

RESEARCH

Open Access

# Exact median-tree inference for unrooted reconciliation costs



Paweł Górecki<sup>1\*</sup>, Alexey Markin<sup>2</sup> and Oliver Eulenstein<sup>2</sup>

From The 18th Asia Pacific Bioinformatics Conference  
Seoul, Korea. 18-20 August 2020

## Abstract

**Background:** Solving median tree problems under tree reconciliation costs is a classic and well-studied approach for inferring species trees from collections of discordant gene trees. These problems are NP-hard, and therefore are, in practice, typically addressed by local search heuristics. So far, however, such heuristics lack any provable correctness or precision. Further, even for small phylogenetic studies, it has been demonstrated that local search heuristics may only provide sub-optimal solutions. Obviating such heuristic uncertainties are exact dynamic programming solutions that allow solving tree reconciliation problems for smaller phylogenetic studies. Despite these promises, such exact solutions are only suitable for credibly rooted input gene trees, which constitute only a tiny fraction of the readily available gene trees. Standard gene tree inference approaches provide only unrooted gene trees and accurately rooting such trees is often difficult, if not impossible.

**Results:** Here, we describe complex dynamic programming solutions that represent the first nonnaïve exact solutions for solving the tree reconciliation problems for unrooted input gene trees. Further, we show that the asymptotic runtime of the proposed solutions does not increase when compared to the most time-efficient dynamic programming solutions for rooted input trees.

**Conclusions:** In an experimental evaluation, we demonstrate that the described solutions for unrooted gene trees are, like the solutions for rooted input gene trees, suitable for smaller phylogenetic studies. Finally, for the first time, we study the accuracy of classic local search heuristics for unrooted tree reconciliation problems.

**Keywords:** Median tree, Reconciliation, Gene duplication, Gene loss, Deep coalescence, Exact solution

## Background

Phylogenetic trees visualize estimates of the evolutionary relationships among multiple biological entities such as molecular sequences, genomes, and species. For biologists, such trees present a fundamental tool for analyzing how distinct biological entities have evolved but are full of complexities and seemingly irreconcilable differences [1]. The implications and potential applications of

phylogenetic analyses are widespread and are concerning a wide variety of central research areas including biology, ecology, epidemiology, and conservation biology (e.g., [2–6]).

Conventional phylogenetic tree inference samples an individual gene (i.e., a gene family) for a collection of species and reconstructs the evolutionary history, called a *gene tree*, of this gene. The gene is a portion of the species' genomes, and it is assumed that the corresponding gene tree is mimicking the evolution of the species. Accordingly, the gene tree is identified with the species tree.

\*Correspondence: [gorecki@mimuw.edu.pl](mailto:gorecki@mimuw.edu.pl)

<sup>1</sup>University of Warsaw, Faculty of Mathematics, Informatics and Mechanics, Banacha 2, 02-097 Warsaw, Poland

Full list of author information is available at the end of the article



© The Author(s). 2020 **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>. The Creative Commons Public Domain Dedication waiver (<http://creativecommons.org/publicdomain/zero/1.0/>) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

However, it is well-recognized that distinct gene families frequently yield incongruent gene trees due to intricate evolutionary processes which are affecting genomic locations of the genes in different ways [7, 8]. Such processes include gene duplications, gene loss, and deep coalescence. Consequently, identifying a gene tree with its species tree can result in largely misleading phylogenetic analyses [9, 10].

Today, algorithmic advances have given rise to powerful computational tools for inferring credible species tree estimates from collections of discordant gene trees by solving median tree problems [11]. Given a set of gene trees, *median tree problems* (also referred to as supertree problems [12]) seek a tree, called a *median tree*, that is minimizing the overall cost function in regards to the input trees (using a problem-specific cost function). Such cost functions, for example, include the classic and well-studied tree reconciliation costs, which are informed by evolutionary models that explain incongruence between gene trees and species trees. Tree reconciliation costs account for the minimum number of evolutionary events required to reconcile the discordance between a gene tree and a species tree, where both trees are rooted. For the *tree reconciliation costs*, we consider here the fundamental evolutionary events of gene duplication, gene loss [13–16], and deep coalescence [17]. The median tree problems under the tree reconciliation costs, which we refer to as (*median*) *tree reconciliation problems*, will be the focus of this work.

Similar to the in practice established median tree problems [11], the tree reconciliation problems are also NP-hard [18, 19], and therefore, are frequently addressed by local search heuristics [20–25]. The challenging task of such heuristic approaches is to efficiently search a globally optimal species tree in a solution landscape that is literally of astronomical size (i.e., the search space increases super-exponentially with the number of taxa [26]). Moreover, this landscape has typically numerous local optima that can easily trap heuristic approaches [27]. So far, these local search heuristics lack any provable correctness or precision [28, 29].

To obviate such heuristic uncertainties, exact dynamic programming solutions have been described for the tree reconciliation problems [20, 28]. These solutions are asymptotically significantly faster than the naïve enumeration of the solution space allowing to compute median trees suitable for smaller studies. Furthermore, such exact solutions have demonstrated that local search heuristics can fail to find optimal solutions for even small-sized studies (i.e., eight species [30]). This makes exact dynamic programming solutions the most appealing choice for phylogenetic studies involving smaller numbers of taxa (e.g., up to 25 taxa [31]).

Unfortunately, the exact dynamic programming solutions are not applicable to the large number of unrooted gene trees that are typically inferred from molecular sequences. Standard tree inference methods that are used in practice, such as maximum parsimony or maximum likelihood, infer only unrooted gene trees, and identifying credible rootings in such trees is often challenging, if not unattainable [32, 33]. Outgroup rooting, for instance, can produce inaccurate rootings when evolutionary events cause heterogeneity in the gene trees. Rooting gene trees under the molecular clock hypothesis, or likewise by using midpoint rooting, also can result in an error when there is a molecular rate variation throughout the tree [34, 35].

However, there has been an increased interest in rooting unrooted gene trees using a credibly rooted species tree such that a chosen tree reconciliation cost is minimized [36–38]. The corresponding extension of the standard reconciliation costs to costs between an unrooted gene tree and a rooted species tree are well-studied and referred to as the *unrooted gene tree reconciliation cost*. For example, consider the standard deep coalescence cost defined between a rooted gene tree and a rooted species tree; the *unrooted deep coalescence cost* between an unrooted gene tree and a rooted species tree is then defined as the minimum rooted deep coalescence cost between any rooting of the unrooted gene tree and the species tree (i.e., the minimum cost over all rootings).

Naïvely, the tree reconciliation problems for the unrooted gene tree reconciliation costs can be solved by a complete enumeration of the input trees' rooting scenarios and selecting a scenario that results in a median tree with the minimum tree reconciliation cost using an exact dynamic programming solution for rooted input gene trees. However, the super-exponential runtime increase caused by the enumeration makes this approach unsuitable even for smaller sized phylogenetic studies involving, e.g., more than five gene trees (see the “*Scalability analysis*” section).

Here, we describe an exact and non-naïve method, termed UrExact, for solving the tree reconciliation problems for complete<sup>1</sup> and unrooted input gene trees, i.e., using the unrooted reconciliation costs. Perhaps surprisingly, UrExact has the same asymptotic runtime as the best know exact method [28], termed RExact, for the tree reconciliation problems for rooted input gene trees. In practice, however, gene trees also can be incomplete. This can be addressed by filling in the gene trees (e.g., [29]), and then applying UrExact. Another option is to apply the *minus method* [39]; that is when computing the pairwise unrooted reconciliation costs between a gene tree and a candidate species tree, the species tree is constrained to the host-species of the gene tree's leaf-genes. We show

<sup>1</sup>By “complete” we imply that the trees have full taxon sets.

how to handle the minus method in the “Results” section, which requires an extension of our approach. Further, using comparative experimental studies, we showcase the runtime differences between the naïve enumeration solution using RExact and our solutions in practice. These studies demonstrate that, in contrast to the naïve solution, UrExact is suitable for exact inference on smaller phylogenetic studies (i.e., up to 11 taxa for 100 input gene trees). Finally, we study the accuracy of classic local search heuristics for unrooted median tree reconciliation problems, and demonstrate that it typically requires a large number of independent heuristic runs to find an exact median tree for already small scale phylogenetic studies. This suggests that such heuristic approaches may often not be suitable for credible phylogenetic studies, in particular for large-scale studies.

### Related work

Our exact dynamic programming solution for the unrooted tree reconciliation problems, UrExact, are building on research areas related to (i) the tree reconciliation costs, (ii) the time complexities of the tree reconciliation problems, and (iii) the unrooted tree reconciliation costs. Therefore, here we will provide more detail for each of these areas.

### Tree reconciliation costs

The pioneering work of Goodman et al. [15], almost 40 years ago, introduced tree reconciliation as a fundamental approach of explaining discord between a gene tree and a species tree by evolutionary means. *Tree reconciliation* is an approach that is embedding a rooted gene tree into a rooted species tree using a *mapping function* that relates every gene in the gene tree to the most recent species in the species tree, referred to as the *gene’s host-species*, that could have contained the gene. In practice, the mapping of a gene tree’s leaf-genes is known to be the species from which the genes were sampled. The mapping function allows identifying the evolutionary events gene duplication, gene loss, and deep coalescence. *Gene duplications* are genes in the gene tree that have the same host-species as one of their children. *Gene losses* are accounted for by the maximum subtrees in the species tree that have no host-species. Associated with the number of these events are the tree reconciliation costs which are termed *gene duplication cost*, *gene loss cost*, and *gene duplication and loss cost* accounting for the number of gene duplications, gene losses, and gene duplications plus losses, respectively. Another classic reconciliation cost is the *deep coalescence cost* that accounts for each edge in the species tree the number of embedded lineages from the gene tree minus one, which are assumed to be the additional lineages caused by deep coalescence events [17].

### Time complexity of the tree reconciliation problems

For rooted input gene trees, the tree reconciliation problems are NP-hard under the costs for (i) gene duplication, gene loss, and gene plus loss [18], and (ii) deep coalescence [19]. However, RExact can solve these problems in  $O(3^{mn}/b)$  time and with  $O(2^n + mn/b)$  space, using dynamic programming, a bit-vector encoding of size  $b$  (ideally representing the register size of the computation machine used) and Gray encoding, where  $n$  is the number of input trees and  $m$  the number of unique rooted splits in the input trees [28]. The time complexities of the tree reconciliation problems for unrooted input gene trees are open, though, we conjecture that the problems are NP-hard.

### Unrooted tree reconciliation costs

Initially, all gene tree parsimony costs were defined only for comparing rooted gene trees with rooted species trees. [15, 17]. However, in general, unrooted trees can be compared with rooted trees by identifying the rootings of the unrooted tree that is minimizing any provided cost function between a pair of rooted trees. Further, the gene tree parsimony costs satisfy the plateau property, which is sufficient for the linear time identification of all optimal rootings and rooting costs in the unrooted gene tree. The *plateau property* is satisfied when all optimal rootings of the unrooted gene tree form a subtree in this tree, and the rootings along every path toward a leaf have monotonically increased costs [38].

## Methods

### Basic definitions

A *rooted tree* is a rooted binary tree whose leaves are labeled by species names (not necessarily uniquely). Let  $T$  be a gene tree. By  $L(T)$  we denote the set of all leaf labels (i.e., species) present in  $T$ . The root of  $T$  is denoted  $\text{root}(T)$ . A node  $n$  is called *internal* if it has two children, which are denoted by  $\text{Ch}(n)$ . A cluster of a node  $g$ , denoted by  $\text{clu}(g)$ , is the set of leaf labels *below*  $g$ . A *species tree* is a rooted tree whose leaves are *uniquely* labeled (that is, there are no two leaves with the same species label).

Let  $S = \langle V_S, E_S \rangle$  be a species tree. For nodes  $a, b \in V_S$ , by  $a + b$  we denote the least common ancestor of  $a$  and  $b$  in  $S$ . We also use the binary order relation  $a \leq b$  if  $b$  is a node on the path between  $a$  and  $\text{root}(S)$  (note that  $a \leq a$ ). Two nodes  $a$  and  $b$  are called *siblings* if they are children of  $a + b$ .

For a rooted tree  $G$ , called here a *rooted gene tree*, and a species tree  $S$  such that  $L(G) \subseteq L(S)$ , a *least common ancestor mapping*, or lca-mapping, is a function from the nodes of  $G$  to the nodes of  $S$  such that  $M(g) = s$  if  $g$  and  $s$  are leaves with the same label, or  $M(g) = M(a) + M(b)$  if  $g$  is an internal node of  $G$  such that  $a$  and  $b$  are the children of  $g$ .

An *unrooted gene tree* is a tree whose internal nodes are of degree 3 and species names label the leaves. An unrooted gene tree  $G$  can be rooted by placing the root on an edge  $e$  (that is, by subdividing edge  $e$  with a new node  $\rho$  and designating it to be the root). Such a *rooting* (i.e., a resulting rooted gene tree) is denoted  $G_e$ .

**Cost functions**

Now, we introduce several cost functions used when reconciling a rooted gene tree  $G$  and a species tree  $S$ . An internal node  $g \in V_G$  is a (*gene*) *duplication* if  $M(g) = M(g')$  for  $g'$  a child of  $g$ . The total number of gene duplications is called *duplication cost* and denoted by  $D(G, S)$ . The *deep coalescence* cost is defined as follows [17]:  $DC(G, S) = \sum_{a,b \text{ siblings in } G} (|\pi(M(a), M(b))| - 1)$ , where  $\pi(x, y)$  is the set of all nodes on the shortest path connecting  $x$  and  $y$  in  $S$ . Note that the standard definition of the *deep coalescence* cost function [40] simply differs by  $1 - |V_G|$  from our definition, which is a constant value for a fixed  $G$ , and the results presented in this work can be easily adapted for the standard definition. Next, we define the *the loss cost* based on the formula derived in [19]:  $L(G, S) = DC(G, S) + 2 \cdot D(G, S) - |V_G| + 1$  and the *duplication-loss cost*:  $DL(G, S) = D(G, S) + L(G, S)$ .

The above cost functions can be naturally extended to unrooted gene trees. For every cost function  $c$  defined above for rooted gene trees, we define its unrooted counterpart as  $\min_{e \in E(G)} c(G_e, S)$ . For convenience, we adopt the same notation (i.e.,  $D, DC, L, DL$ ) to denote unrooted cost functions. The edge  $e$ , such that  $G_e$  has the minimal cost  $c(G_e, S)$ , is called *optimal* (for  $c$ ).

For a given rooted or unrooted gene tree  $G$  and a species tree  $S$ ,  $c(G, S)$  can be computed in linear time [41, 42].

**Problems**

We say that a species tree  $S$  is *over a set of species*  $I$  if  $L(S) = I$ . Let  $Q$  be a collection of unrooted gene trees  $G_1, G_2, \dots, G_n$ . By  $L(Q)$  we denote the set of all species present in the trees of  $Q$ . We extend the notion of the cost function to collections of gene trees. For a given species tree  $S$  over  $L(Q)$ , by  $c(Q, S)$  we denote the total cost  $\sum_{G \in Q} c(G, S)$ .

**Problem 1** (uMinST - Minimal Species Tree) *Given a collection of unrooted gene trees  $Q$  and a cost function  $c$  find a species tree  $S^{\min}$  that minimizes the total cost  $c(Q, S)$  in the set of all species trees  $S$  over  $L(Q)$ .*

We call a species tree  $S^{\min}$  *minimal* (for  $Q$  and  $c$ ) and we denote the respective minimal cost by  $c^{\min}(Q)$ . We also define a simpler variant of the previous problem that does not require finding a tree  $S^{\min}$  explicitly.

**Problem 2** (uMinCC - Minimal Cost Computation) *Given a collection of unrooted gene trees  $Q$  and a cost function  $c$  compute  $c^{\min}(Q)$ .*

Similarly, we define problems for the collections of rooted trees, called rMinST and rMinCC, respectively. Note that [28] provides a dynamic programming solution to rMinST and rMinCC.

**Solution to rMinST and rMinCC - overview**

In this section, we show how rMinCC is solved in [28]. Any internal node  $v$  of a gene or species tree determines a split  $A|B$ , where  $A$  and  $B$  are the clusters of children of  $v$ . For a collection  $Q$  of rooted gene trees let  $r(Q)$  be the multiset of all rooted splits present in trees of  $Q$ . We also set  $r(T)$  to be  $r(\{T\})$  for any rooted tree  $T$ . For brevity, here we present the dynamic programming formulations only for  $D$  and  $DC$  costs. For  $DL$  and  $L$ , please refer to [28].

For a collection of gene trees  $Q$  and a species  $s$ ,  $\Lambda(Q, s)$  is the total cost contribution of the nodes from  $Q$  to a leaf of some species tree over  $L(Q)$  labeled by  $s$ . Let now  $X$  and  $Y$  be two disjoint sets of species. Then  $\Gamma(Q, X, Y)$  is the total cost contribution of the nodes from  $Q$  to an internal node  $v$  of some species tree over  $L(Q)$  such that the cluster of  $v$  is  $X \cup Y$  and  $v$  has two children whose clusters are  $X$  and  $Y$ . Given a species leaf  $s$  and a split  $X|Y$  and a cost  $c \in \{D, DC, DL\}$ :

$$\Lambda^c(Q, s) = \sum_{q \in r(Q)} \lambda^c(q, s),$$

$$\Gamma^c(Q, X|Y) = \sum_{q \in r(Q)} \gamma^c(q, X|Y),$$

where

$$\lambda^D(A|B, s) = \lambda^{DL}(A|B, s) = \mathbf{1}[A \cup B = \{s\}],$$

$$\lambda^{DC}(A|B, s) = \mathbf{1}[\exists i: A_i = \{s\} \neq A_{i+1}]$$

$$\gamma^D(A_1|A_2, X_1|X_2) = \mathbf{1}[A_1 \cup A_2 \subseteq X_1 \cup X_2 \wedge \exists i: X_1 \not\supseteq A_i \not\subseteq X_2],$$

$$\gamma^{DC}(A_1|A_2, X_1|X_2) = \mathbf{1}[\exists i, j: X_j \not\supseteq A_i \subseteq X_1 \cup X_2 \not\supseteq A_{i+1}].$$

The above functions can be used to compute the cost as follows:

$$c(Q, S) = \sum_{s \in L(S)} \Lambda^c(Q, s) + \sum_{q \in r(S)} \Gamma^c(Q, q). \tag{1}$$

The dynamic programming solution to rMinCC from [28] is as follows:

$$\Delta^c(Q, Z) = \begin{cases} \Lambda^c(Q, s) & \text{if } Z = \{s\}, \\ \min_{X|Y \in \text{splits}(Z)} \Delta^c(Q, X) + \Delta^c(Q, Y) + \Gamma^c(Q, X, Y) & \text{otherwise,} \end{cases}$$

where  $\text{splits}(Z)$  is the set of all splits (2-partitions) of  $Z$  and the solution is given by  $\Delta^c(Q, L(Q))$ .

**Unrooted reconciliation**

We now summarize the previous structural results on reconciliation of an unrooted gene tree with a species tree.

These results will be used for the design of our main dynamic programming solutions to uMinST and uMinCC problems.

Without loss of generality, we assume that every unrooted/rooted gene tree has at least 3 distinct labels and it has all labels from a species tree, i.e.,  $L(G) = L(S) \geq 3$ . Let  $G$  be an unrooted gene tree and  $S$  be a fixed species tree. The split of the root of  $S$  we call *top-split*.

A set of species  $Z$  is *simple* if  $Z$  is a subset of a cluster from the top-split. Let  $\zeta(Z)$  be a predicate that is true if  $Z$  is simple. Further, let  $G$  be an unrooted tree. Any internal node  $g$  of  $G$  determines a star  $A|B|C$ , where  $A, B$  and  $C$  are the leaves of three subtrees obtained from  $G$  by removing  $g$ . Note that  $A \cup B \cup C = L(S)$ . Let  $\bar{A} = B \cup C, \bar{B} = C \cup A$  and  $\bar{C} = A \cup B$ . Then, it follows from [42, 43], that given a top-split we have five disjoint types of stars (see Fig. 1) in unrooted gene trees (reordering of  $A, B$  and  $C$  may be required; e.g., the two stars in  $G_2$  from Fig. 2 are represented as  $a|b|c$  (or  $a|c|b$ ) and  $bc|a|a$  in the context of the top-split  $a|bc$ ):

- S1 if  $\neg\zeta(A) \wedge \zeta(\bar{A})$ ,
- S2 if  $\zeta(A) \wedge \zeta(\bar{A})$ ,
- S3 if  $\neg\zeta(A) \wedge \neg\zeta(\bar{A}) \wedge \zeta(B) \wedge \zeta(C)$ ,
- S4 if  $\neg\zeta(A) \wedge \neg\zeta(B) \wedge \neg\zeta(C)$ ,
- S5 if  $\zeta(A) \wedge \neg\zeta(B) \wedge \neg\zeta(C)$ .

Given a species tree  $S$ , an edge of an unrooted tree  $G$  is *symmetric* if removing the edge from  $G$  creates two trees whose top-clusters are either both simple or both are not simple. The remaining edges are called *asymmetric*. E.g., all edges of S4 are symmetric, while all edges of S1

are asymmetric. We have the main theorem for unrooted reconciliation theory.

**Theorem 1** (Adopted from [42–45]) *If  $e$  is a symmetric edge in any star or  $e$  is the asymmetric edge from S5 then  $c(G_e, S) = c(G, S)$ , i.e.,  $e$  is optimal for every cost  $c \in \{D, DC, L, DL\}$ .*

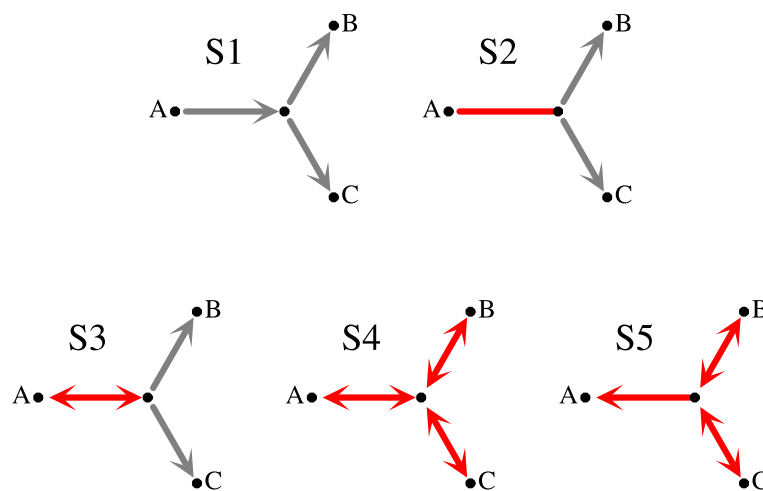
The edges satisfying the conditions from the above theorem induce a connected unrooted subtree in  $G$  called a *plateau*. The plateau in our article equals the DL-plateau from [42], where the  $c$ -plateau is the graph induced by the set of optimal rooting edges for the cost  $c$ . Since any optimal rooting for DL is optimal for D, DC and DL [42], to solve our problems it is sufficient to focus on the plateau (or DL-plateau) rootings only. Examples are depicted in Fig. 2.

**Decomposing unrooted gene trees**

Before we start with the main results, we show how given a species tree (or its top-split only), an unrooted gene tree can be decomposed into two parts: one part which is of the rooted nature and the second part related to the plateau. Note that the notion of a cost function refers to D, DC, L or DL.

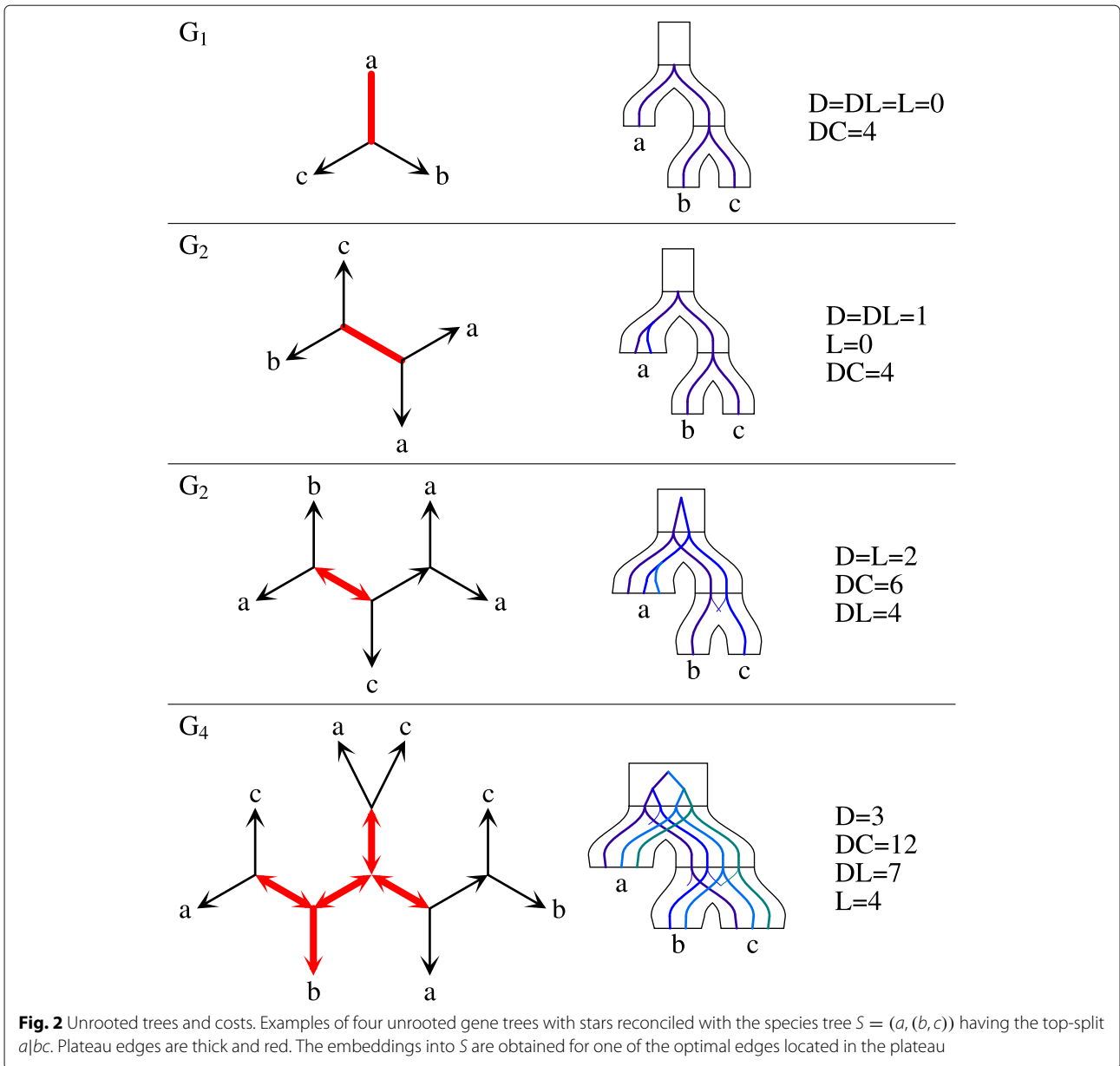
We start with the following observation, which follows from Theorem 1 and stars definitions.

**Lemma 1** *The top-split of the species tree is sufficient to determine the optimal rooting of a given unrooted gene tree.*



**Fig. 1** Stars S1-S5. The edges denote the relation  $\zeta$  as follows:  $A \rightarrow \bar{A}$  is  $\neg\zeta(A) \wedge \zeta(\bar{A})$ ,  $A \leftrightarrow \bar{A}$  is  $\neg\zeta(A) \wedge \neg\zeta(\bar{A})$  and  $A - \bar{A}$  is  $\zeta(A) \wedge \zeta(\bar{A})$ . The nodes whose clusters are  $B$  and  $C$  we call *right-hand* in a given star. A red edge is an element of the plateau, i.e., its rooting is optimal (see Thm. 1)





A *rooted subtree* of an unrooted gene tree  $G$  is a proper subtree of some rooting of  $G$ . Then, we have the following property.

**Lemma 2** Let  $T$  be a rooted subtree of an unrooted gene tree  $G$  and let  $S$  be a species tree over  $L(G)$ .  $T$  is a subtree in every plateau rooting of  $G$  if and only if the following conditions are satisfied:

- $M(t) < \text{root}(S)$ ,
- or  $M(t) = \text{root}(S)$  and  $t$  is a not a duplication,

where  $t = \text{root}(T)$ .

*Proof* ( $\Leftarrow$ ) The statement is obvious if  $t$  is a leaf. We show that no edge in  $T$  is symmetric in  $G$  if  $t$  is an internal node in  $G$ . If  $M(t) < \text{root}(S)$  then  $t$  is a center of  $S_1$  or  $S_2$  and the children of  $t$  in  $T$  are right-hand in the star. Thus, the remaining edges in  $T$  are asymmetric (only  $S_1$  can be present in  $T$ ) with the star-arrows directed towards the leaves of  $T$ . A similar property holds in the second case with the difference that  $t$  is a center of  $S_3$ . Thus, the edges of  $T$  are disjoint with the plateau and  $T$  is a rooted subtree in every plateau rooting of  $G$ . ( $\Rightarrow$ ) If  $t$  is mapped below the root of  $S$ , the tree contains only asymmetric non-plateau edges. Otherwise,  $t$  is a duplication mapped

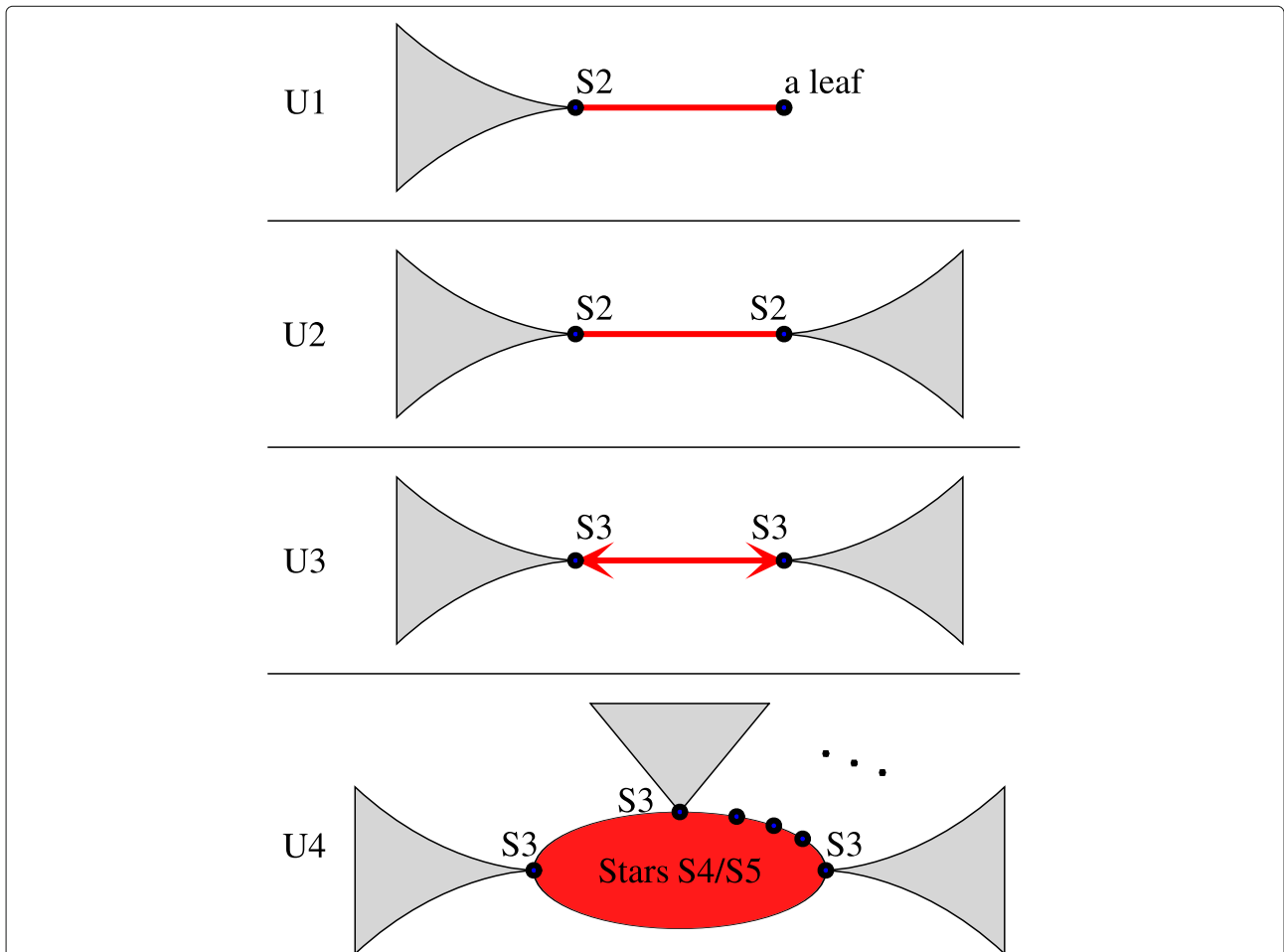
to the root, then at least one edge  $e'$  connecting  $t$  with its child is symmetric. Thus,  $T$  is not present in the rooting placed on  $e'$ . A contradiction.  $\square$

We conclude that every unrooted gene tree  $G$  can be decomposed into two parts: a *plateau*  $G^*$  and the *rooted forest*  $\check{G}$  obtained from  $G$  by removing the internal nodes and edges of  $G^*$ . Note that  $G^*$  is an unrooted tree with at least one edge and  $\check{G}$  is a forest whose edges are asymmetric in  $G$ . Moreover, if  $\check{G}$  contains a tree with a non-root internal node, then this node is a center of S1. The leaves of  $G^*$ , which are also the roots of  $\check{G}$ , are called *border nodes*. Since not every two stars can share an edge, possible topologies of stars in gene tree are limited. Figure 3 depicts all possible types of gene trees depending on the stars (see also Fig. 2).

**Cost contribution functions**

For an unrooted gene tree  $G$  let  $\text{stars}(G)$  be the multi-set of all stars present in  $G$  (similarly it is defined for

collections of gene trees). Here we define the cost contribution functions for our four standard costs. For every cost, we define  $\hat{\lambda}$  as the contribution of a given star to a species (i.e., a leaf of the species tree) in the context of a top-split,  $\hat{\gamma}$  as the contribution of a given star to a non-root internal node of a species tree (a split  $X|Y$ ) in context of a top-split,  $\sigma$  as the contribution of a given star to the root of a species tree in context of a top-split, and  $\epsilon$  which is the cost correction depending on the gene tree. For an unrooted gene tree  $G$ , a species  $s$  and a top-split  $\top$ ,  $\hat{\Lambda}(G, s, \top)$  is the total cost contribution of the stars (in the context of a top-split  $\top$ ) from  $G$  to a leaf of some species tree over  $L(G)$  labeled by  $s$ . Given two disjoint sets of species  $X$  and  $Y$  and a top-split  $\top$ ,  $\hat{\Gamma}(G, X|Y, \top)$  is the total cost contribution of the stars (in the context of a top-split  $\top$ ) from  $G$  to an internal node  $\nu$  of some species tree over  $L(G)$  such that (i) the cluster of  $\nu$  is  $X \cup Y$  and (ii)  $\nu$  has two children whose clusters are  $X$  and  $Y$ , respectively. Then, the total cost contributions are defined:



**Fig. 3** Types of unrooted trees. Four possible types of unrooted gene trees  $G$  (under the assumption that  $|L(G)| \geq 3$  and  $L(S) \geq 3$ ). Red color (in the center) denotes  $G^*$  (plateau) while grey subtrees denote the rooted forest  $\check{G}$ . Border nodes are marked by black circles and decorated with its possible star types (when applicable). See also Fig. 2

$$\hat{\Lambda}^c(G, s, \top) = \sum_{a \in \text{stars}(G)} \hat{\lambda}^c(a, s, \top),$$

$$\hat{\Gamma}^c(G, X|Y, \top) = \sum_{a \in \text{stars}(G)} \begin{cases} \sigma^c(a, \top) & X|Y = \top, \\ \hat{\gamma}^c(a, X|Y, \top) & \text{otherwise,} \end{cases}$$

where the contribution functions  $\hat{\lambda}^c$ ,  $\hat{\gamma}^c$  and  $\sigma^c$  for  $c \in \{D, DL, DC\}$  are depicted below. Here,  $\top$  is a top-split of some species tree  $S$ ,  $X|Y$  is a split of some node from  $S$ ,  $s$  is a species,  $G$  is an unrooted gene tree and  $\mathcal{A} = A|B|C$  is a star of type  $\tau$  from  $G$ .

- Duplication cost (D):

$$\hat{\lambda}^D(\mathcal{A}, s, \top) = \mathbf{1}[B = C = \{s\} \wedge \tau \in \{S1, S2\}] \tag{2}$$

$$\hat{\gamma}^D(\mathcal{A}, X|Y, \top) = \gamma^D(B, C, X|Y) \tag{3}$$

$$\epsilon^D(G) = 1. \tag{4}$$

$$\sigma^D(\mathcal{A}, \top) = \begin{cases} 1 & \tau \in \{S4, S5\} & (5a) \\ -1 & \tau = S2 \text{ and } |A| = 1 & (5b) \\ -0.5 & \tau = S2 \text{ and } |A| \neq 1 \neq |\bar{A}| & (5c) \\ 0 & \text{otherwise} & (5d) \end{cases}$$

- Deep Coalescence cost (DC):

$$\hat{\lambda}^{DC}(\mathcal{A}, s, \top) = \begin{cases} \lambda^{DC}(B|C, s) & \text{if } \tau \in \{S1, S3\} \\ \lambda^{DC}(A|\bar{A}, s) & \text{if } \tau = S5 \\ \max\{\lambda^{DC}(B|C, s), \lambda^{DC}(A|\bar{A}, s)\} & \text{if } \tau = S2 \wedge \neg(B=C \wedge |B|=1) \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{\gamma}^{DC}(\mathcal{A}, X|Y, \top) = \begin{cases} \gamma^{DC}(B, C, X|Y) & \tau \in \{S1, S3\} \\ \gamma^{DC}(A, \bar{A}, X|Y) & \tau = S5 \\ \gamma^{DC}(B, C, X|Y) + \frac{1+|A|=1}{2} \gamma^{DC}(A, \bar{A}, X|Y) & \tau = S2 \text{ and } \neg(B=C \wedge |B|=1) \\ 0 & \text{otherwise} \end{cases}$$

$$\sigma^{DC}(\mathcal{A}, \top) = 0,$$

$$\epsilon^{DC}(G) = 0.$$

- Loss and Duplication-Loss cost; for  $c \in \{L, DL\}$ , let  $\omega_L = 2$  and  $\omega_{DL} = 3$  in

$$\hat{\lambda}^c(\mathcal{A}, s, \top) = \hat{\lambda}^{DC}(\mathcal{A}, s, \top) + \omega_c \cdot \hat{\lambda}^D(\mathcal{A}, s, \top)$$

$$\hat{\gamma}^c(\mathcal{A}, X|Y, \top) = \hat{\gamma}^{DC}(\mathcal{A}, X|Y, \top) + \omega_c \cdot \hat{\gamma}^D(\mathcal{A}, X|Y, \top)$$

$$\sigma^c(\mathcal{A}, \top) = \sigma^{DC}(\mathcal{A}, \top) + \omega_c \cdot \sigma^D(\mathcal{A}, \top)$$

$$\epsilon^c(G) = 1 - |V_G| + \omega_c$$

The next theorem defines how these functions can be used to compute the unrooted cost given an unrooted gene tree and a species tree. This result is the unrooted analogue of Eq. (1).

**Theorem 2** Let  $G$  be an unrooted gene tree with  $|L(G)| \geq 3$  and  $S$  be a species tree over  $L(G)$ . If  $c$  is a cost function  $\{D, DL, L, DC\}$  and  $\top$  is the top-split of  $S$ , then

$$c(G, S) = \sum_{s \in L(S)} \hat{\Lambda}^c(G, s, \top) + \sum_{x, y \text{ siblings in } S} \hat{\Gamma}^c(G, clu(x)|clu(y), \top) + \epsilon^c(G).$$

*Proof* Let  $R$  denote the right-hand side of the above formula. We start with the duplication cost and gene trees of type U4.

**Cost D vs. U4:** The plateau of  $G$  contains stars S4/S5. Then, the remaining stars have type S1/S3. We show that  $D(G_e, S) = R$ , where  $e$  is a plateau edge. Let  $g$  be an internal node of  $G_e$ ,  $M(g) = s$  and, if  $g \neq \text{root}(G_e)$ , then  $g$  is a center of star  $A|B|C$  of type  $\tau$ .

(U4.a) If  $s$  is a leaf. Then,  $g$  is a duplication and  $\tau$  is S1 with  $B = C = \{s\}$ . The case is incorporated in  $\hat{\lambda}^D$  (see (2)). Note that  $\tau$  cannot be S3/S4/S5 under the assumption that  $L(S) \geq 3$ .

(U4.b) If  $s$  is an internal non-root node of  $S$  with the split  $X|Y$ . In this case,  $B \cup C \subseteq X \cup Y$  and  $X \cup Y$  is a subset of the element from  $\top$ . Then,  $\tau = S1$ . In such a case,  $g$  is a duplication if and only if  $\gamma^D(B|C, X|Y, \top) = 1$  (See (3), where  $\gamma^D$  was defined in “[Solution to rMinST and rMinCC - overview](#)” section).

(U4.c) The remaining case is when  $s = \text{root}(S)$ . We have three subcases:  $\tau = S3$  (border of  $G^*$ ),  $\tau \in \{S4, S5\}$  (internal of  $G^*$ ) or  $g = \text{root}(G_e)$ . If  $\tau = S3$ , then,  $g$  is a border node and by Lemma 2, it is not a duplication. In our formula, the contribution of stars S3 is 0. If  $\tau \in \{S4, S5\}$  then  $g$  is a duplication. This case is contributed in (5a). Finally, if  $g = \text{root}(G_e)$  then it is also a duplication counted in  $\epsilon^D(G)$ . Note that, given a plateau with  $n$  internal nodes (or equivalently,  $n$  stars S4/S5), the total number of duplications at the root of  $S$ , equals  $n + 1$  (one duplication for each star S4/S5 plus one for the root of  $G$ ). Thus, our formulas count exactly the number of duplications in such a case.

This completes the proof for gene trees of the type U4.

**Gene tree type U3:** The proof is almost the same as in the previous case. The only difference is that the plateau consists of a single edge ( $n = 0$ ; no stars in the plateau). Thus, the third subcase in (U4.c) can be omitted.

**Gene tree type U1-U2:** Now, the gene tree contains one or two stars S2 (sharing an edge) and the remaining stars are of type S1 (if present).

(U1-2.a) If  $s$  is a leaf, then  $\tau$  is S1 or S2. Then,  $g$  is a duplication and the rest is similar to (U4.a).

(U1-2.b) The same as (U4.b) plus the case when  $\tau = S2$ .

(U1-2.c) If  $g = \text{root}(G_e)$ , then it is not a duplication, and its contribution is 0.

Since there is no duplication mapped to  $\text{root}(S)$  if  $G$  contains S2, the above cases count every duplication present in  $G_e$ . However, the presence of  $\epsilon^D(G) = 1$  which is needed for proper calculation in other cases, requires correction of the total contribution. The correction by  $-1$  is only performed when S2 is present in  $G$ , but the difficulty



is that  $G$  can have one or two such stars. This correction is embedded in (5b) and (5c) as follows. Assume that the first star  $S_2$  is  $A|B|C$ . If the second star  $S_2$  is present we assume it is  $\bar{A}|B'|C'$ , where  $B' \cup C' = A$ .

- $|A| = 1$  and  $G$  has only one star  $S_2$ . Then, the star contributes  $-1$  in (5b).
- $|A| = 1$  and  $G$  has two stars  $S_2$ . The first star contributes  $-1$  from (5b), while the second star contributes  $0$  from (5d).
- $|A| \neq 1 \neq |\bar{A}|$ . In such a case  $G$  has two stars  $S_2$ , each contributes  $-0.5$  from (5c).

Thus, the total correction is  $-1$  in every case. This completes the proof for the case of duplication cost.

**Cost DC:** The technical proof for DC is analogous to D.

**Cost DL and L:** Note that the formulas for DL and L are linear combinations of D and DC with the scalar addition of  $1 - |V_G|$  (see “Cost functions” section). Further, the optimal rootings are shared across all costs; therefore, the proof for this case follows from D and DC proofs and the corresponding contribution formulas.  $\square$

**Dynamic programming solution to uMinCC**

Let  $Q$  be a collection of unrooted gene trees,  $c$  be a cost function and  $Z \subseteq L(Q)$ . Here we extend  $\hat{\Gamma}$ ,  $\hat{\Lambda}$  and  $\epsilon$  to collections of unrooted gene trees, e.g.,  $\hat{\Gamma}(Q) = \sum_{G \in Q} \hat{\Gamma}(G)$  and so on. The dynamic programming formulas for the solution to the uMinCC problem is defined as follows (the superscript  $c$  is omitted):

$$\Upsilon(Q, Z, \top) = \begin{cases} \hat{\Lambda}(Q, s, \top) & \text{if } Z = \{s\}, \\ \min_{X|Y \in \text{splits}(Z)} \Upsilon^*(Q, X|Y, \top) & \text{otherwise,} \end{cases}$$

$$\Upsilon^*(Q, X|Y, \top) = \Upsilon(Q, X, \top) + \Upsilon(Q, Y, \top) + \hat{\Gamma}(Q, X|Y, \top).$$

Informally, in the above recurrence, for each top-split  $X|Y$  and for each  $Z$  such that  $Z \subseteq T$  or  $Z \subseteq T'$ ,  $\Upsilon(Q, Z, \top)$  is the minimal cost contribution of a non-plateau parts of input gene trees in the set of all species trees over  $L(Q)$  having a node  $v$  whose cluster is  $Z$ , where the contribution is only calculated for the nodes strictly descendant from  $v$ . The formula for  $\Upsilon^*(Q, X|Y, \top)$  extends the  $\Upsilon$  by including the contribution of the nodes whose split equals  $X|Y$ . Note that the contribution of the plateau is incorporated in  $\Upsilon^*$  in the special case when  $X|Y = Q$  (see def. of  $\hat{\Gamma}$ ).

**Theorem 3** *Let  $c$  be a cost function and  $Q$  be a collection of unrooted gene trees such that  $|L(Q)| \geq 3$ , and  $L(Q) = L(G)$  for every gene tree  $G$  from  $Q$ . Then, the solution to uMinCC is  $\min_{\top \in \text{splits}(L(Q))} \Upsilon^*(Q, \top, \top) + \epsilon(Q)$ .*

*Proof* Without loss of generality, we may assume that  $Q$  consists of one gene tree  $G$ . Given a species tree  $S$  with top-split  $\top$  and having a node  $s$  with a cluster  $Z$ , by par-

tial  $Z$ -contribution of  $S$  we denote the partial cost defined recursively as follows:

$$c_Z(G, S) = \begin{cases} \hat{\Lambda}^c(G, s, \top) & \text{if } Z = \{s\}, \\ c_X(G, S) + c_Y(G, S) + \hat{\Gamma}^c(G, X|Y, \top) & \text{if } X|Y \text{ is the split of } s. \end{cases}$$

It follows from Theorem 2, that  $c(G, S) = c_{\top}(G, S) + \epsilon^c(G)$ .

Let  $\top = T|T'$ . It is sufficient to prove that, for each  $Z \subseteq T$  or  $Z \subseteq T'$ ,  $\Upsilon^c(Q, Z, \top)$  is the minimal partial  $Z$ -contribution in the set of all species trees over  $L(Q)$  having a node whose cluster is  $Z$ . Then, it follows that  $\Upsilon^*(Q, \top, \top)$  is the minimal partial  $\top$ -contribution in the set of all species trees over  $L(Q)$  having a top-split  $\top$ .

The proof follows by induction from Theorem. 2. We omit technical details.  $\square$

Solving uMinST follows directly from the values of  $\Upsilon$ : it is sufficient to track which partitions of the cluster  $Z$  into  $X$  and  $Y$  under a given top-split yield the minimal value. Such partitions determine clusters in optimal species trees and they can be used to infer one or all optimal species trees.

**Theorem 4 (Complexity)** *Given a collection of  $Q$  unrooted gene trees with  $m$  leaves and  $n$  species. The time complexity of the dynamic programming formula is  $O(nm3^n)$  and the space complexity is  $O(3^n + m)$ .*

*Proof* We have the following identities:  $|\text{stars}(Q)| = m - 2|Q|$ , the number of splits of  $|X| = k$  equals  $\frac{1}{2} \sum_{k=1}^{n-1} \binom{n}{k} = 2^{k-1} - 1$ . Then, computing stars requires  $O(m)$  time, while each value of  $\lambda$ ,  $\gamma$ , etc. requires  $O(n)$  time. Thus, a single  $\hat{\Gamma}$  or  $\hat{\Lambda}$  computation needs  $O(mn)$  time. Note that  $Z$  must be a nonempty subset of an element from  $\top$  in  $\Upsilon$ . Thus, the size of  $\Upsilon$  array is  $\sum_{k=1}^{n-1} \binom{n}{k} (2^k - 1) \approx 3^n$ . Similarly, the size of  $\hat{\Gamma}$  array is  $\sum_{k=1}^{n-1} \binom{n}{k} (2^{k-1} - 1) \approx 3^n$ . Thus, the time complexity of the algorithm is  $O(mn3^n)$  and the space complexity is  $O(3^n + m)$ .  $\square$

Now, we compare the above complexity with the naïve approach i.e., try all rooting variants to compute the minimum cost. We may assume that every gene tree has the size of  $n$ . Then, we have  $2n - 3$  rooting variants of a single gene tree. As the REXACT requires  $O(mn3^n)$  time, we need  $O((2n - 3)^m mn3^n)$  time for the naïve algorithm.

**The general case when  $L(G) \subseteq L(S)$**

When  $L(G) \subseteq L(S)$ , the split needed to determine the star type can be located below the root of  $S$ . For example, the star  $ab|c|d$  cannot be determined for the split  $abcde|f$ , since the set of labels of the star is a subset of an element of the split. To determine the type of a star  $A|B|C$  the species tree split  $X|Y$  has to satisfy the following condition:  $L \subseteq X \cup Y$ ,  $L \cap X \neq \emptyset$  and  $L \cap Y \neq \emptyset$ , where

$L = A \cup B \cup C$ . The split  $X|Y$  satisfying above condition will be called a *rooting split* for  $L$ . Then, Theorem 2 can be reformulated as follows

**Theorem 5** Let  $G$  be an unrooted gene tree with  $|L(G)| \geq 3$  and  $S$  be a species tree such that  $L(G) \subseteq L(S)$ . If  $c$  is a cost function  $\{D, DL, L, DC\}$  and  $T$  is a split in  $S$  such that  $T$  is a rooting split for  $L(G)$ , then

$$c(G, S) = \sum_{s \in L(S)} \hat{\Lambda}^c(G, s, T) + \sum_{x, y \text{ siblings in } S} \hat{\Gamma}^c(G, clu(x)|clu(y), T) + \epsilon^c(G).$$

*Proof* Note that the rooting split is uniquely determined. The rest follows from the proof of Theorem 2.  $\square$

It follows from the above theorem that the rooting split is crucial for proper computation of the cost. Therefore, the dynamic programming formula has to be modified to capture possible different rooting splits. Let  $\tau$  be the set of pairs  $(a, R)$ , where  $a$  is a star and  $R$  is its rooting split. Then, we have the following dynamic programming formula:

$$\Upsilon(Q, Z, \tau) = \begin{cases} \sum_{(a, R) \in \tau} \lambda^c(a, s, R) & \text{if } Z = \{s\}, \\ \min_{X|Y \in \text{splits}(Z)} \Upsilon^*(Q, X|Y, \tau') & \text{otherwise,} \end{cases}$$

$$\begin{aligned} \Upsilon^*(Q, X|Y, \tau) &= \Upsilon(Q, X, \tau) + \Upsilon(Q, Y, \tau) \\ &+ \sum_{(q, X|Y) \in \tau} \sigma(a, X|Y) + \\ &+ \sum_{(q, R) \in \tau, R \neq X|Y} \gamma(a, X|Y, R), \end{aligned}$$

where  $\tau' = \tau \cup \{(a, X|Y) : \text{if } X|Y \text{ is a rooting split for } a \in \text{stars}(Q)\}$ . Now, the time complexity of the algorithm is  $O(mn3^{2n})$  as the  $\tau$  is an additional component with  $O(m3^n)$  possible values.

Similarly, we can adopt the dynamic programming algorithm to solve the species tree inference using the minus method in which the cost between a gene tree  $G$  and a species tree  $S$  such that  $L(G) \subseteq L(S)$  is defined as  $c(G, S|_{L(G)})$ , where  $S|_{L(G)}$  is the species tree obtained from  $S$  by contracting the set of species in  $S$  to  $L(G)$ . Then, the above dynamic programming algorithm solves the problem under the minus method setting; however, the contribution formulas require modification. For the rooted component, the formulas ( $\lambda$  and  $\gamma$ ) are provided in [28], while for the plateau component ( $\sigma$ ) they remain unchanged for all our cost functions.

## Results

### Scalability analysis

We present a scalability study that compares the runtimes of our proposed dynamic programming method

with the previously best-known (naïve) method to solve the median tree reconciliation problems for unrooted input gene trees. Recall that, as presented in the introduction, the naïve method is enumerating all possible combinations of rootings of the input trees and solving the respective rooted problem for each such combination. The number of such combinations grows exponentially fast with the increase of the number of input trees (i.e.,  $n^k$ , where  $n$  is the number of taxa and  $k$  is the number of trees). We (i) showcase that our method is significantly more scalable than the naïve approach and then (ii) we study the scalability of our method under conditions, which are infeasible for the naïve method.

### Experimental setup

Both the developed here *unrooted* exact method and the previously introduced *rooted* exact method [28] were implemented in Python. Recall that the naïve method uses the rooted exact method as a subroutine. The overall naïve method was implemented in Python as well.

The scalability studies were performed under Ubuntu 14.04.5 LTS with 1.4GHz CPUs.

### Discussion

To compare the implementations of UREXACT and the naïve method we study the scalability on input instances with a fixed number of taxa ( $n = 9$ ) and a varying number of input trees ( $k \in \{2, 3, 4, 5\}$ ). The input trees for each such instance were generated randomly using the pure-birth process; the roots were then removed from trees. For consistency, we generated five datasets for each different  $k$ .

The scalability results (runtimes averaged over five datasets) are presented in Table 1. Note that for  $k = 2$  and  $k = 3$  the naïve approach proved to be somewhat faster than UREXACT (this can be explained by ). However, already for  $k = 5$ , the exponential run-time growth of the naïve approach resulted in the approximately 120-fold advantage of the UREXACT method. Further, naïve method could not complete for  $k = 6$  in under 24 hours.

As shown above the naïve method is infeasible for more than five input trees; it is important to note that most of phylogenetic studies in practice involve significantly more than five input trees. Next, we present the scalability study of UREXACT under more realistic conditions. Namely, we generated several input datasets with the number of trees

**Table 1** Average runtimes of the UREXACT and naïve methods; (s) stands for seconds, (m) for minutes, and (h) for hours;  $k$  denotes the number of trees

	$k = 2$	$k = 3$	$k = 4$	$k = 5$
UREXACT	75.9(s)	113.8(s)	2.6(m)	0.05(h)
naïve	7.7(s)	111.7(s)	26.1(m)	6(h)

$k = 20$  and  $k = 100$  and the number of taxa varying from  $n = 5$  to  $n = 11$ . Similar to the previous study, to present consistent results we generated five datasets for each combination of  $k$  and  $n$  and averaged the runtime of our method over those five instances; the input trees were generated using the pure-birth process.

The average runtime results are depicted in Table 2. It shows that our method was able to complete computations of exact median trees for all generated datasets. In the worst case ( $n = 11$  and  $k = 100$ ) UREXACT took about 26 hours to complete. Note that Tables 1 and 2 confirm that the scalability of the presented method grows *linearly* with the number of trees (as implied by the asymptotic analysis in “Dynamic programming solution to uMinCC” section). This is a significant achievement, given that the naïve approach assumes exponential growth.

### Accuracy of heuristics

Local search heuristics have become a standard when addressing median tree problems (e.g., [20, 46–49]). Consequently, analyzing the accuracy of such heuristics is of critical importance. Recent algorithmic advances in exact computing solutions for median tree reconciliation problems made it possible to evaluate the accuracy of local search heuristics addressing these problems [31, 50]. However, these algorithmic advances address only tree reconciliation problems for rooted input trees. Our presented algorithms solve the tree reconciliation problems exactly for unrooted trees with up to 11 taxa (see “Scalability analysis” section). Thus, for the first time, these algorithms (implemented in UREXACT) make it possible to analyze the accuracy of local search heuristics for reconciliation problems for unrooted input trees suitable for smaller phylogenetic studies. Here, we analyze the accuracy of URHEUR [37], one of the theoretically most advanced local search heuristics for unrooted tree reconciliation problems that are implementing the elementary “valley-terraces theorems” [38].

A local search heuristic initiates its median tree search for a given set of input trees under some distance from an additionally provided starting tree. The heuristic searches the local neighborhood of the candidate tree for a tree with the minimum overall distance to the input trees, which constitutes a local search step. Then, the tree found in a local search step becomes the starting point for

the following local search step, and so on, until a local minimum is reached, which is reported by the heuristic. The local neighborhood of a tree is the set of trees into which it can be transformed by applying at most one tree edit-operation of a fixed type.

URHEUR implements the above described local search heuristic for median tree reconciliation problems under the gene duplication (D) and the duplication-loss (DL) costs, using the classic Rooted Nearest Neighbor Interchange tree edit operation for rooted trees [51] to define the local neighborhood. Further, URHEUR offers the option to choose the starting tree to be randomly selected under the Yule random tree model, called *random mode*, or using a cluster-based approach called *clusters mode*. The clusters mode selects the clusters for the starting tree following the greedy consensus strategy when generalized to incomplete input trees. If the resulting tree is not binary, then the tree is binarized randomly by adding clusters under the Yule model.

### Experimental setup

Similarly to the scalability study in “Scalability analysis” section, we generated 5 independent gene tree sets (datasets) for each  $n = 5, 6, \dots, 11$  (where  $n$  indicates the number of taxa). Each such dataset contains 20 trees generated via the pure-birth process.

Since URHEUR is a randomized local search method that performs a single local search run, we executed it 10000 times on each dataset. Additionally, we executed UREXACT on each dataset. That way we were able to evaluate the *accuracy* of URHEUR. That is, for each dataset we computed how many times (out of the 10000 runs) did the heuristic find a *global optimum* (i.e., a true median tree). We report the resulting percentage (averaged over 5 independent datasets for each  $n$ ) as the *accuracy* of the heuristic.

### Results and discussion

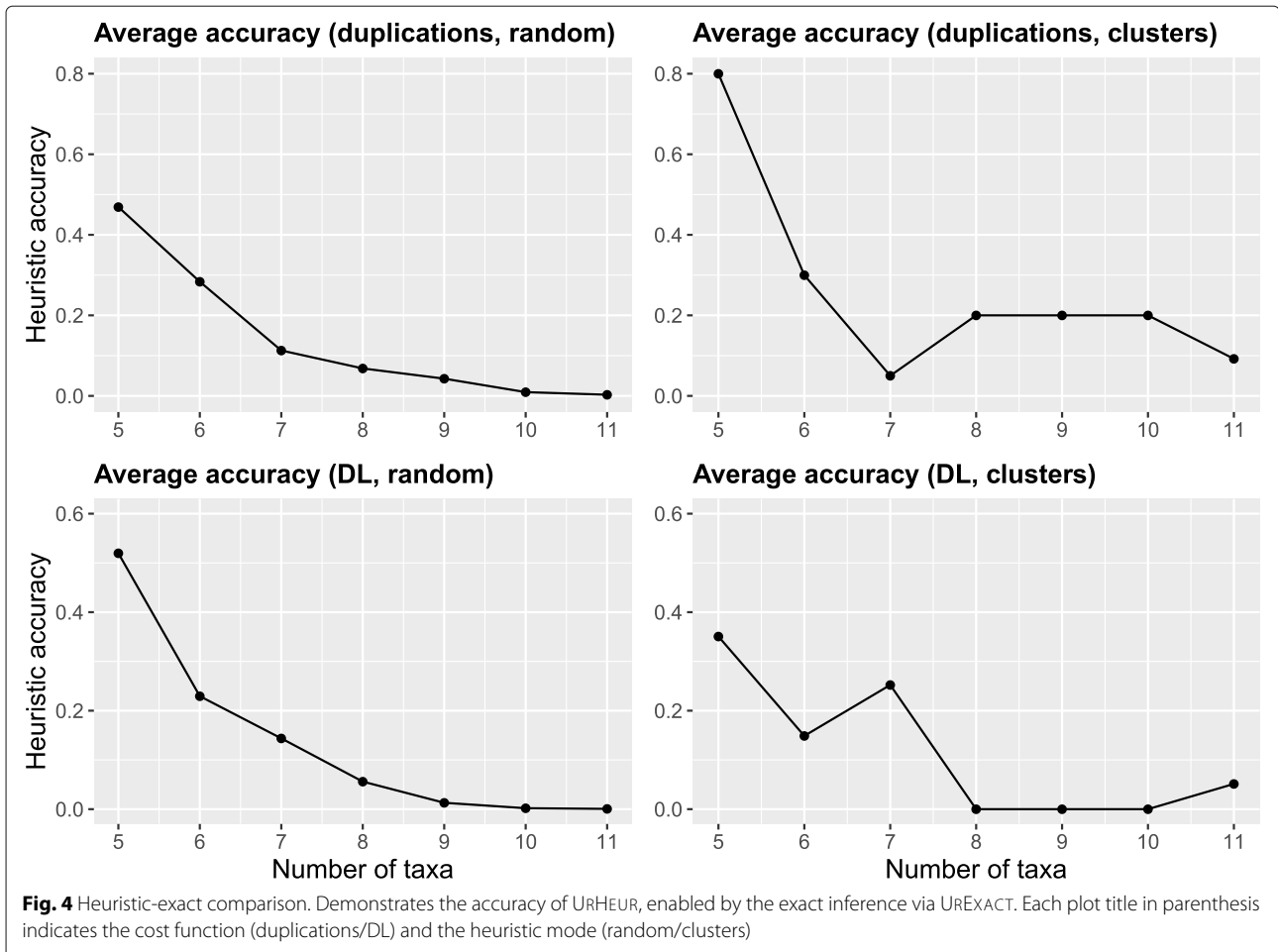
Figure 4 showcases the obtained results, which were compute using a standard laptop in less than 24 hours. Note that we performed our comparison on the examples of the classic duplication (D) and duplication-loss (DL) costs. Further, we executed the heuristic in the modes random and cluster.

Figure 4 demonstrates that, while the accuracy of the heuristic is quite high for smaller taxa numbers, it quickly deteriorates. That is, it typically requires a large number of independent runs of URHEUR to find a global optimum for  $n \geq 9$ . Further, URHEUR is never guaranteed to find a global optimum; for example, it never does so in the (DL, clusters) condition for  $n \in \{8, 9, 10\}$ .

The trends shown in Fig. 4 suggest that, for larger number of taxa, it is crucial to have exact solutions, since the heuristic is never guaranteed to find a global optimum.

**Table 2** Average runtimes of the UREXACT method; (s) stands for seconds, (m) for minutes, and (h) for hours;  $n$  denotes the number of taxa

	$n = 5$	$n = 6$	$n = 7$	$n = 8$	$n = 9$	$n = 10$	$n = 11$
$k = 20$	0.7(s)	4.1(s)	24.9(s)	154.2(s)	11.7(m)	66.8(m)	7.8(h)
$k = 100$	3.1(s)	19.3(s)	116.2(s)	678.8(s)	62.1(m)	314.7(m)	25.7(h)



## Discussion

In this work, we presented the first method that is capable of computing exact median trees (under tree reconciliation costs) from collections of unrooted gene trees. In particular, we introduced a novel algorithm that solves median tree reconciliation problems under the gene duplication, deep coalescence, gene loss, and gene duplication-loss cost functions. This dynamic programming solution represents the first step towards enabling the efficient inference of large-scale median trees using the cluster-constrained paradigm (for more details, see ASTRAL [52]).

Our method, UREXACT, represents a complex dynamic programming approach; it builds on top of the previously introduced REXACT method [28] via employing, and extending, the plateau properties (see “Unrooted tree reconciliation costs” section). The main observation used for the design of the algorithm is that an optimum rooting of an input gene tree only depends on the top two clusters of the species tree (i.e., the clusters/clades of the two nodes immediately below the root); we refer to such top two clusters as the *top-split*. Then the dynamic

programming formulas were expressed through a comprehensive analysis of different configurations of plateaus (and the corresponding *star types* [38]) that could appear in the input gene trees. For each unrooted reconciliation cost function, we propose a novel way to compute its cost using contribution functions that depend only the star and the split from the species tree. Having this, we design a unified dynamic programming algorithm, that can be parametrized by the set of contribution functions. Given the intricacy of the contribution functions’ formulations, we provide a formal proof of their correctness focusing mainly on the gene duplication cost function, allowing their verification.

It is important to note that our method UREXACT runs in  $O(mn3^n)$  time for complete input gene trees, which is asymptotically the same as for REXACT. We show how to handle the minus method in “The general case when  $L(G) \subseteq L(S)$ ” section, adding a factor of  $O(m3^n)$  to the overall time complexity.

Further, we showcase two scalability studies demonstrating the ability of UREXACT. In the first study, we compare UREXACT with the naïve method (implemented



via enumerating all possible rootings of the input trees) and demonstrate that the naïve approach becomes infeasible when the number of input trees exceeds five (which is a very small number of trees for studies in practice). The second study demonstrates the scalability of UREXACT under more realistic conditions – we fix the number of input trees to be either 20 or 100. UREXACT was able to compute exact median trees with up to 11 taxa.

Finally, we evaluated the significance of UREXACT in comparison to an inexact, but fast, local search heuristic, URHEUR [37]. In particular, having the exact median trees computed by UREXACT, we were able to evaluate the accuracy of URHEUR on up to 11 taxa. Our results suggest that it is crucial to develop exact dynamic programming solutions for precise evolutionary inference, given that the accuracy of local search heuristics quickly deteriorates.

## Conclusion

Supertree approaches, and in particular median tree approaches, proved to be valuable tools for inference of phylogenetic species trees. Unfortunately, to the best of our knowledge, there is no median tree problem of interest that was shown to be efficiently solvable (i.e., in polynomial time). Therefore, in practice, median tree problems are typically addressed by using local search heuristics. Despite the vast popularity and often appealing scalability of such heuristics, they have several significant drawbacks. The most concerning drawback is that there is typically no guarantee on the quality of supertrees found by such heuristics; that is, computed supertrees can be arbitrarily far from the truth.

To overcome the drawbacks of local search heuristics, the method REXACT that solves median tree reconciliation problems has been proposed previously. REXACT is guaranteed to find the best median trees under a tree reconciliation cost function of interest. To do that REXACT requires an input of *rooted* gene trees. In practice, however, gene trees are typically unrooted. This factor incurs an additional layer of complexity to solving tree reconciliation problems. That is, since the tree reconciliation costs are defined for rooted gene trees, solving tree reconciliation problems naïvely (using REXACT) would require a full enumeration of all rooting combinations of the input trees. Such an approach is infeasible both in theory and in practice – as our scalability study shows.

We propose an exact novel method, UREXACT, that is capable of computing exact median trees for collections of unrooted gene trees. The dynamic programming algorithm developed in this work is enabled by the crucial so-called *plateau properties* [38]. Despite the intricacy of the developed algorithm, the asymptotic runtime of UREXACT is the same as the asymptotic runtime of REXACT when input gene trees are complete. Perhaps, most

surprisingly, this implies that the runtime of UREXACT grows *linearly* as the number of input trees grows.

Finally, note that the proposed dynamic programming formulation can be now applied to much larger datasets (i.e., with hundreds of taxa) using the *cluster-constrained approach*. This approach entails selecting a set of clusters that are likely to appear in the “true” species tree, and constraining the dynamic programming procedure to work only with the selected clusters. This approach proved to be highly efficient, for example, with the popular ASTRAL software [52, 53]. The implementation of this powerful idea is part of our future research efforts.

## Abbreviations

D: Duplication; L: Loss; DL: Duplication and loss; DC: Deep coalescence; uMinST: Unrooted minimal species tree; uMinCC: Unrooted minimal cost computation; rMinST: Rooted minimal species tree; rMinCC: Rooted minimal cost computation

## Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable suggestions and comments.

## About this supplement

This article has been published as part of *BMC Evolutionary Biology Volume 20 Supplement 1, 2020: Selected articles from the 18th Asia Pacific Bioinformatics Conference (APBC 2020): evolutionary biology*. The full contents of the supplement are available online at <https://bmcevolbiol.biomedcentral.com/articles/supplements/volume-20-supplement-1>.

## Authors' contributions

PG developed the algorithm, proved its properties and provided Python implementation. AM performed experimental evaluations and analyzed the data. OE formulated the problems. All authors contributed to the writing. All authors read and approved the final manuscript.

## Funding

PG has been supported by the National Science Center grants 2017/27/B/ST6/02720 and 2015/19/B/ST6/00726. The work of AM and OE have been supported by the National Science Foundation under Grant No. 1617626. Publication costs are funded by the National Science Center grant 2017/27/B/ST6/02720.

## Availability of data and materials

The software package is available at <https://bitbucket.org/pgor17/urexact>.

## Ethics approval and consent to participate

Not applicable.

## Consent for publication

Not applicable.

## Competing interests

The authors declare that they have no competing interests.

## Author details

<sup>1</sup>University of Warsaw, Faculty of Mathematics, Informatics and Mechanics, Banacha 2, 02-097 Warsaw, Poland. <sup>2</sup>Department of Computer Science, Iowa State University, Atanasoff Hall 212, 50011 Ames, USA.

Published: 28 October 2020

## References

- Cracraft J, Donoghue MJ. *Assembling the Tree of Life*. New York: Oxford University Press; 2004.



2. Nik-Zainal S, Van Loo P, Wedge DC, Alexandrov LB, Greenman CD, Lau KW, Raine K, Jones D, Marshall J, Ramakrishna M, et al. The life history of 21 breast cancers. *Cell*. 2012;149(5):994–1007.
3. Hufbauer RA, Marrs RA, Jackson AK, Sforza R, Bais HP, Vivanco JM, Carney SE. Population structure, ploidy levels and allelopathy of *Centaurea maculosa* (spotted knapweed) and *C. diffusa* (diffuse knapweed) in North America and Eurasia. In: Proceedings of the XI International Symposium on Biological Control of Weeds, Canberra Australia. Morgantown, WV: USDA Forest Service. Forest Health Technology Enterprise Team; 2003. p. 121–6.
4. Roux JLL, Wicczorek AM, Ramadan MM, Tran CT. Resolving the native provenance of invasive fireweed (*Senecio madagascariensis* Poir.) in the Hawaiian Islands as inferred from Poir. in the Hawaiian Islands as inferred from phylogenetic analysis. *Divers Distrib*. 2006;12:694–702.
5. Harris SR, Cartwright EJP, Török ME, Holden MTG, Brown NM, Ogilvy-Stuart AL, Ellington MJ, Quail MA, Bentley SD, Parkhill J, Peacock SJ. Whole-genome sequencing for analysis of an outbreak of methicillin-resistant *Staphylococcus aureus*: a descriptive study. *Lancet Infect Dis*. 2013;13(2):130–6.
6. Forster P, Renfrew C. *Phylogenetic Methods and the Prehistory of Languages*. Cambridge, England, United Kingdom: McDonald Inst. of Archaeol. Reas; 2006.
7. Cotton JA, Page RDM. Rates and patterns of gene duplication and loss in the human genome. *Proc R Soc B Biol Sci*. 2005;272(1560):277–83.
8. Page RDM. GeneTree: comparing gene and species phylogenies using reconciled trees. *Bioinformatics*. 1998;14(9):819–20.
9. Lynch M, Conery JS. The evolutionary fate and consequences of duplicate genes. *Science*. 2000;290(5494):1151–5.
10. Ohno S. *Evolution by Gene Duplication*. Berlin: Springer; 1970.
11. Bininda-Emonds ORP, Gittleman JL, Steel MA. The (super)tree of life: Procedures, problems, and prospects. *Annu Rev Ecol Syst*. 2002;33(1):265–89.
12. Bininda-Emonds ORP, Vol. 4. *Phylogenetic supertrees: combining information to reveal the tree of life*: Springer; 2004.
13. Bonizzoni P, Della Vedova G, Dondi R. Reconciling a gene tree to a species tree under the duplication cost model. *Theor Comput Sci*. 2005;347(1–2):36–53.
14. Eulenstein O, Huzurbazar S, Liberles DA. Reconciling phylogenetic trees. *Evolution after Gene Duplication*. 2010, pp. 185–206.
15. Goodman M, Czelusniak J, Moore GW, Romero-Herrera AE, Matsuda G. Fitting the gene lineage into its species lineage. A parsimony strategy illustrated by cladograms constructed from globin sequences. *Syst Zool*. 1979;28(2):132–63.
16. Górecki P, Tiuryn J. DLS-trees: A model of evolutionary scenarios. *Theor Comput Sci*. 2006;359(1–3):378–99.
17. Maddison WP. Gene trees in species trees. *Syst Biol*. 1997;46:523–36.
18. Ma B, Li M, Zhang L. From gene trees to species trees. *SIAM J Comput*. 2000;30(3):729–52.
19. Zhang L. From gene trees to species trees II: Species tree inference by minimizing deep coalescence events. *IEEE/ACM Trans Comput Biol Bioinform*. 2011;8(6):1685–91.
20. Than C, Nakhleh L. Species tree inference by minimizing deep coalescences. *PLoS Comput Biol*. 2009;5(9):1000501.
21. Bansal MS, Eulenstein O. Algorithms for genome-scale phylogenetics using gene tree parsimony. *IEEE/ACM Trans Comput Biol Bioinform*. 2013;10(4):939–56.
22. Wehe A, Burleigh JG, Eulenstein O. Efficient algorithms for knowledge-enhanced supertree and supermatrix phylogenetic problems. *IEEE/ACM Trans Comput Biol Bioinform*. 2013;10(6):1432–41.
23. Cotton JA, Page RDM. Going nuclear: gene family evolution and vertebrate phylogeny reconciled. *Proc Biol Sci*. 2002;269(1500):1555–61.
24. Martin AP, Burg TMM. Perils of paralogy: using HSP70 genes for inferring organismal phylogenies. *Syst Biol*. 2002;51(4):570–87.
25. Page RDM. Extracting species trees from complex gene trees: reconciled trees and vertebrate phylogeny. *Mol Phylogenet Evol*. 2000;14:89–106.
26. Felsenstein J. The number of evolutionary trees. *Syst Zool*. 1978;27(1):27–33. <http://oup.prod.sis.lan/sysbio/article-pdf/27/1/27/4599600/27-1-27.pdf>.
27. Stadler PF. *Fitness landscapes*. In: *Biological Evolution and Statistical Physics*. Berlin, Heidelberg: Springer; 2002. p. 183–204.
28. Chang W-C, Górecki P, Eulenstein O. Exact solutions for species tree inference from discordant gene trees. *J Bioinform Comput Biol*. 2013;11(5):1342005.
29. Moon J, Eulenstein O. Synthesizing large-scale species trees using the strict consensus approach. *J Bioinform Comput Biol*. 2017;15(03):1740002. PMID: 28513253.
30. Moon J, Eulenstein O. Synthesizing species trees from unrooted gene trees: A parameterized approach. In: Proceedings of the 8th ACM-BCB. New York: ACM; 2017. p. 253–62.
31. Tabaszewski P, Górecki P, Markin A, Anderson T, Eulenstein O. Consensus of all Solutions for Intractable Phylogenetic Tree Inference. *IEEE/ACM Trans Comput Biol Bioinform*. 2019;1–14. <https://doi.org/10.1109/TCBB.2019.2947051>.
32. Boykin LM, Kubatko LS, Lowrey TK. Comparison of methods for rooting phylogenetic trees: A case study using *Orcuttieae* (Poaceae: Chloridoideae). *Mol Phylogenet Evol*. 2010;54(3):687–700.
33. Burleigh JG, Bansal MS, Eulenstein O, Hartmann S, Wehe A, Vision TJ. Genome-scale phylogenetics: inferring the plant tree of life from 18,896 gene trees. *Syst Biol*. 2010;60(2):117–25. <http://oup.prod.sis.lan/sysbio/article-pdf/60/2/117/24552312/syq072.pdf>.
34. Huelsenbeck JP, Bollback JP, Levine AM. Inferring the root of a phylogenetic tree. *Syst Biol*. 2002;51(1):32–43. <http://oup.prod.sis.lan/sysbio/article-pdf/51/1/32/19502830/51-1-32.pdf>.
35. Holland BR, Penny D, Hendy MD. Outgroup misplacement and phylogenetic inaccuracy under a molecular clock? a simulation study. *Syst Biol*. 2003;52(2):229–38. <http://oup.prod.sis.lan/sysbio/article-pdf/52/2/229/19502992/52-2-229.pdf>.
36. Chen K, Durand D, Farach-Colton M. NOTUNG: a program for dating gene duplications and optimizing gene family trees. *J Comput Biol*. 2000;7(3–4):429–47.
37. Górecki P, Burleigh JG, Eulenstein O. GTP supertrees from unrooted gene trees: linear time algorithms for NNI based local searches. In: International Symposium on Bioinformatics Research and Applications; 2012. p. 102–14.
38. Górecki P, Eulenstein O, Tiuryn J. Unrooted tree reconciliation: a unified approach. *IEEE/ACM Trans Comput Biol Bioinform*. 2013;10(2):522–36.
39. Wilkerson M, Cotton JA, Lapointe F-J, Pisani D. Properties of supertree methods in the consensus setting. *Syst Biol*. 2007;56(2):330–7.
40. Bansal MS, Burleigh JG, Eulenstein O. Efficient genome-scale phylogenetic analysis under the duplication-loss and deep coalescence cost models. *BMC Bioinformatics*. 2010;11(Suppl 1):42.
41. Ma B, Li M, Zhang L. On reconstructing species trees from gene trees in term of duplications and losses. In: Proceedings of RECOMB 98. ACM; 1998. p. 182–91.
42. Górecki P, Eulenstein O, Tiuryn J. Unrooted tree reconciliation: A unified approach. *IEEE/ACM Trans Comput Biol Bioinform*. 2013;10(2):522–36.
43. Górecki P, Tiuryn J. Inferring phylogeny from whole genomes. *Bioinformatics*. 2007;23(2):116–22.
44. Górecki P, Eulenstein O. Deep coalescence reconciliation with unrooted gene trees: Linear time algorithms. In: International Computing and Combinatorics Conference; 2012. p. 531–42.
45. Górecki P, Eulenstein O. A Robinson-Foulds measure to compare unrooted trees with rooted trees. In: International Symposium on Bioinformatics Research and Applications; 2012. p. 115–26.
46. Maddison WP, Knowles LL. Inferring phylogeny despite incomplete lineage sorting. *Syst Biol*. 2006;55(1):21–30.
47. Bansal MS, Burleigh JG, Eulenstein O. Efficient genome-scale phylogenetic analysis under the duplication-loss and deep coalescence cost models. *BMC Bioinformatics*. 2010;11(Suppl 1):42.
48. Chaudhary R, Bansal MS, Wehe A, Fernández-Baca D, Eulenstein O. iGTP: a software package for large-scale gene tree parsimony analysis. *BMC Bioinformatics*. 2010;11:1–7.
49. Lin HT, Burleigh JG, Eulenstein O. Consensus properties for the deep coalescence problem and their application for scalable tree search. *BMC Bioinformatics*. 2012;13(Suppl 10):12.
50. Moon J, Eulenstein O. Synthesizing large-scale species trees using the strict consensus approach. *J Bioinform Comput Biol*. 2017;15(3):1740002. <https://doi.org/10.1142/S0219720017400029>.
51. Chen D, Eulenstein O, Fernández-Baca D, Burleigh JG. Improved heuristics for minimum-flip supertree construction. *Evol Bioinforma*.

2006;2:117693430600200003. <https://doi.org/10.1177/117693430600200003>.

52. Mirarab S, Warnow T. ASTRAL-II: coalescent-based species tree estimation with many hundreds of taxa and thousands of genes. *Bioinformatics*. 2015;31(12):44–52.
53. Mirarab S, Reaz R, Bayzid MS, Zimmermann T, Swenson MS, Warnow T. ASTRAL: genome-scale coalescent-based species tree estimation. *Bioinformatics*. 2014;30(17):541–8.

### **Publisher's Note**

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

**Ready to submit your research? Choose BMC and benefit from:**

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

**At BMC, research is always in progress.**

Learn more [biomedcentral.com/submissions](https://biomedcentral.com/submissions)

