Journal of Discrete Algorithms ••• (••••) •••–•••

Contents lists available at SciVerse ScienceDirect

Journal of Discrete Algorithms

www.elsevier.com/locate/jda

Gene tree correction for reconciliation and species tree inference: Complexity and algorithms ☆

Riccardo Dondi [a,*], Nadia El-Mabrouk [b], Krister M. Swenson [b,c]

[a] *Dipartimento di Scienze Umane e Sociali, Università degli Studi di Bergamo, Bergamo, Italy*
[b] *Département d'Informatique et Recherche Opérationnelle, Université de Montréal, Montréal, Canada*
[c] *Department of Computer Science, McGill, Montréal, Canada*

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Bioinformatics
Gene trees
Reconciliation of gene trees and species trees
Algorithms
Computational complexity

ABSTRACT

Reconciliation consists in mapping a gene tree T into a species tree S, and explaining the incongruence between the two as evidence for duplication, loss and other events shaping the gene family represented by the leaves of T. When S is unknown, the *Species Tree Inference Problem* is to infer, from a set of gene trees, a species tree leading to a minimum reconciliation cost. As reconciliation is very sensitive to errors in T, gene tree correction prior to reconciliation is a fundamental task. In this paper, we investigate the complexity of four different combinatorial approaches for deleting misplaced leaves from T. First, we consider two problems (Minimum Leaf Removal and Minimum Species Removal) related to the reconciliation of T with a known species tree S. In the former (latter respectively) we want to remove the minimum number of leaves (species respectively) so that T is "MD-consistent" with S. Second, we consider two problems (Minimum Leaf Removal Inference and Minimum Species Removal Inference) related to species tree inference. In the former (latter respectively) we want to remove the minimum number of leaves (species respectively) from T so that there exists a species tree S such that T is MD-consistent with S. We prove that Minimum Leaf Removal and Minimum Species Removal are APX-hard, even when each label has at most two occurrences in the input gene tree, and we present fixed-parameter algorithms for the two problems. We prove that Minimum Leaf Removal Inference is not only NP-hard, but also W[2]-hard and inapproximable within factor $c \ln n$, where n is the number of leaves in the gene tree. Finally, we show that Minimum Species Removal Inference is NP-hard and W[2]-hard, when parameterized by the size of the solution, that is the minimum number of species removals.

1. Introduction

Duplication followed by modification is a major mechanism driving evolution [28]. The footprint in present-day genomes is the presence of many copies of the same gene in a single genome. Inferring duplication and loss histories for a gene family is crucial for deciphering the evolutionary relationship between gene copies, with important implication towards the annotation and functional specificity of genes. In 1979 Goodman et al. [20] introduced gene tree and species tree reconciliation as a method to infer such a history. A typical reconciliation study first constructs a gene family by identifying genes among a set of genomes that share certain sequence similarity [19]. Such genes are assumed to be homologs, i.e. originating from a single ancestral gene. A gene tree T that best reflects the evolution of the sequences is then constructed.

☆ A preliminary version of this paper appeared in Dondi and El-Mabrouk (2012) [15].

* Corresponding author.
E-mail addresses: riccardo.dondi@unibg.it (R. Dondi), mabrouk@iro.umontreal.ca (N. El-Mabrouk), swensonk@iro.umontreal.ca (K.M. Swenson).

1570-8667/$ – see front matter
http://dx.doi.org/10.1016/j.jda.2013.06.001

Please cite this article in press as: R. Dondi et al., Gene tree correction for reconciliation and species tree inference: Complexity and algorithms, Journal of Discrete Algorithms (2013), http://dx.doi.org/10.1016/j.jda.2013.06.001

A reconciliation consists in "embedding" this gene tree into the species tree, and interpreting the incongruence between the two as a description of gene family evolution through duplication, loss and other events such as horizontal gene transfer [1], incomplete lineage sorting [14] and hybridization [38]. Probabilistic [3] or combinatorial [26] criteria are used to choose the right "reconciliation". When no preliminary knowledge on the species tree is given, a natural problem, known as the *Species Tree Inference Problem*, is to infer, from a set of gene trees, a species tree leading to a most parsimonious evolutionary scenario [12,26]. The species tree inference problem is known to be NP-hard [26], even when the number of input gene trees is bounded by 5 [7].

A strict prerequisite for reconciliation is to have a gene tree free from error, as a few misplaced leaves can lead to a completely different history, possibly with significantly more duplications and losses [24,32]. Consequently, various practical solutions have been considered such as manually curating some orthologs (TreeFam [25]), manually correcting gene trees (PANTHER [36]), or avoiding reconciliation by integrating the orthology identification procedure or the species tree information in the construction of gene trees [3,5,9,31,35,37]. Another strategy is to explore the space of gene trees obtained from the original one by performing rearrangements around weakly-supported edges and select the tree giving rise to the minimum reconciliation cost [13]. This method has been generalized in [21] to unrooted gene trees.

A different strategy for preprocessing a gene tree T prior to reconciliation or species tree inference, is to prune misplaced leaves. Criteria for identifying such leaves were given in [12]. The duplication nodes of T with respect to a species tree S can be subdivided into apparent (called "observed" in [33]) and non-apparent duplication (NAD) nodes, where the latter class has been flagged as potentially resulting from the misplacement of leaves in the gene tree. The reason is that each one of the NAD nodes reflects a phylogenetic contradiction with the species tree that is not due to the presence of duplicated gene copies. In [16], we considered the Minimum Leaf Removal Problem which consists in removing, from a given gene tree, the minimum number of leaves leading to an MD (Minimum Duplication) tree, e.g. a tree without any NAD node with respect to S. An exact polynomial-time algorithm has been described for two special classes of gene trees, and a polynomial-time heuristic with no guarantee of optimality, has been presented for the general case. In [34], we presented heuristics for a generalized version of the problem, the Minimum Species Removal Problem, which consists in removing from T the minimum number of leaf labels (species) leading to an MD-tree.

As for species tree inference, it has been shown in [12] that deciding whether a gene tree T is MD-consistent with at least one species tree, i.e. there is a species tree S such that T has no NAD nodes when mapped in S, can be done in polynomial time, as well as computing such a species tree. In the case that a tree T is not an MD-tree, a natural problem is to find the minimum number of leaves/species that have to be removed from T to become an MD-tree. The Minimum Leaf Removal Inference (Minimum Species Removal Inference respectively) is the problem of removing the minimum number of leaves (species respectively) from T, so that there exists a species tree S such that T is MD-consistent with S. Heuristics for these two problems have been presented in [34]. In this paper, we study the computational, parameterized and approximation complexity of the four problems stated above. We recall that APX is the class of those problems that can be approximated within constant factor. Moreover, when a problem is APX-hard, it does not admit a Polynomial Time Approximation Scheme (PTAS). A PTAS is an algorithm that, given as input an instance I of a problem and an error parameter $\varepsilon > 0$, computes in polynomial time a solution within a factor $1 + \varepsilon$ from the value of an optimal solution in case of minimization problem, and within factor $1 - \varepsilon$ from the value of an optimal solution in case of maximization problem.

First, we prove in Section 3 that Minimum Leaf Removal and Minimum Species Removal are APX-hard even if each label is associated with at most two leaves of the gene tree. Then, we prove in Section 4 that Minimum Leaf Removal Inference is not only NP-hard, but also W[2]-hard (when parameterized by the size of the solution, that is the minimum number of leaf removals) and inapproximable within factor $c \ln n$, for any $c > 0$, where n denotes the number of leaves of the input gene tree. We also prove that Minimum Species Removal Inference is NP-hard and W[2]-hard when parameterized by the size of the solution, that is the minimum number of label (species) removals. On the positive side, in Section 5 we present fixed-parameter algorithms for Minimum Leaf Removal and Minimum Species Removal, where the parameters are the size of the solution (minimum number of leaf/label removals) and the number of genomes containing multiple gene copies.

2. Preliminary definitions

2.1. Trees

Let $\Gamma = \{1, 2, \ldots, \gamma\}$ be a set of labels representing γ different species (genomes). We consider two kinds of rooted binary trees leaf-labelled by the elements of Γ: a *species tree* S is a tree where each element of Γ labels exactly one leaf, while a *gene tree* T is a tree where each element of Γ may label more than one leaf (Fig. 1(a) and (b)). A gene tree represents a gene family, where each leaf labelled x represents a gene copy located on genome x.

Given a tree U, we denote by $L(U)$ the set of its leaves and by $V(U)$ the set of its nodes. For a species tree S leaf-labelled by Γ, there is a bijection between $L(S)$ and Γ (notice that this may not be the case for a gene tree as it is not necessarily uniquely leaf-labelled). Given an internal node x of U, we denote by x_l and x_r respectively, the left and right child of x, by $U(x)$ the subtree of U rooted at x, and by $\Gamma(U(x))$ the set of leaf labels of $U(x)$. If there is no ambiguity on the tree being considered, we denote $\mathcal{C}(x) = \Gamma(U(x))$; $\mathcal{C}(x)$ is called the *cluster* of x. A *triplet* is a uniquely leaf-labelled binary rooted tree on three leaves. An *ancestor* of a node x of U is any node on the path from the root of U to x.

Please cite this article in press as: R. Dondi et al., Gene tree correction for reconciliation and species tree inference: Complexity and algorithms, Journal of Discrete Algorithms (2013), http://dx.doi.org/10.1016/j.jda.2013.06.001

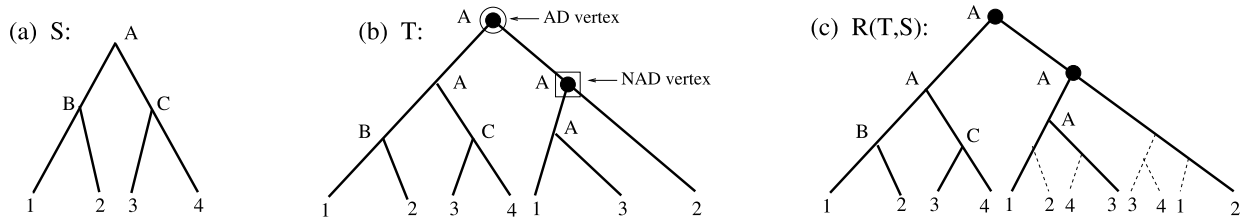

Fig. 1. (a) A species tree S for $\Gamma = \{1, 2, 3, 4\}$; (b) A gene tree T. A leaf label g indicates a gene copy in genome g. Internal nodes are labelled according to the LCA mapping between T and S. Flagged nodes are duplication nodes, as they map to node A of S, and they both have a child mapping to A as well. The root is an AD node as its left and right subtrees share at least one common leaf (they share 2 and 3). The node indicated by a square is a NAD node as its left and right subtrees have empty leaf-set intersection. All other nodes are speciation nodes. (c) A reconciliation $R(T, S)$ of T and S. Dotted lines represent subtree insertions. Flagged nodes are duplication nodes, and all others are speciation nodes. This reconciliation reflects a history of the gene family with two gene duplications preceding the first speciation event, and four losses.

Given a tree U, a *leaf removal* consists in removing a given leaf l of U, and suppressing the resulting degree two node (that is the parent of l). A *label removal* consists in removing all the leaves of U associated with a given label $\sigma \in \Gamma$, and suppressing the resulting degree two nodes. If a tree U' is obtained from U through a sequence of leaf/label removals, then U' is *included* in U.

A *subtree insertion* in U consists in creating a new node x on a branch (a, b) (joining node a to node b, b being the child of a), making b the left child of x, setting the parent of x to a, and grafting the subtree being inserted as the second child of x (create an edge from x to the root of the subtree). An *extension* of U is a tree obtained from U through a sequence of subtree insertions.

2.2. Reconciliation

Several definitions of reconciliation exist in the literature. The one we use utilizes tree extensions [12,17]. A *reconciliation* $R(T, S)$ of a gene tree T with respect to a species tree S is an extension of T such that for each internal node x of $R(T, S)$: (i) there is a node x' in S such that $\mathcal{C}(x) = \mathcal{C}(x')$, and (ii) either $\mathcal{C}(x_l) = \mathcal{C}(x_r)$ (duplication node) or $\mathcal{C}(x_l) \cap \mathcal{C}(x_r) = \emptyset$ (speciation node). An example is given in Fig. 1(c).

A history of duplications and losses can immediately be inferred from a reconciliation. Different algorithms have been developed for recovering a reconciliation minimizing a duplication and/or loss cost [8,12,18,22,23,26,29,30], most of them based on a method called *LCA mapping*.

The LCA mapping between a gene tree T and a species tree S, denoted by $\text{LCA}_{T,S}$, maps every node x of T to the Lowest Common Ancestor (LCA) of $\mathcal{C}(x)$ in S. Formally, $\text{LCA}_{T,S}(x) = y$, where y is the node of S that has the minimum cluster such that $\mathcal{C}(x) \subseteq \mathcal{C}(y)$. A *duplication* occurs in a node x of T (or x is a duplication), if x and at least one of its children are mapped by $\text{LCA}_{T,S}$ in the same node y of the species tree S. If x is not a duplication node, then x is a *speciation*. See Fig. 1(b) for an example.

2.3. Duplication nodes and MD-trees

The notations of this section are those used in [12,16]. Let x be a node of a gene tree T verifying $\mathcal{C}(x_l) \cap \mathcal{C}(x_r) \neq \emptyset$. Then, for any species tree S, x is guaranteed to be a duplication node. Such a node x is called an *Apparent Duplication node* (*AD node* for short). Given a species tree S, a duplication node x which is not an AD node is called a *Non-Apparent Duplication node* (*NAD node* for short). A gene tree T is *MD-consistent* (MD holds for "Minimum Duplication") with a species tree S if and only if each node of T is either a speciation or an AD node.

As explained in [16], NAD nodes point to disagreements between a gene tree T and a species tree S that are not due to the presence of repeated leaf labels, i.e. duplicated gene copies (see Fig. 1(b)). It has therefore been suggested, and supported by simulations in [12], that NAD nodes may point to gene copies that are erroneously placed in T. Notice that a misplaced gene in a gene tree T does not necessarily lead to a NAD node. In other words, NAD nodes can only point to a subset of misplaced leaves. However, in the context of reconciliation and species tree inference, the damage caused by a misplaced leaf leading to a NAD node is to significantly increase the real duplication and/or loss cost of the tree. Following these observations, the Minimum Leaf Removal Problem, given below, has been considered in [16] for error-correction in gene trees.

Problem 1 *(Minimum Leaf Removal Problem [MinLeafRem]).*
Input: A gene tree T and a species tree S, both leaf-labelled by Γ.
Output: A tree T^* MD-consistent with S such that T^* is obtained from T by a minimum number of leaf removals.

Please cite this article in press as: R. Dondi et al., Gene tree correction for reconciliation and species tree inference: Complexity and algorithms, Journal of Discrete Algorithms (2013), http://dx.doi.org/10.1016/j.jda.2013.06.001

A more conservative strategy that can be used when full confidence is not put in the species tree, is to remove the minimum number of species from γ such that T restricted to the new set is MD-consistent with S. The following combinatorial problem has first been introduced in [34].

Problem 2 *(Minimum Species Removal Problem [MinSpecRem]).*
Input: A gene tree T and a species tree S, both leaf-labelled by Γ.
Output: A tree T^* MD-consistent with S such that T^* is obtained from T by a minimum number of label removals.

As for the species tree inference problem, natural generalizations of the MinLeafRem and MinSpecRem problems, first introduced in [34], are given below.

Problem 3 *(Minimum Leaf Removal Inference Problem [MinLeafRemInf]).*
Input: A gene tree T leaf-labelled by Γ.
Output: A gene tree T^* obtained from T by a minimum number of leaf removals and MD-consistent with some species tree S.

Problem 4 *(Minimum Species Removal Inference Problem [MinSpecRemInf]).*
Input: A gene tree T leaf-labelled by Γ.
Output: A gene tree T^* obtained from T by a minimum number of label removals and MD-consistent with some species tree S.

3. Hardness of Minimum Leaf Removal and Minimum Species Removal

In this section we consider the computational (and approximation) complexity of MinLeafRem and MinSpecRem. We show that both problems are APX-hard, even in the restricted case that each label is associated with at most two leaves of T. We denote these restrictions of the problems by MinLeafRem(2) and MinSpecRem(2).

First, we prove that MinLeafRem(2) is APX-hard, by giving an L-reduction from the Minimum Vertex Cover Problem on Cubic graphs (MVCC). Notice that MVCC is known to be APX-hard [4]. The proof that MinSpecRem(2) is APX-hard follows closely, as discussed in Theorem 2.

Problem 5 *(Minimum Vertex Cover Problem on Cubic Graphs [MVCC]).*
Input: A cubic graph $G = (V, E)$ where $V = \{v_1, \ldots, v_n\}$ is the set of vertices and E the set of edges of G (in a cubic graph, each vertex has degree 3).
Output: A minimum cardinality set $V' \subseteq V$, such that for each edge $\{v_i, v_j\} \in E$, at least one of v_i, v_j belongs to V'.

Let $G = (V, E)$ be an instance of MVCC. We define an instance of MinLeafRem(2) associated with G, consisting of a gene tree T and a species tree S, both leaf-labelled by Γ. The set Γ is defined as follows, where $t = 4|V| + |E| + 1$:

$$\Gamma = \{v_{i,l}\colon\ v_i \in V, 1 \leqslant l \leqslant 4\} \cup \left\{v_i^j\colon\ v_i \in V, \{v_i, v_j\} \in E\right\} \cup \left\{e_{i,j}\colon\ \{v_i, v_j\} \in E, i < j\right\}$$
$$\cup \{z_i\colon\ 1 \leqslant i \leqslant t\} \cup \{\alpha\}.$$

We denote $Z = \{z_i\colon\ 1 \leqslant i \leqslant t\}$. Let U be a tree, which is either the gene tree T, the species tree S, or a tree included in T with a leaf labelled by α. We define the *spine* of U as the path from the root of U to the unique leaf of U labelled by α.

Next, we define an ordering on the edges E of G. Consider the edges $\{v_i, v_j\}$, with $i < j$, and $\{v_h, v_k\}$, with $h < k$, then $\{v_i, v_j\} < \{v_h, v_k\}$, iff $i \leqslant h$, and $j < k$ if $i = h$. Denote with $\{v_p, v_q\}$ the last edge in such ordering of E.

The gene tree T is defined as in Fig. 2. It contains the following kinds of subtrees (these subtrees are inserted in the spine starting from the leaf labelled by α to the root): (1) a subtree T_{v_i}, for each vertex $v_i \in V$; (2) a subtree $T_{e_{ij}}$ and a leaf $e_{i,j}$, for each edge $\{v_i, v_j\} \in E$; (3) a tree T_Z, which is a caterpillar tree of size t with leaves uniquely leaf-labelled by the set Z. Notice that the order in which the subtrees $T_{e_{ij}}$ and the leaf $e_{i,j}$ appear in T, depends on the order of the corresponding edges of E. Moreover, notice that the order assigned to the labels v_i^j, v_i^h v_i^k in T_{v_i} is arbitrary.

The species tree S is defined in Fig. 3. It contains the three following kinds of subtrees: (1) a subtree S_{v_i}, for each vertex $v_i \in V$; (2) a single leaf labelled by $e_{i,j}$, for each edge $\{v_i, v_j\} \in E$; (3) a tree S_Z, which is a caterpillar tree of size t uniquely leaf-labelled by the set Z.

It is easy to see that S is a species tree uniquely leaf-labelled by Γ. Moreover, notice that T is a gene tree where each label in Γ is associated with at most two leaves of T. Indeed, each subtree T_{v_i}, $T_{e_{i,j}}$, T_Z is uniquely leaf-labelled and each label in $\{v_{i,l}\colon\ v_i \in V, 1 \leqslant l \leqslant 4\} \cup \{z_i\colon\ 1 \leqslant i \leqslant t\} \cup \{\alpha\}$ is associated with exactly one leaf of T. Each label in $\{v_i^j\colon\ v_i \in V, \{v_i, v_j\} \in E\}$ is associated with a leaf of the subtree T_{v_i} and a leaf of the subtree $T_{e_{i,j}}$. Each leaf in $\{e_{i,j}\colon\ \{v_i, v_j\} \in E\}$ is associated with a leaf of the subtree $T_{e_{i,j}}$ and with a singleton leaf connected to the spine of T.

The following properties of T are directly deduced from the construction of T.

Please cite this article in press as: R. Dondi et al., Gene tree correction for reconciliation and species tree inference: Complexity and algorithms, Journal of Discrete Algorithms (2013), http://dx.doi.org/10.1016/j.jda.2013.06.001

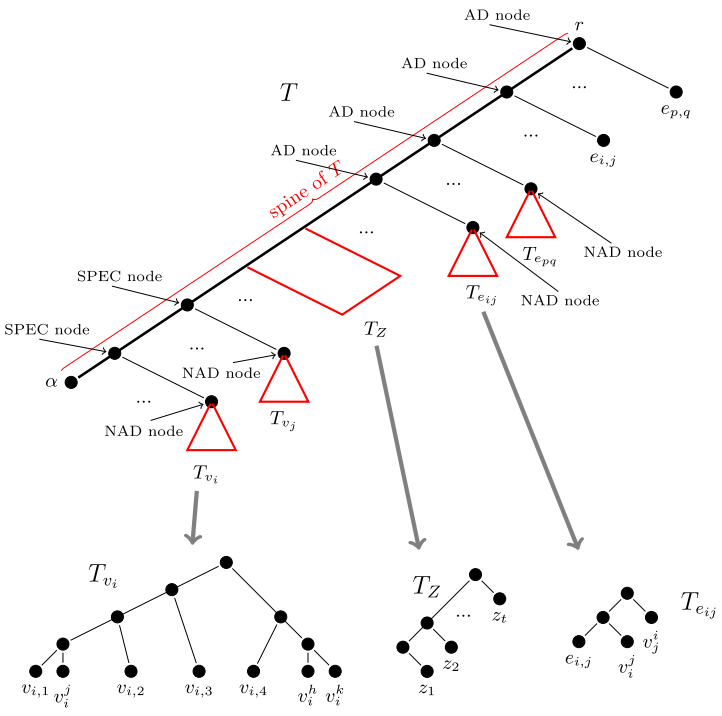

Fig. 2. The gene tree T, and the subtrees T_{v_i}, T_Z and $T_{e_{ij}}$ of T. Notice that $i < j$, hence T_{v_j} is closer to the root than T_{v_i}. SPEC nodes are speciation nodes.

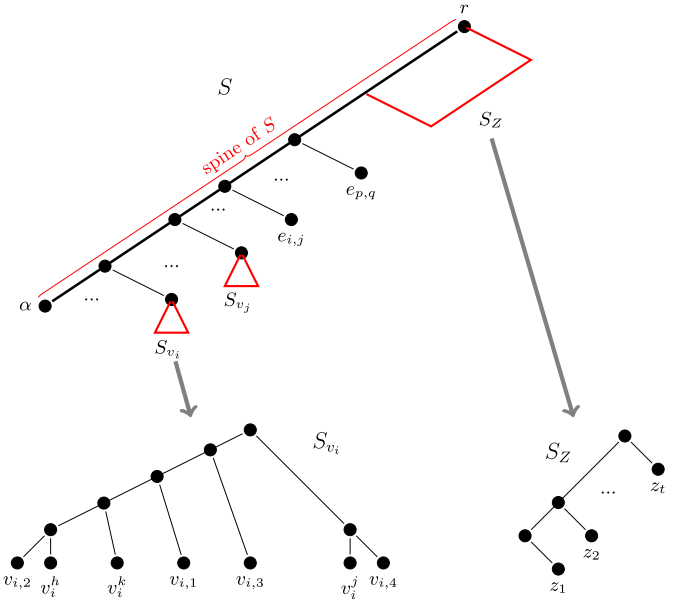

Fig. 3. The species tree S, and the subtrees S_{v_i}, S_Z of S. Notice that $i < j$, hence S_{v_j} is closer to the root than S_{v_i}.

Remark 1. The root of T_Z and all its ancestors are mapped (by the LCA mapping) to the root r of S. Consequently, all the ancestors of the root of T_Z are duplication nodes. Moreover, we deduce from the non-empty intersection of the left and right leaf sets that all these nodes are AD nodes.

Remark 2. For each $\{v_i, v_j\} \in E$, the root of the corresponding subtree $T_{e_{i,j}}$ is a NAD node. Indeed, it is mapped to the same node of S as its left child, and it does not contain any duplicated leaf label.

The next remark follows from Fig. 4.

Please cite this article in press as: R. Dondi et al., Gene tree correction for reconciliation and species tree inference: Complexity and algorithms, Journal of Discrete Algorithms (2013), http://dx.doi.org/10.1016/j.jda.2013.06.001

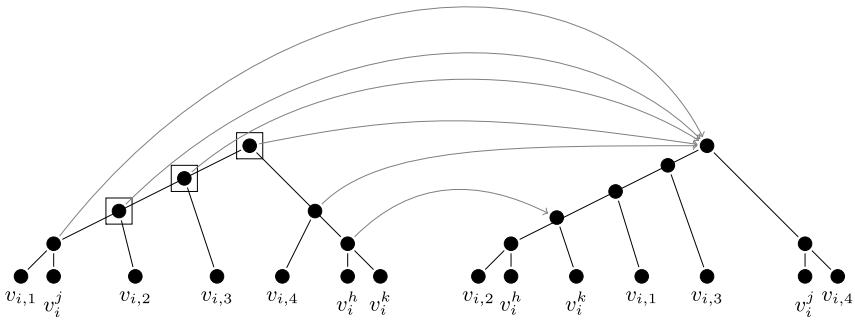

Fig. 4. LCA mapping of T_{v_i} to S_{v_i}.

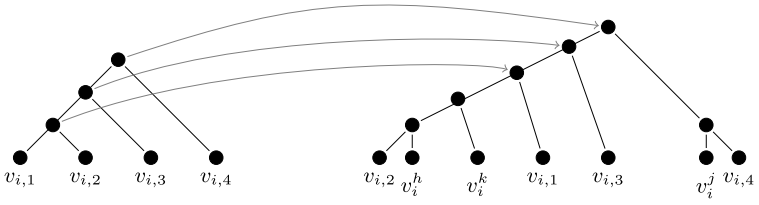

Fig. 5. LCA mapping from T_{v_i} after the removal of leaves with labels v_i^j, v_i^h, v_i^k to S_{v_i}.

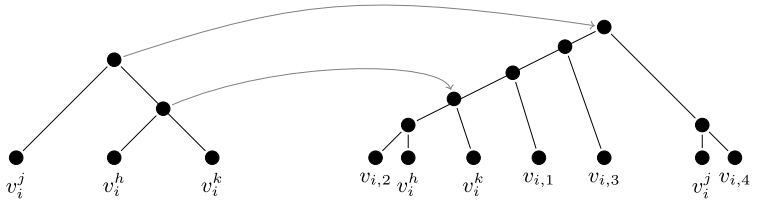

Fig. 6. LCA mapping from T_{v_i} after the removal of leaves with labels $\{v_{i,1}, v_{i,2}, v_{i,3}, v_{i,4}\}$ to S_{v_i}.

Remark 3. For each $v_i \in V$, the corresponding subtree T_{v_i} contains three NAD nodes.

It follows from Remark 3 that appropriate leaves must be removed from each T_{v_i} in order to obtain a solution of MinLeafRem(2) over instance (T, S). The main goal of the rest of this section is to explain the required removals. We begin by giving an overview of the results.

First, from Remark 1, each node v of T such that $\mathcal{C}(v) \supseteq Z$ is mapped (by the LCA mapping) to the root of S. Furthermore, we show in Lemma 3 that we can assume that a solution T^* of MinLeafRem contains all the leaves of T_Z. Hence each ancestor of T_Z in T^* is a duplication node, which must be an AD node (in order for T^* to be MD-consistent with S). Consider an ancestor v of T_Z connecting a subtree $T_{e_{ij}}$ to the spine of T. Since v is mapped in the root of S, to be an AD node in the solution, we should remove from $T_{e_{ij}}$ exactly one leaf with label in $\{v_i^j, v_j^i\}$, assume w.l.o.g. v_j^i, and keep in T_{v_i} the leaf v_i^j (Lemma 5), so that the right subtree and left subtree of $T(v)$ both contain a leaf labelled by v_i^j.

The next remark follows from a direct inspection of Figs. 5 and 6.

Remark 4. Let v_i be a vertex of G. Then: (1) the tree obtained from T_{v_i} by removing the leaves with labels v_i^j, v_i^h, v_i^k is MD-consistent with S_{v_i}; (2) the tree obtained from T_{v_i} by removing the leaves with labels $v_{i,1}$, $v_{i,2}$, $v_{i,3}$, $v_{i,4}$ is MD-consistent with S_{v_i}.

Lemma 1. *Let v_i be a vertex of G. Then*: (1) *in a solution of* MinLeafRem(2) *over instance (T, S) at least three leaves are removed from T_{v_i}*; (2) *a solution of* MinLeafRem(2) *over instance (T, S) that contains a leaf of T_{v_i} with a label in $\{v_i^j, v_i^h, v_i^k\}$, contains at most three leaves of T_{v_i}.*

Proof. Denote by T^* a solution of MinLeafRem(2) over instance (T, S) and denote by $T^*_{v_i}$ the subtree of T^* that is obtained after the removal of leaves from T_{v_i}. Denote the root of T_{v_i} and S_{v_i} respectively, by r and r'.

Please cite this article in press as: R. Dondi et al., Gene tree correction for reconciliation and species tree inference: Complexity and algorithms, Journal of Discrete Algorithms (2013), http://dx.doi.org/10.1016/j.jda.2013.06.001

(1) We prove the first part of the lemma. If $T^*_{v_i}$ is a subtree of $T_{v_i}(r_l)$ or a subtree of $T_{v_i}(r_r)$, then at least three leaves of T_{v_i} have been removed, hence the first part of the lemma holds. Otherwise, $T^*_{v_i}$ has leafset labelled either by a subset of $(\mathcal{C}(r_l) \cap \mathcal{C}(r'_l)) \cup (\mathcal{C}(r_r) \cap \mathcal{C}(r'_r))$, or by a subset of $(\mathcal{C}(r_l) \cap \mathcal{C}(r'_r)) \cup (\mathcal{C}(r_r) \cap \mathcal{C}(r'_l))$, and again the first part of the lemma holds.

(2) Now, we prove the second part of the lemma. If $T^*_{v_i}$ is a subtree of $T_{v_i}(r_r)$, since $T_{v_i}(r_r)$ contains three leaves, in this case the second part of the lemma holds. Assume $T^*_{v_i}$ is a subtree of $T_{v_i}(r_l)$. By construction $|L(T_{v_i}(r_l))| = 4$. Since $T_{v_i}(r_l)$ is not MD-consistent with S_{v_i}, in this case the second part of the lemma holds.

Assume that $T^*_{v_i}$ has leafset labelled by a subset of $(\mathcal{C}(r_l) \cap \mathcal{C}(r'_l)) \cup (\mathcal{C}(r_r) \cap \mathcal{C}(r'_r))$, then its leaves are labelled by a subset of $\{v_{i,1}, v_{i,2}, v_{i,3}, v_{i,4}\}$, hence in this case the second part of the lemma holds.

Finally, assume that $T^*_{v_i}$ has leafset labelled by a subset of $(\mathcal{C}(r_l) \cap \mathcal{C}(r'_r)) \cup (\mathcal{C}(r_r) \cap \mathcal{C}(r'_l))$, then its leaves are labelled by a subset of $\{v_i^j, v_i^h, v_i^k\}$, hence in this case the second part of the lemma holds. □

It follows from Remark 4 and Lemma 1 that a solution of MinLeafRem(2) over instance (T, S) is obtained by removing leaves from each T_{v_i} in essentially two possible ways: (1) either remove the four leaves $\{v_{i,1}, v_{i,2}, v_{i,3}, v_{i,4}\}$, or (2) remove the three leaves $\{v_i^j, v_i^h, v_i^k\}$. Indeed if we remove a different set of leaves from T_{v_i}, then this set must contain at least four leaves, and we can assume that this set is $\{v_{i,1}, v_{i,2}, v_{i,3}, v_{i,4}\}$, since these four labels belong only to leaves of the subtree T_{v_i}, hence their removal does not affect other subtrees of $T(v_i)$. We will relate the former case to the vertex v_i being included in a vertex cover V' of G, and the latter case to the vertex v_i being in $V \setminus V'$ (Lemma 4 and Lemma 5). We first give two preliminary lemmas.

Lemma 2. *At least one leaf from $T_{e_{ij}}$, for each $e_{i,j} \in E$, is removed in each solution of* MinLeafRem(2) *over instance (T, S).*

Proof. Direct corollary of Remark 2. □

The following lemma will be used to show that the caterpillar tree T_Z is kept in a solution of MinLeafRem(2).

Lemma 3. *There is no optimal solution of* MinLeafRem(2) *over instance (T, S) that is obtained by removing less than $4|V| + |E| + 1$ leaves, one of them being a leaf of T_Z.*

Proof. Let T^* be a solution of MinLeafRem(2) over instance (T, S) obtained from T by removing less than $4|V| + |E| + 1$ leaves. Notice that, since $|Z| = 4|V| + |E| + 1$, at least one leaf with a label in the set Z must be in T^*. Assume that a leaf f with label z_h is removed from T. The (re)-insertion of leaf f in T^* does not affect other nodes of T^*, that is the insertion of the leaf with label z_h does not cause any AD node of T^* to become a NAD node. □

We are now ready to show the two main technical results of the reduction.

Lemma 4. *Let $G = (V, E)$ be an instance of* MVCC *and let (T, S) be the corresponding instance of* MinLeafRem(2). *Then, starting from a vertex cover V' of G, we can compute in polynomial time a solution of* MinLeafRem(2) *over instance (T, S) that is obtained by removing $3|V| + |V'| + |E|$ leaves from T.*

Proof. Let $V' \subseteq V$ be a vertex cover of $G = (V, E)$. Then we define a solution T^* by removing some leaves of the subtrees of T as follows:

- for each $v_i \in V'$, remove from the subtree T_{v_i} the set of leaves labelled by $\{v_{i,1}, v_{i,2}, v_{i,3}, v_{i,4}\}$ (hence this subtree $T^*_{v_i}$ of T^* has leafset labelled by $\{v_i^j, v_i^h, v_i^k\}$);
- for each $v_i \in V \setminus V'$, remove from the subtree T_{v_i} the set of leaves labelled by $\{v_i^j, v_i^h, v_i^k\}$ (hence this subtree $T^*_{v_i}$ of T^* has leafset labelled by $\{v_{i,1}, v_{i,2}, v_{i,3}, v_{i,4}\}$);
- for each $\{v_i, v_j\} \in E$, if $v_i \in V'$, then remove from $T_{e_{ij}}$ the leaf labelled by v_j^i (hence this subtree $T^*_{e_{i,j}}$ has leafset labelled by $\{e_{i,j}, v_i^j\}$), otherwise remove from $T_{e_{ij}}$ the leaf labelled by v_i^j (hence this subtree $T^*_{e_{i,j}}$ has leafset labelled by $\{e_{i,j}, v_j^i\}$). Notice that if both $v_i, v_j \in V'$, then we remove from $T_{e_{ij}}$ the leaf labelled by v_j^i and not the leaf labelled by v_i^j.

The tree T^* is computed by removing four leaves of T_{v_i}, for each $v_i \in V'$, three leaves of T_{v_i}, for each $v_i \in V \setminus V'$, and one leaf of $T_{e_{ij}}$, for each $\{v_i, v_j\} \in E$, hence the overall number of leaves removed from T to obtain T^* is $3|V| + |E| + |V'|$.

Now, from Remark 4 and by construction, it follows that the subtrees $T^*_{v_i}$, for each $v_i \in V$, and $T^*_{e_{i,j}}$, for each $\{v_i, v_j\} \in E$, of T^* are MD-consistent with S.

Now, consider the other internal nodes of T^*, that is the nodes on the spine of T^*. If one of these nodes joins a subtree $T^*_{v_i}$ to the spine of T^*, by construction is a speciation.

Please cite this article in press as: R. Dondi et al., Gene tree correction for reconciliation and species tree inference: Complexity and algorithms, Journal of Discrete Algorithms (2013), http://dx.doi.org/10.1016/j.jda.2013.06.001

Consider the node p joining subtree $T_{e_{ij}}$ to the spine of T^*. Since V' is a vertex cover, it follows that a node with either label v_i^j or v_j^i belongs to both clusters $\mathcal{C}(p_l)$ and to $\mathcal{C}(p_r)$ of the right child and the left child of p. Hence p is an AD node.

Consider the node p joining a leaf labelled by $e_{i,j}$ to the spine of T^*. Then, p is an AD node, since both clusters of the right child and the left child of p contain label $e_{i,j}$, formally $e_{i,j} \in \mathcal{C}(p_l)$ and $e_{i,j} \in \mathcal{C}(p_r)$.

Finally, by construction each node joining a leaf with a label in Z to the spine of T^* is a speciation. Hence we can conclude that T^* is MD-consistent with S. □

Lemma 5. *Let $G = (V, E)$ be an instance of* MVCC *and let (T, S) be the corresponding instance of* MinLeafRem(2). *Then starting from a solution of* MinLeafRem(2) *over instance (T, S) that is obtained by removing at most $3|V| + |E| + c$ leaves from T, with $1 \leqslant c \leqslant |V|$, we can compute in polynomial time a vertex cover V' of G such that $|V'| \leqslant c$.*

Proof. Let T^* be a solution of MinLeafRem(2) over instance (T, S) obtained by removing at most $3|V| + |E| + c$ leaves from T, with $1 \leqslant c \leqslant |V|$. First, consider the subtrees $T^*_{v_i}$, with $v_i \in V$, obtained by removing some leaves from T_{v_i}. By Lemma 1, we can assume that T^* is obtained by removing at most four leaves for each subtree T_{v_i} of T. Indeed, if more than four leaves are removed from a subtree T_{v_i} of T, we can replace $T^*_{v_i}$ with the subtree of T_{v_i} containing the leaves labelled by v_i^j, v_i^h, v_i^k.

If $T^*_{v_i}$ is obtained by removing exactly three leaves from T_{v_i}, then by Lemma 1 $T^*_{v_i}$ contains the leaves with labels $v_{i,1}, v_{i,2}, v_{i,3}, v_{i,4}$. Hence in what follows, we can assume by Lemma 4 and by Lemma 1 that $T^*_{v_i}$, with $v_i \in V$, is leaf-labelled either by the set $\{v_{i,1}, v_{i,2}, v_{i,3}, v_{i,4}\}$, or by the set $\{v_i^j, v_i^h, v_i^k\}$. Moreover, by Lemma 3, we can assume that T^* contains each leaf with a label in the set Z.

Assume that for each subtree $T_{e_{ij}}$, with $\{v_i, v_j\} \in E$, T^* contains a subtree $T^*_{e_{i,j}}$. Assume that $T^*_{e_{i,j}}$ does not contain the leaf labelled by $e_{i,j}$, and let $l_{i,j}$ be the single leaf with label $e_{i,j}$ that is connected to the spine of T. It follows that $l_{i,j}$ does not belong to T^*. Indeed, since $e_{i,j} \notin \Gamma(T^*_{e_{i,j}})$, by Remark 1 the ancestor of $l_{i,j}$ would be a NAD node, hence $T^*_{e_{i,j}}$ must contain the leaf labelled by $e_{i,j}$.

If $e_{i,j} \notin \Gamma(T^*_{e_{i,j}})$, we show that we can compute in polynomial time from T^* a solution T^+ of MinLeafRem(2) over instance (T, S) that it is obtained by removing from T at most as many leaves as T^* and such that $l_{i,j} \in L(T^+)$. The solution T^+ is computed as follows: (1) remove from the subtree T_{v_i} exactly the leafset having labels $\{v_{i,1}, v_{i,2}, v_{i,3}, v_{i,4}\}$ (hence T^+ contains a subtree $T^+_{v_i}$ having leaves leaf-labelled by $\{v_i^j, v_i^h, v_i^k\}$); (2) remove from $T_{e_{ij}}$ the leaf of label v_j^i; (3) the single leaf $l_{i,j}$ with label $e_{i,j}$ that is connected to the spine of T belongs to T^+ is (re)-inserted in T^+. Comparing the number of leaves removed to obtain T^+ with the number of leaves removed to obtain T^*, we have: modification (1) increases the number of leaves removed by at most one; modification (2) by Lemma 2 does not change the number of leaves removed; modification (3) decreases the number of leaves removed by at least one. It follows that we have computed in polynomial time from T^* a solution T^+ of MinLeafRem(2) that is obtained by removing from T at most as many leaves as T^* and such that $l_{i,j} \in L(T^+)$.

Hence we assume in what follows that T^* does not remove the leaf labelled by $e_{i,j}$ from $T_{e_{ij}}$, and consequently that the single leaf $l_{i,j}$ is not removed to obtain T^*.

Now we prove that T^* contains at least one of the leaves with label v_i^j, v_j^i of $T_{e_{ij}}$. Assume that T^* contains only the leaf labelled by $e_{i,j}$ of $T_{e_{ij}}$. Then by Remark 1, the node p where $T_{e_{ij}}$ joins the spine of T^* is a NAD node, since the two children of p have disjoint clusters. Hence in what follows we can assume that, for each subtree $T_{e_{ij}}$, T^* contains a subtree $T^*_{e_{i,j}}$ that is obtained by removing exactly one leaf having label in the set $\{v_i^j, v_j^i\}$ from $T_{e_{ij}}$.

Consider the node p that joins the spine of T^* to the subtree $T^*_{e_{i,j}}$. By Remark 1, the node p and its left child p_l are both mapped to the root of S. It follows that p must be an AD node, hence the cluster of its left child and the cluster of its right child must contain a same label. This implies that $T^*_{v_i}$ contains a leaf labelled by v_i^j or that $T^*_{v_j}$ contains a leaf labelled by v_j^i. Notice that by Lemma 1, if $T^*_{v_i}$ contains a leaf labelled by v_i^j, then $T^*_{v_i}$ contains three leaves that are leaf-labelled by the set $\{v_i^j, v_i^h, v_i^k\}$.

Now, we are ready to define a vertex cover V' of T.

$$V' = \left\{v_i\colon\ T^*_{v_i} \text{ contains the leaves labelled by } \left\{v_i^j, v_i^h, v_i^k\right\}\right\}.$$

V' is a vertex cover, since for each subtree $T_{e_{ij}}$ associated with $\{v_i, v_j\} \in E$, $T^*_{v_i}$ has three leaves leaf-labelled by $\{v_i^j, v_i^h, v_i^k\}$ or $T^*_{v_j}$ has three leaves leaf-labelled by $\{v_i^j, v_j^f, v_j^g\}$.

Now, $|V'|$ has size at most c, since T^* is obtained by removing four leaves from each subtree $T^*_{v_i}$ with three leaves leaf-labelled by $\{v_i^j, v_i^h, v_i^k\}$, three leaves from each subtree $T^*_{v_j}$ with four leaves leaf-labelled by $\{v_{i,1}, v_{i,2}, v_{i,3}, v_{i,4}\}$ and exactly one leaf from each subtree $T_{e_{ij}}$. □

Theorem 1. MinLeafRem(2) *is APX-hard.*

Please cite this article in press as: R. Dondi et al., Gene tree correction for reconciliation and species tree inference: Complexity and algorithms, Journal of Discrete Algorithms (2013), http://dx.doi.org/10.1016/j.jda.2013.06.001

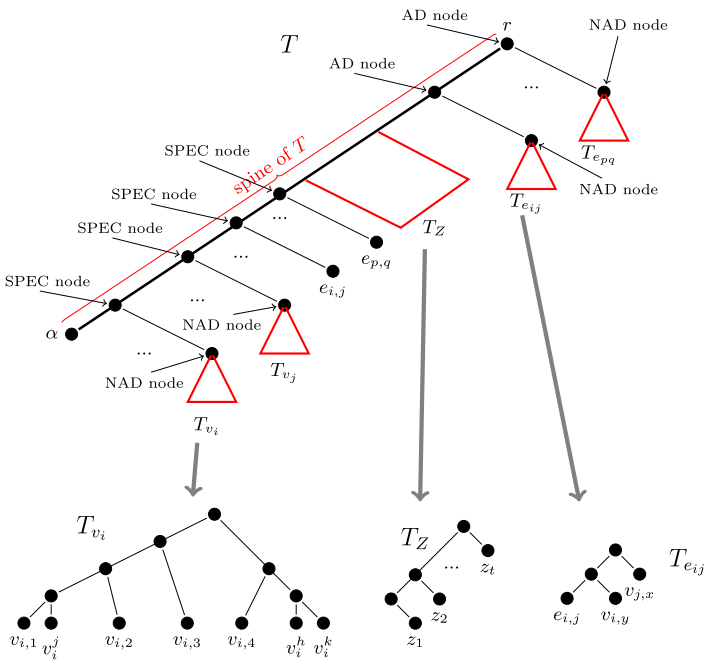

Fig. 7. The gene tree T, and the subtrees T_{v_i}, T_Z and $T_{e_{ij}}$ of T. Notice that $i < j$, hence T_{v_j} is closer to the root than T_{v_i}. SPEC nodes are speciation nodes. In the subtree $T_{e_{ij}}$, $x, y \in \{1, 2, 3, 4\}$.

Proof. It follows from Lemma 4 and from Lemma 5, that we have designed an L-reduction from MVCC to MinLeafRem(2). Since MVCC is APX-hard [4], it follows that also MinLeafRem(2) is APX-hard. □

By using a similar reduction, we prove that MinSpecRem(2) is APX-hard.

Theorem 2. MinSpecRem(2) *is APX-hard.*

Proof. The result is implied by a reduction from MVCC similar to that for MinLeafRem(2). Starting from an instance $G = (V, E)$ of MVCC, we construct an instance of MinSpecRem(2) almost identical to the instance of MinLeafRem(2) described in the previous reduction.

The species tree S is identical to the species tree defined in Fig. 3. The gene tree T is defined as in Fig. 7. Notice that in the construction of T, each of the subtrees $T(e_{i,j})$, $T(e_{i,h})$, $T(e_{i,k})$, with $\{v_i, v_j\}$, $\{v_i, v_h\}$, $\{v_i, v_k\} \in E$, is associated with a distinct label in $\{v_{i,1}, \ldots, v_{i,4}\}$.

By construction the only labels that are associated with more than one leaf are those labels $v_{i,x}$, with $x \in \{1, \ldots, 4\}$, and e_{ij}, with $\{v_i, v_j\} \in E$. For each other label, a leaf removal is equivalent to a label removal.

Consider a subtree $T(v_i)$. By Lemma 1, we can assume that either the set of labels $\{v_{i,1}, v_{i,2}, v_{i,3}, v_{i,4}\}$ or the set of labels $\{v_i^j, v_i^h, v_i^k\}$ is removed from $T(v_i)$.

Now, consider the subtree $T(e_{i,j})$, with $\{v_i, v_j\} \in E$, and let $v_{i,x}$, $v_{j,y}$, for some values $1 \leqslant x \leqslant 4$ and $1 \leqslant y \leqslant 4$, be two labels associated with leaves of $T(e_{i,j})$. We claim that there exists an optimal solution of MinSpecRem(2) over instance (T, S), where at least one of $v_{i,x}$, $v_{j,y}$ is removed. Assume to the contrary that none of the labels $v_{i,x}$, $v_{j,y}$ is removed, then label e_{ij} must be removed, since by construction there is a NAD node in $T(e_{i,j})$. Moreover, by Lemma 1, we can assume that the set of labels $\{v_i^j, v_i^h, v_i^k\}$ is removed from T_{v_i} and that the set of labels $\{v_j^i, v_j^f, v_j^g\}$ is removed from T_{v_i}. Then, we can compute in polynomial time a solution of MinSpecRem(2) over instance (T, S) that does not increase the number of labels removed as follows: we remove the labels $v_{i,1}$, $v_{i,2}$, $v_{i,3}$, $v_{i,4}$, and we put back in the solution labels v_i^j, v_i^h, v_i^k, $e_{i,j}$.

As a consequence, for each $\{v_i, v_j\} \in E$, at least one of the sets $\{v_{i,1}, v_{i,2}, v_{i,3}, v_{i,4}\}$, $\{v_{j,1}, v_{j,2}, v_{j,3}, v_{j,4}\}$ is removed. Hence, it is easy to see that the set of labels $\{v_{i,1}, v_{i,2}, v_{i,3}, v_{i,4}\}$ removed by the solution of MinSpecRem(2) over instance (T, S) corresponds to the vertex v_i in a vertex cover of G.

Then, it follows that starting from a vertex cover V' of G we can compute in polynomial time a solution of MinSpecRem(2) that removes at most $3|V| + |V'|$ labels (similarly to Lemma 4), and that starting from a solution of MinSpecRem(2) that removes at most $3|V| + c$ labels, we can compute in polynomial time a vertex cover $|V'| \leqslant c$ of G (similarly to Lemma 5).

Hence we have designed an L-reduction from MVCC to MinSpecRem(2). Since MVCC is APX-hard [4], it follows that also MinSpecRem(2) is APX-hard. □

Please cite this article in press as: R. Dondi et al., Gene tree correction for reconciliation and species tree inference: Complexity and algorithms, Journal of Discrete Algorithms (2013), http://dx.doi.org/10.1016/j.jda.2013.06.001

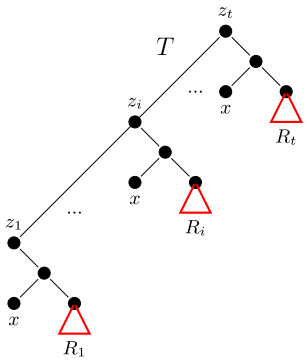

Fig. 8. The gene tree T associated with the set R of triplets.

4. Hardness of Minimum Leaf Removal Inference and Minimum Species Removal Inference

First, we give a reduction from Minimum Rooted Triplets Inconsistency Problem [11] (the dual of the Maximum Compatible Subset of Rooted Triples Problem introduced by Bryant [10]) to MinLeafRemInf. The reduction implies that MinLeafRemInf is not only NP-hard, but it is not approximable within factor $c \ln n$ for some constant $c > 0$, where n is the number of leaves of the input gene tree, unless P = NP, and that the problem is W[2]-hard, if the parameter is the number of leaves that must be removed. We then show that MinSpecRemInf is not only NP-complete, but also W[2]-hard if the parameter is the number of labels that must be removed, with a reduction from the Complement Maximum Agreement Supertree Problem.

Denote by $T|L$ the tree included in T obtained by removing all leaves not in L. A set of trees $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$ is *consistent* if there exists a tree T such that $T|L(T_i) = T_i$ for all i. Otherwise, the set is *inconsistent*. The following problem statement is equivalent to the one given in Byrka et al. [11], since given a set of compatible triplets, we can reconstruct a compatible tree using the algorithm of Aho et al. [2].

Problem 6 *(Minimum Rooted Triplets Inconsistency Problem [11] [MinTripIncon]).*
Input: A set R of rooted triplets over leafset Γ.
Output: A subset $R' \subseteq R$ such that $R \setminus R'$ is consistent, and $|R'|$ is minimum.

Theorem 1. *The* Minimum Leaf Removal Inference *problem is*: (1) *not approximable within factor $c \ln n$, where n is the number of leaves in the input tree, for some constant $c > 0$, unless $P = NP$*; (2) *W[2]-hard, if parameterized by the number of leaves that must be removed.*

Proof. We present a parameterized and approximation preserving reduction from MinTripIncon to MinLeafRemInf. Create the gene tree T from the set of triplets $R = \{R_1, R_2, \ldots, R_t\}$ over leafset Γ by taking the caterpillar on t leaves, having internal nodes $z_1, \ldots, z_t$, and replacing the i-th leaf, $1 \leqslant i \leqslant t$, with the subtree (R_i, x), where x is a label not in Γ (see Fig. 8).

Every internal node in $\{z_2, \ldots, z_t\}$ of T is an AD node (because of the multiple occurrences of label x). The nodes in a subtree R_i, $1 \leqslant i \leqslant t$, of T can be NAD nodes with respect to any species tree, because the set R of triplets is possibly inconsistent.

Consider a solution R' of MinTripIncon, with $|R'| = h$, then we compute in polynomial time a solution T^* of MinLeafRemInf that removes h leaves. For each triplet $R_i \in R'$, we construct T^* by removing an arbitrary leaf from the corresponding subtree of T. Consider the tree U that is consistent with $R \setminus R'$ (we assume that it contains all the labels, otherwise we add them on the top of the tree); then, a species tree S MD-consistent with T^* is defined connecting the root of S with two children: the root of the subtree U and a leaf labelled x. By construction T^* is MD-consistent with S.

Consider a solution T^* of MinLeafRemInf that removes at most h leaves, then we compute in polynomial time a solution R' of MinTripIncon such that $|R'| = h$. Consider the subtree R_i^*, $1 \leqslant i \leqslant t$, of T^*. All such trees are uniquely leaf-labelled, and they must be MD-consistent with some species tree S, hence such subtrees correspond to consistent triplets. Moreover, we can assume that at most one leaf is removed from each subtree R_i^*. Hence, the triplets associated with subtrees where no leaf has been removed are consistent.

Since MinTripIncon is not approximable within factor $d \log |\Gamma|$ [11], for some constant $d > 0$, and $n \leqslant |\Gamma^3| + 1$, it follows that MinLeafRemInf is not approximable within factor $c \log n$, for some constant $c > 0$. Moreover, notice the reduction given in [11] implies that MinTripIncon is W[2]-hard, hence MinLeafRemInf is W[2]-hard. □

We now turn our attention to the MinSpecRemInf problem. We will present a reduction from CMASP (see below) to MinSpecRemInf. Consider a set of trees $\mathcal{U} = \{U_1, U_2, \ldots, U_t\}$ leaf labelled by $\Gamma_1, \Gamma_2, \ldots, \Gamma_t$ respectively. Let $\Gamma = \bigcup_{i=1}^{t} \Gamma_i$. An *agreement supertree* for $\mathcal{U}$ is a tree X leaf labelled by $M \subseteq \Gamma$ such that $X|\Gamma_i = U_i|M$.

Please cite this article in press as: R. Dondi et al., Gene tree correction for reconciliation and species tree inference: Complexity and algorithms, Journal of Discrete Algorithms (2013), http://dx.doi.org/10.1016/j.jda.2013.06.001

Problem 7 *(Complement Maximum Agreement Supertree Problem [CMASP]).*
Input: A set of trees $\mathcal{U} = \{U_1, U_2, \ldots, U_t\}$ leaf labelled by $\Gamma_1, \Gamma_2, \ldots, \Gamma_t$ respectively, and $\Gamma = \bigcup_{i=1}^{t} \Gamma_i$.
Output: An agreement supertree X for $\mathcal{U}$, such that X is over leafset $M \subseteq \Gamma$, and $|\Gamma - M|$ is minimum.

Berry and Nicolas [6] showed that CMASP is NP-hard and W[2]-hard if the parameter is the number of labels that must be removed (that is $|\Gamma - M|$), even when the input trees are triplets. Applying the same construction of Theorem 1, Theorem 2 follows.

Theorem 2. *The* Minimum Species Removal Inference *problem is NP-hard and W*[2]*-hard, if parameterized by the number of labels that must be removed.*

5. Fixed-parameter algorithms for MinLeafRem and MinSpecRem

Since the MinLeafRem and MinSpecRem are APX-hard, it is interesting to see if the problems become tractable under some biological meaningful parameterizations (for an introduction to parameterized complexity see [27]). In this section we focus on the two following parameterizations: (1) the size of the solution of MinLeafRem (that is the number of leaves removed from T in order to obtain a tree MD-consistent with S), and MinSpecRem (that is the number of labels removed from T in order to obtain a tree MD-consistent with S); (2) the number of labels in Γ associated with multiple leaves of T (i.e. the number of genomes containing multiple gene copies). We will give fixed-parameter algorithms for MinLeafRem and MinSpecRem under these two parameterizations.

Notice that a third natural parameter would be the maximum number of leaves in T associated with a single label of Γ (i.e. the maximum number of gene copies in a given genome). However, we have proved in Section 3 that MinLeafRem and MinSpecRem are already APX-hard when each label has at most two occurrences in the gene tree T.

5.1. MinLeafRem *and* MinSpecRem *parameterized by the number of leaves removed*

In this section, we investigate the parameterized complexity of MinLeafRem and MinSpecRem, when parameterized by the size of the solution, that is the number of leaves/species removed from T. We present a fixed-parameter algorithm for MinLeafRem that is based on the depth-bounded search tree technique. The algorithm for MinSpecRem is similar, and we briefly discuss it in Corollary 4.

Denote by c the size of the solution, that is the number of leaves that have to be removed from T in order to get a tree T^* which is MD-consistent with the species tree S.

First, we notice that if T does not contain NAD nodes, then T is MD-consistent with S and it requires no leaf removal. Hence in what follows we assume that T contains at least one NAD node.

Now, consider a NAD node v of T. Let s be the node of S where v is mapped, and let s_l and s_r be the left child and the right child respectively of s. Since v is a NAD node, it follows that at least one of its children, denoted as v_l and v_r, is mapped by $\text{LCA}_{T,S}$ to s. Assume w.l.o.g. that v_l is mapped to s, that is $\text{LCA}_{T,S}(v_l) = s$. Since $\text{LCA}_{T,S}(v_l) = s$, it follows that $\mathcal{C}(v_l) \subseteq \mathcal{C}(s)$, $\mathcal{C}(v_l) \cap \mathcal{C}(s_l) = X_1 \neq \emptyset$ and $\mathcal{C}(v_l) \cap \mathcal{C}(s_r) = X_2 \neq \emptyset$. Hence, in order to obtain a tree T^* that is MD-consistent with S, we have to remove from T: (1) the leaves of $T(v_l)$ having labels in X_1, or (2) the leaves of $T(v_l)$ having labels in X_2, or (3) the leaves of $T(v_r)$. We formally prove this property in the following lemma.

Lemma 6. *Let v be a NAD node of a gene tree T, and let v_l, v_r be the children of v, such that $\text{LCA}_{T,S}(v) = \text{LCA}_{T,S}(v_x) = s$, for some $x \in \{r, l\}$. Let s_l, s_r be the children of s. Then, there is no subtree included in T that is MD-consistent with S and that contains a leaf of $T(v_x)$, for some $x \in \{r, l\}$, with a label in $X_1 = \mathcal{C}(v_x) \cap \mathcal{C}(s_l)$, a leaf of $T(v_x)$ with a label in $X_2 = \mathcal{C}(v_x) \cap \mathcal{C}(s_r)$, and a leaf of $T(v_y)$, with $y \in (\{r, l\} \setminus \{x\})$.*

Proof. Assume that T' is a subtree of T that contains a leaf l_3 of $T(v_y)$ and that is MD-consistent with S. We will show that all the leaves of $T(v_x)$ with a label in X_1 or all the leaves of $T(v_x)$ with a label in X_2 do not belong to T'.

Assume that T' contains one leaf l_1 of $T(v_x)$ with a label in X_1 and one leaf l_2 of $T(v_x)$ with a label in X_2. Let v'_x be the LCA in T' of l_1 and l_2. Let v' be the LCA in T' of l_1 (or l_2) and l_3. It follows that $s = \text{LCA}_{T',S}(v') = \text{LCA}_{T',S}(v'_x)$, hence v' would be a NAD, and T' would not be MD-consistent with S. □

Due to Lemma 6, we can design a fixed-parameter algorithm for MinLeafRem parameterized by the number of leaves c that have to be removed. Let $Dup(T) = \langle v^1, \ldots, v^z \rangle$ be the ordered list of NAD nodes of T in a breadth-first visit of T. The algorithm at each step chooses the first node v^1 of $Dup(T)$. Let $\text{LCA}_{T,S}(v^1) = s$, and let s_l and s_r be the two children of s. Consider a child v^1_x, with $v^1_x \in \{v^1_l, v^1_r\}$, of v^1 that is mapped to s, and let $v^1_{\bar{x}}$ be the other child of v^1. Let $\mathcal{C}(v^1_x) \cap \mathcal{C}(s_l) = X_1 \neq \emptyset$, $\mathcal{C}(v^1_x) \cap \mathcal{C}(s_r) = X_2 \neq \emptyset$.

Now, the algorithm branches on the following cases:

1. Remove the leaves of $T(v^1_x)$ with labels in X_1 from $L(T)$ and suppress the resulting degree two nodes;

2. Remove the leaves of $T(v_{\bar{x}}^1)$ with labels in X_2 from $L(T)$ and suppress the resulting degree two nodes;
3. Remove the subtree $T(v_{\bar{x}}^1)$ from T, and suppress the resulting degree two nodes.

After the branching, the algorithm outputs a subtree T' of T. Then the LCA mapping $\text{LCA}_{T',S}$ between T' and S is computed (in polynomial time), and the ordered list $Dup(T')$ of NAD nodes of T' is computed (again in polynomial time). The algorithm stops either when it finds a subtree T' of T that is MD-consistent with S, or when there is no subtree included in T that can be obtained with c leaf removals.

Theorem 3. *The algorithm computes if there exists a solution of size at most c for* MinLeafRem *in time* $O(3^c \text{ poly}(|V(T)|, |V(S)|))$.

Proof. The correctness of the algorithm follows from Lemma 6.

Now, we focus on the time complexity of the algorithm. At each step the algorithm branches in three possible cases, and for each of these cases at least one leaf is removed. As the depth of the search tree is bounded by c, the size of the search tree is bounded by $O(3^c)$. Since after each branching we require at most time $O(\text{poly}(|V(T)||V(S)|))$ to compute T', $\text{LCA}_{T',S}$, and $Dup(T')$, it follows that the overall time complexity of the algorithm is $O(3^c \text{ poly}(|V(T)||V(S)|))$. □

Next, we show a similar result for MinSpecRem.

Corollary 4. *There exists a fixed-parameter tractable algorithm that computes if there exists a solution of size at most c for* MinSpecRem *in time* $O(3^c \text{ poly}(|V(T)|, |V(S)|))$.

Proof. The algorithm follows very closely the algorithm for MinLeafRem, except that, given a NAD node, in this case the algorithm considers the possible *label removals*, instead of *leaf removals*.

Formally, consider a NAD node v of T. Let s be the node of S where v is mapped. Since v is a NAD node, and assuming that v_l is mapped in s, it follows that $\mathcal{C}(v_l) \cap \mathcal{C}(s_l) = X_1 \neq \emptyset$ and $\mathcal{C}(v_l) \cap \mathcal{C}(s_r) = X_2 \neq \emptyset$. Hence, in order to obtain a tree T^* that is MD-consistent with S, we have to remove: (1) the leaves of T having labels in X_1, or (2) the leaves of T having labels in X_2, or (3) the leaves of T having labels in $\Gamma(T(v_r))$.

The correctness of the algorithm follows closely to that of MinLeafRem. □

5.2. MinLeafRem *and* MinSpecRem *parameterized by the number of labels with multiple copies*

In this section we give a fixed-parameter algorithm for MinLeafRem and MinSpecRem, when the parameter is the number of labels associated with multiple leaves of T. Denote by $\Gamma_D \subseteq \Gamma$, the subset of labels associated with multiple leaves of T. Next, we describe the algorithm for MinLeafRem, then we discuss the case of MinSpecRem in Theorem 6.

The algorithm is based on dynamic programming. Let x be a node of T, having children x_l, x_r, and let y be a node of S, with children y_l, y_r. Given $\Gamma'_D \subseteq \Gamma_D$, we define $M[T(x), S(y), \Gamma'_D]$ as the minimum number of leaves that have to be removed to obtain a tree T' included in $T(x)$ such that (1) T' is MD-consistent with $S(y)$ and (2) the subset $\Gamma'_D \subseteq \Gamma(T')$. We can compute $M[T(x), S(y), \Gamma'_D]$ applying the following dynamic programming recurrence:

$$M\big[T(x), S(y), \Gamma'_D\big] = \min_{\substack{\Gamma'_{1,D} \subseteq \Gamma'_D, \\ \Gamma'_{2,D} \subseteq \Gamma'_D, \\ \Gamma'_{1,D} \cup \Gamma'_{2,D} = \Gamma'_D}} \begin{cases} M[T(x_l), S(y_l), \Gamma'_{1,D}] + M[T(x_r), S(y_r), \Gamma'_{2,D}] \\ \quad \text{if } \Gamma'_{1,D} \cap \Gamma'_{2,D} = \emptyset, \\ M[T(x_l), S(y_r), \Gamma'_{1,D}] + M[T(x_r), S(y_l), \Gamma'_{2,D}] \\ \quad \text{if } \Gamma'_{1,D} \cap \Gamma'_{2,D} = \emptyset, \\ M[T(x_l), S(y), \Gamma'_{1,D}] + M[T(x_r), S(y), \Gamma'_{2,D}] \\ \quad \text{if } \Gamma'_{1,D} \cap \Gamma'_{2,D} \neq \emptyset, \\ M[T(x_l), S(y), \Gamma'_D] + |L(T(x_r))|, \\ M[T(x_r), S(y), \Gamma'_D] + |L(T(x_l))|, \\ M[T(x), S(y_l), \Gamma'_D], \\ M[T(x), S(y_r), \Gamma'_D]. \end{cases} \tag{1}$$

Now, we define the base cases of the recurrence, when each of $T(x)$ and $S(y)$ is a single leaf, with $\Gamma(T(x)) = \lambda_G$ and $\Gamma(S(y)) = \lambda_S$. If $\lambda_G = \lambda_S$, then $M[T(x), S(y), \Gamma'_D] = 0$ if $\Gamma'_D \in (\emptyset \cup \{\lambda_G\})$, otherwise if $\lambda_G = \lambda_S$ and $\Gamma'_D \notin (\emptyset \cup \{\lambda_G\})$, then $M[T(x), S(y), \Gamma'_D] = +\infty$. If $\lambda_G \neq \lambda_S$, then $M[T(x), S(y), \Gamma'_D] = 1$ if $\Gamma'_D = \emptyset$, otherwise $M[T(x), S(y), \Gamma'_D] = +\infty$.

The correctness of Recurrence (1), is proved in the following.

Lemma 7. *Let T be a gene tree, let S be a species tree, and let $\Gamma_D \subseteq \Gamma$ be the set of labels associated with multiple leaves of T. Let x be a node of T and y be a node of S, and consider a subset $\Gamma'_D \subseteq \Gamma_D$. Then if $M[T(x), S(y), \Gamma'_D] = c$, there exists a tree T' included in $T(x)$ such that* (i) *T' is MD-consistent with $S(y)$*; (ii) *T' is obtained by removing c leaves*; *and* (iii) *$\Gamma'_D \subseteq \Gamma(T')$.*

Please cite this article in press as: R. Dondi et al., Gene tree correction for reconciliation and species tree inference: Complexity and algorithms, Journal of Discrete Algorithms (2013), http://dx.doi.org/10.1016/j.jda.2013.06.001

Proof. We prove the lemma by induction on the number of leaves of $|L(T(x))| + |L(S(y))|$.

Assume that $|L(T(x))| + |L(S(y))| = 2$ and $\Gamma(T(x)) = \Gamma(S(y)) = \lambda$. Then, if $\Gamma'_D = \{\lambda\}$ it follows that $M[T(x), S(y), \lambda'_D] = 0$, if $\Gamma'_D = \emptyset$ it follows that $M[T(x), S(y), \lambda'_D] = 1$, otherwise $M[T(x), S(y), \Gamma'_D] = +\infty$. If $\Gamma(T(x)) = \lambda_1$ and $\Gamma(S(y)) = \lambda_2$ with $\lambda_1 \neq \lambda_2$, then $M[T(x), S(y), \Gamma'_D] = 1$ if $\Gamma'_D = \emptyset$, otherwise $M[T(x), S(y), \Gamma'_D] = +\infty$.

Assume now that the lemma holds when $|L(T(x))| + |L(S(y))| \leqslant n$, we show that the lemma holds when $|L(T(x))| + |L(S(y))| = n + 1$. Assume that $M[T(x), S(y), \Gamma'_D] = c$, we show that there exists a subtree $T'(x)$ included in $T(x)$ such that (i) $T'(x)$ is MD-consistent with $S(y)$; (ii) $T'(x)$ is obtained from $T(x)$ by removing c leaves; (iii) $\Gamma'_D \subseteq \Gamma(T'(x))$.

Assume that $M[T(x), S(y), \Gamma'_D]$ is obtained by one of the first two cases of Recurrence (1), w.l.o.g. the first case. It holds that $M[T(x_l), S(y_l), \Gamma'_{1,D}] = c_1$, $M[T(x_r), S(y_r), \Gamma'_{2,D}] = c_2$, where $c_1 + c_2 = c$. By the induction hypothesis there exists a subtree $T'(x_l)$ ($T'(x_r)$ respectively) included in $T(x_l)$ (in $T(x_r)$ respectively) that is MD-consistent with $S(y_l)$ ($S(y_r)$ respectively), obtained by removing c_1 leaves (c_2 leaves respectively) and containing a set of leaves labelled by $\Gamma'_{1,D}$ (by $\Gamma'_{2,D}$ respectively). Since $\Gamma'_{1,D} \cap \Gamma'_{2,D} = \emptyset$ and $\Gamma(S(y_l)) \cap \Gamma(S(y_r)) = \emptyset$, it follows that the subtree $T'(x)$ of $T(x)$ consisting of a root that joins the two subtrees $T'(x_l)$ and $T'(x_r)$, has the following property: $\Gamma(T'(x_l)) \cap \Gamma(T'(x_r)) = \emptyset$. Furthermore, by induction, $T'(x)$ is MD-consistent with $S(y)$ and it is obtained by removing $c = c_1 + c_2$ leaves from $T(x)$.

Assume that $M[T(x), S(y), \Gamma'_D]$ is obtained by case 3 of Recurrence (1). It holds that $M[T(x_l), S(y), \Gamma'_{1,D}] = c_1$, $M[T(x_r), S(y), \Gamma'_{2,D}] = c_2$, where $c_1 + c_2 = c$. Then, by the induction hypothesis there is a subtree $T'(x_l)$ ($T'(x_r)$ respectively) included in $T(x_l)$ ($T(x_r)$ respectively) that is MD-consistent with $S(y)$, obtained by removing c_1 leaves (c_2 leaves respectively) and containing a set of leaves labelled by $\Gamma'_{1,D}$ (by $\Gamma'_{2,D}$ respectively), with $\Gamma'_{1,D} \cap \Gamma'_{2,D} \neq \emptyset$. It follows that the subtree $T'(x)$ of $T(x)$ consisting of a root that joins the two subtrees $T'(x_l)$ and $T'(x_r)$, is MD-consistent with $S(y)$, and it is obtained by removing $c = c_1 + c_2$ leaves.

Assume that $M[T(x), S(y), \Gamma'_D]$ is obtained by case 4 or 5 of Recurrence (1), w.l.o.g. take case 4. It holds that $M[T(x_l), S(y), \Gamma'_D] = c_1$, for some value $c_1 \geqslant 0$, hence there is a subtree $T'(x_l)$ of $T(x_l)$ obtained by removing c_1 leaves of $T(x_l)$ and such that all the leaves of $T(x_r)$, with $c_1 + |L(T(x_r))| = c$. Furthermore, by induction the subtree $T'(x_l)$ is MD-consistent with $S(y)$.

Assume that $M[T(x), S(y), \Gamma'_D]$ is obtained by case 6 or 7 of Recurrence (1), w.l.o.g. take case 6. It holds that $M[T(x), S(y_l), \Gamma'_D] = c$, hence there is a subtree $T'(x)$ of $T(x)$ obtained by removing c leaves of $T(x)$ such that $L(T(x')) \subseteq L(S(y_l))$. Furthermore, by induction the subtree $T'(x)$ is MD-consistent with $S(y_l)$, hence also with $S(y)$. □

Lemma 8. *Let T be a gene tree, let S be a species tree, and let $\Gamma_D \subseteq \Gamma$ be the set of labels associated with multiple leaves of T. Let x be a node of T and y be a node of S, and consider a subset $\Gamma'_D \subseteq \Gamma_D$. Then if there exists a tree T' included in $T(x)$ such that* (i) *T' is MD-consistent with $S(y)$*; (ii) *T' is obtained by removing c leaves*; *and* (iii) *$\Gamma'_D \subseteq \Gamma(T')$, it follows that $M[T(x), S(y), \Gamma'_D] = c$.*

Proof. We prove the lemma by induction.

Assume that $|L(T(x))| + |L(S(y))| = 2$, and $\Gamma(T(x)) = \Gamma(S(y)) = \lambda$. If $\Gamma'_D = \{\lambda\}$, then, $M[T(x), S(y), \lambda'_D] = 0$; if $\Gamma'_D = \emptyset$, then $M[T(x), S(y), \lambda'_D] = 1$; else $M[T(x), S(y), \Gamma'_D] = +\infty$. If $\Gamma(T(x)) = \lambda_1$, $\Gamma(S(y)) = \lambda_2$, with $\lambda_1 \neq \lambda_2$, then if $\Gamma'_D = \emptyset$, $M[T(x), S(y), \Gamma'_D] = 1$, else $M[T(x), S(y), \Gamma'_D] = +\infty$.

Assume that there exists a subtree $T'(x)$ included in $T(x)$ such that (i) $T'(x)$ is MD-consistent with $S(y)$; (ii) $T'(x)$ is obtained by removing c leaves; and (iii) $\Gamma'_D \subseteq \Gamma(T'(x))$. Then, we will show that $M[T(x), S(y), \Gamma'_D] = c$. Denote with r' the root of $T'(x)$.

Notice that if $T'(x)$ is a subtree included in $T(x_l)$ ($T(x_r)$ respectively), then case 4 (case 5 respectively) of Recurrence (1) applies, and by induction the lemma holds. Similarly, if $\Gamma(T'(x)) \subseteq \Gamma(S(y_l))$ ($\Gamma(T'(x)) \subseteq L(S(y_r))$ respectively) holds, then case 6 (case 7 respectively) of Recurrence (1) applies, and by induction the lemma holds.

Now, consider the root r' of $T'(x)$. Let r'_l and r'_r be the two children of r'. Notice that $T'(x)$ is not a subtree of $T(x_l)$ or $T(x_r)$, otherwise by the previous argument the lemma holds. Since $T'(x)$ is MD-consistent with $S(y)$, then r' can be either a speciation or an AD node. First, assume that r' is a speciation. As a consequence r'_l and r'_r, are mapped by $\text{LCA}_{T'(x),S(y)}$ to nodes of distinct subtrees $S(y_l)$ and $S(y_r)$. Assume w.l.o.g. that r'_l and r'_r are mapped by $\text{LCA}_{T'(x),S(y)}$ to nodes of $S(y_l)$ and $S(y_r)$ respectively. Since $T'(x)$ is obtained by removing $c \geqslant 0$ leaves from $T(x)$, then $T'(r'_l)$ is obtained by removing $c_1 \geqslant 0$ leaves from $T(x_l)$, while $T'(r'_r)$ is obtained by removing $c_2 \geqslant 0$ leaves from $T(x_r)$, with $c_1 + c_2 = c$. Furthermore, $T'(r'_l)$ is MD-consistent with $S(y_l)$ and contains a set of leaves labelled by $\Gamma'_{1,D}$, while $T'(r'_r)$ is MD-consistent with $S(y_r)$ and contains a set of leaves labelled by $\Gamma'_{2,D}$. Hence $T'(x)$ contains a set of leaves labelled by Γ'_D, with $\Gamma'_D = \Gamma'_{1,D} \cup \Gamma'_{2,D}$. By induction hypothesis, $M[T(x_l), S(y_l), \Gamma'_{1,D}] = c_1$, $M[T(x_r), S(y_r), \Gamma'_{2,D}] = c_2$, with $c_1 + c_2 = c$. By case 1 of Recurrence (1) the lemma holds. The case that r'_l and r'_r are mapped in nodes of $S(y_r)$ and $S(y_l)$ respectively, is similar, and by case 2 of Recurrence (1) the lemma holds.

Assume now that r' is an AD node, that is $\Gamma(T'(r'_l)) \cap \Gamma(T'(r'_l)) = \Gamma^+_D \neq \emptyset$. Now, consider the subtrees $T'(r'_l)$, $T'(r'_r)$, which w.l.o.g. are subtrees included in $T(x_l)$, $T(x_r)$ respectively. Then $T'(r'_l)$ is obtained by removing $c_1 \geqslant 0$ leaves of $T(x_l)$, while $T'(r'_r)$ is obtained by removing $c_2 \geqslant 0$ leaves of $T(x_r)$, with $c_1 + c_2 = c$. Furthermore, since r' is an AD node, there exist two subsets $\Gamma'_{1,D}, \Gamma'_{2,D}$ with $\Gamma'_{1,D} \subseteq \Gamma(T'(r'_l))$, and $\Gamma'_{2,D} \subseteq \Gamma(T'(r'_r))$ such that $\Gamma'_{1,D} \cap \Gamma'_{2,D} \neq \emptyset$. By induction hypothesis, $M[T(x), S(y), \Gamma'_{1,D}] = c_1$ and $M[T(x_r), S(y), \Gamma'_{2,D}] = c_2$, with $c_1 + c_2 = c$. Then by case 3 of Recurrence (1), $M[T(x), S(y), \Gamma'_D] = c$, with $\Gamma'_D = \Gamma'_{1,D} \cup \Gamma'_{2,D}$, and the lemma follows. □

Please cite this article in press as: R. Dondi et al., Gene tree correction for reconciliation and species tree inference: Complexity and algorithms, Journal of Discrete Algorithms (2013), http://dx.doi.org/10.1016/j.jda.2013.06.001

Theorem 5. *Given a gene tree T and a species tree S, let $\Gamma_D \subseteq \Gamma$ be the set of labels associated with multiple leaves of T. Then an optimal solution of* MinLeafRem *over instance (T, S) can be computed in time $O(4^{|\Gamma_D|}\,\text{poly}(|V(T)||V(S)|))$.*

Proof. By Lemma 7 a solution of MinLeafRem over instance (T, S) is obtained looking for the minimum of the values $M[T(r_T), S(r_S), \Gamma'_D]$, for each $\Gamma'_D \subseteq \Gamma_D$, where r_T (r_S respectively) is the root of T (S respectively).

Now, we prove in the following that the time complexity of the algorithm is $O(4^{|\Gamma_D|}\,\text{poly}(|V(T)||V(S)|))$. It is easy to see that the time complexity to compute Recurrence (1) is dominated by case 3. The entries $M[T(x), S(y), \Gamma'_D]$ are $O(2^{|\Gamma_D|}|V(T)||V(S)|)$. For each pair of nodes $x \in V(T)$, $y \in V(S)$, we have to consider $O(4^{|\Gamma_D|})$ possible combinations. Indeed, the number of subsets $\Gamma'_{1,D}, \Gamma'_{2,D} \subseteq \Gamma'_D$, with $\Gamma'_D = \Gamma'_{1,D} \cup \Gamma'_{2,D}$, is $4^{|\Gamma_D|}$, since we have to consider all possible subsets Γ'_D of Γ_D and, for each subset Γ'_D, we have to consider all possible subsets $\Gamma'_{1,D}, \Gamma'_{2,D} \subseteq \Gamma'_D$, with $\Gamma'_D = \Gamma'_{1,D} \cup \Gamma'_{2,D}$. It follows that we have to consider $4^{|\Gamma_D|}$ combinations, since there are $4^{|\Gamma_D|}$ possible ways to split set Γ_D into four disjoint subsets (in this case the subsets are $\Gamma_D \setminus \Gamma'_D$, $\Gamma'_{1,D} \setminus \Gamma'_{2,D}$, $\Gamma'_{2,D} \setminus \Gamma'_{1,D}$, and $\Gamma'_{1,D} \cap \Gamma'_{2,D}$). For each combination, the recursion can be computed in constant time.

Finding the minimum value in the entries $M[T(r_G), S(r_S), \Gamma'_D]$ requires time $O(2^{|\Gamma_D|}|V(T)||V(S)|)$, hence the overall time complexity to find an optimal solution of MinLeafRem over instance (T, S), is $O(4^{|\Gamma_D|}\,\text{poly}(|V(T)||V(S)|))$. □

Next, we show that an equivalent result can be proved for MinSpecRem.

Theorem 6. *Given a gene tree T and a species tree S, let $\Gamma_D \subseteq \Gamma$ be the set of labels associated with multiple leaves of T. Then an optimal solution of* MinSpecRem *over instance (T, S) can be computed in time $O(4^{|\Gamma_D|}\,\text{poly}(|V(T)||V(S)|))$.*

Proof. Similarly to MinLeafRem, define $M[T(x), S(y), \Gamma'_D]$ as the minimum number of *labels* that have to be removed to obtain a tree $T'(x)$ included in $T(x)$ such that (1) T' is MD-consistent with $S(y)$, (2) the subset $\Gamma'_D \subseteq \Gamma(T')$, and (3) a leaf of $L(T(x))$ labelled by Γ_D belongs of $T'(x)$ if and only if it is associated with a label in Γ'_D.

The dynamic programming algorithm for MinSpecRem is based on the following recurrence:

$$M\big[T(x), S(y), \Gamma'_D\big] = \min_{\substack{\Gamma'_{1,D} \subseteq \Gamma'_D,\\ \Gamma'_{2,D} \subseteq \Gamma'_D,\\ \Gamma'_{1,D} \cup \Gamma'_{2,D} = \Gamma'_D}} \begin{cases} M[T(x_l), S(y_l), \Gamma'_{1,D}] + M[T(x_r), S(y_r), \Gamma'_{2,D}] \\ \quad \text{if } \Gamma'_{1,D} \cap \Gamma'_{2,D} = \emptyset,\ \Gamma'_{2,D} \cap \Gamma(T(x_l)) = \emptyset, \text{ and } \Gamma'_{1,D} \cap \Gamma(T(x_r)) = \emptyset, \\ M[T(x_l), S(y_r), \Gamma'_{1,D}] + M[T(x_r), S(y_l), \Gamma'_{2,D}] \\ \quad \text{if } \Gamma'_{1,D} \cap \Gamma'_{2,D} = \emptyset,\ \Gamma'_{2,D} \cap \Gamma(T(x_l)) = \emptyset, \text{ and } \Gamma'_{1,D} \cap \Gamma(T(x_r)) = \emptyset, \\ M[T(x_l), S(y), \Gamma'_{1,D}] + M[T(x_r), S(y), \Gamma'_{2,D}] \\ \quad \text{if } \Gamma'_{1,D} \cap \Gamma'_{2,D} \neq \emptyset,\ (\Gamma'_{2,D} \setminus \Gamma'_{1,D}) \cap \Gamma(T(x_l)) = \emptyset, \text{ and} \\ \quad (\Gamma'_{1,D} \setminus \Gamma'_{2,D}) \cap \Gamma(T(x_r)) = \emptyset, \\ M[T(x_l), S(y), \Gamma'_D] + |\Gamma(T(x_r))| \\ \quad \text{if } \Gamma'_D \cap \Gamma(T(x_r)) = \emptyset, \\ M[T(x_r), S(y), \Gamma'_D] + |\Gamma(T(x_l))| \\ \quad \text{if } \Gamma'_D \cap \Gamma(T(x_l)) = \emptyset, \\ M[T(x), S(y_l), \Gamma'_D], \\ M[T(x), S(y_r), \Gamma'_D]. \end{cases} \tag{2}$$

The base cases of the recurrence are identical to those for MinLeafRem.

The dynamic programming closely follows that for MinLeafRem, except for the conditions that ensures that, when a label is removed, there is no occurrence of that label associated with some leaves of the solution.

We consider the first case. The condition $\Gamma'_{2,D} \cap \Gamma(T(x_l)) = \emptyset$ ensures that no leaf l having a label σ in $\Gamma'_{1,D}$ belongs to $T(x_r)$, otherwise this label (hence also l) would be removed. Indeed, by definition of $M[T(x), S(y), \Gamma'_D]$, among the leaves labelled by Γ_D, only the leaves of $T(x_l)$ having labels in $\Gamma'_{1,D}$, with $\Gamma'_{1,D} \cap \Gamma'_{2,D} = \emptyset$, are not removed. But then, if l is not removed, there would be a leaf with a label $\sigma \in \Gamma'_{1,D}$ that belongs to the solution, and one label with label σ that has been removed. □

6. Conclusion

In this paper, we present complexity results for problems related to the preprocessing of gene trees for use in reconciliation and species tree inference, following the approach of [12]. We prove that two combinatorial problems presented MinLeafRem, MinSpecRem are APX-hard, even when each label is associated with at most two leaves of the input gene tree, and we present fixed-parameter algorithms for MinLeafRem and MinSpecRem. Furthermore, we prove that MinLeafRemInf is not only NP-hard, but also W[2]-hard (when parameterized by the size of the solution, that is the minimum number of leaf

Please cite this article in press as: R. Dondi et al., Gene tree correction for reconciliation and species tree inference: Complexity and algorithms, Journal of Discrete Algorithms (2013), http://dx.doi.org/10.1016/j.jda.2013.06.001

removals) and inapproximable within factor $c \ln n$, where n is the number of leaves in the gene tree. Finally we show that MinSpecRemInf is NP-hard and W[2]-hard, when parameterized by the size of the solution, that is the minimum number of species removals.

An interesting open problem is to further study the approximation complexity of MinLeafRem and MinSpecRem. Is it possible to have constant factor approximation algorithms for the two problems? Another interesting open problem is to improve the time complexity of the parameterized algorithms given in Section 5.1. Finally, kernelization issues of MinLeafRem and MinSpecRem are left completely unexplored.

References

[1] S. Abby, E. Tannier, M. Gouy, V. Daubin, Lateral gene transfer as a support for the tree of life, Proceedings of the National Academy of Sciences of the United States of America 109 (2012) 4962–4967.
[2] A.V. Aho, Y. Sagiv, T.G. Szymanski, J.D. Ullman, Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions, SIAM Journal on Computing 10 (3) (1981) 405–421.
[3] O. Akerborg, B. Sennblad, L. Arvestad, J. Lagergren, Simultaneous bayesian gene tree reconstruction and reconciliation analysis, Proceedings of the National Academy of Sciences of the United States of America 106 (14) (2009) 5714–5719.
[4] P. Alimonti, V. Kann, Some APX-completeness results for cubic graphs, Theoretical Computer Science 237 (1–2) (2000) 123–134.
[5] L. Arvestad, A.C. Berglung, J. Lagergren, B. Sennblad, Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution, in: D. Gusfield (Ed.), RECOMB 2004, ACM, New York, 2004, pp. 326–335.
[6] V. Berry, F. Nicolas, Maximum agreement and compatible supertrees, Journal of Discrete Algorithms 5 (3) (2007) 564–591.
[7] G. Blin, P. Bonizzoni, R. Dondi, R. Rizzi, F. Sikora, Complexity insights of the minimum duplication problem, in: M. Bieliková, G. Friedrich, G. Gottlob, S. Katzenbeisser, G. Turán (Eds.), SOFSEM 2012, in: LNCS, vol. 7147, Springer, Heidelberg, 2012, pp. 153–164.
[8] P. Bonizzoni, G. Della Vedova, R. Dondi, Reconciling a gene tree to a species tree under the duplication cost model, Theoretical Computer Science 347 (2005) 36–53.
[9] B. Boussau, G. Szollosi, et al., L.D.: Genome-scale coestimation of species and gene trees, Genome Research 23 (2013) 323–330.
[10] D. Bryant, Building trees, hunting for trees and comparing trees: Theory and method in phylogenetic analysis, PhD dissertation, Department of Mathematics, University of Canterbury, UK, 1997.
[11] J. Byrka, S. Guillemot, J. Jansson, New results on optimizing rooted triplets consistency, Discrete Applied Mathematics 158 (11) (2010) 1136–1147.
[12] C. Chauve, N. El-Mabrouk, New perspectives on gene family evolution: losses in reconciliation and a link with supertrees, in: S. Batzoglou (Ed.), RECOMB 2009, in: LNCS, vol. 5541, Springer, Heidelberg, 2009, pp. 46–58.
[13] K. Chen, D. Durand, M. Farach-Colton, Notung: Dating gene duplications using gene family trees, Journal of Computational Biology 7 (2000) 429–447.
[14] Y. Chung, C. Ané, Comparing two bayesian methods for gene tree/species tree reconstruction: Simulations with incomplete lineage sorting and horizontal gene transfer, Systematic Biology 60 (3) (2011) 261–275, http://sysbio.oxfordjournals.org/content/60/3/261.abstract.
[15] R. Dondi, N. El-Mabrouk, Minimum leaf removal for reconciliation: Complexity and algorithms, in: J. Kärkkäinen, J. Stoye (Eds.), CPM, in: LNCS, vol. 7354, Springer, 2012, pp. 399–412.
[16] A. Doroftei, N. El-Mabrouk, Removing noise from gene trees, in: T.M. Przytycka, M.F. Sagot (Eds.), WABI 2011, in: LNBI, vol. 6833, Springer, Heidelberg, 2011, pp. 76–91.
[17] J. Doyon, V. Ranwez, V. Daubin, V. Berry, Models, algorithms and programs for phylogeny reconciliation, Briefings in Bioinformatics 12 (5) (2011) 392–400.
[18] D. Durand, B. Haldórsson, B. Vernot, A hybrid micro-macroevolutionary approach to gene tree reconstruction, Journal of Computational Biology 13 (2006) 320–335.
[19] G. Fang, N. Bhardwaj, R. Robilotto, M. Gerstein, Getting started in gene orthology and functional analysis, PLoS Computational Biology 6 (3) (2010) e1000703, http://dx.doi.org/10.1371/journal.pcbi.1000703.s001.
[20] M. Goodman, J. Czelusniak, G. 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 Zoology 28 (1979) 132–163.
[21] P. Górecki, O. Eulenstein, A linear time algorithm for error-corrected reconciliation of unrooted gene trees, in: J. Chen, J. Wang, A. Zelikovsky (Eds.), ISBRA 2012, in: LNCS, vol. 6674, Springer, Heidelberg, 2011, pp. 148–159.
[22] P. Gorecki, J. Tiuryn, DLS-trees: a model of evolutionary scenarios, Theoretical Computer Science 359 (2006) 378–399.
[23] R. Guigó, I. Muchnik, T. Smith, Reconstruction of ancient molecular phylogeny, Molecular Phylogenetics and Evolution 6 (1996) 189–213.
[24] M. Hahn, Bias in phylogenetic tree reconciliation methods: implications for vertebrate genome evolution, Genome Biology 8 (2007) R141.
[25] H. Li, A. Coghlan, J. Ruan, L. Coin, et al., TreeFam: a curated database of phylogenetic trees of animal gene families, Nucleic Acids Research 34 (2006) D572–D580.
[26] B. Ma, M. Li, L. Zhang, From gene trees to species trees, SIAM Journal on Computing 30 (2000) 729–752.
[27] R. Niedermeier, Invitation to Fixed-Parameter Algorithms, Oxford University Press, Oxford, 2006.
[28] S. Ohno, Evolution by Gene Duplication, Springer, Berlin, 1970.
[29] R. Page, Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas, Systematic Biology 43 (1994) 58–77.
[30] R. Page, M. Charleston, Reconciled trees and incongruent gene and species trees, DIMACS Series in Discrete Mathematics and Theoretical Computer Science 37 (1997) 57–70.
[31] M. Rasmussen, M. Kellis, A bayesian approach for fast and accurate gene tree reconstruction, Molecular Biology and Evolution 28 (1) (2011) 273–290.
[32] M. Sanderson, M. McMahon, Inferring angiosperm phylogeny from EST data with widespread gene duplication, BMC Evolutionary Biology 7 (2007) S3.
[33] C. Scornavacca, V. Berry, V. Ranwez, From gene trees to species trees through a supertree approach, in: The Third Int. Conf. on Language and Automata Theory and Applications (LATA), in: LNCS, vol. 5457, Springer, 2009, pp. 702–714.
[34] K. Swenson, A. Doroftei, N. El-Mabrouk, Gene tree correction for reconciliation and species tree inference, Algorithms for Molecular Biology 7 (2012) 31.
[35] P. Thomas, GIGA: a simple, efficient algorithm for gene tree inference in the genomic age, BMC Bioinformatics 11 (2010) 312.
[36] P. Thomas, M. Campbell, et al., A.K.: PANTHER: a library of protein families and subfamilies indexed for function, Genome Research 13 (2003) 2129–2141.
[37] I. Wapinski, A. Pfeffer, N. Friedman, A. Regev, Natural history and evolutionary principles of gene duplication in fungi, Nature 449 (2007) 54–61.
[38] Y. Yu, J.H. Degnan, L. Nakhleh, The probability of a gene tree topology within a phylogenetic network with applications to hybridization detection, PLoS Genetics 8 (4) (2012) e1002660, http://dx.doi.org/10.1371/journal.pgen.1002660.

Please cite this article in press as: R. Dondi et al., Gene tree correction for reconciliation and species tree inference: Complexity and algorithms, Journal of Discrete Algorithms (2013), http://dx.doi.org/10.1016/j.jda.2013.06.001