Inferring incomplete lineage sorting, duplications, transfers and losses with reconciliations

Yao-Ban Chan, Vincent Ranwez, Celine Scornavacca

Inferring incomplete lineage sorting, duplications, transfers and losses with reconciliations

Yao-ban Chan[a,*], Vincent Ranwez[b], Céline Scornavacca[c,d]

[a]*School of Mathematics and Statistics, The University of Melbourne, Melbourne, Australia*
[b]*Montpellier SupAgro, UMR AGAP, F-34060 Montpellier, France*
[c]*Institut des Sciences de l'Evolution (ISEM), Université de Montpellier, Montpellier, France*
[d]*Institut de Biologie Computationnelle (IBC), Montpellier, France*

Abstract

Gene trees and species trees can be discordant due to several processes. Standard models of reconciliations consider macro-evolutionary events at the gene level: duplications, losses and transfers of genes. However, another common source of gene tree-species tree discordance is incomplete lineage sorting (ILS), whereby gene divergences corresponding to speciations occur "out of order". However, ILS is seldom considered in reconciliation models. In this paper, we devise a unified formal $\mathbb{IDTL}$ reconciliation model which includes all the abovementioned processes. We show how to properly cost ILS under this model, and then give a fixed-parameter tractable (FPT) algorithm which calculates the most parsimonious $\mathbb{IDTL}$ reconciliation, with guaranteed time-consistency of transfer events. Provided that the number of branches in contiguous regions of the species tree in which ILS is allowed is bounded by a constant, this algorithm is linear in the number of genes and quadratic in the number of species. This provides a formal foundation to the inference of ILS in a reconciliation framework.

Keywords: reconciliation, gene duplication, gene transfer, incomplete lineage sorting, parsimony

*Corresponding author
Email address: yaoban@unimelb.edu.au (Yao-ban Chan)

1. Introduction

Macro-evolutionary events at the species level (i.e., speciation) impact the genomes of the individuals belonging to the involved species. Hence, the evolutionary history of a group of species strongly influences the evolutionary history of its genes. However, even though species evolution strongly shapes each gene history, it does not fully determine it, and the discrepancy between the two histories provides clues about gene-specific evolutionary events such as gene duplication, gene transfer and gene loss.

Many methods have been proposed to *reconcile* the (inferred) evolutionary history of a gene (depicted as a gene tree) with that of the corresponding species (depicted as a species tree), using gene-specific events. In general, these methods fall into two paradigms: probabilistic methods (e.g., [1, 20]), which find the most likely reconciliation under a statistical model of evolution, and parsimony-based methods (e.g., [8, 4, 7]), which minimise the number (or total cost under a penalisation scheme) of the gene-specific events. In this paper, we concentrate on the latter paradigm for reasons of efficiency and scalability.

Gene transfers are particularly difficult to take into account due to the time constraints they induce [6]. Thus, reconciliation methods differ mainly by the way they handle transfer events. Some simply ignore them, relying on the fact that transfers almost never occur in a large part of the animal kingdom [29, 27]. Some search for optimal reconciliations without considering the time constraints induced by transfers and, if needed, they either modify the inferred solution to satisfy these constraints — with no guarantee of global optimality — or they check for *time-consistency* of the transfers *a posteriori* and return an optimal solution that is time-consistent, but only if any exists [16, 23]. Finally, some fully handle transfer events and the associated time constraints in polynomial time, but require that the dates of speciations are provided [4, 7, 25].

In addition to discrepancies caused by duplications, transfers and losses, an additional source of discordance between gene and species trees arises from incomplete lineage sorting [13]. In theory, incomplete lineage sorting is not a true "gene event" such as a duplication or a transfer, since nothing "happens" to the gene during incomplete lineage sorting. Still, it is a phenomenon that can lead to a gene tree differing from the species tree containing it. In order to explain how ILS affects gene histories, we recall how a speciation acts on populations.

A speciation can be seen as the division of a population into (two) sub-populations that

will evolve separately and hence *fix* potentially different gene variants (alleles) so that those alleles are somehow *sorted* from the originating population in the two sub-populations that eventually become the two new species. For instance, in Figure 1, the ancestral population giving rise to species B and C, prior to the speciation, contains blue and green alleles for the considered locus; the speciation leads to two populations, one containing only blue alleles (species B) and the other only green alleles (species C).

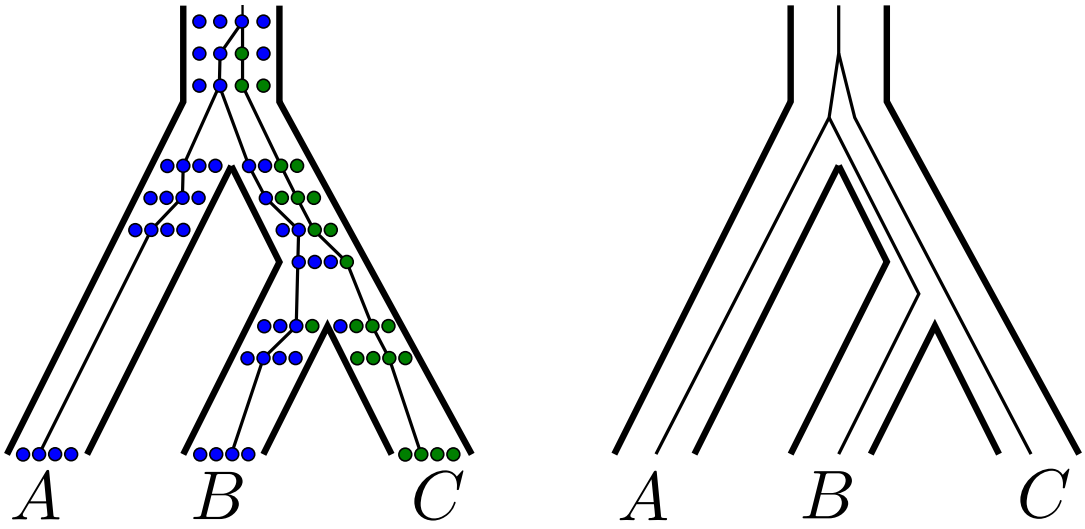
Such a "sorting" is not instantaneous, and if another speciation event occurs soon after the first one, a locus may be *incompletely sorted* at the time of the second speciation. In such a case, we can observe — in the two new species originating from the second speciation — individuals that carry genes whose most recent common ancestor predates the first speciation event. This results in the appearance of the two speciation events being "swapped" in the gene tree, as shown in Figure 1.

The likelihood of an ILS is mainly related to the ancestral effective population size, which can be hard to estimate, and the time elapsed between the two or more successive speciation events, corresponding to the branch length of a dated species tree. However, in theory, given any species tree, all possible gene tree topologies where each species has exactly one copy of the gene can be explained by ILS alone.

The existence of ILS as a reason for discordance between gene and species trees has been known for some time, and is often used in species tree inference from gene trees [12, 5]. In these cases, the multispecies coalescent, arising from Kingman's coalescent in population genetics [9, 10], provides a statistical model under which the likelihood of ILS can be evaluated.

Inference of ILS via reconciliation is less common. In a seminal paper, Maddison [13] suggested the parsimonious criterion of minimising deep coalescences (MDC) for reconciliation, where the total number of "extra lineages" in all branches is minimised. An algorithm to solve this problem was constructed by Than and Nakhleh [24], and extended for the presence of hybridization in [28].

These papers did not consider macro-events such as duplications and losses, and indeed very few papers attempt to combine both ILS and macro-events in a unified framework. Combining these events is relevant from a biological perspective, as recent studies have shown that ILS and gene introgression (although not specifically LGT) can both occur in the history of a species [11, 14, 15]. More generally, with the increasing availability of data and efficiency



(a) Population view (b) Reconciliation representation

Figure 1: Impact of incomplete lineage sorting on simple populations of 4 haploid individuals. The originating population contains a single blue allele for the considered gene. First, a mutation leads to a new green allele at this locus, then a first speciation takes place, rapidly followed by a second one. As the blue and green alleles still co-exist when the second speciation takes place, both alleles still have a chance to be fixed in the resulting child species B and C. For these species, the history of this gene will hence differ from the species history due to ILS.

of algorithms, the number of species that can be considered in one tree is increasing rapidly. In consequence, even processes which mainly occur in different parts of the species tree must now be considered together, in order to capture all possible processes.

Of interest are a series of papers by Wu, Rasmussen and Kellis [21, 27], who model ILS together with duplications and losses using a coalescent model. They devised both a probabilistic algorithm (which was found to be very slow in practice), and a parsimony-based algorithm based on dynamic programming. However, their model does not include transfers and so does not need to consider the associated issues of time-consistency.

Another series of papers by Stolzer *et al.* [26, 23] formulated a full model with duplications, transfers, losses and ILS, and devised an algorithm to calculate the most parsimonious reconciliation for this model. Their algorithm starts by contracting short branches of the species tree into multifurcating nodes (polytomies). These are considered the only places where ILS can occur. Since ILS is not penalised in their model, discordance explainable by ILS is always associated to ILS. The remaining discrepancies are then explained by duplications, losses and transfers. However, their treatment of transfers does not guarantee a time-consistent reconciliation; this must be checked *a posteriori* and thus their algorithm may fail to return a

solution.

The precise complexity of the problem of finding an optimal reconciliation in the full model with duplication, transfers, losses and ILS is unclear. Firstly, there can be slight variations in the formulation of the model which may have an unknown and potentially drastic impact on the complexity. Some information may be gleaned from complexity studies of various special cases, which are necessarily no more complex than the full model. It is known that the optimal $\mathbb{DTL}$ reconciliation problem is NP-hard when the species tree is undated [18], but is polynomial-time if the species tree is dated [7]. Likewise, it was proven recently [2] that finding an optimal reconciliation in the duplication-loss-ILS model is also NP-hard, although this complexity does not change if the species tree is dated or not. As the full model contains this model as a special case (with transfers disallowed), it is likely that the problem we study is also NP-hard, but as the two formulations do not correspond exactly, we cannot say this with complete certainty.

In this paper, we formalise a model of reconciliation accounting for duplications, transfers, losses and ILS. This $\mathbb{IDTL}$ model is based on the standard $\mathbb{DTL}$ model, formalised in [19], with extensions to account for ILS. We then present an algorithm that calculates the most parsimonious reconciliation for our model, and prove its correctness. This algorithm ensures time-consistency through means of subdividing the species tree, as was done in [7], and thus always returns an optimal time-consistent reconciliation. A detailed comparison with the models and algorithms of [27] and [23] is also provided.

2. Preliminaries

Given a tree T, its node set, branches, and leaf set are respectively denoted $V(T)$, $E(T)$, $L(T)$. The label of each leaf u is a name (typically an extant gene or species) or an identifier associated with that leaf, denoted by $\mathcal{L}(u)$, while the set of labels of leaves of T is denoted by $\mathcal{L}(T)$.

If T is rooted, we denote its root by $r(T)$. Given a node $u \in V(T)$, we denote its parent by u_p, and the subtree of T rooted at u by T_u. Given two nodes u and v of T, we write $u \leq_T v$ ($u <_T v$) if and only if v is on the unique path from $r(T)$ to u (and $u \neq v$); in such a case, u is said to be a (strict) descendant of v. The *height* of T, denoted $h(T)$, is the length, in nodes, of the longest path from $r(T)$ to any leaf of T. From now on, unless otherwise specified, we

assume that all trees are rooted.

If a node in a tree T has more than two children, we call it a *polytomy*. If $u \in V(T)$ is not polytomous, we denote its children by $\{u_l, u_r\}$; if u has just one child, u_r is understood to be undefined. In this paper all trees are considered as unordered, so u_l and u_r are arbitrarily assigned.

We define a *clade* of T as a set of leaves of T. The clade *generated* by the node u, denoted $C(u)$, is the set $L(T_u)$. We define $\mathcal{C}(T)$ as the set of all clades generated by nodes in T; for a set $\mathcal{T}$ of trees, $\mathcal{C}(\mathcal{T}) = \cup_{T \in \mathcal{T}} \mathcal{C}(T)$. The LCA of a clade is the internal node which is the lowest common ancestor of the elements of the clade.

If u is a binary internal node, we define the *tripartition* generated by u, denoted by $\Pi(u)$, as the clade triplet $(C(u), C(u_l), C(u_r))$. The latter two clades of a tripartition are a partition of the first one — called *the parent clade* — since, for any internal node u, $C(u) = C(u_l) \cup C(u_r)$ and $C(u_l) \cap C(u_r) = \emptyset$. If u is an internal node with a single child, it generates the tripartition $(C(u), C(u_l), \emptyset)$, while leaf nodes generate no tripartitions. We define $\Pi(T)$ as the set of all tripartitions generated by nodes in T; for a set $\mathcal{T}$ of trees, $\Pi(\mathcal{T}) = \cup_{t \in \mathcal{T}} \Pi(T)$.

A tree T is said to be *dated* when there exists a *time function* $\theta_T : V(T) \to \mathbb{R}^+$ that associates each of its nodes with a non-negative value so that, for any two nodes $x, y \in V(T)$, if $y < x$ then $\theta_T(y) < \theta_T(x)$. Moreover, $\theta_S(x) = 0 \ \forall x \in L(T)$.

If a tree T is dated, then each clade it generates can be associated with the time of the generating node. We therefore say that the tree generates *dated clades*, denoted by tuples (C, t). An internal node u generates the dated clade $(C(u), \theta_T(u))$, and *dated tripartitions* are generated similarly. We denote by $\mathcal{C}_\theta(T)$ and $\Pi_\theta(T)$ respectively the sets of dated clades and tripartitions generated by T.

We define the *subdivision* of a dated binary tree T with time function θ_T to be the unary-binary tree T' obtained from T by adding a new unary node y on each branch $(x_p, x) \in E(T)$ such that there exists $z \in V(T)$ with $\theta_T(x) < \theta_T(z) < \theta_T(x_p)$; the time $\theta_{T'}(y)$ is set to $\theta_T(z)$ (for all nodes u already in T, we have $\theta_{T'}(u) := \theta_T(u)$). These unary nodes are called *artificial* nodes of T'. It is understood that the "+1" and "-1" operators, when applied to a time t, indicate the lowest time in T' greater than t, and the highest time in T' lower than t, respectively.

We define a *gene tree* G as a tree where each leaf represents an extant gene. Similarly,

a *species tree* S is defined as a tree in which each leaf represents a distinct extant species. Each extant gene is associated to its host species by a function $s : \mathcal{L}(G) \to \mathcal{L}(S)$, called the *species labelling* of G. Note that s does not have to be either injective (several genes can be contained in the same species due to duplication or transfers) or surjective (some species may not contain any copy of the gene in question). The set of species labels of the leaves of G is denoted $\mathcal{S}(G)$. In this paper, unless otherwise specified, we assume that gene and species trees are rooted and binary. We will generally require that the species tree be dated, but the gene tree need not be.

3. The model

In this section, we start by extending the $\mathbb{DTL}$ model formalised in [19] (an efficient algorithm for this model was presented in [7] for a single rooted gene tree, and in [22] for several, potentially unrooted, gene trees) to include incomplete lineage sorting of speciations, to give an $\mathbb{IDTL}$ model. Then, we associate a cost to each ILS occurrence and present a scoring scheme for $\mathbb{IDTL}$ reconciliations. The algorithm to compute a most parsimonious reconciliation under this costing scheme will be given in the next section.

Firstly, we discuss (informally) how we model the events, to ease the understanding of the formal definition (Definition 3). To construct a reconciliation, we map each gene tree node to a sequence of dated clades of the species tree. We can consider each dated clade to represent two things: a node in the subdivided species tree (essentially the "location" of the clade, including its time), and a set of extant species into which the gene lineage will eventually be "sorted" (descend), barring further events.

For example, consider Figure 2. Here, a simple $\mathbb{I}$ event (identical to that shown in Figure 1) causes the divergence between the genes in species B and C to occur before the speciation at time 2. The reconciliation is as follows: the root gene is mapped to the dated clade $(\{A, B, C\}, 2)$ and, after the initial $\mathbb{I}$ divergence, its descendants are mapped to $(\{A, B\}, 2)$ and $(C, 2)$, meaning that they will eventually be "sorted" into species A and B and species C respectively, at the appropriate time. This is considered a valid mapping even though neither of these two clades correspond to nodes of the species tree (or its subdivision). When these lineages go forward in time, the clades to which they are mapped descend in the species tree, and, eventually, will correspond to species tree nodes (here, at $(A, 1), (B, 0)$ and $(C, 0)$). At

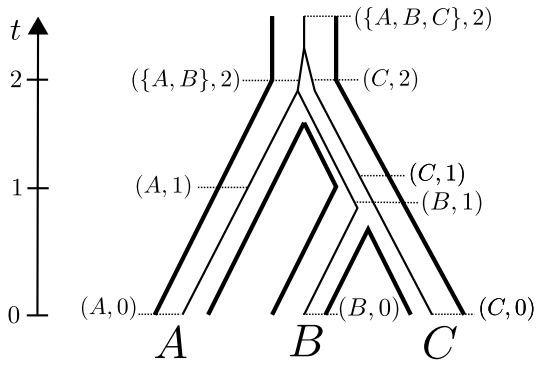


Figure 2: The thicker lines represent the species tree, while the thinner ones depict the gene tree. Dated clades represent the location and time of the gene lineage and also the set of species it will be sorted into. A, B, C are abbreviations for $\{A\}$, $\{B\}$, $\{C\}$ respectively.

this point, the ILS is considered as resolved.

We note that in the absence of ILS, the set of species that a gene lineage is "sorted" into is simply the descendants of the internal species node in which it resides. In this case, the only possible clades that a gene can be mapped to are the clades generated by the nodes of S'. These are clearly in bijection with the nodes of S' themselves. Therefore (for no ILS only) we can directly compare our model and previous models of reconciliations [7, 19] which map a gene to a sequence of nodes of S'. It is not too difficult to see that they are exactly equivalent in this case.

To model events, we consider the effect of each event on the two objects represented by a dated clade (C, t), namely the leaf set (i.e., the set of descendant species) C and the time t. We first consider their impact on the leaf set C. In some situations, C may be partitioned into two subsets; we say this partitioning is *in accordance* with the species tree if it can be ascribed to a speciation. In other words, if C is partitioned into $C = C_1 \cup C_2$, then this partitioning is in accordance with the species tree if there exists a node $x \in V(S)$ such that $C \subseteq C(x)$, $C_1 \subseteq C(x_l)$, and $C_2 \subseteq C(x_r)$ (or vice versa with x_l and x_r interchanged). Then the events have the following impact on the leaf set (the examples in parentheses refer to the species tree in Figure 2, with the caveat that C represents the species named C, and not a clade):

- Speciations ($\mathbb{S}$) will create two gene lineages partitioning C into two subsets, in accordance with the species tree (e.g., $\{A, B, C\}$ can be partitioned into $\{A\}$ and $\{B, C\}$);
- Incomplete lineage sorting ($\mathbb{I}$) will create two gene lineages partitioning C into two

subsets, in a way that is not in accordance with the species tree (e.g., $\{A, B, C\}$ can be partitioned into $\{A, B\}$ and $\{C\}$);

- Duplications ($\mathbb{D}$) will create two gene lineages, each having the same set of descendant species as the original lineage;
- Transfers ($\mathbb{T}$) will create two gene lineages, one having the same set of descendant species as the original clade, and the other corresponding to a node of the species tree at time t (e.g., at $t = 1$, $\{B, C\}$ is a possible choice since $(\{B, C\}, 1)$ exists in the species tree);
- Speciation-losses ($\mathbb{SL}$) will result in a gene with a leaf set which is a subset of C, in accordance with the species tree (e.g., $\{A, B, C\}$ can result in $\{B, C\}$ with $\{A\}$ lost);
- ILS-losses ($\mathbb{IL}$) will result in a gene with a leaf set which is a subset of C, in a way that is not in accordance with the species tree (e.g., $\{A, B, C\}$ can result in $\{A, B\}$ with $\{C\}$ lost);
- Transfer-losses ($\mathbb{TL}$) will result in a gene corresponding to a node of the species tree at time t;
- Null events ($\varnothing$) will result in a gene having the same set of descendant species as the original clade;
- $\mathbb{C}$ events map an extant gene to an extant species containing the gene.

Note that we do not consider losses as separate events. Since it is impossible to distinguish between a single loss and a subtree whose leaves are all lost, we only consider losses as part of atomic $\mathbb{SL}$, $\mathbb{IL}$ and $\mathbb{TL}$ events.

The impact on the time of the events listed above is simple: the gene lineages created by $\mathbb{S}$, $\mathbb{SL}$ and $\varnothing$ events have time $t - 1$, while in all other cases the new lineages have time t.

Before giving a formal definition of our model, we need to define the set of all possible dated clades of a dated tree.

Definition 1. *Consider a dated species tree (S, θ_S). The* expanded clade set *of (S, θ_S), denoted $\mathcal{C}'_\theta(S)$, is the set of all dated clades (C, t) such that $C \subseteq \mathcal{L}(S)$, $t \geq \theta_S(LCA(C))$, and there exists some node $x \in V(S)$ with $\theta_S(x) = t$.*

For example, for the dated species tree in Figure 2, we have $\mathcal{C}'_\theta(S) = (\{A\}, 0), (\{A\}, 1), (\{A\}, 2), (\{B\}, 0), (\{B\}, 1), (\{B\}, 2), (\{C\}, 0), (\{C\}, 1), (\{C\}, 2), (\{B, C\}, 1), (\{B, C\}, 2), (\{A, B\}, 2), (\{A, C\}, 2), (\{A, B, C\}, 2)$. Note that $\mathcal{C}'_\theta(S)$ equates to the set $\mathcal{C}_\theta(S')$ (the set of all dated clades generated by S') augmented with all clades that are possible due to ILS.

To aid interpretability, we now formalise the clade-to-node conversion:

Definition 2. *We define $n : \mathcal{C}'_\theta(S) \to V(S')$ to be the function where $n(C, t) = x$ if:*

- $x \geq LCA(C)$;
- $\theta_S(x) = t$.

It is easy to see that this mapping is well-defined: all nodes have only one ancestor that exists at a given time, so $n(\cdot)$ is unique, and, from the definition of $\mathcal{C}'_\theta(S)$, its value always exists. For example, for the species tree S depicted in Figure 2, we have that $n(\{A, B\}, 2)$ is equal to $r(S)$.

We are now ready to formally define a reconciliation, extending Definition 25 of [19]. As in previous models, a reconciliation can be thought of as "drawing a gene tree inside a species tree"; each branch of the gene tree forms a lineage which resides in the species tree, which must follow the species tree (i.e., descend into that species' descendants), but may also be affected by gene-specific events ($\mathbb{D}$, $\mathbb{T}$, $\mathbb{L}$), and ILS.

Definition 3 (Reconciliation)**.** *Consider a gene tree G and a dated species tree (S, θ_S). Let $\alpha : V(G) \to \cup_{i=1}^{\infty}[\mathcal{C}'_\theta(S)]^i$ be a function which maps each node of G to an ordered sequence of dated clades in $\mathcal{C}'_\theta(S)$ of length at least 1. Let $\alpha_i(u)$ denote the ith element of $\alpha(u)$. Then α is a* reconciliation *between G and (S, θ_S) if and only if exactly one of the following mutually exclusive cases occurs for each element $\alpha_i(u)$ (with $(C, t) := \alpha_i(u)$ and $x := n(C, t)$ in the following):*

- *$\alpha_i(u)$ is the last element of $\alpha(u)$ and exactly one of the cases below is true:*
  1. *$x \in L(S')$, $u \in L(G)$, and $\mathcal{L}(x) = s(\mathcal{L}(u))$;* ($\mathbb{C}$ *event*)
  2. *x is not artificial and $\{\alpha_1(u_l), \alpha_1(u_r)\} = \{(C \cap C(x_l), t-1), (C \cap C(x_r), t-1)\}$, with $C \cap C(x_l), C \cap C(x_r) \neq \emptyset$;* ($\mathbb{S}$ *event*)
  3. *$\alpha_1(u_l) = \alpha_1(u_r) = (C, t)$;* ($\mathbb{D}$ *event*)

4. $\alpha_1(u_l) = (C, t)$ *and* $\alpha_1(u_r) = (C(y), t)$*, where* $y \in V(S')$ *has* $\theta_S(y) = t$ *and* $y \neq x$*;* ($\mathbb{T}$ *event*)

5. $\{\alpha_1(u_l), \alpha_1(u_r)\} = \{(C_1, t), (C_2, t)\}$*, where* $C = C_1 \cup C_2$ *and* $C_1, C_2 \neq \emptyset$ *and* $\nexists y \in V(S')$ *with* $\{C_1, C_2\} = \{C \cap C(y_l), C \cap C(y_r)\}$*;* ($\mathbb{I}$ *event*)

- *otherwise, exactly one of the cases below is true:*

1. x *is not artificial and* $\alpha_{i+1}(u) \in \{(C \cap C(x_l), t-1), (C \cap C(x_r), t-1)\}$*, with* $C \cap C(x_l), C \cap C(x_r) \neq \emptyset$*;* ($\mathbb{SL}$ *event*)

2. $\alpha_{i+1}(u) = (C(y), t)$*, where* $y \in V(S')$ *has* $\theta_S(y) = t$ *and* $y \neq x$*;* ($\mathbb{TL}$ *event*)

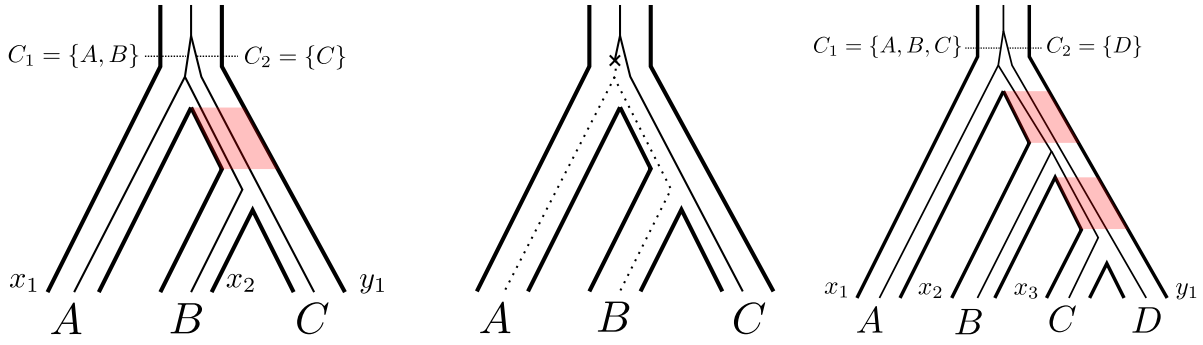
3. $\alpha_{i+1}(u) = (C_1, t)$*, where* $C_1 \subset C$ *and* $\nexists y \in V(S')$ *with* $\{C_1, C \setminus C_1\} = \{C \cap C(y_l), C \cap C(y_r)\}$*;* ($\mathbb{IL}$ *event*)

4. $\alpha_{i+1}(u) = (C, t-1)$*, where one of* $C \cap C(x_l), C \cap C(x_r)$ *is* $\emptyset$*, and if* $u = r(G)$*, then* $i \neq 1$. ($\varnothing$ *event*)

Remarks:.

1. We will sometimes write an event as "$A \to B, C$", where A, B and C are dated clades; this means that for the affected gene v, we have $\alpha_\ell(v) = A$ (where $\ell := |\alpha(v)|$) and $\{\alpha_1(v_l), \alpha_1(v_r)\} = \{B, C\}$.
2. When considering transfer targets, we only consider clades that are generated by S' and not all the clades in the expanded clade set. The reason for this is that we do not allow transfers into a species that fix in some descendants of the species but not in others. In [27], this is referred to (albeit in the context of duplication) as *hemiplasy* and is also not allowed. Thus, we assume that if a transfer occurs, it is fixed immediately in the recipient (if subsequent children do not contain the gene, it must be due to a further loss event). On the other hand, we do allow clades not generated by S' (i.e., incompletely sorted alleles) to be sources of transfers.
3. We allow the root of the gene tree to be mapped initially to any node of the species tree; we do not force it to be mapped to the root of the species tree.

In order to calculate a most parsimonious reconciliation, we must now define the cost of a reconciliation. It is straightforward to cost events not involving ILS: we set δ, τ and λ to be the cost of a duplication, a transfer and a loss respectively. Then $\mathbb{S}$, $\mathbb{D}$, $\mathbb{T}$, $\mathbb{SL}$ and $\mathbb{TL}$



(a) cost ι (b) cost $\iota + \lambda$ (c) cost 2ι

Figure 3: Costing ILS in various scenarios. In (a) the $\mathbb{I}$ event implies that multiple alleles are present in one branch of the species tree, and therefore its cost is ι. In (b) one lineage is lost immediately after the initial divergence (cost λ), but the cost of the ILS event itself is the same as in (a), i.e, ι. In (c) the initial divergence implies that multiple alleles are present in two branches of the species tree and therefore the cost of this ILS event is 2ι. The notation of Definition 4 is shown in (a) and (c), and branches inferred to contain multiple alleles are shaded.

events cost 0, δ, τ, λ and $\tau + \lambda$ respectively. However, it is less straightforward to cost $\mathbb{I}$ (and $\mathbb{IL}$) events. This is because not all $\mathbb{I}$ events are equally likely; an ILS which is resolved "quickly" is more likely to occur than one which induces gene tree-species tree discordance for a long period of time. We follow the MDC criterion of Maddison [13]; every $\mathbb{I}$ event creates incompletely sorted (i.e., multiple) alleles in some branches of the species tree. We seek to minimise the number of these "extra lineages", and therefore set the cost of an $\mathbb{I}$ event to be proportional to the number of tree branches in which incompletely sorted alleles (deep coalescences) are created, barring further events. Note that further events may cause some alleles to be lost, but this does not affect the likelihood (and therefore the cost) of the $\mathbb{I}$ event.

We define the cost of an $\mathbb{I}$ event as follows:

Definition 4 (Cost of an $\mathbb{I}$ event)**.** *Let $(C, t) \to (C_1, t), (C_2, t)$ be an $\mathbb{I}$ event. Let $x_1, \ldots, x_n \in V(S)$ be the set of nodes such that $C_1 = \cup_{i=1}^{n} C(x_i)$ and the $C(x_i)$ are maximal, i.e., there does not exist $x' \in V(S)$ with $C(x_i) \subset C(x') \subseteq C_1$. Define $y_1, \ldots, y_m$ similarly for C_2. Then the cost of this $\mathbb{I}$ event is the number of complete branches in S which are present in both a path from $n(C, t)$ to a $(x_i)_p$, and a path from $n(C, t)$ to a $(y_i)_p$, multiplied by ι.*

We define the cost in this way because the clade generated by each x_i (respectively y_i) is a subset of the set of species into which the gene is "sorted", i.e., C_1 (respectively C_2). Thus the gene lineages descend from $n(C, t)$ to all $(x_i)_p$ and $(y_i)_p$. Branches which contain both lineages must contain multiple alleles and are costed accordingly. At the points $(x_i)_p$

and $(y_i)_p$, the genes are fixed in all descendant species; thus the species no longer contains multiple alleles and we should not consider branches further down.

Note also that here we count branches in S, not S'; that is, branches in the un-subdivided species tree. This is logical as the presence of a speciation in a different part of the species tree should not affect the cost of an ILS.

The cost of an $\mathbb{IL}$ event is defined in a similar manner. See Figure 3 for some examples.

4. The algorithm

The model defined in the previous section allows ILS to occur in all parts of the tree. Unfortunately, this produces an exponential explosion in the number of possible clades, let alone reconciliations. It is impractical to find an optimal reconciliation under this model without some restrictions.

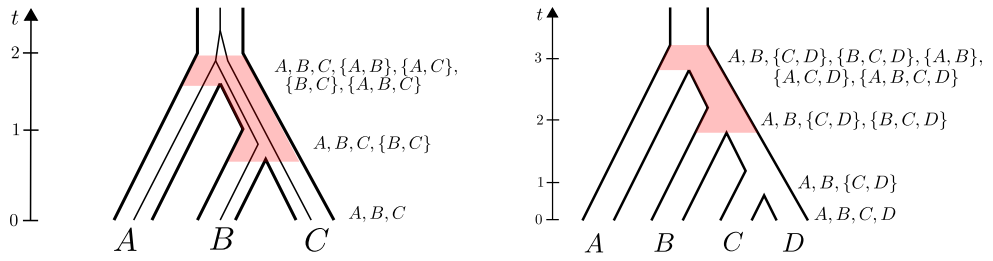
In order to make our model tractable, we restrict ILS to only occur in certain branches. More precisely, we only allow ILS to happen on branches of length not superior to a certain threshold, which we denote by $ILSlength$. Furthermore, ILS can only happen in an internal branch: leaf branches can never contain (observed) incompletely sorted alleles. We note that this is not the only reasonable way to designate branches on which ILS may occur, and, in theory, any method that designates certain branches that can contain ILS can be used in this algorithm.

In this section we now describe an algorithm to compute the minimum cost of a reconciliation between a gene tree G and a dated species tree (S, θ_S), subject to the above restriction. It is an extension of the algorithm of [7] with modifications to allow for incomplete lineage sorting. In that algorithm, time-consistency of transfers is ensured by subdividing the species tree and only allowing transfers within time "slices". We also take this approach here.

The first step in the algorithm computes (under the length restriction):

- all possible dated clades, denoted $\mathcal{C}'_\theta(S)$;
- all possible tripartitions, denoted $\Pi'_\theta(S)$;
- the cost associated to these tripartitions, stored in the function $cost : \Pi'_\theta(S) \to \mathbb{R}^+$.

If ILS is not considered, these sets are simple to define: each clade of $\mathcal{C}'_\theta(S)$ corresponds to



(a) Simple 3-taxa tree (b) 4-taxa tree

Figure 4: All possible clades for some simple trees. ILS branches are shaded. In (a), there is one ILS subtree with child clades A, B, C. In (b), there is one ILS subtree with child clades A, B, $\{C, D\}$. In both figures, clade dates are omitted for brevity and can be inferred from the positioning of the text.

a node of S', and each tripartition of $\Pi'_\theta(S)$ corresponds to a node of S', with elements that are (respectively) that node and its children.

However, when we take into account the possibility of ILS, the situation is much more complicated; we still start by subdividing the species tree and calculating the clades and tripartitions generated by it, but then we must augment these sets. To do so, we start by scanning the species tree and marking each internal branch that can contain ILS (i.e., has a length shorter than $ILSlength$). Each connected set of marked branches is considered as a single *ILS subtree* over which ILS can happen anywhere. Each child of the leaves of this subtree generates a clade, which we call *child clades*. Each child clade can be considered as a single unit with respect to this ILS subtree: it is impossible for ILS occurring in this subtree to split any of its child clades. On the other hand, the clade generated by the root of the ILS subtree can be resolved via ILS in any binary fashion that preserves the child clades. This means that any possible union of child clades is a possible clade of the ILS subtree, to which several times can be associated: the earliest possible time is the time of the root of the ILS subtree, while the latest is the time of the LCA of its elements.

We generate all possible dated clades for each ILS subtree as described above, and we add them to all clades generated by S' to form the set $\mathcal{C}'_\theta(S)$. See Figure 4 for some simple examples.

We next construct the set of possible tripartitions, which represent the possible ways in which a gene lineage can diverge due to speciation or ILS. Tripartitions corresponding to speciations not in ILS subtrees are defined as described in Section 2. We then consider each ILS subtree in turn, with each child clade of the subtree as an indivisible unit. Every clade

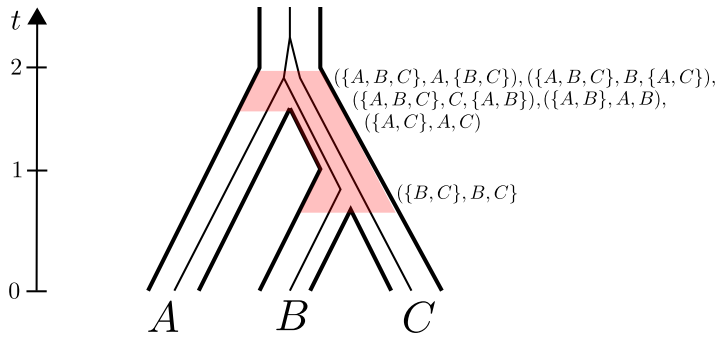


Figure 5: All possible tripartitions for the 3-taxa tree. Again, dates are omitted for brevity. Note that the tripartition $((\{B, C\}, 2), (B, 2), (C, 2))$ is not allowed as it corresponds to a speciation dated earlier than the LCA of B and C.

located in the ILS subtree can be split in any way which preserves the integrity of all child clades. If this partition can be ascribed to a speciation, then we assume it can only happen at the LCA of the species of its parent clade; otherwise, it can happen at any time that the parent clade exists in $\mathcal{C}'_\theta(S)$. In the latter case, this tripartition corresponds to an ILS. For a simple example, see Figure 5.

It still remains to describe how to calculate (and store) the costs of the ILS tripartitions. This can be done in a recursive manner for each ILS subtree. Consider an ILS tripartition $((C, t), (C_1, t), (C_2, t))$ (with $C = C_1 \cup C_2$). This tripartition will be located at $x := n(C, t)$, which we assume without loss of generality to be non-artificial. We must now consider which of the branches (x, x_l) and (x, x_r) contain alleles due to this tripartition (where x_l, x_r are the descendants of x in S). If both $C_1 \cap C(x_l)$ and $C_2 \cap C(x_l)$ are non-empty, then there will be alleles in (x, x_l); otherwise there will not, incurring no cost. If there are alleles, this adds a cost of ι to the tripartition. We then calculate the remaining number of branches with unsorted alleles below x_l by recursing on the tripartition $((C \cap C(x_l), \theta(x_l)), (C_1 \cap C(x_l), \theta(x_l)), (C_2 \cap C(x_l), \theta(x_l))$, which may have a cost of 0. The cost for the alleles in the branch (x, x_r) and descendants is calculated in an identical manner, and these costs are summed to obtain the entire cost for the tripartition. By calculating these costs in order of increasing size of C and increasing time, we can calculate the costs of all ILS tripartitions. See Figure 6 for an example.

The formal pseudocode to generate the clade and tripartition sets and ILS costs is given in Algorithm 1.

Once we have computed the clade and tripartition sets, we proceed in a fashion that is

Algorithm 1 COMPUTE $\mathcal{C}'_\theta(S)$, $\Pi'_\theta(S)$ and the *cost* function for the dated species tree (S, θ_S), given a non-negative length $ILSlength$.

Mark each internal branch of S with length $\leq ILSlength$.
$\mathcal{C}'_\theta(S), \Pi'_\theta(S) \leftarrow \emptyset$

for each leaf node $x \in V(S)$ **do** ▷ initialise with leaf clades
  $\mathcal{C}'_\theta(S) \leftarrow \mathcal{C}'_\theta(S) \cup \{(C(x), 0)\}$
end for

for each non-leaf node $x \in V(S)$ in reverse time order **do** ▷ calculate child clades of ILS subtrees
  $ILSclades(x) \leftarrow \emptyset$
  for each $x_c \in \{x_l, x_r\}$ **do**
    if (x, x_c) is marked **then**
      $ILSclades(x) \leftarrow ILSclades(x) \cup ILSclades(x_c)$
    else
      $ILSclades(x) \leftarrow ILSclades(x) \cup \{C(x_c)\}$
    end if
  end for
end for

for each internal node $r \in V(S)$ whose parent branch is not marked **do** ▷ ordinary node or root of ILS subtree
  for each non-empty subset $\{C_1, \ldots, C_n\} \subseteq ILSclades(r)$ in order of increasing size (n) **do**
    $C \leftarrow \cup_{j=1}^{n} C_j$
    $v \leftarrow LCA(C)$
    $t_C \leftarrow \theta(v)$
    for each time $t = t_C + 1, \ldots, \theta(r)$ **do**
      $\Pi'_\theta(S) \leftarrow \Pi'_\theta(S) \cup \{((C,t),(C,t-1),\emptyset)\}$ ▷ pass through artificial node or be sorted by a speciation
      $cost((C,t),(C,t-1),\emptyset) \leftarrow 0$ ▷ no cost
    end for
    for each non-empty subset $\{C_{i_1}, \ldots, C_{i_m}\} \subset \{C_1, \ldots, C_n\}$ **do**
      $C_1 \leftarrow \cup_{j=1}^{m} C_{i_j}$
      $C_2 \leftarrow C \setminus C_1$
      if $\theta(LCA(C_1)) < t_C$ and $\theta(LCA(C_2)) < t_C$ **then**
        $\Pi'_\theta(S) \leftarrow \Pi'_\theta(S) \cup \{((C,t_C),(C_1,t_C-1),(C_2,t_C-1))\}$ ▷ divergence corresponding to speciation
        $cost((C,t_C),(C_1,t_C-1),(C_2,t_C-1)) \leftarrow 0$ ▷ no cost
      else
        for each time $t = t_C, \ldots, \theta(r)$ **do**
          $\Pi'_\theta(S) \leftarrow \Pi'_\theta(S) \cup \{((C,t),(C_1,t),(C_2,t))\}$ ▷ divergence corresponding to ILS
          $w \leftarrow$ oldest non-artificial node that is $\leq n(C,t)$ ▷ calculate ILS cost
          for $(w_c, c_c) \in \{(w_l, c_l), (w_r, c_r)\}$ **do**
            if $C_1 \cap C(w_c), C_2 \cap C(w_c) \neq \emptyset$ **then**
              $c_c \leftarrow cost((C \cap C(w_c), \theta(w_c)), (C_1 \cap C(w_c), \theta(w_c)), (C_2 \cap C(w_c), \theta(w_c))) + \iota$
            else
              $c_c \leftarrow 0$
            end if
          end for
          $cost((C,t),(C_1,t),(C_2,t)) \leftarrow c_l + c_r$
        end for
      end if
    end for
  end for
end for

for each tripartition $\pi \in \Pi'_\theta(S)$ **do** ▷ assemble clade set
  $\mathcal{C}'_\theta(S) \leftarrow \mathcal{C}'_\theta(S) \cup \pi[1]$
end for

return $\mathcal{C}'_\theta(S), \Pi'_\theta(S), cost$

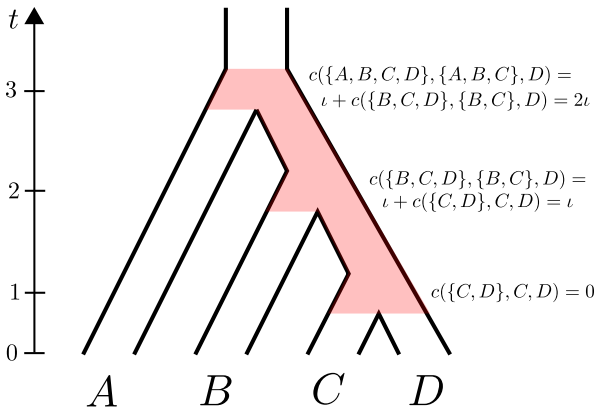


Figure 6: An example for costing the ILS tripartition $((\{A, B, C, D\}, 3), (\{A, B, C\}, 3), (D, 3))$ in a 4-taxa tree.

very similar to the original algorithm of [7], except that we must also apply the appropriate cost to ILS as specified in the previous section and calculated above. Of particular note is the calculation of the best transfer target for a given gene node v when initially mapped to a dated clade D; this is stored in $BR(v, D)$ (where BR stands for Best Receiver). The pseudocode is given in Algorithm 2. We remind the reader here that the set of clades which are transfer targets is the set of clades generated by the subdivided species tree S', i.e., $\mathcal{C}_\theta(S')$. We do not allow other clades (generated by ILS) to be transfer targets, in accordance with Definition 3.

We now show that our algorithm does indeed compute the optimal reconciliation cost (and, by backtracking, an optimal reconciliation).

Theorem 5. *Consider a gene tree G and dated species tree (S, θ_S). Then $c(G, S)$ as computed by Algorithm 2 is the minimum cost of all reconciliations between G and (S, θ_S).*

Proof. We begin by noting that in Algorithm 2, for a gene tree node v and species tree clade D, $c(v, D)$ calculates the minimum cost of reconciling the subtree of G generated by v to (S, θ_S) on the condition that $\alpha_1(v) = D$, i.e., v is first mapped to D.

For the most part, the correctness of the algorithm is then self-evident, as at each stage it merely enumerates all possible ways for the reconciliation to proceed (as according to Definition 3). Likewise, it is easy to see that the costs evaluated by Algorithm 1 are correct.

The only non-trivial issue arises from the treatment of $\mathbb{TL}$ and $\mathbb{IL}$ events. In the $\mathbb{DTL}$ model of [7], $\mathbb{TL}$ events had to be treated differently to prevent infinite loops, because $\mathbb{TL}$ events are the only events which do not change either the gene (node) or the time of the species (node). Thus when calculating $c(v, x)$, where v and x are nodes of G and S' respectively, the

Algorithm 2 COMPUTE $c(G, S)$ given positive costs δ, τ, and λ, respectively for $\mathbb{D}$, $\mathbb{T}$, $\mathbb{L}$ events, a cost ι for ILS, and a non-negative length $ILSlength$.

Compute $\mathcal{C}'_\theta(S)$, $\Pi'_\theta(S)$ and *cost* according to Algorithm 1.

for each node $v \in V(G)$ in bottom-up order **do**
  for $t \in \{0, 1, \ldots, h(S')\}$ in increasing order **do**

    for each dated clade $D = (C, t) \in \mathcal{C}'_\theta(S)$, in order of increasing size of C **do**
      for $e \in \{\mathbb{S}, \mathbb{D}, \mathbb{T}, \varnothing, \mathbb{SL}, \mathbb{TL}\}$ **do**
        $c_e \leftarrow \infty$ ▷ initialise event costs
      end for

      if $v \in L(G)$, D is a leaf clade and $s(\mathcal{L}(v)) = \mathcal{L}(D)$ **then** ▷ see lines $3-5$ of Algorithm 1
        $c(v, D) \leftarrow 0$ ▷ $\mathbb{C}$ event
        goto line 5
      end if

      $c_{\mathbb{D}} \leftarrow \min\{c_{\mathbb{D}}, c(v_l, D) + c(v_r, D) + \delta\}$ ▷ $\mathbb{D}$ event
      $c_{\mathbb{T}} \leftarrow \min\{c_{\mathbb{T}}, c(v_l, D) + c(v_r, BR(v_r, D)) + \tau, c(v_l, BR(v_l, D)) + c(v_r, D) + \tau\}$ ▷ $\mathbb{T}$ event

      for each dated tripartition $\rho \in \Pi'_\theta(S)$ with $\rho[1] = D$ **do** ▷ "divergence" in species
        if $\rho[3] = \emptyset$ **then**
          $c_{\varnothing} \leftarrow \min\{c_{\varnothing}, c(v, \rho[2])\}$ ▷ $\varnothing$ event
        else
          $c_{\mathbb{SL}} \leftarrow \min\{c_{\mathbb{SL}}, c(v, \rho[2]) + \lambda + cost(\rho), c(v, \rho[3]) + \lambda + cost(\rho)\}$ ▷ $\mathbb{SL}$ or $\mathbb{IL}$ event
        end if
      end for

      for each dated tripartition $\rho \in \Pi'_\theta(S)$ with $\rho[1] = D$ **do**
        if $\rho[3] \neq \emptyset$ **then**
          $c_{\mathbb{S}} \leftarrow \min\{c_{\mathbb{S}}, c(v_l, \rho[2]) + c(v_r, \rho[3]) + cost(\rho), c(v_l, \rho[3]) + c(v_r, \rho[2]) + cost(\rho)\}$ ▷ $\mathbb{S}$ or $\mathbb{I}$ event
        end if
      end for

      $c(v, D) \leftarrow \min\{c_e : e \in \{\mathbb{S}, \mathbb{D}, \mathbb{T}, \varnothing, \mathbb{SL}\}\}$ ▷ suboptimal cost: does not consider $\mathbb{TL}$ events
    end for

    for each dated clade $D = (C, t) \in \mathcal{C}'_\theta(S)$ with time t **do**
      $BR(v, D) \leftarrow \underset{Y=(G,t)\in\mathcal{C}_\theta(S'),\ \text{s.t. } G\not\supseteq C}{\arg\min} c(v, Y)$ ▷ find the Best Receiver for transferring v at time t
      $c_{\mathbb{TL}} \leftarrow c(v, BR(v, D)) + \tau + \lambda$ ▷ $\mathbb{TL}$ event
      if $t \neq h(S')$ **then**
        $c_{2\mathbb{TL}} \leftarrow c(v, Z) + 2\tau + 2\lambda$, where $Z = (G, t) \in \mathcal{C}_\theta(S')$ s.t. $G \supseteq C$ ▷ $\mathbb{TL}$-$\mathbb{TL}$ to original species
      end if
      $c(v, D) \leftarrow \min\{c_{\mathbb{TL}}, c_{2\mathbb{TL}}, c(v, D)\}$ ▷ Final cost for $c(v, D)$
    end for

  end for
end for

return $\min\{c(r(G), D) : D \in \mathcal{C}_\theta(S')\}$

cost of assigning a $\mathbb{TL}$ event to v cannot be calculated together with the other events as there is no guarantee that the subtree costs will already be calculated. In [7], this was accounted for by observing that it is never most parsimonious to have two consecutive $\mathbb{TL}$ events. Thus the cost of assigning all events except $\mathbb{TL}$ to v are found for all species tree nodes at the same time as x, and then the cost of assigning a $\mathbb{TL}$ event is found by calculating the best transfer target for v from x based on these calculated costs; since two consecutive $\mathbb{TL}$ events cannot occur, this is optimal.

There is a similar but more complicated situation in our algorithm. Here both $\mathbb{TL}$ and $\mathbb{IL}$ events do not change either the gene (node) or time of the species (clade). However, an $\mathbb{IL}$ event does reduce the size of the species clade, hence preventing the infinite loop problem and ensuring the availability of needed cost values as long as smaller clades are processed first.

In addition, because it is possible for a gene node to be mapped to an incompletely sorted species clade (i.e., one which is not generated by a species tree node), it is sometimes possible for two consecutive $\mathbb{TL}$ events to be most parsimonious. This can only happen if the original species clade is incompletely sorted, then two consecutive $\mathbb{TL}$ events transfer the gene to another species (provided one exists) and then back to the original species (the gene is now fixed in the entire species). This is the only scenario in which two consecutive $\mathbb{TL}$ events can be most parsimonious; note that three consecutive $\mathbb{TL}$ events can never be most parsimonious.

In order to accommodate these, we apply a method that is similar to what is done in [7]. When calculating the cost of assigning a gene tree node v to a dated species tree clade D, we calculate the cost of assigning any event except for $\mathbb{TL}$ to v. Because we calculate these costs in order of increasing size of D, we are sure that $\mathbb{IL}$ events only reference costs which are already calculated. After these costs are calculated for all clades of the same time, we then calculate the costs of assigning a $\mathbb{TL}$ event (line 29; the $G \not\supseteq C$ restriction prevents transfers back to the same species), or two consecutive $\mathbb{TL}$ events leading back to the original species (line 32; here Z is the dated clade resulting from two $\mathbb{TL}$ events back to the original species), based on the previously calculated costs. These costs are then compared to the no-$\mathbb{TL}$ cost to calculate the final cost.

This provides an optimal cost because, in addition to it never being most parsimonious to have additional $\mathbb{TL}$ events, it is also never most parsimonious to have an $\mathbb{IL}$-$\mathbb{TL}$ sequence (removing the $\mathbb{IL}$ will result in the same effect for a lower cost). Therefore we can calculate

the cost of assigning an $\mathbb{IL}$ event to v before calculating the cost of a $\mathbb{TL}$ event for v at the same time.

We note that it is also never most parsimonious to have a sequence of consecutive $\mathbb{IL}$ events, but this is not built into the algorithm. While this does not stop the algorithm from being correct (we can never have an infinite loop of $\mathbb{IL}$ events only), it is a potential unused source of optimisation.

□

The algorithm we have presented applies to the simplest case of a full, rooted, binary gene tree reconciled to a dated (and thereby rooted), binary species tree. There are several extensions to the algorithm of [7] in cases where these conditions are not met, and we discuss the analogous extensions to our algorithm below.

Amalgamating multiple gene trees

In [22], an algorithm to amalgamate a set of rooted or unrooted gene trees $\mathcal{G}$ while simultaneously reconciling with a dated species tree was presented. The basic idea is to cope with gene tree uncertainty by considering not a single binary tree per gene but a set of realistic alternative trees (e.g., those obtained by a bootstrap procedure). The amalgamation process then selects compatible clades from this set of possible trees to build up a (possibly new) gene tree minimising the reconciliation cost and made only of realistic clades.

We can adapt our algorithm to this case in a similar manner: instead of defining a reconciliation as a mapping from the nodes of G to clades of S, we decompose the gene trees in $\mathcal{G}$ into their generated clades, then define a reconciliation as a mapping from gene clades to species clades. The algorithm can then be used by matching each gene tripartition present in the set of gene trees to either a species tree tripartition or a genetic event. This results in a reconciliation which identifies the optimal gene tripartitions and thus defines an amalgamated gene tree which contains only tripartitions which are present in the set of gene trees. Pseudocode for this extension is given in Appendix A.

Unrooted gene trees

The extension above can also be adapted for use with an unrooted gene tree (or amalgamating multiple unrooted gene trees). Here, we consider all clades and all gene tripartitions present in all possible rootings of the gene tree(s), and then proceed as before.

Undated species tree

If the species tree is undated, we first assign each node a date of 0. Thus any branch can contain a transfer to any other branch. In this case, we have no way of ensuring that the optimal reconciliation produced is time-consistent; this must be checked *post hoc*, and if it is found to contain a time paradox, another optimal reconciliation must be tried. Indeed, it is possible that all optimal reconciliations are not time-consistent, in which case the algorithm will fail.

Note that in this scenario, we must also have some alternative way of designating "ILS branches", as the species tree is undated. As observed before, it is impractical for all branches to be ILS branches.

5. Complexity

Let k be the maximum polytomy degree (that is, the number of branches in the largest ILS subtree plus 2) and n_k the number of polytomies.

We first count the number of clades. There are $O(|S|)$ nodes (counting internal nodes) present in S, but due to subdivision these are replicated to $O(|S|^2)$ nodes in S'.

Now consider a single ILS subtree with k child clades. At the root there are $2^k - 1$ possible clades to consider, but (some of) these clades are replicated throughout the ILS subtree. The ILS subtree can have up to $k-2$ non-artificial levels, but these may be subdivided from other nodes outside the subtree; the best we can say is that the subtree has at most $|S|$ levels. Therefore the number of clades generated by this tree is $O(|S|2^k)$, and the total number of clades is

$$O(|S|^2 + n_k|S|2^k).$$

In Algorithm 2, there are three nested loops: we loop over all nodes of G (of which there are $O(|G|)$, then over all clades in $\mathcal{C}'_\theta(S)$ (the number of which we have calculated above), then over all tripartitions corresponding to the species clade. (The complexity of the loop for handling $\mathbb{TL}$ events is clearly dominated by this loop over all tripartitions.) The maximum number of these tripartitions for any clade is 2^k. Putting these three steps together, we find that the total complexity for this algorithm is

$$O(|G|(|S|^2 + |S|n_k 2^k)2^k).$$

In comparison, the NOTUNG algorithm of Stolzer *et al.* [23] has an efficiency of

$$O(|G|(|S|+n_k 2^k)^2(h_S+k)),$$

where h_S is the height of the species tree (which, in the worst case, is $O(|S|)$). This is slightly worse than our algorithm, depending on the relative values of $|S|$ and k. In the case where k is held fixed and $|S|\to\infty$, our algorithm is $O(|G||S|^2)$ and NOTUNG is $O(|G||S|^3)$. NOTUNG additionally has other disavantages as detailed in the next section.

If we are amalgamating m gene trees, the outermost loop would be over all clades appearing in those trees, which requires at most $m|G|$ iterations instead of the $|G|$ previously for a single gene tree, so the complexity of the algorithm is

$$O(m|G|(|S|^2+|S|n_k 2^k)2^k).$$

If the gene tree(s) is unrooted, the complexity of the algorithm does not change; the number of possible gene clades is multiplied by a constant factor of 2, and the number of possible gene tripartitions by a factor of 3.

If the species tree is undated, we lose a factor of $|S|$ from the number of clades generated by S'. We also replace the same factor from the ILS clades by the maximum height of an ILS subtree (k). Therefore the complexity of the algorithm is

$$O(|G|(|S|+kn_k 2^k)2^k),$$

with an extra factor of m if amalgamating m gene trees.

6. Comparisons with other models

Several algorithms to incorporate incomplete lineage sorting into reconciliation models have been proposed before. In this section, we compare our model and algorithm with other methods. It is important to note that often it is the model of incomplete lineage sorting which differs slightly from author to author; each algorithm is formulated to solve the respective model proposed, rather than a universally consistent model. This can make a direct comparison between algorithms less meaningful.

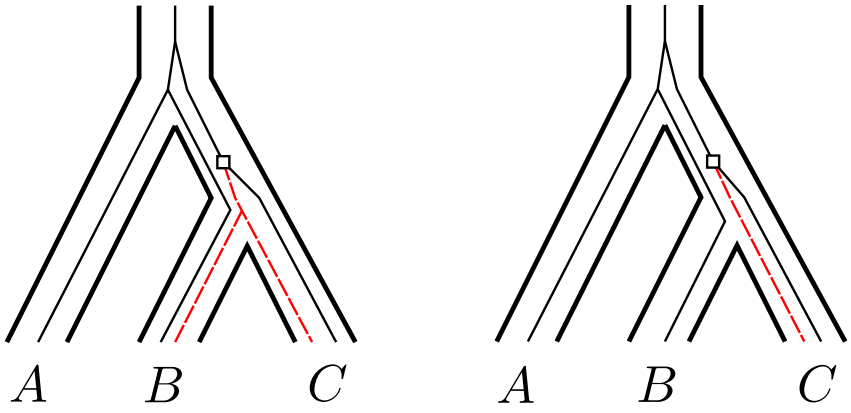
Rasmussen and Kellis [21] proposed a model of incomplete lineage sorting (named DLCoal) based on a coalescent model, and proposed a probabilistic reconciliation method (DLCoalRecon) which also incorporated duplication and loss. Wu, Rasmussen and Kellis [27] used the same model, but devised a parsimonious method (DLCpar) instead, which is a more direct relation to our method. A feature of their methods is that they keep direct track of the (inferred) locus of the genes; this allows them to separate orthologous genes (arising from speciation or ILS) from paralogous genes (arising from duplication).

Our model of ILS is largely similar the DLCoal model, albeit with some subtle differences. While both models allow ILS to interact with other events (i.e., it is possible for genes which are not fully resolved in a species to be duplicated or lost), the manner in which they interact differs.

- The DLCoal model does not allow "hemiplasy", i.e., if a gene duplicates, it is fixed in all descendant species at the new locus. Thereafter, the duplicated gene evolves independently from the original. If the original gene is fixed in the species at the time of duplication (i.e., it is not part of an ILS), then this is identical to our model. On the other hand, if the original gene is not yet fixed (due to ILS) in its species, in our model the duplicated copy is enforced to remain in the same individuals as the original gene; thus, barring further events, it will become fixed in exactly the same species as the original gene. This is depicted in Figure 7. It is possible to enforce this in our algorithm, because it keeps track of all the species into which each gene will eventually be sorted. We also do not allow hemiplasy, but in the sense that the duplicated gene cannot fix in some, but not all of the species that its parent fixes in.

  An alternative way of viewing this is that our model does not allow recombination — if a gene duplicates, the duplicated copies must evolve together rather than independently, as they appear in the same individuals. In contrast, the DLCoal model allows free recombination — once a gene appears at a different locus, it is considered for all intents and purpose as a new gene. It is not immediately clear which model is more realistic, or indeed if recombination should be allowed but penalised in some way. This would introduce another layer of complexity to the model, and we do not consider it in this paper.

  These two different perspectives on duplication give slightly different costs for various



(a) DLCoal model (b) Our model

Figure 7: Different treatments of duplication within an ILS. In each case there is only one ILS and one duplication (denoted by a square). In (a), the duplicated gene (dashed) fixes in species B and C. In (b), the duplicated gene fixes only in species C, as its parent fixes only in C.

scenarios. For example, in Figure 7a, the DLCpar algorithm costs this scenario at $1\iota+1\delta$, whereas we would not allow it all (we could mimic the effect of fixing the duplicated gene with two consecutive $\mathbb{TL}$ events, leading to a total cost of $1\iota + 1\delta + 2\tau + 2\lambda$, but this is a substantially different biological scenario). In contrast, in Figure 7b, our algorithm costs this scenario at $1\iota + 1\delta$, but DLCpar infers an extra loss, for a total cost of $1\iota + 1\delta + 1\lambda$.

- Because the DLCoal model is based on a coalescent perspective, it treats duplications from this viewpoint: running forwards in time, an allele is created (resulting in a divergence in the gene tree), which then simultaneously changes locus and becomes lost at the original locus. The end result is that the gene is duplicated at a new locus, but because there may be a delay between the creation of the allele and the change of locus, it is possible to have incomplete lineage sorting between a duplication and a single speciation (see for example Figure 1C of [27]). In our model, we consider duplications to be instantaneous events which do not create alleles, and so this cannot happen.

- A similar scenario happens with the way losses interact with ILS. Because the DLCoal model arises from a coalescent perspective, it only "observes" (and thus costs) ILS when it infers two incompletely sorted alleles in the same locus (in one branch). It does not, and cannot, account for the possibility that a gene occurs in some of the population, while the rest of the population has no copies of the gene due to loss. This is allowed,

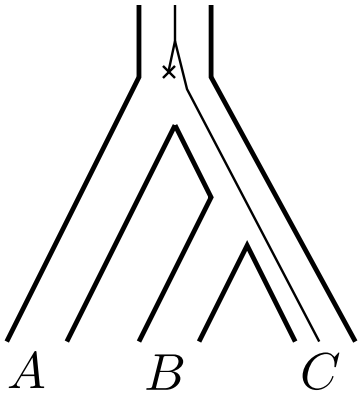


Figure 8: Loss within an ILS. Here is an ILS in the root species, but one allele (which would otherwise fix into species A and B) is immediately lost.

and accounted for, in our model. This is depicted in Figure 8; this scenario would be impossible in the DLCoal model.

- The DLCoal model allows incomplete lineage sorting to happen on all branches. This is necessary for this model, because as we discuss above, it does not require close speciations in order to have ILS. However, the probability of ILS between two speciations decreases quickly as the branch length increases. This is accounted for in DLCoalRecon, but not in DLCpar; there, the penalty for ILS is invariant to the length of the branch. Our algorithm allows ILS to occur only in the presence of rapid successive speciation events, i.e., over branches of small length (under a threshold).

While there are some differences in specifics, overall our model of ILS is largely similar to the DLCoal model. On the other hand, our reconciliation algorithm is entirely different to DLCpar, owing almost entirely to the fact that they do not consider transfers in their model. It is well known that including transfers in a reconciliation model makes it much harder; for example, the $\mathbb{DL}$ reconciliation model (without ILS) is easily solvable using the LCA mapping, but the $\mathbb{DTL}$ model is known to be NP-complete in general [25], and even with a dated species tree requires a polynomial-time algorithm [7]. DLCpar itself requires a dynamic programming algorithm to solve the ILS model with duplication and loss only, but the dynamic programming is iterated over the possible loci for genes in a branch, while each branch is solved more or less independently. Our algorithm iterates over the branches; while it is less efficient, this is an unavoidable result of the increased complexity from including transfers.

While our ILS model differs from DLCoal, it is more or less identical to that of Stolzer *et al.* [23]. However, our method has some significant differences from theirs (NOTUNG):

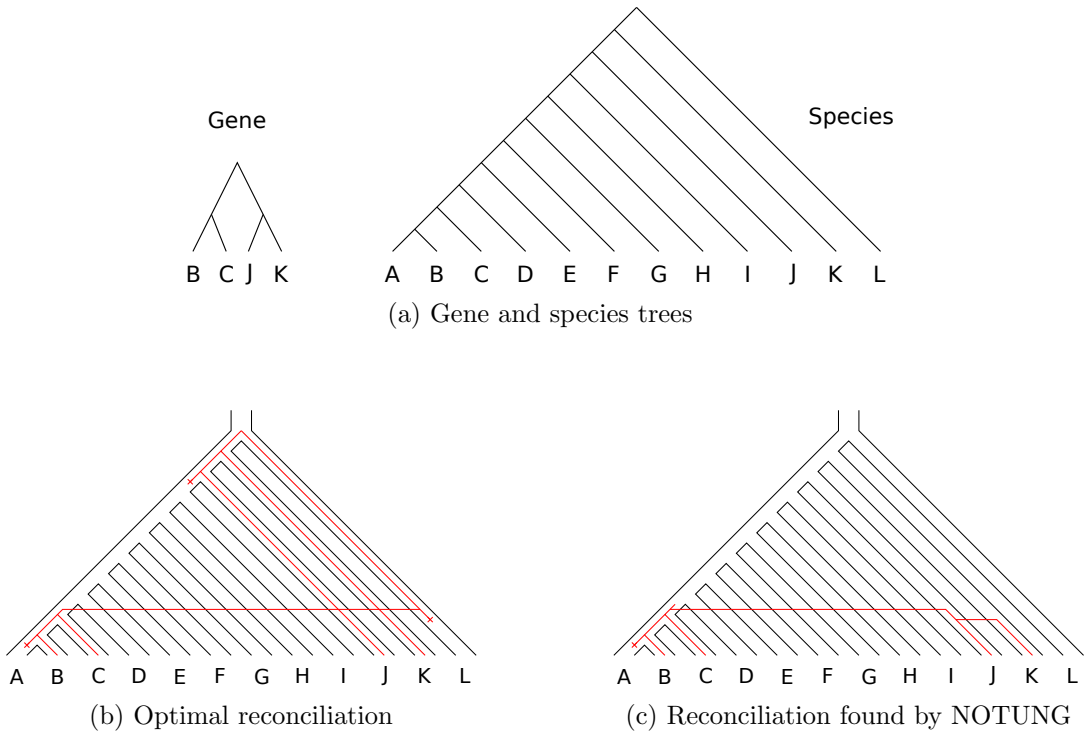
- Our method constructs time-consistent solutions by incorporating the dates of the species tree into the algorithm. In contrast, NOTUNG constructs reconciliations without respect to time-consistency, then filters out time-inconsistent solutions *a posteriori*. This can result in no solution being returned.

- Our method takes a binary species tree, and infers ILS (or other events) when there is incongruence between the gene and species tree. NOTUNG first collapses all short branches into polytomies, and allows ILS on only those branches. However the presence of ILS is unpenalised and so there is no difference between when there is incongruence with the species tree due to ILS and when there is no incongruence.

- Our method has a lower time complexity, as detailed in the previous section.

In addition, NOTUNG makes an implicit but unwarranted assumption that if a gene is present in a species, then it must survive to at least one extant descendant of that species. This is untrue, as it is possible that a gene can be transferred to another species and subsequently lost (a $\mathbb{TL}$ event). This can cause NOTUNG to sometimes produce a suboptimal reconciliation. See Figure 9: when we set the costs to $D = 2, T = 3, L = 1$, then NOTUNG infers the reconciliation in Figure 9c for a cost of 7, whereas the reconciliation in Figure 9b has a lower cost of 6. NOTUNG fails to infer this reconciliation because it contains a $\mathbb{TL}$ event. It is known [3] that the $\mathbb{DTL}$ model simplifies significantly if $\mathbb{TL}$ events are disallowed.

The *mowgliNNI* algorithm of [17], and other algorithms for dealing with gene tree error, could also be used in the context of incomplete lineage sorting, as it (heuristically) modifies the gene trees using nearest-neighbour interchanges (NNI), which mimics the basic effect of ILS. However, the two underlying problems are not equivalent, since *mowgliNNI* allows NNI on pre-selected (unreliable) branches of the gene tree, whereas to correctly account for ILS, NNI should be considered on pre-selected (short) branches of the species tree. Moreover, our algorithm is an exact solution of the most parsimonious $\mathbb{IDTL}$ model, rather than a heuristic to search a broader reconciliation space. Lastly, the effect of ILS is not limited solely to NNI, especially in interactions with $\mathbb{DTL}$ events.



(a) Gene and species trees

(b) Optimal reconciliation

(c) Reconciliation found by NOTUNG

Figure 9: A scenario for which NOTUNG fails to identify the optimal reconciliation when costs are set to $\delta = 2, \tau = 3, \lambda = 1$.

7. Conclusion

In this paper, we have shown how to model incomplete lineage sorting in a reconciliation context. We have created a formal unified framework under which duplications, losses, transfers and ILS can all be accounted for. We have also developed an extension to the algorithm of Doyon *et al.* which calculates a globally most parsimonious $\mathbb{IDTL}$ reconciliation, with guaranteed time-consistency of transfers. This algorithm is efficient if ILS is not allowed on too many branches.

This work develops a foundation for the practical inference of ILS, by showing that it can be performed efficiently and how to do so. With these tools we can analyse real databases to measure how prevalent ILS is in evolutionary history and its relative importance to the other macro-events. We can also estimate the effect of ILS on the accuracy of phylogenetic and reconciliation inference. Finally, we can clearly distinguish between orthologous and paralogous genes. These analyses are the subject of future works.

The fixed-parameter tractable algorithm developed here is exponential only in k, the size of the largest ILS subtree. This means that it is a practical solution for most realistic cases

where ILS is mostly concentrated on few branches of the tree due to rapid successive speciation events. Unlike the DLCpar algorithm, which allows ILS everywhere with no differentiation, we allow it only on short branches. Ideally, we would like to allow ILS on all branches but with a higher cost for ILS on longer branches; however, this would introduce another level of complexity to the algorithm.

A potential way to make the algorithm more efficient is to only allow $\mathbb{I}$ events which cost less than a certain threshold, instead of or in addition to limiting branches on which ILS is allowed. This would limit the number of possible clades generated by ILS and thus escape the exponential dependence on maximum polytomy size, but again introduces more complexity to the algorithm, and we have not explored it further here.

We lastly note that although ILS is not due to errors in gene tree inference, it is possible that the algorithm here could be modified in order to find reconciliations which account for gene tree error rather than ILS, as they both have similar effects on the gene tree (i.e., nearest neighbour interchange).

Acknowledgments

We would like to thank Nicolas Galtier and Edwin Jacox for fruitful discussions. This work was supported by the French Agence Nationale de la Recherche Investissements d'Avenir/Bioinformatique (ANR-10-BINF-01-01, ANR-10-BINF-01-02, Ancestrome).

References

[1] L. Arvestad, A.-C. Berglund, J. Lagergren, and B. Sennblad. Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution. In *Proceedings of the Eighth Annual International Conference on Research in Computational Molecular Biology*, pages 326–335. ACM, 2004.

[2] 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.*, 12(1):6, 2017.

[3] Y. Chan, V. Ranwez, and C. Scornavacca. Exploring the space of gene/species reconciliations with transfers. *J. Math. Biol.*, 71(5):1179–1209, 2015.

[4] L. A. David and E. J. Alm. Rapid evolutionary innovation during an Archaean genetic expansion. *Nature*, 469(7328):93–96, 2010.

[5] J. H. Degnan and N. A. Rosenberg. Gene tree discordance, phylogenetic inference and the multispecies coalescent. *Trends Ecol. Evol.*, 24(6):332–340, 2009.

[6] J. Doyon, V. Ranwez, V. Daubin, and V. Berry. Models, algorithms and programs for phylogeny reconciliation. *Brief. Bioinform.*, 12(5):392–400, 2011.

[7] J.-P. Doyon, C. Scornavacca, K. Y. Gorbunov, G. J. Szöllősi, V. Ranwez, and V. Berry. An efficient algorithm for gene/species trees parsimonious reconciliation with losses, duplications and transfers. In *Proceedings of the 2010 International Conference on Comparative Genomics*, RECOMB-CG'10, pages 93–108, Berlin, Heidelberg, 2011.

[8] P. Górecki and J. Tiuryn. DLS-trees: a model of evolutionary scenarios. *Theor. Comput. Sci.*, 359(1):378–399, 2006.

[9] J. F. C. Kingman. The coalescent. *Stoc. Proc. Appl.*, 13(3):235–248, 1982.

[10] J. F. C. Kingman. On the genealogy of large populations. *J. Appl. Probab.*, pages 27–43, 1982.

[11] V. E. Kutschera, T. Bidon, F. Hailer, J. L. Rodi, S. R. Fain, and A. Janke. Bears in a forest of gene trees: phylogenetic inference is complicated by incomplete lineage sorting and gene flow. *Mol. Biol. Evol.*, 31(8):2004–2017, 2014.

[12] L. Liu, L. Yu, L. Kubatko, D. K. Pearl, and S. V. Edwards. Coalescent methods for estimating phylogenetic trees. *Mol. Phylogenet. Evol.*, 53(1):320–328, 2009.

[13] W. P. Maddison. Gene trees in species trees. *Syst. Biol.*, 46(3):523–536, 1997.

[14] J. A. McGuire, C. W. Linkem, M. S. Koo, D. W. Hutchison, A. K. Lappin, D. I. Orange, J. Lemos-Espinal, B. R. Riddle, and J. R. Jaeger. Mitochondrial introgression and incomplete lineage sorting through space and time: phylogenetics of crotaphytid lizards. *Evolution*, 61(12):2879–2897, 2007.

[15] B. S. McLean, D. J. Jackson, and J. A. Cook. Rapid divergence and gene flow at high latitudes shape the history of holarctic ground squirrels (urocitellus). *Mol. Phylogenet. Evol.*, 102:174–188, 2016.

[16] D. Merkle, M. Middendorf, and N. Wieseke. A parameter-adaptive dynamic programming approach for inferring cophylogenies. *BMC Bioinformatics*, 11(1):1, 2010.

[17] T. H. Nguyen, V. Ranwez, S. Pointet, A.-M. A. Chifolleau, J.-P. Doyon, and V. Berry. Reconciliation and local gene tree rearrangement can be of mutual profit. *Algorithm. Mol. Biol.*, 8(1):12, 2013.

[18] Y. Ovadia, D. Fielder, C. Conow, and R. Libeskind-Hadas. The cophylogeny reconstruction problem is np-complete. *J. Comput. Biol.*, 18(1):59–65, 2011.

[19] V. Ranwez, C. Scornavacca, J.-P. Doyon, and V. Berry. Inferring gene duplications, transfers and losses can be done in a discrete framework. *J. Math. Biol.*, pages 1–34, 2015.

[20] M. D. Rasmussen and M. Kellis. A Bayesian approach for fast and accurate gene tree reconstruction. *Mol. Biol. Evol.*, 28(1):273–290, 2011.

[21] M. D. Rasmussen and M. Kellis. Unified modeling of gene duplication, loss, and coalescence using a locus tree. *Genome Res.*, 22(4):755–765, 2012.

[22] C. Scornavacca, E. Jacox, and G. J. Szöllősi. Joint amalgamation of most parsimonious reconciled gene trees. *Bioinformatics*, 31(6):841–848, 2015.

[23] M. Stolzer, H. Lai, M. Xu, D. Sathaye, B. Vernot, and D. Durand. Inferring duplications, losses, transfers and incomplete lineage sorting with nonbinary species trees. *Bioinformatics*, 28(18):i409–i415, 2012.

[24] C. Than and L. Nakhleh. Species tree inference by minimizing deep coalescences. *PLoS Comput. Biol.*, 5(9):e1000501, 2009.

[25] A. Tofigh, M. Hallett, and J. Lagergren. Simultaneous identification of duplications and lateral gene transfers. *IEEE-ACM T. Comput. Bi.*, 8(2):517–535, 2011.

[26] B. Vernot, M. Stolzer, A. Goldman, and D. Durand. Reconciliation with non-binary species trees. *J. Comput. Biol.*, 15(8):981–1006, 2008.

[27] 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.*, 24(3):475–486, 2014.

[28] Y. Yu, R. M. Barnett, and L. Nakhleh. Parsimonious inference of hybridization in the presence of incomplete lineage sorting. *Syst. Biol.*, page syt037, 2013.

[29] C. M. Zmasek and S. R. Eddy. A simple algorithm to infer gene duplication and speciation events on a gene tree. *Bioinformatics*, 17(9):821–828, 2001.

Appendix A. Pseudocode for multiple gene trees

Algorithm 3 COMPUTE $c(\mathcal{G}, S)$ given positive costs δ, τ, and λ, respectively for $\mathbb{D}$, $\mathbb{T}$, $\mathbb{L}$ events, a cost ι for ILS, and a non-negative length $ILSlength$.

Compute $\mathcal{C}'_\theta(S)$, $\Pi'_\theta(S)$ and *cost* according to Algorithm 1.

for each clade $C \in \mathcal{C}(\mathcal{G})$ in order of increasing size **do**
  for $t \in \{0, 1, \ldots, h(S')\}$ in increasing order **do**

    for each dated clade $D = (G, t) \in \mathcal{C}'_\theta(S)$ with time t, in order of increasing size of D **do**
      for $e \in \{\mathbb{S}, \mathbb{D}, \mathbb{T}, \varnothing, \mathbb{SL}, \mathbb{TL}\}$ **do**
        $c_e \leftarrow \infty$ ▷ initialise event costs
      end for

      if C and D are leaf clades and $s(\mathcal{L}(C)) = \mathcal{L}(D)$ **then** ▷ see lines $3-5$ of Algorithm 1
        $c(C, D) \leftarrow 0$ ▷ $\mathbb{C}$ event
        goto line 5
      end if

      for each tripartition $\pi \in \Pi(C)$ **do** ▷ divergence in gene
        $c_{\mathbb{D}} \leftarrow \min\{c_{\mathbb{D}}, c(\pi[2], D) + c(\pi[3], D) + \delta\}$ ▷ $\mathbb{D}$ event
        $c_{\mathbb{T}} \leftarrow \min\{c_{\mathbb{T}}, c(\pi[2], D) + c(\pi[3], BR(\pi[3], D)) + \tau, c(\pi[2], BR(\pi[2], D)) + c(\pi[3], D) + \tau\}$ ▷ $\mathbb{T}$ event
      end for

      for each dated tripartition $\rho \in \Pi'_\theta(S)$ with $\rho[1] = D$ **do** ▷ "divergence" in species
        if $\rho[3] = \emptyset$ **then**
          $c_{\varnothing} \leftarrow \min\{c_{\varnothing}, c(C, \rho[2])\}$ ▷ $\varnothing$ event
        else
          $c_{\mathbb{SL}} \leftarrow \min\{c_{\mathbb{SL}}, c(C, \rho[2]) + \lambda + cost(\rho), c(C, \rho[3]) + \lambda + cost(\rho)\}$ ▷ $\mathbb{SL}$ or $\mathbb{IL}$ event
        end if
      end for

      for each tripartition $\pi \in \Pi(C)$ **do** ▷ divergence in gene and species
        for each dated tripartition $\rho \in \Pi'_\theta(S)$ with $\rho[1] = D$ **do**
          if $\rho[3] \neq \emptyset$ **then**
            $c_{\mathbb{S}} \leftarrow \min\{c_{\mathbb{S}}, c(\pi[2], \rho[2]) + c(\pi[3], \rho[3]) + cost(\rho), c(\pi[2], \rho[3]) + c(\pi[3], \rho[2]) + cost(\rho)\}$ ▷ $\mathbb{S}$ or $\mathbb{I}$ event
          end if
        end for
      end for

      $c(C, D) \leftarrow \min\{c_e : e \in \{\mathbb{S}, \mathbb{D}, \mathbb{T}, \varnothing, \mathbb{SL}\}\}$ ▷ suboptimal cost: does not consider $\mathbb{TL}$ events
    end for

    for each dated clade $D = (G, t) \in \mathcal{C}'_\theta(S)$ with time t **do**
      $BR(C, D) \leftarrow \underset{Y=(H,t)\in\mathcal{C}_\theta(S'),\ \text{s.t. } H\not\supseteq G}{\arg\min} c(C, Y)$ ▷ find the Best Receiver for transferring C at time t
      $c_{\mathbb{TL}} \leftarrow c(C, BR(C, D)) + \tau + \lambda$ ▷ $\mathbb{TL}$ event
      if $t \neq h(S')$ **then**
        $c_{2\mathbb{TL}} \leftarrow c(C, Z) + 2\tau + 2\lambda$, where $Z = (H, t) \in \mathcal{C}_\theta(S')$ s.t. $H \supseteq G$ ▷ $\mathbb{TL}$-$\mathbb{TL}$ to original species
      end if
      $c(C, D) \leftarrow \min\{c_{\mathbb{TL}}, c_{2\mathbb{TL}}, c(C, D)\}$ ▷ Final cost for $c(C, D)$
    end for

  end for
end for

return $\min\{c(L(\mathcal{G}), D) : D \in \mathcal{C}_\theta(S')\}$