Nøjgaard *et al. Algorithms Mol Biol (2018) 13:2*
https://doi.org/10.1186/s13015-018-0121-8

Algorithms for Molecular Biology

RESEARCH Open Access

Time-consistent reconciliation maps and forbidden time travel

Nikolai Nøjgaard[1,2], Manuela Geiß[5], Daniel Merkle[2], Peter F. Stadler[5,6,7,8,9,10,11], Nicolas Wieseke[3] and Marc Hellmuth[1,4*]

Abstract

Background: In the absence of horizontal gene transfer it is possible to reconstruct the history of gene families from empirically determined orthology relations, which are equivalent to *event-labeled* gene trees. Knowledge of the event labels considerably simplifies the problem of reconciling a gene tree T with a species trees S, relative to the reconciliation problem without prior knowledge of the event types. It is well-known that optimal reconciliations in the unlabeled case may violate time-consistency and thus are not biologically feasible. Here we investigate the mathematical structure of the event labeled reconciliation problem with horizontal transfer.

Results: We investigate the issue of time-consistency for the event-labeled version of the reconciliation problem, provide a convenient axiomatic framework, and derive a complete characterization of time-consistent reconciliations. This characterization depends on certain weak conditions on the event-labeled gene trees that reflect conditions under which evolutionary events are observable at least in principle. We give an $\mathcal{O}(|V(T)|\log(|V(S)|))$-time algorithm to decide whether a time-consistent reconciliation map exists. It does not require the construction of explicit timing maps, but relies entirely on the comparably easy task of checking whether a small auxiliary graph is acyclic. The algorithms are implemented in C++ using the boost graph library and are freely available at https://github.com/Nojgaard/tc-recon.

Significance: The combinatorial characterization of time consistency and thus biologically feasible reconciliation is an important step towards the inference of gene family histories with horizontal transfer from orthology data, i.e., without presupposed gene and species trees. The fast algorithm to decide time consistency is useful in a broader context because it constitutes an attractive component for all tools that address tree reconciliation problems.

Keywords: Tree reconciliation, Horizontal gene transfer, Reconciliation map, Time-consistency, History of gene families

Background

Modern molecular biology describes the evolution of species in terms of the evolution of the genes that collectively form an organism's genome. In this picture, genes are viewed as atomic units whose evolutionary history *by definition* forms a tree. The phylogeny of species also forms a tree. This species tree is either interpreted as a consensus of the gene trees or it is inferred from other data. An interesting formal manner to define a species tree independent of genes and genetic data is discussed, e.g. in [1].

In this contribution, we assume that gene and species trees are given independently of each other. The relationship between gene and species evolution is therefore given by a reconciliation map that describes how the gene tree is embedded in the species tree: after all, genes reside in organisms, and thus at each point in time can be assigned to a species.

From a formal point of view, a reconciliation map μ identifies vertices of a gene tree with vertices and edges in the species tree in such a way that (partial) ancestor relations given by the genes are preserved by μ. Vertices in

*Correspondence: mhellmuth@mailbox.org
[1] Institute of Mathematics and Computer Science, University of Greifswald, Walther-Rathenau-Strasse 47, 17487 Greifswald, Germany
Full list of author information is available at the end of the article

the species tree correspond to speciation events. By definition, in a speciation event all genes are faithfully transmitted from the parent species into both (all) daughter species. Some of the vertices in the gene tree therefore correspond to speciation events. In gene duplications, two copies of a gene are formed from a single ancestral gene and then keep residing in the same species. In horizontal gene transfer (HGT) events, the original remains within the parental species, while the offspring copy "jumps" into a different branch of the species tree. Given a gene tree with event types assigned to its interior vertices, it is customary to define pairwise relations between genes depending on the event type of their last common ancestor [2–4].

Most of the literature on this topic assumes that both the gene tree and the species tree are known but no information is available of the type of events [5–8]. The aim is then to find a mapping of the gene tree T into the species tree S and, at least implicitly, an event-labeling on the vertices of the gene tree T. Here we take a different point of view and assume that T and the types of evolutionary events on T are known. This setting has ample practical relevance because event-labeled gene trees can be derived from the pairwise orthology relation [4, 9]. These relations in turn can be estimated directly from sequence data using a variety of algorithmic approaches that are based on the pairwise best match criterion and hence do not require any *a priori* knowledge of the topology of either the gene tree or the species tree, see e.g. [10–13].

Genes that share a common origin (homologs) can be classified into orthologs, paralogs, and xenologs depending whether they originated by a speciation, duplication or horizontal gene transfer (HGT) event [2, 4]. Recent advances in mathematical phylogenetics [9, 14] have shown that the knowledge of these event-relations (orthologs, paralogs and xenologs) suffices to construct event-labeled gene trees and, in some case, also a species tree [3, 15, 16].

Conceptually, both the gene tree and species tree are associated with a timing of each event. Reconciliation maps must preserve this timing information because there are *biologically infeasible* event labeled gene trees that cannot be reconciled with any species tree. In the absence of HGT, biologically feasibility can be characterized in terms of certain triples (rooted binary trees on three leaves) that are displayed by the gene trees [16]. In the presence of HGT such triples give at least necessary conditions for a gene tree being biologically feasible [15]. In particular, the timing information must be taken into account explicitly in the presence of HGT. That is, gene trees with HGT that must be mapped to species trees only in such a way that some genes do not travel back in time.

There have been several attempts in the literature to handle this issue, see e.g. [17] for a review. In [18, 19] a *single* HGT adds timing constraints to a time map for a reconciliation to be found. Time-consistency is then defined as the existence of a topological order of the digraph reflecting all the time constraints. In [20] NP-hardness was shown for finding a parsimonious time-consistent reconciliation based on a definition for time-consistency that in essence considers *pairs* of HGTs. However, the latter definitions are explicitly designed for *binary* gene trees and do not apply to non-binary gene trees, which are used here to model incomplete knowledge of the exact gene phylogenies. Different algorithmic approaches for tackling time-consistency exist [17] such as the inclusion of "time-zones" known for specific evolutionary events. It is worth noting that *a posteriori* modifications of time-inconsistent solutions will in general violate parsimony [18]. So far, no results have become available to determine the *existence* of time-consistent reconciliation maps given the (undated) species tree and the event-labeled gene tree.

Here, we introduce an axiomatic framework for time-consistent reconciliation maps and characterize for given event-labeled gene trees T and species trees S whether there exists a time-consistent reconciliation map. We provide an $\mathcal{O}(|V(T)|\log(|V(S)|))$-time algorithm that constructs a time-consistent reconciliation map if one exists.

Notation and preliminaries

We consider *rooted trees* $T=(V,E)$ (on L_T) with root $\rho_T \in V$ and leaf set $L_T \subseteq V$. A vertex $v \in V$ is called a *descendant* of $u \in V$, $v \preceq_T u$, and u is an *ancestor* of v, $u \succeq_T v$, if u lies on the path from ρ_T to v. As usual, we write $v \prec_T u$ and $u \succ_T v$ to mean $v \preceq_T u$ and $u \neq v$. The partial order $\succeq_T$ is known as the *ancestor order* of T; the root is the unique maximal element w.r.t $\succeq_T$. If $u \preceq_T v$ or $v \preceq_T u$ then u and v are *comparable* and otherwise, *incomparable*. We consider edges of rooted trees to be directed away from the root, that is, the notation for edges (u, v) of a tree is chosen such that $u \succ_T v$. If (u, v) is an edge in T, then u is called *parent* of v and v *child* of u. It will be convenient for the discussion below to extend the ancestor relation $\preceq_T$ on V to the union of the edge and vertex sets of T. More precisely, for the edge $e=(u,v) \in E$ we put $x \prec_T e$ if and only if $x \preceq_T v$ and $e \prec_T x$ if and only if $u \preceq_T x$. For edges $e=(u,v)$ and $f=(a,b)$ in T we put $e \preceq_T f$ if and only if $v \preceq_T b$. For $x \in V$, we write $L_T(x) := \{y \in L_T \mid y \preceq_T x\}$ for the set of leaves in the subtree $T(x)$ of T rooted in x.

For a non-empty subset of leaves $A \subseteq L$, we define $\mathrm{lca}_T(A)$, or the *least common ancestor of* A, to be the unique $\preceq_T$-minimal vertex of T that is an ancestor of every vertex in A. In case $A = \{u, v\}$, we put $\mathrm{lca}_T(u, v) := \mathrm{lca}_T(\{u, v\})$. We have in particular $u = \mathrm{lca}_T(L_T(u))$ for all $u \in V$. We will also frequently use that for any two non-empty vertex sets A, B of a tree, it holds that $\mathrm{lca}(A \cup B) = \mathrm{lca}(\mathrm{lca}(A), \mathrm{lca}(B))$.

A *phylogenetic tree* is a rooted tree such that no interior vertex in $v \in V \setminus L_T$ has degree two, except possibly the root. If L_T corresponds to a *set of genes* $\mathbb{G}$ or *species* $\mathbb{S}$, we call a phylogenetic tree on L_T *gene tree* or *species tree*, respectively. In this contribution we will *not* restrict the gene or species trees to be binary, although this assumption is made implicitly or explicitly in much of the literature on the topic. The more general setting allows us to model incomplete knowledge of the exact gene or species phylogenies. Of course, all mathematical results proved here also hold for the special case of binary phylogenetic trees.

In our setting a gene tree $T = (V, E)$ on $\mathbb{G}$ is equipped with an *event-labeling* map $t : V \cup E \to I \cup \{0, 1\}$ with $I = \{\bullet, \square, \triangle, \odot\}$ that assigns to each interior vertex v of T a value $t(v) \in I$ indicating whether v is a speciation event ($\bullet$), duplication event ($\square$) or HGT event ($\triangle$). It is convenient to use the special label $\odot$ for the leaves x of T. Moreover, to each edge e a value $t(e) \in \{0, 1\}$ is added that indicates whether e is a *transfer edge* (1) or not (0). Note, only edges (x, y) for which $t(x) = \triangle$ might be labeled as transfer edge. We write $\mathcal{E} = \{e \in E \mid t(e) = 1\}$ for the set of transfer edges in T. We assume here that all edges labeled "0" transmit the genetic material vertically, that is, from an ancestral species to its descendants.

We remark that the restriction $t_{|V}$ of t to the vertex set V coincides with the "symbolic dating maps" introduced in [21]; these have a close relationship with cographs [14, 22, 23]. Furthermore, there is a map $\sigma : \mathbb{G} \to \mathbb{S}$ that assigns to each gene the species in which it resides. The set $\sigma(M)$, $M \subseteq \mathbb{G}$, is the set of species from which the genes M are taken. We write $(T; t, \sigma)$ for the gene tree $T = (V, E)$ with event-labeling t and corresponding map σ.

Removal of the transfer edges from $(T; t, \sigma)$ yields a forest $T_{\overline{\mathcal{E}}} := (V, E \setminus \mathcal{E})$ that inherits the ancestor order on its connected components, i.e., $\preceq_{T_{\overline{\mathcal{E}}}}$ iff $x \preceq_T y$ and x, y are in same subtree of $T_{\overline{\mathcal{E}}}$ [20]. Clearly $\preceq_{T_{\overline{\mathcal{E}}}}$ uniquely defines a root for each subtree and the set of descendant leaf nodes $L_{T_{\overline{\mathcal{E}}}}(x)$.

In order to account for duplication events that occurred before the first speciation event, we need to add an extra vertex and an extra edge "above" the last common ancestor of all species in the species tree $S = (V, E)$. Hence, we add an additional vertex to V (that is now the new root ρ_S of S) and the additional edge $(\rho_S, \mathrm{lca}_S(\mathbb{S}))$ to E. Strictly speaking S is not a phylogenetic tree in the usual sense, however, it will be convenient to work with these augmented trees. For simplicity, we omit drawing the augmenting edge $(\rho_S, \mathrm{lca}_S(\mathbb{S}))$ in our examples.

Observable scenarios

The true history of a gene family, as it is considered here, is an arbitrary sequence of speciation, duplication, HGT, and gene loss events. The applications we envision for the theory developed, here, however assume that the gene tree and its event labels are inferred from (sequence) data, i.e., $(T; t, \sigma)$ is restricted to those labeled trees that can be constructed at least in principle from observable data. The issue here are gene losses that may completely eradicate the information on parts of the history. Specifically, we require that $(T; t, \sigma)$ satisfies the following three conditions:

(O1) Every internal vertex v has degree at least 3, except possibly the root which has degree at least 2.

(O2) Every HGT node has at least one transfer edge, $t(e) = 1$, and at least one non-transfer edge, $t(e) = 0$;

(O3)

- **(a)** If x is a speciation vertex, then there are at least two distinct children v, w of x such that the species V and W that contain v and w, resp., are incomparable in S.
- **(b)** If (v, w) is a transfer edge in T, then the species V and W that contain v and w, resp., are incomparable in S.

Condition **(O1)** ensures that every event leaves a historical trace in the sense that there are at least two children that have survived in at least two of its subtrees. If this were not the case, no evidence would be left for all but one descendant tree, i.e., we would have no evidence that event v ever happened. We note that this condition was used, e.g. in [16] for scenarios without HGT. Condition **(O2)** ensures that for an HGT event a historical trace remains of both the transferred and the non-transferred copy. If there is no transfer edge, we have no evidence to classify v as a HGT node. Conversely, if all edges were transfers, no evidence of the lineage of origin would be available and any reasonable inference of the gene tree from data would assume that the gene family was vertically transmitted in at least one of the lineages in which it is observed. In particular, Condition **(O2)** implies that for each internal vertex there is a path consisting entirely of non-transfer edges to some leaf. This excludes in particular scenarios in which a gene is transferred to a different "host" and later reverts back to descendants of the original lineage without any surviving offspring in the intermittent host lineage. Furthermore, a speciation vertex x cannot be observed from data if it does not "separate" lineages, that is, there are two leaf descendants of distinct children of x that are in distinct species. However, here we only assume to have the weaker Condition **(O3.a)** which ensures that any "observable" speciation vertex x separates at least locally two lineages. In other words, if all children of x would be contained in species that are comparable in S or, equivalently, in the same lineage of S, then there is no clear historical trace that justifies x to be a speciation vertex. In particular, most-likely there are two leaf descendants of distinct children of x that are in the same species even if only $T_{\overline{\mathcal{E}}}$ is considered. Hence, x would rather be classified as a duplication than as a speciation upon inference of the event labels from actual data. Analogously, if $(v, w) \in \mathcal{E}$ then v signifies the transfer event itself but w refers to the next (visible) event in the gene tree T. Given that (v, w) is a HGT-edge in the observable part, in a "true history" v is contained in a species V that transmits its genetic material (maybe along a path of transfers) to a contemporary species Z that is an ancestor of the species W containing w. Clearly, the latter allows to have $V \succeq_S W$ which happens if the path of transfers points back to the descendant lineage of V in S. In this case the transfer edge (v, w) must be placed in the species tree such that $\mu(v)$ and $\mu(w)$ are comparable in S. However, then there is no evidence that this transfer ever happened, and thus v would be rather classified as speciation or duplication vertex.

Assuming that **(O2)** is satisfied, we obtain the following useful result:

Lemma 1 *Let $\mathcal{T}_1, \dots, \mathcal{T}_k$ be the connected components of $T_{\overline{\mathcal{E}}}$ with roots $\rho_1, \dots, \rho_k$, respectively. If* **(O2)** *holds, then, $\{L_{T_{\overline{\mathcal{E}}}}(\rho_1), \dots, L_{T_{\overline{\mathcal{E}}}}(\rho_k)\}$ forms a partition of $\mathbb{G}$.*

Proof Since $L_{T_{\overline{\mathcal{E}}}}(\rho_i) \subseteq V(T)$, it suffices to show that $L_{T_{\overline{\mathcal{E}}}}(\rho_i)$ does not contain vertices of $V(T) \setminus \mathbb{G}$. Note, $x \in L_{T_{\overline{\mathcal{E}}}}(\rho_i)$ with $x \notin \mathbb{G}$ is only possible if all edges (x, y) are removed.

Let $x \in V$ with $t(x) = \triangle$ such that all edges (x, y) are removed. Thus, all such edges (x, y) are contained in $\mathcal{E}$. Therefore, every edge of the form (x, y) is a transfer edge; a contradiction to **(O2)**. □

We will show in Proposition 1 that **(O1)**, **(O2)**, and **(O3)** together imply two important properties of event labeled species trees, **(Σ1)** and **(Σ2)**, which play a crucial role for the results reported here.

(Σ1) If $t(x) = \bullet$, then there are distinct children v, w of x in T such that $\sigma(L_{T_{\overline{\mathcal{E}}}}(v)) \cap \sigma(L_{T_{\overline{\mathcal{E}}}}(w)) = \emptyset$.
(Σ2) If $(v, w) \in \mathcal{E}$, then $\sigma(L_{T_{\overline{\mathcal{E}}}}(v)) \cap \sigma(L_{T_{\overline{\mathcal{E}}}}(w)) = \emptyset$.

Intuitively, **(Σ1)** is true because within a component $T_{\overline{\mathcal{E}}}$ no genetic material is exchanged between non-comparable nodes. Thus, a gene separated in a speciation event necessarily ends up in distinct species in the absence of horizontal transfer. It is important to note that we do not require the converse: $\sigma(L_{T_{\overline{\mathcal{E}}}}(y)) \cap \sigma(L_{T_{\overline{\mathcal{E}}}}(y')) = \emptyset$ does not imply $t(\mathrm{lca}_T(L_{T_{\overline{\mathcal{E}}}}(y) \cup L_{T_{\overline{\mathcal{E}}}}(y')) = \bullet$, that is, the last common ancestor of two sets of genes from different species is not necessarily a speciation vertex.

Now consider a transfer edge $(v, w) \in \mathcal{E}$, i.e., $t(v) = \triangle$. Then $T_{\overline{\mathcal{E}}}(v)$ and $T_{\overline{\mathcal{E}}}(w)$ are subtrees of distinct connected components of $T_{\overline{\mathcal{E}}}$. Since HGT amounts to the transfer of genetic material *across* distinct species, the genes v and w must be contained in distinct species X and Y, respectively. Since no genetic material is transferred between contemporary species X' and Y' in $T_{\overline{\mathcal{E}}}$, where X' and Y' is a descendant of X and Y, respectively we derive **(Σ1)**.

Proposition 1 *Conditions* **(O1)**–**(O3)** *imply* **(Σ1)** *and* **(Σ2)**.

Proof Since **(O2)** is satisfied we can apply Lemma 1 and conclude that neither $\sigma(L_{T_{\overline{\mathcal{E}}}}(v)) = \emptyset$ nor $\sigma(L_{T_{\overline{\mathcal{E}}}}(w)) = \emptyset$. Let $x \in V(T)$ with $t(x) = \bullet$. By Condition (O1) x has (at least two) children. Moreover, **(O3)** implies that there are (at least) two children v and w in T that are contained in distinct species V and W that are incomparable in S. Note, the edges (x, v) and (x, w) remain in $T_{\overline{\mathcal{E}}}$, since only transfer edges are removed. Since no transfer is contained in $T_{\overline{\mathcal{E}}}$, the genetic material v and w of V and W, respectively, is always vertically transmitted. Therefore, for any leaf $v' \in L_{T_{\overline{\mathcal{E}}}}(v)$ we have $\sigma(v') \preceq_S V$ and for any leaf $w' \in L_{T_{\overline{\mathcal{E}}}}(w)$ we have $\sigma(w') \preceq_S W$ in S. Assume now for contradiction, that $\sigma(L_{T_{\overline{\mathcal{E}}}}(v)) \cap \sigma(L_{T_{\overline{\mathcal{E}}}}(w)) \neq \emptyset$. Let $z_1 \in L_{T_{\overline{\mathcal{E}}}}(v)$ and $z_2 \in L_{T_{\overline{\mathcal{E}}}}(w)$ with $\sigma(z_1) = \sigma(z_2) = Z$. Since $Z \preceq_S V, W$ and S is a tree, the species V and W must be comparable in S; a contradiction to **(O3)**. Hence, Condition **(Σ1)** is satisfied.

To see **(Σ2)**, note that since **(O2)** is satisfied we can apply Lemma 1 and conclude that neither $\sigma(L_{T_{\overline{\mathcal{E}}}}(v)) = \emptyset$ nor $\sigma(L_{T_{\overline{\mathcal{E}}}}(w)) = \emptyset$. Let $(v, w) \in \mathcal{E}$. By **(O3)** the species containing V and W are incomparable in S. Now we can argue along the same lines as in the proof for **(Σ2)** to conclude that $\sigma(L_{T_{\overline{\mathcal{E}}}}(v)) \cap \sigma(L_{T_{\overline{\mathcal{E}}}}(w)) = \emptyset$. □

From here on we simplify the notation a bit and write $\sigma_{T_{\overline{\mathcal{E}}}}(u) := \sigma(L_{T_{\overline{\mathcal{E}}}}(u))$. We are aware of the fact that condition **(O3)** cannot be checked directly for a given event-labeled gene tree. In contrast, **(Σ1)** and **(Σ2)** are easily determined. Hence, in the remainder of this paper we consider the more general case, that is, gene trees that satisfy **(O1)**, **(O2)**, **(Σ1)**, and **(Σ1)**.

DTL-scenario and time-consistent reconciliation maps

In case that the event-labeling of T is unknown, but the gene tree T and a species tree S are given, the authors in [20, 24] provide an axiom set, called DTL-scenario, to reconcile T with S. This reconciliation is then used to infer the event-labeling t of T. Instead of defining a DTL-scenario as octuple [20, 24], we use here the notation established above:

Definition 1 (*DTL-scenario*) For a given gene tree $(T; t, \sigma)$ on $\mathbb{G}$ and a species tree S on $\mathbb{S}$ the map $\gamma : V(T) \to V(S)$ maps the gene tree into the species tree such that

(I) For each leaf $x \in \mathbb{G}, \gamma(u) = \sigma(u)$.
(II) If $u \in V(T) \setminus \mathbb{G}$ with children v, w, then

(a) $\gamma(u)$ is not a proper descendant of $\gamma(v)$ or $\gamma(w)$, and
(b) At least one of $\gamma(v)$ or $\gamma(w)$ is a descendant of $\gamma(u)$.

(III) (u, v) is a transfer edge if and only if $\gamma(u)$ and $\gamma(v)$ are incomparable.
(IV) If $u \in V(T) \setminus \mathbb{G}$ with children v, w, then

(a) $t(u) = \triangle$ if and only if either (u, v) or (u, w) is a transfer-edge,
(b) If $t(u) = \bullet$, then $\gamma(u) = \mathrm{lca}_S(\gamma(v), \gamma(w))$ and $\gamma(v), \gamma(w)$ are incomparable,
(c) If $t(u) = \square$, then $\gamma(u) \succeq \mathrm{lca}_S(\gamma(v), \gamma(w))$.

DTL-scenarios are explicitly defined for fully resolved binary gene and species trees. Indeed, Fig. 1 (right) shows a valid reconciliation between a gene tree T and a species tree S that is not consistent with DTL-scenario. To see this, let us call the duplication vertex v. The vertex v and the leaf a are both children of the speciation vertex ρ_T. Condition (IVb) implies that a and v must be incomparable. However, this is not possible since $\gamma(v) \succeq_S \mathrm{lca}_S(B, C)$ (Cond. (IVc)) and $\gamma(a) = A$ (Cond. (I)) and therefore, $\gamma(v) \succeq_S \mathrm{lca}_S(B, C) = \mathrm{lca}_S(A, B, C) \succ_S \gamma(a)$.

The problem of reconciliations between gene trees and species tree is formalized in terms of so-called DTL-scenarios in the literature [20, 24]. This framework, however, usually assumes that the event labels t on T are unknown, while a species tree S is given. The "usual" DTL axioms, furthermore, explicitly refer to binary, fully resolved gene and species trees. We therefore use a different axiom set here that is a natural generalization of the framework introduced in [16] for the HGT-free case:

Definition 2 Let $T = (V, E)$ and $S = (W, F)$ be phylogenetic trees on $\mathbb{G}$ and $\mathbb{S}$, resp., $\sigma : \mathbb{G} \to \mathbb{S}$ the assignment of genes to species and $t : V \cup E \to \{\bullet, \square, \triangle, \odot\} \cup \{0, 1\}$ an event labeling on T. A map $\mu : V \to W \cup F$ is a reconciliation map if for all $v \in V$ it holds that:

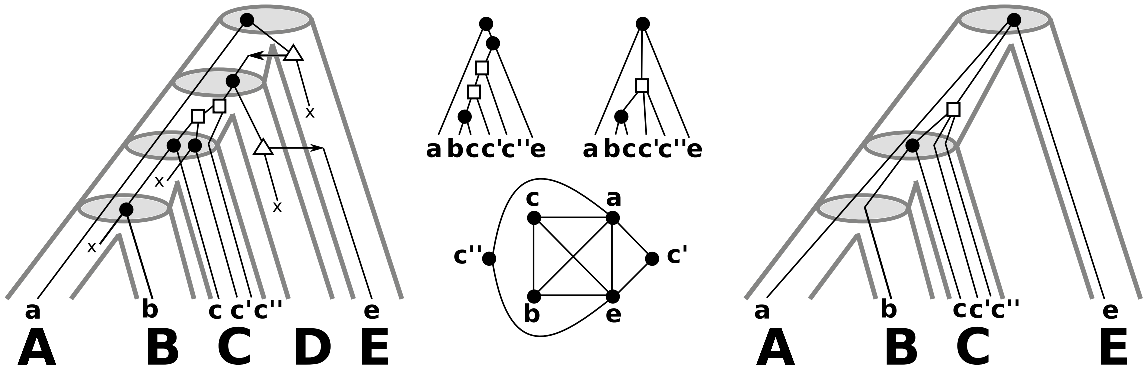

Fig. 1 Left: A "true" evolutionary scenario for a gene tree with leaf set $\mathbb{G}$ evolving along the tube-like species trees is shown. The symbol "x" denotes losses. All speciations along the path from the root ρ_T to the leaf a are followed by losses and we omit drawing them. Middle: The observable gene tree is shown in the upper-left. The orthology graph $G = (\mathbb{G}, E)$ (edges are placed between genes x, y for which $t(\mathrm{lca}(x, y)) = \bullet$) is drawn in the lower part. This graph is a cograph and the corresponding *non-binary* gene tree T on $\mathbb{G}$ that can be constructed from such data is given in the upper-right part (cf. [3, 4, 14] for further details). Right: Shown is species trees S on $\mathbb{S} = \sigma(\mathbb{G})$ with reconciled gene tree T. The reconciliation map μ for T and S is given implicitly by drawing the gene tree T within S. Note, this reconciliation is not consistent with DTL-scenarios [20, 24]. A DTL-scenario would require that the duplication vertex and the leaf a are incomparable in S. Note, a non-binary duplication or HGT vertex v can always be "binary resolved" such that the newly created vertices are placed on the same edge $\mu(v)$ as v. However, there are cases that show that non-binary speciation vertices cannot be "binary resolved". For instance, for the non-binary gene tree T there is no way to resolve its root without violating the conditions of a reconciliation map (cf. [15, Fig. 3]). Yet, such cases strongly imply that the speciation event must have been followed by (several) duplication/HGT events that are not observable due to losses

(M1) Leaf Constraint. If $t(v) = \odot$, then $\mu(v) = \sigma(v)$.
(M2) Event Constraint.

(i) If $t(v) = \bullet$, then $\mu(v) = \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v))$.
(ii) If $t(v) \in \{\square, \triangle\}$, then $\mu(v) \in F$.
(iii) If $t(v) = \triangle$ and $(v, w) \in \mathcal{E}$, then $\mu(v)$ and $\mu(w)$ are incomparable in S.

(M3) Ancestor Constraint.
Suppose $v, w \in V$ with $v \prec_{T_{\overline{\mathcal{E}}}} w$.

(i) If $t(v), t(w) \in \{\square, \triangle\}$, then $\mu(v) \preceq_S \mu(w)$,
(ii) Otherwise, i.e., at least one of $t(v)$ and $t(w)$ is a speciation $\bullet$, $\mu(v) \prec_S \mu(w)$.

We say that S is a species tree for $(T; t, \sigma)$ if a reconciliation map $\mu : V \to W \cup F$ exists.

For the special case that gene and species trees are binary, Definition 2 is equivalent to the definition of a DTL-scenario, which is summarized in the following

Theorem 1 *For a binary gene tree $(T; t, \sigma)$ and a binary species tree S there is a DTL-scenario if and only if there is a reconciliation μ for $(T; t, \sigma)$ and S.*

The proof of Theorem 1 is a straightforward but tedious case-by-case analysis. In order to keep this section readable, we relegate the proof of Theorem 1 to "Proof of Theorem 1" section. Figure 1 shows an example of a biologically plausible reconciliation of non-binary trees that is valid w.r.t. Definition 2 but does not satisfy the conditions of a DTL-scenario.

Condition **(M1)** ensures that each leaf of T, i.e., an extant gene in $\mathbb{G}$, is mapped to the species in which it resides. Conditions **(M2.i)** and **(M2.ii)** ensure that each inner vertex of T is either mapped to a vertex or an edge in S such that a vertex of T is mapped to an interior vertex of S if and only if it is a speciation vertex. Condition **(M2.i)** might seem overly restrictive, an issue to which we will return below. Condition **(M2.iii)** satisfies condition **(O3)** and maps the vertices of a transfer edge in a way that they are incomparable in the species tree, since a HGT occurs between distinct (co-existing) species. It becomes void in the absence of HGT; thus Definition 2 reduces to the definition of reconciliation maps given in [16] for the HGT-free case. Importantly, condition **(M3)** refers only to the connected components of $T_{\overline{\mathcal{E}}}$ since comparability w.r.t. $\prec_{T_{\overline{\mathcal{E}}}}$ implies that the path between x and y in T does not contain transfer edges. It ensures that the ancestor order $\preceq_T$ of T is preserved along all paths that do not contain transfer edges.

We will make use of the following bound that effectively restricts how close to the leafs the image of a vertex in the gene tree can be located.

Lemma 2 *If $\mu : (T; t, \sigma) \to S$ satisfies* **(M1)** *and* **(M3)**, *then $\mu(u) \succeq_S \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))$ for any $u \in V(T)$.*

Proof If u is a leaf, then by Condition **(M1)** $\mu(u) = \sigma(u)$ and we are done. Thus, let u be an interior vertex. By Condition **(M3)**, $z \preceq_S \mu(u)$ for all $z \in \sigma_{T_{\overline{\mathcal{E}}}}(u)$. Hence, if $\mu(u) \prec_S \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))$ or if $\mu(u)$ and $\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u)))$ are incomparable in S, then there is a $z \in \sigma_{T_{\overline{\mathcal{E}}}}(u)$ such that z and $\mu(u)$ are incomparable; contradicting **(M3)**. □

Condition **(M2.i)** implies in particular the weaker property "(M2.i') if $t(v) = \bullet$ then $\mu(v) \in W$". In the light of Lemma 2, $\mu(v) = \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v))$ is the lowest possible choice for the image of a speciation vertex. Clearly, this restricts the possibly exponentially many reconciliation maps for which $\mu(v) \succ_S \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v))$ for a speciation vertices v to only those that satisfy **(M2.i)**. However, the latter is justified by the observation that if v is a speciation vertex with children u, w, then there is only one unique piece of information given by the gene tree to place $\mu(v)$, that is, the unique vertex x in S with children y, z such that $\sigma_{T_{\overline{\mathcal{E}}}}(u) \subseteq L_S(y)$ and $\sigma_{T_{\overline{\mathcal{E}}}}(w) \subseteq L_S(z)$. The latter arguments easily generalizes to the case that v has more than two children in T. Moreover, any *observable* speciation node $v' \succ_T v$ closer to the root than v must be mapped to a node ancestral to $\mu(v)$ due to (M3.ii). Therefore, we require $\mu(v) = x = \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v))$ here.

If S is a species tree for the gene tree (T, t, σ) then there is no freedom in the construction of a reconciliation map μ on the set $\{x \in V(T) \mid t(x) \in \{\bullet, \odot\}\}$. The duplication and HGT vertices of T, however, can be placed differently. As a consequence there is a possibly exponentially large set of reconciliation maps from (T, t, σ) to S.

From a biological point of view, however, the notion of reconciliation used so far is too weak. In the absence of HGT, subtrees evolve independently and hence, the linear order of points along each path from root to leaf is consistent with a global time axis. This is no longer true in the presence of HGT events, because HGT events imply additional time-consistency conditions. These stem from the fact that the appearance of the HGT copy in a distant subtree of S is concurrent with the HGT event. To investigate this issue in detail, we introduce time maps and the notion of time-consistency, see Figs. 2, 3, 4 for illustrative examples.

Definition 3 (*Time Map*) The map $\tau_T : V(T) \to \mathbb{R}$ is a time map for the rooted tree T if $x \prec_T y$ implies $\tau_T(x) > \tau_T(y)$ for all $x, y \in V(T)$.

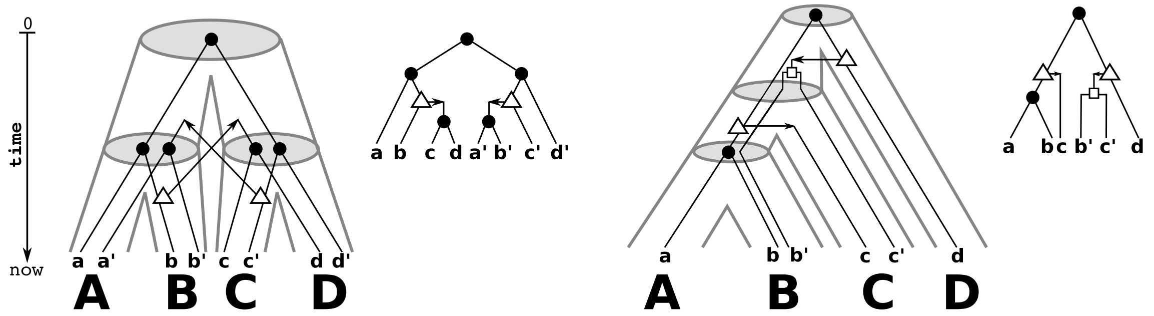

Fig. 2 Shown are two (tube-like) species trees with reconciled gene trees. The reconciliation map μ for T and S is given implicitly by drawing the gene tree (upper right to the respective species tree) within the species tree. In the left example, the map μ is unique. However, μ is not time-consistent and thus, there is no time consistent reconciliation for T and S. In the example on the right hand side, μ is time-consistent

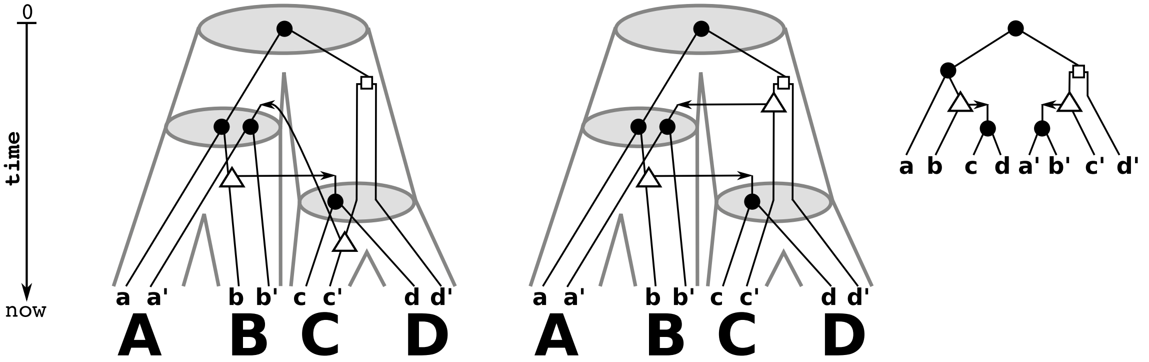

Fig. 3 Shown are a gene tree $(T; t, \sigma)$ (right) and two identical (tube-like) species trees S (left and middle). There are two possible reconciliation maps for T and S that are given implicitly by drawing T within the species tree S. These two reconciliation maps differ only in the choice of placing the HGT-event either on the edge $(\mathrm{lca}_S(C,D), C)$ or on the edge $(\mathrm{lca}_S(\{A,B,C,D\}), \mathrm{lca}_S(C,D))$. In the first case, it is easy to see that μ would not be time-consistent, i.e., there are no time maps τ_T and τ_S that satisfy **(C1)** and **(C2)**. The reconciliation map μ shown in the middle is time-consistent

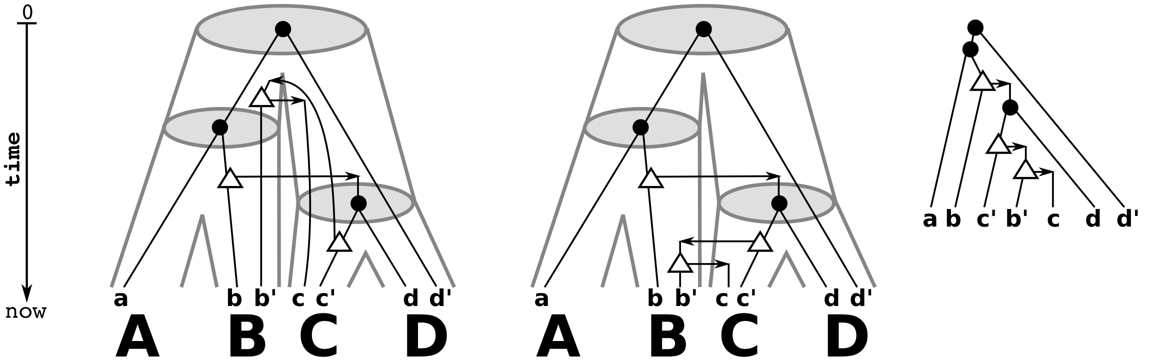

Fig. 4 Shown are a gene tree $(T; t, \sigma)$ (right) and two identical (tube-like) species trees S (left and middle). There are two possible reconciliation maps for T and S that are given implicitly by drawing T within the species tree S. The left reconciliation maps each gene tree vertex as high as possible into the species tree. However, in this case only the middle reconciliation map is time-consistent

Definition 4 A reconciliation map μ from $(T; t, \sigma)$ to S is time-consistent if there are time maps τ_T for T and τ_S for S for all $u \in V(T)$ satisfying the following conditions:

(C1) *If* $t(u) \in \{\bullet, \odot\}$, *then* $\tau_T(u) = \tau_S(\mu(u))$.

(C2) *If* $t(u) \in \{\square, \triangle\}$ *and, thus* $\mu(u) = (x, y) \in E(S)$, *then* $\tau_S(y) > \tau_T(u) > \tau_S(x)$.

Condition **(C1)** is used to identify the time-points of speciation vertices and leaves u in the gene tree with the time-points of their respective images $\mu(u)$ in the species trees. In particular, all genes u that reside in the same species must be assigned the same time point $\tau_T(u) = \tau_S(\sigma(u))$. Analogously, all speciation vertices in T that are mapped to the same speciation in S are assigned matching time stamps, i.e., if $t(u) = t(v) = \bullet$ and $\mu(u) = \mu(v)$ then $\tau_T(u) = \tau_T(v) = \tau_S(\mu(u))$.

To understand the intuition behind **(C2)** consider a duplication or HGT vertex u. By construction of μ it is mapped to an edge of S, i.e., $\mu(u) = (x, y)$ in S. The time point of u must thus lie between time points of x and y. Now suppose $(u, v) \in \mathcal{E}$ is a transfer edge. By construction, u signifies the transfer event itself. The node v, however, refers to the next (visible) event in the gene tree. Thus $\tau_T(u) < \tau_T(v)$. In particular, $\tau_T(v)$ must not be misinterpreted as the time of introducing the HGT-duplicate into the new lineage. While this time of course exists (and in our model coincides with the timing of the transfer event) it is not marked by a visible event in the new lineage, and hence there is no corresponding node in the gene tree T.

W.l.o.g. we fix the time axis so that $\tau_T(\rho_T) = 0$ and $\tau_S(\rho_S) = -1$. Thus, $\tau_S(\rho_S) < \tau_T(\rho_T) < \tau_T(u)$ for all $u \in V(T) \setminus \{\rho_T\}$.

Clearly, a necessary condition to have biologically feasible gene trees is the existence of a reconciliation map μ. However, not all reconciliation maps are time-consistent, see Fig. 2.

Definition 5 An event-labeled gene tree $(T; t, \sigma)$ is biologically feasible if there exists a time-consistent reconciliation map from $(T; t, \sigma)$ to some species tree S.

As a main result of this contribution, we provide simple conditions that characterize (the existence of) time-consistent reconciliation maps and thus, provides a first step towards the characterization of biologically feasible gene trees.

Theorem 2 *Let μ be a reconciliation map from $(T; t, \sigma)$ to S. There is a time-consistent reconciliation map from $(T; t, \sigma)$ to S if and only if there are two time-maps τ_T and τ_S for T and S, respectively, such that the following conditions are satisfied for all $x \in V(S)$:*

(D1) *If $\mu(u) = x$, for some $u \in V(T)$, then $\tau_T(u) = \tau_S(x)$.*
(D2) *If $x \preceq_S \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))$ for some $u \in V(T)$ with $t(u) \in \{\square, \triangle\}$, then $\tau_S(x) > \tau_T(u)$.*
(D3) *If $\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u) \cup \sigma_{T_{\overline{\mathcal{E}}}}(v)) \preceq_S x$ for some $(u, v) \in \mathcal{E}$, then $\tau_T(u) > \tau_S(x)$.*

Proof In what follows, x and u denote vertices in S and T, respectively.

Assume that there is a time-consistent reconciliation map μ from $(T; t, \sigma)$ to S, and thus two time-maps τ_S and τ_T for S and T, respectively, that satisfy **(C1)** and **(C2)**.

To see **(D1)**, observe that if $\mu(u) = x \in V(S)$, then **(M1)** and **(M2)** imply that $t(u) \in \{\bullet, \odot\}$. Now apply **(C1)**.

To show **(D2)**, assume that $t(u) \in \{\square, \triangle\}$ and $x \preceq_S lca_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))$. By Condition **(M2)** it holds that $\mu(u) = (y, z) \in E(S)$. Together with Lemma 2 we obtain that $x \preceq_S \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u)) \preceq_S z \prec_S \mu(u)$. By the properties of τ_S we have

$$\tau_S(x) \geq \tau_S(\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u)) \geq \tau_S(z) \overset{(C2)}{>} \tau_T(u).$$

To see **(D3)**, assume that $(u, v) \in \mathcal{E}$ and $z := \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u) \cup \sigma_{T_{\overline{\mathcal{E}}}}(v)) \preceq_S x$. Since $t(u) = \triangle$ and by **(M2ii)**, we have $\mu(u) = (y, y') \in E(S)$. Thus, $\mu(u) \prec_S y$. By **(M2iii)** $\mu(u)$ and $\mu(v)$ are incomparable and therefore, we have either $\mu(v) \prec_S y$ or $\mu(v)$ and y are incomparable. In either case we see that $y \preceq_S z$, since Lemma 2 implies that $\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u)) \preceq_S \mu(u)$ and $\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v)) \preceq_S \mu(v)$. In summary, $\mu(u) \prec_S y \preceq_S z \preceq_S x$. Therefore,

$$\tau_T(u) \overset{(C2)}{>} \tau_S(y) \geq \tau_S(z) \geq \tau_S(x).$$

Hence, conditions **(D1)**–**(D3)** are satisfied.

To prove the converse, assume that there exists a reconciliation map μ that satisfies **(D1)**–**(D3)** for some time-maps τ_T and τ_S. In the following we will make use of τ_S and τ_T to construct a time-consistent reconciliation map μ'.

First we define "anchor points" by $\mu'(v) = \mu(v)$ for all $v \in V(T)$ with $t(v) \in \{\bullet, \odot\}$. Condition **(D1)** implies $\tau_T(v) = \tau_S(\mu(v))$ for these vertices, and therefore μ' satisfies **(C1)**.

The next step will be to show that for each vertex $u \in V(T)$ with $t(u) \in \{\square, \triangle\}$ there is a unique edge (x, y) along the path from $\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))$ to ρ_S with $\tau_S(x) < \tau_T(u) < \tau_S(y)$. We set $\mu'(u) = (x, y)$ for these points. In the final step we will show that μ' is a valid reconciliation map.

Consider the unique path $\mathcal{P}_u$ from $\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))$ to ρ_S. By construction, $\tau_S(\rho_S) < \tau_T(\rho_T) \leq \tau_T(u)$ and by Condition **(D2)** we have $\tau_T(u) < \tau_S(\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u)))$. Since τ_S is a time map for S, every edge $(x, y) \in E(S)$ satisfies $\tau_S(x) < \tau_S(y)$. Therefore, there is a unique edge $(x_u, y_u) \in E(S)$ along $\mathcal{P}_u$ such that either $\tau_S(x_u) < \tau_T(u) < \tau_S(y_u)$, $\tau_S(x_u) = \tau_T(u) < \tau_S(y_u)$, or $\tau_S(x_u) < \tau_T(u) = \tau_S(y_u)$. The addition of a sufficiently small perturbation ϵ_u to $\tau_T(u)$ does not violate the conditions for τ_T being a time-map for T. Clearly ϵ_u can be chosen to break the equalities in the latter two cases in such a way that $\tau_S(x_u) < \tau_T(u) < \tau_S(y_u)$ for each vertex $u \in V(T)$ with $t(u) \in \{\square, \triangle\}$. We then continue with the perturbed version of τ_T and set $\mu'(u) = (x_u, y_u)$. By construction, μ' satisfies **(C2)**.

It remains to show that μ' is a valid reconciliation map from $(T; t, \sigma_{T_{\overline{\mathcal{E}}}})$ to S. Again, let $\mathcal{P}_u$ denote the unique path from $\operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))$ to ρ_S for any $u \in V(T)$.

By construction, Conditions **(M1)**, **(M2i)**, **(M2ii)** are satisfied. To check condition **(M2iii)**, assume $(u, v) \in \mathcal{E}$. The original map μ is a valid reconciliation map, and thus, Lemma 2 implies that $\operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u)) \prec_S \mu(u)$ and $\operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v)) \preceq_S \mu(v)$. Since $\mu(u)$ and $\mu(v)$ are incomparable in S and $\operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u) \cup \sigma_{T_{\overline{\mathcal{E}}}}(v))$ lies on both paths $\mathcal{P}_u$ and $\mathcal{P}_v$ we have $\mu(u), \mu(v) \preceq_S \operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u) \cup \sigma_{T_{\overline{\mathcal{E}}}}(v)) =: x$. In particular, $x \neq \operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))$ and $x \neq \operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v))$.

Conditions **(D1)** and **(D2)** imply that $\tau_S(x) < \tau_T(u) < \tau_S(\operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u)))$ and $\tau_S(x) < \tau_T(v) \leq \tau_S(\operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v)))$. By construction of μ', the vertex u is mapped to a unique edge $e_u = (x_u, y_u)$ and v is mapped either to $\operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v)) \neq x$ or to the unique edge $e_v = (x_v, y_v)$, respectively. In particular, $\mu'(u)$ lies on the path $\mathcal{P}'$ from x to $\operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))$ and $\mu'(v)$ lies one the path $\mathcal{P}''$ from x to $\operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v))$. The paths $\mathcal{P}'$ and $\mathcal{P}''$ are edge-disjoint and have x as their only common vertex. Hence, $\mu'(u)$ and $\mu'(v)$ are incomparable in S, and **(M2iii)** is satisfied.

In order to show **(M3)**, assume that $u \prec_{T_{\overline{\mathcal{E}}}} v$. Since $u \prec_{T_{\overline{\mathcal{E}}}} v$, we have $\sigma_{T_{\overline{\mathcal{E}}}}(u) \subseteq \sigma_{T_{\overline{\mathcal{E}}}}(v)$. Hence, $\operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u)) \preceq \operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v)) \preceq_S \rho_S$. In other words, $\operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v))$ lies on the path $\mathcal{P}_u$ and thus, $\mathcal{P}_v$ is a subpath of $\mathcal{P}_u$. By construction of μ', both $\mu'(u)$ and $\mu'(v)$ are comparable in S. Moreover, since $\tau_T(u) > \tau_T(v)$ and by construction of μ', it immediately follows that $\mu'(u) \preceq_S \mu'(v)$.

Its now an easy task to verify that **(M3)** is fulfilled by considering the distinct event-labels in **(M3i)** and **(M3ii)**, which we leave to the reader. □

Interestingly, the existence of a time-consistent reconciliation map from a gene tree T to a species tree S can be characterized in terms of a time map defined on T, only.

Theorem 3 *Let μ be a reconciliation map from $(T; t, \sigma)$ to S. There is a time-consistent reconciliation map $(T; t, \sigma)$ to S if and only if there is a time map τ_T such that for all $u, v, w \in V(T)$:*

(T1) If $t(u) = t(v) \in \{\bullet, \odot\}$ then

(a) If $\mu(u) = \mu(v)$, then $\tau_T(u) = \tau_T(v)$.
(b) If $\mu(u) \prec_S \mu(v)$, then $\tau_T(u) > \tau_T(v)$.

(T2) If $t(u) \in \{\bullet, \odot\}$, $t(v) \in \{\square, \triangle\}$ and $\mu(u) \preceq_S \operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v))$, then $\tau_T(u) > \tau_T(v)$.
(T3) If $(u, v) \in \mathcal{E}$ and $\operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u) \cup \sigma_{T_{\overline{\mathcal{E}}}}(v)) \preceq_S \operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(w))$ for some $w \in V(T)$, then $\tau_T(u) > \tau_T(w)$.

Proof Suppose that μ is a time-consistent reconciliation map from $(T; t, \sigma)$ to S. By Definition 4 and Theorem 2, there are two time maps τ_T and τ_S that satisfy **(D1)**–**(D3)**. We first show that τ_T also satisfies **(T1)**–**(T3)**, for all $u, v \in V(T)$. Condition **(T1a)** is trivially implied by **(D1)**. Let $t(u), t(v) \in \{\bullet, \odot\}$, and $\mu(u) \prec_S \mu(v)$. Since τ_T and τ_S are time maps, we may conclude that

$$\tau_T(u) \overset{(D1)}{=} \tau_S(\mu(u)) < \tau_S(\mu(v)) \overset{(D1)}{=} \tau_T(v).$$

Hence, **(T1b)** is satisfied.

Now, assume that $t(u) \in \{\bullet, \odot\}$, $t(v) \in \{\square, \triangle\}$ and $\mu(u) \preceq_S \operatorname{lca}(\sigma_{T_{\overline{\mathcal{E}}}}(v))$. By the properties of τ_S, we have:

$$\tau_T(u) \overset{(D1)}{=} \tau_S(\mu(u)) \overset{(D2)}{>} \tau_T(v).$$

Hence, **(T2)** is fulfilled.

Finally, assume that $(u, v) \in \mathcal{E}$, and $x := \operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u) \cup \sigma_{T_{\overline{\mathcal{E}}}}(v)) \preceq_S \operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(w))$ for some $w \in V(T)$. Lemma 2 implies that $\operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(w)) \preceq_S \mu(w)$ and we obtain

$$\tau_T(w) \overset{(D2)}{<} \tau_S(x) \leq \tau_S(\operatorname{lca}(\sigma_{T_{\overline{\mathcal{E}}}}(w))) \overset{(D3)}{<} \tau_T(u).$$

Hence, **(T3)** is fulfilled.

To see the converse, assume that there exists a reconciliation map μ that satisfies **(T1)**–**(T3)** for some time map τ_T. In the following we construct a time map τ_S for S that satisfies **(D1)**–**(D3)**. To this end, we first set

$$\tau_S(x) = \begin{cases} -1 & \text{if } x = \rho_S \\ \tau_T(v) & \text{else if } v \in \mu^{-1}(x) \\ * & \text{else, i.e., } \mu^{-1}(x) = \emptyset \text{ and } x \neq \rho_S. \end{cases}$$

We use the symbol $*$ to denote the fact that so far no value has been assigned to $\tau_S(x)$. Note, by **(M2i)** and **(T1a)** the value $\tau_S(x)$ is uniquely determined and thus, by construction, **(D1)** is satisfied. Moreover, if $x, y \in V(S)$ have non-empty preimages w.r.t. μ and $x \prec_S y$, then we can use the fact that τ_T is a time map for T together with condition **(T1)** to conclude that $\tau_S(x) > \tau_S(y)$.

If $x \in V(S)$ with $a \in \mu^{-1}(x)$, then **(T2)** implies **(D2)** [by **(D1)** and setting $u = a$ in **(T2)** and **(T3)** implies **(D3)** [by

(D1) and setting $w = a$ in **(T3)**]. Thus, **(D2)** and **(D3)** is satisfied for all $x \in V(S)$ with $\mu^{-1}(x) \neq \emptyset$.

Using our choices $\tau_S(\rho_T) = 0$ and $\tau_S(\rho_S) = -1$ for the augmented root of S, we must have $\mu^{-1}(\rho_S) = \emptyset$. Thus, $\rho_S \succ_S \operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v))$ for any $v \in V(T)$. Hence, **(D2)** is trivially satisfied for ρ_S. Moreover, $\tau_T(\rho_T) = 0$ implies $\tau_T(u) > \tau_S(\rho_S)$ for any $u \in V(T)$. Hence, **(D3)** is always satisfied for ρ_S.

In summary, Conditions **(D1)**–**(D3)** are met for any vertex $x \in V(S)$ that up to this point has been assigned a value, i.e., $\tau_S(x) \neq *$.

We will now assign to all vertices $x \in V(S)$ with $\mu^{-1}(x) = \emptyset$ a value $\tau_S(x)$ in a stepwise manner. To this end, we give upper and lower bounds for the possible values that can be assigned to $\tau_S(x)$. Let $x \in V(S)$ with $\tau_S(x) = *$. Set

$$\mathsf{LO}(x) = \{\tau_S(y) \mid x \prec_S y, y \in V(S) \text{ and } \tau_S(y) \neq *\}$$
$$\mathsf{UP}(x) = \{\tau_S(y) \mid x \succ_S y, y \in V(S) \text{ and } \tau_S(y) \neq *\}.$$

We note that $\mathsf{LO}(x) \neq \emptyset$ and $\mathsf{UP}(x) \neq \emptyset$ because the root and the leaves of S already have been assigned a value τ_S in the initial step. In order to construct a valid time map τ_S we must ensure $\max(\mathsf{LO}(x)) < \tau_S(x) < \min(\mathsf{UP}(x))$.

Moreover, we strengthen the bounds as follows. Put

$$\begin{aligned}\mathsf{lo}(x) &= \{\tau_T(u) \mid t(u) \in \{\square, \triangle\}, x \preceq_S \operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))\}\\ \mathsf{up}(x) &= \{\tau_T(u) \mid (u,v) \in \mathcal{E} \text{ and}\\ &\qquad \operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u) \cup \sigma_{T_{\overline{\mathcal{E}}}}(v)) \preceq_S x\,\}.\end{aligned}$$

Observe that $\max(\mathsf{lo}(x)) < \min(\mathsf{up}(x))$, since otherwise there are vertices $u, w \in V(T)$ with $\tau_T(w) \in \mathsf{lo}(x)$ and $\tau_T(u) \in \mathsf{up}(x)$ and $\tau_T(w) \geq \tau_T(u)$. However, this implies that $\operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u) \cup \sigma_{T_{\overline{\mathcal{E}}}}(v)) \preceq_S x \preceq \operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(w))$; a contradiction to **(T3)**.

Since **(D2)** is satisfied for all vertices y that obtained a value $\tau_S(y) \neq *$, we have $\max(\mathsf{lo}(x)) < \min(\mathsf{UP}(x))$. Likewise because of **(D3)**, it holds that $\max(\mathsf{LO}(x)) < \min(\mathsf{up}(x))$. Thus we set $\tau_S(x)$ to an arbitrary value such that

$$\max(\mathsf{LO}(x) \cup \mathsf{lo}(x)) < \tau_S(x) < \min(\mathsf{UP}(x) \cup \mathsf{up}(x)).$$

By construction, **(D1)**, **(D2)**, and **(D3)** are satisfied for all vertices in $V(S)$ that have already obtained a time value distinct from $*$. Moreover, for all such vertices with $x \prec_T y$ we have $\tau_S(x) > \tau_S(y)$. In each step we chose a vertex x with $\tau_S(x) = *$ that obtains then a real-valued time stamp. Hence, in each step the number of vertices that have value $*$ is reduced by one. Therefore, repeating the latter procedure will eventually assign to all vertices a real-valued time stamp such that, in particular, τ_S satisfies **(D1)**, **(D2)**, and **(D3)** and thus is indeed a time map for S. □

From the algorithmic point of view it is desirable to design methods that allow to check whether a reconciliation map is time-consistent. Moreover, given a gene tree T and species tree S we wish to decide whether there exists a time-consistent reconciliation map μ, and if so, we should be able to construct μ.

To this end, observe that any constraints given by Definition 3, Theorem 2 **(D2)**–**(D3)**, and Definition 4 **(C2)** can be expressed as a total order on $V(S) \cup V(T)$, while the constraints **(C1)** and **(D1)** together suggest that we can treat the preimage of any vertex in the species tree as a "single vertex". In fact we can create an auxiliary graph in order to answer questions that are concerned with time-consistent reconciliation maps.

Definition 6 Let μ be a reconciliation map from $(T; t, \sigma)$ to S. The auxiliary graph A is defined as a directed graph with a vertex set $V(A) = V(S) \cup V(T)$ and an edge-set $E(A)$ that is constructed as follows

(A1) *For each* $(u, v) \in E(T)$ *we have* $(u', v') \in E(A)$, *where*

$$u' = \begin{cases} \mu(u) & \text{if } t(u) \in \{\odot, \bullet\} \\ u & \text{otherwise} \end{cases}$$

and

$$v' = \begin{cases} \mu(v) & \text{if } t(v) \in \{\odot, \bullet\} \\ v & \text{otherwise} \end{cases},$$

(A2) *For each* $(x, y) \in E(S)$ *we have* $(x, y) \in E(A)$..

(A3) *For each* $u \in V(T)$ *with* $t(u) \in \{\square, \triangle\}$ *we have* $(u, lca_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))) \in E(A)$.

(A4) *For each* $(u, v) \in \mathcal{E}$ *we have* $(lca_S(\sigma_{T_{\overline{\mathcal{E}}}}(u) \cup \sigma_{T_{\overline{\mathcal{E}}}}(v)), u) \in E(A)$.

(A5) *For each* $u \in V(T)$ *with* $t(u) \in \{\triangle, \square\}$ *and* $\mu(u) = (x, y) \in E(S)$ *we have* $(x, u) \in E(A)$ *and* $(u, y) \in E(A)$.

We define A_1 and A_2 as the subgraphs of A that contain only the edges defined by **(A1)**, **(A2)**, **(A5)** and **(A1)**, **(A2)**, **(A3)**, **(A4)**, respectively.

We note that the edge sets defined by conditions **(A1)** through **(A5)** are not necessarily disjoint. The mapping of vertices in T to edges in S is considered only in condition **(A5)**. The following two theorems are the key results of this contribution.

Theorem 4 *Let* μ *be a reconciliation map from* $(T; t, \sigma)$ *to* S. *The map* μ *is time-consistent if and only if the auxiliary graph* A_1 *is a directed acyclic graph (DAG).*

Proof Assume that μ is time-consistent. By Theorem 2, there are two time-maps τ_T and τ_S satisfying **(C1)** and **(C2)**. Let $\tau = \tau_T \cup \tau_S$ be the map from $V(T) \cup V(S) \to \mathbb{R}$. Let A' be the directed graph with $V(A') = V(S) \cup V(T)$ and set for all $x, y \in V(A')$: $(x, y) \in E(A')$ if and only if $\tau(x) < \tau(y)$. By construction A' is a DAG since τ provides a topological order on A' [25].

We continue to show that A' contains all edges of A_1.

To see that **(A1)** is satisfied for $E(A')$ let $(u, v) \in E(T)$. Note, $\tau(v) > \tau(u)$, since τ_T is a time map for T and by construction of τ. Hence, all edges $(u, v) \in E(T)$ are also contained in A', independent from the respective event-labels $t(u)$, $t(v)$. Moreover, if $t(u)$ or $t(v)$ are speciation vertices or leaves, then **(C1)** implies that $\tau_S(\mu(u)) = \tau_T(u) > \tau_T(v)$ or $\tau_T(u) > \tau_T(v) = \tau_S(\mu(v))$. By construction of τ, all edges satisfying **(A1)** are contained in $E(A')$. Since τ_S is a time map for S, all edges as in **(A2)** are contained in $E(A')$. Finally, **(C2)** implies that all edges satisfying **(A5)** are contained in $E(A')$.

Although, A' might have more edges than required by **(A1)**, **(A2)** and **(A5)**, the graph A_1 is a subgraph of A'. Since A' is a DAG, also A_1 is a DAG.

For the converse assume that A_1 is a directed graph with $V(A_1) = V(S) \cup V(T)$ and edge set $E(A_1)$ as constructed in Definition 6 **(A1)**, **(A2)** and **(A5)**. Moreover, assume that A_1 is a DAG. Hence, there is is a topological order τ on A_1 with $\tau(x) < \tau(y)$ whenever $(x, y) \in E(A_1)$. In what follows we construct the time-maps τ_T and τ_S such that they satisfy **(C1)** and **(C2)**. Set $\tau_S(x) = \tau(x)$ for all $x \in V(S)$. Additionally, set for all $u \in V(T)$:

$$\tau_T(u) = \begin{cases} \tau(\mu(u)) & \text{if } t(u) \in \{\odot, \bullet\} \\ \tau(u) & \text{otherwise.} \end{cases}$$

By construction it follows that (C1) is satisfied. Due to (A2), τ_S is a valid time map for S. It follows from the construction and (A1) that τ_T is a valid time map for T. Assume now that $u \in V(T)$, $t(u) \in \{\square, \triangle\}$, and $\mu(u) = (x, y) \in E(S)$. Since τ provides a topological order we have:

$$\tau(x) \overset{\textbf{(A5)}}{<} \tau(u) \overset{\textbf{(A5)}}{<} \tau(y).$$

By construction, it follows that $\tau_S(x) < \tau_T(u) < \tau_S(y)$ satisfying **(C2)**. □

Theorem 5 *Assume there is a reconciliation map μ from $(T; t, \sigma)$ to S. There is a time-consistent reconciliation map, possibly different from μ, from $(T; t, \sigma)$ to S if and only if the auxiliary graph A_2 (defined on μ) is a DAG.*

Proof Let μ be a reconciliation map for $(T; t, \sigma)$ and S and μ' be a time-consistent reconciliation map for $(T; t, \sigma)$ and S. Let A_2 and A_2' be the auxiliary graphs that satisfy Definition 6 **(A1)** – **(A4)** for μ and μ', respectively. Since $\mu(u) = \mu'(u)$ for all $u \in V(T)$ with $t(u) \in \{\odot, \bullet\}$ and **(A2)** – **(A4)** don't rely on the explicit reconciliation map, it is easy to see that $A_2 = A_2'$.

Now we can re-use similar arguments as in the proof of Theorem 4. Assume there is a time-consistent reconciliation map $(T; t, \sigma)$ to S. By Theorem 2, there are two time-maps τ_T and τ_S satisfying **(D1)**-**(D3)**. Let τ and A' be defined as in the proof of Theorem 4.

Analogously to the proof of Theorem 4, we show that A' contains all edges of A_2. Application of **(D1)** immediately implies that all edges satisfying **(A1)** and **(A2)** are contained in $E(A')$. By condition **(D2)**, it yields $(u, lca_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))) \in E(A')$ and **(D3)** implies $(lca_S(\sigma_{T_{\overline{\mathcal{E}}}}(u) \cup \sigma_{T_{\overline{\mathcal{E}}}}(v)), u) \in E(A')$. We conclude by the same arguments as before that the graph A_2 is a DAG.

For the converse, assume we are given the directed acyclic graph A_2. As before, there is is a topological order τ on A_2 with $\tau(x) < \tau(y)$ only if $(x, y) \in E(A_2)$. The time-maps τ_T and τ_S are given as in the proof of Theorem 1.

By construction, it follows that **(D1)** is satisfied. Again, by construction and the Properties **(A1)** and **(A2)**, τ_S and τ_T are valid time-maps for S and T respectively.

Assume now that $u \in V(T)$, $t(u) \in \{\square, \triangle\}$, and $x \preceq_S \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))$ for some $x \in V(S)$. Since there is a topological order on $V(A_2)$, we have

$$\tau(x) \overset{\textbf{(A2)}}{\geq} \tau(\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))) \overset{\textbf{(A3)}}{>} \tau(u).$$

By construction, it follows that $\tau_S(x) > \tau_T(u)$. Thus, **(D2)** is satisfied.

Finally assume that $(u, v) \in \mathcal{E}$ and $\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u) \cup \sigma_{T_{\overline{\mathcal{E}}}}(v)) \preceq_S x$ for some $x \in V(S)$. Again, since τ provides a topological order, we have:

$$\tau(x) \overset{\textbf{(A2)}}{\leq} \tau(\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u) \cup \sigma_{T_{\overline{\mathcal{E}}}}(v))) \overset{\textbf{(A4)}}{<} \tau(u).$$

By construction, it follows that $\tau_S(x) < \tau_T(u)$, satisfying **(D3)**.

Thus τ_T and τ_S are valid time maps satisfying **(D1)**–**(D3)**. □

Naturally, Theorems 4 or 5 can be used to devise algorithms for deciding time-consistency. To this end, the efficient computation of $\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))$ for all $u \in V(T)$ is necessary. This can be achieved with Algorithm 2 in $O(|V(T)|\log(|V(S)|))$. More precisely, we have the following statement:

Lemma 3 *For a given gene tree $(T = (V, E); t, \sigma)$ and a species tree $S = (W, F)$, Algorithm 2 correctly computes $\ell(u) = \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))$ for all $u \in V(T)$ in $O(|V|\log(|W|))$ time.*

Proof Let $u \in V(T)$. In what follows, we show that $\ell(u) = \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))$. In fact, the algorithm is (almost) a depth first search through T that assigns the (species tree) vertex $\ell(u)$ to u if and only if every child v of u has obtained an assignment $\ell(v)$ (cf. Line (9)–(10)). That there are children v with non-empty $\ell(v)$ at some point is ensured by Line (7). That is, if $t(u) = \odot$, then $\ell(u) = \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u)) = \sigma(u)$. Now, assume there is an interior vertex $u \in V(T)$, where every child v has been assigned a value $\ell(v)$, then

$$
\begin{aligned}
&\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))\\
&= \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(\{\sigma_{T_{\overline{\mathcal{E}}}}(v) \mid (u,v) \in E(T) \text{ and } t(u,v) = 0\}))\\
&= \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(\{\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v)) \mid (u,v) \in E(T) \text{ and } t(u,v) = 0\}))\\
&= \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(\{\ell(v) \mid (u,v) \in E(T) \text{ and } t(u,v) = 0\}))
\end{aligned}
$$

The latter is achieved by Line (10).

Since T is a tree and the algorithm is in effect a depth first search through T, the while loop runs at most $O(V(T) + E(T))$ times, and thus in $O(V(T))$ time.

The only non-constant operation within the while loop is the computation of lca_S in Line (10). Clearly lca_S of a set of vertices $C = \{c_1, c_2 \dots c_k\}$, where $c_i \in V(S)$, for all $c_i \in C$ can be computed as sequence of lca_S operations taking two vertices: $\mathrm{lca}_S(c_1, \mathrm{lca}_S(c_2, \dots \mathrm{lca}_S(c_{k-1}, c_k)))$, each taking $O(\lg(|V(S)|))$ time. Note however, that since Line (10) is called exactly once for each vertex in T, the number of lca_S operations taking two vertices is called at most $|E(T)|$ times through the entire algorithm. Hence, the total time complexity is $O(|V(T)|\lg(|V(S)|))$. □

Let S be a species tree for $(T; t, \sigma)$, that is, there is a valid reconciliation between the two trees. Algorithm 1 describes a method to construct a time-consistent reconciliation map for $(T; t, \sigma)$ and S, if one exists, else "No time-consistent reconciliation map exists" is returned. First, an arbitrary reconciliation map μ that satisfies the condition of Definition 2 is computed. Second, Theorem 5 is utilized and it is checked whether the auxiliary graph A_2 is not a DAG in which case no time-consistent map μ exists for $(T; t, \sigma)$ and S. Finally, if A_2 is a DAG, then we continue to adjust μ to become time-consistent. The latter is based on Theorem 2, see the proof of Theorems 2 and 6 for details.

Theorem 6 *Let $S = (W, F)$ be species tree for the gene tree $(T = (V, E); t, \sigma)$. Algorithm 1 correctly determines whether there is a time-consistent reconciliation map μ and in the positive case, returns such a μ in $O(|V|\log(|W|))$ time.*

Proof In order to produce a time-consistent reconciliation map, we first construct some valid reconciliation map μ from $(T; t, \sigma)$ to S. Using the lca-map ℓ from Algorithm 2, μ will be adjusted to become time-consistent, if possible.

By assumption, there is a reconciliation map from $(T; t, \sigma)$ to S. The for-loop (Line (3)–(5)) ensures that each vertex $u \in V$ obtained a value $\mu(u)$. We continue to show that μ is a valid reconciliation map satisfying **(M1)–(M3)**.

Algorithm 1 Check existence and construct time-consistent reconciliation map.

```
Precondition: $S = (W,F)$ is a species tree for $T = (V,E)$.
 1: $\ell \leftarrow$ ComputeLcaSigma($(T;t,\sigma),S$)
 2: $\mu(u) \leftarrow \emptyset$ for all $u \in V$                ▷ "$\emptyset$" means uninitialized
 3: for all $u \in V$ do
 4:     if $t(u) \in \{\bullet,\odot\}$ then $\mu(u) \leftarrow \ell(u)$
 5:     else $\mu(u) \leftarrow (p(\ell(u)),\ell(u))$   ▷ $p(\ell(u))$ denotes the parent of $\ell(u)$
 6: Compute the auxiliary graph $A_2$
 7: if $A_2$ contains a cycle then return "No time-consistent reconciliation map exists."
 8: Let $\tau : V(A_2) \rightarrow \mathbb{R}$ such that if $(x,y) \in E(A_2)$ then $\tau(x) < \tau(y)$
 9:  ▷ W.l.o.g. we can assume that $\tau(x) \neq \tau(y)$ for all $x,y \in V(A_2)$
10: $\tau_S \leftarrow$ A time map such that $\tau_S(x) = \tau(x)$ for all $x \in W$
11: $\tau_T \leftarrow$ A time map such that $\tau_T(u) = \tau(\mu(u))$ if $t(u) \in \{\bullet,\odot\}$, otherwise $\tau_T(u) = \tau(u)$ for all $u \in V$.
12: for $u \in V$ where $t(u) \in \{\square,\triangle\}$ do
13:     while it does not hold that $\tau_S(x) < \tau_T(u) < \tau_S(y)$ for $(x,y) = \mu(u)$ do
14:         $\mu(u) \leftarrow (p(x),x)$
15: return $\mu$
```

Algorithm 2 Compute $\ell(u) = \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))$ for all $u \in V(T)$

```
 1: function ComputeLcaSigma($(T;t,\sigma),S$)
 2:     $\ell(u) \leftarrow \emptyset$ for all $u \in V(T)$             ▷ "$\emptyset$" means uninitialized
 3:     $A \leftarrow$ empty stack
 4:     $A.push(\rho_T)$
 5:     while $A$ is not empty do
 6:         $u \leftarrow A.pop()$
 7:         if $t(u) = \odot$ then $\ell(u) \leftarrow \sigma(u)$
 8:         else if $\ell(v) = \emptyset$ for some child $v$ of $u$ then $A.push(u)$, $A.push(v)$
 9:         else
10:             $\ell(u) \leftarrow \mathrm{lca}_S(\{\ell(v) \| (u,v) \in E(T)$ and $t((u,v)) = 0\})$
11:     return $\ell$
```

Assume that $t(u) = \odot$, in this case $\ell(u) = \sigma(u)$, and thus **(M1)** is satisfied. If $t(u) = \bullet$, it holds that $\mu(u) = \ell(u) = \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))$, thus satisfying **(M2i)**. Note that $\rho_S \succ_S \ell(u)$, and hence, $\mu(u) \in F$ by Line (5), implying that **(M2ii)** is satisfied. Now, assume $t(u) = \triangle$ and $(u, v) \in \mathcal{E}$. By assumption, we know there exists a reconciliation map from T to S, thus by (**Σ1**):

$$\sigma_{T_{\overline{\mathcal{E}}}}(u) \cap \sigma_{T_{\overline{\mathcal{E}}}}(v) = \emptyset$$

It follows that, $\ell(u)$ is incomparable to $\ell(v)$, satisfying **(M2iii)**.

Now assume that $u, v \in V$ and $u \prec_{T_{\overline{\mathcal{E}}}} v$. Note that $\sigma_{T_{\overline{\mathcal{E}}}}(u) \subseteq \sigma_{T_{\overline{\mathcal{E}}}}(v)$. It follows that $\ell(u) = \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u)) \preceq_S \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v)) = \ell(v)$. By construction, **(M3)** is satisfied. Thus, μ is a valid reconciliation map.

By Theorem 5, two time maps τ_T and τ_S satisfying **(D1)–(D3)** only exists if the auxiliary graph A build on Line (7) is a DAG. Thus if $A := A_2$ contains a cycle, no such time-maps exists and the statement "No time-consistent reconciliation map exists." is returned (Line (7)). On the other hand, if A is a DAG, the construction in Line (8)–(11) is identical to the construction used in the proof of Theorem 5. Hence correctness of this part of the algorithm follows directly from the proof of Theorem 5.

Finally, we adjust μ to become a time-consistent reconciliation map.. By the latter arguments, τ_T and τ_S satisfy **(D1)–(D3)** w.r.t. to μ. Note, that μ is chosen to be the "lowest point" where a vertex $u \in V$ with $t(u) \in \{\square, \triangle\}$ can be mapped, that is, $\mu(u)$ is set to $(p(x), x)$ where $x = \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))$. However, by the arguments in the proof of Theorem 2, there is a unique edge $(y, z) \in W$ on the path from x to ρ_S such that $\tau_S(y) < \tau_T(u) < \tau_S(z)$. The latter is ensured by choosing a different value for distinct vertices in $V(A)$, see comment in Line (9). Hence, Line (14) ensures, that $\mu(u)$ is mapped on the correct edge such that **(C2)** is satisfied. It follows that adjusted μ is a valid time-consistent reconciliation map.

We are now concerned with the time-complexity. By Lemma 3, computation of ℓ in Line (1) takes $O(|V|\log(|W|))$ time and the for-loop (Line (3)-(5)) takes $O(|V|)$ time. We continue to show that the auxiliary graph A (Line (6)) can be constructed in $O(|V|\log(|W|))$ time.

Since we know $\ell(u) = \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))$ for all $u \in V$ and since T and S are trees, the subgraph with edges satisfying **(A1)–(A3)** can be constructed in $O(|V| + |W| + |E| + |F|) = O(|V| + |W|)$ time. To ensure **(A4)**, we must compute for a possible transfer edges $(u, v) \in \mathcal{E}$ the vertex $\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u) \cup \sigma_{T_{\overline{\mathcal{E}}}}(v))$. which can be done in $O(\log(|W|))$ time. Note, the number of transfer edges is bounded by the number of possible transfer event $O(|V|)$. Hence, generating all edges satisfying **(A4)** takes $O(|V|(\log(|W|))$ time. In summary, computing A can done in $O(|V| + |W| + |V|(\log(|W|)) = O(|V|(\log(|W|))$ time.

To detect whether A contains cycles one has to determine whether there is a topological order τ on $V(A)$ which can be done via depth first search in $O(|V(A)| + |E(A)|)$ time. Since $|V(A)| = |V| + |W|$ and $O(|E(A)|) = O(|F| + |E| + |W| + |V|)$ and S, T are trees, the latter task can be done in $O(|V| + |W|)$ time. Clearly, Line (10)-(11) can be performed on $O(|V| + |W|)$ time.

Finally, we have to adjust μ according to τ_T and τ_S. Note, that for each $u \in V$ with $t(u) \in \{\square, \triangle\}$ (Line (12)) we have possibly adjust μ to the next edge $(p(x), x)$. However, the possibilities for the choice of $(p(x), x)$ is bounded by by the height of S, which is in the worst case $\log(|W|)$. Hence, the for-loop in Line (12) has total-time complexity $O(|V|\log(|W|))$.

In summary, the overall time complexity of Algorithm 1 is $O(|V|\log(|W|))$. □

So far, we have shown how to find a time consistent reconciliation map μ given a species tree S and a single gene tree T. In practical applications, however, one often considers more than one gene family, and thus, a set of gene trees $F = \{(T_1; t_1, \sigma_1), \dots, (T_n; t_n, \sigma_n)\}$ that has to be reconciled with one and the same species tree S.

In this case it is possible to aggregate all gene trees $(T_i; t_i, \sigma_i) \in F$ to a single gene tree $(T; t, \sigma)$ that is constructed from F by introducing an artificial duplication as the new root of all T_i. More precisely, $T = (V, E)$ is constructed from F such that $V = \{\rho_T\} \cup \bigcup_{i=1}^{n} V(T_i)$ and $E = \bigcup_{i=1}^{n} (E(T_i) \cup \{(\rho_T, \rho_{T_i})\})$. Moreover, the event-labeling map t is defined as

$$t(x) = \begin{cases} t_i(x) & \text{if } x \in V(T_i) \cup E(T_i) \\ \square & \text{if } x = t(\rho_T) \\ 0 & \text{if } x = (\rho_T, \rho_{T_i}) \end{cases}$$

Finally, $\sigma(x) = \sigma_i(x)$ for all $x \in L_{T_i}$.

Finding a time consistent reconciliation for a species tree S and a set of gene trees F then corresponds to finding a time map τ_S for S and a time map τ_T for the aggregated gene tree $(T; t, \sigma)$, such that **(D1)–(D3)** are satisfied.

If there exists a time consistent reconciliation map μ from $(T; t, \sigma)$ to S then, by Theorem 2, there exists the two time maps τ_T and τ_S that satisfy **(D1)–(D3)**. But then τ_T and τ_S also satisfy **(D1)–(D3)** w.r.t. any $(T_i; t_i, \sigma_i) \in F$ and therefore, μ immediately gives a time-consistent reconciliation map for each $(T_i; t_i, \sigma_i) \in F$.

Outlook and summary

We have characterized here whether a given event-labeled gene tree $(T; t, \sigma)$ and species tree S can be reconciled in a time-consistent manner in terms of two auxiliary graphs A_1 and A_2 that must be DAGs. These are defined in terms of given reconciliation maps. This condition yields an $O(|V|\log(|W|))$-time algorithm to check whether a given reconciliation map μ is time-consistent, and an algorithm with the same time complexity for the construction of a time-consistent reconciliation maps, provided one exists.

Our results depend on three conditions on the event-labeled gene trees that are motivated by the fact that event-labels can be assigned to internal vertices of gene trees only if there is observable information on the event. The question which event-labeled gene trees are actually observable given an arbitrary, true evolutionary scenario deserves further investigation in future work. Here we have used conditions that arguable are satisfied when gene trees are inferred using sequence comparison and synteny information. A more formal theory of observability is still missing, however.

Our results point to an efficient way of deciding whether a *given* pair of gene and species tree can be time-consistently reconciled. Such gene and species trees can be obtained from genomic sequence data using the following workflow: (i) Estimate putative orthologs and HGT events using e.g. one of the methods detailed in [11, 12, 26–38], respectively. Importantly, this step uses only sequence data as input and does not require the construction of either gene or species trees. (ii) Correct these estimates in order to derive "biologically feasible" homology relations as described in [15, 16, 26, 39–44]. The result of this step are (not necessarily fully resolved) gene trees together with event-labels. (iii) Extract "informative triples" from the event-labeled gene tree. These imply necessary conditions for gene trees to be biologically feasible [15, 16].

In general, there will be exponentially many putative species trees. This begs the question whether there is *at least one* species tree S for a gene tree and if so, how to construct S. In the absence of HGT, the answer is known: time-consistent reconciliation maps are fully characterized in terms of "informative triples" [16]. Hence, the central open problem that needs to be addressed in further research are sufficient conditions for the existence of a time-consistent species tree *given* an event-labeled gene tree with HGT.

Proof of Theorem 1

We show that Definition 2 is is equivalent to the traditional definition of a DTL-scenario [20, 24] in the special case that both the gene tree and species trees are binary. To this end we establish a series of lemmas detailing some useful properties of reconciliation maps.

Lemma 4 *Let μ be a reconciliation map from $(T;t,\sigma)$ to S and assume that T is binary. Then the following conditions are satisfied:*

1. *If $v,w\in V(T)$ are in the same connected component of $T_{\overline{\mathcal{E}}}$, then $\mu(\operatorname{lca}_{T_{\overline{\mathcal{E}}}}(v,w))\succeq_S \operatorname{lca}_S(\mu(v),\mu(w))$. Let u be an arbitrary interior vertex of T with children v, w, then:*
2. *$\mu(u)$ and $\mu(v)$ are incomparable in S if and only if $(u,v)\in\mathcal{E}$.*
3. *If $t(u)=\bullet$, then $\mu(v)$ and $\mu(w)$ are incomparable in S.*
4. *If $\mu(v),\mu(w)$ are comparable or $\mu(u)\succ_S \operatorname{lca}_S(\mu(v),\mu(w))$, then $t(u)=\square$.*

Proof We prove the Items 1 – 4 separately. Recall, Lemma 1 implies that $\sigma(L_{T_{\overline{\mathcal{E}}}}(x))\neq\emptyset$ for all $x\in V(T)$.

Proof of Item 1: Let v and w be distinct vertices of T that are in the same connected component of $T_{\overline{\mathcal{E}}}$. Consider the unique path P connecting w with v in $T_{\overline{\mathcal{E}}}$. This path P is uniquely subdivided into a path P' and a path P'' from $\operatorname{lca}_{T_{\overline{\mathcal{E}}}}(v,w)$ to v and w, respectively. Condition **(M3)** implies that the images of the vertices of P' and P'' under μ, resp., are ordered in S with regards to $\preceq_S$ and hence, are contained in the intervals Q' and Q'' that connect $\mu(\operatorname{lca}_{T_{\overline{\mathcal{E}}}}(v,w))$ with $\mu(v)$ and $\mu(w)$, respectively. In particular, $\mu(\operatorname{lca}_{T_{\overline{\mathcal{E}}}}(v,w))$ is the largest element (w.r.t. $\preceq_S$) in the union of $Q'\cup Q''$ which contains the unique path from $\mu(v)$ to $\mu(w)$ and hence also $\operatorname{lca}_S(\mu(v),\mu(w))$.

Proof of Item 2: If $(u,v)\in\mathcal{E}$ then, $t(u)=\triangle$ and **(M2iii)** implies that $\mu(u)$ and $\mu(v)$ are incomparable.

To see the converse, let $\mu(u)$ and $\mu(v)$ be incomparable in S. Item **(M3)** implies that for any edge $(x,y)\in E(T_{\overline{\mathcal{E}}})$ we have $\mu(y)\preceq_S\mu(x)$. However, since $\mu(u)$ and $\mu(v)$ are incomparable it must hold that $(u,v)\notin E(T_{\overline{\mathcal{E}}})$. Since (u, v) is an edge in the gene tree T, $(u,v)\in\mathcal{E}$ is a transfer edge.

Proof of Item 3: Let $t(u)=\bullet$. Since none of (u, v) and (u, w) are transfer-edges, it follows that both edges are contained in $T_{\overline{\mathcal{E}}}$.

Then, since T is a binary tree, it follows that $L_{T_{\overline{\mathcal{E}}}}(u)=L_{T_{\overline{\mathcal{E}}}}(v)\cup L_{T_{\overline{\mathcal{E}}}}(w)$ and therefore, $\sigma_{T_{\overline{\mathcal{E}}}}(u)=\sigma_{T_{\overline{\mathcal{E}}}}(v)\cup\sigma_{T_{\overline{\mathcal{E}}}}(w)$.

Therefore and by Item **(M2i)**,

$$\begin{aligned}\mu(u)&=\operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))=\operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v)\cup\sigma_{T_{\overline{\mathcal{E}}}}(w))\\&=\operatorname{lca}_S(\operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v)),\operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(w))).\end{aligned}$$

Assume for contradiction that $\mu(v)$ and $\mu(w)$ are comparable, say, $\mu(w)\succeq_S\mu(v)$. By Lemma 2, $\mu(w)\succeq_S\mu(v)\succeq_S\operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v))$ and $\mu(w)\succeq_S\operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(w))$. Thus,

$$\mu(w)\succeq_S\operatorname{lca}_S(\operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v)),\operatorname{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(w))).$$

Thus, $\mu(w)\succeq_S\mu(u)$; a contradiction to **(M3ii)**.

Proof of Item 4: Let $\mu(v),\mu(w)$ be comparable in S. Item 3 implies that $t(u)\neq\bullet$. Assume for contradiction that $t(u)=\triangle$. Since by **(O2)** only one of the edges (u, v)

and (u, w) is a transfer edge, we have either $(u, v) \in \mathcal{E}$ or $(u, w) \in \mathcal{E}$. W.l.o.g. let $(u, v) \in \mathcal{E}$ and $(u, w) \in E(T_{\overline{\mathcal{E}}})$. By Condition **(M3)**, $\mu(u) \succeq_S \mu(w)$. However, since $\mu(v)$ and $\mu(w)$ are comparable in S, also $\mu(u)$ and $\mu(v)$ are comparable in S; a contradiction to Item 2. Thus, $t(u) \neq \triangle$. Since each interior vertex is labeled with one event, we have $t(u) = \square$.

Assume now that $\mu(u) \succ_S \mathrm{lca}_S(\mu(v), \mu(w))$. Hence, $\mu(u)$ is comparable to both $\mu(v)$ and $\mu(w)$ and thus, **(M2iii)** implies that $t(u) \neq \triangle$. Lemma 2 implies $\mu(v) \succeq_S \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v))$ and $\mu(w) \succeq_S \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(w))$. Hence,

$$\begin{aligned}\mathrm{lca}_S(\mu(v), \mu(w)) &\succeq_S \mathrm{lca}_S(\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v)), \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(w)))\\ &= \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v) \cup \sigma_{T_{\overline{\mathcal{E}}}}(w)).\end{aligned}$$

Since $T(u) \neq \triangle$ it follows that neither $(u, v) \in \mathcal{E}$ nor $(u, w) \in \mathcal{E}$ and hence, both edges are contained in $T_{\overline{\mathcal{E}}}$. By the same argumentation as in Item 3 it follows that $\sigma_{T_{\overline{\mathcal{E}}}}(u) = \sigma_{T_{\overline{\mathcal{E}}}}(v) \cup \sigma_{T_{\overline{\mathcal{E}}}}(w)$ and therefore, $\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v) \cup \sigma_{T_{\overline{\mathcal{E}}}}(w)) = \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))$. Hence, $\mu(u) \succ_S \mathrm{lca}_S(\mu(v), \mu(w)) \succeq_S \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))$. Now, **(M2i)** implies $t(u) \neq \bullet$. Since each interior vertex is labeled with one event, we have $t(u) = \square$. □

Lemma 5 *Let μ be a reconciliation map for the gene tree $(T; t, \sigma)$ and the species tree S as in Definition 2. Moreover, assume that T and S are binary. Set for all $u \in V(T)$:*

$$\gamma(u) = \begin{cases} \mu(u), & \text{if } \mu(u) \in V(S) \\ y, & \text{if } \mu(u) = (x, y) \in E(S)\end{cases}$$

Then $\gamma : V(T) \to V(S)$ is a map according to the DTL-scenario.

Proof We first emphasize that, by construction, $\mu(u) \succeq_S \gamma(u)$ for all $u \in V(T)$. Moreover, $\mu(u) = \mu(v)$ implies that $\gamma(u) = \gamma(v)$, and $\gamma(u) = \gamma(v)$ implies that $\mu(u)$ and $\mu(v)$ are comparable. Furthermore, $\mu(u) \prec_S \mu(v)$ implies $\gamma(u) \preceq_S \gamma(v)$, while $\gamma(u) \prec_S \gamma(v)$ implies that $\mu(u) \prec_S \mu(v)$. Thus, $\mu(u)$ and $\mu(v)$ are comparable if and only if $\gamma(u)$ and $\gamma(v)$ are comparable.

Item (I) and **(M1)** are equivalent.

For Item (II) let $u \in V(T) \setminus \mathbb{G}$ be an interior vertex with children v, w. If $(u, w) \notin \mathcal{E}$, then $w \prec_{T_{\overline{\mathcal{E}}}} u$. Applying Condition **(M3)** yields $\mu(w) \preceq_S \mu(u)$ and thus, by construction, $\gamma(w) \preceq_S \gamma(u)$. Therefore, $\gamma(u)$ is not a proper descendant of $\gamma(w)$ and $\gamma(w)$ is a descendant of $\gamma(u)$. If one of the edges, say (u, v), is a transfer edge, then $t(u) = \triangle$ and by Condition **(M2iii)** $\mu(u)$ and $\mu(v)$ are incomparable. Hence, $\gamma(u)$ and $\gamma(v)$ are incomparable. Therefore, $\gamma(u)$ is no proper descendant of $\gamma(v)$. Note that **(O2)** implies that for each vertex $u \in V(T) \setminus \mathbb{G}$ at least one of its outgoing edges must be a non-transfer edge, which implies that $\gamma(w) \preceq_S \gamma(u)$ or $\gamma(v) \preceq_S \gamma(u)$ as shown before. Hence, Item (IIa) and (IIb) are satisfied.

For Item (III) assume first that $(u, v) \in \mathcal{E}$ and therefore $t(u) = \triangle$. Then, **(M2iii)** implies that $\mu(u)$ and $\mu(v)$ are incomparable and thus, $\gamma(u)$ and $\gamma(v)$ are incomparable. Now assume that (u, v) is an edge in the gene tree T and $\gamma(u)$ and $\gamma(v)$ are incomparable. Therefore, $\mu(u)$ and $\mu(v)$ are incomparable. Now, apply Lemma 4(2).

Item (IVa) is clear by the event-labeling t of T and since **(O2)**. Now assume for (IVb) that $t(u) = \bullet$. Lemma 4(3) implies that $\mu(v)$ and $\mu(w)$ are incomparable and thus, $\gamma(v)$ and $\gamma(w)$ must be incomparable as well. Furthermore, Condition **(M2i)** implies that $\mu(u) = \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))$. Lemma 2 implies that $\mu(v) \succeq_S \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v))$ and $\mu(w) \succeq_S \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(w))$. The latter together with the incomparability of $\mu(v)$ and $\mu(u)$ implies that

$$\begin{aligned}\mathrm{lca}_S(\mu(v), \mu(w)) &= \mathrm{lca}_S(\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v)), \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(w)))\\ &= \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(v) \cup \sigma_{T_{\overline{\mathcal{E}}}}(w))\\ &= \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u)) = \mu(u).\end{aligned}$$

If $\mu(v)$ is mapped on the edge (x, y) in T, then $\gamma(v) = y$. By definition of lca for edges, $\mathrm{lca}_S(\mu(v), \gamma(w)) = \mathrm{lca}_S(y, \gamma(w)) = \mathrm{lca}_S(\gamma(v), \gamma(w))$. The same argument applies if $\mu(w)$ is mapped on an edge. Since for all $z \in V(T)$ either $\mu(z) \succ_S \gamma(z)$ (if $\mu(z)$ is mapped on an edge) or $\mu(z) = \gamma(z)$, we always have

$$lca_S(\gamma(v), \gamma(w)) = \mathrm{lca}_S(\mu(v), \mu(w)) = \mu(u).$$

Since $t(u) = \bullet$, **(M2i)** implies that $\mu(u) \in V(S)$ and therefore, by construction of γ it holds that $\mu(u) = \gamma(u)$. Thus, $\gamma(u) = \mathrm{lca}_S(\gamma(v), \gamma(w))$. For (IVc) assume that $t(u) = \square$. Condition **(M3)** implies that $\mu(u) \succeq_S \mu(v), \mu(w)$ and therefore, $\gamma(u) \succeq_S \gamma(v), \gamma(w)$. If $\gamma(v)$ and $\gamma(w)$ are incomparable, then $\gamma(u) \succeq_S \gamma(v), \gamma(w)$ implies that $\gamma(u) \succeq_S \mathrm{lca}_S(\gamma(v), \gamma(w))$. If $\gamma(v)$ and $\gamma(w)$ are comparable, say $\gamma(v) \succeq_S \gamma(w)$, then $\gamma(u) \succeq_S \gamma(v) = \mathrm{lca}_S(\gamma(v), \gamma(w))$. Hence, Statement (IVc) is satisfied. □

Lemma 6 *Let $\gamma : V(T) \to V(S)$ be a map according to the DTL-scenario for the binary the gene tree $(T; t, \sigma)$ and the binary species tree S. For all $u \in V(T)$ define:*

$$\mu(u) = \begin{cases} \gamma(u), & \text{if } t(u) \in \{\bullet, \odot\} \\ (x, \gamma(u)) \in E(S), & \text{if } t(u) \in \{\triangle, \square\}\end{cases}$$

Then $\mu : V(T) \to V(S) \cup E(S)$ is a reconciliation map according to Definition 2.

Proof Let $\gamma : V(T) \to V(S)$ be a map a DTL-scenario for the binary the gene tree $(T; t, \sigma)$ and the species tree S.

Condition **(M1)** is equivalent to (I).

For **(M3)** assume that $v \preceq_{T_{\overline{\mathcal{E}}}} w$. The path P from v to w in $T_{\overline{\mathcal{E}}}$ does not contain transfer edges. Thus, by (III) all vertices along P are comparable. Moreover, by (IIa) we have that $\gamma(w)$ is not a proper descendant of the image of its child in S, and therefore, by repeating these arguments along the vertices x in P_{wv}, we obtain $\gamma(v) \preceq_S \gamma(x) \preceq_S \gamma(w)$.

If $\gamma(v) \prec_S \gamma(w)$, then by construction of μ, it follows that $\mu(v) \prec_S \mu(w)$. Thus, **(M3)** is satisfied, whenever $\gamma(v) \prec_S \gamma(w)$. Assume now that $\gamma(v) = \gamma(w)$. If $t(v), t(w) \in \{\square, \triangle\}$ then $\mu(v) = (x, \gamma(v)) = (x, \gamma(w)) = \mu(w)$ and thus **(M3i)** is satisfied. If $t(v) = \bullet$ and $t(w) \neq \bullet$ then since $\mu(v) = \gamma(v)$ and $\mu(w) = (x, \gamma(w))$. Thus $\mu(v) \prec_S \mu(w)$.

Now assume that $\gamma(v) = \gamma(w)$ and w is a speciation vertex. Since $t(w) = \bullet$, for its two children w' and w'' the images $\gamma(w')$ and $\gamma(w'')$ must be incomparable due to (IVb). W.l.o.g. assume that w' is a vertex of P_{wv}. Since $\gamma(v) \preceq_S \gamma(x) \preceq_S \gamma(w)$ for any vertex x along P_{wv} and $\gamma(v) = \gamma(w)$, we obtain $\gamma(w') = \gamma(w)$. However, since $\gamma(w'') \preceq_S \gamma(w)$, the vertices $\gamma(w')$ and $\gamma(w'')$ are comparable in S; contradicting (IVb). Thus, whenever w is a speciation vertex, $\gamma(w') = \gamma(w)$ is not possible. Therefore, $\gamma(v) \preceq_S \gamma(w') \prec_S \gamma(w)$ and, by construction of μ, $\mu(v) \prec_S \mu(w)$. Thus, **(M3ii)** is satisfied.

Finally, we show that **(M2)** is satisfied. To this end, observe first that **(M2ii)** is fulfilled by construction of μ and **(M2iii)** is an immediate consequence of (III). Thus, it remains to show that **(M2i)** is satisfied. Thus, for a given speciation vertex u we need to show that $\mu(u) = \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))$. By construction, $\mu(u) = \gamma(u)$. Note, $T_{\overline{\mathcal{E}}}$ does not contain transfer edges. Applying (III) implies that for all edges (x, y) in $T_{\overline{\mathcal{E}}}$ the images $\gamma(x)$ and $\gamma(y)$ must be comparable. The latter and (IIa) implies that for all edges (x, y) in $T_{\overline{\mathcal{E}}}$ we have $\gamma(y) \preceq_S \gamma(x)$. Take the latter together, $\sigma(z) = \gamma(z) \preceq_S \gamma(u)$ for any leaf $z \in L_{T_{\overline{\mathcal{E}}}}(u)$. Therefore $\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u)) \preceq_S \gamma(u) = \mu(u)$. Assume for contradiction that $\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u)) \prec_S \gamma(u) = \mu(u)$. Consider the two children u' and u'' of u in $T_{\overline{\mathcal{E}}}$. Since neither $(u, u') \in \mathcal{E}$ nor $(u, u'') \in \mathcal{E}$ and T is a binary tree, it follows that $L_{T_{\overline{\mathcal{E}}}}(u) = L_{T_{\overline{\mathcal{E}}}}(u') \cup L_{T_{\overline{\mathcal{E}}}}(u'')$ and we obtain that $\sigma_{T_{\overline{\mathcal{E}}}}(u) = \sigma_{T_{\overline{\mathcal{E}}}}(u') \cup \sigma_{T_{\overline{\mathcal{E}}}}(u'')$. Moreover, re-using the arguments above, $\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u')) \preceq_S \gamma(u')$ and $\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u'')) \preceq_S \gamma(u'')$. By the arguments we used in the proof for **(M3)**, we have $\gamma(u') \prec_S \gamma(u)$ and $\gamma(u'') \prec_S \gamma(u)$. In particular, $\gamma(u')$ and $\gamma(u'')$ must be contained in the subtree of S that is rooted in the child a of $\gamma(u)$ in S with $\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u)) \preceq_S a$, as otherwise, $\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u')) \not\preceq_S \gamma(u')$ or $\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u'')) \not\preceq_S \gamma(u'')$. Moreover, neither $\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u)) \preceq_S \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u'))$ nor $\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u)) \preceq_S \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u''))$ is possible since then $\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u')) \preceq_S \gamma(u')$ and $\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u'')) \preceq_S \gamma(u'')$ implies that $\gamma(u')$ and $\gamma(u'')$ would be comparable; contradicting (IVb). Hence, there remains only one way to locate $\gamma(u')$ and $\gamma(u'')$, that is, they must be located in the subtree of S that is rooted in $\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u))$. But then we have $\mathrm{lca}_S(\gamma(u'), \gamma(u'')) \preceq_S \mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u)) \prec_S \gamma(u)$; a contradiction to (IVb) $\gamma(u) = \mathrm{lca}_S(\gamma(u'), \gamma(u''))$. Therefore, $\mathrm{lca}_S(\sigma_{T_{\overline{\mathcal{E}}}}(u)) = \gamma(u) = \mu(u)$ and **(M2i)** is satisfied. □

Finally, Lemmas 5 and 6 imply Theorem 1.

Authors' contributions

NN and MH designed the study. NN implemented and designed thealgorithms. All authors collaborated in research and the writing of themanuscript. All authors read and approved the final manuscript.

Author details

[1] Institute of Mathematics and Computer Science, University of Greifswald, Walther-Rathenau-Strasse 47, 17487 Greifswald, Germany. [2] Department of Mathematics and Computer Science, University of Southern Denmark, Campusvej 55, 5230 Odense M, Denmark. [3] Parallel Computing and Complex Systems Group, Department of Computer Science, Leipzig University, Augustusplatz 10, 04109 Leipzig, Germany. [4] Center for Bioinformatics, Saarland University, Building E 2.1, P.O. Box 151150, 66041 Saarbrücken, Germany. [5] Bioinformatics Group, Department of Computer Science, University of Leipzig, Häartelstraße 16-18, 04107 Leipzig, Germany. [6] Interdisciplinary Center for Bioinformatics, Universität Leipzig, Häartelstraße 16-18, 04107 Leipzig, Germany. [7] Max-Planck-Institute for Mathematics in the Sciences, Inselstraße 22, 04103 Leipzig, Germany. [8] Fraunhofer Institut for Cell Therapy and Immunology, Perlickstraße 1, 04103 Leipzig, Germany. [9] Inst. f. Theoretical Chemistry, University of Vienna, Wäahringerstraße 17, 1090 Wien, Austria. [10] Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe, NM 87501, USA. [11] Center for non-coding RNA in Technology and Health, Grønegåardsvej 3, 1870 Frederiksberg C, Denmark.

Acknowledgements

We thank the organizers of the 32nd TBI Winterseminar 2017 in Bled(Slovenia), where the authors met and jointly drafted the main ideas of thispaper with the help of an unknown number of cold and tasty cans of redUnion, or was it green Laško?

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

Not applicable.

Consent for publication

Not applicable.

Ethics approval and consent to participate

Not applicable.

Funding

Supported in part by the Danish Council for Independent Research, NaturalSciences, Grants DFF-1323-00247 and DFF-7014-00041.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 21 October 2017 Accepted: 20 January 2018
Published online: 06 February 2018

References

1. Dress A, Moulton V, Steel M, Wu T. Species, clusters and the 'tree of life': a graph-theoretic perspective. J Theor Biol. 2010;265:535–42.
2. Fitch WM. Homology: a personal view on some of the problems. Trends Genet. 2000;16:227–31.
3. Hellmuth M, Stadler PF, Wieseke N. The mathematics of xenology: dicographs, symbolic ultrametrics, 2-structures and tree- representable systems of binary relations. J Math Biol. 2016;75(1):199–237. https://doi.org/10.1007/s00285-016-1084-3.
4. Hellmuth M, Wieseke N. From sequence data including orthologs, paralogs, and xenologs to gene and species trees. In: Pontarotti P, editor. Evolutionary Biology: convergent evolution, evolution of complex traits, concepts and methods. Cham: Springer; 2016. p. 373–92.
5. Guigó R, Muchnik I, Smith T. Reconstruction of ancient molecular phylogeny. Mol Phylogenet Evol. 1996;6:189–213.
6. Page RDM, Charleston MA. Trees within trees: phylogeny and historical associations. Trends Ecol Evol. 1998;13:356–9.
7. Zmasek C, Eddy S. A simple algorithm to infer gene duplication and speciation events on a gene tree. Bioinformatics. 2001;17:821–8.
8. Vernot B, Stolzer M, Goldman A, Durand D. Reconciliation with non-binary species trees. J Comput Biol. 2008;15:981–1006. https://doi.org/10.1089/cmb.2008.0092.
9. Hellmuth M, Wieseke N, Lechner M, Lenhof H-P, Middendorf M, Stadler PF. Phylogenomics with paralogs. Proc Natl Acad Sci. 2015;112(7):2058–63. https://doi.org/10.1073/pnas.1412770112.
10. Roth ACJ, Gonnet GH, Dessimoz C. Algorithm of OMA for large-scale orthology inference. BMC Bioinf. 2008;9:518.
11. Altenhoff AM, Dessimoz C. Phylogenetic and functional assessment of orthologs inference projects and methods. PLoS Comput Biol. 2009;5:1000262.
12. Lechner M, Hernandez-Rosales M, Doerr D, Wieseke N, Thévenin A, Stoye J, Hartmann RK, Prohaska SJ, Stadler PF. Orthology detection combining clustering and synteny for very large datasets. PLoS ONE. 2014;9(8):105015.
13. Altenhoff AM, Boeckmann B, Capella-Gutierrez S, Dalquen DA, DeLuca T, Forslund K, Huerta-Cepas J, Linard B, Pereira C, Pryszcz LP, Schreiber F, da Silva AS, Szklarczyk D, Train CM, Bork P, Lecompte O, von Mering C, Xenarios I, Sjölander K, Jensen LJ, Martin MJ, Muffato M, Gabaldón T, Lewis SE, Thomas PD, Sonnhammer E, Dessimoz C. Standardized benchmarking in the quest for orthologs. Nat Methods. 2016;13:425–30.
14. Hellmuth M, Hernandez-Rosales M, Huber KT, Moulton V, Stadler PF, Wieseke N. Orthology relations, symbolic ultrametrics, and cographs. J Math Biol. 2013;66(1–2):399–420.
15. Hellmuth M. Biologically feasible gene trees, reconciliation maps and informative triples. Algorithms Mol Biol. 2017;12(1):23.
16. Hernandez-Rosales M, Hellmuth M, Wieseke N, Huber KT, Moulton V, Stadler PF. From event-labeled gene trees to species trees. BMC Bioinf. 2012;13(Suppl 19):6.
17. Doyon J-P, Ranwez V, Daubin V, Berry V. Models, algorithms and programs for phylogeny reconciliation. Brief Bioinf. 2011;12(5):392.
18. Merkle D, Middendorf M. Reconstruction of the cophylogenetic history of related phylogenetic trees with divergence timing information. Theor Biosci. 2005;4:277–99.
19. Charleston MA. Jungles: a new solution to the host/parasite phylogeny reconciliation problem. Math Biosci. 1998;149(2):191–223.
20. Tofigh A, Hallett M, Lagergren J. Simultaneous identification of duplications and lateral gene transfers. IEEE/ACM Trans Comput Biol Bioinf. 2011;8(2):517–35.
21. Böcker S, Dress AWM. Recovering symbolically dated, rooted trees from symbolic ultrametrics. Adv Math. 1998;138:105–25.
22. Hellmuth M, Wieseke N. On symbolic ultrametrics, cotree representations, and cograph edge decompositions and partitions., Proceedings COCOON 2015Cham: Springer; 2015. p. 609–23.
23. Hellmuth M, Wieseke N. On tree representations of relations and graphs: Symbolic ultrametrics and cograph edge decompositions. J Comb Optim. 2017; https://doi.org/10.1007/s10878-017-0111-7.
24. Bansal MS, Alm EJ, Kellis M. Efficient algorithms for the reconciliation problem with gene duplication, horizontal transfer and loss. Bioinformatics. 2012;28(12):283–91.
25. Kahn AB. Topological sorting of large networks. Commun ACM. 1962;5(11):558–62.
26. Altenhoff AM, Gil M, Gonnet GH, Dessimoz C. Inferring hierarchical orthologous groups from orthologous gene pairs. PLoS ONE. 2013;8(1):53786.
27. Altenhoff AM, et al. The OMA orthology database in 2015: function predictions, better plant support, synteny view and other improvements. Nucleic Acids Res. 2015;43(D1):240–9.
28. Chen F, Mackey AJ, Stoeckert CJ, Roos DS. OrthoMCL-db: querying a comprehensive multi-species collection of ortholog groups. Nucleic Acids Res. 2006;34(S1):363–8.
29. Lechner M, Findeiß S, Steiner L, Marz M, Stadler PF, Prohaska SJ. `Proteinortho`: detection of (co-)orthologs in large-scale analysis. BMC Bioinf. 2011;12:124.
30. Östlund G, Schmitt T, Forslund K, Köstler T, Messina DN, Roopra S, Frings O, Sonnhammer ELL. `InParanoid 7`: new algorithms and tools for eukaryotic orthology analysis. Nucleic Acids Res. 2010;38(suppl 1):196–203.
31. Tatusov RL, Galperin MY, Natale DA, Koonin EV. The COG database: a tool for genome-scale analysis of protein functions and evolution. Nucleic Acids Res. 2000;28(1):33–6.
32. Trachana K, Larsson TA, Powell S, Chen W-H, Doerks T, Muller J, Bork P. Orthology prediction methods: a quality assessment using curated protein families. BioEssays. 2011;33(10):769–80.
33. Wheeler DL, Barrett T, Benson DA, Bryant SH, Canese K, Chetvernin V, Church DM, Dicuccio M, Edgar R, Federhen S, Feolo M, Geer LY, Helmberg W, Kapustin Y, Khovayko O, Landsman D, Lipman DJ, Madden TL, Maglott DR, Miller V, Ostell J, Pruitt KD, Schuler GD, Shumway M, Sequeira E, Sherry ST, Sirotkin K, Souvorov A, Starchenko G, Tatusov RL, Tatusova TA, Wagner L, Yaschenko E. Database resources of the national center for biotechnology information. Nucleic Acids Res. 2008;36:13–21.
34. Clarke GDP, Beiko RG, Ragan MA, Charlebois RL. Inferring genome trees by using a filter to eliminate phylogenetically discordant sequences and a distance matrix based on mean normalized BLASTP scores. J Bacteriol. 2002;184(8):2072–80.
35. Dessimoz C, Margadant D, Gonnet GH. DLIGHT—lateral gene transfer detection using pairwise evolutionary distances in a statistical framework. In: Proceedings RECOMB 2008, pp. 315–330. Springer, Berlin; 2008.
36. Lawrence JG, Hartl DL. Inference of horizontal genetic transfer from molecular data: an approach using the bootstrap. Genetics. 1992;131(3):753–60.
37. Pellegrini M, Marcotte EM, Thompson MJ, Eisenberg D, Yeates TO. Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. Proc Natl Acad Sci USA. 1999;96(8):4285–8.
38. Ravenhall M, Škunca N, Lassalle F, Dessimoz C. Inferring horizontal gene transfer. PLoS Comput Biol. 2015;11(5):1004095.
39. Dondi R, Lafond M, El-Mabrouk N. Approximating the correction of weighted and unweighted orthology and paralogy relations. Algorithms Mol Biol. 2017;12(1):4.
40. Lafond M, El-Mabrouk N. Orthology and paralogy constraints: satisfiability and consistency. BMC Genom. 2014;15(6):12.
41. Lafond M, El-Mabrouk N. Orthology relation and gene tree correction: complexity results. In: International workshop on algorithms in bioinformatics, Berlin: Springer; 2015. p. 66–79.
42. Dondi R, El-Mabrouk N, Lafond M. Correction of weighted orthology and paralogy relations-complexity and algorithmic results. In: International workshop on algorithms in bioinformatics, Berlin: Springer; 2016. p. 121–36.
43. Dondi R, Mauri G, Zoppis I. Orthology correction for gene tree reconstruction: Theoretical and experimental results. Procedia Computer Science. International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland. p. 1115–24.
44. Lafond M, Dondi R, El-Mabrouk N. The link between orthology relations and gene trees: a correction perspective. Algorithms Mol Biol. 2016;11(1):1.