Fast algorithm for the reconciliation of gene trees and LGT networks

Celine Scornavacca, Joan Carles Pons Mayol, Gabriel Cardona

Fast algorithm for the reconciliation of gene trees and LGT networks

Celine Scornavacca[1]

Institut des Sciences de l'Evolution, Université de Montpellier, CNRS, IRD, EPHE
34095 Montpellier Cedex 5 - France
Celine.Scornavacca@umontpellier.fr

Joan Carles Pons Mayol, Gabriel Cardona

Department of Mathematics and Computer Science,
University of the Balearic Islands
E-07122 Palma - Spain
{joancarles.pons, gabriel.cardona}@uib.es

Abstract

In phylogenomics, *reconciliations* aim at explaining the discrepancies between the evolutionary histories of genes and species. Several reconciliation models are available when the evolution of the species of interest is modelled via phylogenetic trees; the most commonly used are the $\mathbb{DL}$ model, accounting for duplications and losses in gene evolution and yielding polynomially-solvable problems, and the $\mathbb{DTL}$ model, which also accounts for gene transfers and implies NP-hard problems. However, when dealing with non-tree-like evolutionary events such as hybridisations, phylogenetic networks – and not phylogenetic trees – should be used to model species evolution. Reconciliation models involving phylogenetic networks are still at their early days.

In this paper, we propose a new reconciliation model in which the evolution of species is modelled by a special kind of phylogenetic networks – the *LGT networks*. Our model considers duplications, losses and transfers of genes, but restricts transfers to happen through some specific arcs of the network, called secondary arcs. Moreover, we provide a polynomial algorithm to compute the most parsimonious reconciliation between a gene tree and an LGT network under this model. Our method, when combined with quartet decomposition methods to detect putative "highways" of transfers, permits to refine their analyses by allowing to examine the two possible directions of a highway and even consider combinations of highways.

Keywords: Phylogenetic tree, Phylogenetic network, Reconciliation method, Highways of transfers.

---

[1]Corresponding author

1. Introduction

Speciation events constitute only part of the events shaping a gene history, others being, for example, duplications, losses and transfers of genes. Reconciliation methods aim at finding an evolutionary scenario – a *reconciliation* – explaining the discrepancies between gene and species trees that are caused by evolutionary events other than speciations. In the parsimony framework, a cost is associated to each evolutionary event and the cost of an evolutionary scenario is the sum of the costs of all events that it implies; the aim here is to obtain a most parsimonious reconciliation, i.e. one minimizing the cost over the set of all possible evolutionary scenarios.

Several reconciliation models have been proposed in the last decades when the species evolution is tree-like [1, 2, 3, 4, 5, 6, 7, 8], and the most commonly used are the $\mathbb{DL}$ and the $\mathbb{DTL}$ models. In the former, duplications and losses are the only evolutionary events considered (along with speciations, of course), and a most parsimonious reconciliation can be found in linear time [9]. In the latter, transfers are also considered, and finding a most parsimonious reconciliation becomes NP-hard [10, 4], the crux lying in the difficulty of ensuring that transfers happen only between co-existing species. To palliate this problem, two alternative models have been suggested that lead to polynomially-solvable problems: either to use a dated species tree [11, 12, 1, 3], or to use a priori information to fix which pairs of branches are allowed to be involved in a transfer event (a representation called lateral transfer scheme in [10] and species graph in [13]).

When the species history involves a significant amount of reticulate events such as hybridisation or reassortment, phylogenetic trees are less suited to model species evolution, and phylogenetic networks should be used instead [14, 15]. For now, little work has been done on the problem of finding parsimonious reconciliations between gene trees and species networks. In [16], the authors presented an extension of a model very closely related to the $\mathbb{DTL}$ model, namely the cophylogeny model [17, 18, 19, 20], on dated phylogenetic networks, taking into account codivergence and host switching (respectively, speciation and transfer in the $\mathbb{DTL}$ jargon), along with duplication and loss events.

Motivated by the high time complexity of the solution proposed in [16] to solve the $\mathbb{DTL}$ model on networks, the authors of [21] discarded transfers and proposed an extension of the $\mathbb{DL}$ model, which can be solved faster. (The authors of [21] also provided a model seeking an optimum reconciliation between a gene tree and a *tree displayed by the species network*; this latter concept will be detailed in due course).

Here, we extend the models of [21] by allowing transfer events via some particular arcs of the network – in a similar flavour of what is proposed in [13] – we will see that our models and algorithms are more general and faster than the ones presented in [13]. To do so, we will make use of LGT networks [22], networks in which some arcs model transfer events and the remaining ones model descend with modification.

The main application of the models presented here is the estimation of species networks: if the LGT network is not known but highways of trans-

fers are suspected for the data, our reconciliation models can be used to score candidate LGT networks. Combining this with local search techniques such as hill climbing leads to an LGT networks estimation framework of a similar flavour to what was done in PHYLDOG [23] to estimate a species tree from a set of genes.

2. Basic notations

In this paper we focus on binary rooted phylogenetic networks, *networks* for short, which are directed acyclic graphs (DAGs) that have a single indegree-0 node (the *root*), outdegree-0 nodes (the *leaves*) that are labelled, and internal nodes with either indegree-1 and outdegree-2 (*principal* nodes) or indegree-2 and outdegree-1 (*secondary* nodes or *reticulations*). Binary rooted phylogenetic trees, *trees* for short, are binary rooted phylogenetic networks with no indegree-2 outdegree-1 nodes.

Given a network N, we denote its root by $r(N)$ and the set of its nodes, internal nodes, arcs, leaves and labels respectively by $V(N)$, $I(N)$, $E(N)$, $L(N)$ and $\mathcal{L}(N)$. Each leaf is labeled via a function $l : L(N) \to \mathcal{L}(N)$; we will omit l in the following, and we will refer to N rather than (N, l). An internal node u of N has one child (u_l) or two interchangeable children (u_l, u_r). The topology of N induces a partial order on its nodes. Given two nodes u and v of N, u is said to be a descendant of v (denoted as $u \leqslant_N v$) if, and only if, there exists a path in N going from v to u. By extension, u is said to be a proper descendant of v if, and only if, $u \leqslant_N v$ and $u \neq v$. For a node u of N, we denote N_u the subgraph induced by the nodes accessible from u. A *cut node* or *cut arc* is a node or arc, respectively, whose removal disconnects the graph. A *biconnected component* is a maximal connected subgraph that is induced by a set of arcs and does not contain a cut node. If every biconnected component of N has at most k reticulation nodes, we say that N is of *level-k* [24].

Given a network N, a *switching* S of N is obtained from N by choosing, for each reticulation, an incoming arc to *switch on* and the others to *switch off*. Once this is done, we also recursively switch off all switched-on arcs whose target node has only switched-off outgoing arcs. We denote by $V_{on}(S)$ the set of nodes of a switching S that are not an endpoint (i.e. source or target) of any switched-off arc. A tree T is said to *be displayed* by a network if T is isomorphic to the tree obtained by using only the switched-on arcs of a switching S of N and by contracting all nodes not in $V_{on}(S)$[2]. A *path* of S is a path of N that uses only switched-on arcs.

A *species network* N is a network such that each element of $\mathcal{L}(N)$ represents an extant species. A *gene tree* G is a tree such that each element of $\mathcal{L}(G)$ represents a contemporary gene and each leaf u is associated to a species, denoted

[2] *Contracting* an indegree-1 outdegree-1 node u consists in first adding the arc (u_p, u_l) then removing (u_p, u), (u, u_l) and u, where u_p is the parent node of u. Contracting a degree-1 node consists in deleting it.

$s(u)$. From now on, we consider a species network N and a gene tree G such that $\{s(u)|u \in L(G)\} \subseteq \mathcal{L}(N)$.

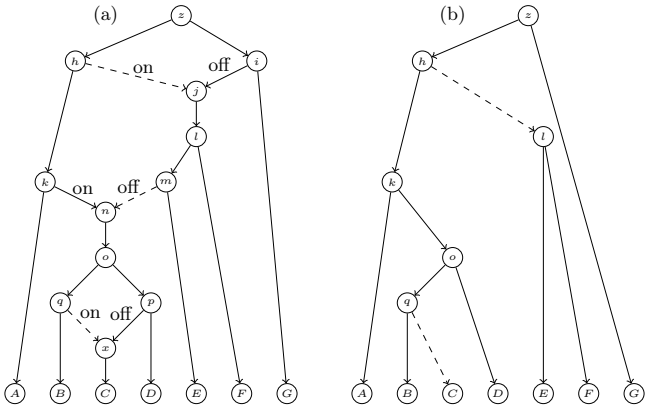

Figure 1: (a) An example of a switching for a level-2 LGT network, and (b) the tree displayed by this switching. The solid lines represent the principal arcs and the dashed ones represent the secondary arcs. The labels "on" and "off" on the arcs in (a) represent the switched-on and switched-off arcs, respectively. Note that the LGT network depicted in (a) is time-consistent.

An *LGT network* N [22] is a species network along with a partition of $E(N)$ in a set of *principal arcs* E_p and a set of *secondary arcs* E_s, such that $T_0(N) = (V, E_p)$ is a tree, once that all indegree-1 outdegree-1 nodes have been contracted. It is easy to see that this implies that each internal node of N has at least one principal node as child. Note that LGT networks are *tree-based networks*, where $T_0(N)$ is a *distinguished base tree* [25]. Note also that our definition of LGT network differs slightly from the one given in [22] because here we consider only binary networks. A *species tree* is an LGT network with $E_s = \emptyset$. We say that an LGT network N is *time-consistent* if there is a function $t : V(N) \to \mathbb{N}$ such that:

- $t(u) = t(v)$, if $(u, v) \in E_s$, and
- $t(u) < t(v)$, if $(u, v) \in E_p$.

Informally, $t(u)$ is a time-stamp of the evolutionary event associated to the node u that increases between speciation events and remains constant for transfer events. A similar definition of time-consistency is used in [13]. In the following, we will only consider time-consistent LGT networks. See Figures 1 and 2 for an illustration of several concepts introduced in this section.

3. Former reconciliation models

The aim of this paper is to extend the models presented in [21] to model transfer events via secondary arcs of an LGT network. We start by giving an

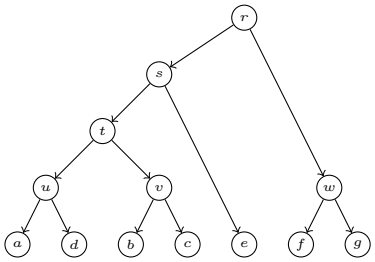

Figure 2: A gene tree.

overlook of various reconciliation models present in the literature.

3.1. $\mathbb{DL}$/$\mathbb{DTL}$ reconciliations between a species tree and a gene tree

Recalling the definition of a $\mathbb{DL}$/$\mathbb{DTL}$ reconciliation between a species tree and a gene tree will be very useful in the next section.

Given a species tree S and a gene tree G, a reconciliation α is a function that maps each node u of G onto an ordered sequence of nodes of S, denoted $\alpha(u) = (\alpha_1(u), \alpha_2(u), \ldots, \alpha_\ell(u))$. A reconciliation depicts an evolutionary history for a gene family with a given gene tree, evolving within a given species tree. Possible mappings are restricted by conditions that aim at having an evolutionary history coherent with the chosen evolutionary model. A $\mathbb{DTL}$ reconciliation is a reconciliation for the evolutionary model considering speciation, duplication, loss and horizontal gene transfer as possible evolutionary events for genes (for more information of this model, see [26, 27, 28], among others). More formally:

Definition 1. *Given a species tree S and a gene tree G, α is said to be a $\mathbb{DTL}$ reconciliation between G and S if and only if exactly one of the following events occurs for each pair of nodes u of G and $\alpha_i(u)$ of S (for simplicity, let $x := \alpha_i(u)$ below):*

a) if x is the last node of $\alpha(u)$, one of the cases below is true:

1. *$u \in L(G)$, $x \in L(S)$ and $s(x) = s(u)$;* *(extant leaf)*
2. *$\{\alpha_1(u_l), \alpha_1(u_r)\} = \{x_l, x_r\}$;* ($\mathbb{S}$)
3. *$\alpha_1(u_l) = x$ and $\alpha_1(u_r) = x$;* ($\mathbb{D}$)
4. *$\alpha_1(u_l) = x$, and $\alpha_1(u_r)$ is any species node that is not a descendant or ancestor of x (or symmetrically interchanging the roles of u_l and u_r);* ($\mathbb{T}$)

b) otherwise, one of the cases below is true:

5. *$\alpha_{i+1}(u) \in \{x_l, x_r\}$;* ($\mathbb{SL}$)

6. $\alpha_{i+1}(u)$ is any node that is not a descendant or ancestor of x; ($\mathbb{TL}$)

Speciation ($\mathbb{S}$) and duplication ($\mathbb{D}$) events are self-explanatory. A speciation-loss ($\mathbb{SL}$) is a speciation where the original gene is absent from one of the two species resulting from the speciation. A transfer ($\mathbb{T}$) corresponds to transferring the lineage of a child of a gene to another branch of the species tree, while the sibling lineage still evolves within the lineage of the parent. A transfer-loss ($\mathbb{TL}$) is a transfer of one of the two descendants of a gene combined with the loss of its sibling lineage. Note that each loss is coupled with either a speciation or a transfer. Indeed, duplication-loss events – never parsimonious and leaving no trace in the data – are not taken into account in the model. α is said to be time-consistent if all $\mathbb{T}$ events can be guaranteed to happen between co-existing species. A $\mathbb{DL}$ reconciliation is a $\mathbb{DTL}$ one where $\mathbb{T}$ and $\mathbb{TL}$ events are not allowed[3].

Given costs δ, τ and λ for respectively $\mathbb{D}$, $\mathbb{T}$ and $\mathbb{L}$ events, the cost of a reconciliation α is defined as the sum of the costs of all events it implies. The most parsimonious reconciliation between S and G is one with minimum cost over all possible time-consistent reconciliations. We denote the minimum cost as $cost(G, S)$.

3.2. $\mathbb{DL}$ reconciliations between a species network and a gene tree

In [21], reconciliations between gene trees and species trees are generalised to consider species networks instead of trees. In this setting, reticulations in the species network represent hybridisations while principal nodes represent speciations; the evolutionary events peculiar to genes are duplications and losses. In this setting, the authors focus on two different problems. First, they look for the switching of the network – and consequently, the tree displayed by the network – having a most parsimonious reconciliation with the given gene tree according to the $\mathbb{DL}$ model [2]. Second, they extend the $\mathbb{DL}$ model to species networks and they seek a most parsimonious reconciliation between the given species network and the given gene tree. The first model is more adapted for ancient duplication, for which often only one copy of the gene has been retained, while the latter is more suitable for recent duplications.

4. $\mathbb{DTL}$ reconciliations of gene trees and LGT networks

The models presented in [21] are not adapted to model transfer events via secondary arcs. Indeed, in these models, there is an intrinsic symmetry in the species network: each gene of a hybrid species is inherited from one of the two parental species with the same probability and at no cost, and the evolutionary events peculiar to genes are duplications and losses. On the other hand, in the models that will be presented in this section, principal arcs are used

[3]This is not the common definition of a $\mathbb{DL}$ reconciliation but our definition is equivalent [28] to the more widespread one, which is used for example in [21].

to represent descent with modification while secondary arcs represent possible transfers: the evolutionary events peculiar to genes are transfers (via secondary arcs), duplications and losses. Thus, there is an intrinsic asymmetry in the species network: genes normally evolve by means of vertical descent but they can also take highways of transfers to move "horizontally"; in this case, they have to "pay a toll".

At a first look, one could think that allowing transfers only along secondary arcs is simply a weakening of the $\mathbb{DTL}$ model described in Definition 1, and that transfers that do not occur along the secondary arcs will result in incorrect reconciliations. But one should not forget that the main application of the models presented here is the estimation of species networks, and thus the estimation of the set of secondary arcs that explain the data best.

As in [21], here we shall focus on two different problems. First, we shall describe how to find the switching of the network – and consequently, the tree displayed by the network – having a most parsimonious reconciliation with the given gene tree according to the $\mathbb{DTL}$ model [3]. Second, we shall extend the $\mathbb{DTL}$ model to species networks where transfers are allowed only on secondary arcs – similarly to what was proposed in [13] – and we shall seek for a most parsimonious reconciliation between the given species network and gene tree. Roughly speaking, the two problems differ in the way they deal with the two incoming arcs of each reticulation: in the first case, all genes of a same gene family are forced to use at most one of the two incoming arcs (since the other is off), while in the second case different genes of a same gene family can use different incoming arcs. This implies that that any solution under the first model is a solution under the second.

4.1. Finding the best tree in the network

In this section, we present an algorithm to find the tree displayed by a network N having a most parsimonious $\mathbb{DTL}$ reconciliation with a given gene tree G. Note that this is different from searching the most parsimonious $\mathbb{DTL}$ reconciliation between the distinguished base tree of N and G, and only allowing transfer along secondary arcs. Indeed, in a tree displayed by a network, for each reticulation only one of the two incoming arcs is kept.

We start by extending the $\mathbb{DTL}$ reconciliation definition given in the previous section for a species tree and a gene tree to the case of a switching S of an LGT network and a gene tree G (remember that here we allow transfers only via secondary arcs):

Definition 2. *Given an LGT network N, a switching S of N and a gene tree G, α is said to be a $\mathbb{DTL}$ reconciliation between G and S if and only if exactly one of the following events occurs for each pair of nodes u of G and $\alpha_i(u)$ of S (for simplicity, let $x := \alpha_i(u)$ below):*

a) if x is the last node of $\alpha(u)$, one of the cases below is true:

1. $u \in L(G)$, $x \in L(S)$ and $s(x) = s(u)$; *(extant leaf)*

2. $\{\alpha_1(u_l), \alpha_1(u_r)\} = \{x_l, x_r\}$ *and both outgoing arcs of* x *are in* E_p *and switched-on;* ($\mathbb{S}$)

3. $\alpha_1(u_l) = x$ *and* $\alpha_1(u_r) = x$*;* ($\mathbb{D}$)

4. $\alpha_1(u_l) = x$*,* $\alpha_1(u_r) = y$ *and* (x, y) *is in* E_s *and switched-on (or symmetrically interchanging the roles of* u_l *and* u_r*);*

or $\alpha_1(u_l) = x_l$*,* $\alpha_1(u_r) = x_r$ *and* (x, x_r) *is in* E_s *and both outgoing arcs of* x *are switched-on (or symmetrically interchanging the roles of* u_l *and* u_r*, and* x_l *and* x_r*, so 4 possibilities);* ($\mathbb{T}$)

b) otherwise, one of the cases below is true:

5. $\alpha_{i+1}(u) = y$*,* (x, y) *is in* E_p *and* x *has two outgoing arcs that are switched-on;* ($\mathbb{SL}$)

6. $\alpha_{i+1}(u) = y$*,* (x, y) *is in* E_s *and* x *has two outgoing arcs that are switched-on;* ($\mathbb{TL}$)

7. $\alpha_{i+1}(u) = y$*,* (x, y) *is in* E_p *and is the only switched-on arc of* x*;* ($\emptyset$)

8. $\alpha_{i+1}(u) = y$*,* (x, y) *is in* E_s *and is the only switched-on arc of* x*;* ($\mathbb{T}$)

In Figure 3, an example of a $\mathbb{DTL}$ reconciliation between a gene tree and a switching is depicted.

Note that Definition 2 is equivalent to Definition 1, apart from the fact that transfers are allowed only on secondary arcs (conditions 4, 6 and 8), and that losses are not counted on nodes not in $V_{on}(S)$ (conditions 7 and 8). This way of counting losses implies that, if transfers are allowed only on secondary arcs, for a switching S and its associated tree S' displayed by the network, we have $cost(G, S) = cost(G, S')$. Thus, to find the optimal tree displayed by the network, we can focus on switchings, as done in [21]:

Problem 1 (Best Switching)**.**

Input: A gene tree G*, an LGT network* N*, the costs* δ*,* τ *and* λ *for respectively* $\mathbb{D}$*,* $\mathbb{T}$ *and* $\mathbb{L}$ *events.*

Output: A switching S *of* N *such that* $cost(G, S)$ *is minimum over all switchings of* N*.*

In the following, we use $\alpha_l(u)$ to denote the last node of $\alpha(u)$. In this article, we focus on reconciliations such that $|\alpha(r(G))| = 1$, since this is the case for all most parsimonious reconciliations. Indeed, since Definition 2 allows to map the root of G to any node of the switching S, any reconciliation α where $|\alpha(r(G))| > 1$ can be changed into another one with the same cost or lower simply by mapping $r(G)$ to the last node of $\alpha(r(G))$. Note that, since both nodes of a secondary arc have the same time-stamp, all lateral gene transfers are *time-consistent*, i.e. are guaranteed to happen between co-existing species.

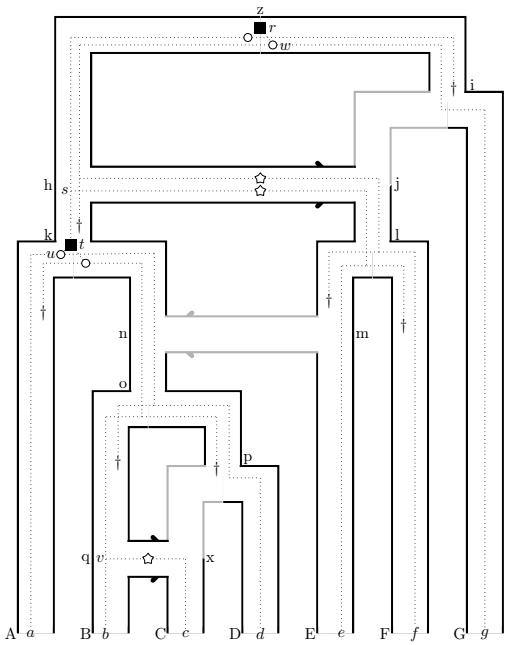

Figure 3: A $\mathbb{DTL}$ reconciliation α between the gene tree depicted in Fig. 2 and the switching depicted in Fig. 1. Solid lines represent network arcs, where secondary arcs are depicted horizontally and with "winglets" and grey ones represent switched-off arcs. The dotted lines represent the gene evolution. The $\mathbb{D}$, $\mathbb{T}$ and $\mathbb{L}$ event are represented as filled square, star and cross dagger, respectively. Speciations are indicated by circles. Each gene label is represented in the figure and mapped to the element $\alpha_l(\cdot)$. As an example, the reconciliation α maps the nodes v and f of G as follows: $\alpha(v) = (k, n, o, q)$, where $\alpha_1(v) = k$, $\alpha_2(v) = n$, $\alpha_3(v) = o$, $\alpha_4(v) = q$ are respectively an $\mathbb{SL}$, a $\emptyset$, an $\mathbb{SL}$ and a $\mathbb{T}$ event; $\alpha(f) = (h, j, l, F)$, where $\alpha_1(f) = h$, $\alpha_2(f) = j$, $\alpha_3(f) = l$, $\alpha_4(f) = F$ are respectively a $\mathbb{TL}$, a $\emptyset$, an $\mathbb{SL}$ and a "exant leaf" event.

In [21], the authors pointed out that, when all arcs are principal, the most parsimonious mapping between S and G is completely determined and the most parsimonious reconciliation is always the LCA reconciliation. We can adapt the usual definition of an LCA reconciliation to our case as follows[4]:

1. for each node $u \in L(G)$, $\alpha_l(u)$ is defined as the only node $x \in L(S)$ such that $s(u) = s(x)$;
2. for each node $u \in I(G)$ with child nodes $\{u_1, u_2\}$, $\alpha_l(u) := LCA_S(\alpha_l(u_1), \alpha_l(u_2))$;
3. once $\alpha_l(u)$ is fixed for all nodes u of G, $\alpha(u)$ is completed as follows. First, we insert in $\alpha(u)$ – before $\alpha_l(u)$ – the ordered list of nodes composing the unique path in S – extremes excluded – between $\alpha_l(u_p)$ and $\alpha_l(u)$, where u_p is the parent node of u. Then, if
   - $\alpha_l(u) \neq \alpha_l(u_p)$,
   - $\alpha_l(u') = \alpha_l(u_p)$ where u' is the sibling of u,
   - $(\alpha_l(u_p), \alpha_1(u))$ is not a secondary arc,

   we insert $\alpha_l(u_p)$ before $\alpha_1(u)$ in $\alpha(u)$.

The LCA reconciliation can be found in $O(|G|)$ time, along with its cost [29, 30]. Now, it is easy to see that, even when some arcs are secondary, once the switched-on/switched-off arcs are fixed, the most parsimonious mapping is still completely determined. (This observation permits us to use several of the results in [21]). We denote the minimum cost over all possible $\mathbb{DTL}$ reconciliations between G and S as $cost(G, S)$, which is thus the cost of the LCA reconciliation.

Note that in [21], the authors solve Problem 1 when all arcs of the network are considered as principal. This implies that all the results and algorithms in the *Best switching* section of [21] are valid to find a valid reconciliation according to Definition 2, but may fail to optimise the cost of transfers. Indeed, since both incoming arcs of a reticulation are considered as principal, in their model of hybridisations there is no difference in taking one rather than the other incoming arc of a reticulation.

In order to optimise the cost of transfers correctly and solve Problem 1, we need to adapt the results of [21] to correctly take into account the "asymmetry" of LGT networks.

We start by recalling some definitions introduced in [21]:

- Given a biconnected component B that is not a leaf of N, the network $N(B)$ consists of B and all cut arcs coming out from B.
- The mapping $B(\cdot)$ associates every $u \in V(G)$ to the lowest (w.r.t. the relation $\leqslant_N$) biconnected component B of N such that $\mathcal{L}(N_{r(B)}) \supseteq \{s(u) | u \in L(G_u)\}$.

[4]Note that in our definition, $\alpha(u)$ for each u in $V(G)$ is exhaustive, while in [21] only the $\alpha_l(u)$ were defined.

- The tree G_N is obtained from G by applying the following procedure for each child u' of each internal node u of G. If there are k biconnected components $B_1, \ldots, B_k$ properly below $B(u)$ and properly above $B(u')$ in N, we add k indegree-1 outdegree-1 nodes on the arc (u, u'), respectively mapped to $B_1, \ldots, B_k$.

- Given a biconnected component B different from a leaf, we denote by G_B the set of all maximal connected subgraphs H of G_N satisfying that $B(u) = B$ for every internal node u of H. Note that these subgraphs are necessarily binary trees or arcs (see Lemma 2 of [21]); we accordingly decompose G_B as $G_B^t \sqcup G_B^e$, where $\sqcup$ denotes the disjoint union.

See Figures 4 and 5 for an illustration of these concepts.

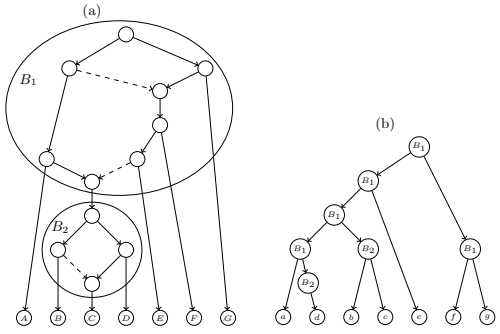

Figure 4: (a) The LGT network depicted in Fig. 1 with the two non trivial biconnected components B_1 and B_2 highlighted. (b) The tree G_N along with the labelling $B(\cdot)$ where the gene tree G is the one depicted in Fig. 2. Notice that an artificial node labelled B_2 is added on the arc (u, d) because $B(u) = B_1$, $B(d) = B_2$ and $B_1 > d$.

We shall see that, in order to adapt the tools proposed in [21] to solve Problem 1, it suffices to redefine the cost function introduced in [21] to correctly count transfers, i.e. secondary arcs used in the reconciliations.

Let α be a reconciliation between G and S, for two nodes x and y in S, $lgt(x, y)$ and $dist(x, y)$ are defined as follows: If $y \leqslant_N x$, $lgt(x, y)$ and $dist(x, y)$ are defined respectively as the number of switched-on secondary arcs, and as the number of nodes in $V_{on}(S)$, in the path from x to y; otherwise, $dist(x, y) = lgt(x, y) = \infty$. We denote $t_\alpha(u)$ the number of transfer events in α associated to an internal node u in G; then we have $t_\alpha(u) = lgt(\alpha_l(u), \alpha_l(u_l)) + lgt(\alpha_l(u), \alpha_l(u_r))$. Consequently, the number of transfers of the reconciliation α, denoted by $t(\alpha)$, is the sum of $t_\alpha(\cdot)$ for all internal nodes of G. We denote by $d(\alpha)$ and $l(\alpha)$ respectively the number of duplications and losses of α.

Note that, as done in [21], we suppose that no internal node of G has all its descendant leaves associated to the same species $\bar{s}$. Indeed, if such a node exists in G, say u, it is easy to see that the most parsimonious way to reconcile G_u is via $(|L(G_u)| - 1)$ duplications. Thus, we can replace G_u by a leaf x such that

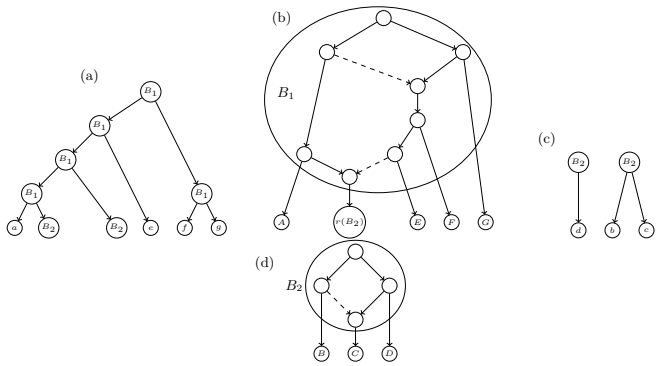

Figure 5: Given the LGT network of Fig. 1 and the gene tree of Fig. 2 we find: (a) G_{B_1}; (b) $N(B_1)$; (c) G_{B_2}; (d) $N(B_2)$.

$s(x) = \bar{s}$, reconcile the resulting tree and, a posteriori, add back G_u, along with all the duplications it implies, to the reconciliation.

Given a biconnected component B_i of N different from a leaf, a switching S_i of $N(B_i)$ and a tree H in $G^t_{B_i}$, we denote by $\beta^H_{S_i}$ the LCA reconciliation between H and S_i where, for each leaf u of H, $s(u) := r(B(u))$. Now, for $H \in G_{B_i}$, we define $cost(H, S_i)$ as follows:

- $\forall H \in G^t_{B_i}$, if $B_i = B(r(G))$, $cost(H, S_i) = cost(\beta^H_{S_i})$,
- $\forall H \in G^t_{B_i}$, if $B_i \neq B(r(G))$, $cost(H, S_i) = cost(\beta^H_{S_i}) + \tau \cdot lgt(r(S_i), \beta^H_{S_i}(r(H))) + \lambda \cdot dist(r(S_i), \beta^H_{S_i}(r(H)))$,
- $\forall H \in G^e_{B_i}$ with u the only leaf of H, $cost(H, S_i) = \tau \cdot lgt(r(S_i), r(B(u))) + \lambda \cdot dist(r(S_i), r(B(u)))$.

Then, the following proposition permits to analyse independently each biconnected component of N and, hence, its cost.

Proposition 1. *Let $B_1, \ldots, B_p$ be the biconnected components of N that are not leaf nodes, and let S be a switching of N. Moreover, for each elementary network $N(B_i)$, let be S_i its switching induced by S.Then, $cost(G, S) = \sum_{i=1}^{p} \sum_{H \in G_{B_i}} cost(H, S_i)$.*

Proof. Let α be a reconciliation between G and S with minimum cost. We denote by $d_\alpha(S_i)$ the number of duplications in S_i and by $l_\alpha(u, S_i)$ the number of losses in S_i associated with $u \in I(G)$. Moreover, given two nodes in S such that $y \leqslant_N x$, we define $lgt_{S_i}(x, y)$ as the number of switched-on secondary arcs in S_i on the path from x to y. Note that, given $u \in L(S_i)$, the single arc whose target is u is never a secondary arc. Then, using lgt_{S_i} instead of lgt, we can define the number of transfers associated with u in S_i, denoted by $t_\alpha(u, S_i)$, in the same way as $t_\alpha(u)$. Then, $t_\alpha(u) = \sum_{i=1}^{p} t_\alpha(u, S_i)$.

Now, given u in $I(G)$, by Lemma 3 of [21], there exists $H \in G^t_{B(u)}$ such that $u \in I(H)$ and $\alpha_l(u) = \beta^H_{S(B(u))}(u)$. This also implies that $\alpha_l(u)$ and $\alpha_l(u')$ are respectively contained in $B(u)$ and $B(u')$, where u' is a child of u.

Note that $t_\alpha(u, S_i) > 0$ only if the path from $\alpha_l(u)$ to $\alpha_l(u')$ (again, being u' a child of u) in S contains at least one switched-on secondary arc of B_i. Then, $t_\alpha(u, S_i) \geqslant 0$ can hold only for the three sets of nodes V_i^1, V_i^2, V_i^3 of $I(G)$ defined below:

- $V_i^1 := \{u \in I(G) : \alpha_l(u) \in B_i\}$.
- $V_i^2 := \{u \in I(G) : \alpha_l(u)$ is above $r(B_i)$ and either $\alpha_l(u_l)$ or $\alpha_l(u_r)$ are in $B_i\}$.
- $V_i^3 := \{u \in I(G) : \alpha_l(u)$ is above $r(B_i)$ and either $\alpha_l(u_l)$ or $\alpha_l(u_r)$ are below $B_i\}$.

Note that V_i^2 and V_i^3 are empty if $B_i = B(r(G))$. By construction V_1 is disjoint from V_2 and V_3; moreover, V_2 and V_3 are disjoint because if the two children of u have their α_l one in B_i and one below B_i, then $\alpha_l(u)$ must be in B_i and thus cannot be above $r(B_i)$.
Thus,

$$t(\alpha) = \sum_{i=1}^{p} \sum_{u \in V_i^1 \cup V_i^2 \cup V_i^3} t_\alpha(u, S_i).$$

Recall that the contribution to $t_\alpha(u, S_i)$ of any child u' of u consists of the secondary arcs in the path between $\alpha_l(u)$ and $\alpha_l(u')$ contained in S_i.

Given $u \in V_i^1$, by Lemma 3 in [21], there exists $H \in G^t_{B_i}$ such that $\alpha_l(u) = \beta^H_{S_i}(u)$ while, for each child u' of u, $\alpha_l(u')$ is somewhere in $B(u')$. If $B(u') = B(u)$, then by definition of G_N, u' is a child of u in H. If $B(u') \neq B(u)$, let B_j be the highest biconnected component in N such that $r(B_i) > r(B_j) \geqslant r(B(u'))$. Then, by definition of G_N and of $\beta^H_{S_i}$, $r(B_j)$ will label a leaf l of H that is a child of u. Thus $t_\alpha(u, S_i) = t_{\beta^H_{S_i}}(u)$. Then:

$$\sum_{u \in V_i^1} \tau \cdot t_\alpha(u, S_i) = \sum_{H \in G^t_{B_i}} \tau \cdot \sum_{u \in I(H)} t_{\beta^H_{S_i}}(u) = \sum_{H \in G^t_{B_i}} \tau \cdot t(\beta^H_{S_i}).$$

The first equivalence holds because 1) any internal node u of a tree $H \in G^t_{B_i}$ is a node of G that is mapped to B_i and, by definition of G_N, is an internal node, thus $u \in V_i^1$; 2) any such $u \in V_i^1$ is an internal node for some $H \in G^t_{B_i}$. In a similar way, from [21], we have:

$$\sum_{u \in V_i^1} \lambda \cdot l_\alpha(u, S_i) = \sum_{H \in G^t_{B_i}} \lambda \cdot l(\beta^H_{S_i}),$$

and $d_\alpha(S_i) = \sum_{H \in G^t_{B_i}} d(\beta^H_{S_i})$. Then, the following holds:

$$\delta \cdot \sum_{H \in G^t_{B_i}} d(\beta^H_{S_i}) + \sum_{u \in V_i^1} \Big(\tau \cdot t_\alpha(u, S_i) + \lambda \cdot l_\alpha(u, S_i)\Big) = \sum_{H \in G^t_{B_i}} cost(\beta^H_{S_i}).$$

Given $u \in V_i^2$ and u' being any of the children of u with $B(u') = B_i$, by Lemma 3 of [21], there exists $H \in G^t_{B_i}$ such that $\alpha_l(u') = \beta^H_{S_i}(u')$. Moreover, by definition of G_N, u' is the root of H, thus the contribution of u' to $t_\alpha(u, S_i)$ is $lgt(r(S_i), \beta^H_{S_i}(r(H)))$. From [21], we know that $l_\alpha(u, S_i) = dist(r(S_i), \beta^H_{S_i}(r(H)))$, thus, joining the cases for $u \in V_i^1$ and $u \in V_i^2$, we have:

$$\delta \cdot \sum_{H \in G^t_{B_i}} d(\beta^H_i) + \sum_{u \in V_i^1 \cup V_i^2} \Big(\tau \cdot t_\alpha(u, S_i) + \lambda \cdot l_\alpha(u, S_i)\Big) \overset{(*1)}{=} \sum_{H \in G^t_{B_i}} cost(H, S_i).$$

Again, the last equivalence holds because any internal node $\bar{u}$ of a tree $H \in G^t_{B_i}$ is a node in V_i^2, with $B_i \neq B(r(G))$, and vice versa.

Given $u \in V_i^3$ and u' being any of the children of u below B_i, let B_j be the first biconnected component of N between B_i and the component containing $\alpha_l(u')$. Then the number of secondary arcs in the path between $\alpha_l(u)$ and $\alpha_l(u')$ contained in S_i is, by construction, equal to $lgt(r(S_i), r(B_j))$. Since, as noted above, $\alpha_l(u)$ and $\alpha_l(u')$ are respectively contained in $B(u)$ and $B(u')$, then, by definition of G_N, each arc (u_a, u_b) in $G^e_{B_i}$ corresponds to exactly one of the biconnected component, namely $B(u_b)$, properly between $\alpha_l(u)$ and $\alpha_l(u')$ for a given u in V_i^3. Then, the following holds:

$$\sum_{u \in V_i^3} \tau \cdot t_\alpha(u, S_i) = \sum_{H := (u_a, u_b) \in G^e_{B_i}} \tau \cdot lgt(r(S_i), r(B(u_b))).$$

Similarly, from [21], we have:

$$\sum_{u \in V_i^3} \lambda \cdot l_\alpha(u, S_i) = \sum_{H := (u_a, u_b) \in G^e_{B_i}} \lambda \cdot dist(r(S_i), r(B(u_b))).$$

Then, from the definition of $cost(H, S_i)$, the following holds:

$$\sum_{u \in V_i^3} \Big(\tau \cdot t_\alpha(u, S_i) + \lambda \cdot l_\alpha(u, S_i)\Big) \overset{(*2)}{=} \sum_{H \in G^e_{B_i}} cost(H, S_i).$$

Now, by Lemma 4 in [21], $d(\alpha) = \sum_{i=1}^p d_\alpha(S_i) = \sum_{i=1}^p \sum_{H \in G^t_{B_i}} d(\beta^H_{S_i})$ and $l(\alpha) = \sum_{i=1}^p l_\alpha(S_i) = \sum_{i=1}^p \sum_{u \in V_i^1 \cup V_i^2 \cup V_i^3} l_\alpha(u, S_i)$. Moreover, above we proved that $t(\alpha) = \sum_{i=1}^p \sum_{u \in V_i^1 \cup V_i^2 \cup V_i^3} t_\alpha(u, S_i)$. Then, combining this with (*1) and (*2), we have:

$$
\begin{aligned}
cost(G,S) &= \delta \cdot d(\alpha) + \tau \cdot t(\alpha) + \lambda \cdot l(\alpha) \\
&= \sum_{i=1}^{p} \Big(\delta \cdot d_\alpha(S_i) + \tau \cdot \sum_{u \in I(G)} t_\alpha(u, S_i) + \lambda \cdot \sum_{u \in I(G)} l_\alpha(u, S_i) \Big) \\
&= \sum_{i=1}^{p} \Big(\delta \cdot \sum_{H \in G^t_{B_i}} d(\beta_i^H) + \sum_{u \in V_i^1 \cup V_i^2 \cup V_i^3} \Big(\tau \cdot t_\alpha(u, S_i) + \lambda \cdot l_\alpha(u, S_i) \Big) \Big) \\
&= \sum_{i=1}^{p} \Big([\delta \cdot \sum_{H \in G^t_{B_i}} d(\beta_i^H) + \sum_{u \in V_i^1 \cup V_i^2} \Big(\tau \cdot t_\alpha(u, S_i) + \lambda \cdot l_\alpha(u, S_i) \Big)] + \\
&+ [\sum_{u \in V_i^3} \Big(\tau \cdot t_\alpha(u, S_i) + \lambda \cdot l_\alpha(u, S_i) \Big)] \Big) \\
&= \sum_{i=1}^{p} \Big(\sum_{H \in G^t_{B_i}} cost(H, S_i) + \sum_{H \in G^e_{B_i}} cost(H, S_i) \Big) \\
&= \sum_{i=1}^{p} \sum_{H \in G_{B_i}} cost(H, S_i)
\end{aligned}
$$

This concludes the proof. □

Given a gene tree G, a level-k LGT network N with p biconnected components, the costs δ, τ and λ for respectively $\mathbb{D}$, $\mathbb{T}$ and $\mathbb{L}$ events, then, by Proposition 1, Problem 1 can be solved by Algorithm 1 in [21] using the new definition of $cost(H, S_i)$ given above. The complexity stays the same: $O(|V(N)| + 2^k \cdot p \cdot |V(G)|)$. See Theorem 2 of [21] for the correctness and running time analysis.

4.2. Finding the best reconciliation with the network

In the previous section, we showed how to find the tree in the network that has a most parsimonious $\mathbb{DTL}$ reconciliation with a given gene tree G. Now, if G contains several copies of a gene tree, each of it following a different evolutionary scenario, this model may not be the more adapted one. Another way to approach the problem is to drop the requirement of reconciling with a switching, and directly reconcile the gene tree with the LGT network instead. What we want to do is to extend Definition 1 to networks, allowing transfer events only via secondary arcs. To obtain this, we can simply modify Definition 2 as follows. First, we consider all arcs of N as switched-on. Second, condition 7 is modified so that $\alpha_{i+1}(u)$ is the only child of x through a principal arc. Note that, since in an LGT network all internal nodes have at least a principal outgoing arc, and here we consider all arcs of N as switched-on, condition 8 will never be fulfilled. Hence:

Definition 3. *Given an LGT network N and a gene tree G, α is said to be a $\mathbb{DTL}$ reconciliation between G and N if and only if exactly one of the following events occurs for each pair of nodes u of G and $\alpha_i(u)$ of S (for simplicity, let $x := \alpha_i(u)$ below):*

a) if x is the last node of $\alpha(u)$, one of the cases below is true:

1. *$u \in L(G)$, $x \in L(S)$ and $s(x) = s(u)$;* *(extant leaf)*
2. *$\{\alpha_1(u_l), \alpha_1(u_r)\} = \{x_l, x_r\}$;* $(\mathbb{S})$
3. *$\alpha_1(u_l) = x$ and $\alpha_1(u_r) = x$;* $(\mathbb{D})$
4. *$\alpha_1(u_l) = x$, $\alpha_1(u_r) = y$ and (x, y) is in E_s (or symmetrically interchanging the roles of u_l and u_r);*

   or $\alpha_1(u_l) = x_l$, $\alpha_1(u_r) = x_r$ and (x, x_r) is in E_s (or symmetrically interchanging the roles of u_l and u_r, and x_l and x_r, so 4 possibilities); $(\mathbb{T})$

b) otherwise, one of the cases below is true:

5. *$\alpha_{i+1}(u) = y$, (x, y) is in E_p;* $(\mathbb{SL})$
6. *$\alpha_{i+1}(u) = y$, (x, y) is in E_s;* $(\mathbb{TL})$
7. *$\alpha_{i+1}(u) = y$ and (x, y) is the only outgoing arc of x in E_p;* $(\emptyset)$

We denote the minimum cost of $\mathbb{DTL}$ reconciliation between G and N as $cost(G, N)$. We thus face the following problem:

Problem 2 (Best Reconciliation)**.**

Input: A gene tree G, an LGT network N, the costs δ, τ and λ for respectively $\mathbb{D}$, $\mathbb{T}$ and $\mathbb{L}$ events.

Output: A $\mathbb{DTL}$ reconciliation between G and N with cost $cost(G, N)$.

In Algorithm 1 we present a method to solve this problem, which is an adaptation of the algorithm presented in [3] to consider only transfers on secondary arcs. Note that in this algorithm we use the fact that N is a DAG and thus a bottom-up traversal of N exists. Again, since N is a time-consistent network, all lateral gene transfers are time-consistent.

Theorem 2. *Given a gene tree G, an LGT network N, the costs δ, τ and λ for respectively $\mathbb{D}$, $\mathbb{T}$ and $\mathbb{L}$ events, Algorithm 1 solves Problem 2 in $O(n \cdot m)$ space and time, where $n = |V(N)|$ and $m = |V(G)|$.*

Proof. We first prove the correctness of the algorithm. The algorithm fills a matrix $c : V(G) \times V(N) \to \mathbb{N}$ through two nested loops, each one visiting all nodes through a bottom-up traversal of G and N, respectively.

Consider the node u at an iteration of the loop in line 2. This loop (lines 3–28) computes $c(u, x)$ for each node $x \in V(N)$ by considering all six possible events separately. The consistency of the computation of the cost for each of

these events is ensured because for any child u' of u ($u' \in \{u_l, u_r\}$), any child x' of x ($x' \in \{x_l, x_r\}$) and any node $y \in V(N)$, the costs $c(u', y)$ and $c(u, x')$ have been previously computed thanks to the bottom-up traversal of G and N. Then, the final cost for $c(u, x)$ is computed by considering the minimum over all possible events. Since we can assume that a reconciliation with minimum cost maps the root of G to a single node of N, we can find this minimum cost by taking the minimum of $c(root(G), x)$ where $x \in V(N)$.

We now prove the running time and space cost of the algorithm. The loop over the nodes of G (line 2) runs for $O(m)$ iterations. The loop over the nodes of N (line 3) runs for $O(n)$ iterations. Thus, lines 4 to 27 run $O(m \cdot n)$ times. The computations of the costs of all possible events can be done in constant time. As a result, the overall time complexity of the algorithm is in $O(n \cdot m)$. The space complexity is completely determined by the size of the matrix $c(G, N)$, which is $m \times n$, and hence the space requirement is also $O(n \cdot m)$. □

Algorithm 1 Compute $cost(G, N)$ given positive costs δ, τ, and λ, respectively for $\mathbb{D}$, $\mathbb{T}$ $\mathbb{L}$ events

1: **for** each $u \in V(G)$ and $x \in V(N)$, **do** $c(u, x) \leftarrow \infty$ **end for** ▷ Initialize the matrix
2: **for** node $u \in V(G)$ according to a bottom-up traversal **do**
3: **for** node $x \in V(N)$ according to a bottom-up traversal **do**
4: **if** $u \in L(G)$, $x \in L(S)$, and $s(u) = s(x)$ **then**
5: $c(u, x) \leftarrow 0$. Go to the next iteration of the loop at line 3 ▷ Extant leaf
6: **end if**
7: **for all** $e \in \{\mathbb{S}, \mathbb{D}, \mathbb{T}, \varnothing, \mathbb{SL}, \mathbb{TL}\}$ **do** $c_e \leftarrow \infty$ **end for**
8: **if** u has two children **then**
9: $c_{\mathbb{D}} \leftarrow c(u_l, x) + c(u_r, x) + \delta$ ▷ $\mathbb{D}$ event
10: **if** x has two outgoing principal arcs **then**
11: $c_{\mathbb{S}} \leftarrow \min\{c(u_l, x_l) + c(u_r, x_r),\ c(u_l, x_r) + c(u_r, x_l)\}$ ▷ $\mathbb{S}$ event
12: **else if** x has two outgoing arcs and w.l.o.g. (x, x_r) is secondary **then**
13: $c_{\mathbb{T}} \leftarrow \min\{c(u_l, x) + c(u_r, x_r),\ c(u_l, x_r) + c(u_r, x)\} + \tau$ ▷ $\mathbb{T}$ event
14: $c_{\mathbb{T}} \leftarrow \min\{c_{\mathbb{T}}, c(u_l, x_l) + c(u_r, x_r) + \tau,\ c(u_l, x_r) + c(u_r, x_l) + \tau\}$ ▷ $\mathbb{T}$ event
15: **end if**
16: **else**
17: **if** x has two outgoing principal arcs **then**
18: $c_{\mathbb{SL}} \leftarrow \min\{c(u, x_l),\ c(u, x_r)\} + \lambda$ ▷ $\mathbb{SL}$ event
19: **else if** x has two outgoing arcs and w.l.o.g. (x, x_r) is secondary **then**
20: $c_{\mathbb{TL}} \leftarrow c(u, x_r) + \lambda + \tau$ ▷ $\mathbb{TL}$ event
21: $c_{\varnothing} \leftarrow c(u, x_l)$ ▷ $\varnothing$ event
22: **else**
23: $c_{\varnothing} \leftarrow c(u, x_l)$ ▷ $\varnothing$ event
24: **end if**
25: **end if**
26: $c(u, x) \leftarrow \min\{c_e : e \in \{\mathbb{S}, \mathbb{D}, \mathbb{T}, \varnothing, \mathbb{SL}, \mathbb{TL}\}\}$ ▷ Final cost for $c(u, x)$
27: **end for**
28: **end for**
29: **return** $\min\{c(root(G), x) : x \in V(N)\}$.

Note that, when all arcs are principal, Problem 2 coincides with Problem 2 in [21]. This implies that Algorithm 1 solves the latter problem for time-consistent LGT networks in $O(n \cdot m)$ time instead of $O(h^2 \cdot n \cdot m)$, as proposed

in [21], where h is the number of reticulations of N. Moreover, our definition of a time-consistent LGT network coincides with that of a species graph in [13], apart for condition H6 [13, Section 3.2], which is not required here (Actually, we speculate that for any species graph N satisfying conditions $S'1 - S'3$ and time stamp for N satisfying conditions $H1 - H5$ in [13, Section 3.2], the time stamp can be modified to satisfy also condition $H6$. Thus any time-consistent LGT network is a species graph, and vice versa). In [13], Problem 2 was solved in $O(n^3 \cdot m)$ time.

Note that the approach described in this section is very similar to searching the most parsimonious $\mathbb{DTL}$ reconciliation between the distinguished base tree of N and G, and only allowing transfer along secondary arcs. Still, our approach, unlike the distinguished base tree one, is able to ensure the time-consistency of all gene transfers.

5. Experiments

We tested our method on a data set of 1128 genes from 11 cyanobacterial species first studied in [31]. This data set was used later in [32] in order to test a method aiming at detecting pairs of (ancestral) species between which many different genes were horizontally transferred – these pairs are said to be connected by a *highway* [33]. The method in [32] takes as input a set of unrooted gene trees and a rooted species tree, then it decomposes the gene trees into quartet trees – trees with four leaves – and tries to identify pairs of species such that a highway between them explains a great amount of the inconsistency between the quartet trees and the species tree (an improvement of this method that is also based on quartet decomposition is presented in [34]).

In [32], the authors used their method on the set of 1128 unrooted gene trees of [31] and the rooted species tree depicted in black in Figure 6, and suggested several highways of transfers, also depicted in Figure 6.

We applied our method on the same data set, rooting the trees with *Gloeobacter* as outgroup (we discarded the 27 genes not present in this species). We used as cost vector (2,3,1) respectively for duplications, transfers and losses; this cost has been used in several biological studies [35, among other]. Note that our method, unlike the method in [32], easily permits to analyse the two possible directions of a highway. Thus, we analysed each of the 4 possible highways, one at the time and in both directions, generating 8 LGT networks with one secondary arc each that were reconciled against the 1101 gene trees according to both models described in the previous section (see Table 1). Of the 4 putative highways, only the red and the dotted ones have scores that are significantly lower than the score of the species tree, i.e. when transfers are not permitted (line 1 in Table 1). This means that the other 2 highways can be explained – with a similar cost – with duplications and losses and are possibly due to pseudo-orthology (a combination of duplications and asymmetrical losses) rather than transfers. The red highway has a significantly lower score among all the highways, and its score is lower when directed toward Synechococcus, see lines 2-3 of Table 1 (this is true for all transfer costs that we tried, data not shown). On

	Problem 1	Problem 2
species tree	15.53	15.53
red highway from Synechococcus (H1)	13.91	14.88
red highway to Synechococcus (H2)	12.58	12.73
dotted highway to Thermosynechococcus (H3)	15.3	14.91
dotted highway from Thermosynechococcus (H4)	14.65	15.23
H2+H3	12.34	12.11
H2+H4	11.68	12.5

Table 1: Average reconciliation scores for Problem 1 and 2 for several LGT networks on the set of 1101 gene trees described in the main text.

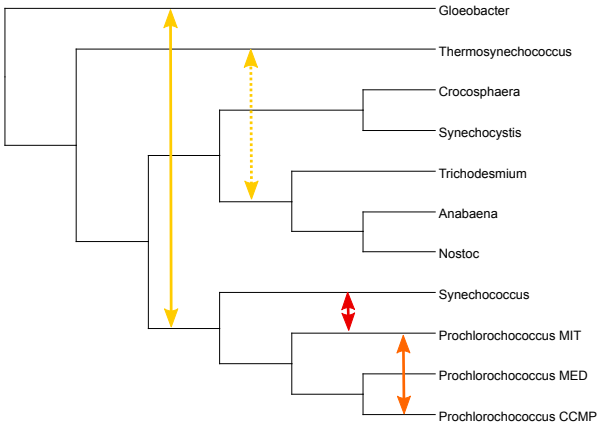

Figure 6: A putative phylogeny of the 11 cyanobacterial species studied in [31] constructed from the 16S ribosomal RNA sequence [36] is depicted in black. Coloured edges represent the putative highways of transfers detected in [32]. The edge in red represents the highway with the highest score, and the one in orange the one with the second highest score. The edges in yellow represents highways with an even lower score. Adapted from [32].

the contrary, the direction of the dotted highway is unclear, since one direction has lower score for Problem 1 and the other for Problem 2 (lines 4-5 of Table 1); this is confirmed when considering this highway in combination with the red highway to Synechococcus in the same LGT network (lines 6-7 of Table 1). Note that, for the cost vector (2,3,1), the combination of the red and the dotted highways has a score lower than the red highway only, adding evidence to its existence. But, when the transfer cost increases, the significance of the gap decreases considerably (data not shown).

6. Conclusion

In this paper we have proposed a model for the reconciliation problem between genes and evolutionary histories of species using the so-called LGT-

networks and considering an scenario with duplications, losses and transfers. Then, we have given a polynomial algorithm that solves the problem of finding the most parsimonious reconciliation. Our algorithms have been tested using biologically significant data. The comparison of our results with previous published results on the same data shows that our model is well suited for the reconciliation of gene trees with evolutive histories involving a high amount of transfers. Also, our experiments prove that our method, when combined with quartet decomposition methods to detect putative highways, permits to refine their analyses since it allows analysing the two possible directions of a highway and considering combinations of highways.

Further work includes the development of a maximum likelihood procedure inspired by our minimum parsimony procedure, and the development of a maximum likelihood framework to estimate LGT networks, similarly to what has been done in PHYLDOG [23] to estimate a species tree from a set of genes.

The methods presented in this paper have been integrated into the ecceTERA software [28], freely available at `http://mbb.univ-montp2.fr/MBB/subsection/softExec.php?soft=eccetera`.

7. Acknowledgements

This work was supported by the Spanish Ministry of Economy and Competitiveness and European Regional Development Fund project DPI2015-67082-P (MINECO/FEDER) (GC and JCP), *Obra Social "La Caixa"* bursary for work stays (JCP), and the French Agence Nationale de la Recherche Investissements d'Avenir/ Bioinformatique (ANR-10-BINF-01-02, Ancestrome) (CS).

The authors would like to thank Olga Zhaxybayeva and Mukul Bansal for providing the data for the experiments section.

Bibliography

[1] K. Y. Gorbunov, V. A. Lyubetsky, Reconstructing genes evolution along a species tree, Mol. Biol. (Mosk.) 43 (2009) 946–958.

[2] J.-P. Doyon, C. Chauve, S. Hamel, Space of gene/species trees reconciliations and parsimonious models, Journal of Computational Biology 16 (10) (2009) 1399–1418.

[3] J.-P. Doyon, C. Scornavacca, K. Y. Gorbunov, G. J. Szöllősi, V. Ranwez, V. Berry, An efficient algorithm for gene/species trees parsimonious reconciliation with losses, duplications and transfers, in: RECOMB International Workshop on Comparative Genomics, Springer, 2010, pp. 93–108.

[4] A. Tofigh, M. Hallett, J. Lagergren, Simultaneous identification of duplications and lateral gene transfers, IEEE/ACM Transactions on Computational Biology and Bioinformatics 8 (2) (2011) 517–535.

[5] M. S. Bansal, E. J. Alm, M. Kellis, Efficient algorithms for the reconciliation problem with gene duplication, horizontal transfer and loss, Bioinformatics 28 (12) (2012) i283–i291.

[6] M. S. Bansal, E. J. Alm, M. Kellis, Reconciliation revisited: Handling multiple optima when reconciling with duplication, transfer, and loss, Journal of Computational Biology 20 (10) (2013) 738–754.

[7] C. Scornavacca, W. Paprotny, V. Berry, V. Ranwez, Representing a set of reconciliations in a compact way, Journal of Bioinformatics and Computational Biology 11 (02) (2013) 1250025.

[8] R. Libeskind-Hadas, Y.-C. Wu, M. S. Bansal, M. Kellis, Pareto-optimal phylogenetic tree reconciliation, Bioinformatics 30 (12) (2014) i87–i95.

[9] L. Zhang, On a Mirkin-Muchnik-smith conjecture for comparing molecular phylogenies, Journal of Computational Biology 4 (2) (1997) 177–187.

[10] M. Hallett, J. Lagergren, A. Tofigh, Simultaneous identification of duplications and lateral transfers, in: Proceedings of the Eighth Annual International Conference on Research in Computational Molecular Biology, ACM, 2004, pp. 347–356.

[11] C. Conow, D. Fielder, Y. Ovadia, R. Libeskind-Hadas, Jane: a new tool for the cophylogeny reconstruction problem, Algorithms for Molecular Biology 5 (1) (2010) 1.

[12] D. Merkle, M. Middendorf, N. Wieseke, A parameter-adaptive dynamic programming approach for inferring cophylogenies, BMC bioinformatics 11 (1) (2010) 1.

[13] P. Górecki, Reconciliation problems for duplication, loss and horizontal gene transfer, in: Proceedings of the Eighth Annual International Conference on Research in Computational Molecular Biology, ACM, 2004, pp. 316–325.

[14] D. A. Morrison, Introduction to Phylogenetic Networks, RJR Productions, Uppsala, Sweden, 2011.

[15] D. Huson, R. Rupp, C. Scornavacca, Phylogenetic Networks. Concepts, Algorithms and Applications, Cambridge University Press, Cambridge, UK, 2010.

[16] R. Libeskind-Hadas, M. A. Charleston, On the computational complexity of the reticulate cophylogeny reconstruction problem, Journal of Computational Biology 16 (1) (2009) 105–117.

[17] R. D. Page, Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas, Systematic Biology 43 (1) (1994) 58–77.

[18] F. Ronquist, Reconstructing the history of host-parasite associations using generalised parsimony, Cladistics 11 (1) (1995) 73–89.

[19] M. Charleston, Jungles: a new solution to the host/parasite phylogeny reconciliation problem, Mathematical Biosciences 149 (2) (1998) 191–223.

[20] D. Merkle, M. Middendorf, Reconstruction of the cophylogenetic history of related phylogenetic trees with divergence timing information, Theory in Biosciences 123 (4) (2005) 277–299.

[21] T.-H. To, C. Scornavacca, Efficient algorithms for reconciling gene trees and species networks via duplication and loss events, BMC Genomics 16 (10) (2015) 1–14.

[22] G. Cardona, J. C. Pons, F. Rosselló, A reconstruction problem for a class of phylogenetic networks with lateral gene transfers, Algorithms for Molecular Biology 10 (1) (2015) 1–15.

[23] B. Boussau, G. J. Szöllősi, L. Duret, M. Gouy, E. Tannier, V. Daubin, Genome-scale coestimation of species and gene trees, Genome Research 23 (2) (2013) 323–330.

[24] C. Choy, J. Jansson, K. Sadakane, W.-K. Sung, Computing the maximum agreement of phylogenetic networks, Theoretical Computer Science 335 (1) (2005) 93–107.

[25] A. R. Francis, M. Steel, Which phylogenetic networks are merely trees with additional arcs?, Systematic biology 64 (5) (2015) 768–777.

[26] J.-P. Doyon, V. Ranwez, V. Daubin, V. Berry, Models, algorithms and programs for phylogeny reconciliation, Briefings in Bioinformatics 12 (5) (2011) 392–400.

[27] V. Ranwez, C. Scornavacca, J.-P. Doyon, V. Berry, Inferring gene duplications, transfers and losses can be done in a discrete framework, Journal of Mathematical Biology (2015) 1–34.

[28] E. Jacox, C. Chauve, G. J. Szöllősi, Y. Ponty, C. Scornavacca, eccetera: comprehensive gene tree-species tree reconciliation using parsimony, Bioinformatics (2016) btw105.

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

[30] M. Goodman, J. Czelusniak, G. W. Moore, A. Romero-Herrera, G. Matsuda, Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences, Systematic Biology 28 (2) (1979) 132–163.

[31] O. Zhaxybayeva, J. P. Gogarten, R. L. Charlebois, W. F. Doolittle, R. T. Papke, Phylogenetic analyses of cyanobacterial genomes: quantification of horizontal gene transfer events, Genome Research 16 (9) (2006) 1099–1108.

[32] M. S. Bansal, G. Banay, J. P. Gogarten, R. Shamir, Detecting highways of horizontal gene transfer, Journal of Computational Biology 18 (9) (2011) 1087–1114.

[33] R. G. Beiko, T. J. Harlow, M. A. Ragan, Highways of gene sharing in prokaryotes, Proceedings of the National Academy of Sciences of the United States of America 102 (40) (2005) 14332–14337.

[34] M. S. Bansal, G. Banay, T. J. Harlow, J. P. Gogarten, R. Shamir, Systematic inference of highways of horizontal gene transfer in prokaryotes, Bioinformatics 29 (5) (2013) 571–579.

[35] L. A. David, E. J. Alm, Rapid evolutionary innovation during an archaean genetic expansion, Nature 469 (7328) (2011) 93–96.

[36] G. P. Fournier, J. P. Gogarten, Rooting the ribosomal tree of life, Molecular Biology and Evolution 27 (8) (2010) 1792–1801.