reconcILS: A gene tree-species tree reconciliation algorithm that allows for incomplete lineage sorting

Sarthak Mishra[1], Megan L. Smith[2], and Matthew W. Hahn[1]

[1] Indiana University, Bloomington IN, USA {samishr,mwh}@iu.edu
[2] Mississippi State University, Starkville MS, USA
ms4438@msstate.edu

Abstract. Reconciliation algorithms provide an accounting of the evolutionary history of individual gene trees given a species tree. Many reconciliation algorithms consider only duplication and loss events (and sometimes horizontal transfer), ignoring effects of the coalescent process, including incomplete lineage sorting (ILS). Here, we present a new algorithm for carrying out reconciliation that accurately accounts for incomplete lineage sorting by treating ILS as a series of nearest neighbor interchange (NNI) events. For discordant branches of the gene tree identified by last common ancestor (LCA) mapping, our algorithm recursively chooses the optimal history by comparing the cost of duplication and loss to the cost of NNI and loss. We demonstrate the accuracy of our new method, which we call `reconcILS`, using a new simulation engine (`dupcoal`) that can accurately generate gene trees produced by the interaction of duplication, ILS, and loss. We show that `reconcILS` is much more accurate than models that ignore ILS, and at least as accurate or better than the leading method that can model ILS, duplication, and loss. We discuss how our method can also be extended in the future for use as a reconciliation algorithm in additional scenarios.
Availability: `reconcILS` is implemented in Python 3 and is available at `https://github.com/smishra677/reconcILS`. The `dupcoal` simulator is implemented in Python 3 and is available at `https://github.com/meganlsmith/dupcoal`

Keywords: reconciliation · duplication · loss · coalescent · hemiplasy · incomplete lineage sorting.

1 Introduction

The sequencing of a large number of genomes from many different species has highlighted the dynamic nature of genes: they are often duplicated, lost, or moved between species via introgression (no gain of an extra copy) or horizontal gene transfer (gain of an extra copy). One of the main goals of evolutionary and comparative genomics has been to quantify this dynamism, both in terms of the number of events of each type and their timing. A popular approach for quantifying both measures is gene tree-species tree reconciliation (reviewed in Boussau and Scornavacca 2020). Given a gene tree topology and a species tree topology, reconciliation algorithms attempt to infer the evolutionary events necessary to explain any discordance in the respective topologies. Many different reconciliation algorithms have been developed, most often considering only duplication and loss (so-called "DL" models; e.g Goodman et al. 1979; Page 1994; Guigo et al. 1996; Page and Charleston 1997; Chen et al. 2000; Zmasek and Eddy 2001; Durand et al. 2006), but also sometimes considering duplication, transfer, and loss ("DTL" models; e.g. Bansal et al. 2012; Szöllősi et al. 2013; Jacox et al. 2016; Morel et al. 2020).

Despite the success—and widespread use—of DL and DTL reconciliation algorithms, they do not consider gene tree discordance due to the coalescent process. The coalescent is a model that describes how different sequences within a population are related (Kingman 1982; Hudson 1983; Tajima 1983), and is relevant even when considering sequences from different species because these must also coalesce in shared ancestral populations. Importantly, gene trees and species trees can be discordant solely due to coalescence, in a process often called incomplete lineage sorting (ILS) or deep coalescence (Maddison 1997). Standard DL and DTL algorithms will incorrectly infer one duplication and three losses for every discordance only due to ILS (Hahn 2007), resulting in misleading results when applying reconciliation methods to trees with ILS (e.g. Stull et al. 2021). Duplication, loss, and coalescence can also interact in complex and non-intuitive ways (Rasmussen and Kellis 2012; Mallo et al. 2014; Li et al. 2021), making it almost impossible to simply exclude gene trees discordant due to ILS from an analysis.

A few different approaches to reconciliation have been used to deal with discordance due to coalescence. An early approach allowed short branches in the species tree—which can lead to ILS—to be collapsed (Vernot et al. 2008; Stolzer et al. 2012). The use of non-binary species trees means that such algorithms will not mistake ILS alone for duplication and loss; however, they do not account for interactions between events. Chan et al. (2017) introduced a reconciliation algorithm that included duplication, loss, transfer, and ILS, but it assumed that new duplicates were completely linked to their parental locus (i.e. no recombination between them). Given that the majority of new gene duplicates are not completely linked (Schrider and Hahn 2010), such an assumption limits the number of genes that can be analyzed. Wu et al. (2014) developed a parsimony-based algorithm (DLCpar) that models duplication, loss, and incomplete lineage sorting; this approach is similar to the probabilistic DLCoalRecon algorithm (Rasmussen and Kellis 2012), but runs much faster and without the need to specify duplication and loss rates ahead of time.

Here, we introduce a new algorithm for finding the most parsimonious reconciliation of a gene tree and a species tree in the presence of duplication, loss, and incomplete lineage sorting. Our goal is to improve upon existing methods by increasing speed and accuracy, as well as by incorporating more coalescent processes (i.e. ILS is not the only cause of discordance due to the coalescent). In addition, although DLCpar has been shown to perform well (Wu et al. 2014; Du et al. 2021), it is not straightforward to extend its algorithm to other tools that also have reconciliation steps (e.g. Thomas et al. 2017; Zhang et al. 2020). We hope that the algorithm introduced here will be easily adaptable to such scenarios.

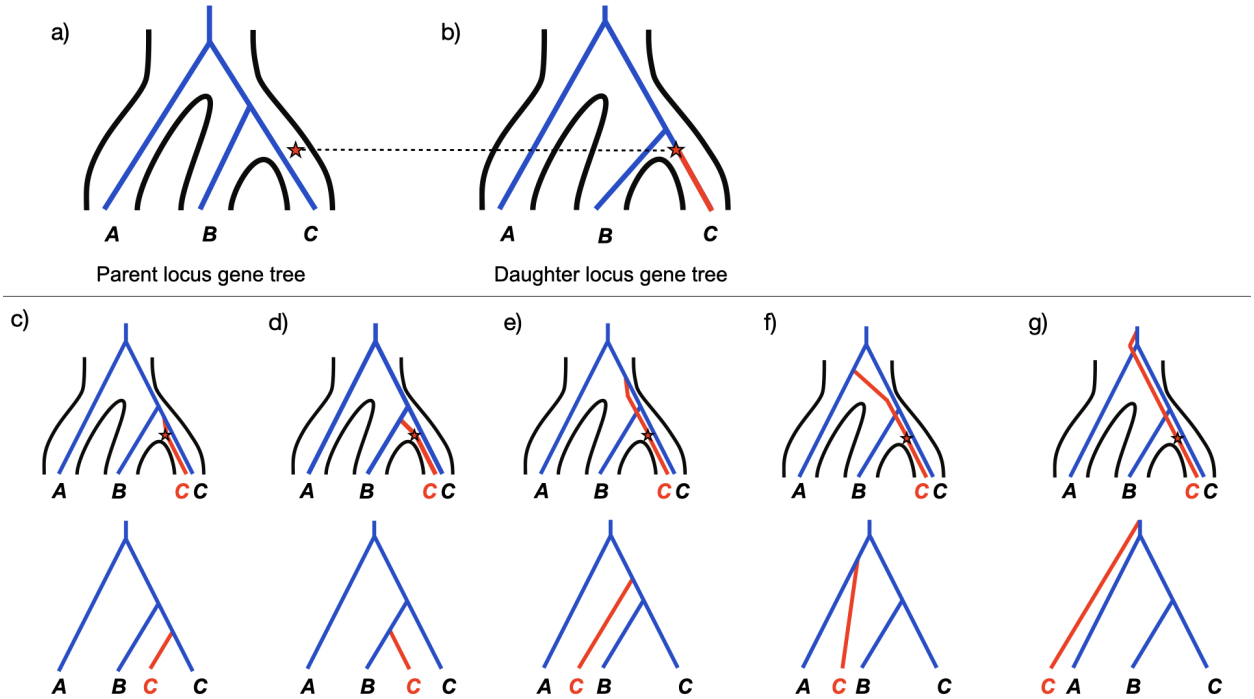

Fig. 1. The multilocus multispecies coalescent model generates full gene trees. a) A gene tree at the parent locus (blue) is drawn from the species tree (outlined in black) via the multispecies coalescent (MSC) model. A duplication event (red star) is drawn from a birth-death model over the species tree, in this case occurring in the population ancestral to species B and C. b) A gene tree at the daughter locus is drawn from the MSC. The duplication is placed on this gene tree at the time at which it occurred in the species tree (shown as a dotted line), choosing one of the two possible branches at random. The inserted DNA now takes on the gene tree branch labeled in red. c)-g) Possible ways that the parent and daughter loci coalesce to form the full gene tree. In all panels the star and red branch below it come from the daughter gene tree, and the blue tree comes from the parent gene tree. The coalescent process joining the two trees is shown as a red branch above the star. The trees in the bottom row are simply unfolded views of the full gene trees.

2 New algorithm

2.1 Multilocus multispecies coalescent simulator

We begin with a description of the simulation program introduced here, `dupcoal`, as it allows us to explain the biological model underlying both the simulations and inferences.

The model we use closely resembles the multilocus multispecies coalescent (MLMSC) model of Li et al. (2021). This model treats each individual locus as evolving via the multispecies coalescent (MSC) model, in which discordance between individual gene trees and the species tree can arise only via incomplete lineage sorting. The MLMSC further models relationships among these loci, in addition to losses at any locus.

To illustrate how duplications and losses occur, and how relationships among loci are generated in `dupcoal`, we use a simplified example (a full description of the simulation algorithm can be found at `https://github.com/meganlsmith/dupcoal`). As a first step, a gene tree is generated via the MSC at the "parent" locus (Figure 1a); this tree exists regardless of whether subsequent duplications occur. Duplications and losses are then generated via a birth-death process on the species tree, with each event occurring at a specific time on a specific branch of the species tree (one such duplication is represented as a star in Figure 1a). For each duplication event, a gene tree is generated via the MSC at a "daughter" locus (Figure 1b). This tree represents the history of the locus at which the duplicated DNA will be inserted, and exists with or without this duplication event; for now we assume that this locus is unlinked from the parent locus. The mutation representing the duplication event is then placed on the gene tree at the daughter locus at the same time that it occurred on the species tree—if multiple branches of the daughter gene tree exist at this point in time on the relevant branch, the mutation is assigned to one at random (star in Figure 1b). If the mutation occurs on a discordant branch of the daughter gene tree (that is, one that does not exist in the species tree topology), this is referred to as "hemiplasy" or "copy-number hemiplasy." Such events are not simulated by SimPhy (Mallo et al. 2016). Regardless of where exactly the mutation occurs on the daughter gene tree, the duplication now takes on the topology and branch lengths of the tree below this point (red branch in Figure 1b).

The process described thus far has produced two gene trees—one at the parent locus and one at the daughter locus—either or both of which may be discordant due to ILS. Because gene duplication is fundamentally a process by which an existing haplotype in a population at one locus is inserted into another location, the inserted DNA at the new daughter locus carries with it a history of coalescence with all of the other haplotypes at the parent locus. This coalescent process joining the two trees can cause additional discordance, even among gene trees that individually cannot be discordant (e.g. when the species tree has only two species). We illustrate this coalescent process here with a simple example.

Figures 1c-g show the five outcomes possible when joining the parent and daughter trees shown in Figures 1a and 1b. Figure 1c is the simplest outcome, as the daughter tree joins the parent tree on the same branch that the mutation occurred on (i.e. the one leading to species C); this is the tree that is assumed to be produced by duplication under standard DL reconciliation. Figures 1d and 1e represent cases where the daughter tree joins the parent tree on a different branch, each of which would require one NNI move to recover the tree topology in Figure 1c. Finally, Figures 1f and 1g represent more extreme cases, in which coalescence between parent and daughter further up the tree requires two NNI moves to recover the topology in Figure 1c.

It should be emphasized here that all of the topologies presented in Figure 1 involve only a single duplication and no losses, and neither the parent tree nor the daughter tree is discordant due to ILS. In fact, the examples in Figures 1d and 1e are not technically due to ILS either, since coalescence is occurring in the same population of the species tree (the ancestor of species B and C); in addition, in this scenario these two trees are actually equiprobable with the topology in Figure 1c. These simple examples demonstrate that the vagaries of the coalescent process lead to a thicket of different tree topologies.

Losses are also simulated on the species tree and placed on each gene tree. Loss events are placed on branches of gene trees in the same manner as duplications, with all branches below such mutations removed. Together, these processes make up the steps by which our simulator can generate full gene trees (i.e. ones with the parent and all daughter trees joined together). We used functions from DendroPy (Sukumaran and Holder 2010) to generate trees under the MSC and to manipulate parent and daughter trees; we used functions from NumPy (Harris et al. 2020) to draw coalescent times, duplication times, and loss times from exponential distributions.

2.2 reconcILS

Inputs and outputs. Given a rooted, bifurcating species tree, S, and a rooted, bifurcating gene tree, G, as input, reconcILS outputs the minimum number of duplications, losses, and NNIs required to reconcile G and S. Inferred events are all also assigned to nodes in the gene tree.

Main algorithm. An overview of reconcILS is given in Algorithm 1 (see Appendix). Our approach combines multiple sub-approaches that are commonly used in DL reconciliation and other phylogenetic inferences. As a first step, we use last common ancestor (LCA) mapping (Zmasek and Eddy 2001) to efficiently map each node in G to a node in S (Figure 2a).

Unlike in standard DL methods, LCA mapping here is used to initialize each round of our algorithm—and is sometimes used iteratively within a round—without representing the final reconciliation. Instead, this step identifies cases in which multiple nodes in the gene tree map to an identical node of the species tree. For instance, in Figure 2a nodes G_1 and G_2 in the gene tree both map to node S_1 in the species tree. We therefore call S_1 a "multiply mapped" node. Additionally, we refer to the branch in the gene tree bracketed by nodes G_1 and G_2 as a "discordant" branch (labeled K_1 in Figure 2a); such branches do not exist in the species tree. The focus of our method is in determining whether discordant branches should be resolved via NNI and loss (Figure 2b) or via duplication and loss (Figure 2c). Choosing among events (NNI, duplication, and loss) is carried out greedily for each branch, with little consideration of a globally optimal set of events among all discordant branches in the gene tree. Consequently, after choosing a set of events for one branch, the process repeats itself until there are no multiply mapped nodes and no discordant branches. Duplications implied by multiply mapped tip nodes in the species trees are inferred via standard methods; inferences of losses are explained below.

Traversing trees. The high-level description given above does not provide details about several important steps in our algorithm. In this section and the next two we provide some more information.

After the initial LCA mapping, our algorithm performs a preorder traversal of the species tree. For each species tree node the algorithm checks how many gene tree nodes are mapped to it. If there are fewer than 2 mappings, the algorithm invokes a function that asks whether speciation or loss can be inferred. Nodes with 1 mapping are speciation events—these do not incur a cost. Nodes that have 0 mappings represent losses. Our function asks whether the parent node of a node with 0 mappings itself has 0 mappings. If not, a loss is inferred; if so, no event is recorded, because the loss must have occurred further up the tree. If there are 2 or more mappings the species tree node is multiply mapped, and a function to carry out "local reconciliation" is called. Local reconciliation is described in more detail in the next section, but here can simply be described as determining the events that have generated the multiple mappings to a single species tree node.

If there are only two adjacent nodes in the gene tree mapped to a node in the species tree (i.e. one discordant branch), then the method proceeds as described below. However, when there are more than two nodes mapped to an internal species tree node—and therefore more than one discordant branch in the gene tree—we must decide which branch to reconcile with the multiply mapped node first. We have found some cases where the order in which they are resolved matters, with only one of the choices leading to an optimal reconciliation. Algorithm 2 (see Appendix) describes how we choose which discordant branch to reconcile first. This algorithm calculates the "bipartition cost" of each possible branch—the number of bipartitions and losses induced by NNI moves applied to each branch (Figure A1). Such a calculation can be carried out quickly, and by choosing the branch with the lowest bipartition cost, we can more accurately proceed with reconciliation.

Once reconciliation for a multiply mapped node is carried out, our algorithm begins a new round. In each round we repeat LCA mapping and continue traversing the species tree, counting mappings at each node.

Choosing among events. In order for our algorithm to more efficiently and optimally choose among different evolutionary events (e.g. Figure 2b vs 2c), we have introduced the concept of a "local reconciliation" for a multiply mapped node. Here, a species tree node is locally reconciled if the number of gene tree nodes mapped to it, m, is reduced; for internal nodes of the species tree, this by definition also reduces the number of discordant branches in the gene tree.

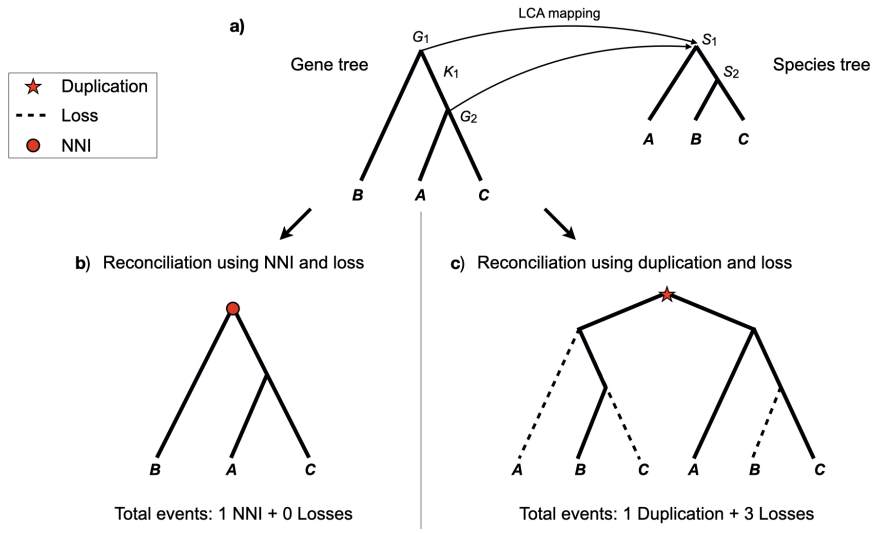

Fig. 2. Summary of `reconcILS`. a) The initial step carries out LCA mapping from the gene tree to the species tree. Here, nodes G_1 and G_2 in the gene tree both map to node S_1 in the species tree; we therefore refer to S_1 as a multiply mapped node. Additionally, G_1 and G_2 together define a discordant branch, K_1, in the gene tree. For clarity, all mappings involving tip nodes have been excluded. b) The algorithm attempts to reconcile the discordant branch using NNI and loss. Here, reconciliation can be carried out with 1 NNI and 0 losses. c) The algorithm attempts to reconcile the discordant branch using duplication and loss. Here, reconciliation can be carried out with 1 duplication and 3 losses. `reconcILS` must decide between the reconcilations in b) and c) for every discordant branch.

The use of local reconciliation has several advantages. Most importantly, it provides a criterion for moving forward: we can determine when reconciliation has been achieved within a single round of the algorithm. It also makes it clear that there are some scenarios that can only be reconciled using duplication and not NNI (e.g. a multiply mapped tip node), in which case choosing events is straightforward. In a single round of our algorithm, we attempt to locally reconcile an identified multiply mapped node using both NNI+loss and duplication+loss (Figure 2).

Note that, given our definition, a species tree node can be locally reconciled by NNI while still remaining multiply mapped; e.g., if $m = 3$ is reduced to $m = 2$, the node still requires more reconciliation. In this case our algorithm will carry out up to $m - 1$ NNI moves within a single round, as long as each one results in a local reconciliation. For example, starting with $m = 3$, the first NNI move will locally reconcile a node if one of the two NNI topologies reduces the number of mappings to the species tree (we have never observed a case where both NNI topologies reduce this number). In this case, $m = 2$, and we now carry out another NNI move on the updated gene tree. Only one move needs to be examined at this point, however, since the back-NNI rearrangement would return us to the starting state. The cost of all of these NNI moves is then compared to the cost of duplication and loss at the end of the round. Duplication+loss naturally allows for more than one of each type of event for a local reconciliation, and is therefore not handled differently when $m > 2$.

As with all parsimony-based reconciliations, `reconcILS` requires a cost for each event type: one for duplications (C_D), one for losses (C_L), and one for NNIs (C_I). Reconciliations can be very sensitive to costs, such that maximum parsimony solutions under one set of costs can be very different than under another set of costs. Although there are few ways to determine costs independently (but see Discussion), there are some approaches for finding the Pareto-optimal costs within a reconciliation context (e.g. Libeskind-Hadas et al. 2014; Mawhorter et al. 2019). Here, as in most commonly used reconcilation algorithms, we allow the user to set costs. The default costs within `reconcILS` are set to $C_D = 1.1$, $C_L = 1$, and $C_I = 1$. (The default costs within DLCpar are $C_D = 2$, $C_L = 2$, and $C_I = 1$.) It is also important to note that DLCpar is specifying a cost of ILS, while `reconcILS` is specifying a cost for each NNI induced by even a single ILS event. This distinction will become important when we test the accuracy of each method below, as we will

be inferring slightly different events with the two methods (we use the same costs in both methods for these comparisons).

Finally, at the end of a round, `reconcILS` will sum the cost of all events in order to choose the optimal local reconciliation. If there are equally optimal reconciliations, `reconcILS` will choose the duplication+loss solution.

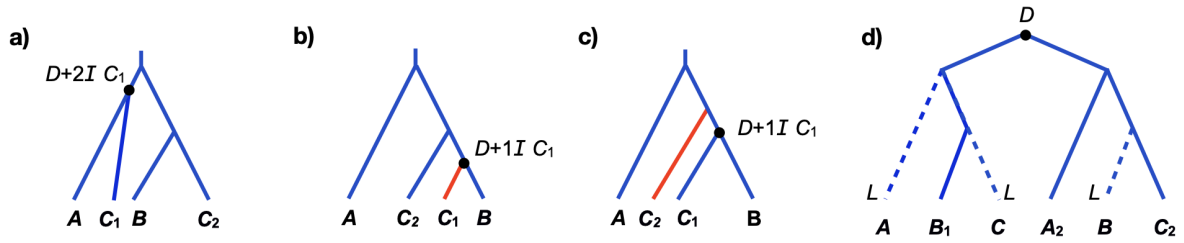

Fig. 3. Labeling reconciled gene trees. a) One possible reconciliation of the gene tree from Figure 1f labeled by `reconcILS`. The single ILS node (I) has been labeled with the branch that has been rearranged by NNI (C_1), the number of NNIs it has undergone (2), and the fact that it also represents a duplication (D). b) The gene tree from Figure 1d labeled by `reconcILS`. This tree is correctly labeled, but note that it is the same topology as in panel c (see main text for further explanation). c) The gene tree from Figure 1e labeled by `reconcILS`. Note that the program has labeled the bottom-most node flanking the discordant branch with D and I, although that is not the true history. d) A gene tree with only duplications and losses (L) labeled by `reconcILS`. Internal branches could also be labeled with L, though none are shown here.

Labeling a reconciled gene tree. A full reconciliation of a gene tree, G, and species tree, S, will have locally reconciled all multiply mapped nodes. The total number of duplications, losses, and NNIs required for this reconciliation is the sum of each event type across all branches; this number is reported by `reconcILS`. We also aim to label the gene tree with these events—identifying where each event occurred in history—as this is an important output of reconciliation algorithms. As `reconcILS` traversed the species tree finding optimal local reconciliations, it also recorded the number and location of all events. Labeling therefore consists of placing these events on the gene tree.

Traditional DL reconciliation algorithms label gene tree nodes as either D (for duplication) or S (for speciation); they may also label the location of losses, L. Algorithms including ILS must additionally label nodes as I. However, there is no standard way to do this labelling: for instance, RecPhyloXML, a format for representing reconciled trees, does not include any way to label ILS events (Duchemin et al. 2018). Consequently, we have created our own labeling system for ILS, one that we think is both understandable and easy to work with.

Our solution for placing the I label on nodes of gene trees in `reconcILS` is to denote both the daughter branch affected and the number of NNI moves that were associated with this branch. A few examples are shown in Figure 3, which also helps to illustrate why these choices were made. Figure 3a labels the gene tree from Figure 1f—with more complete tip names—with output from `reconcILS`. As can be seen, this tree has one node (denoted with a small black dot for clarity) labeled with both duplication and ILS. Further, the labeling tells you which daughter branch was reconciled with NNI (C_1), and how many NNI moves were required to reconcile it (2). Without such information, simply labeling the node with I would not have distinguished histories involving coalescent events along the A branch from those along the C_1 branch.

There are two important caveats to this approach, one perhaps more obvious than the other. First, a specific labeling is associated with a specific reconciliation. That is, if we had found a different optimal reconciliation for the tree in Figure 3a (for instance, one duplication and two losses), we would have had different labeled events. So, as might be expected, a labeling is always reported in the context of a particular reconciliation. Second, for some ILS events it is not always clear which of the two nodes flanking a discordant

branch to label with I: we have arbitrarily set `reconcILS` to always choose the node closest to the tips. In some cases this choice should not matter very much. Consider the toy reconciliation in Figure 2b—should we have placed the NNI event on node G_1 (as shown), or on node G_2? A branch undergoes NNI, not a node, so it is not clear which node flanking such a branch should be chosen; in this case the choice does not seem to imply any difference in evolutionary histories. In contrast, consider the reconciliations shown in Figure 3b and 3c—these represent the output of `reconcILS` applied to the gene trees in Figure 1d and 1e, respectively. While we have left the daughter tree branch in red for clarity, it is important to note that these trees are identical topologically, which is why `reconcILS` has labelled them in exactly the same way. The algorithm has no way to "see" the red branches, and has consequently simply picked the bottom-most node in both cases as the one to label as having undergone ILS and duplication, even though this is not precisely correct in Figure 3c. At the moment this is a limitation of `reconcILS`, but we mention possible solutions in the Discussion.

`reconcILS` also labels duplications and losses in the absence of ILS. Figure 3d shows how both individual duplications and losses are labelled: observe that neither requires a label with a specified number of events (like the I label), nor does either need to specify which branch they act on. `reconcILS` places the label L on any node in the implied gene tree that is not represented by a gene copy; for example, generic tip nodes A, C, and B in Figure 3d do not exist in the input gene tree itself, but are labelled with an L. `reconcILS` is able to do this because every duplication event is represented internally by generating a new copy of the species tree below the duplicated node—if losses occur on these implied trees, we can track their timing as well. Note that internal nodes can also be labelled with an L, so that the timing of loss events must be inferred by comparing the state of parent and daughter nodes (i.e. losses on internal branches will lead the descendant internal node and all of its daughter nodes to be labelled with an L). Finally, any node without a label can be inferred to be a speciation node.

3 Evaluation

3.1 Simulations and ETE3

To test the accuracy and speed of `reconcILS` compared to other methods, we used our simulator (`dupcoal`) to generate gene trees evolving under ILS, duplication, and loss. Data were simulated on a species tree with topology (A:1.5,(B:0.5,C:0.5):1); branch lengths are measured in coalescent units. Given this tree, we expect single-copy gene trees to be concordant approximately 75% of the time. Simulated rates of duplication and loss were $\lambda = 0.3$ and $\mu = 0.3$, respectively, except where stated otherwise. The default costs of `reconcILS` were used for all analyses for all methods ($C_D = 1.1$, $C_L = 1$, $C_I = 1$).

As a simple test of the DL reconciliation algorithm implemented in ETE3 (Huerta-Cepas et al. 2016)—which does not allow for ILS—we ran simulations with the above duplication rate, but with the loss rate set to 0 (this allows us to easily see incorrectly inferred losses). As expected (Hahn 2007), in simulated trees with discordance due to ILS but no duplication or loss, ETE3 always inferred 1 duplication and 3 losses (cf. Figure 2c). In cases with discordance due to coalescence and duplication, but no loss (e.g. Figures 1d-g), ETE3 always inferred the correct number of duplications, but with extra incorrect losses. These incorrect inferences were all returned extremely quickly; regardless, in the rest of the results we focus on comparisons between algorithms that do allow for discordance due to coalescence.

3.2 Accuracy of `reconcILS` and DLCpar in reconstructing duplication and ILS

We ran `reconcILS` and DLCpar (Wu et al. 2014) on 1000 gene trees simulated as described above. Overall, the accuracy of both methods was high and similar for ILS (Figure 4a) and duplication (Figure 4b). The correlation (as measured by Spearman's ρ) between simulated and inferred ILS events was 0.719 for `reconcILS` and 0.590 for DLCpar (Figure 4a). It is important to note, however, that these two methods are inferring slightly different kinds of events, which we account for in these results. `reconcILS` infers the number of NNI events needed to reconcile a gene tree; for example, the gene tree in Figure 3a requires two NNI events. The results reported for `reconcILS` therefore compare simulated NNI events vs. inferred events. In contrast, DLCpar infers the number of ILS events needed, regardless of the number of NNI rearrangements that occur within such an event. The tree in Figure 3a only counts as one ILS event using this method. Results for

reported for DLCpar therefore compare simulated vs. inferred ILS events, to ensure that a fair comparison can be made. In both cases the methods are doing well in their respective inferences, but with a slight boost in accuracy for `reconcILS`.

Inferences about duplication were also highly accurate and highly similar between `reconcILS` and DLCpar (correlations of 0.894 and 0.892, respectively). The slight undercounting in both methods is due to cases in which duplication events are immediately followed by losses (the same can happen with ILS events followed by loss). Such events cannot be inferred by any parsimony-based algorithm, and are likely not inferrable by probabilistic methods either (e.g. Rasmussen and Kellis 2012).

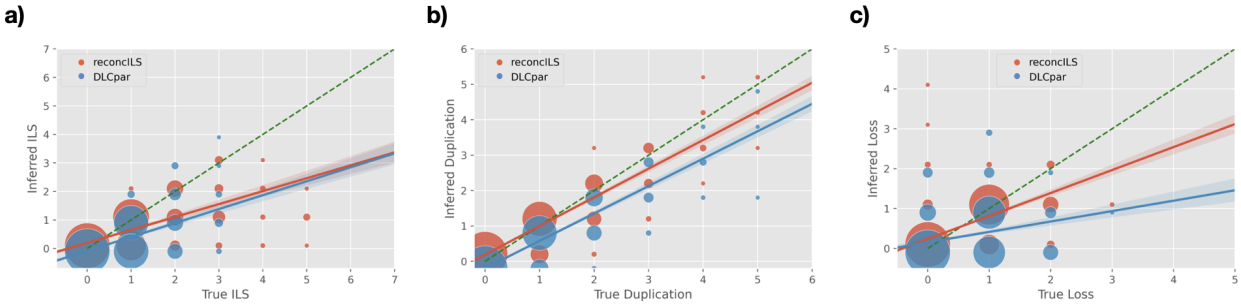

Fig. 4. Accuracy of `reconcILS` and DLCpar. a) Accuracy of inferred ILS events. Orange circles represent results from `reconcILS`. Blue circles represent results from DLCpar. Circle size is proportional to the number of reconciliations with each value, with jitter applied for clarity. Lines represent best-fit regressions (as calculated in Seaborn; Waskom 2021), plus confidence intervals. b) Accuracy of inferred duplication events. c) Accuracy of inferred loss events.

3.3 Accuracy of `reconcILS` and DLCpar in reconstructing loss

There is quite a difference between `reconcILS` and DLCpar in the inference of losses (correlations of 0.729 and 0.334, respectively; Figure 4c). Upon closer inspection, the lower accuracy in DLCpar is due to many loss events of one particular type being missed. Specifically, DLCpar cannot detect losses that occur on either of the branches attaching to the root node of the species tree, as these will have occurred above the root of the resulting gene tree. This means that, for instance, DLCpar does not count the loss that occurred along the A branch in gene trees with the topology (B,C). (Gene trees with only a single tip, i.e. the topology (A), do not run at all in DLCpar, but were not counted here.) We confirmed this behavior on larger trees in DLCpar, as well as in all trees in ETE3. The difference in counting of losses among algorithms arises because `reconcILS` explicitly traverses the species tree, while the other methods do not. This enables our method to track losses that occur above the root of the input gene tree, but below the root of the input species tree.

3.4 Comparison of runtimes

We compared the runtimes of `reconcILS` and DLCpar on the same set of simulated gene trees used above. These comparisons were carried out on an HP Elitebook 840 G3 laptop running Linux Ubuntu 22.04.2 with 16 Gb of memory and four Intel i7 processors. `reconcILS` requires Python 3 or higher to run, while DLCpar requires Python 2.7 (it will not run in Python 3), which was run natively on this computer. We used the "dp" (dynamic programming) option to run DLCpar.

Across all reconciliations, the average runtime for `reconcILS` was 0.066 seconds and for DLCpar was 0.549 seconds. Despite the almost ten-fold increase in runtimes for DLCpar, it showed less variability among gene trees: the standard deviation was 0.037, compared to 0.094 for `reconcILS`. Indeed, two gene trees took longer to reconcile with `reconcILS` than with DLCpar. Therefore, although the current results are very promising with respect to our new software, it is possible that there are areas of parameter space where its runtimes will be more similar to alternative tools.

4 Discussion

We have introduced a new reconciliation algorithm and associated software package (`reconcILS`) that considers discordance due to duplication, loss, and coalescence. To better demonstrate the accuracy of our method, we have also developed a new simulation program (`dupcoal`) that generates gene trees while allowing these processes to interact. Used together, our results demonstrate that `reconcILS` is a fast and accurate approach to gene tree reconciliation, one that offers multiple advantages over alternative approaches, even those that also consider ILS (e.g. DLCpar; Wu et al. 2014). Nevertheless, there are still several caveats to consider, as well as future directions to explore.

Despite our demonstration of good performance in practice, the biggest caveat to our approach is the lack of a theoretical proof that our method finds most parsimonious reconciliations. While such proofs often require strict assumptions—which can be relaxed in a simulation setting—they are very helpful in providing confidence across untested parameter space. Similarly, we have not yet provided an estimate of the time complexity of our algorithm, a measure that can be used as a guideline for expected runtimes across different inputs. Our approach is the first that we know of to use NNI rearrangements in a reconciliation setting, and as such has few predecessors that we can use as benchmarks. Given this, we are still actively working to provide a theoretical underpinning for this novel approach, as well as working to simplify the data structures and memory requirements underlying the `reconcILS` software.

As mentioned above, there is often ambiguity in how to label nodes in gene trees affected by ILS and duplication (e.g. Figures 3b and 3c). This is due in part to how `reconcILS` uses NNI rearrangements—it is not clear which node to pick if a branch is rearranged—and in part to the intrinsic lack of information contained only within tree topologies when coalescence and duplication interact. Nevertheless, labeling gene trees serves multiple important functions, and is often interpreted as biological truth. For example, multiple methods use inferred gene tree labels to identify orthologous genes (genes related by speciation events; Fitch 1970). The general approach taken is to label nodes using reconciliation or similar approaches, treating any node not labeled with a D as a speciation node; genes whose common ancestor is a speciation node are subsequently defined as orthologs. Both standard DL reconciliation and so-called "overlap" methods (e.g. Huerta-Cepas et al. 2007; Emms and Kelly 2019) will consistently label the top-most node flanking a discordant branch with a duplication event. This means that, in cases like the one shown in Figure 3b, these methods will label the wrong node with a D, and subsequently identify the wrong pair of genes as orthologs. (Figure 3c would be labeled correctly; recall that this outcome has an equal probability to the one in Figure 3b in the scenario considered here.) Such cases have been referred to as "unsorted homologs" (Mallo et al. 2014), and will occur any time duplication and coalescence interact, either before or after a speciation event. They can also occur even if only two species are analyzed (e.g. if only species B and C were considered in Figures 3b and 3c), and so are almost certainly more common than the occurrence of pseudoorthologs (a different type of mislabeling) due to duplication and multiple subsequent losses (Smith and Hahn 2022). Although `reconcILS` also currently only labels one node with a D in these cases, in the future we could allow for multiple possible labelings in order to emphasize the inherent ambiguity of some labels.

The algorithm used by `reconcILS` deals at least in part with an additional interaction between duplication, loss, and coalescence: hemiplasy. Hemiplasy is the phenomenon by which a single mutational event on a discordant gene tree will be interpreted as multiple events ("homoplasy") when reconciled to the species tree (Avise and Robinson 2008; Hahn and Nakhleh 2016). In analyses of nucleotide substitutions, generic binary traits, and even continuous traits, hemiplasy requires that mutations occur on a discordant branch of a gene tree (Mendes and Hahn 2016; Guerrero and Hahn 2018; Hibbins et al. 2023). In the context of either duplication or loss, such mutations would have to occur on a discordant branch of the daughter gene tree to lead to "copy-number hemiplasy." Unfortunately, such events are still not accurately inferred by `reconcILS`: we have found that the algorithm consistently proposes an extra loss when hemiplasy occurs (`dupcoal` can track and report simulated hemiplasy events, allowing us to easily make such comparisons). However, there is also a broader definition of copy-number hemiplasy in use: a duplication or loss that does not fix in all descendant lineages (Rasmussen and Kellis 2012; Wu et al. 2014). While this definition encompasses the stricter one given above, it also includes cases such as the one shown in Figure 1. Because the mutation in that figure occurs in the common ancestral population of species B and C, but was only inherited by lineage C, this scenario would be considered to be hemiplasy by the broader definition (although it actually does not involve ILS if the outcome resembles the gene trees in Figure 1c-e, since coalescence is happening in the

same population). The number of each type of event in such cases is accurately inferred using `reconcILS`; DLCpar assumes that this type of hemiplasy does not occur. Note, however, that if the duplication event (the star in Figure 1a) occurred a few generations later—after the split between species B and C, along a tip lineage—the scenario would not be considered hemiplasy by this broader definition, but would still produce the same range of gene trees (i.e. Figure 1c-g). (It would now involve ILS, however.) Regardless of the exact definitions and terms used, it should be clear that duplication, loss, and coalescence can interact to produce a wide array of topologies not considered by other reconciliation approaches.

One relatively straightforward improvement that could be made to `reconcILS` is in the cost of different events. As discussed earlier, it is difficult to determine biologically realistic costs for duplication, loss, and ILS. One approach to determining duplication and loss costs would be to infer the number of such events using an alternative, non-reconciliation approach (e.g. Mendes et al. 2020), setting the cost in `reconcILS` to be inversely related to the inferred number of events of each type. Varying rates of duplication and loss can also be inferred along different branches of the species tree, and so the costs of these events could vary commensurately along the tree. Similarly, we can envision a simple way to scale the costs of NNI with the length of species tree branches: this is because the probability of discordance due to ILS is inversely proportional to internal branch lengths on a species tree (Hudson 1983). We could therefore make NNI events on longer branches of the species tree more costly, which would properly weight the relative probability of such events locally on the tree.

There are also several extensions that could be made to `dupcoal`, our new simulation engine. Currently, `dupcoal` draws a daughter gene tree from the multispecies coalescent (MSC) at random: this implicitly assumes that the parent and daughter trees are completely unlinked. It should be straightforward, however, to draw the daughter tree at a given recombination distance from the parent tree, ranging from fully linked to fully unlinked. In addition, though the software currently draws gene trees from the MSC, we could also include introgression as a source of discordance by drawing trees from the multispecies network coalescent (Meng and Kubatko 2009; Yu et al. 2011). In this case, users would simply input a species network to simulate from, rather than a species tree. The interaction between duplication, loss, and introgression is currently understudied, though possibly biologically very important (e.g. Hsieh et al. 2019). We should also note the differences between the current version of `dupcoal` and alternative simulators. As mentioned previously, SimPhy (Mallo et al. 2016) does not simulate hemiplasy of any kind, as it uses the same three-tree model of Rasmussen and Kellis (2012) and Wu et al. (2014). However, this method does include gene transfers, which are not simulated by `dupcoal`. The MLMSC simulator of Li et al. (2021) simulates both hemiplasy and transfers. `dupcoal` is therefore most similar to this method, with two small differences. First, `dupcoal` does not change the rate of duplication and loss above speciation events to account for ancestral polymorphism (cf. Gillespie and Langley 1979); instead, users can simply extend all nodes $2N$ generations into the past to do so (where N is the effective population size). Second, `dupcoal` does not account for the possibility of further events occurring while duplications and losses are still polymorphic in a population. In order to streamline the simulation process, we have chosen to ignore this squared term.

Finally, we envision multiple ways in which the `reconcILS` algorithm could be used in conjunction with other methods that use reconciliation. For instance, ASTRAL-Pro (Zhang et al. 2020) infers a species tree from gene trees containing duplicates by first labeling nodes in each gene tree via reconciliation. The authors of this software rightfully recognize the problems with labeling discussed here, but nevertheless their reconciliation step might benefit from methods that include ILS. Likewise, GRAMPA (Thomas et al. 2017) allows for reconciliation of polyploidy events, but does not include coalescence as a source of discordance; their algorithm could easily accommodate this extension. There are also multiple approaches that can be used to "correct" gene trees via reconciliation (e.g. El-Mabrouk and Ouangraoua 2017; Christensen et al. 2020). These and many other tasks might benefit from approaches that model duplication, loss, and coalescence in a single framework.

Acknowledgements. We thank Yao-ban Chan for many helpful conversations about the MLMSC model, and Ben Fulton for tips on best programming practices. This work was supported by NSF grant DEB-1936187 to MWH.

5 References

Avise JC, Robinson TJ. 2008. Hemiplasy: A new term in the lexicon of phylogenetics. Systematic Biology 57:503-507.

Bansal MS, Alm EJ, Kellis M. 2012. Efficient algorithms for the reconciliation problem with gene duplication, horizontal transfer and loss. Bioinformatics 28:i283-i291.

Boussau B, Scornavacca C. 2020. Reconciling gene trees with species trees. In: Scornavacca C, Delsuc F, Galtier N, editors. Phylogenetics in the Genomic Era: No commercial publisher — Authors open access book. p. 3.2:1-3.2:23.

Chan Y-b, Ranwez V, Scornavacca C. 2017. Inferring incomplete lineage sorting, duplications, transfers and losses with reconciliations. Journal of Theoretical Biology 432:1-13.

Chen K, Durand D, Farach-Colton M. 2000. NOTUNG: A program for dating gene duplications and optimizing gene family trees. Journal of Computational Biology 7:429-447.

Christensen S, Molloy EK, Vachaspati P, Yammanuru A, Warnow T. 2020. Non-parametric correction of estimated gene trees using TRACTION. Algorithms for Molecular Biology 15:1.

Du H, Ong YS, Knittel M, Mawhorter R, Liu N, Gross G, Tojo R, Libeskind-Hadas R, Wu Y-C. 2021. Multiple optimal reconciliations under the duplication-loss-coalescence model. IEEE/ACM Transactions on Computational Biology and Bioinformatics 18:2144-2156.

Duchemin W, Gence G, Arigon Chifolleau A-M, Arvestad L, Bansal MS, Berry V, Boussau B, Chevenet F, Comte N, Davín AA, et al. 2018. RecPhyloXML: a format for reconciled gene trees. Bioinformatics 34:3646-3652.

Durand D, Halldórsson BV, Vernot B. 2006. A hybrid micro-macroevolutionary approach to gene tree reconstruction. Journal of Computational Biology 13:320-335.

El-Mabrouk N, Ouangraoua A. 2017. A general framework for gene tree correction based on duplication-loss reconciliation. 17th International Workshop on Algorithms in Bioinformatics (WABI 2017). p. 1-14.

Emms DM, Kelly S. 2019. OrthoFinder: phylogenetic orthology inference for comparative genomics. Genome Biology 20:238.

Fitch WM. 1970. Distinguishing homologous from analogous proteins. Systematic Zoology 19:99-113.

Gillespie JH, Langley CH. 1979. Are evolutionary rates really variable? Journal of Molecular Evolution 13:27-34.

Goodman M, Czelusniak J, Moore GW, Romero-Herrera AE, Matsuda G. 1979. Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences. Systematic Zoology 28:132-163.

Guerrero RF, Hahn MW. 2018. Quantifying the risk of hemiplasy in phylogenetic inference. Proceedings of the National Academy of Sciences 115:12787-12792.

Guigo R, Muchnik I, Smith TF. 1996. Reconstruction of ancient molecular phylogeny. Molecular Phylogenetics and Evolution 6:189-213.

Hahn MW. 2007. Bias in phylogenetic tree reconciliation methods: Implications for vertebrate genome evolution. Genome Biology 8:R141.

Hahn MW, Nakhleh L. 2016. Irrational exuberance for resolved species trees. Evolution 70:7-17.

Harris CR, Millman KJ, Van Der Walt SJ, Gommers R, Virtanen P, Cournapeau D, Wieser E, Taylor J, Berg S, Smith NJ, et al. 2020. Array programming with NumPy. Nature 585:357-362.

Hibbins MS, Breithaupt LC, Hahn MW. 2023. Phylogenomic comparative methods: Accurate evolutionary inferences in the presence of gene tree discordance. Proceedings of the National Academy of Sciences 120:e2220389120.

Hsieh P, Vollger MR, Dang V, Porubsky D, Baker C, Cantsilieris S, Hoekzema K, Lewis AP, Munson KM, Sorensen M, et al. 2019. Adaptive archaic introgression of copy number variants and the discovery of previously unknown human genes. Science 366:eaax2083.

Hudson RR. 1983. Testing the constant-rate neutral allele model with protein sequence data. Evolution 37:203-217. Huerta-Cepas J, Dopazo H, Dopazo J, Gabaldón T. 2007. The human phylome. Genome Biology 8:R109.

Huerta-Cepas J, Serra F, Bork P. 2016. ETE 3: Reconstruction, Analysis, and Visualization of Phylogenomic Data. Molecular Biology and Evolution 33:1635-1638.

Jacox E, Chauve C, Szöllősi GJ, Ponty Y, Scornavacca C. 2016. ecceTERA: comprehensive gene tree-species tree reconciliation using parsimony. Bioinformatics 32:2056-2058.

Kingman JFC. 1982. The coalescent. Stochastic Processes and their Applications 13:235-248.

Li Q, Scornavacca C, Galtier N, Chan Y-b. 2020. The multilocus multispecies coalescent: A flexible new model of gene family evolution. Systematic Biology 70:822-837.

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

Maddison WP. 1997. Gene trees in species trees. Systematic Biology 46:523-536.

Mallo D, De Oliveira Martins L, Posada D. 2014. Unsorted homology within locus and species trees. Systematic Biology 63:988-992.

Mallo D, De Oliveira Martins L, Posada D. 2016. SimPhy: Phylogenomic simulation of gene, locus, and species trees. Systematic Biology 65:334-344.

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

Mendes FK, Hahn MW. 2016. Gene tree discordance causes apparent substitution rate variation. Systematic Biology 65:711-721.

Mendes FK, Vanderpool D, Fulton B, Hahn MW. 2021. CAFE 5 models variation in evolutionary rates among gene families. Bioinformatics 36:5516-5518.

Meng C, Kubatko LS. 2009. Detecting hybrid speciation in the presence of incomplete lineage sorting using gene tree incongruence: A model. Theoretical Population Biology 75:35-45.

Morel B, Kozlov AM, Stamatakis A, Szöllősi GJ. 2020. GeneRax: A tool for species-tree-aware maximum likelihood-based gene family tree inference under gene duplication, transfer, and loss. Molecular Biology and Evolution 37:2763-2774.

Page RDM. 1994. Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas. Systematic Biology 43:58-77.

Page RDM, Charleston MA. 1997. From gene to organismal phylogeny: Reconciled trees and the gene tree/species tree problem. Molecular Phylogenetics and Evolution 7:231-240.

Rasmussen MD, Kellis M. 2012. Unified modeling of gene duplication, loss, and coalescence using a locus tree. Genome Research 22:755-765.

Schrider DR, Hahn MW. 2010. Lower linkage disequilibrium at CNVs is due to both recurrent mutation and transposing duplications. Molecular Biology and Evolution 27:103-111.

Smith ML, Hahn MW. 2022. The frequency and topology of pseudoorthologs. Systematic Biology 71:649-659.

Stolzer M, Lai H, Xu M, Sathaye D, Vernot B, Durand D. 2012. Inferring duplications, losses, transfers and incomplete lineage sorting with nonbinary species trees. Bioinformatics 28:i409-i415.

Stull GW, Qu X-J, Parins-Fukuchi C, Yang Y-Y, Yang J-B, Yang Z-Y, Hu Y, Ma H, Soltis PS, Soltis DE, et al. 2021. Gene duplications and phylogenomic conflict underlie major pulses of phenotypic evolution in gymnosperms. Nature Plants 7:1015-1025.

Sukumaran J, Holder MT. 2010. DendroPy: a Python library for phylogenetic computing. Bioinformatics 26:1569-1571.

Szöllősi GJ, Rosikiewicz W, Boussau B, Tannier E, Daubin V. 2013. Efficient exploration of the space of reconciled gene trees. Systematic Biology 62:901-912.

Tajima F. 1983. Evolutionary relationships of DNA sequences in finite populations. Genetics 105:437-460.

Thomas GWC, Ather SH, Hahn MW. 2017. Gene-tree reconciliation with MUL-Trees to resolve polyploidy events. Systematic Biology 66:1007-1018.

Vernot B, Stolzer M, Goldman A, Durand D. 2008. Reconciliation with non-binary species trees. Journal of Computational Biology 15:981-1006.

Waskom M. 2021. seaborn: statistical data visualization. Journal of Open Source Software 6:3021.

Wu Y-C, Rasmussen MD, Bansal MS, Kellis M. 2014. Most parsimonious reconciliation in the presence of gene duplication, loss, and deep coalescence using labeled coalescent trees. Genome Research 24:475-486.

Yu Y, Than C, Degnan JH, Nakhleh L. 2011. Coalescent histories on phylogenetic networks and detection of hybridization despite incomplete lineage sorting. Systematic Biology 60:138-149.

Zhang C, Scornavacca C, Molloy EK, Mirarab S. 2020. ASTRAL-Pro: Quartet-based species-tree inference despite paralogy. Molecular Biology and Evolution 37:3292-3307.

Zmasek CM, Eddy SR. 2001. A simple algorithm to infer gene duplication and speciation events on a gene tree. Bioinformatics 17:821-828.