Inferring gene duplications, transfers and losses can be done in a discrete framework

Vincent Ranwez, Celine Scornavacca, Jean-Philippe Doyon, Vincent Berry

Noname manuscript No.
(will be inserted by the editor)

Inferring gene duplications, transfers and losses can be done in a discrete framework

Vincent Ranwez[1,4], **Celine Scornavacca**[2,4], **Jean-Philippe Doyon**[2,3] **and Vincent Berry**[3,4]

the date of receipt and acceptance should be inserted later

Abstract In the field of phylogenetics, the evolutionary history of a set of organisms is commonly depicted by a species tree – whose internal nodes represent speciation events – while the evolutionary history of a gene family is depicted by a gene tree – whose internal nodes can also represent macro-evolutionary events such as gene duplications and transfers.

As speciation events are only part of the events shaping a gene history, the topology of a gene tree can show incongruences with that of the corresponding species tree. These incongruences can be used to infer the macro-evolutionary events undergone by the gene family. This is done by embedding the gene tree inside the species tree and hence providing a *reconciliation* of those trees. In the past decade, several parsimony-based methods have been developed to infer such reconciliations, accounting for gene duplications ($\mathbb{D}$), transfers ($\mathbb{T}$) and losses ($\mathbb{L}$).

The main contribution of this paper is to formally prove an important assumption implicitly made by previous works on these reconciliations, namely that solving the (maximum) parsimony $\mathbb{DTL}$ reconciliation problem in the discrete framework is equivalent to finding a most parsimonious $\mathbb{DTL}$ scenario in the continuous framework. In the process, we also prove several intermediate results that are useful on their own and constitute a theoretical toolbox that will likely facilitate future theoretical contributions in the field.

Keywords tree reconciliation; tree embedding; gene evolution; phylogenetics; parsimony; equivalence.

1 Introduction

The evolutionary history of organisms is commonly depicted by a *species tree*, whose internal nodes represent speciation events [9,36]. A *gene tree* depicts the evolutionary history of a gene family, i.e., a set of homologous sequences appearing in the genome of different organisms [16]. The history of a gene family is shaped by events affecting simultaneously all genes, such as speciations and whole genome duplications [41,32], but also by locus specific events such as gene duplications, losses and transfers. The presence of numerous gene transfers between organisms has been shown in prokaryotes and at the origins of life [25,18,38,45,42], while the presence of rampant duplications is recognized as a driving force of evolution in eukaryotes [29,31,15]. Finally, gene losses are known to be frequent in both eukaryote and prokaryote genomes [27,40,11,26].

As speciation events are only part of the events intervening in a gene history, a gene tree can be truly incongruent with the species tree. Congruences and incongruences of these trees can be represented by embedding the gene tree inside the species tree, the branches of the latter being displayed as tubes within which evolve parts of the former [30]. In the past decade, several methods have been developed to infer such embeddings [25,2,43,37,1,38,44,45,42] according to various assumptions and paradigms. These *reconciliation* methods explain the incongruences between gene and species trees by gene evolution

[1]SupAgro, UMR-AGAP, 2 place Pierre Viala 34060 Montpellier, France E-mail: vincent.ranwez@supagro.fr · [2]ISEM, UMR 5554 (Univ. Montpellier, CNRS, IRD, EPHE) Place Eugene Bataillon Montpellier, France E-mail: [jean-philippe.doyon,celine.scornavacca]@univ-montp2.fr · [3]LIRMM, CNRS - Univ. Montpellier, France. E-mail: vberry@lirmm.fr · [4]Institut de Biologie Computationnelle, Montpellier, France

events as those listed above. Reconciliation methods find applications in various areas such as functional annotation in genomics [17], studies on habitat areas in biogeography and coevolutionary studies in ecology [8]. Authors from the latter field have contributed several algorithms for reconciling host and parasite evolutionary trees, that directly apply to reconcile gene and species trees (*e.g.*, [7,34]). We refer the reader to [12] for a review of reconciliation models and methods.

Originating from the seminal paper of Goodman [19], parsimony is the most used paradigm for reconciling trees [33,50,7,34,20,48,13,10,3]. The principle is first to associate a cost to each kind of event (duplication, loss, ...); then the costs of the individual events taking place in a tree reconciliation scenario are added up to give the cost of this scenario. Given a gene and a species tree, the goal is then to find a most parsimonious scenario, that is one of lowest cost. When duplications and losses are the only considered events, the parsimony reconciliation problem can be solved in linear time [51] and it remains tractable when the species tree [50] or the gene tree [28] is non-binary. When transfers are added to the model, then we face an NP-complete problem, even for reconciling two binary trees [48]. The crux of the difficulty is to ensure time consistency, *i.e.*, that transfers do happen between co-existing species (each transfer imposes time constraints on nodes of the species tree belonging to different subtrees, and satisfying all those constraints simultaneously is challenging due to their non independence, see [24,48]). Several approaches have been proposed to overcome this difficulty. First, some authors proposed the idea to fix in advance (by external means) the pairs of branches in the species tree S between which transfers are allowed (a representation called *species graph* in [21] and *lateral transfer scheme* in [24]). A partial ordering of these pairs of branches allows to check whether a reconciliation allowing transfers only between these pairs of branches is consistent, and even to compute a parsimonious consistent reconciliation, in polynomial time [22,23]. However, in the absence of *a priori* information on possible transfers, the general problem of computing a species graph inducing a most parsimonious reconciliation is NP-complete [21], an exact exponential algorithm being proposed in [5]. Alternatively, transfers can also be handled by providing a dated tree S rather than a potential set of possible transfers. Indeed, when adding date information to the nodes of the species tree, the problem can again be solved in polynomial time, despite transfers [33,7,34,20,49,13]. Such dates can be obtained by relaxed molecular clock techniques working from molecular sequences [14,39]. Note that relative dates are sufficient for reconciliation, and that inferred dates can also be used to define a species graph.

The parsimony algorithms cited above do not all rely on explicit models of evolution. Some only specify that they assign costs to each kind of considered event [4,10], others only indicate that they consider a mapping from nodes of the gene tree to nodes/edges of the species tree (*e.g.*,[35,50]). Yet, not any mapping represents a biologically meaningful scenario. For instance, a mapping where a gene tree node u is mapped on a species branch pre-dating the branch on which a child of u is mapped is not biologically meaningful. Some of those algorithms rely on models allowing for such time inconsistent scenarios to happen [7,34,48]. Another weakness of these (implicit) proposed models is that they (tacitly) assume the most straightforward scenario between the mapping of a gene tree node and the mapping of its children and hence, *de facto,* exclude scenarios that are biologically meaningful but difficult to reconstruct [5,13,34,10,3]. In the current paper, we consider the case where a binary gene tree and a binary dated species tree are provided as input together with event costs, and we formalize an explicit model of gene evolution including speciations ($\mathbb{S}$) duplications ($\mathbb{D}$), transfers ($\mathbb{T}$) and losses ($\mathbb{L}$), that only allows for biologically realistic scenarios (as far as $\mathbb{S}$, $\mathbb{D}, \mathbb{T}$, and $\mathbb{L}$ are the only considered events). This declarative model hence allows to discriminate between valid and invalid reconciliations, that is to answer the question of whether a given candidate history is a correct reconciliation or not – regardless of the algorithm used to produce this reconciliation. Up to now, valid scenarios were implicitly defined as those explored by a given algorithm This is the first time, to our knowledge, that a reconciliation model with these desirable properties is proposed. This formalization should help to compare proposed methods and ease the dialogue between computer scientists and biologists.

The models on which rely the parsimony algorithms cited above are purely combinatorial models, explaining how the visible part of a gene family history (the extant genes) evolved according to a discrete framework. In contrast, the real history of the gene family involves hidden genes (that were subsequently lost) and evolutionary events that have happened at a precise time, *i.e.*, in a continuous framework. The main contribution of the current paper is to show that solving the (maximum) parsimony reconciliation problem in the discrete framework – the *reconciliation model* – for a gene tree depicting only the history of the extant genes, is equivalent to finding a most parsimonious scenario in the continuous framework – the *dated scenario model* – for the *full* gene tree (which includes extant as well as the lost genes, and their ancestors). Though this result is implicit in many works cited above, it has never been proved explicitly. Thus, this paper demonstrates the soundness of an important assumption made by previous works in the field. The proof is somewhat intricate and includes an intermediate framework (called *sliced scenario model*) where time is discretized in time slices and the full gene tree is considered.

In the process we show that any time-consistent reconciliation can be represented, with no information loss, by mapping gene nodes on the initial species tree (instead, for example, of mapping them on the species tree refinement used to ensure time-consistency [13], or having a supplementary mapping to distinguish between duplication, speciation and transfer events [48]). This provides a more compact way for encoding reconciliations and permits to define equivalence classes for reconciliations. As the class characterization of any reconciliation can easily be obtained, it becomes easier to check if two reconciliations are based on the same set of evolutionary events, up to minor date variations of their predicted events that can not be distinguished in the parsimony setting. More generally, this may serve as a groundwork for future definitions of distances between reconciliations as this ensures that it suffices to compare their class representatives, *i.e.* their canonical elements.

After introducing some basic notations and the three reconciliation models mentioned above (namely: dated scenario, sliced scenario and reconciliation), we prove successively the equivalence of the dated scenario and sliced scenario problems, of the reconciliation and sliced scenario problems. This leads to the equivalence of the dated scenario and reconciliation problems. We conclude by discussing the impact of these results as well as perspectives.

2 Preliminaries

In this paper we focus on leaf labeled rooted trees with nodes with at most two children and rooted by an artificial branch on their top. Given a rooted tree T with labels on its leaves, we denote its root by $r(T)$ and the set of its nodes, edges, leaf nodes and leaf labels respectively by $V(T)$, $E(T)$, $L(T)$ and $\mathcal{L}(T)$. Each leaf u is associated to a species, denoted $s(u)$. An internal node u of T has one child (u_l) or two interchangeable children (u_l, u_r).

For a node u of T, we denote u_p its parent node, (u_p, u) its parent edge, and T_u and $T_{(u_p,u)}$ respectively the subtree of T rooted at node u, and on edge (u_p, u). The topology of T induces a partial order on its nodes. Given two nodes u and v of T, u is said to be a descendant of v (denoted as $u \leq_T v$) if, and only if, $u \in V(T_v)$. By extension, u is said to be a strict descendant of v if, and only if, $u \leq_T v$ and $u \neq_T v$. An internal node u of T is said to be *artificial* if it has only one child (u_l). *Contracting* an artificial node u consists in merging the two adjacent edges of u in a single one by adding the edge (u_p, u_l) then removing (u_p, u), (u, u_l) and u.

Given a tree T, the height of a node u, denoted by $h(u)$, is the maximum number of edges along the path between u and any leaf $v \in L(T_u)$. The height of tree T, denoted $h(T)$, refers to the height of its root. The subset of nodes of T located at height $k \in \{0, 1, \ldots, h(T)\}$ is denoted $V_k(T) = \{u \in V(T) : h(u) = k\}$. Since leaves of T are the only nodes located at height 0, we have that $L(T) = V_0(T)$.

Definition 1 (time function, dated tree) A *time function* for a tree T, denoted by $\theta_T : V(T) \to \mathbb{R}^+$, associates a non-negative time to each node so that $\forall x, x' \in V(T)$, $x' <_T x \Rightarrow \theta_T(x') < \theta_T(x)$.

Definition 2 (contemporary time function) A *contemporary time function* for a tree T, is a time function associating 0 to each leaf of T.

Definition 3 (binary tree, dated tree, gene tree, species tree)

- A *binary tree* is a tree where all internal nodes but the root have outdegree two. The root T can have outdegree one.
- A *dated tree* is a tree T associated with a *time function* θ_T.
- A *species tree* S is a rooted binary tree such that each element of $\mathcal{L}(S)$ represents an extant species labeling exactly one leaf of S. A *dated species tree* (S, θ_S) is a species tree S associated with a contemporary time function θ_S.
- A *gene tree* G is a rooted binary tree whose leaves are labeled. Each leaf corresponds to a contemporary gene or to a gene that was lost. Each internal node of G represents an ancestral descendant of the ancestral gene $r(G)$, and we adopt the convention that the term *gene* refers to a node of G.

Definition 4 (time interval $I_{\theta_T}(.)$) Given a dated tree (T, θ_T), we will denote by $I_{\theta_T}(x)$ the time interval $]\theta_x, \theta_{x_p}[$ associated to a branch (x_p, x) of T.

From now on, we consider a species tree S and a gene tree G such that $\mathcal{L}(G) \subseteq \mathcal{L}(S)$; furthermore to help distinguish between G and S, the terms *node* and *edge* refer to G whereas their synonyms *vertex* and *branch* refer to S.

3 A general model of evolutionary scenario: the dated scenarios

When we consider a gene tree G depicting the history of some genes observed in contemporary genomes, we miss part of the history of the ancestral genes: that of the ancestral genes belonging to subtrees whose leaves all went extinct. For instance in Figure 1 (Right), the leaves hanging from the path between $r(G^o)$ and u and from the path between w and d_1 left no trace in the contemporary genomes since all these genes were lost.

Definition 5 ((non) contemporary leaves) Given a gene tree G^o, the set $L(G^o)$ is partitioned in two sets: the set of contemporary leaves $L_{\mathbb{C}}(G^o)$, and the set of non contemporary leaves $L_{\mathbb{L}}(G^o)$. The latter correspond to loss events.

Definition 6 (Full gene tree, traceable gene tree) Given two gene trees G and G^o, G^o is said to be a *full gene tree* for G – and conversely G is said to be the *traceable gene tree* of G^o – if, and only if:

1. $L_{\mathbb{L}}(G) = \emptyset$;
2. there exists an isomorphism preserving labels between G and the tree obtained from G^o by deleting all the leaves that are not in $L_{\mathbb{C}}(G^o)$ and contracting all artificial nodes created by the deletion.

Hence, the history of a full gene encompasses the history of its traceable gene tree. For the rest of this article, a *gene tree* means a *traceable gene tree.*

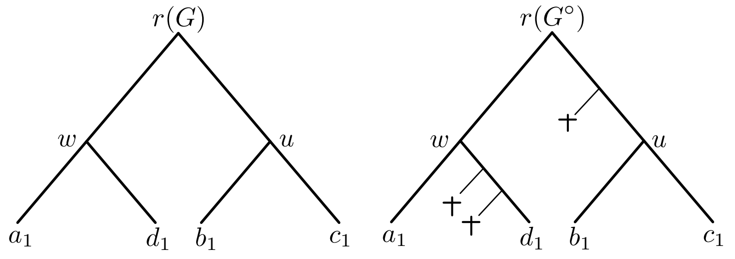


Fig. 1: (Left) A traceable gene tree G with four leaves a_1, b_1, c_1, and d_1 (say respectively belonging to contemporary species A, B, C, and D). (Right) A full gene tree G^o for G, where $L_{\mathbb{C}}(G^o) = \{a_1, b_1, c_1, d_1\}$, and $L_{\mathbb{L}}(G^o)$ consists of the three leaves labelled by the $+$ symbol.

Before detailing the general model, we precise some underlying assumptions on the evolution of species and genes. First, we assume that the evolution of the contemporary species is exactly depicted by the phylogeny S and ignore speciations where one or both lineages went extinct or are not depicted in S. Yet, as considered by Szöllősi et al [46,45], it is possible that other species interact with the ones depicted in S via, for example, transfer events. This consideration can be incorporated in the model presented here without invalidating the results described hereafter. However, to keep the description as simple as possible, results given in this paper do not take this fact into account. Second, we focus on duplication, transfer, loss, and speciation events, respectively denoted $\mathbb{D}$, $\mathbb{T}$, $\mathbb{L}$, and $\mathbb{S}$, to document the gene history. Third, we assume that $\mathbb{D}, \mathbb{T},$ and $\mathbb{L}$ events happen strictly within a lineage, *i.e.*, between speciations, or between a speciation node and a leaf node.

Reconciliation models seek to describe the evolutionary scenario that lead to the observed gene tree G knowing the corresponding species tree. The outcome of such a model is an embedding of G into S

together with evolutionary events depicting the evolution of the ancestral gene $r(G)$ and its descendants as they reside in the genomes of the species in S. This result yields a full gene tree G^o (depending on the software this tree is given explicitly (e.g. [13]) or implied from the embedding of G in S and the outputted set of events (e.g. [48]).

Remark 1 To be biologically meaningful, the evolution of a full gene tree G^o along a species tree S has to respect the following constraints:

1. Each contemporary gene is a leaf of G^o and is associated to the corresponding species of S in which this gene is collected. Such an association is denoted $\mathbb{C}$. Each speciation event in G^o (denoted $\mathbb{S}$) happens at an internal node of S.
2. The evolution of G^o along S goes forward in time from the common ancestral gene toward the contemporary species, i.e. an ancestral gene always existed before its descendants.
3. Each $\mathbb{S}, \mathbb{D},$ and $\mathbb{T}$ event gives birth to exactly two genes.
4. $\mathbb{L}$ events are taken into account explicitly.
5. Each $\mathbb{T}$ event is *locally consistent*, i.e. it happens between two co-existing species.

To get closer to models and algorithms of the reconciliation literature, the constraints of Remark 1 are formalized below in a more operational definition.

Definition 7 (full dated scenario) Consider a dated binary species tree $(S,\ \theta_S)$, and a full binary gene tree G^o. Let M be a function that maps each node u° of G^o to a pair (vertex x of S, time t) and let us denote by M_v and M_t the associated functions returning only the vertex, respectively the time, of the M mapping, i.e. if $M(u^\circ) = (x, t)$, $M_v(u^\circ) = x$ and $M_t(u^\circ) = t$. The mapping M is a *full dated scenario* for the triple (G^o, S, θ_S) if, and only if, for each node u° of G^o exactly one of the following, mutually exclusive, cases occurs (where $(x, t) := M(u^\circ)$):

1. $u^\circ \in L_{\mathbb{C}}(G^o)$, $x \in L(S)$, $s(x) = s(u^\circ)$, and $M_t(u^\circ) = 0$. ($\mathbb{C}^o$ event)
2. $u^\circ \in L_{\mathbb{L}}(G^o)$ and $M_t(u^\circ) \in I_{\theta_S}(x)$. ($\mathbb{L}^o$ event)
3. $\{M_v(u^\circ_l), M_v(u^\circ_r)\} = \{x_l, x_r\}$ and $M_t(u^\circ) = \theta_S(x)$. ($\mathbb{S}^o$ event)
4. $M_v(u^\circ_l) = M_v(u^\circ_r) = x$;
   $M_t(u^\circ) \in I_{\theta_S}(x)$ and $M_t(u^\circ) > max(M_t(u^\circ_l), M_t(u^\circ_r))$. ($\mathbb{D}^o$ event)
5. $M_v(u^\circ_l) = x$, $M_v(u^\circ_r) = y$, $x \neq y$,
   $M_t(u^\circ) \in I_{\theta_S}(x) \cap I_{\theta_S}(y)$ and $M_t(u^\circ) > max(M_t(u^\circ_l), M_t(u^\circ_r))$. ($\mathbb{T}^o$ event)

See Figure 2 for an illustration. Each of the above cases is labeled by the symbol corresponding to the associated evolutionary event and results from the mapping of a gene node u°– and sometimes of its children – into the species tree. For the sake of simplicity, we will say, for instance, that $M(u^\circ)$ is a $\mathbb{D}^o$ event when in fact we mean that the mappings of u° and its children verify the constraints of the line labeled $\mathbb{D}^o$. Note that in the $\mathbb{T}^o$ event case, the transferred gene is chosen, without loss of generality, to be the child u°_r of u°.

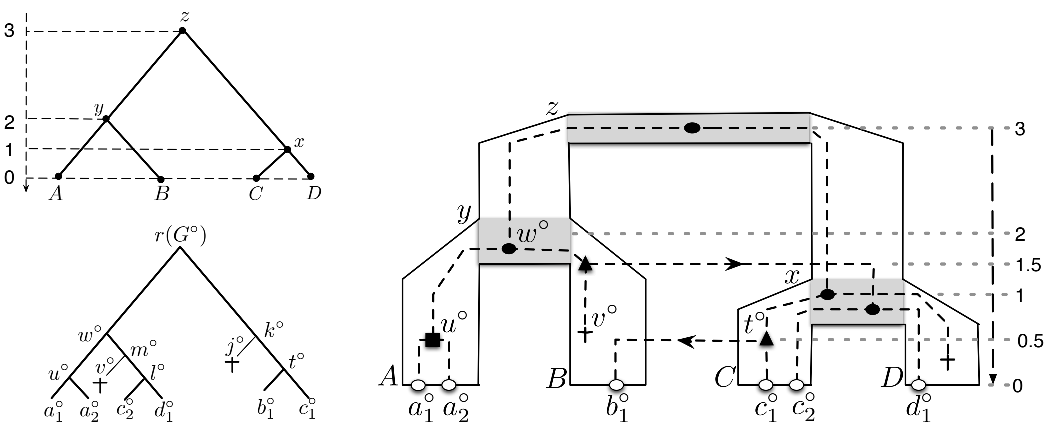
Claim Any full dated scenario M for (G^o, S, θ_S) respects the five constraints of Remark 1.

The proof of the claim is deferred in the appendix.

The constraints on $\mathbb{T}^o$ events in Definition 7 increase the complexity of the problem [24,48], and it is not a surprise that several algorithms and software not ensuring the time consistency of transfers have been proposed [7,24].

Note that because $\mathbb{D}, \mathbb{T}, \mathbb{L},$ and $\mathbb{S}$ events are associated to times in $\mathbb{R}$, there is an infinite number of full dated scenarios between a dated binary species tree S and a full binary gene tree G^o. Moreover, the full dated scenario is defined via G^o and not G, and given a traceable gene tree there is an infinity of full gene trees for it (having *e.g.* lost subtrees of arbitrary size). Relying on G^o rather than G we hence assume that the position of loss events is known, which – except for simulated data sets – is not the case *before* having reconciled G and S. Finally, following [19], the full dated scenario can start at any location in S, since the root of G^o can be mapped on any vertex of S and not only on its root vertex. The following definitions formalize these points.

Definition 8 (dated scenario) A *dated scenario* for the triplet (G, S, θ_S), is a pair (G^o, M) where G^o is a full gene tree for G and M is a full dated scenario for (G^o, S, θ_S).



(a) A dated species tree (S, θ_S) (above) and a full gene tree G^o (below)

(b) A scenario M for G^o and (S, θ_S)

Fig. 2: A scenario M for (G^o, S, θ_S), where the fives events of Definition 7 are represented as follows: a $\mathbb{C}^o$ event ($\circ$), an $\mathbb{L}^o$ event ($+$), an $\mathbb{S}^o$ event ($\bullet$) a $\mathbb{D}^o$ event ($\blacksquare$), and a $\mathbb{T}^o$ event ($\blacktriangle$). The node a_1° is a $\mathbb{C}^o$ event at vertex $M(a_1^\circ) = A$ and time $\theta_{G^o}(a_1^\circ) = 0$, the node v° is an $\mathbb{L}^o$ event at vertex $M(v^\circ) = B$, the node w° is an $\mathbb{S}^o$ event at vertex $M(w^\circ) = y$ and time $\theta_{G^o}(w^\circ) = 2$, the node u° is a $\mathbb{D}^o$ event along the branch ending at vertex $M(u^\circ) = A$, and the node t° is a $\mathbb{T}^o$ event at time $\theta_{G^o}(t^\circ) = 0.5$ from the branch above the vertex $M(t^\circ) = C$ toward the branch above the vertex $M(b_1^\circ) = B$. Note that speciations in part (b) of the figure are represented by grey boxes to show in detail what happens at those places, but the model assigns a precise time to each of them (namely, 3, 2 and 1 respectively, for nodes z, y and x).

1

Definition 9 (cost of a dated scenario) The *cost* of a dated scenario (G^o, M) for (G, S, θ_S) is denoted $C(G^o, M) = d\delta + t\tau + l\lambda$, where δ, τ, and λ respectively denote the positive cost of $\mathbb{D}^o$, $\mathbb{T}^o$, and $\mathbb{L}^o$ events, and d, t, and l denote the respective numbers of these events in (G^o, M).

Definition 10 (optimal dated scenario cost) The *optimal dated scenario cost* for (G, S, θ_S) is $C_{opt}(G, S, \theta_S) = \min\{C(G^o, M) : (G^o, M) \text{ is a dated scenario for } (G, S, \theta_S)\}$.

Note that the above definition implicitly assumes that speciation events have null cost as is often done in the literature. The central optimization problem considered in this paper is called *Most Parsimonious Scenario* (MPS for short) defined below.

Problem 1 MPS problem.
INPUT: a dated species tree (S, θ_S), a binary traceable gene tree G and strictly positive costs for $\mathbb{D}^o$, $\mathbb{T}^o$, and $\mathbb{L}^o$ events, denoted δ, τ, and λ, respectively.
OUTPUT: a dated scenario (G^o, M) of optimal cost for (G, S, θ_S).

A dated scenario that is a solution for the MPS problem is said to be *Most Parsimonious* (MP for short) for (G, S, θ_S).

3.1 Basic Dated Scenarios

As detailed above, a full dated scenario relies on a *full* gene tree G^o including the contemporary genes of G as well as additional lost genes. Among all the possible full gene trees for a traceable gene tree G, some cannot be involved in an MP solution as they contain multiple losses that can be summarized by a single one. The following definitions allow to characterized the subset of G^o trees, hence the subset of dated scenarios, that it suffices to consider for solving MPS.

Definition 11 (dead, lost and traceable genes) Consider a full gene tree G^o, any node u° of G^o is in exactly one of the three following cases:

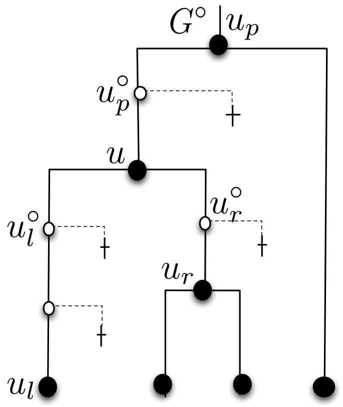


Fig. 3: Illustration of the notations given in Definition 11. Dead genes, ghosts genes, and traceables genes are represented by a cross †, a white circle ∘, and a black circle •, respectively.

1. u° is a *dead gene* if, and only if, neither itself nor one of its descendants is contemporary.
2. u° is a *ghost gene* if, and only if, it has two children and exactly one of them is dead.
3. u° is a *traceable gene* if, and only if, u° is neither a dead gene nor a ghost one (i.e. (a) it is contemporary or (b) none of its children is dead).

See Figure 3 for an example.

Definition 12 (dead subtree) A node u° of G^o is said to be a *dead subtree* if, and only if, u° is a dead gene and u° is not a leaf.

When searching for an MP dated scenario, any dated scenario (G^o, M) where G^o contains dead subtrees can be ignored, as well as scenarios with ghost genes at certain positions. The subset of dated scenarios to consider for solving the MPS problem, are called *basic* and are defined as follows:

Definition 13 (Basic dated scenario) A dated scenario (G^o, M) for (G, S, θ_S) is said to be *basic* if, and only if, it respects the following constraints: (a) the root $r(G^o)$ is a traceable gene. (b) each dead gene of G^o is a leaf. (c) each ghost gene u° is either an $\mathbb{S}^o$ event or a $\mathbb{T}^o$ event that transfers the non dead child of u° (*i.e.*, a $\mathbb{T}^o$ event where u_r° is the non-dead child of u°).

Property 1 Given (G, S, θ_S) and strictly positive costs δ, τ, and λ, any most parsimonious dated scenario (G^o, M) for (G, S, θ_S) is basic.

The proof of the property can be found in the appendix.

3.2 Equivalent dated scenario

Given a dated scenario (G^o, M) for (G, S, θ_S), an infinite number of alternate scenarios can be derived from it by infinitesimal changes in the time of events (that is by changes in M_t). Slight date alterations lead to scenarios that correspond to identical evolutionary histories, except for the precise timing of the events in the species tree branches. Informally, we consider that two dated scenarios for (G, S, θ_S) are equivalent if they are based on the same full gene tree (up to isomorphism) and map each of its nodes on the same vertex of S. Note that, as we shall see, this vertex mapping constraint is sufficient to also ensure that each vertex of the full gene tree is associated to the same event type among the equivalent dated scenarios.

Definition 14 (equivalent (full) dated scenarios)

– Two *full dated scenarios* M and N for (G^o, S, θ_S) are said to be *equivalent* if and only if $\forall u^\circ \in G^o, M_v(u^\circ) = N_v(u^\circ)$.

– Two *dated scenario* (G^o, M) and (H°, N) for (G, S, θ_S) are said to be *equivalent* if, and only if G^o is identical to H° (up to isomorphism) and the full dated scenarios M and N are equivalent.

The time constraints of a full dated scenario definition (Definition 7) ensure the accordance of the vertex mapping and the time mapping of the scenario. When M and N are two full dated scenarios, these constraints can be safely ignored when determining their equivalence.

Property 2 When two full dated scenarios M and N are equivalent, they associate the same event type to each node u° of G^o.

Proof. The proof directly follows from the definition of each event of a full dated scenario (Definition 7). Indeed, for each node u° of G^o, its event type is unambiguously determined as follows by knowing G^o, S and their vertex mapping, which are, by the definition, identical among equivalent full dated scenarios:

1. $u^\circ \in L_{\mathbb{C}}(G^o)$ $(\mathbb{C}^o)$
2. $u^\circ \in L_{\mathbb{L}}(G^o)$ $(\mathbb{L}^o)$
3. $\{M_v(u^\circ_l), M_v(u^\circ_r)\} = \{x_l, x_r\}$ ($\mathbb{S}^o$ event)
4. $M_v(u^\circ_l) = M_v(u^\circ_r) = x;$ ($\mathbb{D}^o$ event)
5. $M_v(u^\circ_l) = x$, $M_v(u^\circ_r) = y$, $x \neq y;$ ($\mathbb{T}^o$ event)

□

Property 3 Two full dated scenarios that are equivalent have the same cost.

Proof. This directly follows from Property 2, which ensures that full dated scenarios that are equivalent have the same number of each kind of event. □

4 An intermediate model: sliced scenarios

As seen in the previous section, the time mapping of a full dated scenario M for (G^o, S, θ_S) allows to ensure the global time consistency of the proposed evolutionary history. However, as already pointed out, slight modifications of the proposed time mapping can lead to equivalent dated scenarios. This pinpoints the fact that the exact time position of an event is not so important, as long as it remains in a time interval ensuring time consistency. The latter point is further emphasized when noting that the constraints on $\mathbb{D}^o$ and $\mathbb{T}^o$ event in Definition 7 are based on the fact that the event time is included in a specific time interval. In this section, we hence introduce an intermediate model that maps each node of G^o on a chunk of a branch rather than at a precise time of this branch. Indeed, the time consistency of transfers, as defined in Remark 1, can be ensured by imposing that the donor and receiver of a transfer are within the same *time slice* (see [13,7,47] and next section). As it will be proved in this paper, this intermediate model allows to transpose our initial continuous problem in a discrete space where it is then much easier to tackle.

Definition 15 (tree subdivision) A tree T' is said to be a *subdivision* of a tree T if the recursive contraction of all artificial nodes of T' yields T (up to isomorphism).

Definition 16 (The dated subdivision of a dated species tree) Consider a dated species tree (S, θ_S), its dated subdivision is a dated species tree (S', θ') obtained as follows: starting with a copy of (S, θ_S), for each initial branch (x_p, x) of S' and for each date $t \in]\theta_S(x), \theta_S(x_p)[$, such that there is a vertex y of S with $\theta_S(y) = t$, an artificial vertex w is inserted along the branch (x_p, x) of S', with $\theta_{S'}(w) = t$.

See Figure 4 for an example of subdivision. Note that, when computing the subdivision S' for the tree S, nodes are only added and never removed. So, any node x of S is present as a node x' in S'; such a case where x and x' *coincide* is denoted hereafter by $x' \cong x$.

The dated subdivision $(S', \theta_{S'})$ of (S, θ_S), splits each branch of S into several ones; the $Br(\cdot)$ function defined below, associates each branch of S' to its corresponding branch of S. As each vertex x of a dated tree has a single parent x_p, x can unambiguously be used to identify the branch (x_p, x). The branch association defined by $Br(\cdot)$ is hence done based on a vertex association, as detailed in the following definition.

Definition 17 ($Br(\cdot)$) For a species tree S and a subdivision S' of S, let $Br : V(S') \to V(S)$ be a function that maps each vertex x' of S' onto the unique vertex x of S such that $x'_1 \leq_{S'} x' <_{S'} x'_2$, where $x'_1 \cong x$ and $x'_2 \cong x_p$ (x_p being the father of x in S).

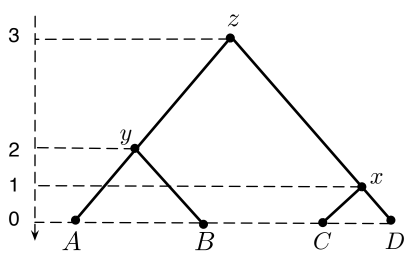
Note that $x' \cong x$ happens whenever $Br(x') = x$ and $\theta_{S'}(x') = \theta_S(x)$.

Definition 18 (Chunk) For a species tree S and a subdivision S', a chunk of the branch (x_p, x) in S is a branch (x'_p, x') in S' such that $Br(x') = x$, which implies that $I_{\theta_{S'}}(x') \subseteq I_{\theta_S}(x)$.

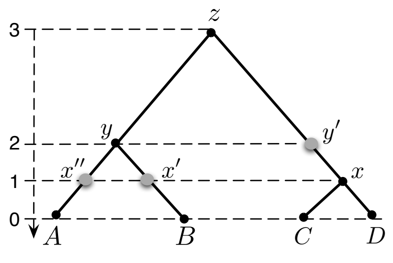
Definition 19 ($Chunk(\cdot,\cdot)$) Given a species tree S, and its dated subdivision S', let $Chunk : V(S) \times \mathbb{R} \to V(S')$ be the function that, given a vertex $x \in V(S)$ and a time $t \in [\theta_S(x), \theta_S(x_p)[$, returns the unique vertex x' such that $Br(x') = x$ and $t \in [\theta_{S'}(x'), \theta_{S'}(x'_p)[$. If the provided time t is not coherent with x, $Chunk(x,t)$ is undefined.

For instance in Figure 4b, $Br(y') = x$ and $Br(x) = x$ and the branch (z, x) of S is split in two chunks (z, y') and (y', x) in S'.

Note that by construction, we have that, for any two vertices x' and y' of S', $\theta_{S'}(x') = \theta_{S'}(y')$ if and only if $h(x') = h(y')$. Since $h(x') = h(y')$ implies that $h(x'_p) = h(y'_p) = h(x') + 1$, it follows that for any two vertices x' and y' of S', either $I_{\theta_{S'}}(x') = I_{\theta_{S'}}(y')$, or $I_{\theta_{S'}}(x') \cap I_{\theta_{S'}}(y') = \emptyset$. In the latter case, we will denote by $I_{\theta_{S'}}(x') > I_{\theta_{S'}}(y')$ the fact that any date within $I_{\theta_{S'}}(y')$ is posterior to any date within $I_{\theta_{S'}}(x')$. Using this notation, $h(x') > h(y') \Leftrightarrow I_{\theta_{S'}}(x') > I_{\theta_{S'}}(y')$.



(a) A species tree S



(b) The subdivision S' of S

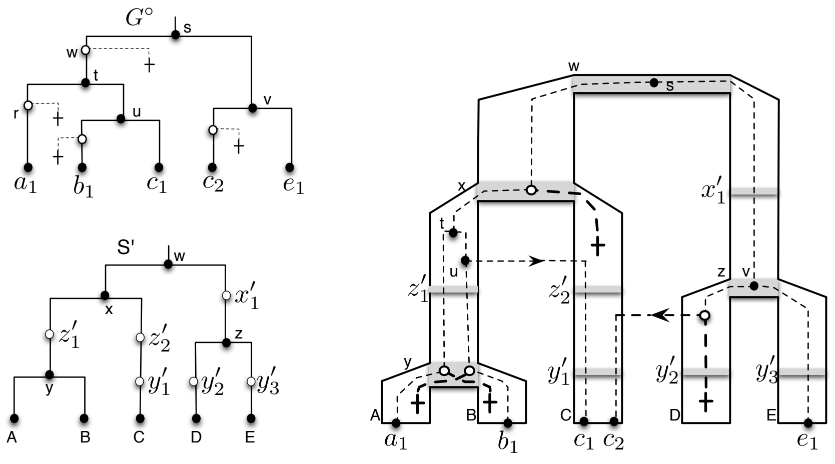
Fig. 4: A species tree S and its subdivision S'. The artificial nodes of S' are represented by gray circles and denoted y', x', and x'', where $\theta_{S'}(x) = \theta_{S'}(x') = \theta_{S'}(x'')$ and $\theta_{S'}(y) = \theta_{S'}(y')$.

The evolutionary history of a gene tree along the species tree S is described in this intermediate model by a *(full) sliced scenario* that maps each node of the (full) gene tree to a chunk of S', as formalized hereafter and illustrated in Figure 5. Since a chunk (x'_p, x') of S' is unambiguously defined by its lowest node x', sliced scenario are in practice define by a node mapping.

Definition 20 (full sliced scenario) Consider a dated binary species tree (S, θ_S), its subdivision $(S', \theta_{S'})$ and a full binary gene tree G^o. Let M'_v be a function that maps each node u° of G^o to a vertex x' of S'. The mapping M'_v is a *full sliced scenario* for (G^o, S', θ'_S) if, and only if, for each node u° of G^o exactly one of the following mutually exclusive cases occurs (where $x' := M'_v(u^\circ)))$ and $x := Br(x')$):

1. $u^\circ \in L_{\mathbb{C}}(G^o)$, $x \in L(S)$, $s(x) = s(u^\circ)$, and $\theta_{S'}(x') = 0$. ($\mathbb{C}^o$ event)
2. $u^\circ \in L_{\mathbb{L}}(G^o)$. ($\mathbb{L}^o$ event)
3. $\{Br(M'_v(u^\circ_l)), Br(M'_v(u^\circ_r))\} = \{x_l, x_r\}$ and $x' \cong x$. ($\mathbb{S}^o$ event)
4. $Br(M'_v(u^\circ_l)) = Br(M'_v(u^\circ_r)) = x$
   and $I_{\theta_{S'}}(x') \geq max\left(I_{\theta_{S'}}(M'_v(u^\circ_l)), I_{\theta_{S'}}(M'_v(u^\circ_r))\right)$. ($\mathbb{D}^o$ event)
5. $Br(M'_v(u^\circ_l)) = x$, $Br(M'_v(u^\circ_r)) = y$ and $x \neq y$
   and $I_{\theta_{S'}}(x') \subseteq I_{\theta_S}(y)$ and $I_{\theta_{S'}}(x') \geq max\left(I_{\theta_{S'}}(M'_v(u^\circ_l)), I_{\theta_{S'}}(M'_v(u^\circ_r))\right)$. ($\mathbb{T}^o$ event)

As might be expected, the definition of a sliced scenario is highly similar to that of a full dated scenario (Definition 7). However, the sliced scenario definition benefits from a simplification because cases 2, 4, 5 no longer require to check the consistency of the time and vertex mapping. Indeed, for dated scenarios we



(a) Trees G^o and S' (b) A sliced scenario (G^o, M')

Fig. 5: A full gene tree G^o, a subdivision S' of a dated species tree S and a full sliced scenario M' for (G^o, S, θ_S).

needed to check that $M_t(u^\circ) \in I_{\theta_S}(Br(x))$, while for sliced scenario we have that $I_{\theta_{S'}}(x') \subseteq I_{\theta_S}(Br(x))$ by construction. Note that, since events are not associated to a specific date, we can not yet ensure the overall time consistency of such a scenario. We only have a partial chronological ordering of the events through the time slices to which they are associated. But, as we will prove hereafter, it is always possible to assign a specific date to each event in a way that respects the slice scenario constraints and leads to a valid dated scenario.

Definition 21 Given (G, S, θ_S), a *sliced scenario* is a pair (G^o, M'_v) where G^o is a full gene tree for G and M'_v is a full sliced scenario for (G^o, S, θ_S).

The definition of the cost of a (full) sliced scenario is an immediate transposition of Definition 9, and that of an *optimal* sliced scenario for (G^o, S, θ_S) derives from Definition 10 provided for dated scenario. The *Most Parsimonious Sliced Scenario* problem is defined as follows:

Problem 2 MPSS problem.
INPUT: a dated species tree (S, θ_S), a binary traceable gene tree G, and strictly positive costs δ, τ, resp. λ for $\mathbb{D}^o$, $\mathbb{T}^o$, resp. $\mathbb{L}^o$ events.
OUTPUT: a sliced scenario (G^o, M'_v) of optimal cost for (G, S, θ_S).

Similarly to what was done in the full dated scenario section, we will now define *basic sliced scenario* and the *equivalence* between basic sliced scenarios. The next sections will resort to these definitions to prove that solving the MPSS problem allows to solve the MPS problem, thanks to a direct correspondence between the equivalent classes of sliced scenarios and those of dated scenarios.

Definition 22 (Basic sliced scenario) A sliced scenario (G^o, M'_v) for (G, S, θ_S) is said to be *basic* if, and only if, it respects the following constraints: (a) the root $r(G^o)$ is a traceable gene; (b) each dead gene of G^o is a leaf; (c) each ghost gene u° is either an $\mathbb{S}^o$ event, or alternatively a $\mathbb{T}^o$ event that transfers the non dead child of u°.

The proof of Properties 4 and 5 below are identical to those detailed for the same properties in the dated scenario context (Properties 1, 2 and 3); they are thus not repeated here.

Property 4 Given a gene tree G and strictly positive costs δ, τ, and λ, any most parsimonious sliced scenario (G^o, M'_v) for (G, S, θ_S) is such that (G^o, M'_v) is a basic sliced scenario for (G, S, θ_S).

Definition 23 (equivalent sliced scenario) Two sliced scenario (G°_1, M'_v) and (G°_2, N'_v) for (G, S, θ_S) are said to be equivalent if, and only if, G°_1 is identical to G°_2 (up to isomorphism) and $\forall u^\circ \in G^\circ_1 = G^\circ_2, Br(M'_v(u^\circ)) = Br(N'_v(u^\circ))$.

Property 5 If two sliced scenarios (G^o, M') and (G^o, N') are equivalent, they associate the same event type to each node u° of G^o and have hence the same cost.

This property arises from the fact that each event in Definition 20 is defined on the basis of $Br(\cdot)$ values and because constraints on time they impose are verified separately by both (G^o, M') and (G^o, N').

Definition 24 (Basic-up scenario) Given (G, S, θ_S), a basic sliced scenario (G^o, M'_v) for (G, S, θ_S) is said to be basic-up if each dead gene is placed in the highest possible time slice (i.e. at the same time slice as its parent node for a $\mathbb{T}^o$ event and in the next time slice for an $\mathbb{S}^o$ event).

Note that each basic sliced scenario as a unique basic-up scenario equivalent to it.

5 A tractable model of evolutionary scenario : reconciliations

In this section, we recall the *reconciliation* (or $\mathbb{DTL}$) model introduced in [13].

The atomic events of the model are: a speciation ($\mathbb{S}$), a duplication ($\mathbb{D}$), a transfer ($\mathbb{T}$), a transfer followed immediately by the loss of the non- transferred child ($\mathbb{TL}$), a speciation followed by the loss of one of the two resulting children ($\mathbb{SL}$), and a contemporary event ($\mathbb{C}$) that associates an extant gene to its corresponding species. Finally, a null event ($\varnothing$), is used to indicate that a gene lineage change of time-slice while remaining in the same branch of the species tree. (Null events are used to indicate that no events happened in a given chunk.) Note that duplication-loss events and transfer followed by the loss of the transferred gene, leave no trace and are therefore undetectable and not taken into account by this model. Such events cannot occur in parsimonious reconciliation as their removal would always lead to an alternative valid reconciliation with lower cost.

We now give the formal definition of a *reconciliation* between a gene tree G and a subdivision S' of a dated species tree (S, θ_S). This definition is adapted from the definition of a reconciliation given in [13][2].

Definition 25 (Reconciliation) Consider a gene tree G, a dated species tree (S, θ_S) and its time subdivision S'. Let α be a function that maps each node u of G onto an ordered sequence of vertices of S', denoted $\alpha(u)$. Let $\alpha_i(u)$ denote the i^{th} element of this sequence (where $1 \le i \le \ell := |\alpha(u)|$). Then α is said to be a *reconciliation* between G and (S, θ_S) if and only if exactly one of the following mutually exclusive cases occurs for each couple of node u of G and vertex $\alpha_i(u)$ of S' (where $\alpha_i(u)$ is denoted x' below):

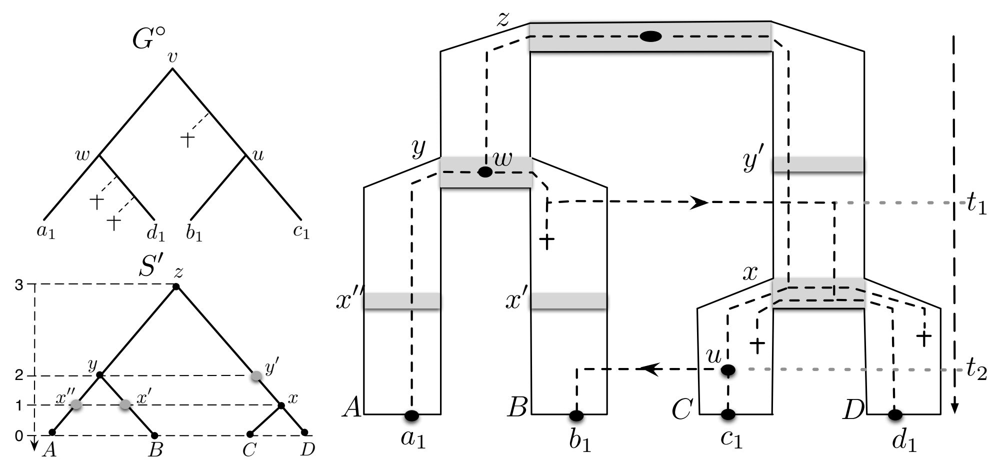
- x' is the last element of $\alpha(u)$ and exactly one of the cases below is true:
  1. $u \in L(G)$, $x' \in L(S')$ and $s(x') = s(u)$; ($\mathbb{C}$ event)
  2. x' is not artificial and $\{\alpha_1(u_l), \alpha_1(u_r)\} = \{x'_l, x'_r\}$; ($\mathbb{S}$ event)
  3. $\alpha_1(u_l) = \alpha_1(u_r) = x'$; ($\mathbb{D}$ event)
  4. $\alpha_1(u_l) = x'$, and $\alpha_1(u_r)$ is any vertex other than x' that is located at height $h(x')$; ($\mathbb{T}$ event)
- otherwise, exactly one of the cases below is true:
  5. x' is not artificial and $\alpha_{i+1}(u) \in \{x'_l, x'_r\}$; ($\mathbb{SL}$ event)
  6. $\alpha_{i+1}(u)$ is any vertex other than x' that is located at height $h(x')$; ($\mathbb{TL}$ event)
  7. x' is an artificial vertex and $\alpha_{i+1}(u)$ is its only child, and, if $u = r(G)$, $i \ne 1$. ($\varnothing$ event)

$\varnothing$ events are events modeling the fact that a gene evolving along a branch of the species tree crosses the boundary of a time slice.

We say that, for example, "$\alpha_i(u)$ *is a* $\mathbb{TL}$ *event*" if $\alpha_i(u)$ satisfies the condition given on the line labelled "$\mathbb{TL}$" in Definition 25. Note that Condition 5 of Definition 25 does not permit to have reconciliations starting with a $\varnothing$ event. A reconciliation α between a gene tree G and a dated species tree (S, θ_S) is depicted in Figure 6. Similarly to the case of sliced scenarios, a branch (v_p, v) of S' is identified by the vertex v in a reconciliation $\alpha(.)$; thus, when α maps a duplication event or the origin of a transfer to a vertex x of S', this means that the specified event happens in the branch just above x.

Note that a reconciliation between G and a dated species tree (S, θ_S) allows to map $r(G)$ on any branch of its subdivision S'. Indeed, the gene whose evolution is depicted by G may fail to be present

[2] Instead of mapping each node of G onto a sequence of *vertices* of S' (as done in Definition 25), the original model maps each edge of G onto a sequence of *branches* of S'. Since an edge of a rooted tree is univocally identified by its bottom node, these two reconciliation models are equivalent.



(a) A full gene tree G^o and a subdivided species tree S'

(b) A reconciliation α between G and S'

Fig. 6: A reconciliation α between a gene tree G and a dated species tree (S, θ_S), where S' is its subdivision and G is the traceable gene tree of G^o and losses are indicated by crosses. The reconciliation α maps the nodes d_1 and u of G as follows: $\alpha(d_1) = [x', x, D]$, where $\alpha_1(d_1) = x'$, $\alpha_2(d_1) = x$, and $\alpha_3(d_1) = D$ are respectively a $\mathbb{TL}$, an $\mathbb{SL}$, and a $\mathbb{C}$ event; $\alpha(u) = [y', x, C]$, where $\alpha_1(u) = y'$, $\alpha_2(u) = x$, and $\alpha_3(u) = C$ are respectively a $\varnothing$, an $\mathbb{SL}$, and a $\mathbb{T}$ event. Note that, although the gene lineage (u, b_1) starts on branch (x, C) after the $\mathbb{T}$ event of its parent u, this branch is not in the $\alpha(b_1)$ sequence and $\alpha(b_1) = [B]$.

in the ancestral lineage represented by $r(S')$. Note also that $\mathbb{T}$ and $\mathbb{TL}$ events only happen between two branches that are in a same slice, hence only time-consistent scenarios are accounted by this tractable model. The next property contains a list of observations that immediately follows from the constrains imposed in Definition 25.

Property 6 Given a reconciliation α between a gene tree G and a dated species tree (S, θ_S):

1. for any node u of G and indices $i < j \in \{1, 2, \ldots, |\alpha(u)|\}$, $h(\alpha_i(u)) \geq h(\alpha_j(u))$.
2. For each internal node u of G, $h(\alpha_\ell(u)) \geq max(h(\alpha_1(u_l)), h(\alpha_1(u_r)))$.
3. For any $u \in V(G)$, $\alpha_i(u)$ is the last element of the sequence $\alpha(u)$ (*i.e.*, $i = \ell$) if and only if it is a $\mathbb{C}$, $\mathbb{S}$, $\mathbb{D}$ or $\mathbb{T}$ event. Consequently, $i < \ell$ if and only if it is a $\varnothing$, an $\mathbb{SL}$ or a $\mathbb{TL}$ event.
4. For any $u \in V(G)$, if $\alpha_i(u)$ is a $\varnothing$ event then let k be the index of the following non $\varnothing$ event in $\alpha(u)$, then $Br(\alpha_i(u)) = Br(\alpha_{i+1}(u)) = \ldots = Br(\alpha_k(u))$. Moreover, if $k < \ell$ (*i.e.*, $\alpha_k(u)$ is not the last event in the sequence), then $Br(\alpha_k(u)) \neq Br(\alpha_{k+1}(u))$.
5. For any node u, if $\alpha_i(u)$ is a non $\varnothing$ event, then let $\mathbb{E}$ be the preceding non $\varnothing$ event in $\alpha(u)$ or be $\alpha_\ell(u_p)$ otherwise[3], and let $k = h(\mathbb{E}) - h(\alpha_i(u))$.
   (a) If $\mathbb{E}$ is an $\mathbb{S}$ or an $\mathbb{SL}$ then there are exactly $k - 1$ $\varnothing$ events just preceding $\alpha_i(u)$, that is $\alpha_{i-(k-1)}(u), \ldots, \alpha_{i-1}(u)$ are $\varnothing$ events and either $i - (k - 1) = 1$ or $\alpha_{i-(k-2)}(u)$ is $\mathbb{E}$.
   (b) when $\mathbb{E}$ is a $\mathbb{D}$, $\mathbb{T}$ or $\mathbb{TL}$ event, there are exactly k $\varnothing$ events just preceding $\alpha_i(u)$, that is $\alpha_{i-k}(u), \ldots, \alpha_{i-1}(u)$ are $\varnothing$ events and either $i - k = 1$ or $\alpha_{i-(k-1)}(u)$ is $\mathbb{E}$.

Reconciliations, as sliced scenarios, mapped nodes of G on chunks of S. A chunk (x'_p, x') can be identified either by the vertex of x' or by its corresponding vertex $x = Br(x')$ of S and its time slice $h(x')$. The reconciliation definition can thus be reformulated as follows:

Definition 26 (Reconciliation via $Br(\cdot)$) Consider a gene tree G, a dated species tree (S, θ_S) and its time subdivision S'. Let α be a function that maps each node u of G onto an ordered sequence of vertices of S', denoted $\alpha(u)$. Let $\alpha_i(u)$ denote the i^{th} element of this sequence (where $1 \leq i \leq \ell := |\alpha(u)|$). Then α is said to be a reconciliation between G and (S, θ_S) if and only if exactly one of the following

[3] if u is the root of G and $\alpha_i(u)$ is the first non $\varnothing$ event of $\alpha(u)$ then there is no preceding element, the property does not apply.

mutually exclusive cases occurs for each couple of node u of G and vertex $\alpha_i(u)$ of S' (with $x' := \alpha_i(u)$ and $x := Br(x')$):

- x' is the last element of $\alpha(u)$, and exactly one of the cases below is true:
  1. $u \in L(G)$, $x \in L(S)$, $s(x) = s(u)$ and $h(x') = 0$; ($\mathbb{C}$ event)
  2. $x' \cong x$, $\{Br(\alpha_1(u_l)), Br(\alpha_1(u_r))\} = \{x_l, x_r\}$ and $h(x') = h(\alpha_1(u_l)) + 1 = h(\alpha_1(u_r)) + 1$; ($\mathbb{S}$ event)
  3. $Br(\alpha_1(u_l)) = Br(\alpha_1(u_r)) = x$ and $h(x') = h(\alpha_1(u_l)) = h(\alpha_1(u_r))$; ($\mathbb{D}$ event)
  4. $Br(\alpha_1(u_l)) = x$, $Br(\alpha_1(u_r)) = y$ and $x \neq y$ and $h(x') = h(\alpha_1(u_l)) = h(\alpha_1(u_r))$; ($\mathbb{T}$ event)
- otherwise, exactly one of the cases below is true:
  5. $x' \cong x$ and $Br(\alpha_{i+1}(u)) \in \{x_l, x_r\}$ and $h(x') = h(\alpha_{i+1}(u)) + 1$; ($\mathbb{SL}$ event)
  6. $Br(\alpha_{i+1}(u)) = y$ and $x \neq y$ and $h(x') = h(\alpha_{i+1}(u))$; ($\mathbb{TL}$ event)
  7. x' is artificial, $Br(\alpha_{i+1}(u_r)) = Br(\alpha_i(u))$ and $h(\alpha_i(u_r)) = h(\alpha_{i+1}(u)) + 1$ and, if $u = r(G)$, $i \neq 1$. ($\varnothing$ event)

As illustrated in Figure 6 the full gene tree G^o is a by-product of the reconciliation α. Starting from G it suffices to add a dead gene and a ghost node each time a $\mathbb{SL}$ or a $\mathbb{TL}$ event is encountered in α. Apart from the $\varnothing$ events, all elements of a reconciliation α hence map a node of G^o on S'. As we will prove hereafter, $\varnothing$ events are only here to connect the dots, that is the other reconciliation events, allowing to solve the MPR problem in polynomial time [13], but are useless to characterize a reconciliation since they are imposed once all other events are known. This leads us to the notion of compact reconciliation or *c-reconciliation* in short:

Definition 27 (c-reconciliation) Consider a gene tree G, a dated species tree (S, θ_S) and its time subdivision S'. Let $\bar{\alpha}$ be a function that maps each node u of G onto an ordered sequence of vertices of S', denoted $\bar{\alpha}(u)$. Let $\bar{\alpha}_j(u)$ denote the j^{th} element of this sequence (where $1 \leq j \leq \eta := |\bar{\alpha}(u)|$). Then $\bar{\alpha}$ is said to be a *c-reconciliation* between G and (S, θ_S) if and only if exactly one of the following mutually exclusive cases occurs for each couple of node u of G and vertex $\bar{\alpha}_j(u)$ of S' (with $x' := \bar{\alpha}_j(u)$ and $x := Br(x')$):

- x' is the last element of $\bar{\alpha}(u)$, and exactly one of the cases below is true:
  1. $u \in L(G)$, $x \in L(S)$, $s(x) = s(u)$ and $h(x') = 0$; ($\mathbb{C}$ event)
  2. $x' \cong x$, $\{Br(\bar{\alpha}_1(u_l)), Br(\bar{\alpha}_1(u_r))\} = \{x_l, x_r\}$ ($\mathbb{S}$ event)
  3. $Br(\bar{\alpha}_1(u_l)) = Br(\bar{\alpha}_1(u_r)) = x$ and $h(x') \geq max(h(\bar{\alpha}_1(u_l)), h(\bar{\alpha}_1(u_r)))$. ($\mathbb{D}$ event)
  4. $Br(\bar{\alpha}_1(u_l))) = x$, $Br(\bar{\alpha}_1(u_r))) = y$ and $x \neq y$, $I_{\theta_{S'}}(x') \subseteq I_{\theta_S}(y)$ and $h(x') \geq max(h(\bar{\alpha}_1(u_l)), h(\bar{\alpha}_1(u_r)))$; ($\mathbb{T}$ event)
- otherwise, exactly one of the cases below is true:
  5. $x' \cong x$ and $Br(\bar{\alpha}_{j+1}(u)) \in \{x_l, x_r\}$; ($\mathbb{SL}$ event)
  6. $Br(\bar{\alpha}_{j+1}(u)) = y$ and $x \neq y$ and $I_{\theta_{S'}}(x') \subseteq I_{\theta_S}(y)$, $h(x') \geq h(\bar{\alpha}_{j+1}(u))$. ($\mathbb{TL}$ event)

For example, for the reconciliation α depicted in Figure 6, we have that $\bar{\alpha}(u) = [x, C]$. Let us introduce some few functions that will allow to prove the that there is a simple bijective transformation between reconciliations and c-reconciliations.

Definition 28 ($ArtificialPath(\cdot,\cdot,\cdot)$; $ArtificialPath(\cdot,\cdot,\cdot,\cdot,\cdot)$) Given the time subdivision S' of a dated species tree (S, θ_S) and a vertex v of S, $ArtificialPath(S', S, v)$ returns the sequence of artificial vertices of S' constituting the path from v'_a to v' in S' where $v' \cong v$ and $v'_a \cong v_p$ (note that in S' v'_a is thus an ancestor of v'). Moreover, the function $ArtificialPath(S', h_{max}, h_{min}, S, v)$ returns the subsequence of those artificial vertices having a height in S' included in $]h_{min}, h_{max}]$ (note that vertices are still returned by decreasing height).

Transforming a reconciliation into a compact one can be done by resorting to a trivial function, called $f_{\alpha 2\bar{\alpha}}$, that for each $u \in V(G)$, removes from $\alpha(u)$ each element associated to a $\varnothing$ event, thus obtaining the corresponding $\bar{\alpha}(u)$. Note that $\bar{\alpha}(u)$ is not empty as the sequence $\alpha(u)$ ends with an event different from a $\varnothing$ event. Inversely, transforming a compact reconciliation into a reconciliation can be done by resorting to the function $f_{\bar{\alpha}2\alpha}$ detailed in Algorithm 1 that inserts artificial nodes, corresponding to $\varnothing$ events, between consecutive elements of $\bar{\alpha}$ (u) so that the output mapping $\alpha(u)$ satisfies Definition 25.

Algorithm 1 $f_{\bar{\alpha}2\alpha}(\bar{\alpha}, G, (S, \theta_S))$ Given a compact reconciliation $\bar{\alpha}$ between a gene tree G and a dated species tree (S, θ_S), this function computes a reconciliation α between these trees such that, up to $\varnothing$ events, the two reconciliations represent the same events.

```
1: $\alpha(r(G)) \leftarrow [\bar{\alpha}(r(G))]$
2: for all $u \in V(G)$ do
3:     // for any non root node $u$, to place $\varnothing$ events we need to know the mapping for $u_p$
4:     if $u \neq r(G)$ then
5:         $A_{\bar{\alpha}} \leftarrow concatSeq(\bar{\alpha}_\eta(u_p), \bar{\alpha}(u))$
6:     else
7:         $A_{\bar{\alpha}} \leftarrow \bar{\alpha}(u)$
8:     end if
9:     //we can now add $\varnothing$ events between each node mapping of $A_{\bar{\alpha}}$ based on its height and event type
10:    //$A_{\bar{\alpha}}[1]$ is skipped as it is either $r(G)$ (and has no $\varnothing$ event above it) or $\bar{\alpha}_\eta(u_p)$ (handled when processing $u_p$)
11:    for $k$ from 2 to $|A_{\bar{\alpha}}|$ do
12:        if $A_{\bar{\alpha}}[k-1]$ is an $\mathbb{S}$ or an $\mathbb{SL}$ event in $\bar{\alpha}$ then
13:            $h_{max} = h(A_{\bar{\alpha}}[k-1]) - 1$
14:        else
15:            $h_{max} = h(A_{\bar{\alpha}}[k-1]))$
16:        end if
17:        $\varnothing_{u,k} \leftarrow ArtificialPath(S', h_{max}, h(A_{\bar{\alpha}}[k]), S, Br(A_{\bar{\alpha}}[k]))$
18:        $\alpha(u) \leftarrow concatSeq(\alpha(u), \varnothing_{u,k}, A_{\bar{\alpha}}[k])$
19:    end for
20: end for
21: return $\alpha$
```

Property 7 (Bijection between reconciliations and c-reconciliations) There is an event-preserving bijection between reconciliations and c-reconciliations.

The cost of a (compact) reconciliation can be defined along the same lines as those for dated and sliced scenarios due to the correspondence of the event kinds in all these models.

Definition 29 ((optimal) reconciliation cost) The *cost* of a (compact) reconciliation α between a gene tree G and a dated species tree (S, θ_S) is denoted $cost(\alpha)$ and is defined as $d\delta + t\tau + l\lambda$, where δ, τ and λ respectively denote the cost of each $\mathbb{D}$, $\mathbb{T}$, and $\mathbb{L}$ event, and d, t, and l denote the number of these events in α. It is important to note that the cases $\mathbb{TL}$ (resp. $\mathbb{SL}$) in the definitions of a (compact) reconciliation count as a $\mathbb{T}$ (resp. $\mathbb{S}$) event plus an $\mathbb{L}$ event. The *optimal reconciliation cost*, is defined as:

$$C(G, S) := \min\{cost(\alpha) : \alpha \text{ is a reconciliation between } G \text{ and } (S, \theta_S)\}.$$

Note that a reconciliation and its *corresponding* c-reconciliation (*i.e.*, obtained by the function $f_{\alpha 2\bar{\alpha}}$) have the same cost since $\varnothing$ events are ignored in the cost definition. Indeed, recall that they are just fake events modeling the fact that a gene evolving along a branch of the species tree crosses the boundary of a time slice.

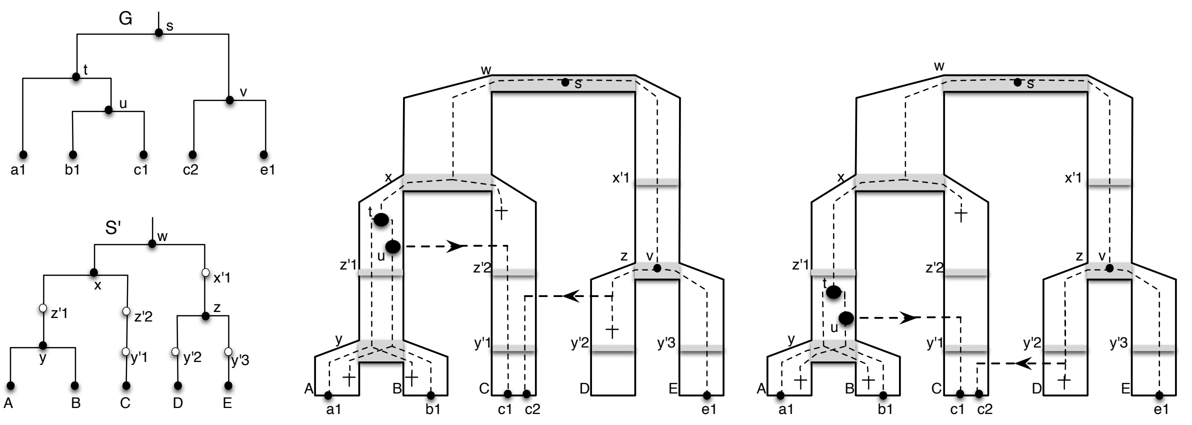
We now define an optimization problem called *Most Parsimonious Reconciliation* (MPR for short) which is the central problem solved by the algorithm in Doyon et al [13]. Many other algorithms have been proposed on similar discrete problems [33, 50, 7, 34, 20, 48, 10, 3].

Problem 3 MPR problem
INPUT: a dated species tree (S, θ_S), a traceable gene tree G, and costs δ, τ, resp. λ for $\mathbb{D}$, $\mathbb{T}$, resp. $\mathbb{L}$ events.
OUTPUT: a (compact) reconciliation α between trees G and (S, θ_S) of optimal cost, $C(G, S)$.

A reconciliation that is a solution for the MPR problem is said to be *Most Parsimonious* (MP for short) for (G, S, θ_S). As shown below, and similarly to the sliced and dated frameworks, any MPR is a basic reconciliation.

Definition 30 (Basic reconciliation) A reconciliation α between trees G and a dated species tree (S, θ_S) is said to be *basic* if and only if $\alpha_1(r(G))$ is not a $\mathbb{TL}$ or an $\mathbb{SL}$ event.

This definition is simpler than those for dated scenarios (section 3) and sliced scenarios (section 4) since reconciliations handle traceable gene trees, that is they don't have to position explicitly dead genes. Hence no equivalent of points *(b)* and *(c)* from Definitions 13 and 22 is needed here.



(a) Trees G and S' (b) Reconciliation $\bar{\alpha}^1$ (c) Reconciliation $\bar{\alpha}^2$

Fig. 7: Equivalent reconciliations between a gene tree G and a dated species tree (S, θ_S). (a) A gene tree G and a subdivision S' of a species tree, where each contemporary gene is denoted by the lower case letter corresponding to the upper case letter of the species to which it belongs (*e.g.*, gene a_1 belongs to species A). (b,c) $\bar{\alpha}^1$ and $\bar{\alpha}^2$ are two equivalent reconciliations.

Property 8 Given (G, S, θ_S) and strictly positive costs δ, τ, resp. λ for duplication, transfer, resp. loss events, any most parsimonious (compact) reconciliation α between G and (S, θ_S) is basic.

Proof. Let us prove this by contradiction. Suppose that we have a non-basic MP reconciliation α, then let β be the reconciliation such that i) $\beta(u) = \alpha(u)$ for any node $u \in V(G)\backslash r(G)$ and ii) $\beta(r(G))$ is equal to the last element of $\alpha(r(G))$. Note that, by definition of a reconciliation, $\beta(r(G))$ is not a $\varnothing$ event, and β also satisfies the definition of a reconciliation for each of the remaining elements (inherited from α). Hence, β would be a reconciliation with an $\mathbb{L}$ event less than α, that is would be of lower cost than α (since $\lambda > 0$), a contradiction. The proof for a compact reconciliation is identical. $\square$

5.1 Equivalence classes for reconciliations

As for sliced scenarios, two (compact) reconciliations can be non-identical with regards to the subdivided species tree S' but still yield the same set of events with regards to the original species tree S. Roughly speaking, two or more (compact) reconciliations are equivalent when each event occurs on (possibly different) vertices of S' that are all located on the same branches of S in both (compact) reconciliations (see Figure 7 for an example). More formally:

Definition 31 (Class of equivalent (compact) reconciliations) Two compact reconciliations $\bar{\alpha}^1$ and $\bar{\alpha}^2$ between a gene tree G and a dated species tree (S, θ_S) are said to be *equivalent*, denoted $\bar{\alpha}^1 \equiv \bar{\alpha}^2$, if and only if for each node u of G the ordered subsequences $\bar{\alpha}^1(u)$ and $\bar{\alpha}^2(u)$ have the same length η and $Br(\bar{\alpha}^1_k(u)) = Br(\bar{\alpha}^2_k(u))$ for each index $k \in \{1, 2, \ldots, \eta\}$.
Two reconciliations (compact or not) are equivalent if their compact representations are.

Since, for a given (G, S, θ_S), equivalence is based on the S mapping (through the $Br(\cdot)$ function in Definition 31), each (compact) reconciliation equivalence class can be characterized by a sequence mapping M_s that associates each node of $V(G)$ to an ordered sequence of vertices in $V(S)$. Different reconciliations in a same class M_s will only differ on the precise chunk in branches of S on which they map nodes of G. The next lemma states that equivalent reconciliations indeed have $\bar{\alpha}$ mappings that induce identical events and have hence the same reconciliation cost.

Lemma 1 *If two compact reconciliations $\bar{\alpha}^1$ and $\bar{\alpha}^2$ between a gene tree G and a dated species tree (S, θ_S) are equivalent then: for each node $u \in V(G)$ and each index $k \in \{1, 2, \ldots, \eta\}$, where $\eta := |\bar{\alpha}^1(u)| = |\bar{\alpha}^2(u)|$, the vertices $x' = \bar{\alpha}^1_k(u)$ and $y' = \bar{\alpha}^2_k(u)$ are such that i) $Br(x') = Br(y')$; ii) the same event type is associated to x' and y'; iii) $cost(\bar{\alpha}^1) = cost(\bar{\alpha}^2)$.*

Proof. Point i) is ensured by the definition of equivalent compact reconciliations. Point ii) follows from the fact that the event type associated to any $\bar{\alpha}_k(u)$ (Definition 27) is imposed by $Br(\bar{\alpha}_.(.))$ which by definition of equivalence is the same for two equivalent reconciliations (other constraints on height only ensure the validity of the compact reconciliation which is already ensured here since $\bar{\alpha}^1$ and $\bar{\alpha}^2$ are valid reconciliations). Point *iii)* directly follows from point *ii)*. □

6 Equivalence of the MP dated scenario and MP sliced scenario problems

The aim of this section is to prove that a most parsimonious full dated scenario can be found by searching for a most parsimonious sliced scenario. Finding an object of the latter kind is easier since this requires to explore a discrete time space rather than a continuous one. This proof relies on a bijection between the equivalent classes partitioning the two search spaces.

Definition 32 (Sets $\mathcal{C}$ and $\mathcal{C}'$ of equivalence classes) The equivalence relationship between two full dated scenarios (Definition 14) defines a set $\mathcal{C}$ of *dated classes*, each containing equivalent basic dated scenarios. This partitions the space of basic dated scenarios for a given (G, S, θ_S).
Similarly, on the basis of Definition 23, we denote by $\mathcal{C}'$ the set of equivalence classes defined on basic sliced scenarios for a given (G, S, θ_S). The latter classes are called *sliced classes.*

Remember that, by definition, two full dated scenarios (G_1^o, M_1) and (G_2^o, M_2) are equivalent if, and only if, G_1^o is identical to G_2^o (up to isomorphism) and any node $u^\circ \in G_1^o = G_2^o$ is mapped on the same branch of the species tree S by the two full scenarios. The same holds for equivalent sliced scenarios (G_1^o, M_v') and (G_2^o, N_v'), since Definition 23 states that, for any $u^\circ \in G^o$, $Br(M_v'(u^\circ)) = Br(N_v'(u^\circ))$; this implies that u° is mapped on the same branch of S by the two sliced scenarios. Thus, for a given (G, S, θ_S), each equivalence class of $\mathcal{C}$ or $\mathcal{C}'$ can be characterized by a pair $(G^o$,$M_v)$ with M_v a mapping of $V(G^o)$ to $V(S)$. Different scenarios in a same dated class $(G^o$,$M_v)$ will differ on their M_t mapping, while different scenarios in a sliced class $(G^o$,$M_v)$ will differ on the precise chunk of branches on which they map nodes of G^o.

Finally, note that not any pair $(G^o$,$M_v)$ – with M_v a mapping from nodes of G^o to nodes of S – represents a class of scenarios. Indeed, when imposing no constraint on M_v, there exists cases where no M_t can be defined to have a valid dated scenario (M_v, M_t); similarly, for some M_v, no valid sliced scenario with mapping $M_v' : V(G^o) \to V(S')$ such $\forall u^\circ \in G^o, Br(M_v'(u^\circ)) = M_v(u^\circ)$ exists.

Definition 33 (The dated-to-sliced function f_{d2s}) The *dated-to-sliced* function, denoted by f_{d2s}, is defined on $\mathcal{C} \to \mathcal{C}'$ and associates to each dated class $c_d = (G^o, M_v) \in \mathcal{C}$ the sliced class $c_s = (G^o, M_v) \in \mathcal{C}'$.

The function f_{d2s} may look like the identity function, but it is not since C and C' are different spaces. Hence we have so far no insurance that the image of a valid dated scenario class always leads to a valid sliced scenario class, and vice versa.

Property 9 (f_{d2s} validity) The dated-to-sliced function f_{d2s} is correctly defined.

Proof. We need to prove that $\forall c_d = (G^o, M_v) \in \mathcal{C}$, the pair $(G^o, M_v) = f_{d2s}(c_d) \in \mathcal{C}'$, *i.e.*, that there is at least one sliced scenario $ss = (G^o, M_v')$ that maps vertices of G^o into vertices of S (when considering $Br(\cdot)$) as done by M_v. Since $c_d \in \mathcal{C}$, there exists at least one dated scenario in this class, denoted $ds = (G^o, M^s)$ such that $M_v^s = M_v$ (since $ds \in c_d$). Consider the mapping M_v' that associates each vertex u° of G^o to the vertex $x' = Chunk(M_v^s(u^\circ), M_t^s(u^\circ))$ of S'.
The pair (G^o, M_v') is a valid sliced scenario, since each vertex $u^\circ \in G^o$ is, by construction, mapped on the same branch of S (via a chunk of it), hence each vertex is associated to the same event type in ss and ds:

1. in ss, each $u^\circ \in G^o$ associated to a $\mathbb{C}^o$, $\mathbb{L}^o$ or $\mathbb{S}^o$ event respects the associated constraints since they were respected by ds (which contains more stringent conditions than in the definition of a sliced scenario) and are unchanged by the dated to sliced transformation;
2. in ss, each $u^\circ \in G^o$ associated to a $\mathbb{D}^o$, or a $\mathbb{T}^o$ event respect the associated constraints for the mapping of u_l° and u_r° since they are also unchanged by the transformation. The respect of the time constraints follows from the fact that $t_1 \geq t_2 \Rightarrow I_{\theta_{S'}}(Chunk(x_1, t_1)) \geq I_{\theta_{S'}}(Chunk(x_2, t_2))$ for all nodes x_1 and x_2 of S such that $t_1 \in [\theta_S(x_1), \theta_S(x_{1p})[$ and $t_2 \in [\theta_S(x_2), \theta_S(x_{2p})[$.

Hence (G^o, M'_v) is a valid sliced scenario, which by construction belongs to the class $c_s = (G^o, M_v) \in \mathcal{C}'$ and f_{d2s} is a valid function from $\mathcal{C} \to \mathcal{C}'$. $\square$

Definition 34 (The sliced-to-dated function f_{s2d}) The *sliced-to-dated* function, denoted by f_{s2d}, is defined on $\mathcal{C}' \to \mathcal{C}$ and associates to each sliced class $c_s = (G^o, M_v) \in \mathcal{C}'$ the dated class $c_d = (G^o, M_v) \in \mathcal{C}$.

Property 10 (f_{s2d} validity) The sliced-to-dated function f_{s2d} is correctly defined.

Proof. We need to prove that for any class $c_s = (G^o, M_v) \in \mathcal{C}'$, the pair $(G^o, M_v) = f_{s2d}(c_s)$ is a class of $\mathcal{C}$, *i.e.*, that there is at least one dated scenario $ds = (G^o,(M^s_v, M^s_t))$ such that $M^s_v = M_v$.
Since $c_s \in \mathcal{C}'$, there exists at least one sliced scenario in this class, denoted by $ss = (G^o, M'_v)$ and hence such that $\forall u^\circ \in G^o : Br(M'_v(u^\circ)) = M_v(u^\circ)$ (since $ss \in c_s$).

For each node $u^\circ \in V(G^o)$ if ds associates u° to a given event $\mathbb{E}$ then cs also has to associate u° to $\mathbb{E}$: indeed *i)* ds and ss map each vertex u° of G^o on the same branch of S since $M^s_v(\cdot) = Br(M'_v(\cdot)) = M_v(\cdot)$; *ii)* the vertex mappings impose the kind of event (*e.g.*, Property 2) and *iii)* the constraints on the vertex mappings are identical for each kind of event $\mathbb{E}$ in Definitions 7 and 20. Now consider a time mapping function M_t that would date each vertex $u^\circ \in G^o$ so that

1. if u° is associated to a $\mathbb{C}^o$ or $\mathbb{S}^o$ then $M_t(u^\circ) = \theta_S(M'_v(u^\circ))$
2. if u° is associated to a $\mathbb{L}^o$ then $M_t(u^\circ) = t_1$ with $t_1 \in I_{\theta_{S'}}(M'(u^\circ))$
3. if u° is associated to a $\mathbb{D}^o$ or a $\mathbb{T}^o$, then $M_t(u^\circ) = t_1$ with $t_1 \in I_{\theta_{S'}}(M'_v(u^\circ)), t_1 > max(M_t(u^\circ_l), M_t(u^\circ_r))$

With such a time mapping $(G^o,(M_v, M_t))$ would obviously be a valid dated scenario, since those constraints ensure i) that any pair $(M_v(u^\circ), M_t(u^\circ))$ is coherent since $M_t(u^\circ) \in I_{\theta_{S'}}(M_v(u^\circ)) \subseteq I_{\theta_S}(M_v(u^\circ))$ holds for $\mathbb{D}^o, \mathbb{T}^o$ and $\mathbb{L}^o$ events. ii) that any node u is mapped at a date posterior to those of its children (and hence of its descendants). So, we need to prove that it is always possible to construct such a time mapping function M_t. This can easily be done as detailed in Algorithm 2, by associating dates of the vertex of G^o in post-order (hence ensuring that a vertex of G^o is considered after all its children/descendants). Using this ordering, when dating node a u° which is mapped in the time slice s_i, it suffices to constraint $M_t(u^\circ)$ to be within $]d_{last}, d_{max}[$ where d_{last} is the last date (hence the maximal one) that was affected to a node in this slice and d_{max} is the upper bound of this time slice. A simple choice for $M_t(u^\circ)$ is to use the middle of this interval. Note that node after node $]d_{last}, d_{max}[$ is smaller and smaller but it is never $\emptyset$ (Zeno's paradox).

It is hence always possible to construct a time mapping function M_t respecting the above mentioned constraint and it hence always exists a valid dated scenario $ss_1 = (G^o, (M_v, M_t))$, which belongs to the class $c_d = (G^o, M_v) \in \mathcal{C}$.Thus, f_{s2d} is a valid function from $\mathcal{C}' \to \mathcal{C}$.

$\square$

Algorithm 2 Given a sliced scenario (M') for (G, S, θ_S), this function computes a dated function M_t compatible with M' and with the topology of G^o.

```
1: for $k$ from 0 to $h(root(G^o))$ do
2:     $date\_last[k] \leftarrow$ the initial lower bound of this time slice
3:     $date\_max[k] \leftarrow$ the upper bound of this time slice
4: end for
5: for $u^\circ$ in $V(G^o)$ considered in post-order do
6:     $x' \leftarrow (M'(u^\circ))$; $x \leftarrow Br(x')$;
7:     if $u^\circ$ is an $\mathbb{S}^o$ or a $\mathbb{C}^o$ in $M'$ then
8:         $M_t(u^\circ) \leftarrow \theta_S(x)$
9:     else
10:        $M_t(u^\circ) \leftarrow \big(date\_last[h(x')] + date\_max[h(x')]\big)/2$
11:        $date\_last[h(x')] \leftarrow M_t(u^\circ)$
12:    end if
13: end for
14: return $M_t$
```

Theorem 1 *The function f_{d2s} is a bijection between the classes of equivalent scenarios defined on dated and sliced scenarios for (G^o, S, θ_S). Moreover this bijection preserves the event types associated to the vertex of G^o and hence the overall cost of the scenario.*

Proof. The function $f_{d2s} : \mathcal{C} \to \mathcal{C}'$ is bijective, if and only if, it exists a function $g : \mathcal{C}' \to \mathcal{C}$ such that $\forall c \in \mathcal{C} : g(f_{d2s}(c)) = c$ and $\forall c' \in \mathcal{C}' : f_{d2s}(g(c')) = c'$. These two conditions obviously hold for $g = f_{s2d}$ since, it directly follows from the definition of f_{d2s} and f_{s2d} that:

– $\forall c = (G^o, M) \in \mathcal{C}$, $f_{d2s}(f_{s2d}(G^o, M)) = f_{d2s}(G^o, M) = (G^o, M)$
– $\forall c' = (G^o, M) \in \mathcal{C}'$, $f_{s2d}(f_{d2s}(G^o, M)) = f_{d2s}(G^o, M) = (G^o, M)$

As already mentioned previously in this section, since f_{d2s} and f_{s2d} preserve G^o and its vertex mapping on S, they also preserve the events associated with each vertex of G^o in the source scenario class and hence the overall cost of the source scenario class. □

7 Equivalence of the MP reconciliation and MP sliced scenario problems

As seen in the previous section, exploring the space of basic-up scenarios is sufficient to find a MP scenario. In this section we will prove that there is a bijection between basic-up sliced scenarios and basic c-reconciliations, which preserves the $\mathbb{D}$ and $\mathbb{T}$ events associated to the vertices of G, and the number of $\mathbb{L}$ events inferred on each branch of S, hence the overall cost. Hence, a most parsimonious sliced scenario can be found by searching for a most parsimonious (compact) reconciliation, which is easier since this does not requires to explore the space of all possible G°.

Note that a branch (u_p, u) of G corresponds to a path in G^o that link (the nodes corresponding to) u_p and u through ghost nodes .

Definition 35 **(*GhostPathInG*$^0(\cdot,\cdot,\cdot)$)** Given a full gene tree G^o for G, and a node u of G, *GhostPath InG*$^0(G^0, G, u)$ returns the sequence of ghost nodes of G^o constituting the path between u_p and u in G^o. Note that for simplicity, we confound here a node u of G and its corresponding traceable node u^0 in G^o.

Algorithm 3 $f_{r2s}(\bar{\alpha}, G, S, \theta_S)$ Given a compact reconciliation $\bar{\alpha}$ between a gene tree G and a dated species tree (S, θ_S), this function computes a basic-up sliced scenario (G^o, M_v) between these trees that, up to $\varnothing$ events, represents the same events as $\bar{\alpha}$.

```
1: $G^o \leftarrow G$;
2: $L_{\mathbb{L}}(G^o) \leftarrow \emptyset$;
3: for all node $u \in V(G)$ do
4:     $\eta \leftarrow |\bar{\alpha}(u)|$;
5:     replace in $G^o$ the edge $(u_p, u)$ with a path $[u_p, g^\circ_1, \ldots, g^\circ_{\eta-1}, u := g^\circ_\eta]$        //ghost nodes of $G^o$
6:     add a new leaf node $d^\circ_k$ to $L_{\mathbb{L}}(G^o)$ and connect it as the left child of $g^\circ_k$        //dead leaf of $G^o$
7:     for $k$ from 1 to $\eta$ do
8:         $M_v(g^\circ_k) \leftarrow \bar{\alpha}_k(u)$
9:     end for
10:    for $k$ from 1 to $\eta$ -1 do
11:        if $\bar{\alpha}_k(u)$ is associated to a $\mathbb{SL}$ then
12:            $M_v(d^\circ_k) \leftarrow$ the child of $\bar{\alpha}_k(u)$ that is not $\bar{\alpha}_{k+1}(u)$        $\mathbb{SL}$
13:        else
14:            $M_v(d^\circ_k) \leftarrow \bar{\alpha}_k(u)$        $\mathbb{TL}$
15:        end if
16:    end for
17: end for
18: return $(G^o, M_v)$
```

Definition 36 (The reconciliation-to-sliced function f_{r2s}) The *reconciliation-to-sliced* function, denoted by f_{r2s} and detailed in Algorithm 3, is defined on $\mathcal{C}^r \to \mathcal{C}^s$ where $\mathcal{C}^r$ denotes the space of basic compact reconciliations and $\mathcal{C}^s$ the space of basic-up sliced scenario. It associates to each basic c-reconciliation $\bar{\alpha}$ a pair (G^o, M_v): G^o is obtained from G by adding dead and ghost nodes (lines 5, 6) and is thus a full gene tree for G, while M_v is a mapping between nodes of G^o and vertices of S' constructed from $\bar{\alpha}$ by adding the mapping of dead (line 8) and ghost (lines 12, 14) nodes so that (G^o, M_v) is a basic-up sliced scenario for G and (S, θ_S), see proof of Property 11 for more details.

Property 11 (f_{r2s} validity) The reconciliation-to-sliced function f_{r2s} is correctly defined and output a basic-up sliced scenario with the same cost as the basic c-reconciliation in input.

Proof. We need to prove that $\forall c_\alpha \in \mathcal{C}^r$, the pair $(G^o, M_v) = f_{r2s}(\bar{\alpha}, G, S, \theta_S)$ is an element of $\mathcal{C}^s$, i.e. that the output pair $ss_1 = (G^o, M_v)$ is indeed a valid basic-up sliced scenario $ss_1 = (G^o,M'_v)$ for (G, S, θ_S), i.e. i) G^o is a full tree for G and ii) all node of G^o satisfies the condition of a basic-up sliced scenario.

The first point is easily proved as G^o is obviously a full gene tree for G by construction (lines 1 and 6 of Algorithm 3).

We will now prove that each non dead gene u° satisfies the constrained of a sliced scenario (Definition 20). This is done based on the event type of $\bar{\alpha}_k(u)$. Recall that each non dead gene u° is mapped such that $M'_v(u^\circ) = \bar{\alpha}_k(u)$ and $u^\circ \in GhostPathInG^0(G^0, G, u)$.

- $\underline{\bar{\alpha}_k(u) \text{ is a } \mathbb{C} \text{ event}}$ then so is u° since $L_{\mathbb{C}}(G^o)$ and $s(.)$ are unchanged and $h(\bar{\alpha}_k(u)) = 0$ implies $\theta_{S'}(M'_v(u^\circ)) = 0$.
- $\underline{\bar{\alpha}_k(u) \text{ is an event } \mathbb{E} \in \{\mathbb{S}, \mathbb{D}, \mathbb{T}\}}$ then $k = |\bar{\alpha}(u)|$ and u° is a node of G. Then, by construction, $M'_v(u^\circ_r) = \bar{\alpha}_1(u^\circ_r)$ and $M'_v(u^\circ_l) = \bar{\alpha}_1(u^\circ_l)$ (lines 7, 8). Thus, since $\bar{\alpha}_{|\bar{\alpha}(u)|}(u)$ respects the (c-reconciliation) conditions of a $\mathbb{E}$ event based on the three mapping $\bar{\alpha}_{|\bar{\alpha}(u)|}(u), \bar{\alpha}_1(u_r), \bar{\alpha}_1(u_l)$, $M'(u^\circ)$ respect the (identical) conditions of a (sliced scenario) $\mathbb{E}^\circ$ event (the condition on $h(.)$ are obviously equivalent to those on $\theta_{S'}(\cdot)$.
- $\underline{\bar{\alpha}_k(u) \text{ is an event } \mathbb{E} \in \{\mathbb{SL}, \mathbb{TL}\}}$ then $k < |\bar{\alpha}(u)|$ and u° is by construction a ghost node.
  - $\underline{\bar{\alpha}_k(u) \text{ is a } \mathbb{SL} \text{ event}}$ then $x := \bar{\alpha}_k(u)$ is non artificial and, by construction, $M'_v(u^\circ_r) = \bar{\alpha}_{k+1}(u)$. Moreover, $Br(\bar{\alpha}_{k+1}(u)) \in \{x_l, x_r\}$ (Definition 27); this plus lines 6 and 12 of Algorithm 3 ensure that $\{Br(M'_v(u^\circ_r)), Br(M'_v(u^\circ_l)\} = \{x_l, x_r\}$ thus completing the required conditions for $M'_v(u^\circ)$ to be a $\mathbb{S}^o$ event.
  - $\underline{\bar{\alpha}_k(u) \text{ is a } \mathbb{TL} \text{ event}}$ then by construction $M'_v(u^\circ_r) = \bar{\alpha}_{k+1}(u)$ and $Br(\bar{\alpha}_{k+1}(u)) = y$ with $y \neq x$ and $h(y) \geq h(x)$ (Definition 27). This ensures that $Br(M'_v(u^\circ_r)) = y$ with $y \neq x$, $I_{\theta_{S'}}(x') \subseteq I_{\theta_S}(y)$ and $I_{\theta_{S'}}(x') \geq I_{\theta_{S'}}(M'_v(u^\circ_r))$. Moreover, lines 6 and 14 of Algorithm 3 ensure that the $M'_v(u^\circ_l) = \bar{\alpha}_k(u) = M'_v(u^\circ)$ thus completing the required conditions for $M'_v(u^\circ)$ to be a $\mathbb{T}^o$ event.

  Moreover, each $\mathbb{SL}$ and $\mathbb{TL}$ event is also associated to a dead leaf of G^o that is present in $L_{\mathbb{L}}(G^o)$ (Algorithm 3 line 6) and is thus a $\mathbb{L}$ event in M'. Note also that all dead nodes are leaves and that all losses are positioned as high as possible, as required in a basic-up sliced scenario.

Hence (G^o, M'_v) is a full sliced scenario for (G, S, θ_S), with the same events than $\bar{\alpha}$ positioned on the same branch of S and they thus have the same cost. Let now conclude the proof by verifying that this sliced scenario is a basic-up one. To be basic, (G^o, M'_v) should satisfy the three requirements of Definition 22. The root $r(G^o)$ is traceable since $\bar{\alpha}$ is basic (requirement a); each dead gene of G^o is a leaf by construction – Algorithm 3 line 6 (requirement b); each ghost gene of the scenario is either a $\mathbb{SL}$ or a $\mathbb{TL}$ events – Algorithm 3 lines 10-15, (requirement c). So (G^o, M'_v) is indeed basic. Finally each dead leaf is mapped by M' on the highest possible chunk of S' (with respect to other non dead node of G^o) since if it originates from a $\mathbb{SL}$ event then it is mapped on a child of $x' \cong x$ (Algorithm 3 line 12) which by definition is in the upper chunk of the child of x; while if it originates from a $\mathbb{TL}$ event it is mapped by M' in the same chunk as u° (Algorithm 3 line 14) and can obviously not be mapped higher without invalidating the basic time constraint that u° must satisfy (i.e. being mapped at least as high as its children). □

Definition 37 (The sliced-to-reconciliation function f_{s2r}) The *sliced-to-reconciliation* function, denoted by f_{s2r} and detailed in Algorithm 4, is defined on $\mathcal{C}^r \to \mathcal{C}^s$ and associates a basic c-reconciliation $\bar{\alpha}$ to each basic-up sliced scenario (G^o, M'_v). This is done by concatenating, for each traceable node u of G^o, the mappings M'_v of all ghost nodes above u, until another traceable node is reached.

Property 12 (f_{s2r} validity) The sliced-to-reconciliation function f_{s2r} is correctly defined and output a basic c-reconciliation with the same events and same cost as the input basic-up reconciliation.

Proof. We need to prove that $\forall (G^o, M'_v) \in \mathcal{C}^s$, the output c-reconciliation $\bar{\alpha} = f_{s2r}(G^o, M'_v)$ is an element of $\mathcal{C}^r$, i.e. that the output $\bar{\alpha}$ is indeed a valid c-reconciliation for (G, S, θ_S), i.e. that for all node of $u \in G$ each element $\bar{\alpha}_k(u)$ (where $1 \leq j \leq \ell := |\bar{\alpha}(u)|$) satisfied the conditions of a c-reconciliation.

Each element $\bar{\alpha}_k(u) \in \bar{\alpha}$ corresponds to a $M'(g^\circ_k)$ for a non dead gene of G^o (Algorithm 4 line 1 and 6). We will now prove that each such non dead gene node $\bar{\alpha}_k(u) \in \bar{\alpha}$ satisfies the constrains of a c-reconciliation (Definition 27), this is done based on the node type and event type of $M'(g^\circ_k)$:

- $\underline{M'(g^\circ_k) \text{ is a } \mathbb{C}^o \text{ event}}$ then g°_k is a traceable node u°, $k = |\bar{\alpha}(u)| = \eta$ and $\bar{\alpha}_\eta(u)$ is a $\mathbb{C}$ event since $L_{\mathbb{C}}(G^o)$ and $s(.)$ are unchanged and $\theta_{S'}(M'(u^\circ)) = 0$ implies $h(\bar{\alpha}_\eta(u)) = 0$.

Algorithm 4 $f_{s2r}(G^o, M')$ Given a basic-up sliced scenario (G^o, M') between a gene tree G and a dated species tree (S, θ_S), this function computes a basic c-reconciliation $\bar{\alpha}$ between these trees with the same events (up to $\varnothing$ events) as (G^o, M') .

```
1: $\bar{\alpha}(r(G)) \leftarrow [M'(r(G^o))]$;
2: for all traceable node $u \in V(G^o) \setminus \{r(G^o)\}$ do
3:     $\bar{\alpha}(u) \leftarrow [\ ]$
4:     $[g_1^\circ, \ldots, g_{\eta-1}^\circ, u := g_\eta^\circ] = concatSeq\left(GhostPathInG^0(G^0, G, u), [u]\right)$
5:     for $k$ from 1 to $\eta$ do
6:         $\bar{\alpha}(u) \leftarrow concatSeq\left(\bar{\alpha}(u), [M'(g_k^\circ)]\right)$
7:     end for
8: end for
9: return $(\bar{\alpha})$
```

– $\underline{M'(g_k^\circ)}$ is associated to an event $\mathbb{E}^\circ \in \{\mathbb{S}^o, \mathbb{D}^o, \mathbb{T}^o\}$ and g_k° is a traceable node u°. Then, by construction, $k = |\bar{\alpha}(u)| = \eta$ (Algorithm 4 lines 2, 4), $\bar{\alpha}_1(u_r^\circ) = M'_v(u_r^\circ)$ and $\bar{\alpha}_1(u_l^\circ) = M'_v(u_l^\circ)$. Since $M'_v(u^\circ)$ respects the (sliced-scenario) conditions of a $\mathbb{E}^\circ$ event, $\bar{\alpha}_\eta(u^\circ)$ respect the (identical) conditions of a (c-reconciliation) $\mathbb{E}$ event (the condition on $\theta_{S'}(\cdot)$ are obviously equivalent to those on $h(\cdot)$).

– $\underline{M'(g_k^\circ)}$ is associated to an event $\mathbb{E}^\circ \in \{\mathbb{S}^o, \mathbb{T}^o\}$ and g_k° is a ghost node with a non dead son g_{k-1}°. Then $k < |\bar{\alpha}(u)| = \eta$ (Algorithm 4 line 2, 4). Note that, since (G^o, M') is basic, a ghost node g_k° can neither be associated to a $\mathbb{D}^o$ event nor to a $\mathbb{T}^o$ event where the transferred child is a dead leaf. From these remarks and (Algorithm 4 lines 4, 6) it follows that $\bar{\alpha}_{k+1}(u) = M'_v(g_{k-1}^\circ)$ then (using the same arguments as in the reciprocal proof) either
  – $\underline{\mathbb{E}^\circ \text{ is a } \mathbb{S}^o \text{ event}}$ and thus $\bar{\alpha}_k(u)$ respect the constraints of a $\mathbb{SL}$ event; or
  – $\underline{\mathbb{E}^\circ \text{ is a } \mathbb{T}^o \text{ event}}$ and thus $\bar{\alpha}_k(u)$ respect the constraints of a $\mathbb{TL}$ event, since the constraints on $h(\cdot)$ obviously hold.

Finally, it is easy to see that the input sliced scenario being basic, so is $\bar{\alpha}$. □

Theorem 2 *The function f_{s2r} is a bijection between the basic-up sliced scenarios and basic c-reconciliations, which preserves the $\mathbb{D}$ and $\mathbb{T}$ events associated to the vertices of G and the number of $\mathbb{L}$ events inferred on each branch of S, hence the overall cost.*

Proof. The function $f_{s2r} : \mathcal{C}^s \to \mathcal{C}^r$ is bijective, if and only if, it exists a function $g : \mathcal{C}^r \to \mathcal{C}^s$ such that $\forall s = (G^o, M') \in \mathcal{C}^s : g(f_{s2r}(s)) = s$ and $\forall r \in \mathcal{C}^r : f_{s2r}(g(r)) = r$. These two conditions hold for $g = f_{s2r}$ since:

– $\underline{f_{s2r}(f_{r2s}(\bar{\alpha}, G, S, \theta_S)) = \bar{\alpha}}$. Given $(G^o, M') = f_{r2s}(\bar{\alpha}, G, S, \theta_S)$, the $\bar{\alpha}$ mapping of $r(G)$ is preserved as $M'(r(G^o))$ and restored by $f_{s2r}(G^o, M')$ (Algorithm 4 line 1). Moreover, for any non root node u of G, the order sequence of S' vertices corresponding to $\bar{\alpha}(u)$ is encoded as the M' mapping of the ordered nodes of G^o along the path $concatSeq(GhostPathInG^\circ(G^o, G, u), [u])$ (Algorithm 4 lines 5 and 8) and the original $\bar{\alpha}(u)$ is restored by f_{s2r}.

– $\underline{f_{r2s}(f_{s2r}(G^o, M'), G, S, \theta_S) = (G^o, M')}$. The structure and mapping of G^o is fully encoded in $\bar{\alpha} = f_{s2r}(G^o, M')$: i) since $r(G^o) \cong r(G)$ and (G^o, M') is basic, the root of G^o and its mapping are encoded in $\bar{\alpha}$ (Algorithm 4 lines 1); ii) since the ordered sequence of mapping along the path $concatSeq(GhostPath\ InG^\circ(G^o, G, u), [u])$ is encoded in the ordered sequenced of $\bar{\alpha}(u)$ (Algorithm 4 lines 4 and 6) and hence identically recreated by $f_{r2s}(\bar{\alpha}, G, S, \theta_S)$ (Algorithm 3 lines 5 and 8); and iii) the remaining part of G^o are dead leaves (since (G^o, M') is basic) and each of them is restored from $\bar{\alpha}$ (Algorithm 3 lines 6) together with its mapping. The original mapping of such a dead leaf u_d is recovered thanks to its unchanged parent mapping (see case i) and to the event type $(\mathbb{SL}, \mathbb{TL})$ of this parent in $\bar{\alpha}$ which encoded the branch of S on which u_d was mapped (Algorithm 3 lines 11 and 13). Finally, as we focus on basic-up scenarios, u_d can only be mapped on the highest possible chunk of the identified S branch (Algorithm 3 lines 12 and 14).

– The fact that the event set is preserved by this bijection directly follows from Properties 11 and 12.

□

8 Equivalence of the MP dated scenario and MP reconciliation problems

The aim of this short section is to prove that a most parsimonious dated scenario can be found by searching for a most parsimonious reconciliation, by chaining several intermediate results proved in the previous sections.

First of all, note that Property 1 ensures that we can focus on basic dated scenarios while Property 8 ensures that we can focus on basic reconciliations. Property 7 ensures that there exist a bijection between (basic) reconciliations and (basic) c-reconciliations. Theorem 1 ensures that there exists a bijection between basic c-reconciliations and the basic-up sliced scenarios, which preserves the event types and the the overall cost.

Moreover, Theorem 2 ensures that there exists a bijection between the classes of (basic) equivalent scenarios defined on dated and sliced scenarios, preserving the event types and the the overall cost. Since per each basic sliced scenario, there exists a basic-up scenario equivalent to it, this proves the claim stated at the beginning of the section.

9 Conclusion

The main contribution of this paper is to have formally proven an important assumption, implicitly made by previous works in the field of reconciliations: that solving the (maximum) parsimony $\mathbb{DTL}$ dated reconciliation problem in the discrete framework is equivalent to finding a most parsimonious dated $\mathbb{DTL}$ scenario in the continuous framework.

In the process, we also introduced new concepts and proved several intermediate results that may be useful on their own. First, we give here an explicit description of the model we used in [13] for the evolution of a full gene tree G^o along a species tree S via $\mathbb{D}, \mathbb{T}, \mathbb{L},$ and $\mathbb{S}$ events. This provides a way to discriminate between valid and invalid reconciliations, regardless of the algorithm used to produce these reconciliations. This explicit description also allows others to compare their models with ours, and to easily pinpoint the differences between the two. For example, the model described in [48] was thought to be equivalent ours, until [12] pointed out that the former model does not take into account $\mathbb{TL}$ events. This could not be noticed easily as the tackled underlying models were not *explicitly* described in [13] and [48]. Second, the concept of equivalence class among reconciliations allows to spot reconciliations that are inherently equivalent, even though not identical. Third, the algorithm given in Section 6 that computes a dated scenario equivalent to a given sliced scenario (hence a reconciliation), provides a practical solution to draw a reconciliation as an embedding of the gene tree within the species tree. Indeed, for such a drawing, each event (node of the gene tree) should be assigned to a precise date, and not just to a time interval on a branch [6].

The concepts, algorithms and proofs provided in this paper constitute a theoretical toolbox that will likely facilitate future theoretical contributions in the field.

Acknowledgements. This work was funded by the French Agence Nationale de la Recherche (ANR) through grants ANR-09-PEXT-000 ”Phylospace” and ANR-10-BINF-01-01 ”Ancestrome”, and by the *Institut de Biologie Computationnelle*. This publication is contribution no. 2015-098 of the Institut des Sciences de l’Evolution de Montpellier (ISEM, UMR 5554).

References

1. Åkerborg, Ö., Sennblad, B., Arvestad, L., Lagergren, J.: Simultaneous Bayesian gene tree reconstruction and reconciliation analysis. Proceedings of the National Academy of Sciences of the United States of America **106**(14), 5714–5719 (2009)
2. Arvestad, L., Berglund, A.C., Lagergren, J., Sennblad, B.: Bayesian gene/species tree reconciliation and orthology analysis using MCMC. Bioinformatics **19 Suppl 1**, 7–15 (2003)
3. Bansal, M.S., Alm, E.J., Kellis, M.: Efficient algorithms for the reconciliation problem with gene duplication, horizontal transfer and loss. Bioinformatics **28**(12), i283–91 (2012). DOI 10.1093/bioinformatics/bts225
4. Berglund, A.C., Steffansson, P., Betts, M.J., Liberles, D.A.: Optimal gene trees from sequences and species trees using a soft interpretation of parsimony. J. Mol. Evol. **63**, 240–250 (2006)
5. Charleston, M.: Jungles: a new solution to the host/parasite phylogeny reconciliation problem. Mathematical Biosciences **149**(2), 191–223 (1998). URL `http://dx.doi.org/10.1016/S0025-5564(97)10012-8`
6. Chevenet, F., Doyon, J.F., Scornavacca, C., Jousselin, E., Berry, V.: Sylvx: a viewer for phylogenetic reconciliations (2015). In preparation

7. Conow, C., Fielder, D., Ovadia, Y., Libeskind-Hadas, R.: Jane: a new tool for the cophylogeny reconstruction problem. Algorithms Mol Biol **5**, 16 (2010)
8. Cotton, J., Page, R.: Rates and patterns of gene duplication and loss in the human genome. Proc Biol Sci **272**(1560), 277–83 (2005)
9. Daubin, V., Moran, N.A., Ochman, H.: Phylogenetics and the cohesion of bacterial genomes. Science **301**, 829–832 (2003)
10. David, L., Alm, E.: Rapid evolutionary innovation during an archaean genetic expansion. Nature **469**(7328), 93–96 (2011)
11. Demuth, J.P., De Bie, T., Stajich, J.E., Cristianini, N., Hahn, M.W.: The evolution of mammalian gene families. PLoS ONE **1**, e85 (2006)
12. Doyon, J., Ranwez, V., Daubin, V., Berry, V.: Models, algorithms and programs for phylogeny reconciliation. Brief. Bioinformatics **12**, 392–400 (2011)
13. Doyon, J.P., Scornavacca, C., Gorbunov, K.Y., Szllosi, G.J., Ranwez, V., Berry, V.: An efficient algorithm for gene/species trees parsimonious reconciliation with losses, duplications and transfers. In: E. Tannier (ed.) RECOMB-CG, *Lecture Notes in Computer Science*, vol. 6398, pp. 93–108. Springer (2010)
14. Drummond, A.J., Ho, S.Y., Phillips, M.J., Rambaut, A.: Relaxed phylogenetics and dating with confidence. PLoS Biology **4**(5) (2006). DOI 10.1371/journal.pbio.0040088
15. Fischer, I., Dainat, J., Ranwez, V., Glemin, S., Dufayard, J.F., Chantret, N.: Impact of recurrent gene duplication on adaptation of plant genomes. BMC Plant Biology **14**(1), 151 (2014). DOI 10.1186/1471-2229-14-151. URL `http://www.biomedcentral.com/1471-2229/14/151`
16. Fitch, W.M.: Homology - a personal view on some of the problems. Trends Genet. **16**(5), 227–231 (2000)
17. Gabaldon, T.: Computational approaches for the prediction of protein function in the mitochondrion. Am J Physiol Cell Physiol **291**(6), C1121–1128 (2006). DOI 10.1152/ajpcell.00225.2006
18. Goldenfeld, N., Woese, C.: Biology's next revolution. Nature **445**, 369 (2007)
19. Goodman, M., Czelusniak, J., Moore, G.W., Herrera, R.A., Matsuda, G.: Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences. Syst. Zool. **28**, 132–163 (1979)
20. Gorbunov, K.Y., Lyubetsky, V.A.: Reconstructing genes evolution along a species tree. Mol. Biol. (Mosk.) **43**, 946–958 (2009)
21. Górecki, P.: Reconciliation problems for duplication, loss and horizontal gene transfer. In: P.E. Bourne, D. Gusfield (eds.) RECOMB, pp. 316–325. ACM (2004). URL `http://dblp.uni-trier.de/db/conf/recomb/recomb2004.html#Gorecki04`
22. Górecki, P.: H-trees: a model of evolutionary scenario with horizontal gene transfer. Fundamenta Informaticae **103**, 105–128 (2010)
23. Górecki, P., Tiuryn, J.: Inferring evolutionary scenarios in the duplication, loss and horizontal gene transfer model. In: R. Constable, A. Silva (eds.) Logic and Program Semantics, *Lecture Notes in Computer Science*, vol. 7230, pp. 83–105. Springer Berlin Heidelberg (2012). DOI 10.1007/978-3-642-29485-3_7. URL `http://dx.doi.org/10.1007/978-3-642-29485-3_7`
24. Hallett, M., Lagergren, J., Tofigh, A.: Simultaneous identification of duplications and lateral transfers. In: RECOMB '04, pp. 347–356. ACM, New York, NY, USA (2004)
25. Hallett, M.T., Lagergren, J.: Efficient algorithms for lateral gene transfer problems. In: Proceedings of the Fifth Annual International Conference on Computational Biology, pp. 149–156. ACM, New York, NY, USA (2001). DOI 10.1145/369133.369188. URL `http://doi.acm.org/10.1145/369133.369188`
26. Han, M.V., Thomas, G.W., Lugo-Martinez, J., Hahn, M.W.: Estimating gene gain and loss rates in the presence of error in genome assembly and annotation using CAFE 3. Mol. Biol. Evol. **30**(8), 1987–1997 (2013)
27. Kunin, V., Ouzounis, C.A.: The balance of driving forces during genome evolution in prokaryotes. Genome Res. **13**(7), 1589–1594 (2003)
28. Lafond, M., Swenson, K., El-Mabrouk, N.: An optimal reconciliation algorithm for gene trees with polytomies. In: B. Raphael, J. Tang (eds.) Algorithms in Bioinformatics, *Lecture Notes in Computer Science*, vol. 7534, pp. 106–122. Springer Berlin Heidelberg (2012). DOI 10.1007/978-3-642-33122-0_9. URL `http://dx.doi.org/10.1007/978-3-642-33122-0_9`
29. Lynch, M., Conery, J.S.: The evolutionary fate and consequences of duplicate genes. Science **290**(5494), 1151–1155 (2000)
30. Maddison, W.P.: Gene trees in species trees. Systematic biology **46**(3), 523–536 (1997)
31. Maere, S., De Bodt, S., Raes, J., Casneuf, T., Van Montagu, M., Kuiper, M., Van de Peer, Y.: Modeling gene and genome duplications in eukaryotes. Proc. Natl. Acad. Sci. U.S.A. **102**(15), 5454–5459 (2005)
32. Makino, T., McLysaght, A.: Positionally-biased gene loss after whole genome duplication: Evidence from human, yeast and plant. Genome Research **22**, 24–27 (2012)
33. Merkle, D., Middendorf, M.: Reconstruction of the cophylogenetic history of related phylogenetic trees with divergence timing information. Theory Biosci **123**(4), 277–299 (2005). DOI 10.1016/j.thbio.2005.01.003
34. Merkle, D., Middendorf, M., Wieseke, N.: A parameter-adaptive dynamic programming approach for inferring cophylogenies. BMC Bioinformatics **11**(Suppl 1), S60 (2010). DOI 10.1186/1471-2105-11-S1-S60
35. Page, R.D.: Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas. Syst. Biol. **43**, 58–77 (1994)
36. Puigbo, P., Wolf, Y., Koonin, E.: Search for a 'tree of life' in the thicket of the phylogenetic forest. Journal of Biology **8**(6), 59 (2009). DOI 10.1186/jbiol159. URL `http://jbiol.com/content/8/6/59`
37. Rasmussen, M.D., Kellis, M.: Accurate gene-tree reconstruction by learning gene- and species-specific substitution rates across multiple complete genomes. Genome Research **17**(12), 1932–1942 (2007). URL `papers2://publication/uuid/499F02A0-86D4-4624-8A32-78C69D8C4F9Cpapers://c574a988-041f-48f0-a387-1407660e65a3/Paper/p872`

38. Rasmussen, M.D., Kellis, M.: Unified modeling of gene duplication, loss, and coalescence using a locus tree. Genome Research **22**(4), 755–765 (2012). URL papers2://publication/uuid/54E62C0F-314A-4173-94B5-5368EEACB2E0papers://c574a988-041f-48f0-a387-1407660e65a3/Paper/p4162http://eutils.ncbi.nlm.nih.gov/entrez/eutils/elink.fcgi?dbfrom=pubmed&id=22271778&retmode=ref&cmd=prlinkspapers2://publication/doi/10.1101/gr.123901.111
39. Sanderson, M.: A nonparametric approach to estimating divergence times in the absence of rate constancy. Molecular Biology and Evolution **14**, 1218–1231 (1997)
40. Sebat, J., Lakshmi, B., Troge, J., Alexander, J., Young, J., Lundin, P., Maner, S., Massa, H., Walker, M., Chi, M., Navin, N., Lucito, R., Healy, J., Hicks, J., Ye, K., Reiner, A., Gilliam, T.C., Trask, B., Patterson, N., Zetterberg, A., Wigler, M.: Large-scale copy number polymorphism in the human genome. Science **305**(5683), 525–528 (2004)
41. Semon, M., Wolfe, K.H.: Consequences of genome duplication. Curr. Opin. Genet. Devel. **17**, 505–512 (2007)
42. Sjöstrand, J., Tofigh, A., Daubin, V., Arvestad, L., Sennblad, B., Lagergren, J.: A bayesian method for analyzing lateral gene transfer. Syst Biol **63**(3), 409–20 (2014). DOI 10.1093/sysbio/syu007
43. Suchard, M.A.: Stochastic Models for Horizontal Gene Transfer: Taking a Random Walk Through Tree Space. Genetics **170**(1), 419–431 (2005). URL papers2://publication/uuid/61E6A9EE-A1FB-484B-9428-BE1834DA4AD5papers://c574a988-041f-48f0-a387-1407660e65a3/Paper/p946papers://c574a988-041f-48f0-a387-1407660e65a3/Paper/p374papers2://publication/uuid/B15BDCD1-3BC5-41B2-97C4-613C1D00FADC
44. Szöllősi, G.J., Daubin, V.: Modeling gene family evolution and reconciling phylogenetic discord. Methods Mol Biol **856**, 29–51 (2012)
45. Szöllősi, G.J., Tannier, E., Lartillot, N., Daubin, V.: Lateral gene transfer from the dead. Systematic Biology **62**(3), 386–397 (2013). DOI 10.1093/sysbio/syt003
46. Szollosi, G.J., Boussau, B., Abby, S.S., Tannier, E., Daubin, V.: Phylogenetic modeling of lateral gene transfer reconstructs the pattern and relative timing of speciations. Proc. Natl. Acad. Sci. U.S.A. **109**(43), 17,513–17,518 (2012)
47. Tofigh, A.: Using trees to capture reticulate evolution, lateral gene transfers and cancer progression. Ph.D. thesis, KTH Royal Institute of Technology, Sweden (2009)
48. Tofigh, A., Hallett, M., Lagergren, J.: Simultaneous identification of duplications and lateral gene transfers. IEEE/ACM TCBB **99** (2010). DOI http://doi.ieeecomputersociety.org/10.1109/TCBB.2010.14
49. Tofigh, A., Sjöstrand, J., Sennblad, B., Arvestad, L., Lagergren, J.: Detecting LGTs using a novel probabilistic model integrating duplications, lgts, losses, rate variation, and sequence evolution (Manuscript)
50. Vernot, B., Stolzer, M., Goldman, A., Durand, D.: Reconciliation with non-binary species trees. J. Comput. Biol. **15**, 981–1006 (2008)
51. Zhang, L.: On a mirkin-muchnik-smith conjecture for comparing molecular phylogenies. Journal of Computational Biology **4**(2), 177–187 (1997)

10 Appendix

Proof of Claim 3.

1. directly follows from $\mathbb{C}^o$ and $\mathbb{S}^o$ event definitions.
2. the fact that $M_t(u^\circ) > M_t(v^\circ)$ for any node u° and any strict descendant v° of u°, can be proved recursively. The constraint holds for leaves of G^o. Assume that it holds for all strict descendant v° of u°, then it also holds for u°, since either i) $M(u^\circ)$ is a $\mathbb{S}^o$ event and $M_t(u^\circ) = \theta_S(x) > max(M_t(u^\circ_l), M_t(u^\circ_r))$ because being mapped on x_l (resp. x_r) impose to be mapped at a time included in $[\theta_S(x_l), \theta_S(x)[$ (resp. $[\theta_S(x_r), \theta_S(x)[$]) no matter the associated event, or ii) $M(u^\circ)$ is a $\mathbb{D}^o$ or $\mathbb{T}^o$ event and $M_t(u^\circ) > max(M_t(u^\circ_l), M_t(u^\circ_r))$ holds by definition. Since any strict descendant v° of u° is either u°_l, u°_r or one of their descendants, the inequality $M_t(u^\circ) > max(M_t(u^\circ_l), M_t(u^\circ_r))$ together with the recurrence hypothesis applied to u°_l and u°_r ensure that $M_t(u^\circ) > M_t(v^\circ)$.
3. follows from the fact that $\mathbb{D}^o$, $\mathbb{S}^o$, and $\mathbb{T}^o$ event definitions involve the two sons of u° in G^o.
4. follows from the explicit definition of $\mathbb{L}^o$ events.
5. follows from the fact that $M_t(u^\circ) \in]\theta_S(x), \theta_S(x_p)[\cap]\theta_S(y), \theta_S(y_p)[$ for any $\mathbb{T}^o$ event.

□

Proof of Property 1.

Let us prove this by contradiction. Suppose that we have an MPS, denoted $S_{opt} = (G^o, M)$, that is not a basic scenario.

If condition (a) is not satisfied then either i) $r(G^o)$ is a dead gene implying that G contains no nodes, impossible; ii) $r(G^o)$ is a ghost gene. In this case, consider an alternative scenario (H°, N) obtained in choosing H° as the subtree $G^o_{u^\circ}$, where u° is the non dead child of $r(G^o)$, and such that the mapping function N is simply the restriction of M to the nodes of H°. Since the sole leaves that have been removed are dead ones, H° is also a full gene tree for G. Moreover, since the mapping and children of each remaining node are unchanged in N compared to M, their associated events, as defined by Definition 7, are also unchanged, while at least a loss event is spared (located in the subtree rooted at the other child of $r(G^o)$). Hence (H°, N) would be a valid dated scenario for (G, S, θ_S) of lower cost than S_{opt}, a contradiction.

If condition (b) is not satisfied, then G^o contains at least one dead gene subtree $G^o_{u^\circ}$. Let us consider the alternative scenario obtained by deleting from G^o all strict descendants of u°, while keeping the exact same mapping for remaining nodes. Using the same argument as in the previous case, we can show that this valid dated scenario would have a lower cost than S_{opt}, a contradiction.

If condition (c) is not satisfied, then G^o contains a ghost gene u° being a $\mathbb{D}^o$ event, or being a $\mathbb{T}^o$ event that transfers the dead child of u°. In both cases, suppose, without loss of generality, that the dead child of u° is u°_r, and consider the full gene tree H° obtained from G^o by i) replacing the edges (u°_p, u°) and (u°, u°_l) by the edge (u°_p, u°_l), ii) deleting all nodes in $G^\circ_{u^\circ_r}$, iii) deleting u°. Let N be the function obtained by restricting M to the nodes of H°. Since $N_v(u^\circ) = N_v(u^\circ_l)$ and $N_t(u^\circ) > N_t(u^\circ_l)$, this modification does not modify the event associated to u°_p, u°_l ,or any other nodes of H°, via Definition 7. So, the scenario (H°, N) would be a valid one and have a lower cost than S_{opt}, a contradiction.

Therefore, if S_{opt} is an MPS, it has to satisfy all three conditions of Definition 13, hence it is a basic scenario. □

prop:recTocRec

Proof of Property 1.

- $\underline{f_{\alpha 2\bar{\alpha}} \text{ is valid.}}$ First note that applying $f_{\alpha 2\bar{\alpha}}$ to a reconciliation gives a function $\bar{\alpha}$ such that $h(\bar{\alpha}_1(u)) \leq h(\alpha_1(u))$ for any node u of G (by Property 6(1)). Additionally, Property 6(4) implies that $Br(\bar{\alpha}_1(u)) = Br(\alpha_1(u))$. Note that any $\bar{\alpha}_j(u)$ of $\bar{\alpha}$ is coming from an element of the sequence of α, say $\alpha_i(u)$. Then we have that $\bar{\alpha}_j(u) = \alpha_i(u)$; moreover, if $\bar{\alpha}_j(u) < |\bar{\alpha}(u)|$, then $Br(\bar{\alpha}_{j+1}(u)) = Br(\alpha_{i+1}(u))$ (by Property 6(3)-(4)) and $h(\bar{\alpha}_{j+1}(u)) \leq h(\alpha_{i+1}(u))$ (Property 6(1)). Finally, if $\alpha_i(u)$ is the last element of $\alpha(u)$, so is $\bar{\alpha}_j(u)$ for $\bar{\alpha}(u)$. Now we consider separately each case of Definition 26 for $\alpha_i(u)$:
  - $\underline{\mathbb{C} \text{ event:}}$ if $\alpha_i(u)$ is in case 1 then so is $\bar{\alpha}_j(u)$ in Definition 27 (same constraints, not altered by the removal of $\varnothing$ events).
  - $\underline{\mathbb{S} \text{ event:}}$ $\alpha_i(u)$ is the last element of $\alpha(u)$ and case 2 of Definition 26 transforms into case 2 of a c-reconciliation due to $Br(\bar{\alpha}_1(u_c)) = Br(\alpha_1(u_c))$ applied to children u_c of u.

  - $\underline{\mathbb{D}\text{ event:}}$ $\alpha_i(u)$ is the last element of $\alpha(u)$ and the constraint on $Br(\cdot)$ of case 3 of Definition 26 transforms readily into that of the same case for a c-reconciliation due to $Br(\bar{\alpha}_1(u)) = Br(\alpha_1(u))$ applied to u and its two children; moreover the constraint on the heights holds from $h(\bar{\alpha}_j(u)) = h(\alpha_i(u)) = h(\alpha_1(u_c)) \geq h(\bar{\alpha}_1(u_c))$ for both the children u_c of u.
  - $\underline{\mathbb{T}\text{ event:}}$ case 4 is similar to case 3.
  - $\underline{\mathbb{SL}\text{ event:}}$ the constraint on the $Br(\cdot)$ in case 5 for $\alpha_i(u)$ transforms into that of case 5 for a c-reconciliation due to $Br(\bar{\alpha}_{j+1}(u)) = Br(\alpha_{i+1}(u))$.
  - $\underline{\mathbb{TL}\text{ event:}}$ The same reason than above explains that the constraint of case 6 on $Br(\cdot)$ for a c-reconciliation holds from the constraint of case 6 for $\alpha_i(u)$; moreover the fact that $h(\bar{\alpha}_{j+1}(u)) \leq h(\alpha_{i+1}(u))$ (see beginning of this proof), ensures that the second part of case 6 for a c-reconciliation also holds.
  - $\underline{\varnothing\text{ event:}}$ these events are removed by $f_{\alpha 2\bar{\alpha}}$, thus are not to be considered.

  In conclusion, any element in a sequence for any node u of G falls into exactly one of the cases of Definition 27, so that $f_{\alpha 2\bar{\alpha}}$ is shown to output a c-reconciliation.
- $\underline{f_{\bar{\alpha}2\alpha}\text{ is valid.}}$ First, note that calls to *ArtificialPath* at line 17 are always ensured to indicate heights h_{min} and h_{max} corresponding to chunks in S' covering the branch (v_p, v) of S, where here $Br(A_{\bar{\alpha}}[k]) \cong v$. Moreover we always have $h_{min} < h_{max}$ for the two parameters given as input to the *ArtificialPath* function: from the definition of a c-reconciliation it is clear that the equivalent of Property 6(1) and (2) hold, thus that $h(A_{\bar{\alpha}}[k]) \leq h(A_{\bar{\alpha}}[k-1])$ holds, and in the particular case of $\mathbb{S}$ and $\mathbb{SL}$ events we have $h(A_{\bar{\alpha}}[k]) \leq h(A_{\bar{\alpha}}[k-1]) - 1$. As a result, the call to *ArtificialPath* function always returns a sequence of artificial nodes in a same branch, hence meeting constraint of case 7 ($\varnothing$ event) in Definition 26. Possibly, this sequence is empty (which happens when $h_{min} = h_{max}$).

  Any other element $\alpha_i(u)$ of α is also an element $\bar{\alpha}_j(u)$ of $\bar{\alpha}$ and should have the same type ($\mathbb{E}$) in α as in $\bar{\alpha}$. Indeed, as any $\varnothing_{u,k}$ is mapped on the same branch of S than u, the mapping constraints satisfied by u (and its children) for $\bar{\alpha}_j(u)$ are also satisfied for $\alpha_i(u)$. Moreover the height constraints associated to the event type ($\mathbb{E}$) are also satisfied by $\alpha_i(u)$:
  - $\underline{\mathbb{E} \in \{\mathbb{TL}, \mathbb{SL}\}}$: $\alpha_{i+1}(u)$ is either a $\varnothing$ event (and then $\varnothing$ events were added up to the adequate height by lines 13 and 15 of the algorithm), or is a non $\varnothing$ event, that is $\bar{\alpha}_{j+1}(u)$ (hence the constraint on the height of Definition 27 ensures the result).
  - $\underline{\mathbb{E} \in \{\mathbb{D}, \mathbb{T}, \mathbb{S}\}}$: for each children u_c of u, $\alpha_1(u_c)$ is either a $\varnothing$ event, in which case lines 5, 13 and 15 of the algorithm ensures that the constraint on the height holds, or is a non $\varnothing$ event, in which case the constraint on the height holds from Definition 27.

  Finally, note that constraints for $\mathbb{C}$ events are identical for reconciliations and c-reconciliations, so any such event in $\bar{\alpha}$ is preserved in α as the mapping of $\alpha_i(u)$ is unchanged by the algorithm.
- $\underline{\bar{\alpha} = f_{\alpha 2\bar{\alpha}}(f_{\bar{\alpha}2\alpha}(\bar{\alpha}))}$. Since a reconciliation $\bar{\alpha}$ contains no $\varnothing$ event and $f_{\bar{\alpha}2\alpha}(.)$ only adds no events while $f_{\alpha 2\bar{\alpha}}(.)$ removes all $\varnothing$ events we obviously have $\bar{\alpha} = f_{\alpha 2\bar{\alpha}}(f_{\bar{\alpha}2\alpha}(\bar{\alpha}))$.
- $\alpha = f_{\bar{\alpha}2\alpha}(f_{\alpha 2\bar{\alpha}}(\alpha))$. Starting from a reconciliation α, $f_{\alpha 2\bar{\alpha}}$ removes from the $\alpha(u)$ sequences all elements corresponding to $\varnothing$ events, leaving the other elements in the same order in $\bar{\alpha}$. Then Algorithm 1 applied to $\bar{\alpha}$ just reintroduces the $\varnothing$ events at the same place and in the same number as they were in α originally. This results from the fact that Algorithm 1 mimics Property 6(5) for placing the $\varnothing$ events between the other elements, common to α and $\bar{\alpha}$: each call to *ArtificialPath* at line 17 returns a sequence $\varnothing_{u,k}$ of $\varnothing$ events preceding a non $\varnothing$ event (namely, $A_{\bar{\alpha}}[k]$) in $\alpha(u)$ and is concatenated after preceding elements in the sequence $\alpha(u)$ (processed for smaller k), followed by the event $A_{\bar{\alpha}}[k]$. Eventually, when the inner loop has been processed, the full sequence $\alpha(u)$ has been produced for the node u examined in the outer loop. Note that $A_{\bar{\alpha}}$ contains $\bar{\alpha}_\eta(u_p)$ at its beginning so that we always know in the inner loop the non $\varnothing$ event *preceding* $A_{\bar{\alpha}}[k]$ in the sense of Property 6(5), even for the first element in $A_{\bar{\alpha}}(u)$ (this allows to write "$k-1$" in lines 13-15). However this element is not added in $\alpha(u)$ due to k starting from 2 in the inner loop (line 11).

□