Simultaneous Identification of Duplications and Lateral Gene Transfers

Ali Tofigh, Michael Hallett, and Jens Lagergren

Abstract

The incongruency between a gene tree and a corresponding species tree can be attributed to evolutionary events such as gene duplication and gene loss. This paper describes a combinatorial model where a so-called DTL-scenario is used to explain the differences between a gene tree and a corresponding species tree taking into account gene duplications, gene losses, and lateral gene transfers (also known as horizontal gene transfers). The reasonable biological constraint that a lateral gene transfer may only occur between contemporary species leads to the notion of acyclic DTL-scenarios. Parsimony methods are introduced by defining appropriate optimization problems. We show that finding most parsimonious acyclic DTL-scenarios is NP-complete. However, by dropping the condition of acyclicity, the problem becomes tractable, and we provide a dynamic programming algorithm as well as a fixed-parameter-tractable algorithm for finding most parsimonious DTL-scenarios.

Index Terms

Trees, Biology and genetics, Combinatorial algorithms, Graph algorithms.

Simultaneous Identification of Duplications and Lateral Gene Transfers

I. Introduction

Gene duplication and lateral gene transfer (LGT) are important evolutionary events that, in interplay with each other as well as with other evolutionary events, shape the genomes of species and thereby also their phenotypes. The role of duplications in creating new functionality has been studied rather extensively. In [1] and [2], the possible fates of a duplicated gene was described by a biological model where the archetypical fates are coined non-functionalization (loss of function due to a disruptive mutation), sub-functionalization (in which the two copies each take on a subset of the original function), and neo-functionalization (where one of the copies, due to point mutations, assumes a new function). Duplications are common in many parts of the tree of life [3], [4], [5], [6], [7]. LGT has also been implicated in how species acquire new functionality and phenotypes. Particular attention has been given to how pathogens have developed through LGT by receiving so called pathogenicity islands, and to the relative importance of LGT and gene loss in pathogen evolution [8]. In contrast to an apparent consensus on the importance of gene duplications, the importance and prevalence of LGT is well known to be unusually controversial (see [9] and references therein).

As always when developing methods for evolutionary studies, there is an interdependence between the underlying models and our knowledge or opinions of the evolutionary processes. When considering LGT, three main views can be identified together with their ramifications for method development. First, one extreme view with few remaining proponents, is that LGT hardly exists, so discrepancies between gene and species trees are due to random effects or insufficiently sophisticated tree reconstruction methods, or possibly due to other events such as duplications. At the other extreme is the view that, at least in some parts of the tree of life, LGT is so rampant that trees are in general not a valid representation of organismal evolution. This latter view is, of course, in conflict with any form of reliance on a species tree when constructing a network or some alternative representation of reticulate evolution caused by LGT. The latter view is, however, fully consistent with the use of gene trees. Finally, there is an intermediate view according to which LGT is common, although not so common among the genes of a species that organismal evolution cannot be meaningfully

described by a tree. When accepting this intermediate view, it becomes desirable to reconstruct species trees, as well as the locations where LGT has occurred, for specific gene families. The parsimony variation of this problem was formalized and treated in [10]. More recently, this approach was applied in an *ad hoc* manner by Bapteste *et al.* [11]. There are also several earlier studies of heuristics for the problem, e.g., [12] and [13], as well as later studies using distance methods [14]. Subsequently, evidence has been provided for the correctness of the intermediate view when considering γ-proteobacteria [15]. So, a species tree can aid in the identification of LGT and, moreover, species trees requiring fewer LGTs to explain the laterally transferred gene can be viewed as more likely than others.

We may, in general, divide methods attempting to reconstruct evolution into explicit, where a direct interpretation in terms of evolutionary events is possible, and implicit, where no such interpretation is available [16]. We may further distinguish between methods for LGT identification and methods for deriving descriptions of reticulate evolution. Two main methodologies have been applied in order to develop methods for identifying laterally transferred genes. First, atypical sequence characteristics of newly transferred genes have been taken advantage of in order to identify LGTs, see e.g., [17]. Second, incongruencies between gene and species trees have been used in so called phylogenetic LGT identification methods. These methods basically consist of comparing a gene tree with an established species tree and identifying gene tree clades with a significantly different placement in the gene tree compared to the corresponding clade in the species tree. Naive phylogenetic methods are still commonly used.

There has been considerable consensus on the importance and prevalence of duplications, for which the biological model has been significantly clearer. Already in 1979, Goodman *et al.* [18] gave a parsimony method for identification of gene duplications as well as for embedding a gene tree into a corresponding species tree in a way that illustrates a possible evolution of the former *inside* the latter. Building on this framework, Guigo *et al.* [19] gave an explicit supertree method attempting to find the species tree that explains a number of given gene trees using a minimum number of duplications. Ma *et al.* [20] proved hardness results for several variations of species tree reconstruction problems. Hallett *et al.* [21] gave an efficient algorithm that is guaranteed to find the optimum species tree under the assumption that one of the gene trees have had a constant number of lineages in each species tree lineage. More recently, methods have emerged for duplication analysis, i.e., identification of duplications and simultaneous construction of a gene tree, as well as embedding the gene tree inside the species tree. Duplication analysis takes advantage of sequence information directly rather

than merely mediated by a gene tree. In [22], an *ad hoc* method for duplication analysis was presented which also takes gene order information into account. In [23], [24], [25], an integrated model for gene duplication, gene loss, and sequence evolution, together with computational tools for duplication analysis based on the same model, has been developed. In the latest contribution to that line of research, the model was extended with an iid model of sequence evolution rate variation across gene tree edges [26].

There appears to be relatively few studies attempting to tease apart the influences of LGT and gene duplications in regions of the tree of life, where LGT are believed to be common. In [27], it was estimated that 16% of the 1425 intra-genome homologs of *E. coli* K12 have been acquired by LGT, implying that the other homologs have been acquired through gene duplication. An analysis of paralog content in 106 bacterial genomes can be found in [28]. A study by Retchless and Lawrence [29] concluded that the complete separation of *E. coli* and *S. enterica* from their common ancestor took tens of millions of years and that gene conversion events due to bacterial recombination occurred between the incipient species during a period of ~70 million years.

Few attempts have been made at devising methods to explicitly detect duplications and LGTs simultaneously. Csűrös *et al.* [30] gave a probabilistic model of gene evolution that considered LGTs, duplications, and losses, but applied it only to gene family sizes. A rather restricted parsimony method was given in [31] where the input is a gene tree and a species tree augmented with additional edges showing where transfers have taken place. The output is then the minimum cost of mapping the gene tree into the augmented species tree/network.

Here, we present a parsimony method that given a species tree and an incongruent gene tree finds reconciliations that explain the incongruences with a minimum number of duplications and lateral gene transfers. A preliminary version, of which this paper is a complete and thorough revision, first appeared in [32]. In section III, we give the definition of a DTL-scenario (Duplication-Transfer-Loss scenario) which is our formal equivalent of a reconciliation: a description of how a gene tree has evolved within a species tree using, in our case, duplications, LGTs, and losses. Care has been taken in defining DTL-scenarios to include all the interesting viable cases of gene evolution, and at the same time to exclude the cases that seem inappropriate or degenerate in a parsimony setting. Our aim is thus to find DTL-scenarios with a minimum number of duplications and LGTs. As we will demonstrate, the (implicitly inferred) number of losses can be used to choose between several existing most parsimonious DTL-scenarios.

Biologically, LGTs only occur between a pair of contemporary species. It may therefore be desirable to enforce this restriction and demand the existence of a temporal order on the species tree vertices such that all LGTs in the reconciliation occur between contemporary species. We term such DTL-scenarios acyclic. In section V, we show that the problem of finding most parsimonious acyclic DTL-scenarios is NP-complete. However, as was shown in [33], cycles are not a major concern in practice, and in sections VI and VII we provide efficient algorithms for finding most parsimonious DTL-scenarios disregarding the notion of cyclicity. More specifically, in section VI, we provide a dynamic programming algorithm for computing the minimum cost of any DTL-scenario reconciling a gene tree and a species tree, and in section VII, we describe a fixed-parameter-tractable algorithm for enumerating all most parsimonious DTL-scenarios. Finally, in section IX, we demonstrate the benefits of our methods by applying them on real biological data.

But first, we start with a description of the notation that we will use in the remainder of this article.

II. Definitions

For a directed graph H, we let $V(H)$ and $A(H)$ be the sets of vertices and arcs of H, respectively. For a tree T, we let $V(T)$, $\mathring{V}(T)$, $L(T)$, and $E(T)$ denote the sets of vertices, internal vertices, leaves, and edges of T, respectively. For a rooted tree T, $\mathrm{root}(T)$ denotes the root vertex. We consider edges of rooted trees to be directed away from the root. An edge of a rooted tree is denoted by an ordered pair of vertices (u_1, u_2) where u_1 is closer to the root than u_2.

Let T be a rooted tree. If (u, v) is an edge of T, then v is called a child of u, and u is called the parent of v denoted by $p_T(v)$. When the tree is clear from context, we will drop the subscript and write $p(v)$. Two distinct vertices u and v are siblings iff $p_T(u) = p_T(v)$; in that case, the two edges $(p_T(u), u)$ and $(p_T(v), v)$ are called sibling edges. A vertex v is a descendant of a vertex u, denoted by $v \leq_T u$, iff there is a directed path from u to v. In that case, we also say that u is an ancestor of v ($u \geq_T v$). We say that v is a proper descendant (proper ancestor) of u iff $v \leq_T u$ ($v \geq_T u$) and $v \neq u$ and denote this by $v <_T u$ ($v >_T u$). An edge whose vertices are ancestors of v is called an ancestral edge of v. Two vertices u and v are incomparable iff $u \not\leq_T v$ and $v \not\leq_T u$. An edge (u, v) is called the incoming edge of v and an outgoing edge of u. The least common ancestor of a set X of vertices of T, denoted $\mathrm{lca}\, X$, is the $\leq_T$-minimal vertex of T that is an ancestor of every vertex in X. By

T_u we denote the subtree of T rooted at $u \in V(T)$.

A binary rooted tree is a rooted tree in which every vertex has at most two children. A full binary tree is a rooted tree in which every vertex has zero or two children. A (full rooted binary) forest is a graph in which every connected component is a (full rooted binary) tree. If T is a tree and $F \subseteq E(T)$ is a set of edges of T, then $T \setminus F$ is the forest obtained from T by removing the edges in F, i.e., $E(T \setminus F) = E(T) \setminus F$.

It will be convenient to describe rooted trees using the Newick format. In this notation, a tree is described using parentheses. If $T_1, \ldots, T_n$ are rooted trees, then $(T_1, \ldots, T_n)$ is the rooted tree obtained from $T_1, \ldots, T_n$ by adding a new root ρ and the edges $(\rho, \mathrm{root}(T_i))$, for $i = 1, \ldots, n$.

Finally, if $f : X \to Y$ is a function from X into Y, and if $R \subseteq X$, then the restriction of f to R is denoted by $f|_R$.

III. DTL-Scenarios

A reconciliation may be thought of as a mapping of a gene tree into a corresponding species tree demonstrating a biologically viable history of the evolution of genes within species. If duplications and losses are the sole culprits in creating incongruencies between the species tree and the gene tree, then there is a unique reconciliation that minimizes both the number of duplications and losses [18] (see also [19], [34], [35]). However, adding LGTs as a possible evolutionary event complicates the notion of a reconciliation; without LGTs, the evolution of the gene tree is conveniently restricted to staying within the edges of the species tree, something that is not true when dealing with LGTs. In fact, when also considering LGTs, there is no simple reconciliation that minimizes the number of evolutionary events. Our approach instead is to define exactly what constitutes a valid reconciliation and devise algorithms that find the most parsimonious ones.

In this section, we define DTL-scenarios (Duplication-Transfer-Loss scenarios) which serve as the formalization of the notion of valid reconciliations. When doing so, we must be careful as to what mappings and combinations of events we wish to allow. We can benefit greatly by carefully defining a reconciliation so as not to allow cases that seem degenerate within a parsimony framework. One example of such a clearly degenerate case is a sequence of LGTs in which the same gene is transferred over and over, and where each transfer is followed by a loss in the species from which the transfer originated. Such a sequence would leave no trace in the intermediate species to which genes have been transferred and would in fact be

represented by a single edge in the gene tree. Such a reconciliation is certainly not interesting in a parsimony setting. Our definition of DTL-scenarios has been carefully crafted to allow biologically viable reconciliations while excluding clearly degenerate cases.

The purpose of a DTL-scenario is to

- assign to each gene tree vertex exactly one event: a speciation, a duplication, or a lateral transfer,
- determine for each transfer vertex exactly one of the outgoing edges as a transfer edge,
- map every vertex of the gene tree into the species tree in a way that is consistent with the previous points and with the temporal order implicitly represented by the trees.

Below, we will formally define a DTL-scenario as an octuple $(S, G, \sigma, \gamma, \Sigma, \Delta, \Theta, \Xi)$. Informally, S and G represent biological data in the form of a species tree and a corresponding gene tree. The correspondence between genes and species is established via a leaf mapping function σ. Every bifurcation of G is the result of one of three events: speciation, duplication, and lateral transfer; the sets Σ, Δ, and Θ contain internal vertices of G representing these events, respectively. The set Ξ contains the edges of G corresponding to lateral transfers. Finally, γ maps the entire gene tree into the species tree showing where the evolutionary events have taken place.

Note that γ will be defined as a function mapping the gene tree vertices to species tree vertices. For a gene tree vertex u, the interpretation of $\gamma(u)$ depends on the type of event represented by u in the DTL-scenario. If u represents a speciation, i.e., $u \in \Sigma$, $\gamma(u)$ is the species tree vertex at which the speciation took place. Otherwise, if u represents a duplication or a lateral transfer, the event represented by u is considered to have occurred somewhere along the incoming edge of $\gamma(u)$ (if $\gamma(u)$ is the root of the species tree, then the event is taken to have occurred before the root). Fig. 1 contains a complete example of a scenario.

Formally, we define a **DTL-scenario** as an octuple $(S, G, \sigma, \gamma, \Sigma, \Delta, \Theta, \Xi)$ where S and G are rooted full binary trees, $\sigma : L(G) \to L(S)$ maps every gene tree leaf to the species in which it is found, $\gamma : V(G) \to V(S)$ maps the gene tree into the species tree, Σ, Δ, and Θ form a partition of $\mathring{V}(G)$, and $\Xi \subset E(G)$ is a subset of the gene tree edges such that:

(I) For each leaf u in the gene tree, $\gamma(u) = \sigma(u)$

(II) If $u \in \mathring{V}(G)$ is a gene tree vertex with children v and w, then

a) $\gamma(u)$ is not a proper descendant of $\gamma(v)$ or $\gamma(w)$

b) At least one of $\gamma(v)$ and $\gamma(w)$ is a descendant of $\gamma(u)$

(III) $(u, v) \in \Xi$ if and only if $\gamma(u)$ is incomparable to $\gamma(v)$

(IV) If $u \in \mathring{V}(G)$ is a gene tree vertex with children v and w, then

a) $u \in \Theta$ if and only if $(u, v) \in \Xi$ or $(u, w) \in \Xi$

b) $u \in \Sigma$ only if $\gamma(u) = \mathrm{lca}\,\{\gamma(v), \gamma(w)\}$ and $\gamma(v)$ and $\gamma(w)$ are incomparable

c) $u \in \Delta$ only if $\gamma(u) \geq \mathrm{lca}\,\{\gamma(v), \gamma(w)\}$

The cost of a DTL-scenario α is denoted $|\alpha|$, and is defined as $|\Delta| + |\Theta|$ (which is equal to $|\Delta| + |\Xi|$). For convenience, we will allow ourselves to use σ as a function mapping a *set* of gene tree leaves to the corresponding *set* of species tree leaves. In this text, the words DTL-scenario and scenario will be used interchangeably.

Condition (I) states that γ is an extension of σ. Condition (IIa) ensures that genes evolve in the direction implied by the trees. Condition (IIb) restricts each bifurcation of G to represent exactly one evolutionary event. Condition (III) determines which edges of the gene tree are to be considered as lateral transfer edges, and condition (IV) states when a gene tree vertex may represent a lateral transfer, speciation, or duplication. Note the overlap between conditions (IVb) and (IVc): given a mapping γ, the set of gene tree vertices that may be labeled as speciations according to condition (IVb) is a subset of those that may be labeled as duplications according to (IVc). Of course, no most parsimonious DTL-scenario will label a gene tree vertex as a duplication if the vertex may just as well be labeled a speciation. But for now, we will allow this slight over-expressiveness of DTL-scenarios.

For convenience, we will adopt the following notational conventions throughout the paper. The symbols S, G, σ, γ, Σ, Δ, Θ, Ξ, and their subscripted versions, will be used exclusively as the elements of DTL-scenarios. If $\alpha_\Box$ is a DTL-scenario, where $\Box$ is some subscript, then the elements of $\alpha_\Box$ are $S_\Box$, $G_\Box$, $\sigma_\Box$, $\gamma_\Box$, $\Sigma_\Box$, $\Delta_\Box$, $\Theta_\Box$, and $\Xi_\Box$, respectively. If a symbol referring to a scenario lacks subscript, then so will its elements. In that case, it will always be clear from context to which scenario the element symbols belong. The expression "$\alpha_\Box$ is a scenario for S, G, and σ" is understood to mean that $S_\Box = S$, $G_\Box = G$, and $\sigma_\Box = \sigma$.

In our scenarios, the interpretation of a transfer edge $(u, v) \in \Xi$ is that a lateral transfer has occurred from the incoming edge of $\gamma(u)$ to some ancestral edge of $\gamma(v)$. For a scenario to be biologically meaningful, we must be able to order the species tree vertices in time in such a way that the incoming edge of $\gamma(u)$ overlaps some ancestral edge of $\gamma(v)$. In fact, if (u', v') is also a transfer edge and $v \geq_G v'$, then we must also ensure that the incoming edge of $\gamma(u)$ overlaps some ancestral edge of $\gamma(v')$. Extending this to include all transfer edges,

we will call a scenario **acyclic** iff

(V) There is a total order $<$ on $V(S)$ such that:

a) if $(x, y) \in E(S)$, then $x < y$

b) if $(u, v), (u', v') \in \Xi$ and $v \geq_G v'$, then $p(\gamma(u)) < \gamma(v')$

See Fig. 1 for an example of a cyclic scenario.

We now present a lemma showing when duplications are forced when considering a mapping of the gene tree into the species tree.

Lemma 1: Let α be a scenario for S, G, and σ, and let $u \in \mathring{V}(G)$ be a gene tree vertex with children v and w.

(a) If $\gamma(v)$ and $\gamma(w)$ are comparable, then $u \in \Delta$.

(b) If $\gamma(u) >_S \text{lca}\,\{\gamma(v), \gamma(w)\}$, then $u \in \Delta$.

Proof:

(a) Assume, without loss of generality, that $\gamma(v) \leq_S \gamma(w)$. By (IVb), $u \notin \Sigma$. From (II), we see that $\gamma(u)$ must be an ancestor of both $\gamma(v)$ and $\gamma(w)$. Therefore, by (III) and (IVa), $u \notin \Theta$. By the definition of a scenario, the sets Σ, Δ, and Ξ partition the internal vertices of G. Hence, having shown that $u \notin \Sigma$ and $u \notin \Theta$, we deduce that $u \in \Delta$.

(b) Assume that $\gamma(u) >_S \text{lca}\,\{\gamma(v), \gamma(w)\}$. Since $\gamma(u)$ is comparable to both $\gamma(v)$ and $\gamma(w)$, $u \notin \Theta$. Since $\gamma(u) \neq \text{lca}\,\{\gamma(v), \gamma(w)\}$, $u \notin \Sigma$. Hence, $u \in \Delta$.

■

The minimum number of losses inferred from a scenario can be computed by considering each non-transfer edge (u, v) of the gene tree and the mapping of its vertices into the species tree. This is similar to how losses are computed in the duplication-loss model. One loss is inferred for each intermediate species tree vertex between $\gamma(u)$ and $\gamma(v)$. A loss is also inferred when $u \in \Delta$ and $\gamma(v) \neq \gamma(u)$. We can make this argument formal as follows. Let α be a scenario for S, G, and σ and define $I_\alpha(e)$, where $e = (u, v) \in E(G)$, to be the number of intermediate species tree vertices between $\gamma(u)$ and $\gamma(v)$:

$$I_\alpha(e) = |\{x \in V(S) : \gamma(v) <_S x <_S \gamma(u)\}|.$$

Note that $I_\alpha(e) = 0$ when $e \in \Xi$. The number of losses inferred by e is

$$\text{loss}_\alpha(e) = \begin{cases} I_\alpha(e) + 1 & \text{if } u \in \Delta \text{ and } \gamma(u) \neq \gamma(v) \\ I_\alpha(e) & \text{otherwise.} \end{cases}$$

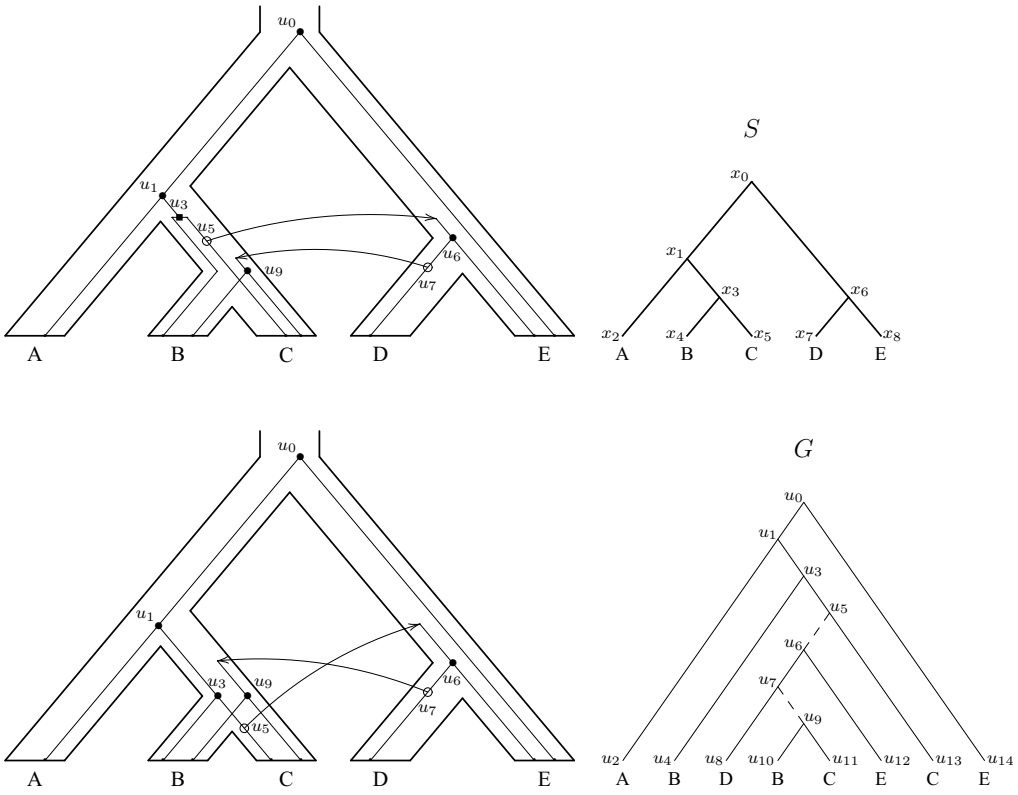


Fig. 1. **An example of a scenario.** A species tree S and a gene tree G are shown on the right side of the figure. The names of the extant species are given below the leaves of S. The extant genes of G are labeled with the name of the species to which they belong. The mapping σ is then derived from these labels:$\sigma(u_2) = x_2, \sigma(u_4) = x_4, \sigma(u_8) = x_7, \sigma(u_{10}) = x_4, \sigma(u_{11}) = x_5, \sigma(u_{12}) = x_8, \sigma(u_{13}) = x_5$, and $\sigma(u_{14}) = x_8$. A DTL-scenario for S, G, and σ is shown on the top left of the figure. The leaves of G are mapped by γ according to σ. For internal vertices of G, we have that $\gamma(u_0) = x_0$, $\gamma(u_1) = x_1$, $\gamma(u_3) = x_3$, $\gamma(u_5) = x_3$, $\gamma(u_6) = x_6$, $\gamma(u_7) = x_7$, and $\gamma(u_9) = x_3$. Two of the edges of the gene tree are transfer edges in this scenario: $\Xi = \{(u_5, u_6), (u_7, u_9)\}$. The sets of speciations, duplications, and transfer vertices are: $\Sigma = \{u_0, u_1, u_6, u_9\}, \Delta = \{u_3\}, \Theta = \{u_5, u_7\}$. It is easy to check that this scenario is acyclic. On the other hand, the scenario depicted in the lower left of the figure is cyclic; we cannot order the species tree vertices in time so that x_6 comes before x_3 and x_3 comes before x_6. Note that in this last scenario the least-common-ancestor mapping was used to map G into S, which is not the case in the scenario on the top left.

The total number of losses of the scenario is then

$$\text{loss}(\alpha) = \sum_{e \in E(G)} \text{loss}_\alpha(e).$$

IV. TRANSFER SETS

In this section we will examine the possible mappings of a gene tree into a corresponding species tree. We will characterize the subsets of $E(G)$ that can serve as a set of transfer edges in a DTL-scenario and show how this characterization leads to a complete understanding of all possible mappings. Closely linked to this characterization is a least-common-ancestor mapping that we will define shortly.

As stated earlier, there is a unique mapping of the gene tree into the species tree under the duplication-loss model that simultaneously minimizes the number of duplications and losses. This mapping is defined as

$$M(u) = \mathrm{lca}\,(\sigma(L(G_u))), \tag{1}$$

for all $u \in V(G)$. We now define a similar mapping that will depend, not only on G, S, and σ, but also on the set of gene tree edges that we have chosen as transfer edges. Intuitively, given a set $F \subset E(G)$ as the set of transfer edges, we first remove from G all the edges of F to obtain a forest of rooted trees. Each tree in the forest is then mapped into S using (1).

Formally, if $F \subset E(G)$ is a set of gene tree edges such that no two edges of F are siblings, we define the function $\lambda_{G\setminus F} : V(G) \to V(S)$, called the least-common-ancestor mapping of G into S, by

$$\lambda_{G\setminus F}(u) = \mathrm{lca}\,(\sigma(L_u)),$$

where L_u is the set of leaves of G_u reachable from u using only edges not in F, i.e., only using edges in $G \setminus F$. Note that if $F = \emptyset$, then $\lambda_{G\setminus F} = M$. The next lemma shows that $\lambda_{G\setminus F}$ can be computed recursively in postorder.

Lemma 2: Let S, G, and σ be given, and let $F \subset E(G)$ be a set of gene tree edges such that no two edges are siblings. Then, for $u \in V(G)$,

$$\lambda_{G\setminus F}(u) = \begin{cases} \sigma(u) & \text{if } u \in L(G), \\ \mathrm{lca}\,\{\lambda_{G\setminus F}(v), \lambda_{G\setminus F}(w)\} & \text{if } (u,v) \notin F \text{ and } (u,w) \notin F, \text{ where } v, w \text{ are the children of } u, \\ \lambda_{G\setminus F}(w) & \text{if } (u,v) \in F \text{ and } (u,w) \notin F, \text{ where } v, w \text{ are the children of } u, \end{cases}$$

Proof: For a vertex $u \in V(G)$, let L_u denote the set of leaves of G_u reachable from u using only edges in $G \setminus F$. The first case follows immediately from the definition of $\lambda_{G\setminus F}$. For the third case, just note that $(u,v) \in F$ implies that $L_u = L_w$. To verify the second case,

note that for vertex sets A and B, $\operatorname{lca} A \cup B = \operatorname{lca}\{\operatorname{lca} A, \operatorname{lca} B\}$, so that

$$\begin{aligned}\lambda_{G\setminus F}(u) &= \operatorname{lca}(\sigma(L_u)) \\ &= \operatorname{lca}(\sigma(L_v) \cup \sigma(L_w)) \\ &= \operatorname{lca}\{\operatorname{lca}(\sigma(L_v)), \operatorname{lca}(\sigma(L_w))\} \\ &= \operatorname{lca}\{\lambda_{G\setminus F}(v), \lambda_{G\setminus F}(w)\}.\end{aligned}$$

■

The importance of the least-common-ancestor mapping just defined is highlighted by the next result, which shows that given a DTL-scenario α, the lowest possible placement for any gene tree vertex u in the species tree is $\lambda_{G\setminus\Xi}(u)$

Lemma 3: If α is a scenario for S, G, and σ, then

$$\gamma(u) \geq_S \lambda_{G\setminus\Xi}(u), \tag{2}$$

for any vertex $u \in V(G)$.

Proof: Assume α is a scenario for S, G, and σ. For $u \in L(G)$, (2) follows immediately from (I) and the definition of $\lambda_{G\setminus\Xi}$.

Let $u \in \mathring{V}(G)$ be a gene tree vertex with children v and w such that $\gamma(u') \geq_S \lambda_{G\setminus\Xi}(u')$ for each proper descendant u' of u. If $u \in \Sigma$ or $u \in \Delta$, then

$$\gamma(u) \geq_S \operatorname{lca}\{\gamma(v), \gamma(w)\} \geq_S \operatorname{lca}\{\lambda_{G\setminus\Xi}(v), \lambda_{G\setminus\Xi}(w)\} = \lambda_{G\setminus\Xi}(u),$$

where the first inequality follows from (IVb) and (IVc), the second inequality follows from our inductive hypothesis, and the third equality follows from Lemma 2. Hence, $\gamma(u) \geq_S \lambda_{G\setminus\Xi}(u)$. Assume that $u \in \Theta$ and, without loss of generality, let $(u, v) \in \Xi$. By Lemma 2, $\lambda_{G\setminus\Xi}(u) = \lambda_{G\setminus\Xi}(w)$. Moreover, $\gamma(u)$ must be an ancestor of $\gamma(w)$, and hence,

$$\gamma(u) \geq_S \gamma(w) \geq_S \lambda_{G\setminus\Xi}(w) = \lambda_{G\setminus\Xi}(u).$$

■

Next, we show that it is possible to characterize all subsets $F \subset E(G)$ for which there is a DTL-scenario such that $\Xi = F$, and that for each such F, there is a DTL-scenario such that $\Xi = F$ and $\gamma = \lambda_{G\setminus F}$.

A set $F \subset E(G)$ of gene tree edges is called a **transfer set** if no pair of edges in F are siblings and $\lambda_{G\setminus F}(u)$ is incomparable to $\lambda_{G\setminus F}(v)$ for each edge $(u, v) \in F$. We will show that for each transfer set F there is a DTL-scenario such that $\Xi = F$, and that the edges Ξ of

any DTL-scenario is a transfer set. For convenience, we define the concept of anchors, which will also be frequently used later when we discuss our algorithms. We say that $u \in \mathring{V}(G)$ is an **anchor** with respect to a transfer set F iff $\lambda_{G\setminus F}(u) \neq \lambda_{G\setminus F}(v)$ for any child v of u. Note that this is equivalent to $\lambda_{G\setminus F}(v)$ being incomparable to $\lambda_{G\setminus F}(w)$ where v, w are the children of u. Also note that if $(u, v) \in F$, then u is not an anchor with respect to F. See the example on the lower left of Fig. 1 where, in fact, the gene tree is mapped into the species tree using $\lambda_{G\setminus\Xi}$. In the example, the anchors with respect to Ξ are exactly the set of speciations of the gene tree.

Lemma 4: Let S, G and σ be given, and let F be a transfer set. If

$$\Theta = \{u \in \mathring{V}(G) : (u, v) \in F \text{ for some child } v \text{ of } u\},$$

$$\Sigma \subseteq \{u \in \mathring{V}(G) : u \text{ is an anchor w.r.t. } F\}, \quad \text{and}$$

$$\Delta = \mathring{V}(G) \setminus (\Theta \cup \Sigma),$$

then the octuple $(S, G, \sigma, \lambda_{G\setminus F}, \Sigma, \Delta, \Theta, F)$ is a DTL-scenario.

Proof: We only need to verify that each requirement of a DTL-scenario is fulfilled. Using Lemma 2, this verification becomes straightforward and is omitted. ■

Lemma 5: Let α be a DTL-scenario for S, G, and σ. Then Ξ is a transfer set.

Proof: assume that $(u, v) \in \Xi$ and $\lambda_{G\setminus\Xi}(u)$ is comparable to $\lambda_{G\setminus\Xi}(v)$. Then, by Lemma 3, $\gamma(u)$ and $\gamma(v)$ must also be comparable, contradicting (III). Hence, Ξ is a transfer set. ■

We now see that the transfer sets induce a natural partition on the space of all DTL-scenarios. Moreover, given a transfer set Ξ we can obtain all possible mappings of G into S by starting with $\lambda_{G\setminus\Xi}$ and placing gene tree vertices closer to the root of S while ensuring that the conditions of a DTL-scenario, especially (II) and (III), are not violated.

In sections VI and VII we will give algorithms for finding scenarios with the least number of duplications and transfers. As we have seen, the transfer sets determine the possible mappings of a gene tree into the species tree, and by using the least-common-ancestor mapping defined above, we can find a mapping that minimizes the number of duplications and losses, just as in the duplication-loss model. Therefore, our intention will not be to find the exact locations within the species tree where events have taken place, but rather to pinpoint what events have taken place in the gene tree.

V. Finding Most Parsimonious Acyclic Scenarios is NP-Hard

In this section, we will prove that the following decision problem is NP-complete:

DTL-RECONCILIATION

Instance: A species tree/gene tree pair S, G with corresponding leaf mapping $\sigma : L(G) \to L(S)$, and a non-negative integer $J \leq |\mathring{V}(G)|$.

Question: Is there an acyclic DTL-scenario for S, G, and σ with cost at most J?

The NP-completeness will be shown by a reduction from the following NP-complete problem [36]:

MINIMUM FEEDBACK ARC SET

Instance: Directed graph H and positive integer $K \leq |A(H)|$.

Question: Is there a subset $A' \subseteq A(H)$ with $|A'| \leq K$ such that A' contains at least one arc from every directed cycle in H?

Let H and K be given, and let $m = |A(H)|$. We will construct S, G, and σ such that there exists an acyclic DTL-scenario for S, G, and σ with cost at most $J = 2m + K$ if and only if H and K form a yes-instance of MINIMUM FEEDBACK ARC SET.

Let $V = \{r_1, r_2, \ldots, r_n\}$ and $A = \{a_1, a_2, \ldots, a_m\}$ be the sets of vertices and arcs of H, respectively. We now give the species tree and gene tree in Newick format. See also Fig. 2 and 3. For each r_j, let S^{r_j} be the subtree defined as

$$S^{r_j} = (x_{j,K+6}, (x_{j,K+5}, (\ldots, (x_{j,2}, x_{j,1}) \ldots))).$$

Our species tree is then

$$S = (a_1, (a_2, (\ldots, (a_{m+1}, (S^{r_1}, (S^{r_2}, (\ldots, (S^{r_n}, x_{n+1}) \ldots)))) \ldots))).$$

For each $a_i = \langle r_j, r_k \rangle$, let G^{a_i} be the subtree

$$G^{a_i} = (v^i_{j,K+4}, (v^i_{j,K+3}, (\ldots, (v^i_{j,1}, (v^i_{k,K+6}, v^i_{k,K+5})) \ldots))).$$

Our gene tree is then

$$G = \Big((b_1, G^{a_1}), \Big((b_2, G^{a_2}), \Big(\ldots, \Big((b_m, G^{a_m}), b_{m+1}\Big) \ldots\Big)\Big)\Big).$$

The function σ will map the leaves of G to leaves of S according to the subscripts of the leaf labels: $\sigma(v^i_{j,l}) = x_{j,l}$ and $\sigma(b_i) = a_i$.

Lemma 6: If H and K form a yes-instance of MINIMUM FEEDBACK ARC SET, then there is a DTL-scenario for S, G, and σ with cost $J = 2m + K$.

Proof: Assume that H and K form a yes-instance of MINIMUM FEEDBACK ARC SET, so that there is a subset $A' \subseteq A$, $|A'| = K$, containing at least one arc from every directed

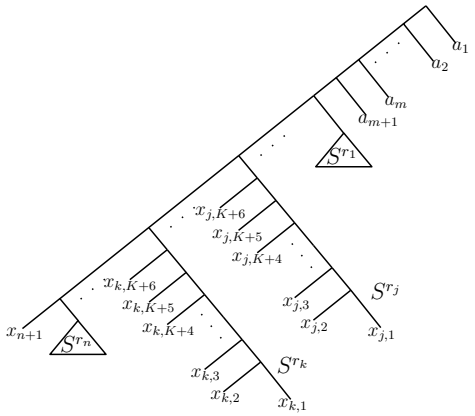


Fig. 2. **The species tree in the NP-completeness proof.** Two subtrees, S^{r_j} and S^{r_k} are shown in full.

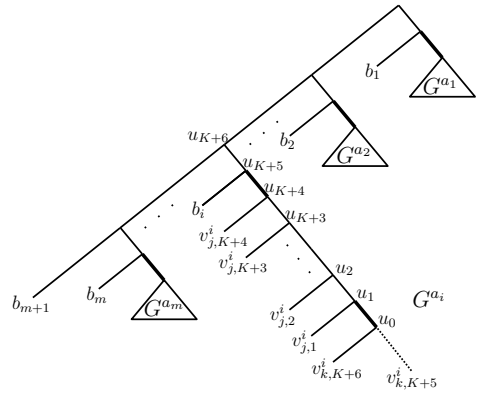


Fig. 3. **The gene tree in the NP-completeness proof.** One subtree G^{a_i} is shown in full, where $a_i = \langle r_j, r_k \rangle$. In Lemma 6, the thick edges are always transfer edges, whereas the dashed edge is a transfer edge iff $a_i \in A'$.

cycle in H. We will now prove that there exists an octuple $\alpha = (S, G, \sigma, \gamma, \Sigma, \Delta, \Theta, \Xi)$ that is an acyclic DTL-scenario with cost $J = 2m + K$.

Let Ξ contain the following J edges of G:

- For each $a_i = \langle r_j, r_k \rangle$ in A, the incoming edge of the root of G^{a_i}.
- For each $a_i = \langle r_j, r_k \rangle$ in A, the incoming edge of $p(v^i_{k,K+5})$.
- For each $a_i = \langle r_j, r_k \rangle$ in A', the incoming edge of $v^i_{k,K+5}$.

See Fig. 3 where the edges of Ξ are highlighted. Let the sets Θ, Σ, and Δ be defined as

$$\Theta = \{u \in \mathring{V}(G) : (u, v) \in \Xi \text{ for some child } v \text{ of } u\},$$

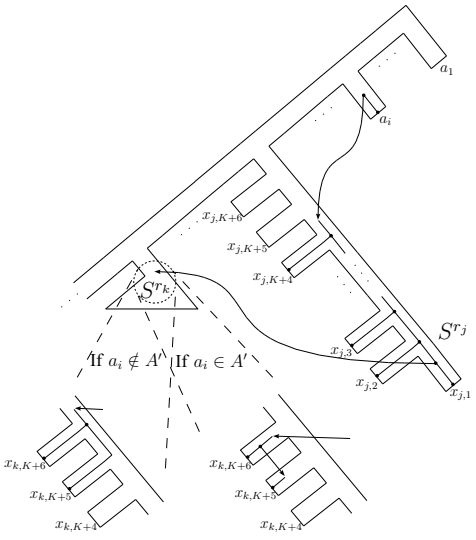


Fig. 4. **Depiction of the mapping of (b_i, G^{a_i}) into S where $a_i = \langle r_j, r_k \rangle$.** Note that the mapping differs depending on whether or not $a_i \in A'$.

$$\Sigma = \mathring{V}(G) \setminus \Theta, \quad \text{and}$$
$$\Delta = \emptyset,$$

and let $\gamma = \lambda_{G \setminus \Xi}$. Note that Θ, Σ, and Δ partition the elements of $\mathring{V}(G)$, and that we are using the least common ancestor mapping, $\lambda_{G \setminus \Xi}$, to map the gene tree into the species tree. Clearly, no two edges of Ξ are sibling edges, so that $\lambda_{G \setminus \Xi}$ is well defined. See Fig. 4 for an illustration of this mapping.

Intuitively, the part of the species tree containing the leaves $x_{j,K+6}$ and $x_{j,K+5}$ should be thought of as the in-section of S^{r_j}, and the part containing $x_{j,K+4}, \ldots, x_{j,1}$ as the out-section of S^{r_j}. The reason, as can be seen in Fig. 4, is that we will recreate arcs of H in our scenario by lateral transfers. An arc $\langle r_j, r_k \rangle$ will result in a lateral transfer from the out-section of S^{r_j} to the in-section of S^{r_k}.

If we can show that (i) Ξ is a transfer set, and that (ii) for each vertex $u \in \Sigma$, u is an anchor w.r.t. Ξ, then we can deduce by Lemma 4 that α is a DTL-scenario.

Consider the subtree (b_i, G^{a_i}) for some $a_i = \langle r_j, r_k \rangle \in A$. For ease of notation, let $u_0, u_1, \ldots, u_{K+6}$ denote the internal vertices of the subtree (b_i, G^{a_i}) and its parent from bottom up, i.e., let u_0 denote $p(v^i_{k,K+5})$ and let u_l denote $p(u_{l-1})$, for $l = 1, \ldots, K+6$. See

Fig. 3.

Claim 6.1:

$$\gamma(u_0) = \begin{cases} x_{k,K+6} & \text{if} \quad a_i \in A', \\ p(x_{k,K+6}) & \text{otherwise,} \end{cases}$$

$$\gamma(u_1) = x_{j,1},$$

$$\gamma(u_l) = p(x_{j,l}), \quad \text{for} \quad l = 2, \ldots, K+4,$$

$$\gamma(u_{K+5}) = a_i,$$

$$\gamma(u_{K+6}) = p(a_i).$$

In proving the above claim we will make frequent use of the result presented in Lemma 2 without repeatedly referring to it.

If $a_i \in A'$, then the edge $(u_0, v^i_{k,K+5})$ is in Ξ, and by Lemma 2, $\gamma(u_0) = \gamma(v^i_{k,K+6}) = x_{k,K+6}$. If $a_i \notin A'$, then $\gamma(u_0) = \mathrm{lca}\{\gamma(v^i_{k,K+5}), \gamma(v^i_{k,K+6})\} = p(x_{k,K+6})$. By our construction, $(u_1, u_0) \in \Xi$, so that $\gamma(u_1) = \gamma(v^i_{j,1}) = x_{j,1}$. Since u_l is not a transfer vertex for $l = 2, \ldots, K+4$, we can show by induction on l that $\gamma(u_l) = \mathrm{lca}\{\gamma(v^i_{j,l}), \gamma(u_{l-1})\} = p(x_{j,l})$. For u_{K+5}, we have that $(u_{K+5}, u_{K+4}) \in \Xi$. Therefore, $\gamma(u_{K+5}) = \gamma(b_i) = a_i$. As a final step consider u_{K+6}. Clearly, $u_{K+6} \in \Sigma$ and $\gamma(u_{K+6}) = p(a_i)$. To see this last point, note that if $i = m$, then $\gamma(u_{K+6}) = \mathrm{lca}\{\gamma(b_{m+1}), \gamma(u_{K+5})\} = \mathrm{lca}\{a_{m+1}, a_m\} = p(a_m)$. We can then show by induction on i that $\gamma(u_{K+6}) = p(a_i)$ for $i = m-1, \ldots, 1$. This ends the proof of the above claim, from which the validity of (i) and (ii) follows (see also Fig. 4). Hence, α is DTL-scenario.

Having shown that α is a DTL-scenario, we now move on to showing that α is acyclic. To do this, we will order the vertices of the species tree such that (V) is fulfilled. Let H' be the DAG that is obtained from H by removing the arcs in A', i.e., $V(H') = V$ and $A(H') = A \setminus A'$. Since H' is a DAG, there is a topological sort of the vertices of H'. Let us fix a topological sort. We now order the vertices of the species tree as follows. The first vertex is $p(a_1)$, followed by $p(a_2)$, and so on until $p(a_{m+1})$. Next, we have $p(\mathrm{root}(S^{r_1}))$, $p(\mathrm{root}(S^{r_2}))$, and so on until $p(\mathrm{root}(S^{r_n}))$. For the rest of the vertices of S, if r_i comes before r_j in the topological sort, then let all internal vertices of S^{r_i} come before the internal vertices of S^{r_j} while respecting the partial order given by the edges of S. Last of all come the leaves of S (in any order). We refer to this ordering as $<$.

Condition (Va) is clearly fulfilled. To show that (Vb) holds, we must consider for each

edge $(u, v) \in \Xi$, all edges $(u', v') \in \Xi$ such that $v \geq_G v'$ and show that $p(\gamma(u)) < \gamma(v')$. Note that, in particular, we need to show that $p(\gamma(u)) < \gamma(v)$. By our choice of transfer edges, we only need to consider three edges for each subtree G^{a_i}.

So, let $a_i = \langle v_j, v_k \rangle \in A$ and, as before, let u_0 denote $p(v^i_{k,K+5})$ and let u_l denote $p(u_{l-1})$, for $l = 1, \ldots, K+6$. The three edges of G^{a_i} that we must consider are (u_{K+5}, u_{K+4}), (u_1, u_0), and $(u_0, v^i_{k,K+5})$. By Claim 6.1, $p(\gamma(u_{K+t})) = p(a_i)$ which is an ancestor of $\gamma(u_{K+4})$, $\gamma(u_0)$, and $\gamma(v^i_{k,K+5})$. By our construction of $<$, it follows that (Vb) is satisfied for the edge (u_{K+5}, u_{K+4}). Now consider the edge (u_1, u_0). By Claim 6.1, $\gamma(u_1) \in V(S^{r_j})$ and $\gamma(u_0) \in V(S^{r_k})$. In fact, we also have that $p(\gamma(u_1)) \in V(S^{r_j})$. If $a_i \notin A'$, then by our construction, all the internal vertices of S^{r_j} come before the internal vertices of S^{r_k}, and therefore, $p(\gamma(u_1)) < \gamma(u_0)$. If, on the other hand, $a_i \in A'$, then $\gamma(u_0) = x_{k,K+6}$ which is a leaf of S and all the leaves come after the internal vertices of S in $<$. Hence, in all cases, $p(\gamma(u_1)) < \gamma(u_0)$. Since $v^i_{k,K+5}$ is mapped by γ to a species tree leaf, and all species tree leaves come after the internal vertices in $<$, (Vb) holds for (u_1, u_0), and $(u_0, v^i_{k,K+5})$. Hence, we have shown that α is an acyclic DTL-scenario. ■

This concludes one direction of our NP-completeness proof. It remains for us to show the opposite direction.

Lemma 7: If there is an acyclic DTL-scenario for S, G, and σ with cost at most J, then H and K form a yes-instance of MINIMUM FEEDBACK ARC SET.

Proof: Let α be an acyclic DTL-scenario for S, G, and σ, with cost $\leq J$. For most of the remainder of the proof we will consider the subtree (b_i, G^{a_i}) for an arbitrary arc $a_i = \langle r_j, r_k \rangle \in A$. As before, let u_0 denote the internal vertex $p(v^i_{k,K+5})$ and let u_l denote $p(u_{l-1})$, for $l = 1 \ldots K+6$. As a first step, we show that there is a cost of at least 2 associated with the gene tree vertices $u_1, \ldots, u_{K+6}$.

Claim 7.1: At most one of u_1 and u_2 is in Σ.

Assume $u_1 \in \Sigma$ and $u_2 \in \Sigma$. Then, by (IVb), $\gamma(u_1)$ must be incomparable to $\gamma(v^i_{j,2}) = x_{j,2}$ and it must be an ancestor of $\gamma(v^i_{j,1}) = x_{j,1}$. There is only one such species tree vertex, namely $x_{j,1}$, which is a leaf. Clearly, by (IVb), u_1 cannot be mapped to a species tree leaf if it is a member of Σ, and we have proved the above claim.

Claim 7.2: At most one of u_{K+5} and u_{K+6} is in Σ.

Assume that $u_{K+5} \in \Sigma$. The children of u_{K+5}, b_i and u_{K+4}, must be mapped by γ to incomparable species tree vertices. Since b_i is a leaf, we know that $\gamma(b_i) = \sigma(b_i) = a_i$. So, $\gamma(u_{K+4})$ must be incomparable to a_i. By Lemma 3, $\gamma(u_{K+4}) \geq_S \lambda_{G \setminus \Xi}(u_{K+4})$. But since

$a_l \notin \sigma(L(G_{u_{K+4}}))$ for $l = 1, \ldots, i$, we conclude that $\gamma(u_{K+4}) \neq a_l$ for $l = 1, \ldots, i$. This, together with the fact that $\gamma(u_{K+4})$ is incomparable to a_i, implies that

$$\gamma(u_{K+4}) \leq_S p(a_{i+1}),$$

so that

$$\begin{aligned}\gamma(u_{K+5}) &= \mathrm{lca}\,\{\gamma(b_i), \gamma(u_{K+4})\} \\ &= \mathrm{lca}\,\{a_i, \gamma(u_{K+4})\} \\ &= p(a_i).\end{aligned}$$

Now, if u_{K+6} is also in Σ, then $\gamma(u_{K+6})$ must be incomparable to $\gamma(u_{K+5})$. But the only vertices incomparable to $p(a_i)$ are $a_1, \ldots, a_{i-1}$, which are leaves of S. Clearly, u_{K+6} cannot be mapped to a leaf if it is a speciation. Hence, if $u_{K+5} \in \Sigma$, then $u_{K+6} \notin \Sigma$.

Our next claim puts a limit on the number of transfers and duplications among the vertices $u_0, \ldots, u_{K+6}$.

Claim 7.3: At most $2+K$ of the vertices $u_0, \ldots, u_{K+6}$ are duplications or transfer vertices.

It follows from the two previous claims that every scenario for S, G and σ has a minimum cost of $2m$. The result then follows by our assumption that our scenario has cost at most $J = 2m + K$.

Next, we will show that there is a transfer event from S^{r_j} to S^{r_k} corresponding to our arc $a_i = \langle r_j, r_k \rangle \in A$.

Let u_q be the $<_G$-minimal vertex in $\{u_1, \ldots, u_{K+3}\}$ such that $(u_q, v^i_{j,q}) \notin \Xi$. Note that such a vertex exists, otherwise $u_l \in \Theta$ for $l = 1, \ldots, K+3$ contradicting claim 7.3.

Claim 7.4: $(u_q, u_{q-1}) \in \Xi$.

Let x be the least common ancestor of the roots S^{r_j} and S^{r_k}. We first show that if $u_q \notin \Theta$, then $\gamma(u_q) \geq_S x$, from which it follows that $u_l \in \Delta$ for $l = q+1, \ldots, K+4$ contradicting Claim 7.3.

Assume that $u_q \notin \Theta$. Now, u_0 has as children two leaves that are mapped by γ to $x_{k,K+5}$ and $x_{k,K+6}$. By (IIb), $\gamma(u_0)$ is an ancestor of at least one of $x_{k,K+5}$ and $x_{k,K+6}$. By the definition of u_q, $(u_p, v^i_{j,p}) \in \Xi$ for $p = 1, \ldots, q-1$. By (IIb) and (III), we have that

$$\gamma(u_{q-1}) \geq_S \gamma(u_{q-2}) \geq_S \cdots \geq_S \gamma(u_0).$$

Hence, $\gamma(u_{q-1})$ is an ancestor of at least one of $x_{k,K+5}$ and $x_{k,K+6}$. From (IVb) and (IVc), we see that irrespective of whether $u_q \in \Delta$ or $u_q \in \Sigma$,

$$\gamma(u_q) \geq_S \mathrm{lca}\,\{\gamma(u_{q-1}), \gamma(v^i_{j,q})\} = \mathrm{lca}\,\{\gamma(u_{q-1}), x_{j,q})\}.$$

Since $\gamma(u_{q-1})$ is an ancestor of one of $x_{k,K+5}$ and $x_{k,K+6}$, we see that $\mathrm{lca}\{\gamma(u_{q-1}), x_{j,q})\} \geq_S x$, i.e.,

$$\gamma(u_q) \geq_S x.$$

The sibling of u_q, i.e., $v^i_{j,q+1}$, is a leaf and is mapped by γ to $x_{j,q+1}$ which is comparable to x. We then get from Lemma 1a that $u_{q+1} \in \Delta$. From (IVc) we get that $\gamma(u_{q+1}) \geq x$. We can now show inductively that $u_l \in \Delta$ for $l = q+1, \ldots, K+4$. This together with the fact that $u_l \in \Theta$ for $l = 1, \ldots, q-1$, contradicts Claim 7.3. Therefore, $u_q \in \Theta$, and by the definition of q, we must have that $(u_q, u_{q-1}) \in \Xi$. This ends the proof of the above claim.

Claim 7.5: $p(\gamma(u_q)) \in \mathring{V}(S^{r_j})$.

Since $u_l \in \Theta$ for $l = 1, \ldots, q$, we deduce from claim 7.3 that $q \leq K+2$ and that at least one vertex among $u_{q+1}, \ldots, u_{K+3}$ is a speciation. Let u_p be the $<_G$-minimal such vertex. We will now show that $\gamma(u_p) = p(x_{j,p})$.

Since $u_p \in \Sigma$, its children, $v_{j,p}$ and u_{p-1}, are mapped by γ to incomparable species tree vertices. Since $v^i_{j,p}$ is a leaf, we know that $\gamma(v^i_{j,p}) = x_{j,p}$. So, $\gamma(u_{p-1})$ is incomparable to $x_{j,p}$. By Lemma 3, $\gamma(u_{p-1}) \geq_S \lambda_{G\setminus\Xi}(u_{p-1})$. The leaves reachable from u_{p-1} in $G \setminus \Xi$ is a subset of $\{v^i_{j,q}, v^i_{j,q+1}, \ldots, v^i_{j,p-1}\}$, so that

$$\lambda_{G\setminus\Xi}(u_{p-1}) \leq_S \mathrm{lca}\{x_{j,q}, x_{j,q+1}, \ldots, x_{j,p-1}\} = p(x_{j,p-1}).$$

Hence, we have that

$$\gamma(u_p) = \mathrm{lca}\{x_{j,p}, \gamma(u_{p-1})\} = p(x_{j,p}).$$

Now, $\gamma(u_q) \geq_S x_{j,q}$, and by (IIa), $\gamma(u_q)$ is not a proper ancestor of $\gamma(u_p) = p(x_{j,p})$. Hence,

$$x_{j,q} \leq_S \gamma(u_q) \leq_S p(x_{j,p}).$$

Clearly, $p(\gamma(u_q)) \in \mathring{V}(S^{r_j})$, and we have proved the claim.

Claim 7.6: if $u_0 \notin \Theta$, then $\gamma(u_{q-1}) \geq_S \mathrm{root}(S^{r_k})$.

Assume that $u_0 \notin \Theta$. By (IVb) and (IVc), we have that

$$\gamma(u_0) \geq \mathrm{lca}\{x_{k,K+5}, x_{k,K+6}\} = \mathrm{root}(S^{r_k}).$$

By the definition of u_q, we have that $(u_l, v^i_{j,l}) \in \Xi$ for $l = 1, \ldots, q-1$, so that

$$\gamma(u_{q-1}) \geq_S \gamma(u_{q-2}) \geq_S \cdots \geq_S \gamma(u_0).$$

Hence, $\gamma(u_{q-1}) \geq_S \mathrm{root}(S^{r_k})$.

Since our scenario α is acyclic, there exists a total order on the vertices of S satisfying condition (V). Let $<$ be such an order. By (V), $p(\gamma(u_q)) < \gamma(u_{q-1})$. Since $\gamma(u_{q-1})$ is an ancestor of the root of S^{r_k}, we have by (Va), that $p(\gamma(u_{q-1})) < z$ for any vertex $z \in V(S^{r_k})$.

To sum things up, we have shown that for any arc $a_i = \langle r_j, r_k \rangle$ of A, if $p(v^i_{k,K+5})$ is not a transfer vertex, then there is a vertex $y \in \mathring{V}(S^{r_j})$ such that $y < z$ for each vertex $z \in V(S^{r_k})$.

All that remains is to construct a subset A' of A such that $|A'| \leq K$, and the graph H', with $V(H') = V$ and $A(H') = A \setminus A'$, is a DAG. We will do this by

$$A' = \{a_i = \langle r_j, r_k \rangle : p(v^i_{k,K+5}) \in \Theta\}.$$

From claims 7.1 and 7.2 it follows that $|A'| \leq K$. Assume that H' contains a directed cycle c. For an arc $a_i = \langle r_j, r_k \rangle$ in c, we know that there is a vertex y of $\mathring{V}(S^{r_j})$ for which $y < z$ for all $z \in \mathring{V}(S^{r_k})$. But this implies that $<$ is a cyclic order on the vertices of S which is absurd. Hence H' is a DAG. ■

Theorem 1: DTL-RECONCILIATION is NP-complete.

Proof: This follows immediately from Lemma 6 and 7. ■

The reader may have noticed that the construction of S, G, and σ may leave some of the leaves of S without corresponding leaves in G. (i.e., $\sigma(L(G)) \neq L(S)$). This does not contradict any of the conditions of a DTL-scenario. However, one may ask whether the hardness result still holds for the special case when $\sigma(L(G)) = L(S)$. The answer is yes, since we can easily reduce the problem of solving the general case to the special case. In fact, it is easy to verify that if S' is obtained from S by removing all subtrees whose leaves have no corresponding leaves in G, then any scenario for S', G, and σ can easily be extended to a scenario for S, G, and σ.

VI. A DYNAMIC PROGRAMMING ALGORITHM

As we saw in the previous section, finding most parsimonious acyclic scenarios is difficult. However, we also saw that the acyclicity requirement was essential in the NP-completeness proof. We will show in the coming sections, that dropping this requirement makes the problem tractable; we are able to find most parsimonious scenarios in polynomial time if we do not require them to be acyclic. We will return to the problem posed by cycles in section VIII. Unless stated otherwise, we will ignore cycles for the time being.

In this section we will present a dynamic programming algorithm that given S, G and σ, computes the cost of a most parsimonious scenario.

We define a counter $c(u,x)$ as the minimum cost of any scenario for G_u and S such that u is mapped to $x \in V(S)$. This, in turn, is the minimum of the minimum costs of having u mapped to x and $u \in \Sigma$, $u \in \Delta$, and $u \in \Theta$. The counters c_1, c_2, and c_3 given below will represent these three mutually exclusive cases. The recursion is as follows:
If $u \in L(G)$, then

$$c(u,x) = \begin{cases} 0 & \text{if } x = \sigma(u), \\ \infty & \text{otherwise.} \end{cases}$$

If $u \in \mathring{V}(G)$ with children v and w, then

$$c(u,x) = \min\{c_1(u,x), c_2(u,x), c_3(u,x)\},$$

where

$$c_1(u,x) = \begin{cases} \min\{c(v,y) + c(w,z) : y \text{ incomparable to } z, \text{ and } \operatorname{lca}\{y,z\} = x\ \} & \text{if } x \in \mathring{V}(S), \\ \infty & \text{otherwise,} \end{cases}$$

$$c_2(u,x) = \min\{1 + c(v,y) + c(w,z) : y \leq_S x,\ z \leq_S x\},$$

$$c_3(u,x) = \min\{1 + c(v,y) + c(w,z) : y \leq_S x \text{ and } z \text{ incomparable to } x\}.$$

The minimum cost of a scenario reconciling S and G is then given by

$$\min_{x \in V(S)} c(\operatorname{root}(G), x).$$

An algorithm for computing the above recursion can easily be implemented to run in time $O(|V(G)| \cdot |V(S)|^2)$. Note that although each minimum in the expressions above is taken over a quadratic number of terms, they can easily be computed in linear time. For example,

$$\begin{aligned} c_2(u,x) &= \min\{1 + c(v,y) + c(w,z) : y \leq_S x, z \leq_S x\} \\ &= \min\{c(v,y) : y \leq_S x\} + \min\{c(w,z) : y \leq_S x\} + 1, \end{aligned}$$

and similarly for the other cases. Algorithm 1 shows explicitly how the recursions may be implemented to achieve the above mentioned time complexity. We will now prove the correctness of our recursion.

For a DTL-scenario α and a gene tree vertex $v \in V(G)$, define the restriction of α to G_v as

$$\alpha|_{G_v} = (S, G_v, \sigma|_{L(G_v)}, \gamma|_{V(G_v)}, \Sigma \cap V(G_v), \Delta \cap V(G_v), \Theta \cap V(G_v), \Xi \cap E(G_v)).$$

Algorithm 1: Dynamic programming algorithm.

Input: S, G, and σ.

Output: Minimum cost of any DTL-scenario for S, G, and σ.

$c \leftarrow \mathsf{Array}\big[1..|V(G)|, 1..|V(S)|\big]$ initialized to ∞
for $u \in L(G)$ **do**
    $c(u, \sigma(u)) \leftarrow 0$
end
for $u \in \mathring{V}(G)$ *in postorder* **do**
    for $x \in V(S)$ *in postorder* **do**
        Let v, w be the children of u
        if $x \in \mathring{V}(S)$ **then**
            Let y, z be the children of x
            $c_1 \leftarrow \min\Big(\min_{y' \leq y} c(v, y') + \min_{z' \leq z} c(w, z'), \min_{y' \leq y} c(w, y') + \min_{z' \leq z} c(v, z')\Big)$
        else
            $c_1 \leftarrow \infty$
        end
        $c_2 \leftarrow \min_{x' \leq x} c(v, x') + \min_{x' \leq x} c(w, x') + 1$
        $c_3 \leftarrow \min\Big(\min_{x' \leq x} c(v, x') + \min_{\substack{x' \not\leq x \\ x' \not\geq x}} c(w, x'), \min_{x' \leq x} c(w, x') + \min_{\substack{x' \not\leq x \\ x' \not\geq x}} c(v, x')\Big)$
        $c(u, x) \leftarrow \min\{c_1, c_2, c_3\}$
    end
end
return $\min_{x \in V(S)} c(\mathrm{root}(G), x)$

It is easy to verify that any restriction of a DTL-scenario is itself a DTL-scenario. The next two lemmas show that we can decompose DTL-scenarios into smaller ones and, given certain natural conditions, we are able to combine smaller DTL-scenarios into larger ones.

Lemma 8: Let S, G, and σ be given, and let v and w be the children of $u = \mathrm{root}(G)$. Fix a species tree vertex x.

Assume that α_0 is a scenario for S, G, and σ such that $\gamma_0(u) = x$, and let $\alpha_1 = \alpha_0|_{G_v}$ and $\alpha_2 = \alpha_0|_{G_w}$.

(a) If $u \in \Sigma_0$, then $\gamma_1(v)$ is incomparable to $\gamma_2(w)$, and $\mathrm{lca}\,\{\gamma_1(v), \gamma_2(w)\} = x$.

(b) If $u \in \Delta_0$, then $\gamma_1(v) \leq_S x$ and $\gamma_2(w) \leq_S x$.

(c) If $u \in \Theta_0$, then $\gamma_1(v) \leq_S x$ and $\gamma_2(w)$ is incomparable to x, or $\gamma_2(w) \leq_S x$ and $\gamma_1(v)$ is incomparable to x.

Proof: If we note that $\gamma_0(v) = \gamma_1(v)$ and $\gamma_0(w) = \gamma_2(w)$, then a, b, and c follow immediately from the definition of DTL-scenarios. ■

Lemma 9: Let S, G, and σ be given and let v and w be the children of $u = \mathrm{root}(G)$. Fix a species tree vertex x.

Assume that α_1 is a DTL-scenario for S, G_v, and $\sigma|_{L(G_v)}$, and α_2 is a DTL-scenario for S, G_w, and $\sigma|_{L(G_w)}$.

(a) If $\gamma_1(v)$ is incomparable to $\gamma_2(w)$, and $\mathrm{lca}\,\{\gamma_1(v), \gamma_2(w)\} = x$, then there is a DTL-scenario α_0 for S, G, and σ such that $\alpha_1 = \alpha_0|_{G_v}$, $\alpha_2 = \alpha_0|_{G_w}$, $\gamma_0(u) = x$, and $u \in \Sigma_0$,

(b) If $\gamma_1(v) \leq_S x$, and $\gamma_2(w) \leq_S x$, then there is a DTL-scenario α_0 for S, G and σ such that $\alpha_1 = \alpha_0|_{G_v}$, $\alpha_2 = \alpha_0|_{G_w}$, $\gamma_0(u) = x$, and $u \in \Delta_0$.

(c) If $\gamma_1(v) \leq_S x$, and $\gamma_2(w)$ is incomparable to x, then there is a DTL-scenario α_0 for S, G and σ such that $\alpha_1 = \alpha_0|_{G_v}$, $\alpha_2 = \alpha_0|_{G_w}$, $\gamma_0(u) = x$, and $u \in \Theta_0$.

Proof: For (a), let α_1 and α_2 be given. Assume that $\gamma_1(v)$ is incomparable to $\gamma_2(w)$, and that $\mathrm{lca}\,\{\gamma_1(v), \gamma_2(w)\} = x$. Let α_0 be the octuple

$$\alpha_0 = (S, G, \sigma, \gamma_0, \Sigma_1 \cup \Sigma_2 \cup \{u\}, \Delta_1 \cup \Delta_2, \Theta_1 \cup \Theta_2, \Xi_1 \cup \Xi_2),$$

where $\gamma_0 : V(G) \to V(S)$ is an extension of both γ_1 and γ_2 with $\gamma_0(u) = x$. It is then straightforward to verify that conditions (I)-(IV) (in particular (IVb)) are fulfilled for u. That the conditions are fulfilled for the rest of the gene tree vertices and edges follows from the fact that α_1 and α_2 are DTL-scenarios.

In the same way (b) and (c) can be proved by instead considering the octuples

$$\alpha_0 = (S, G, \sigma, \gamma_0, \Sigma_1 \cup \Sigma_2, \Delta_1 \cup \Delta_2 \cup \{u\}, \Theta_1 \cup \Theta_2, \Xi_1 \cup \Xi_2), \quad \text{and}$$

$$\alpha_0 = (S, G, \sigma, \gamma_0, \Sigma_1 \cup \Sigma_2, \Delta_1 \cup \Delta_2, \Theta_1 \cup \Theta_2 \cup \{u\}, \Xi_1 \cup \Xi_2 \cup \{(u, w)\}),$$

respectively. ■

Finally, the theorem below proves that the recursions above correctly compute the minimum cost of reconciling a gene tree and species tree.

Theorem 2: Let S, G, and σ be given. For $u \in V(G)$ and $x \in V(S)$ define the set

$$A = \{\alpha \text{ a DTL-scenario for } S,\ G_u, \text{ and } \sigma|_{L(G_u)} : \gamma(u) = x\}.$$

Then,

$$c(u,x) = \begin{cases} \infty & \text{if } A = \emptyset, \\ \min\limits_{\alpha \in A} |\alpha| & \text{otherwise.} \end{cases}$$

Proof: The theorem is clearly true if u is a leaf.

Assume that $u \in \mathring{V}(G)$ and that the theorem is true for all proper descendants of u for all species tree vertices. Let v and w be the children of u and define the following three sets:

$$A_1 = \{\alpha \in A : u \in \Sigma\},$$

$$A_2 = \{\alpha \in A : u \in \Delta\},$$

$$A_3 = \{\alpha \in A : u \in \Theta\}.$$

Claim 9.1:

$$c_1(u,x) = \begin{cases} \infty & \text{if } A_1 = \emptyset, \\ \min\limits_{\alpha \in A_1} |\alpha| & \text{otherwise.} \end{cases}$$

Assume $c_1(u,x) \neq \infty$. Then, by the definition of $c_1(u,x)$, there is a pair y, z of incomparable vertices in S such that $\operatorname{lca}\{y,z\} = x$, $c(v,y) \neq \infty$, and $c(w,z) \neq \infty$. By our inductive hypothesis, this implies that there are scenarios α_1 for S, G_v, $\sigma|_{L(G_v)}$, and α_2 for S, G_w, $\sigma|_{L(G_w)}$, such that $\gamma_1(v) = y$ and $\gamma_2(w) = z$. From Lemma 9a, we then see that $A_1 \neq \emptyset$. Therefore, if $A_1 = \emptyset$, then $c_1(u,x) = \infty$.

Assume $A_1 \neq \emptyset$. Let $\alpha \in A_1$ be a scenario with minimum cost and consider the restrictions $\alpha_1 = \alpha|_{G_v}$ and $\alpha_2 = \alpha|_{G_w}$ of α. Clearly, by our inductive hypothesis, $c(v, \gamma_1(v)) \leq |\alpha_1|$. In fact, since α is a minimum-cost scenario in A, $c(v, \gamma_1(v)) = |\alpha_1|$. Similarly, $|\alpha_2| = c(w, \gamma_2(w))$. From Lemma 8a, we deduce that

$$c_1(u,x) \leq c(v, \gamma_1(v)) + c(v, \gamma_2(w)) = |\alpha|.$$

Assume that the above inequality is strict, i.e., $c_1(u,x) < |\alpha|$. Then there is a pair of incomparable vertices y, z in S such that $c(v,y) + c(w,z) < |\alpha|$ and $\operatorname{lca}\{y,z\} = x$. By our inductive hypothesis, this implies that there are scenarios α_3 for S, G_v, $\sigma|_{L(G_v)}$, and α_4 for S, G_w, $\sigma|_{L(G_w)}$, such that $\gamma_3(v) = y$, $\gamma_4(w) = z$, and $|\alpha_3| + |\alpha_4| < |\alpha|$. From Lemma 9a, we see that there is a scenario α_0 with $\alpha_3 = \alpha_0|_{G_v}$, $\alpha_4 = \alpha_0|_{G_w}$, and $u \in \Sigma_0$. Clearly, $\alpha_0 \in A_1$. The fact that $\alpha_0 \in A_1$ and

$$|\alpha_0| = |\alpha_3| + |\alpha_4| < |\alpha|,$$

produces a contradiction. Therefore, $c_1(u,x) = |\alpha|$, and we have proved the above claim.

The next two claims are stated without proofs as the structure of their proofs are very similar to that of the above claim.

Claim 9.2:

$$c_2(u,x) = \begin{cases} \infty & \text{if } A_2 = \emptyset, \\ \min\limits_{\alpha \in A_2} |\alpha| & \text{otherwise.} \end{cases}$$

Claim 9.3:

$$c_3(u,x) = \begin{cases} \infty & \text{if } A_3 = \emptyset, \\ \min\limits_{\alpha \in A_3} |\alpha| & \text{otherwise.} \end{cases}$$

The theorem follows immediately from the above claims. ■

A. A Dynamic Programming Algorithm for the DT-cost set problem

In the previous section we saw how to compute the minimum cost of reconciling S, G, and σ. It can, however, be desirable to know the number of duplications and transfers that are involved in an optimal scenario separately. We can use the same recursive idea as in the previous section to find all pairs (d,t) such that d is the number of duplications and t is the number of transfers in some optimal scenario. Instead of using a counter to keep track of the minimal cost, we will use sets containing pairs of numbers. The recursion is given below without proof. Note that we define $\operatorname{argmin}_{x \in X} f(x)$ to be the *set* of arguments where $f(x)$ attains its minimum value. Also, similar to the counters c_1, c_2, and c_3 above, the sets A_1, A_2, and A_3 defined below correspond to costs of the gene tree vertex under consideration being a speciation, duplication, and transfer vertex respectively. An algorithm for computing the recursion below can easily be implemented to run in time $O(mn(m+n^2))$, where $m = |V(S)|$ and $n = |V(G)|$.

If $u \in L(G)$

$$c(u,x) = \begin{cases} \{(0,0)\} & \text{if } x = \sigma(u) \\ \emptyset & \text{otherwise.} \end{cases}$$

If $u \in \mathring{V}(G)$

$$c(u,x) = \operatorname*{argmin}_{(d,t) \in A_1 \cup A_2 \cup A_3} d + t,$$

where the sets A_1, A_2, and A_3 are defined as:

$$A_1 = \{(d_1 + d_2, t_1 + t_2) : (d_1, t_1) \in c(v, y), (d_2, t_2) \in c(w, z),$$
$$\text{where } y \text{ is incomparable to } z, \text{ and } \mathrm{lca}\,\{y, z\} = x\},$$
$$A_2 = \{(d_1 + d_2 + 1, t_1 + t_2) : (d_1, t_1) \in c(v, y), (d_2, t_2) \in c(w, z),$$
$$\text{where } y \leq_S x, \text{ and } z \leq_S x\},$$
$$A_3 = \{(d_1 + d_2, t_1 + t_2 + 1) : (d_1, t_1) \in c(v, y), (d_2, t_2) \in c(w, z),$$
$$\text{where } y \leq_S x \text{ and } z \text{ incomparable to } x\}.$$

VII. A Fixed-Parameter-Tractable Algorithm

In this section we present a fixed-parameter-tractable algorithm for reconciling S, G, and σ, which is able to enumerate all optimal reconciliations. The time complexity of the algorithm will be shown to be polynomial in the size of the input when considering the minimum cost of reconciling S and G as a fixed parameter.

As we saw in section IV, the transfer sets induce a natural partition of the space of DTL-scenarios, and the main idea behind the FPT-algorithm is to use transfer sets as a basis for searching in this space. We know that there is a DTL-scenario for each transfer set, but every transfer set also entails certain restrictions as to the placement of gene tree vertices within the species tree as seen in Lemma 3. This, in turn, forces the introduction of duplications by Lemma 1. Hence, choosing to include or not to include an edge in a transfer set has consequences in terms of forced duplications. Below, we will define the notion of candidates which will be central to our search strategy. The purpose of a candidate is to keep track of forced duplications with respect to transfer sets and anchors. We remind the reader that a gene tree vertex is called an anchor w.r.t. a transfer set F if its children are mapped to incomparable species tree vertices by $\lambda_{G\setminus F}$. It can then be seen from (IVb) and Lemma 4 that for each transfer set F, there is a DTL-scenario in which all anchors w.r.t. F are speciations.

A tuple (D, F) is called a **candidate** for S, G, and σ iff $D \subseteq \mathring{V}(G)$, F is a transfer set, and for each $u \in D$, $(u, v) \notin F$ for any child v of u. An internal gene tree vertex u is **unmarked** with respect to a candidate (D, F) iff $u \notin D$ and $(u, v) \notin F$ for any child v of u. A candidate (D, F) is **final** iff each unmarked vertex u is an anchor w.r.t. F. We will say that u is an anchor w.r.t. a candidate (D, F), when and only when u is an anchor w.r.t. F. The cost of a candidate (D, F) is defined as $|(D, F)| = |D| + |F|$. A final candidate with

minimal cost is called **optimal**. Finally, we write $(D_1, F_1) \preceq (D_2, F_2)$ when $D_1 \subseteq D_2$ and $F_1 \subseteq F_2$.

We now show that (D, F) is a final candidate if and only if there is a DTL-scenario such that $D = \Delta$ and $F = \Xi$. It then follows that given any final candidate (D, F), there is a set of corresponding scenarios $\{\alpha : \Delta = D, \Xi = F\}$ in which the only difference between two distinct scenarios is the mapping of the gene tree into the species tree. There is, in general, an exponential number of ways to map a gene tree into a species tree even when keeping Δ, Ξ, and Σ fixed.

Lemma 10: If (D, F) is a final candidate for S, G, and σ, then there is a DTL-scenario α for S, G, and σ such that $\Delta = D$ and $\Xi = F$.

Proof: Assume that (D, F) is a final candidate for S, G, and σ. Let

$$\Theta = \{u \in \mathring{V}(G) : (u, v) \in F \text{ for some child } v \text{ of } u\},$$

$$\Delta = D, \qquad \text{and}$$

$$\Sigma = \{u \in \mathring{V}(G) : u \notin \Theta, u \notin \Delta\}.$$

For each $u \in \Sigma$, u is unmarked w.r.t. (D, F), and since (D, F) is final, u is an anchor w.r.t. F. By Lemma 4, we see that the octuple $(S, G, \sigma, \lambda_{G \setminus F}, \Sigma, \Delta, \Theta, F)$ is a DTL-scenario. ■

Lemma 11: If α is a scenario for S, G, and σ, then (Δ, Ξ) is a final candidate for S, G, and σ.

Proof: Assume that α is a scenario for S, G, and σ. Since $\Delta \cap \Theta = \emptyset$ and Ξ is a transfer set, we have that (Δ, Ξ) is a candidate. Clearly, Σ is the set of unmarked vertices of (Δ, Ξ). Let $u \in \Sigma$ with children v and w. We need to show that u is an anchor w.r.t. Ξ. From Lemma 3, we have that $\lambda_{G \setminus \Xi}(v) \leq_S \gamma(v)$ and $\lambda_{G \setminus \Xi}(w) \leq_S \gamma(w)$. Therefore, since $\gamma(v)$ is incomparable to $\gamma(w)$ by (IVb), $\lambda_{G \setminus \Xi}(v)$ is incomparable to $\lambda_{G \setminus \Xi}(w)$ and u is an anchor w.r.t. Ξ. Hence, (Δ, Ξ) is a final candidate. ■

As stated earlier, our main interest is not determining the exact location within the species tree where events have taken place, but rather to identify the set of duplications and transfers in the gene tree by finding parsimonious scenarios. Therefore, we define an equivalence relation on the set of scenarios such that two scenarios α_1 and α_2 for S, G, and σ are equivalent iff $\Sigma_1 = \Sigma_2$, $\Delta_1 = \Delta_2$, and $\Xi_1 = \Xi_2$. It is then clear that every final candidate can be taken as the representative of an equivalence class of scenarios.

In the remainder of this section we will give an algorithm that, given S, G, and σ, enumerates all optimal candidates. We can think of this computation as search for optimal

candidates in the space of candidates for S, G, and σ. Consider a search tree in the space of all candidates where the root is the empty candidate, $(\emptyset, \emptyset)$, and each candidate C has as its children exactly the candidates C' such that $C \preceq C'$ and $|C'| = |C| + 1$ (so each child of C is obtained by adding exactly one duplication or one transfer to C). We will show that we can prune this search tree in such a way that no candidate has more than three children and that all optimal candidates are reachable from the root. The algorithm we present below performs an implicit breadth-first search in this "pruned" search tree.

Let $C = (D, F)$ be a candidate for S, G, and σ and let f be the set of transfer vertices of C, i.e., $f = \{u \in V(G) : (u, v) \in F \text{ for some child } v \text{ of } u\}$. Consider the forest $G \setminus F$. The vertices with only one outgoing edge in $G \setminus F$ are exactly the set of transfer vertices of C. We will often need to speak of pairs of gene tree vertices $u \notin f, v \notin f$, such that u is a proper ancestor of v in $G \setminus F$ and for each vertex w such that $u >_G w >_G v$, w is a transfer vertex. For ease of notation, we will now introduce the rooted forest $G \natural F$ that is obtained from $G \setminus F$ by contracting any paths that contain only transfer vertices into a single edge. Note that (u, v) is an edge of $G \natural F$ iff u is a proper ancestor of v in $G \setminus F$ and every vertex on the path from u to v that is distinct from u and v is a transfer vertex. This implies that if (u, v) is an edge of $G \natural F$ and $u >_G w >_G v$, then $\lambda_{G\setminus F}(w) = \lambda_{G\setminus F}(v)$. Also, note that $G \natural F$ is a full rooted binary forest, i.e., every vertex of $G \natural F$ has two outgoing edges in $G \natural F$.

Let $C = (D, F)$ be a candidate for S, G, and σ. An unmarked anchor u w.r.t. C is called an **s-move** iff $(p, u) \in E(G \natural F)$ for some unmarked vertex p and $\lambda_{G\setminus F}(p) = \lambda_{G\setminus F}(u)$. An unmarked vertex u of C is called a **d-move** iff $(u, v) \in E(G \natural F)$ for some vertex $v \in D$ and $\lambda_{G\setminus F}(u) = \lambda_{G\setminus F}(v)$. As we will see, we must decide for each s-move of a candidate C whether it is to become a speciation, in which case its parent in $G \natural F$ is a forced duplication, or to have it be a transfer vertex. A d-move is simply a vertex that must be a duplication due to the choices made so far. Given these definitions, we will now show that Algorithm 2 enumerates all optimal candidates and can be implemented to run in time $O(m + n \cdot 3^c)$, where $n = |V(G)|$, $m = |V(S)|$, and c is the minimum cost of any final candidate for S, G, and σ. Therefore, keeping the cost c fixed, the algorithm runs in polynomial time in m and n.

First, we give two technical lemmas that will be needed later. Then we will show that a candidate is final iff it contains no d-moves or s-moves. This is the main idea behind Algorithm 2; we identify d-moves and s-moves and eliminate them by introducing duplica-

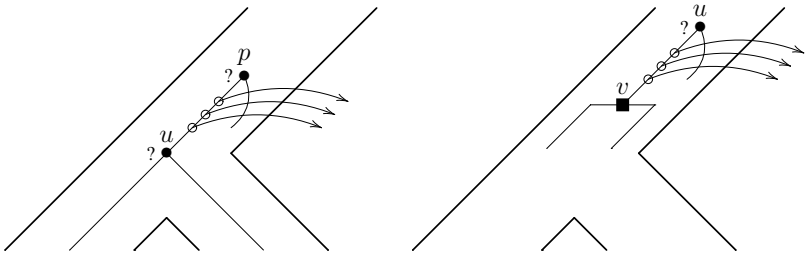


Fig. 5. **Illustration of the d- and s-moves of a candidate** $C = (D, F)$**.** The left figure shows a portion of the gene tree mapped inside the species tree using $\lambda_{G\setminus F}$. The vertex u is an anchor, and both u and p are unmarked and mapped to the same species tree vertex. The vertices in the path between p and u are all transfer vertices of C. As proved in the text, every optimal final candidate $C' \succeq C$ either has one of the outgoing edges of u as a transfer edge, or u is unmarked and $p \in D'$. The right figure illustrates a d-move. Both u and v are mapped to the same vertex by $\lambda_{G\setminus F}$, $v \in D$, and u is unmarked. Every optimal final candidate $C' \succeq C$ will have $u \in D'$.

tions and transfers. Note that when eliminating a move, at most three candidates need to be considered. As we will show further below, the order in which we consider the moves is irrelevant.

Lemma 12: A gene tree vertex u is an anchor w.r.t. a candidate (D, F) iff $\lambda_{G\setminus F}(u) \neq \lambda_{G\setminus F}(v)$ for each edge (u, v) of $G \setminus\!\!\setminus F$.

Proof: Let $C = (D, F)$ be a candidate. Assume u is an anchor w.r.t. C and let (u, v) be an edge of $G \setminus\!\!\setminus F$. Consider the child v' of u that is an ancestor of v. By the definition of $G \setminus\!\!\setminus F$, we have that $\lambda_{G\setminus F}(v') = \lambda_{G\setminus F}(v)$. Since u is an anchor, $\lambda_{G\setminus F}(v') \neq \lambda_{G\setminus F}(u)$, and hence, $\lambda_{G\setminus F}(v) \neq \lambda_{G\setminus F}(u)$.

Assume that $\lambda_{G\setminus F}(u) \neq \lambda_{G\setminus F}(v)$ for an edge (u, v) of $G \setminus\!\!\setminus F$. Consider the child v' of u that is an ancestor of v. By the definition of $G \setminus\!\!\setminus F$, $\lambda_{G\setminus F}(v') = \lambda_{G\setminus F}(v) \neq \lambda_{G\setminus F}(u)$. Hence, if u is a vertex of $G \setminus\!\!\setminus F$ such that $\lambda_{G\setminus F}(u) \neq \lambda_{G\setminus F}(v)$ for each edge (u, v) of $G \setminus\!\!\setminus F$, then $\lambda_{G\setminus F}(v') \neq \lambda_{G\setminus F}(u)$ for each child v' of u, and u is an anchor. ■

Lemma 13: Let $C = (D, F)$ and $C' = (D', F')$ be candidates for S, G, and σ such that $C \preceq C'$. We then have that

(a) $\lambda_{G\setminus F'}(u) \leq_S \lambda_{G\setminus F}(u)$ for any gene tree vertex u.

(b) If u is unmarked in C, but is a transfer vertex in C', then $\lambda_{G\setminus F'}(u) <_S \lambda_{G\setminus F}(u)$.

(c) If $\lambda_{G\setminus F}(p_G(u)) = \lambda_{G\setminus F}(u)$ and $\lambda_{G\setminus F'}(u) = \lambda_{G\setminus F}(u)$, then $\lambda_{G\setminus F'}(p_G(u)) = \lambda_{G\setminus F}(p_G(u))$.

(d) If $\lambda_{G\setminus F'}(u) \neq \lambda_{G\setminus F}(u)$, then there is an anchor v w.r.t. C such that $\lambda_{G\setminus F}(v) = \lambda_{G\setminus F}(u)$

Algorithm 2: Fixed-parameter-tractable algorithm.

Input: S, G, and σ.

Output: All optimal final candidates for S, G, and σ.

```
1  opt ← ∞
2  Q ← {(∅, ∅)} ;                                   // Initialize queue
3  while Q ≠ ∅ do
4      (D, F) ← dequeue(Q)
5      if |(D, F)| ≤ opt then
6          if there is an s-move u in (D, F) then
7              let v, w be the children of u
8              C1 ← (D, F ∪ {(u, v)})
9              C2 ← (D, F ∪ {(u, w)})
10             C3 ← (D ∪ {p_{G\F}(u)}, F)
11             Q ← Q ∪ {C1, C2, C3}
12         else if there is a d-move u in (D, F) then
13             C1 ← (D ∪ {u}, F)
14             Q ← Q ∪ {C1}
15         else
16             solutions ← solutions ∪ {(D, F)}
17             opt ← |(D, F)|
18         end
19     end
20 end
21 return solutions
```

and v is a transfer vertex in C'.

Proof:

(a) Since $C \preceq C'$, we have that $F \subseteq F'$ and the result follows from Lemma 2.

(b) Assume that u is unmarked in C but is a transfer vertex in C'. Let $(u, v) \in F$ for some child v of u. From (a), we already know that $\lambda_{G\setminus F'}(u) \leq_S \lambda_{G\setminus F}(u)$. Since u is unmarked in C, $\lambda_{G\setminus F}(u)$ is comparable to $\lambda_{G\setminus F}(v)$. From (a), we have that $\lambda_{G\setminus F}(v) \leq_S \lambda_{G\setminus F'}(v)$. Therefore, if $\lambda_{G\setminus F'}(u) = \lambda_{G\setminus F}(u)$, then $\lambda_{G\setminus F'}(u)$ is comparable to $\lambda_{G\setminus F'}(v)$ so that F'

is not a transfer set. Hence, $\lambda_{G\backslash F'}(u) \neq \lambda_{G\backslash F}(u)$.

(c) Assume that $\lambda_{G\backslash F}(p_G(u)) = \lambda_{G\backslash F}(u)$ and $\lambda_{G\backslash F'}(u) = \lambda_{G\backslash F}(u)$. We see from (a) that $\lambda_{G\backslash F'}(p_G(u)) \leq_S \lambda_{G\backslash F}(p_G(u)) = \lambda_{G\backslash F}(u)$. Since F' is a transfer set, it follows from Lemma 2 that $\lambda_{G\backslash F'}(p_G(u))$ is not a proper descendant of $\lambda_{G\backslash F'}(u) = \lambda_{G\backslash F}(u)$. Hence $\lambda_{G\backslash F'}(p_G(u)) = \lambda_{G\backslash F}(u) = \lambda_{G\backslash F}(p_G(u))$.

(d) Assume that $\lambda_{G\backslash F'}(u) \neq \lambda_{G\backslash F}(u)$. Let $u = u_1, u_2, \ldots, u_n$ be a path in G such that $\lambda_{G\backslash F}(u_i) = \lambda_{G\backslash F}(u)$ and u_n is an anchor w.r.t. C. If u_n is not a transfer vertex in C', then $\lambda_{G\backslash F'}(u_n) = \lambda_{G\backslash F}(u_n)$. From (c), we see that $\lambda_{G\backslash F'}(u_i) = \lambda_{G\backslash F}(u_i)$ for $i = 1, \ldots, n$ which is a contradiction. Hence, u_n is a transfer vertex in C'.

■

Using the rather technical lemmas above, we can now show that final candidates correspond exactly to candidates with no moves.

Lemma 14: C is a final candidate iff C is a candidate with no d-moves or s-moves.

Proof: Assume that C is final. Each unmarked vertex of C is an anchor w.r.t. C. If u is a d-move in C, then there is a vertex v such that $(u, v) \in E(G \natural F)$ and $\lambda_{G\backslash F}(v) = \lambda_{G\backslash F}(u)$, and by Lemma 12, u is not an anchor. Hence, there are no d-moves in C. If u is an s-move in C, there is an edge $(p, u) \in E(G \natural F)$ such that p is unmarked and $\lambda_{G\backslash F}(p) = \lambda_{G\backslash F}(u)$, and by Lemma 12, p is not an anchor. Hence, there are no s-moves in C.

For the other direction, assume that $C = (D, F)$ is a candidate with no d-moves or s-moves. Assume that there is an unmarked vertex in C that is not an anchor and let u be a $<_G$-minimal such vertex. By Lemma 12, there is an edge (u, v) in $G \natural F$ such that $\lambda_{G\backslash F}(u) = \lambda_{G\backslash F}(v)$. Clearly, v is not a transfer vertex of C. If v is unmarked in C, then by the $<_G$-minimality of u, v is an anchor, implying that v is an s-move, producing a contradiction. If $v \in D$, then u is a d-move which is also a contradiction. Since we get a contradiction in all cases, we must have that each unmarked vertex of C is an anchor, i.e., C is final. ■

The next two lemmas show that there are three mutually exclusive ways to resolve an s-move. Either the s-move is kept as a speciation, in which case its parent in $G \natural F$ must be labeled a duplication, or one of its outgoing edges is added as a transfer edge.

Lemma 15: Let $C = (D, F)$ be a candidate for S, G, and σ. If u is an s-move in C with children v and w, then for any optimal final candidate $C^* = (D^*, F^*)$ such that $C \preceq C^*$

(a) $u \notin D^*$, and

(b) if u is not a transfer vertex in C^*, then $p_{G\natural F}(u) \in D^*$.

Proof: Assume that u is an s-move in C with children v, w and let $C^* = (D^*, F^*)$ be an optimal candidate such that $C \preceq C^*$. Let $p = p_{G\setminus F}(u)$.

By the definition of an s-move, u is an anchor w.r.t. C implying that $\lambda_{G\setminus F}(v)$ is incomparable to $\lambda_{G\setminus F}(w)$. By Lemma 13a, $\lambda_{G\setminus F^*}(v) \leq_S \lambda_{G\setminus F}(v)$ and $\lambda_{G\setminus F^*}(w) \leq_S \lambda_{G\setminus F}(w)$. Therefore, $\lambda_{G\setminus F^*}(v)$ is incomparable to $\lambda_{G\setminus F^*}(w)$. It follows that if u is not a transfer vertex in C^*, then $\lambda_{G\setminus F^*}(u) = \lambda_{G\setminus F}(u)$, and by repeated application of Lemma 13c to the ancestors of u, we have that $\lambda_{G\setminus F^*}(p) = \lambda_{G\setminus F}(p)$, so that p is not a transfer vertex in C^* by Lemma 13b.

(a) Assume $u \in D^*$. By the discussion above, p is not a transfer vertex in C^*. If p is unmarked in C^*, then p is a d-move in C^*, which is a contradiction. So $p \in D^*$. But then, since C^* has no moves, $(D^* \setminus \{u\}, F^*)$ has no moves and is therefore final contradicting the optimality of C^*. Hence, $u \notin D^*$.

(b) Assume u is not a transfer vertex in C^*. Then p is not a transfer vertex in C^*. If p is unmarked in C^*, then p is a d-move in C^* which is a contradiction. Hence, $p \in D^*$.

■

Lemma 16: Only candidates are inserted into Q in Algorithm 2.

Proof: Clearly, at the start of the first iteration of the while-loop on line 3, Q contains only candidates.

Assume that Q only contains candidates at the start of some iteration of the loop. Then, (D, F) dequeued on line 4 is a candidate. Clearly, for any unmarked vertex u, $(D \cup \{u\}, F)$ is a candidate. Now, let u be an s-move with children v, w. Let $F' = F \cup \{(u, w)\}$. We will now show that F' is a transfer set. Let (u', u'') be an edge in F' distinct from (u, w). Since F is a transfer set, $\lambda_{G\setminus F}(u')$ is incomparable to $\lambda_{G\setminus F}(u'')$, and by Lemma 13a, we have that $\lambda_{G\setminus F'}(u') \leq \lambda_{G\setminus F}(u')$ and $\lambda_{G\setminus F'}(u'') \leq \lambda_{G\setminus F}(u'')$. Therefore, $\lambda_{G\setminus F'}(u')$ is incomparable to $\lambda_{G\setminus F'}(u'')$. Now consider the edge (u, w). By the definition of an s-move, $\lambda_{G\setminus F}(w)$ is incomparable to $\lambda_{G\setminus F}(v)$, so that $\lambda_{G\setminus F'}(w)$ is incomparable to $\lambda_{G\setminus F'}(v)$. Since $\lambda_{G\setminus F'}(u) = \lambda_{G\setminus F'}(v)$, we see that $\lambda_{G\setminus F'}(u)$ is incomparable to $\lambda_{G\setminus F'}(w)$. Hence, F' is a transfer set and (D, F') is a candidate.

We can now see that, in all cases, only candidates are inserted into Q during the while-loop.

■

All that remains is for us to show that the search we are performing really will find all optimal candidates. The proof of the next Lemma contains a detailed argument that shows that the order in which we consider the moves is not important.

Lemma 17: Each optimal final candidate for S, G, and σ will be inserted into Q at some point during the execution of Algorithm 2.

Proof: Let C^* be an optimal candidate for S, G, and σ. If $|C^*| = 0$, then C^* is inserted into Q on line 2. Assume that $|C^*| > 0$. Consider one execution of Algorithm 2 and define a sequence of candidates $C_0, C_1, \ldots, C_m$, with $C_i = (D_i, F_i)$, as follows:

$$C_0 = (\emptyset, \emptyset).$$

If C_i contains an s-move, then let u with children v, w be the s-move chosen on line 6 and define C_{i+1} as:

$$C_{i+1} = \begin{cases} (D_i, F_i \cup \{(u,v)\}) & \text{if } (u,v) \in F^*, \\ (D_i, F_i \cup \{(u,w)\}) & \text{if } (u,w) \in F^*, \\ (D_i \cup \{p_{G \setminus F}(u)\}) & \text{otherwise.} \end{cases}$$

From Lemma 15, it is clear that $C_0, \ldots, C_m$ are all well defined, $C_0 \preceq C_1 \preceq \cdots \preceq C_m \preceq C^*$, and that C_i contains an s-move for $i = 0, 1, \ldots, m-1$. Also, it is clear that $C_0, \ldots, C_m$ will all be inserted into Q at some point during the execution of Algorithm 2.

We will now show that $F_m = F^*$. We already know that $F_m \subseteq F^*$. It remains for us to show that $F^* \subseteq F_m$. Consider the set β of transfer vertices in C^* that are not transfer vertices in C, i.e., $\beta = \{u : (u,v) \in F^* \setminus F_m\}$. Assume $\beta \neq \emptyset$. By Lemma 13b and d, β contains an anchor w.r.t. C_m. Now, let $u \in \beta$ be an anchor w.r.t. C_m that is $<_S$-maximally placed in C_m, by which we mean that if $u' \in \beta$ is an anchor w.r.t. C_m, then $\lambda_{G \setminus F_m}(u')$ is not a proper ancestor of $\lambda_{G \setminus F_m}(u)$. Let v, w be the children of u and assume, without loss of generality, that $(u,v) \in F^*$.

Case 1. u is a root in the forest $G \setminus F_m$. C^* contains no moves, and clearly, neither will the candidate $(D^*, F^* \setminus \{(u,v)\})$ contradicting the optimality of C^*.

Case 2. u is not a root in $G \setminus F_m$ and $\lambda_{G \setminus F_m}(p_{G \setminus F_m}(u)) = \lambda_{G \setminus F_m}(u)$. Since C_m contains no s-moves, $p_{G \setminus F_m}(u) \in D_m$. By the definition of $C_0, \ldots, C_m$, there is a C_i, $i < m$, such that u is an s-move in C_i and $p_{G \setminus F_i}(u) = p_{G \setminus F_m}(u)$. Again, by definition, $(u,v) \notin F^*$, which is a contradiction.

Case 3. u is not a root in $G \setminus F_m$ and $\lambda_{G \setminus F_m}(p_{G \setminus F_m}(u)) \neq \lambda_{G \setminus F_m}(u)$. Let $p = p_{G \setminus F_m}(u)$. Clearly, $\lambda_{G \setminus F_m}(p)$ is a proper ancestor of $\lambda_{G \setminus F_m}(u)$. By the definition of u, any anchor u' w.r.t. C_m such that $\lambda_{G \setminus F_m}(u') = \lambda_{G \setminus F_m}(p)$ is not a transfer vertex in C^*. By Lemma 13d, $\lambda_{G \setminus F^*}(p) = \lambda_{G \setminus F_m}(p)$. It is now clear that since C^* contains no moves, then $(D^*, F^* \setminus (u,v))$ contains no moves, contradicting the optimality of C^*.

In all cases, our assumption that $\beta \neq \emptyset$ leads to a contradiction. Hence, $F_m = F^*$.

If $D_m = D^*$, then $C_m = C^*$ and we are done. Assume instead that $D_m \subset D^*$. Define C_{m+i} for $i = 1, \ldots, n$ as follows. If C_{m+i} contains a d-move, then let u be the d-move chosen on line 12 and define C_{m+i+1} as

$$C_{m+i+1} = (D_{m+i} \cup \{u\}, F_{m+i}).$$

If C_{m+i} contains no d-move, then $i = n$.

Clearly, $C_m \prec C^*$. Assume that $C_{m+i} \prec C^*$. Since C_{m+i} contains no s-moves and is not final, it contains a d-move u. If $u \notin D^*$, then u is a d-move of C^*. Therefore, $u \in D^*$. Hence, $C_{m+i+1} \preceq C^*$. By induction, $C_{m+n} \preceq C^*$. But C_{m+n} contains no s-move or d-move and is therefore final. Hence, $C_{m+n} = C^*$. ■

Finally, we prove the time complexity of the algorithm. Note that the proof contains detailed descriptions of how the search for s-moves and d-moves can be implemented to achieve the time complexity.

Theorem 3: Given S, G, and σ, Algorithm 2 returns the set of optimal candidates and can be implemented to run in time $O(m + n \cdot 3^c)$, where $m = |V(S)|$, $n = |V(G)|$, and c is the minimum cost of any final candidate for S, G, and σ.

Proof: The correctness of the algorithm is an easy consequence of Lemma 16 and Lemma 17. We will now proceed to show how the algorithm can be implemented to run in time $O(m + n \cdot 3^c)$.

By doing a one-time precomputation in time $O(m)$, it is possible to compute $\mathrm{lca}\,\{x, y\}$ for any pair of vertices $x, y \in V(S)$ in time $O(1)$. A clear exposition of how this can be done can be found in [37].

The while-loop of the algorithm is executed at most $O(3^c)$ times. To see this, note that each candidate C that is dequeued on line 4 is replaced by at most three other candidates, each having a cost of $|C| + 1$. No candidate with cost exceeding $c + 1$ is ever inserted into Q. Hence the while-loop is executed at most 3^{c+1} times.

It only remains for us to show that every operation within the while-loop can be performed in time $O(n)$. More specifically, we have to show how to find an s-move or d-move in time $O(n)$.

We will assume that we can determine whether an arbitrary gene tree vertex u is unmarked in constant time. First, we will precompute $\lambda_{G \setminus F}(u)$ for all vertices of G. Using the recursion in Lemma 2, we see that this can be done in time $O(n)$.

Next we precompute $p_{G \setminus F}(u)$. This can also be done in time $O(n)$ using the following recursion:

$$P[u] = \begin{cases} \emptyset & \text{if } u = \text{root}(G) \text{ or } (p_G(u), u) \in F, \\ P[p_G(u)] & \text{if } (p_G(u), u') \in F,\ u' \text{ sibling of } u, \\ p_G(u) & \text{otherwise.} \end{cases}$$

Clearly, if $u \in G \setminus F$, then $P[u] = p_{G \setminus F}(u)$.

Now, a gene tree vertex u with children v, w is an s-move iff u is unmarked, $p_{G \setminus F}(u)$ is unmarked, $\lambda_{G \setminus F}(u) \neq \lambda_{G \setminus F}(v)$, $\lambda_{G \setminus F}(u) \neq \lambda_{G \setminus F}(w)$, and $\lambda_{G \setminus F}(p_{G \setminus F}(u)) = \lambda_{G \setminus F}(u)$. To check these conditions for each internal gene tree vertex takes time $O(n)$.

To find a d-move we simply check, for each vertex $u \in D \cup L(G)$, whether $p_{G \setminus F}(u)$ is unmarked and $\lambda_{G \setminus F}(p_{G \setminus F}(u)) = \lambda_{G \setminus F}(u)$. If so, $p_{G \setminus F}(u)$ is a d-move. Clearly, this can also be done in time $O(n)$.

Hence, Algorithm 2 can be implemented to run in time $O(m + n \cdot 3^c)$. ■

VIII. A Note on Cycles and Gene Losses

As mentioned earlier, the interpretation of a transfer edge (u, v) in a scenario is that a lateral transfer has occurred from the incoming edge of $\gamma(u)$ to some ancestral edge of $\gamma(v)$ that is not ancestral to $\gamma(u)$. So, although the starting point of each transfer in the species tree is made explicit in a scenario, the end point is not; the end point could be anywhere between $\gamma(v)$ and $\text{lca}\{\gamma(v), \gamma(u)\}$. This is in contrast to earlier work [32], where both the start and end points of lateral transfer events were made explicit. As such, the notion of cyclic scenarios is somewhat different.

In [32], a path was defined as a sequence $x_1, x_2, \ldots, x_n$ of species tree vertices where, for each pair of consecutive vertices x_i and x_{i+1}, either (x_i, x_{i+1}) is an edge of the species tree, or x_i and x_{i+1} represent the edges between which a lateral transfer event has occurred. In the latter case, the transfer could be in any direction, i.e., either from the incoming edge of x_i to the incoming edge of x_{i+1}, or vice versa. In this way, whenever we move from one vertex to another in a path we never move backward in time. A cycle is a path that starts and ends in the same vertex, and if a scenario contains a cycle, then the scenario is biologically infeasible. With our new definitions, a cyclic DTL-scenario would contain a cycle no matter where the end points of lateral transfer events were placed. We note here, that as pointed out in [32], cycles seem not to be a problem in practice.

Note, however, that the issue of transfer end points is the only essential difference between the old and new definitions. In fact, for each DTL-scenario we can find a corresponding set of old-style scenarios by explicitly choosing all possible end points for each transfer in the DTL-scenario. Hence, the set of most parsimonious scenarios are essentially the same irrespective of which definition we use.

We mentioned in section IV that there are many ways a gene tree could be mapped into a species tree, even when keeping the set of events, i.e., the sets Σ, Δ, and Ξ, fixed. In our present context, we do not view the exact placement of the gene tree vertices within the species tree as an important question. Such questions are better answered by, for example, using sequence data. However, a more interesting question is whether or not every scenario with a fixed set of events is cyclic. In terms of the definitions in section VII, the question can be phrased: Given a final candidate (D, F) for S, G, and σ, is there an acyclic scenario for S, G, and σ such that $\Delta = D$ and $\Xi = F$?

The name of DTL-scenarios comes from the fact that we are using three biological events—duplications, lateral transfers, and gene losses—to explain the differences between species trees and gene trees. In the algorithms we have presented, however, we optimize the number of duplications and transfers only. There are several reasons for this. First, gene loss occurs frequently during evolution of genomes, where many duplications are followed by gene loss. Secondly, minimizing the number of losses can lead to the introduction of unnecessary and artificial transfer events. A duplication near the root of the species tree, for example, can lead to many losses further down the tree. However, a transfer could instead move the gene tree bifurcation closer to the leaves of the species tree, thereby eliminating losses. Instead, we propose to use the number of losses as a second criteria to choose between the different most parsimonious scenarios. A conservative approach for the detection of lateral transfer events would be to choose among the most parsimonious scenarios the ones with the fewest transfers, and among these, the ones with the fewest number of losses.

IX. Empirical Performance

In this section we will analyze a biological dataset using the algorithms described previously. We will see how to handle some of the difficulties that may arise in dealing with real data, such as dealing with unrooted gene trees, and how to overcome them. Our results are comparable to other analyses done on the same data set, but we have the added advantage of being able to also take into account the role of duplications.

In [38], Matte-Tailliez *et al.* constructed phylogenies for 14 archaeal species: one based on the concatenation of 53 ribosomal proteins (7175 positions), which we will call S_{RP}, and one based on the concatenation of SSU and LSU rRNA (3933 positions), which we will call S_{rRNA}. These two species trees are shown in Fig. 6. In the same article, the authors also analyzed the impact of LGTs in their dataset and concluded that 8 genes may have undergone lateral gene transfer events. One of these is the *rpl12e* ribosomal protein, which we will study in this section. The same dataset was also analyzed by Jin *et al.* in [39].

The aligned rpl12e sequences were generously provided by Hervé Philippe. We used MrBayes [40] to obtain the ML gene tree shown in Fig. 7 (the tree with maximum posterior likelihood and the consensus tree were the same), which is identical to the one presented in [38] and has high edge posterior probabilities. We were not able to find any outgroup sequences that aligned well with the archaeal sequences, so instead of using an outgroup, we rooted the gene tree in all possible ways, thereby obtaining 25 candidate rooted gene tree $G_1, \ldots, G_{25}$.

Reconciling any of the rooted gene trees with S_{RP} without using transfers, i.e., according to the duplication loss model, requires at least 7 duplications and 27 losses. For S_{rRNA}, the numbers are 6 duplications and 25 losses. We analyzed each pair of rooted gene tree and species tree using our algorithms. Note that the gene tree in Fig. 7 nicely groups the Crenarchaeota (*S. solfataricus, A. pernix,* and *P. aerophilum*), but internally, this clade is in conflict with each of the species trees in Fig. 6. This conflict can be reconciled using either one duplication and 3 losses or just one transfer. We will take the conservative approach here and use a duplication to explain the difference. A similar remark applies to the *Pyrococcus* clade which is different from that of S_{RP}.

We examined each of the scenarios obtained and discarded those that were considered highly unlikely and those with transfers within the Crenarchaeota or the *Pyrococcus* clade; For S_{RP} at least 5 events, i.e., transfers or duplications, were needed. Among the cases with d duplications and t transfers, satisfying $d + t = 5$, we kept only those with a minimum number of losses. One gene tree whose root was placed in a very unlikely position, inside the *Pyrococcus* clade, was discarded. A similar analysis was performed for S_{rRNA} where a minimum of 4 events were required. One scenario with 4 LGTs and no duplications turned out to be cyclic and was discarded. The results of the undiscarded scenarios for both species trees are summarized in Table I. The roots of the gene trees in Table I are highlighted in Fig. 7.

TABLE I

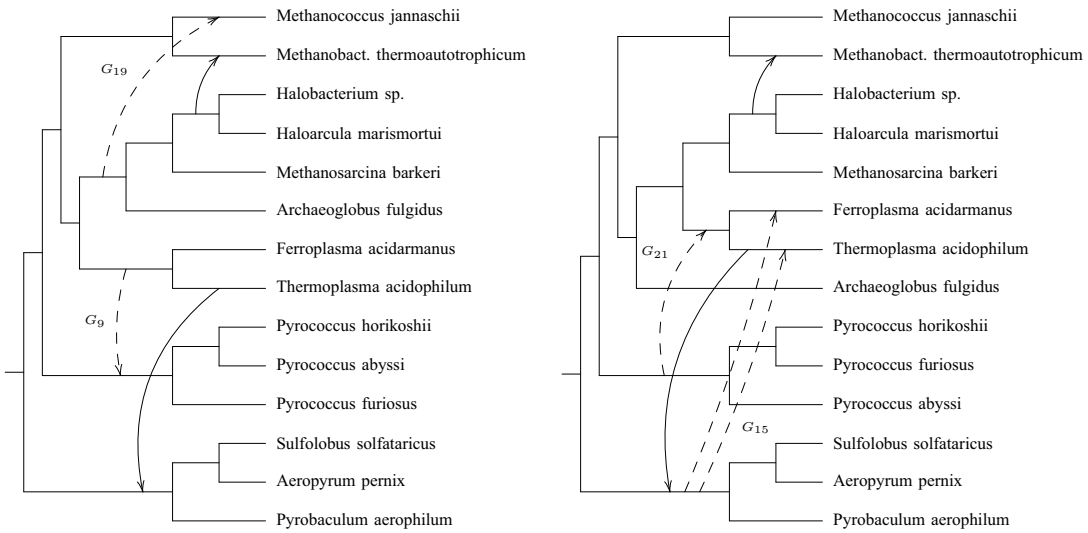
SUMMARY OF THE DTL-SCENARIOS RECONCILING THE ROOTED GENE TREES WITH THE TWO SPECIES TREES. THE COLUMNS D, T, AND, L CORRESPOND TO THE NUMBER OF DUPLICATIONS, TRANSFERS, AND LOSSES RESPECTIVELY.

S_{RP}				S_{rRNA}			
Gene Tree	D	T	L	Gene Tree	D	T	L
G_{17}	4	1	16	G_{17}	3	1	15
G_{19}	3	2	10	G_{19}	2	2	8
G_9	2	3	7	G_{15}	1	3	5
G_{19}	2	3	7	G_{21}	1	3	5

For each pair of gene tree and species tree in Table I, we examined all most parsimonious scenarios and looked for common features. For S_{RP}, every most parsimonious scenario has a transfer to *Methanobacter thermoautotrophicum*, and every scenario except G_{17} that has only one transfer, has also a transfer to the Crenachaeota. These are indicated with solid arrows in Fig. 6a. A scenario for G_{19} has a transfer from the parent of *Archaeoglobus fulgidus* to *Methanococcus janaschii*, and a scenario for G_9 has a transfer from the parent of *Thermoplasma acidophilum* to the *Pyrococcus* clade. The latter two transfers are indicated with dashed arrows in Fig. 6a.

All scenarios for S_{rRNA} has a transfer to *Methanobacter thermoautotrophicum*, just as for S_{RP}, and all scenarios except for G_{15} and G_{17} (the latter has only one transfer) has a transfer to the Crenarchaeota. The scenario for G_{15} has instead two transfers in the opposite direction: from the least common ancestor of the Crenarchaeota to *Thermoplasma acidophilum* and to *Ferroplasma acidarmanus*. The scenario for G_{21} has a transfer from the *Pyrococcus* clade to the parent of *Thermoplasma acidophilum*. See Fig. 6b.

There are similarities between our analysis of the data and that in [39]: e.g., there is a transfer to *M. thermautotrophicum*, and also a transfer from the Crenarchaeota to the least common ancestor of *T. acidophilium* and *F. acidarmanus* (similar to our scenario for G_{15}). A transfer within the *Pyrococcus* clade is present in [39], but as discussed above, we chose not to examine it further based on the fact that it may be just as easily explained by one duplication. The method given in [39] cannot handle duplications, and so uses transfers to explain any incongruency between the gene evolution model, the species tree, and the sequences. Finally, a transfer is indicated in [39] from the least common ancestor of *T. acidophilium* and *F. acidarmanus* to the Crenarchaeota *A. pernix*. No such transfer is



(a) S_{RP} (b) S_{rRNA}

Fig. 6. The two organismal phylogenies from [38]. Fig 6a is based on 53 ribosomal proteins and 6b is based on SSU and LSU rRNA. See main text for the explanation of the transfer edges indicated in the figure.

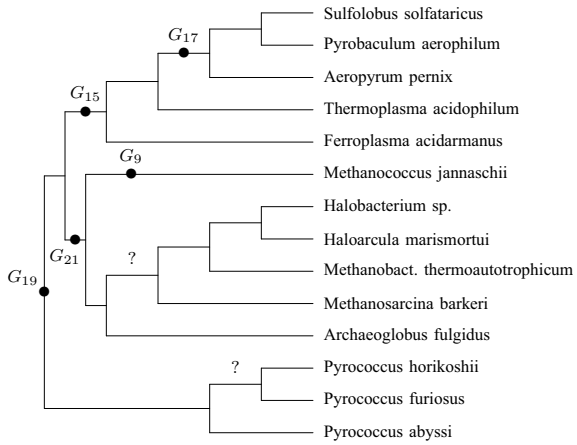


Fig. 7. The gene tree obtained from the archaeal rpl12e sequences. Note that the gene tree is unrooted. The highlighted positions show the roots of the indicated rooted gene trees. The edge posterior probabilities are all above 0.9 except for two edges that are indicated by question marks; the posterior probability of the edge in the *Pyrococcus* clade was 0.69, and for the other edge it was 0.86.

present in our analysis, and in fact, such a transfer contradicts the gene tree in Fig. 7.

Overall, there are significant similarities between the two analyses, although some transfers present in [39] clearly contradict the gene tree. For example, the scenario in [39] groups *A. pernix* together with *T. acidophilum* and *F. acidarmanus*. The posterior probability of such a clade (in the output from MrBayes) is close to zero. Our method also benefits from being able to consider duplications and transfers simultaneously.

X. Open Problems and Discussion

In this paper, we have given a sound and biologically relevant definition of reconciliations between gene trees and species trees and devised algorithms for detecting most parsimonious reconciliations. For reasons that we explained in section VIII, we do not attempt to minimize the number of losses. But as we showed in our empirical tests, the number of losses can be used to choose among the most parsimonious scenarios when more than one exist. Moreover, we have seen that it can be of great use when the root of the gene tree under consideration is hard to determine.

The FPT-algorithm has a potential to be expanded in future work. First, if the minimum cost of reconciling G and S is known, then the algorithm can be easily modified to do a depth-first instead of a breadth-first search, something that will minimize the amount of memory needed. Second, it may be possible to extend the algorithm to search beyond the optimal scenarios and thereby provide the user the ability to search for non-parsimonious solutions if the most parsimonious ones are not satisfactory in light of other biological data.

It is also interesting to consider weighting duplications and LGTs differently. Ideally, such a weighting should reflect the likelihood of each event under a probabilistic model of evolution. An interesting question is whether there is an efficient algorithm for computing the optimal number of duplications and transfers for all weighting schemes simultaneously.

References

[1] M. Lynch and A. Force, "The Probability of Duplicate Gene Preservation by Subfunctionalization," *Genetics*, vol. 154, no. 1, pp. 459–473, 2000.

[2] A. Force, M. Lynch, F. Pickett, A. Amores, Y. Yan, and J. Postlethwait, "Preservation of duplicate genes by complementary, degenerative mutations," *Genetics*, vol. 151, no. 4, pp. 1531–1545, Apr 1999.

[3] M. Lynch and J. Conery, "The evolutionary fate and consequences of duplicate genes," *Science*, vol. 290, no. 5494, pp. 1151–1155, Nov 2000.

[4] ——, "The evolutionary demography of duplicate genes," *J Struct Funct Genomics*, vol. 3, no. 1-4, pp. 35–44, 2003.

[5] M. Hahn, T. De Bie, J. Stajich, C. Nguyen, and N. Cristianini, "Estimating the tempo and mode of gene family evolution from comparative genomic data," *Genome Res*, vol. 15, no. 8, pp. 1153–1160, Aug 2005.

[6] J. Demuth, T. De Bie, J. Stajich, N. Cristianini, and M. Hahn, "The evolution of mammalian gene families," *PLoS ONE*, vol. 1, p. e85, 2006.

[7] J. Cotton and R. Page, "Rates and patterns of gene duplication and loss in the human genome," *Proc Biol Sci*, vol. 272, no. 1560, pp. 277–283, Feb 2005.

[8] J. Lawrence, "Horizontal and vertical gene transfer: the life history of pathogens," *Contrib Microbiol*, vol. 12, pp. 255–271, 2005.

[9] W. Doolittle and E. Bapteste, "Pattern pluralism and the tree of life hypothesis," *Proc Natl Acad Sci U S A*, vol. 104, no. 7, pp. 2043–2049, Feb 2007.

[10] M. Hallett and J. Lagergren, "Efficient algorithms for lateral gene transfer problems," *Proceedings of the Fifth Annual International Conference on Research in Computational Biology, April 22-25, 2001, Montreal, Canada*, 2001.

[11] E. Bapteste, E. Susko, J. Leigh, D. MacLeod, R. Charlebois, and W. Doolittle, "Do orthologous gene phylogenies really support tree-thinking?" *BMC Evol Biol*, vol. 5, no. 1, p. 33, 2005.

[12] M. Charleston, "Jungles: a new solution to the host/parasite phylogeny reconciliation problem," *Math Biosci*, vol. 149, no. 2, pp. 191–223, May 1998.

[13] L. Nakhleh, D. Ruths, and L. Wang, "Riata-hgt: A fast and accurate heuristic for reconstructing horizontal gene transfer," *Proceedings of the Eleventh International Computing and Combinatorics Conference (COCOON 05)*, pp. 84–93, 2005.

[14] V. Makarenkov and P. Legendre, "From a phylogenetic tree to a reticulated network," *J Comput Biol*, vol. 11, no. 1, pp. 195–212, 2004.

[15] E. Lerat, V. Daubin, H. Ochman, and N. Moran, "Evolutionary origins of genomic repertoires in bacteria," *PLoS Biol*, vol. 3, no. 5, p. e130, May 2005.

[16] D. Huson, *Split Networks and Reticulate Networks*. Oxford University Press, 2007, pp. 247–276.

[17] R. Azad and J. Lawrence, "Detecting laterally transferred genes: use of entropic clustering methods and genome position," *Nucleic Acids Res*, vol. 35, no. 14, pp. 4629–4639, 2007.

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

[19] R. Guigó, I. Muchnik, and T. Smith, "Reconstruction of ancient molecular phylogeny," *Mol Phylogenet Evol*, vol. 6, no. 2, pp. 189–213, Oct 1996.

[20] B. Ma, M. Li, and L. Zhang, "From gene trees to species trees," *SIAM Journal on Computing*, vol. 30, no. 3, pp. 729–752, 2000.

[21] M. Hallett and J. Lagergren, "New algorithms for the duplication-loss model," *Proceedings of the fourth annual international conference on Research in Computational Molecular Biology*, pp. 138–146, 2000.

[22] I. Wapinski, A. Pfeffer, N. Friedman, and A. Regev, "Natural history and evolutionary principles of gene duplication in fungi," *Nature*, vol. 449, no. 7158, pp. 54–61, Sep 2007.

[23] L. Arvestad, A. Berglund, J. Lagergren, and B. Sennblad, "Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution," *Proceedings of the eighth annual international conference on Research in Computational Molecular Biology*, pp. 326–335, 2004.

[24] L. Arvestad, J. Lagergren, and B. Sennblad, "The gene evolution model and computing its associated probabilities," *J ACM*, vol. 56, no. 2, pp. 1–44, 2009.

[25] B. Sennblad and J. Lagergren, "Probabilistic orthology analysis," *submitted*, 2008.

[26] Ö. Åkerborg, B. Sennblad, L. Arvestad, and J. Lagergren, "Simultaneous bayesian gene tree reconstruction and reconciliation analysis," *Proc Natl Acad Sci U S A*, vol. 106, no. 14, pp. 5714–5719, Apr 2009.

[27] J. Lawrence and H. Ochman, "Molecular archaeology of the Escherichia coli genome," *Proc Natl Acad Sci U S A*, vol. 95, no. 16, pp. 9413–9417, Aug 1998.

[28] D. Gevers, K. Vandepoele, C. Simillon, and Y. Van de Peer, "Gene duplication and biased functional retention of paralogs in bacterial genomes," *Trends Microbiol*, vol. 12, no. 4, pp. 148–154, Apr 2004.

[29] A. Retchless and J. Lawrence, "Temporal fragmentation of speciation in bacteria," *Science*, vol. 317, no. 5841, pp. 1093–1096, Aug 2007.

[30] M. Csűrös and I. Miklós, "A probabilistic model for gene content evolution with duplication, loss and horizontal transfer," in *In Tenth Annual International Conference on Research in Computational Molecular Biology (RECOMB)*. Springer, 2006, pp. 206–220.

[31] P. Górecki, "Reconciliation problems for duplication, loss and horizontal gene transfer," in *RECOMB '04: Proceedings of the eighth annual international conference on Resaerch in computational molecular biology*. New York, NY, USA: ACM, 2004, pp. 316–325.

[32] M. Hallett, J. Lagergren, and A. Tofigh, "Simultaneous identification of duplications and lateral transfers," *Proceedings of the eighth annual international conference on Research in Computational Molecular Biology*, pp. 347–356, 2004.

[33] L. Addario-Berry, M. Hallett, and J. Lagergren, "Towards identifying lateral gene transfer events," *Proc 8th Pacific Symp on Biocomputing (PSB03)*, pp. 279–290, 2003.

[34] R. D. M. Page, "Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas," *Systematic Biology*, vol. 43, no. 1, pp. 58–77, 1994.

[35] B. Mirkin, I. Muchnik, and T. Smith, "A biologically consistent model for comparing molecular phylogenies." *J Comput Biol*, vol. 2, no. 4, p. 493, 1995.

[36] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1979.

[37] M. A. Bender, M. Farach-Colton, G. Pemmasani, S. Skiena, and P. Sumazin, "Lowest common ancestors in trees and directed acyclic graphs," *J Algorithms*, vol. 57, no. 2, pp. 75–94, 2005.

[38] O. Matte-Tailliez, C. Brochier, P. Forterre, and H. Philippe, "Archaeal phylogeny based on ribosomal proteins," *Mol Biol Evol*, vol. 19, no. 5, pp. 631–639, May 2002.

[39] G. Jin, L. Nakhleh, S. Snir, and T. Tuller, "Maximum likelihood of phylogenetic networks," *Bioinformatics*, vol. 22, no. 21, pp. 2604–2611, Nov 2006.

[40] F. Ronquist and J. Huelsenbeck, "MrBayes 3: Bayesian phylogenetic inference under mixed models," *Bioinformatics*, vol. 19, no. 12, pp. 1572–1574, Aug 2003.