this section, we show that the problems of computing a single optimal reconciliation, counting the number of optimal reconciliations, and sampling from the set of optimal reconciliations uniformly at random are fixed-parameter tractable by extending the DLCpar algorithm and analyzing its running time.

3.1 Computing the Number of Optimal Reconciliations

Before we turn to the problem of counting optimal reconciliations, we introduce a corollary of Theorem 2.3. Given a species map $\mathcal{M}$ and locus map $\mathcal{L}$, for each species node $s \in V(S)$ and each locus $l \in mother\_loci(s)$, let a *local order* be a partial order over $O(s,l)$ and a *duplication order* be a total order over $D(s,l)$.[4] A local order (duplication order) is said to be optimal if it induces the minimum number of coalescence at duplication events (as again, other event types do not depend on the partial order).

Corollary (Number of Optimal Partial Orders for a Single Locus). Given a species map $\mathcal{M}$ and locus map $\mathcal{L}$, for each species node $s \in V(S)$ and each locus $l \in mother\_loci(s)$, the number of optimal local orders is equal to the number of optimal duplication orders. □

The proof is provided in Supplemental Section S2, available online. As before, neither the given species map nor locus map need be optimal.

We now describe how to count optimal reconciliations. In the DLCpar algorithm, there are three places where we might choose from multiple optimal options and thus need to keep track of the number of solutions:

C1. For each sub-locus map for a species, there may exist multiple optimal sub-partial orders.

C2. For each relative locus pair for a species, there may exist multiple optimal sub-locus maps.

C3. When using dynamic programming to determine an optimal assignment of top and bottom loci for each species, there may exist multiple optimal assignments and multiple optimal paths.

Next, we describe how to count each of these sources of multiplicity.

First, for each sub-locus map for a species, we consider each locus $l \in mother\_loci(s)$ separately. Via the above corollary, for this locus, the number of optimal local orders is equal to the number of optimal duplication orders. When considering partial orders, DLCpar constructs all sets of duplication orders, so it is easy to count the subset that is optimal. Then, as each locus evolves independently, the number of optimal sub-partial orders for the sub-locus map is the product of the number of duplication orders for each locus (Figs. 2B and 2D, gray highlight). To keep track of these counts, for a species s and a set $\mathbf{L}(s)$ of locus maps for that species, let $\mathcal{N}^{s,\mathcal{O}} : \mathbf{L}(s) \rightarrow \mathbb{N}^+$ map each locus map to the number of optimal partial orders for that locus map.

Second, when DLCpar propagates locus assignments across species, it computes a relative locus pair for each locus map. For each relative locus pair (l_t, l_b), it is therefore straightforward to count the subset $\mathcal{X}$ of underlying sub-locus maps with minimum reconciliation cost. Then, because each of these underlying sub-locus maps could have multiple optimal sub-partial orders, we sum the number of optimal sub-partial orders for each sub-locus map in $\mathcal{X}$ (Figs. 2C and 2D, gray highlight). Formally, for a species s and a set $\mathbf{RLP}(s)$ of relative locus pairs for that species, let $\mathcal{N}^{s,\mathcal{L}} : \mathbf{RLP}(s) \rightarrow \mathbb{N}^+$ map each relative locus pair to the number of optimal reconciliations (consisting of a sub-locus map and sub-partial order) for that relative locus pair. Then

$$\mathcal{N}^{s,\mathcal{L}}(l_t, l_b) = \sum_{\hat{\mathcal{L}} \in \mathcal{X}} \mathcal{N}^{s,\mathcal{O}}(\hat{\mathcal{L}}).$$

Third, we must account for multiple optimal assignments and paths during the dynamic programming step (Fig. 2F). During this step, DLCpar now constructs two additional tables, N^b and N^t that are analogous to F^b and F^t. That is, the entries $N^b(s,l)$ and $N^t(s,l)$ track the number of optimal reconciliations that assign bottom loci b to $bottoms(s)$ or top loci l to $tops(s)$, respectively, where again, the reconciliations include events along all descendant species branches. To compute $N^b(s,l)$, there are again two cases to consider. If $s \in L(S)$, then there is only one valid assignment of bottom loci. Otherwise, DLCpar can use any sub-solution that assigns l as top loci of one child species s' and any sub-solution that assigns l as top loci of the other child species s''

$$N^b(s,l) = \begin{cases} 1, & \text{if } s \in L(G) \\ N^t(s',l) \times N^t(s,'',l), & \text{otherwise} \end{cases}.$$

To compute $N^t(s,l)$, recall that DLCpar combines a bottom loci with a relative locus pair that has the same bottom loci. So for a single bottom loci, we multiply the corresponding counts. We must then sum over the set $\mathcal{Y}$ of bottom loci with minimum cost

$$N^t(s,l) = \sum_{\hat{l}_b \in \mathcal{Y}} \left\{ N^b(s,\hat{l}_b) \times \mathcal{N}^{s,\mathcal{L}}(l,\hat{l}_b) \right\}.$$

Finally, at the species root, there is only one possible assignment l of top loci. The number of optimal reconciliations is, therefore, $N^t(r(S),l)$.

3.2 Sampling Optimal Reconciliations Uniformly at Random

Now that we have a process for counting the number of equally optimal reconciliations, we turn to the problem of uniform sampling among the multiple optima. Our method for uniform sampling parallels our method for counting optima. In particular, we consider each point in the algorithm where we choose from multiple optimal options. Instead of using random sampling, we now consider a weighted sampling of these choices:

S1. For each sub-locus map for a species, there may exist multiple optimal sub-partial orders. For each locus, each local order is defined by its duplication order, and there is no choice for the node partition.

[4] The terms *local partial order* and *local duplication order* are more precise, but for simplicity, we will understand that a local order is partial and a duplication order is local.

Therefore, we first use uniform weights to select from the set of optimal local orders for each locus. Then, as each locus evolves independently, we combine the selected local orders for each locus to arrive at an optimal sub-partial order for the sub-locus map.

S2. For each relative locus pair for a species, there may exist multiple optimal sub-locus maps. We weight each sub-locus map according to the number of associated optimal sub-partial orders. Formally, for a relative locus pair (l_t, l_b), a locus map $\hat{\mathcal{L}}$ that induces (l_t, l_b) is sampled with probability $\frac{\mathcal{N}^{s,\mathcal{O}}(\hat{\mathcal{L}})}{\mathcal{N}^{s,\mathcal{L}}(l_t,l_b)}$.

S3. When using dynamic programming to determine an optimal assignment of top and bottom loci for each species, there may exist multiple optimal assignments and multiple optimal paths. To be precise, whereas counting the number of optimal reconciliations resulted in additional dynamic programming tables, sampling a reconciliation results in changes during traceback through the tables. When assigning optimal bottom loci, the bottom loci are either known for extant species or set to the top loci of children species. Thus, there is no selection to be made. However, when assigning top loci, there may exist multiple optimal bottom loci. The weights for each bottom loci must account for the number of solutions for assigning bottom loci and the number of solutions for the locus map. Formally, for top loci l, a bottom loci $\hat{l}_b$ is sampled with probability $\frac{N^b(s,\hat{l}_b)\times\mathcal{N}^{s,\mathcal{L}}(l,\hat{l}_b)}{N^t(s,l)}$.

3.3 Correctness

The proof of correctness of the DLCpar algorithm, including the augmentations described above, is straightforward. By Theorem 2.1, the species map is optimal. Then, when constructing tiles and propagating locus assignments using relative locus pairs, DLCpar enumerates all possible sub-locus maps and sub-partial orders. For the augmentations, the number of solutions associated with each sub-locus map or each relative locus pair must be exactly the sum over sub-partial orders and sub-locus maps, respectively. Additionally, each sub-partial order or sub-locus map is sampled at random based on its probability mass. Likewise, when determining an optimal assignment of top loci and bottom loci for each species, DLCpar calculates the minimum cost of each sub-solution, combines sub-solutions over disjoint parts of the subtree, and samples each sub-solution with probability equal to its probability mass.

3.4 Time Complexity

Let m denote the number of leaves in the species tree, n denote the number of leaves in the gene tree, and c denote the maximum number of speciation nodes at any species branch. In this section, we show that the MPR problem is fixed-parameter tractable by showing that the running time of DLCpar is $\mathbf{O}(m(f(c)+n))$ for some function f that depends only on c.[5] The following analysis uses loose upper-bounds for $f(c)$; the value of $f(c)$ can be improved with more detailed analysis.

Lemma 3.1. *Given a species $s \in V(S)$ and a sub-locus map and sub-partial order, the reconciliation cost for the species branch $e(s)$ can be computed in time $\mathbf{O}(c^2)$.*

Proof. First, note that the subtrees of the gene tree that exist within $e(s)$ form a forest $\mathcal{F}$ that contains at most c roots and c leaves. Thus, $\mathcal{F}$ contains $\mathbf{O}(c)$ additional gene tree nodes and $\mathbf{O}(c)$ gene tree branches.

Duplications can be counted in $\mathbf{O}(c)$ time by simply traversing $\mathcal{F}$. Losses can be counted in $\mathbf{O}(c)$ time by first traversing $\mathcal{F}$ to collect the starting nodes of each locus in $e(s)$. Then, from the set of starting nodes of each locus, the gene tree subgraph is traversed downwards to determine if there is a path to a bottom node $g \in bottoms(s)$ that does not pass through a duplication. If there is no such path, that locus is lost.

Coalescences at speciation can be counted in $\mathbf{O}(c)$ time by counting the number of top nodes $g \in tops(s)$ that are on the same relative locus. For coalescences at duplication, $\mathcal{F}$ is traversed in $\mathbf{O}(c)$ time to construct the sets $O(s,l)$ and the set of starting nodes of each locus. Then, for each locus $l \in mother\_loci(s)$ and each duplication node $d \in D(s,l) \subseteq O(s,l)$, the number of branches contemporaneous with d is counted by processing $O(s,l)$ in the order specified by the sub-partial order. Since there can be $\mathbf{O}(c)$ duplications across all mother loci, and each scan over $O(s,l)$ takes $\mathbf{O}(c)$ time, the total cost is $\mathbf{O}(c^2)$. □

Theorem 3.2. *The worst-case running time of the DLCpar algorithm is $\mathbf{O}(m(f(c)+n))$ where $f(c) = B_c 2^{2c}(2c)!c^2$ and B_c denotes the cth Bell number.*

Proof. We give an upper-bound on the running time of DLCpar by considering the separate parts of the algorithm. First, the LCA mapping between the gene tree and species tree can be computed in $\mathbf{O}(mn)$ time [9].

Next, DLCpar constructs a set of tiles for each species, which consists of a sub-locus map with an associated optimal sub-partial order and reconciliation cost. For a species s, since there are at most c nodes in $tops(s)$, the number of distinct top loci is bounded by B_c, the cth Bell number. Because there are at most c nodes in $bottoms(s)$, the subtrees of the gene tree that exist within $e(s)$ form a forest with at most c leaves, resulting in at most $c-1$ internal nodes and at most $2c$ branches within $e(s)$. In an MPR, at most one duplication can be placed on each gene branch and an optimal partial order places duplications as early as possible, resulting in at most 2^{2c} distinct duplication placements and at most $(2c)!$ distinct orderings. Thus, the total number of sub-locus maps with associated sub-partial orders is bounded by $\mathbf{O}(B_c 2^{2c}(2c)!)$. Since permutations and partitions can be generated in-place in amortized constant time [41], [42], and each sub-locus map with associated sub-partial order has size $\mathbf{O}(c)$, each sub-locus map and sub-partial order can be explicitly enumerated in amortized time $\mathbf{O}(c)$, and then, by Lemma 3.1, the reconciliation cost can be computed in time $\mathbf{O}(c^2)$. Thus, this step of the algorithm is bounded by time $\mathbf{O}(B_c 2^{2c}(2c)!c^2)$.

Next, DLCpar computes the relative locus pair and reconciliation cost for each sub-locus map. Since there are at most c nodes in $tops(s)$ and c nodes in $bottoms(s)$, the relative locus pair for a locus map can be computed in $\mathbf{O}(c)$

[5] Big O is denoted using bold-face $\mathbf{O}$ to differentiate it from O in the set $O(s,l)$.

time, but this time is subsumed by the time to compute the reconciliation cost. Then, the algorithm constructs tables C^s and $\mathcal{N}^{s,\mathcal{L}}$ to map each relative locus pair to its optimal reconciliation cost and number of sub-locus maps with that cost. These $B_c \times B_c$ tables are filled by scanning the list of sub-locus maps with associated partial orders and thus takes time $\mathbf{O}(B_c 2^{2c}(2c)!)$. Note that the $\mathbf{O}({B_c}^2)$ time to initialize the table is subsumed by the time to scan the list since $B_c \in \mathbf{O}(c!)$. Thus, the cost of enumerating sub-locus maps and sub-partial orders, computing their reconciliation costs, and storing them in tables takes time $\mathbf{O}(B_c 2^{2c}(2c)!c^2)$, and repeating this process for each of m species takes time $\mathbf{O}(m(B_c 2^{2c}(2c)!c^2))$.

Each dynamic programming table tracks the assignment of top and bottom loci for each species and thus has dimensions $m \times B_c$. For each entry $F^b(s,l)$ or $N^b(s,l)$, the value is either known in the base case ($s \in L(G)$) or uses the values (cost or number of solutions) from assigning l as top loci for the two children branches. For each entry $F^t(s,l)$ or $N^t(s,l)$, the algorithm considers each of the at most B_c bottom loci assignments and, for each such assignment, looks up the cost or number of solutions in other tables. Thus, each entry can be computed in time $\mathbf{O}(B_c)$. Altogether, the running time of the dynamic programming step is bounded by $\mathbf{O}(m{B_c}^2)$, which is subsumed by the previous $\mathbf{O}(m(B_c 2^{2c}(2c)!c^2))$ term.

Putting these components together, the total running time of DLCpar is $\mathbf{O}(m(B_c 2^{2c}(2c)!c^2 + n))$. □

This theorem implies that the MPR problem is fixed-parameter tractable, where the parameter, c, is the maximum number of speciation nodes at any species branch in the LCA mapping. While $f(c)$ grows exponentially with c, the value of c is induced by the LCA mapping, with $c = 1$ if the two trees are congruent, and $c = n$ in the worst case (when the entire gene tree is mapped within a single species). In general, c is small for relatively congruent trees and large for relatively incongruent trees.

4 Results

To investigate the solution space of DLC reconciliations, we used a biological data set of 5,351 gene families across 16 fungal genomes [43] that has been used to evaluate numerous phylogenetic algorithms [13], [24], [25], [36]. All gene families contain at least four genes; thus, multiple gene trees and reconciliations can be inferred for each family. We reconstructed gene trees using TreeFix [44] then ran DLCpar with the default event costs (duplication and loss cost of 1, coalescence cost of 0.5). For each gene family, we determined the number of optimal reconciliations, and for gene families with multiple optima, we also uniformly sampled 100 optimal reconciliations. Some gene families were very large or highly incongruent to the species tree and thus not able to be reconciled (0.2 percent of gene families are omitted from our analysis).

4.1 Number of Optimal Reconciliations

The majority (66.9 percent) of gene families have a unique optimal reconciliation. This large percentage can be attributed to three factors. One, 24.5 percent of all gene trees are congruent to the species tree, and so there exists a single unique optimal reconciliation that requires no events. Two, an additional 32.3 percent of gene families have at most one gene per species. Their corresponding reconciliations require no duplications, and without duplications, only one locus map (with a trivial partial order) is optimal. Three, the remaining families with a single reconciliation occur when the gene tree is reconciled using one duplication. With one duplication, there can exist two optimal locus maps (each with one optimal partial order) that differ only in that lineages labeled with the mother locus and lineages with the daughter locus are interchanged. However, for these gene trees, only one of these locus maps is valid due to the requirement of complete coalescence within the daughter locus; interchanging the mother and daughter lineages would result in incomplete coalescence within the daughter locus.

Despite the prevalence of families with a unique optimal reconciliation, many gene families have multiple optimal reconciliations. We found that 33.1 percent of families have multiple optima, with 4.8 percent (1.9 percent) of families with more than 10 (100) optima and one family with more than 7.9 million optima (Fig. 3A). Furthermore, the number of optimal reconciliations tends to increase exponentially with gene tree size (Fig. 3B), making it impractical to enumerate all optimal reconciliations for larger datasets.

We also observed that when a gene family has multiple optimal reconciliations, the number of optima tends to be a power of two. This property is true for 98.2 percent of gene families with multiple optima. Again, recall that each duplication in a species branch can yield two optimal locus maps that differ only in the lineages labeled with the mother and daughter locus. When there is complete coalescence of lineages within both the mother and daughter locus, either labeling is allowed. Thus, with d such duplications, there exists 2^d distinct optimal locus maps with associated partial orders. (In contrast, as discussed earlier, when there is incomplete coalescence of lineages within the mother locus, then interchanging the mother and daughter lineages is invalid, resulting in less than 2^d optima.)

Interestingly, our results suggest that the space of MPRs under the DLC model can both differ from and be similar to the space of MPRs under the DTL model. A case study of 4,735 gene trees and 100 (predominantly prokaryotic) species from the Tree of Life [15] found that only 17 percent of the gene trees have a unique optimal reconciliation and more than 50 percent have more than 100 optima, but similarly, the number of optima increases exponentially with gene tree size [29]. Some of the observed differences can likely be attributed to the DTL study using a larger data set. Whereas our study considered 16 species, with median and mean leaf set sizes for gene trees of 16 and 15.4, the DTL study considered 100 species, with median and mean leaf set sizes of 18 and 35.1. However, the similarity in median gene tree size suggests that the space of MPRs under the DLC model may be smaller, and thus a single MPR more representative, even for data sets of similar size.

4.2 Event Support Across Multiple Samples

Despite the presence of multiple optimal reconciliations, it may be that the reconciliations are similar in the sense that the underlying locus tree topology and events are largely the same. For gene families with multiple optimal reconciliations,

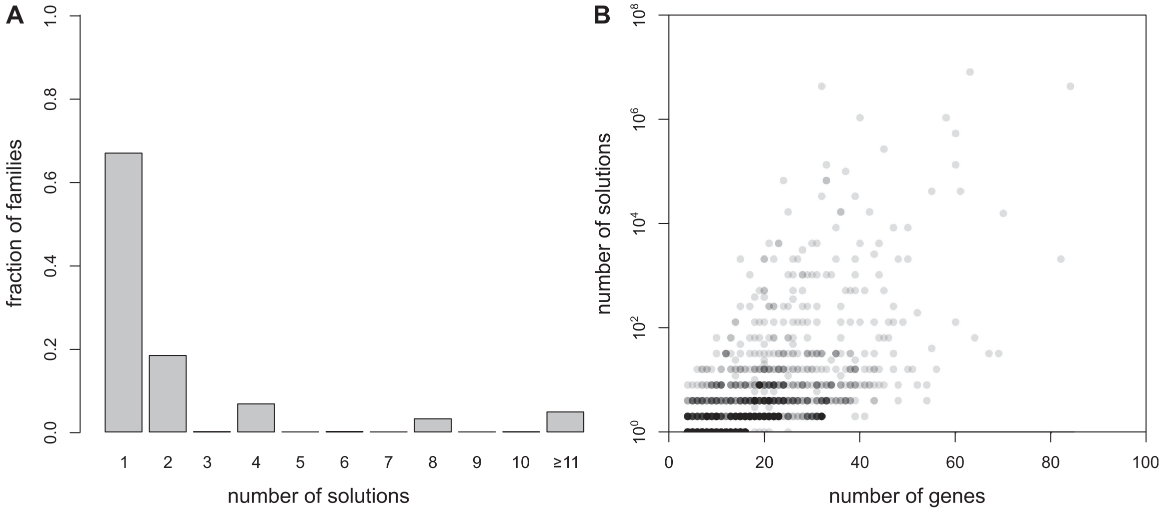

Fig. 3. *Number of optimal reconciliations.* (A) The distribution of the number of optimal reconciliations across all gene families, and (B) the number of optimal reconciliations as the number of genes per family varies.

we addressed this question of similarity by sampling 100 optimal reconciliations for each gene tree, extracting the locus tree topology and events for each sampled reconciliation, and computing the support of the locus tree topology and events for a plurality reconciliation.

We found that the locus tree topology and events are well-supported. 91.8 percent of gene families have a locus tree topology that is fully supported across the 100 samples. Additionally, 99.0 percent of locus tree branches are fully supported, and the average support across all locus tree branches is 99.4 percent. For the 29,551 speciations, 4,053 duplications, and 2,266 losses, 99.4, 97.4, and 97.2 percent of events are fully supported, with average supports of 99.7, 98.7, and 98.6 percent, respectively. (We did not compute support for coalescences because, in most applications, deep coalescences are "nuisance" events that are irrelevant to the user.) These results imply that the locus tree and events, and consequently orthologs and paralogs, inferred using DLCpar are likely to be representative of MPR space. Interestingly, similar results, though with weaker support values, were found for MPRs under the Duplication-Transfer-Loss model [29], [30] despite the DLC and DTL models using different underlying events.

We found that, surprisingly, support for locus tree branches and events increases with increasing number of optimal reconciliations (Fig. 4). This result suggests that the number of MPRs cannot adequately measure the variability within MPR space. That is, a gene family may have many MPRs that mostly share the same events, in which case a single plurality reconciliation may be enough to summarize the events in MPR space. Or a gene family may have few MPRs that differ substantially from one another, in which case it may be necessary to enumerate or sample multiple solutions.

4.3 Varying Event Costs

A limitation of parsimonious reconciliation approaches is the need for the user to explicitly set costs for each event. We studied the effect of using different costs on the MPR space and found that our results are robust to the cost setting (Table 1). The most substantial deviation occurs when all events have equal cost, which yields lower locus tree and event support. We hypothesize that with equal costs, events are more "fungible" in the sense that a group of events can be swapped with another (equally-sized) group of events.

4.4 Runtime

The average (median) runtime for a gene family was 1.89 (0.08) sec to count the number of optima and, for gene families with multiple optima, 1.47 (0.10) sec to sample 100 reconciliations.[6] As expected, runtime increases with number of genes and number of speciation nodes (Fig. 5).

5 Comparison with Other Models

In this work, we have used the DLCoal model for gene family evolution; however, other models exist. Vernot et al. [45] proposed a model for reconciling gene trees with non-binary species trees under a duplication-loss parsimony framework while allowing ILS (due to deep coalescence) at non-binary nodes in the species tree. Stolzer et al. [19] later extended this model and parsimony method to allow for transfers as well. Their algorithms are fixed-parameter tractable when the size of the largest polytomy in the species tree is fixed. More recently, Chan et al. [46] proposed a model for reconciling gene trees with binary species trees that allows for duplications, transfers, losses, and ILS but also penalizes the degree of ILS (e.g., the number of extra lineages) as well as ensuring a time-consistent solution (i.e., in which transfers do not induce contradictory constraints on the relative order of the internal nodes). Their algorithm for inferring a most parsimonious reconciliation marks certain internal branches that can contain ILS, then connects sets of marked branches into ILS subtrees. Its complexity was also shown to be fixed-parameter tractable when the size of the largest ILS subtree is fixed.

A detailed comparison of these models is provided in Chan et al. [46], which notes that the model of ILS is often the key difference, and, because each algorithm solves their own model, direct comparisons may be less informative.

[6] Experiments were performed on 64-core cluster consisting of four AMD Opteron 6276 CPUs, each with 16 cores at 2.3 GHz, and a total of 512 GB of DDR3-1600 RAM.

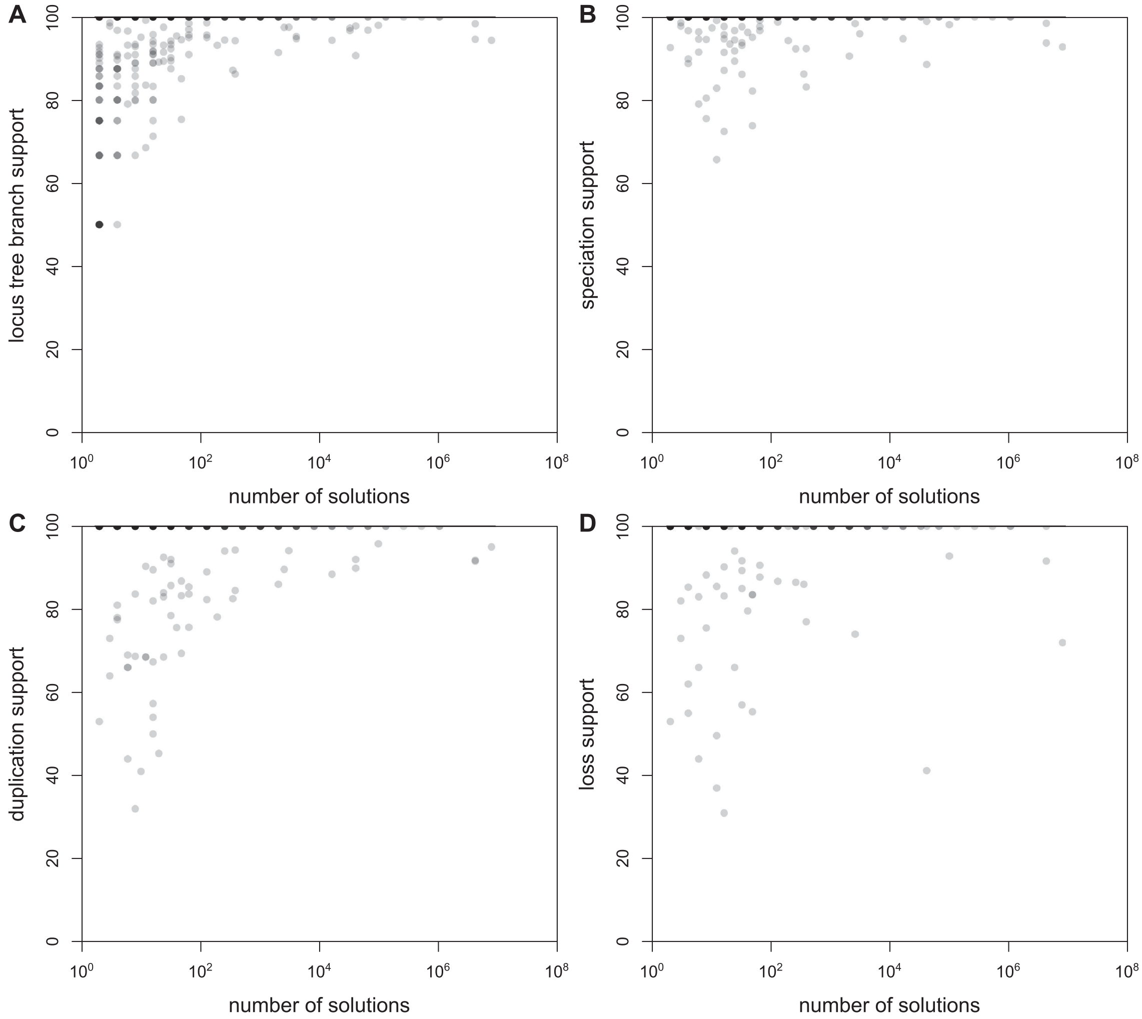

Fig. 4. *Event support.* For gene families with multiple optimal reconciliations, across 100 uniformly sampled reconciliations, average support of (A) locus tree branches, (B) speciations, (C) duplications, and (D) losses in a plurality optimal reconciliation.

Here, we highlight some differences that advantage and disadvantage the model used here.

- The DLCoal model is the only one based on the multispecies coalescent. Duplications and losses start in one allele and drift to fixation or extinction. In the Vernot-Stolzer and Chan models, duplications are considered instantaneous, so deep coalescence at duplications is not allowed. However, DLCoal must observe two incompletely sorted alleles (in the same locus and same species branch); other models allow for ILS in which one allele is immediately lost.
- The DLCoal model decouples the effects of the duplication-loss and multispecies coalescent processes through the concept of a locus tree, allowing reconciliations under this model to directly track the locus of genes and therefore to distinguish orthologs and paralogs. While events are mapped onto the gene tree

TABLE 1
Impact of Event Costs

costs[a]			reconciliations[b]		full support[c]					average support[d]			
D	*L*	*C*	single sol	max sol	locus tree	branches	specs	dups	losses	branches	specs	dups	losses
1	1	1	66.7	8.6b	74.2	96.9	97.0	88.7	86.6	98.5	98.7	93.7	93.9
1	1	0.75	66.7	8.6b	93.8	99.4	99.7	99.2	98.2	99.6	99.9	99.6	99.3
1	1	0.5	66.7	7.9m	91.8	99.0	99.4	97.4	97.2	99.4	99.7	98.7	98.6
1	1	0.25	66.8	8.6b	89.7	98.4	98.9	95.3	97.0	99.1	99.5	97.9	98.5
2	1	0.5	66.7	4.2m	92.1	99.1	99.5	97.7	98.0	99.5	99.8	98.8	99.1

[a] *The costs of duplications, losses and coalescences.*
[b] *Percentage of families (out of 5351 families) with a single reconciliation. Maximum number of reconciliations (in billions or millions) across all families.*
[c] *Across 100 sampled reconciliations for gene families with multiple optima, percentage of families with a single locus tree and percentage of locus tree branches, speciation events, duplication events, and loss events in a plurality optimal reconciliation with full support.*
[d] *Across 100 sampled reconciliations for gene families with multiple optima, average support for locus tree branches, speciation events, duplication events, and loss events in a plurality optimal reconciliation.*

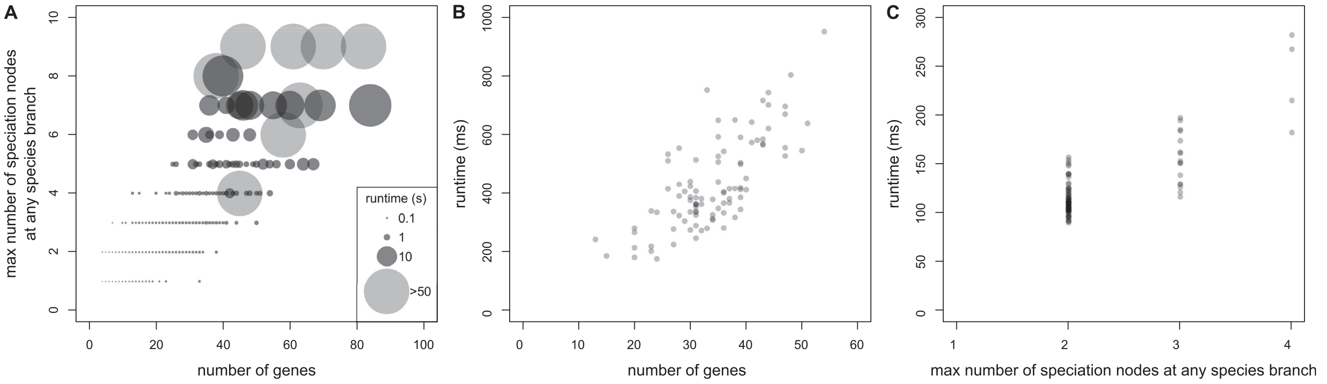

Fig. 5. *Runtime.* (A) Median runtime. (B) Runtime for 98 families with a maximum of four speciation nodes at any species branch. (C) Runtime for 98 families with 20 genes.

in the Vernot-Stolzer and Chan models, simply considering pairs of genes in the two subtrees of the mapped node will fail to account for the effects of ILS.

- The DLCoal model allows deep coalescence along any species branch whereas the Vernot-Stolzer and Chan models restrict the phenomena to certain parts of the species tree, in particular, at polytomies in the former and at marked branches in the latter. In contrast, DLCpar allows ILS anywhere and is unable to account for the probability of deep coalescence decreasing with branch length. It is unclear whether allowing deep coalescence only within certain species branches would invalidate Theorems 2.1, 2.2, and 2.3 on properties of MPRs under the DLCoal model.
- The Stolzer and Chan models allow transfers. While it would be straightforward to unify the duplication-transfer-loss and multispecies coalescent model (a "DTLCoal" model), the specifics of an associated reconciliation algorithm are unclear. A probabilistic framework for DTL reconciliation [26] could be substituted for the embedded DL reconciliation component in DLCoalRecon, but such an algorithm would require estimates of several additional parameters such as population sizes, species tree branch lengths, and duplication, transfer, and loss rates and likely be prohibitively slow in practice. A key efficiency of the DLCpar algorithm in the parsimony framework is that the optimal species map is the LCA map. This theorem almost certainly does not hold when transfers are included.

In previous work, we used simulated data sets to compare the performance of DLCpar, NOTUNG (which implements reconciliation under Vernot-Stolzer model), and LCA (the classic method for inferring MPRs under a duplication-loss-only model or a coalescent-only model). Events inferred by DLCpar had both higher precision and sensitivity compared to LCA. In contrast, while we found that NOTUNG correctly identifies spurious duplications due to ILS, the sensitivity of inferred duplications was similar to that of the LCA, and loss sensitivity and precision were often worse than that of LCA. To our knowledge, no implementation exists for reconciliation under the Chan model.

6 Discussion

In this work, we have presented new algorithms for understanding the space of maximum parsimony reconciliations under the DLC model. Specifically, we have shown how to compute the size of MPR space and to sample from this space uniformly at random. Our algorithms are efficient in practice, with runtime polynomial in the size of the species and gene tree when the number of genes that map to any given species is fixed. Our analysis of a biological data set provides some key insights into MPR space. In particular, we show the majority of gene families have a unique optimal reconciliation, and for gene families with multiple optima, events in a plurality reconciliation tend to be well-supported. These results suggest that reconciliations returned by DLCpar are likely to be representative of MPR space.

Our work represents a first step towards understanding MPR space, and there are several directions for future work, especially for gene trees with multiple optima. For example, while we have summarized MPR space through sampling, several other approaches are possible. For MPRs under the DTL model, methods exist not only for sampling [29] but also for compactly representing the space of all MPRs [30], computing a medoid MPR [31], finding a set of reconciliations that collectively cover the most frequently occurring events in MPR space [32], implicitly clustering MPR space [33], and computing the diameter of MPR space [34]. We expect that it may be possible to similarly explore MPR space under the DLC model.

Acknowledgments

The authors thank Eliot Bush for use of his cluster and Mukul Bansal for his helpful discussions. 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 US National Science Foundation under Grant No. IIS-1751399 to YW and IIS-1419739 to RLH. H. Du and Y. S. Ong contributed equally to this work.

References

[1] S. Ohno, *Evolution by Gene Duplication*. New York, NY, USA: Springer-Verlag, 1970.

[2] M. Lynch and J. S. Conery, "The evolutionary fate and consequences of duplicate genes," *Sci.*, vol. 290, no. 5494, pp. 1151–1155, Nov. 2000. [Online]. Available: http://www.sciencemag.org/cgi/content/full/290/5494/1151

[3] E. V. Koonin, "Orthologs, paralogs, and evolutionary genomics," *Annu. Rev. Genetics*, vol. 39, no. 1, pp. 309–338, 2005.

[4] M. E. Peterson, F. Chen, J. G. Saven, D. S. Roos, P. C. Babbitt, and A. Sali, "Evolutionary constraints on structural similarity in orthologs and paralogs," *Protein Sci.*, vol. 18, no. 6, pp. 1306–1315, 2009. [Online]. Available: http://dx.doi.org/10.1002/pro.143

[5] Y. Niimura and M. Nei, “Extensive gains and losses of olfactory receptor genes in mammalian evolution,” *PLoS One*, vol. 2, no. 1, 2007, Art. no. e708. [Online]. Available: http://dx.doi.org/10.1371/journal.pone.0000708

[6] M. Goodman, J. Czelusniak, G. W. Moore, A. Romero-Herrera, and G. Matsuda, “Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences,” *Systematic Zoology*, vol. 28, no. 2, pp. 132–163, 1979.

[7] R. D. Page, “Maps between trees and cladistic analysis of historical associations among genes,organisms, and areas,” *Systematic Biol.*, vol. 43, no. 1, pp. 58–77, Mar. 1994. [Online]. Available: http://sysbio.oxfordjournals.org/content/43/1/58.abstract

[8] K. Chen, D. Durand, and M. Farach-Colton, “NOTUNG: A program for dating gene duplications and optimizing gene family trees,” *J. Comput. Biol.*, vol. 7, no. 3/4, pp. 429–447, Aug. 2000. [Online]. Available: http://dx.doi.org/10.1089/106652700750050871

[9] C. M. Zmasek and S. R. Eddy, “A simple algorithm to infer gene duplication and speciation events on a gene tree,” *Bioinf.*, vol. 17, no. 9, pp. 821–828, Sep. 2001. [Online]. Available: http://bioinformatics.oxfordjournals.org/content/17/9/821.abstract

[10] P. Górecki and J. Tiuryn, “DLS-trees: A model of evolutionary scenarios,” *Theoretical Comput. Sci.*, vol. 359, no. 1–3, pp. 378–399, Aug. 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0304397506003185

[11] C. Chauve, J.-P. Doyon, and N. El-Mabrouk, “Gene family evolution by duplication, speciation, and loss,” *J. Comput. Biol.*, vol. 15, no. 8, pp. 1043–1062, Oct. 2008.

[12] O. Åkerborg, B. Sennblad, L. Arvestad, and J. Lagergren, “Simultaneous Bayesian gene tree reconstruction and reconciliation analysis,” *Proc. Nat. Academy Sci. United States America*, vol. 106, no. 14, pp. 5714–5719, Apr. 2009. [Online]. Available: http://www.pnas.org/content/106/14/5714.abstract

[13] M. D. Rasmussen and M. Kellis, “A Bayesian approach for fast and accurate gene tree reconstruction,” *Mol. Biol. Evol.*, vol. 28, no. 1, pp. 273–290, Jan. 2011. [Online]. Available: http://mbe.oxfordjournals.org/content/28/1/273.abstract

[14] C. Conow, D. Fielder, Y. Ovadia, and R. Libeskind-Hadas, “Jane: A new tool for the cophylogeny reconstruction problem,” *Algorithm Mol. Biol.*, vol. 5, no. 16, 2010. [Online]. Available: http://www.almob.org/content/5/1/16

[15] L. A. David and E. J. Alm, “Rapid evolutionary innovation during an Archaean genetic expansion,” *Nature*, vol. 469, no. 7328, pp. 93–96, Jan. 2011. [Online]. Available: http://dx.doi.org/10.1038/nature09649

[16] J.-P. Doyon, C. Scornavacca, K. Gorbunov, G. J. Szöllősi, V. Ranwez, and V. Berry, “An efficient algorithm for gene/species trees parsimonious reconciliation with losses, duplications and transfers,” in *Comparative Genomics*, E. Tannier, Ed. Berlin, Germany: Springer, 2011, pp. 93–108. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-16181-0_9

[17] A. Tofigh, M. Hallett, and J. Lagergren, “Simultaneous identification of duplications and lateral gene transfers,” *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 8, no. 2, pp. 517–535, Mar./Apr. 2011. [Online]. Available: http://dx.doi.org/10.1109/TCBB.2010.14

[18] M. S. Bansal, E. J. Alm, and M. Kellis, “Efficient algorithms for the reconciliation problem with gene duplication, horizontal transfer and loss,” *Bioinf.*, vol. 28, no. 12, pp. i283–i291, Jun. 2012. [Online]. Available: http://bioinformatics.oxfordjournals.org/content/28/12/i283.abstract

[19] 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,” *Bioinf.*, vol. 28, no. 18, pp. 409–415, 2012.

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

[21] B. Rannala and Z. Yang, “Bayes estimation of species divergence times and ancestral population sizes using DNA sequences from multiple loci,” *Genetics*, vol. 164, no. 4, pp. 1645–1656, Aug. 2003.

[22] 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. Suppl 9, 2011, Art. no. S7. [Online]. Available: http://www.biomedcentral.com/1471–2105/12/S9/S7

[23] L. Zhang, “From gene trees to species trees II: Species tree inference by minimizing deep coalescence events,” *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 8, no. 6, pp. 1685–1691, Nov./Dec. 2011. [Online]. Available: http://dx.doi.org/10.1109/TCBB.2011.83

[24] M. D. Rasmussen and M. Kellis, “Unified modeling of gene duplication, loss, and coalescence using a locus tree,” *Genome Res.*, vol. 22, pp. 755–765, 2012. [Online]. Available: http://genome. cshlp.org/content/early/2012/01/23/gr.123901.111.abstract

[25] Y.-C. Wu, M. D. Rasmussen, M. S. Bansal, and M. Kellis, “Most parsimonious reconciliation in the presence of gene duplication, loss, and deep coalescence using labeled coalescent trees,” *Genome Res.*, vol. 24, no. 3, pp. 475–486, Mar. 2014. [Online]. Available: http://genome.cshlp.org/content/24/3/475.abstract

[26] A. Tofigh, “Using trees to capture reticulate evolution : Lateral gene transfers and cancer progression,” PhD dissertation, KTH Royal Institute of Technology, 2009. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-10608

[27] B. Donati, C. Baudet, B. Sinaimeri, P. Crescenzi, and M.-F. Sagot, “EUCALYPT: Efficient tree reconciliation enumerator,” *Algorithm Mol. Biol.*, vol. 10, no. 1, 2015, Art. no. 3. [Online]. Available: https://doi.org/10.1186/s13015-014-0031-3

[28] E. Jacox, C. Chauve, G. J. Szöllősi, Y. Ponty, and C. Scornavacca, “ecceTERA: Comprehensive gene tree-species tree reconciliation using parsimony,” *Bioinf.*, vol. 32, no. 13, pp. 2056–2058, Jul. 2016. [Online]. Available: http://dx.doi.org/10.1093/bioinformatics/btw105

[29] 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, Oct. 2013. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3791060/

[30] C. Scornavacca, W. Paprotny, V. Berry, and V. Ranwez, “Representing a set of reconciliations in a compact way,” *J. Bioinf. Com-put. Biol.*, vol. 11, no. 2, Apr. 2013, Art. no. 1250025, pMID: 23600816. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S0219720012500254

[31] T.-H. Nguyen, V. Ranwez, V. Berry, and C. Scornavacca, “Support measures to estimate the reliability of evolutionary events predicted by reconciliation methods,” *PLoS One*, vol. 8, no. 10, Oct. 2013, Art. no. e73667. [Online]. Available: https://doi.org/10.1371/journal.pone.0073667

[32] W. Ma, D. Smirnov, J. Forman, A. Schweickart, C. Slocum, S. Srinivasan, and R. Libeskind-Hadas, “DTL-RnB: Algorithms and tools for summarizing the space of DTL reconciliations,” *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 15, no. 2, pp. 411–421, Mar./Apr. 2018.

[33] A. Ozdemir, M. Sheely, D. Bork, R. Cheng, R. Hulett, J. Sung, J. Wang, and R. Libeskind-Hadas, “Clustering the space of maximum parsimony reconciliations in the duplication-transfer-loss model,” in *Proc. Int. Conf. Algorithms Comput. Biol.*, 2017, pp. 127–139.

[34] J. Haack, E. Zupke, N. Ramirez, Y.-C. Wu, and R. Libeskind-Hadas, “Computing the diameter of the space of maximum parsimony reconciliations in the duplication-transfer-loss model,” *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 16, no. 1, pp. 14–22, Jan. 2019. doi: 10.1109/TCBB.2018.2849732.

[35] D. Bork, R. Cheng, J. Wang, J. Sung, and R. Libeskind-Hadas, “On the computational complexity of the maximum parsimony reconciliation problem in the duplication-loss-coalescence model,” *Algorithm Mol. Biol.*, vol. 12, 2017, Art. no. 6.

[36] I. Wapinski, A. Pfeffer, N. Friedman, and A. Regev, “Natural history and evolutionary principles of gene duplication in fungi,” *Nature*, vol. 449, no. 7158, pp. 54–61, Sep. 2007. [Online]. Available: http://dx.doi.org/10.1038/nature06107

[37] F. Tajima, “Evolutionary relationship of DNA sequences in finite populations,” *Genetics*, vol. 105, no. 2, pp. 437–460, Oct. 1983. [Online]. Available: http://www.genetics.org/content/105/2/437.abstract

[38] P. Pamilo and M. Nei, “Relationships between gene trees and species trees,” *Mol. Biol. Evol.*, vol. 5, no. 5, pp. 568–583, Sep. 1988. [Online]. Available: http://mbe.oxfordjournals.org/content/5/5/568.abstract

[39] N. A. Rosenberg, “The probability of topological concordance of gene trees and species trees,” *Theoretical Population Biol.*, vol. 61, no. 2, pp. 225–247, Mar. 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0040580901915680

[40] J. H. Degnan and N. A. Rosenberg, “Gene tree discordance, phylogenetic inference and the multispecies coalescent,” *Trends Ecology Evol.*, vol. 24, no. 6, pp. 332–340, Jun. 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169534709000846

[41] A. Itai, “Generating permutations and combinations in lexicographical order,” *J Brazilian Comput. Soc.*, vol. 7, pp. 65–68, 2001. [Online]. Available: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0104–65002001000200009&nrm=iso

[42] M. Orlov, "Efficient generating of set partitions," pp. 1–6. [Online]. Available: http://www.informatik.uni-ulm.de/ni/Lehre/WS03/DMM/Software/partitions.pdf

[43] G. Butler, M. D. Rasmussen, M. F. Lin, M. A. S. Santos, S. Sakthikumar, C. A. Munro, E. Rheinbay, M. Grabherr, A. Forche, J. L. Reedy, I. Agrafioti, M. B. Arnaud, S. Bates, A. J. P. Brown, S. Brunke, M. C. Costanzo, D. A. Fitzpatrick, P. W. J. de Groot, D. Harris, L. L. Hoyer, B. Hube, F. M. Klis, C. Kodira, N. Lennard, M. E. Logue, R. Martin, A. M. Neiman, E. Nikolaou, M. A. Quail, J. Quinn, M. C. Santos, F. F. Schmitzberger, G. Sherlock, P. Shah, K. A. T. Silverstein, M. S. Skrzypek, D. Soll, R. Staggs, I. Stansfield, M. P. H. Stumpf, P. E. Sudbery, T. Srikantha, Q. Zeng, J. Berman, M. Berriman, J. Heitman, N. A. R. Gow, M. C. Lorenz, B. W. Birren, M. Kellis, and C. A. Cuomo, "Evolution of pathogenicity and sexual reproduction in eight *Candida* genomes," *Nature*, vol. 459, no. 7247, pp. 657–662, Jun. 2009. [Online]. Available: http://dx.doi.org/10.1038/nature08064

[44] Y.-C. Wu, M. D. Rasmussen, M. S. Bansal, and M. Kellis, "TreeFix: Statistically informed gene tree error correction using species trees," *Systematic Biol.*, vol. 62, no. 1, pp. 110–120, Jan. 2013. [Online]. Available: http://sysbio.oxfordjournals.org/content/62/1/110.abstract

[45] B. Vernot, M. Stolzer, A. Goldman, and D. Durand, "Reconciliation with non-binary species trees," *J. Comput. Biol.*, vol. 15, no. 8, pp. 981–1006, Sep. 2008. [Online]. Available: http://www.liebertonline.com/doi/abs/10.1089/cmb.2008.0092

[46] Y.-B. Chan, V. Ranwez, and C. Scornavacca, "Inferring incomplete lineage sorting, duplications, transfers and losses with reconciliations," *J. Theoretical Biol.*, vol. 432, pp. 1–13, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022519317303740

Haoxing Du is working toward the BS degree in physics at Harvey Mudd College.

Yi Sheng Ong is working toward the BS degree in computer science and mathematics at Harvey Mudd College.

Marina Knittel received the BS degree in computer science and mathematics from Harvey Mudd College, in 2018. She is working toward the PhD degree in computer science at the University of Maryland.

Ross Mawhorter received the BS degree in computer science and mathematics from Harvey Mudd College in 2019.

Nuo Liu is working toward the BS degree in mathematical and computational biology at Harvey Mudd College.

Gianluca Gross is working toward the BSE degree in computer science at the University of Pennsylvania.

Reiko Tojo received the BS degree in computer science and mathematics from Harvey Mudd College in 2018.

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 assistant professor of computer science with Harvey Mudd College.