HAL
open science

Branch-and-Bound Approach for Parsimonious Inference of a Species Tree From a Set of Gene Family Trees

Jean-Philippe Doyon, Cedric Chauve

To cite this version:

Jean-Philippe Doyon, Cedric Chauve. Branch-and-Bound Approach for Parsimonious Inference of a Species Tree From a Set of Gene Family Trees. RR-10001, 2010, pp.18. lirmm-00448481v1

HAL Id: lirmm-00448481

https://hal-lirmm.ccsd.cnrs.fr/lirmm-00448481v1

Submitted on 19 Jan 2010 (v1), last revised 3 May 2010 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Branch-and-Bound approach for parsimonious inference of a species tree from a set of gene family trees

Jean-Philippe Doyon[1,3] and Cedric Chauve[2⋆]

[1] LIRMM, Université Montpellier2 and CNRS, UMR 5506 - CC 477, 161 rue Ada 34392 Montpellier Cedex 5, France. Jean-philippe.Doyon@lirmm.fr

[2] Department of Mathematics, Simon Fraser University, 8888 University Drive, V5A 1S6, Burnaby (BC), Canada, cedric.chauve@sfu.ca

[3] DIRO, Université de Montréal, CP6128, succ. Centre-Ville, H3C 3J7, Montréal (QC),Canada,

Abstract. We describe a Branch-and-Bound algorithm for computing a parsimonious species tree given a set of gene family trees. Our algorithm can compute a parsimonious species tree for three cost measures: number of gene duplications, number of gene losses, and both combined. Moreover, to cope with intrinsic limitations of Branch-and-Bound algorithms for species trees inference regarding the number of taxa that can be considered, our algorithm can naturally take into account predefined relationships between sets of taxa. We test our algorithm on a dataset of eukaryotic gene families spanning 29 taxa.

Keywords. Comparative genomics, Evolution and phylogenetics.

1 Introduction

Speciation is the fundamental mechanism of genome evolution, especially for eukaryotic genomes. However, other events can happen, that do not result immediately in the creation of new species but act as fundamental evolutionary mechanisms, such as gene duplication and loss [9][4]. Duplication is the genomic process where one or more genes of a single genome are copied, resulting in two copies of each duplicated gene. Gene duplication allows one copy to possibly develop a new biological function through point mutation, while the other copy often preserves its original role. A gene is considered to be lost when the corresponding sequence has been deleted by a genomic rearrangement or has completely lost any functional role (i.e. has become a pseudogene). (See [9] for example). Genes of contemporary species that evolved from a common ancestor, through speciations and duplications, are said to be homologs [7] and are grouped into a gene family. Such gene families are in general inferred using protein sequence comparison.

The availability of large datasets of gene families makes now possible to perform genome-scale phylogenetic analyses. A widely used approach, named Gene Tree Parsimony (GTP for short), is based on the notion of *reconciliation* between a gene tree and a species tree introduced in [8], and seeks a species tree with a minimum overall reconciliation cost with the whole set of input gene trees. Given a gene tree G and a species tree S for the corresponding taxa, the reconciliation cost is the minimum number of duplications, losses, or mutations (duplications plus losses) that is needed to explain the (possible) discrepancies between G and S. Computing a most parsimonious reconciliation between a given gene tree and a species tree can be done in linear time [22], but inferring a parsimonious species tree is an NP-complete problem for both duplication and mutation criteria [12], although fixed-parameter tractable algorithms have been described in [13, 20]. Hence, in most cases, studies based on GTP use either a brute-force approach when the number of taxa is

⋆ Contact author

[4] Other genomic events such as lateral gene transfer, that occurs mostly in bacterial genomes, will not be considered here.

low, as in [18], greedy heuristics [4, 5] or the local search approach with edit operations on species trees such as the Subtree Pruning and Regrafting (rSPR) [14, 1] or Nearest-Neighbour Interchange (NNI) [2]. Although such heuristics, especially local-search ones, are fast and proved to be effective on large datasets [21], they do not guarantee to infer an optimal species tree.

A Branch-and-Bound approach is a classical method when dealing with hard species tree inference problems, as it implicitly explores the space of species trees and guarantees to find an optimal phylogeny for a given criterion. The optimality of such an approach requires that the objective function is nondecreasing during a downward phase of the exploration space and its effectiveness relies significantly on the required time to compute the cost measure associated to a newly visited species tree given the previous one. This method has been used [10] for evolutionary criteria such as Maximum Parsimony and Maximum Likelihood (the reader is referred to [6] for a brief overview), but it has not been considered up to now for the GTP. In the present work, we present a Branch-and-Bound algorithm that guarantees to find a species tree S with the minimum cost for a given gene tree G, and works for the three usual criteria (duplications, losses, and mutations). Our algorithm relies on a new way to explore the space of species trees that allows to update efficiently the cost (for the three considered costs) of a partial species tree and to naturally account for prior knowledge of the species phylogeny, and then process datasets that span a significant number of taxa.

The plan of the paper is as follows. In Section 2, formal notations are defined. In Section 3, we describe our Branch-and-Bound algorithm. In Section 4, we apply our algorithm on a dataset of 1111 gene families from 29 animal genomes taken from the TreeFam database [11, 17].

2 Preliminaries

Gene trees, species trees, forests. Except when indicated, any considered tree is rooted, binary, unordered and leaf-labeled. For simplicity, we consider that each leaf label is an integer. For a given tree T, let $V(T)$, $r(T)$, $L(T)$, $\Lambda(T)$ and $I(T)$ respectively denote its vertex set, its root, its leaf set, its label set (the set of integers that appear at its leaves), and its internal vertex set (that is $V(T) \setminus L(T)$). The depth of a vertex is the length of the unique path to $r(T)$ and the height of T, denoted $h(T)$, is its maximal depth. We will adopt the convention that the root is at the top of the tree and the leaves at the bottom.

Given two vertices u and v of T, $u \leq_T v$ (resp. $u <_T v$) if and only if v is on the unique path from u to $r(T)$ (resp. and $u \neq v$); in such a case, u is said to be a *(resp. strict) descendant* of v. For a vertex u of T, we denote by u_1 and u_2 its children (when $u \notin L(T)$), by $p(u)$ its parent, by $s(u)$ its sibling (when $u \neq r(T)$), and by T_u the subtree of T rooted at u. It is important to point out that because T is an unordered tree, the children u_1 and u_2 of an internal vertex u of T are interchangeable, that is u_1 may arbitrarily be selected as the unique children of u that respects a given constraint. The distance between two vertices u and v of a tree T, where $u <_T v$, is denoted $d_T(u, v)$ and is the number of vertices on the path from u to v in T, excluding u and v.

A *forest* $\mathcal{T}$ is a set of trees. The notations $V(\mathcal{T})$, $r(\mathcal{T})$, $L(\mathcal{T})$, $\Lambda(\mathcal{T})$ and $I(\mathcal{T})$ have the same definitions as for single trees, except that $r(\mathcal{T})$ is the set of roots of all the trees in $\mathcal{T}$. A forest is said to be ordered if there is a total order on the trees it contains. Given two trees S_1 and S_2 of a forest, we denote by $(S_1 + S_2)$ the trees obtained by joining S_1 and S_2 under a common (binary) root x (i.e. creating two edges from x to the roots of S_1 and S_2).

A *species tree* S is a tree such that each element of $\Lambda(S)$ represents an extant species and labels exactly one leaf of S (there is a bijection between $L(S)$ and $\Lambda(S)$). A *species forest* $\mathcal{F}$ is simply a

set of trees with disjoint label sets. A *gene tree* G is a tree such that $\Lambda(G) \subseteq \Lambda(S)$ (each leaf of G represents an extant gene that belongs to a species of $\Lambda(S)$).

Reconciliation between a gene tree and a species tree. A reconciliation between a gene tree G and a species tree S maps each internal vertex of G onto a vertex of S and induces an evolutionary history in term of gene duplications and losses. The Lowest Common Ancestor mapping (LCA-mapping), that maps a gene u of G onto the most recent species of S that is ancestor of all genomes that contain a gene descendant of u, is the most widely used mapping. It depicts a parsimonious evolutionary process for each of the three usual combinatorial criteria, which is also the unique parsimonious scenario for the number of losses and the number of mutations [5], while there can be several parsimonious reconciliations for the number of duplications.

Definitions 2 to 4 below define how to read the different costs associated to the reconciliation between a given gene tree G and a given species tree S.

Definition 1. The LCA-mapping between a gene tree G and a species tree S, denoted $M_S : V(G) \to V(S)$, is defined as follows: given a vertex u of G, $M_S(u)$ is the unique vertex x of S such that $\Lambda(G_u) \subseteq \Lambda(S_x)$ and either x is a leaf of S, or $\Lambda(G_u) \not\subseteq \Lambda(S_{x_1})$ and $\Lambda(G_u) \not\subseteq \Lambda(S_{x_2})$. $\diamond$

Definition 2. An internal vertex $u \in I(G)$ is a duplication if $M_S(u) = M_S(u_1)$ and/or $M_S(u) = M_S(u_2)$. The duplication cost of the reconciliation between G and S is $d(G,S) = \sum_{u \in I(G)} d(u,S)$, where $d(u,S)$ has value 1 if and only if $u \in I(G)$ is a duplication and 0 otherwise. $\diamond$

Definition 3. The loss cost of the reconciliation between G and S is $l(G,S) = \sum_{u \in I(G)} l(u,S)$, where $l(u,S)$ is defined as follows

$$l(u,S) = \begin{cases} 0 & \text{(1) if } M_S(u) = M_S(u_1) = M_S(u_2); \\ d_S(M_S(u_1), M_S(u)) + 1 & \text{(2) if } M_S(u_1) \neq M_S(u) \text{ and } M_S(u_2) = M_S(u); \\ d_S(M_S(u_1), M_S(u)) + d_S(M_S(u_2), M_S(u)) & \text{(3) if } M_S(u_1) \neq M_S(u) \text{ and } M_S(u_2) \neq M_S(u). \end{cases}$$

$\diamond$

Definition 4. The mutation cost of the reconciliation between G and S is $m(G,S) = l(G,S) + d(G,S)$. $\diamond$

Note that some internal vertices of G correspond to duplications for any given species tree S. Such vertices, that are called *apparent duplications* (also sometimes *forced duplications*) are defined as the vertices u such that $\Lambda(G_{u_1}) \cap \Lambda(G_{u_2}) \neq \emptyset$. Internal vertices u such that $|\Lambda(G_u)| = 1$ are obviously apparent duplications, and then, from now, without loss of generality, we consider that such vertices are replaced by a single leaf, which implies that there is no extant gene u (leaf) of G that has the same label as its sibling $s(u)$.

The LCA-mapping between a gene tree G and a species forest $\mathcal{F}$, denoted $M_{\mathcal{F}} : V(G) \to V(\mathcal{F})$, is defined similarly to the case of a single species tree (see Definition 1). For a given vertex $u \in V(G)$, $M_{\mathcal{F}}(u) = M_S(u)$, if there is a species tree S of $\mathcal{F}$ such that $M_S(u)$ is defined. Otherwise, $M_{\mathcal{F}}(u)$ is said to be undefined (which is denoted $M_{\mathcal{F}}(u) = \varnothing$ from now).

The goal of the current work is the design of an exact method to solve the following optimization problem, given a cost measure c (either d, l, or m) for the reconciliation between a gene tree and a species tree.

Minimum C Species Tree Problem
Input. A gene tree forest $\mathcal{G} = \{G_1, \ldots, G_k\}$
Output. A species tree S such that $\sum_{i=1}^{k} c(G_i, S)$ is minimized.

3 A Branch-and-Bound algorithm for the GTP

We now describe our Branch-and-Bound algorithm. We assume that there are n taxa, denoted by $\{1, 2, \ldots, n\}$, and denote by $\mathcal{K}^n$ the set of all possible species trees on these n taxa. Without loss of generality, we describe our algorithm for a single gene tree G.

The algorithm is based on the exploration of a rooted tree denoted $\mathcal{T}^n$, where each vertex corresponds to a forest of species trees, and such that each internal forest corresponds to an incomplete species tree for n species, and each leaf forest to a complete species tree S of $\mathcal{K}^n$. The Branch-and-Bound explores this tree and each time it visits a forest denoted $\mathcal{F}$, it computes a lower bound on the cost $c(G, S)$ (where $c = l$ or $c = d$) of any species tree $S \in \mathcal{K}^n$ located in $\mathcal{T}^n_{\mathcal{F}}$, that is the subtree of $\mathcal{T}^n$ rooted at $\mathcal{F}$. To ensure the optimality of this approach, such a lower bound has to respect the following definition.

Definition 5. Let $\pi : \mathcal{K}^n \to \mathbb{N}$ be an objective function that we seek to minimize. A function $\omega : V(\mathcal{T}^n) \to \mathbb{N}$ is a Consistent Lower Bound (CLB) for π if and only if (1) it is non-decreasing along the path that starts at $r(\mathcal{T}^n)$ and ends at any leaf $\{S\}$ of $\mathcal{T}^n$ and (2) $\omega(\{S\}) = \pi(S)$. $\diamond$

Given a CLB, denoted $c(G, \mathcal{F})$, for the considered cost $c(G, S)$, then $\mathcal{T}^n_{\mathcal{F}}$ is explored if and only if $c(G, \mathcal{F}) < c(G, S_{min})$, where $S_{min} \in \mathcal{K}^n$ is the best solution found since the beginning of the exploration of $\mathcal{T}^n$ and is updated when a species tree with a lower cost is found. Such a Branch-and-Bound guarantees to find an optimal species tree S_{min} such that $c(G, S_{min})$ is minimum.

The plan of this section is as follows: first, we formally define the space tree $\mathcal{T}^n$ and give important combinatorial properties that are central in the design of the Branch-and-Bound; second, we describe an algorithm that updates the LCA-mapping $M_{\mathcal{F}} : V(G) \to V(\mathcal{F})$ for the current visited forest $\mathcal{F}$ given the mapping of the previously visited forest; third, we define the two CLBs for the costs $d(G, S)$ and $l(G, S)$; and finally, we explain how $\mathcal{T}^n$ can easily be adapted to account for prior knowledge on the species phylogeny.

Combinatorial structure of $\mathcal{T}^n$ (i.e. the species trees space exploration tree). The main structural feature of $\mathcal{T}^n$ is that a child $\mathcal{F}'$ of an internal forest $\mathcal{F}$ is defined by joining two of its trees under a (new) vertex (thus forming a new clade). This architecture is different from the classical one used in a Branch-and-Bound approach for phylogenetic inference, where the exploration starts with a tree with two leaves and one internal node, an then iteratively add a leaf and an edge until a complete tree is obtained. The advantage of the architecture of $\mathcal{T}^n$ over the classical one is that it is more adapted to efficiently compute the LCA-mapping during its traversal, which is essential to rapidly explore the space and solve our problem. Definition 7 formally describes the architecture of $\mathcal{T}^n$, illustrated by Figure 1, and Property 1 shows that it is an appropriate structure for the exploration of $\mathcal{K}^n$. Below, given a species tree S over $\{1, \ldots, n\}$, $min(\Lambda(S))$ denotes the minimum label of the leaves of S. We also define an order on the trees of a forest as follows.

Definition 6. Given two trees S and S' of a forest $\mathcal{F}$, $S \prec_{\mathcal{F}} S'$ (resp. $\preceq_{\mathcal{F}}$) if and only if $min(\Lambda(S)) < min(\Lambda(S'))$ (resp. $\leq$). $\diamond$

Definition 7. $\mathcal{T}^n$ is an ordered and rooted tree where each vertex is an ordered species forest $\mathcal{F}$ on $\{1, \ldots, n\}$, with a distinguished tree called the *branching tree* $\beta(\mathcal{F})$. The branching structure of $\mathcal{T}^n$ is defined below.

1. The root forest $r(\mathcal{T}^n)$ is the forest composed of n trees $\{S_1, \ldots, S_n\}$, where S_i is the tree reduced to a single vertex labeled i, and its branching tree is the tree S_1.

2. Each leaf of $\mathcal{T}^n$ is a forest $\mathcal{F}$ containing a single tree that is a species tree from $\mathcal{K}^n$.
3. A forest $\mathcal{F}_x$ is a child of an internal forest $\mathcal{F}$ if and only if there exists two trees S_{x_1} and S_{x_2} in $\mathcal{F}$, with $S_{x_1} \prec_{\mathcal{F}} S_{x_2}$, such that
   (a) $\mathcal{F}_x = \mathcal{F} - \{S_{x_1}, S_{x_2}\} \cup \{S_x\}$, where $S_x = (S_{x_1} + S_{x_2})$ is the branching tree of $\mathcal{F}_x$,
   (b) and either $S_{x_2} = \beta(\mathcal{F})$ or $\beta(\mathcal{F}) \preceq_{\mathcal{F}} S_{x_1}$.

Finally, the children of an internal vertex (i.e. forest) $\mathcal{F}$ of $\mathcal{T}^n$ are totally ordered as follows: if $\mathcal{F}_x$ and $\mathcal{F}_y$ are two children of $\mathcal{F}$, where the corresponding branching trees are respectively $S_x = (S_{x_1} + S_{x_2})$ and $S_y = (S_{y_1} + S_{y_2})$, then, $\mathcal{F}_x$ precedes $\mathcal{F}_y$ if and only if either i) $S_{x_1} \prec_{\mathcal{F}} S_{y_1}$ or ii) $S_{x_1} = S_{y_1}$ and $S_{x_2} \prec_{\mathcal{F}} S_{y_2}$. $\diamond$

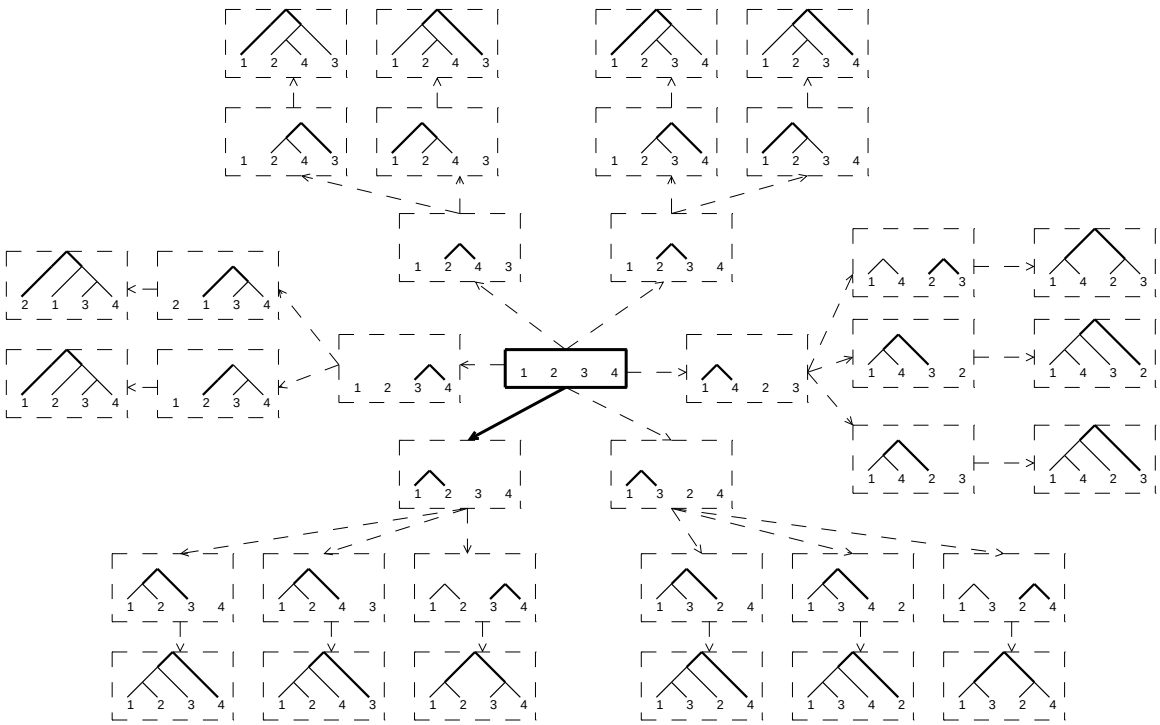

Fig. 1. For $n = 4$, representation of the space tree $\mathcal{T}^n$, where each square represents a forest, an arrow indicates one of its child, the root $r(\mathcal{T}^n)$ is the darkest square and its first child (according to the total order) is represented by the darkest arrow. The children of a forest are ordered according to the anti-clockwise direction. The darkest edges of a forest $\mathcal{F}_x$ indicate the two subtrees S_{x_1} and S_{x_2} of $\mathcal{F}$ that are joined together to form the new tree S_x of $\mathcal{F}_x$.

Property 1 below follows immediately from the structure of $\mathcal{T}^n$ described in Definition 7, and in particular from the order on the trees in a forest, that ensures that no two different paths from the root can lead to the same species forest.

Property 1. The tree $\mathcal{T}^n$ is such that (1) there are no two nodes that represent the same species forest; (2) $L(\mathcal{T}^n) = \mathcal{K}^n$; (3) its height is $\Theta(n)$; and (4) the number of children of each internal vertex is bounded by $O(n^2)$.

The general principle of our Branch-and-Bound algorithm is to visit $\mathcal{T}^n$ starting at its root, and then to recursively visit the children of the starting vertex forest $\mathcal{F}$ according to the order described

in Definition 7. We now explain how a subtree of $\mathcal{T}^n$ can be explored in time linear in the number of species forests it contains. There are two key points:

- To visit the children of an internal vertex $\mathcal{F}$ according to the order defined in Definition 7, it is sufficient that the trees of $\mathcal{F}$ are ordered according to $\prec_{\mathcal{F}}$.
- To maintain the order $\prec_{\mathcal{F}}$ in constant time when visiting a child forest $\mathcal{F}_x$ (of $\mathcal{F}$) with $S_x = (S_{x_1} + S_{x_2})$ as its branching tree, S_{x_2} is removed from the ordered forest and S_x replaces S_{x_1} (as $min(\Lambda(S_x)) = min(\Lambda(S_{x_1}))$). And then, when the traversal goes back to $\mathcal{F}$ after visiting the subtree $\mathcal{T}^n_{\mathcal{F}_x}$, the ordered forest $\mathcal{F}$ can be retrieved (in constant time) assuming that both the position of S_{x_2} in $\mathcal{F}$ and $S_x = (S_{x_1} + S_{x_2})$ (the used branching tree) were previously saved, which can be implemented easily using lists and pointers.

Together that the height of $\mathcal{T}^n$ is in $\Theta(n)$, this proves the following result.

Proposition 1. *The complete exploration of a subtree $\mathcal{T}$ of $\mathcal{T}^n$ can be implemented to run in time $\Theta(|V(\mathcal{T})|)$ and space $\Theta(n)$.*

We finally introduce some combinatorial properties on the architecture of $\mathcal{T}^n$ that will be used to define a CLB for the cost $l(G, S)$. According to Definition 3, the cost $l(u, S)$ induced by an internal vertex u of G depends on the distance in S between $M_S(u)$ and $M_S(u_1)$ (resp. $M_S(u_2)$). Hence, the main idea behind a CLB for $l(G, S)$ resides on the definition of a CLB for the distance $d_S(M_S(u_1), M_S(u))$ (resp. $d_S(M_S(u_2), M_S(u))$). Formally, considering a forest $\mathcal{F}$ of $\mathcal{T}^n$, a non-root vertex u of G, and any species tree $S \in \mathcal{K}^n$ that is located at a leaf of $\mathcal{T}^n_{\mathcal{F}}$, the question is as follows: how can a CLB for $d_S(M_S(u), M_S(p(u)))$ be efficiently computed during the traversal of $\mathcal{T}^n$ along the path that connects $r(\mathcal{T}^n)$ and $\mathcal{F}$? To define such a CLB, we introduce *incremental forests*.

Definition 8. Let $\mathcal{F}$ be a forest of $\mathcal{T}^n$ and $\mathcal{F}_x$ one of its children, whose branching tree is $S_x = (S_{x_1} + S_{x_2})$. Given a non-root vertex u of G, if the mapping $M_{\mathcal{F}}(u)$ is defined in either S_{x_1} or S_{x_2} and $M_{\mathcal{F}_x}(p(u))$ is not defined, then $\mathcal{F}_x$ is said to be an incremental forest for u. $\diamond$

It is easy to see that each incremental forest for u located between $r(\mathcal{T}^n)$ and $\mathcal{F}$ (including $\mathcal{F}$) corresponds to an increment of one on $d_S(M_S(u), M_S(p(u)))$. If $d_{\mathcal{F}}(u)$ denotes the number of such incremental forests, then the property below immediately follows from the usual LCA-mapping (between G and S) and Definition 7.

Property 2. Given a leaf forest $\{S\}$ of $\mathcal{T}^n$ and a vertex u of $V(G) \setminus \{r(G)\}$, $d_S(M_S(u), M_S(p(u))) = d_{\{S\}}(u)$, and then $d_{\mathcal{F}}(u)$ is a CLB for $d_S(M_S(u), M_S(p(u)))$.

Updating the LCA-mapping. Let $\mathcal{F}$ be an internal forest of $\mathcal{T}^n$, and $\mathcal{F}_x$ be one of its child forest and the next one to be visited during the traversal of the space tree. The main issue here is to detect as efficiently as possible the vertices u of G for which the LCA-mapping was undefined in $\mathcal{F}$ but is now defined in $\mathcal{F}_x$. Definition 9 below describes the smallest forest of subtrees of G that contains all these vertices, Figure 2 depicts a simple example of this architecture, and then we explain how it can be explored in linear time.

Definition 9. Let $\mathcal{F}$ be an internal forest of $\mathcal{T}^n$ and $\mathcal{F}_x$ be one of its child, where $S_x = (S_{x_1} + S_{x_2})$ is the branching tree (see Definition 7). $\mathcal{G}_x$ denotes the forest of subtrees of G such that its root set is $r(\mathcal{G}_x) = \mathcal{M}_{\mathcal{S}_x} = \{u \in V(G) \setminus \{r(G)\} : M_{S_x}(u) \neq \varnothing \text{ and } M_{S_x}(s(u)) = \varnothing\}$ and its leaf set is $L(\mathcal{G}_x) = \mathcal{M}_{\mathcal{S}_{x_1}} \cup \mathcal{M}_{\mathcal{S}_{x_2}}$. $\diamond$

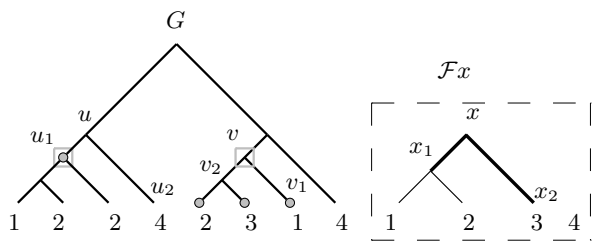

Fig. 2. Representation of the forest $\mathcal{G}_x$, given a gene tree G and a forest $\mathcal{F}x$ of $\mathcal{T}^n$, where $n = 4$. The vertices of G that are in $r(\mathcal{G}_x)$ (resp. $L(\mathcal{G}_x)$) are represented by a grey square (resp. circle). The two trees S_{x_1} and S_{x_2} of $\mathcal{F}$ used to define the branching tree S_x of $\mathcal{F}_x$ are indicated by larger edges. Because u_1 is mapped in S_x and u_2 is not, u_1 is both a leaf and a root of $\mathcal{G}_x$.

The main problem for the exploration of $\mathcal{G}_x$ is that it is not explicitly defined when the forest $\mathcal{F}_x$ is visited during the traversal of $\mathcal{T}^n$. However, as $\mathcal{F}$ is the previous forest and the parent of $\mathcal{F}_x$, we can assume that both sets $\mathcal{M}_{\mathcal{S}_{x_1}}$ and $\mathcal{M}_{\mathcal{S}_{x_2}}$ are computed, together with the LCA-mappings of their vertices, and that the leaf set $L(\mathcal{G}_x)$ is available (see Definition 9). The traversal of the whole forest $\mathcal{G}_x$ is done by a bottom-up approach in such a way that each vertex of $\mathcal{G}_x$ located at a given depth in G is visited before any vertex located at a lower depth. Such a traversal ensures that when a vertex u of $\mathcal{G}_x$ is visited, its mapping $M_{\mathcal{F}_x}(u)$ is defined (in S_x), and if the mapping $M_{\mathcal{F}_x}(s(u))$ of its sibling is not defined, then $s(u)$ is not in $\mathcal{G}_x$ and u is a root of this forest. This observation is formally stated in the Property 3 below, which also describes the calculations that are required during the traversal of $\mathcal{G}_x$ in such a way that the root set $r(\mathcal{G}_x)$ and the LCA-mappings of its vertices are computed.

Property 3. Let u be the current visited vertex during the bottom-up traversal of $\mathcal{G}_x$ and assume that its mapping $M_{\mathcal{F}_x}(u)$ is defined. If $M_{\mathcal{F}_x}(s(u))$ is defined and is in S_x, then both $s(u)$ and $p(u)$ are in $\mathcal{G}_x$ and $M_{\mathcal{F}_x}(p(u)) = x$. Otherwise, neither $s(u)$ nor $p(u)$ is in $\mathcal{G}_x$ and $u \in \mathcal{M}_{\mathcal{S}_x}$ $(= r(\mathcal{G}_x))$.

Proposition 2. *Let $\mathcal{F}_x$ be a child of an internal forest $\mathcal{F}$ of $\mathcal{T}^n$, with $S_x = (S_{x_1} + S_{x_2})$ as the branching tree, and suppose that $\mathcal{M}_{\mathcal{S}_{x_1}}$ and $\mathcal{M}_{\mathcal{S}_{x_2}}$ are given with their vertices ordered in decreasing order of their depth in G. The bottom-up traversal of $\mathcal{G}_x$, which computes the set $\mathcal{M}_{\mathcal{S}_x}$ (with the same order described above) and the LCA-mapping $M_{\mathcal{F}}(u)$ for each vertex $u \in V(\mathcal{G}_x)$, can be implemented to run in time $\Theta(|V(\mathcal{G}_x)|)$ and space $O(|V(\mathcal{G}_x)|) + \Theta(|V(\mathcal{F}_x)|)$.*

Proof. The bottom-up traversal of $\mathcal{G}_x$ is briefly described below.

1. Let Q be a queue (with the usual order) of vertices of G, initialized with $\mathcal{M}_{\mathcal{S}_{x_1}}$ and $\mathcal{M}_{\mathcal{S}_{x_2}}$.
2. A first loop on the vertices of Q is performed.
   (a) Let Q' be an empty queue.
   (b) A second loop over all the vertices u located at the front of Q and at the same depth in G, where the calculations described in Property 3 are done and $p(u)$ is inserted in Q' if and only if $p(u) \in V(\mathcal{G}_x)$.
   (c) All the vertices of Q' are moved at the front of Q.

As line 1 is done in $\Theta(|L(\mathcal{G}_x|))$ time, and all operations of Property 3, as well as line 2c, are done in constant time, the expected time complexity is immediate. The expected space complexity follows immediately from the traversal of $\mathcal{G}_x$ described above.

Definition and computation of the two CLBs. For both duplication and loss criteria, we formally define a cost for a forest $\mathcal{F}$ of $\mathcal{T}^n$, prove that it is a CLB for the considered criterion, and explain how it is computed in linear time when a child forest $\mathcal{F}_x$ of $\mathcal{F}$ is visited. Below, $S_x = (S_{x_1} + S_{x_2})$ denotes the branching tree of $\mathcal{F}_x$, $I'(G)$ denotes the subset $\{u \in V(G) \setminus L(G) : M_{\mathcal{F}}(u) \neq \varnothing\}$, and when the context is unambiguous, S refers to the species tree of $\mathcal{F}$ where the mapping $M_{\mathcal{F}}(u)$, of a vertex $u \in I'(G)$, is defined. We denote by k_G the number of apparent duplications in G, by $k_{G,\mathcal{F}}$ the number of such apparent duplications whose LCA-mapping is defined in $\mathcal{F}$, and by $d_{G,\mathcal{F}}$ the number of internal vertices of G that are duplications according to $\mathcal{F}$, that is $d_{G,\mathcal{F}} = \sum_{u \in I'(G)} d(u, S)$.

Definition 10. The duplication cost between a forest $\mathcal{F}$ of $\mathcal{T}^n$ and a gene tree G is denoted $d(G, \mathcal{F})$ and is defined as follows: $d(G, \mathcal{F}) = d_{G,\mathcal{F}} + k_G - k_{G,\mathcal{F}}$. $\diamond$

Together with Definition 2 and the fact that each apparent duplication $u \in I(G)$ is such that $d(u, S) = 1$ for any species tree $S \in \mathcal{K}^n$, $d(G, \mathcal{F})$ is obviously a CLB for $d(G, S)$. The inconvenient of this CLB, when considering only the vertices of G that are not apparent duplication, is that it requires the LCA-mapping of such a vertex to determine if it is a duplicated gene for any species tree located below the considered forest $\mathcal{F}$. Moreover, it is important to assess the efficiency of this CLB against the one that does not take advantage of the constant cost induced by the apparent duplications (such a CLB is equal to $d_{G,\mathcal{F}}$). Recall that all k_G apparent duplications are considered in the cost $d(G, \mathcal{F})$ of any forest $\mathcal{F}$ of $\mathcal{T}^n$, which means that the CLB described in Definition 10 can be reduced solely to the non-apparent duplications. Formally, if $\mathcal{F}$ is the current visited forest and S^* is the best species tree found since the beginning of the traversal of $\mathcal{T}^n$, then $\mathcal{F}$ is pruned if and only if $d_{G,\mathcal{F}} - k_{G,\mathcal{F}} \geq d(G, S^*) - k_G$. In other words, the advantage given by the k_G apparent duplications present in G can not be evaluated by how large k_G is, as the efficiency of the CLB $d(G, \mathcal{F})$ to cut the subtree $\mathcal{T}^n_{\mathcal{F}}$ depends on the number of non-apparent duplications induced by $\mathcal{F}$ (i.e. $d_{G,\mathcal{F}} - k_{G,\mathcal{F}}$) and S^* (i.e. $d(G, S^*) - k_G$).

The corollary below explains how the CLB $d(G, \mathcal{F})$ can be computed with the same complexities as for the traversal of $\mathcal{G}_x$ (see Proposition 2).

Corollary 1. *If the duplication cost $d(G, \mathcal{F})$ is given, then $d(G, \mathcal{F}_x)$ can be computed in time $\Theta(|V(\mathcal{G}_x)|)$ and space $O(|V(\mathcal{G}_x)|) + \Theta(|V(\mathcal{F}_x)|)$.*

Proof. Let $d(\mathcal{G}_x)$ denotes the number of vertex u of G that is a duplication according to S_x and such that its LCA-mapping is (resp. not) defined in $\mathcal{F}_x$ (resp. $\mathcal{F}$), that is $M_{\mathcal{F}_x}(u) = x$, $M_{\mathcal{F}}(u) = \varnothing$ and $d(u, S_x) = 1$. Hence, according to Definition 9, u is in $\mathcal{G}_x$. It is then immediate that $d(G, \mathcal{F}_x) = d(G, \mathcal{F}) + d(\mathcal{G}_x)$, where $d(\mathcal{G}_x)$ is computed by the traversal of $\mathcal{G}_x$, and the expected complexities follow immediately from Proposition 2. To handle apparent duplications, detecting them can be done during a preprocessing phase, which implies that updating $k_{G,\mathcal{F}}$ when visiting a new forest can be done in time linear in the number of apparent duplications whose mapping is defined.

Definition 11. The loss cost between a forest $\mathcal{F}$ of $\mathcal{T}^n$ and a gene tree G is denoted $l(G, \mathcal{F})$ and is defined as follows: $l(G, \mathcal{F}) = \sum_{u \in I'(G)} l(u, S) + \sum_{u \in I(G) \setminus I'(G)} (d_{\mathcal{F}}(u_1) + d_{\mathcal{F}}(u_2))$. $\diamond$

From Definitions 3 and 5 and Property 2, Corollary 2 below is immediate.

Corollary 2. *$l(G, \mathcal{F})$ is a CLB for $l(G, S)$.*

Corollary 3. *If the loss cost $l(G, \mathcal{F})$ is given, then $l(G, \mathcal{F}_x)$ can be computed in time $\Theta(|V(\mathcal{G}_x)|)$ and space $O(|V(\mathcal{G}_x)|) + \Theta(|V(\mathcal{F}_x)|)$.*

Proof. Let $l(\mathcal{G}_x)$ denotes the number of vertex u of $\mathcal{G}_x$ such that either $l(u, S_x)$ corresponds to the second case of Definition 3 or $\mathcal{F}_x$ is an incremental forest for u (see Definition 8). It is then immediate that $l(G, \mathcal{F}_x) = l(G, \mathcal{F}) + l(\mathcal{G}_x)$, where $l(\mathcal{G}_x)$ is computed by the traversal of $\mathcal{G}_x$. Because both cases above can be verified in constant time ($\mathcal{F}_x$ is an incremental forest for u if and only if u is both a leaf and a root of $\mathcal{G}_x$), the expected complexities follow immediately from Proposition 2.

Proposition 1, together with Corollaries 1, 2 and 3, gives the fundamental properties of the complexity of our Branch-and-Bound algorithm: (1) the exploration of the visited species forests requires a time linear in the number of these forests, (2) visiting a new species forest while updating the appropriate CLB requires a time linear in the number of vertices of gene forest whose LCA-mapping is updated and (3) the total space complexity is linear in the number of considered taxa. We do not have theoretical properties of the two CLBs we introduced, and we will assess how efficient they are to cut large subspaces of $\mathcal{T}^n$ experimentally in Section 4.

Accounting for prior knowledge on the considered species tree. In the context of some prior knowledge on the species tree for the considered n genomes, let S be a multifurcating (non-binary) species tree that describes such prior and $\mathcal{K}^n(S)$ be the subset of $\mathcal{K}^n$ that contains each (binary) species tree that is consistent with S. Such prior information can for example consists in well defined clades of taxa. The question that we address here is how to define an architecture that allows to exhaustively explore the subset $\mathcal{K}^n(S)$? Note that it can lead to a species tree that is not globally optimal, but is optimal among the species tree of $\mathcal{K}^n(S)$. The solution resides in the use of Definition 7 to exhaustively enumerate all the binary species trees for the m children of an internal vertex x of S that is not resolved. Formally, let $\{t_1, t_2, \ldots, t_m\}$ be the set of the m subtrees, where $m > 2$, rooted at the children of x. The original architecture $\mathcal{T}^m$ (see Definition 7) is used to exhaustively enumerate all the binary species trees for the unresolved vertex x of S as follows: $\mathcal{T}^m$ is rooted at the forest $\{t_1, t_2, \ldots, t_m\}$, and each leaf forest is composed of a single binary tree that contains one and only one leaf labeled by the subtree t_i, for each $1 \leq i \leq m$.

By recursively applying this process on all the unresolved vertices of S, it is easy to see that the induced architecture has $\mathcal{K}^n(S)$ as its leaf set. Moreover, given that all such unresolved vertices of S are ordered according to the prefix traversal of S, the time and space complexities for the exploration of this adapted architecture are the same as in Proposition 1.

4 Experimental results

We considered 1111 gene trees from the TreeFam database [11, 17], more specifically the ones of the TreeFam-B families that have been manually corrected by experts and contain gene families from 29 eukaryotic genomes. Table 3 (Appendix) lists the considered species, together with their abbreviations, and Figure 5 (Appendix) describes the size distribution of the gene trees. The corresponding reference species tree, denoted by S_0, is depicted in Appendix (Figure 4) and corresponds to the NCBI taxonomy tree [19], except that three nodes of the tree were considered as multifurcations due to different phylogenetic hypothesis regarding the corresponding clades (see [11, 17]).

Study of a subset of 8 *species.* First, to gain some insight on the whole space of species trees for a given dataset, we selected $n = 8$ species from the 29 considered genomes (see Figure 4), removed from the gene trees all genes from other species, and performed an exhaustive exploration of the 135135 species trees. The aim is to study the shape of the space $\mathcal{K}^n$ according to the three combinatorial criteria we consider in order to evaluate their performance for phylogenetic inference. For a species tree S of $\mathcal{K}^n$, $c(G, S)$ denotes the considered cost (either l, d, or m), S_{min} (resp. S_{max})

a tree of $\mathcal{K}^n$ that minimizes (resp. maximizes) this cost and $c_{norm}(G,S)$ the normalized cost over the range of possible values and defined as follows: $c_{norm}(G,S) = (c(G,S) - c(G,S_{min}))/(c(G,S_{max}) - c(G,S_{min}))$. The similarity between S and the species tree S_0 is evaluated by the classical Robinson and Foulds distance between phylogenetic trees [15, 16], denoted $RF(S_0,S)$. For the three costs, Figures 3 (left) below depicts the distribution of each tree $S \in \mathcal{K}^n$ according to the normalized cost $l_{norm}(G,S)$.

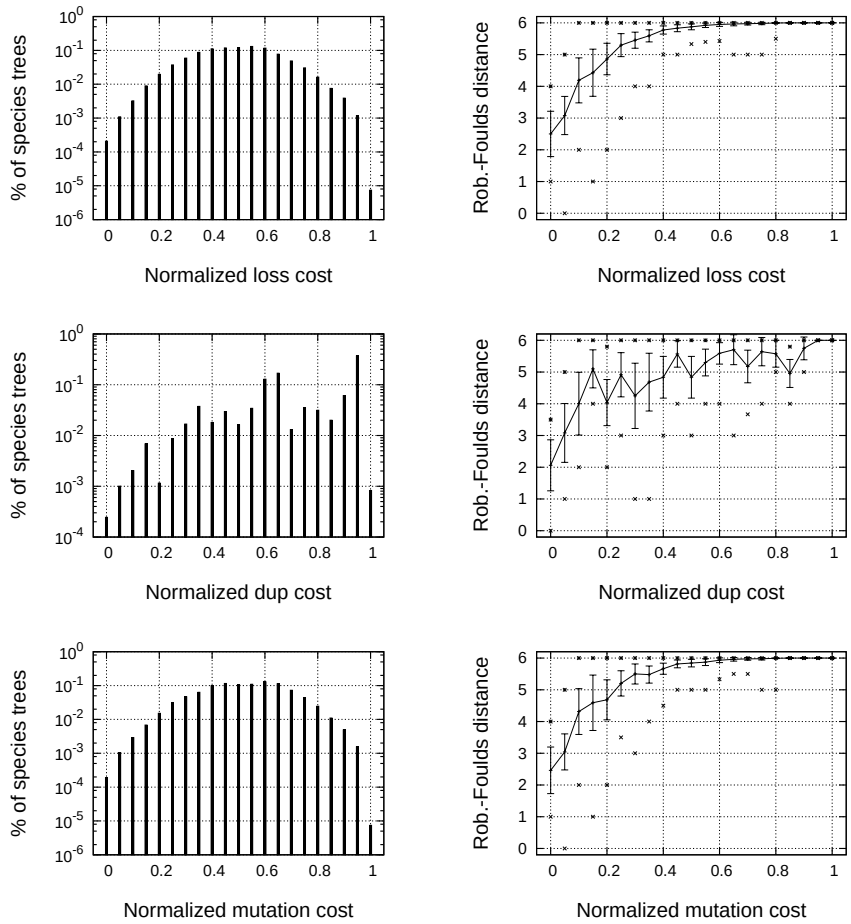

Fig. 3. (Left) The percentage of species tree S of $\mathcal{K}^n$ (y axis) such that $0.05\ i \leq c_{norm}(G,S) < 0.05(i+1)$ (x axis) and (Right) the average, standard deviation, and minimum and maximum values of the Robinson and Foulds distance with S_0 for each such tree. Top: loss cost. Middle: duplication cost. Bottom: mutation cost.

It appears clearly from Table 1 that, on this dataset, the duplication cost seems to be slightly better than the two other criteria, although in terms of normalized cost, the difference is relatively marginal. We can also observe that for the loss and mutation cost, the loss cost distribution is similar to a normal and almost symmetrical distribution with a mean located around 0.5, while the distribution for the duplication cost is less smooth. Over each species tree S of $\mathcal{K}^n$ with the same normalized cost $c_{norm}(G,S)$, we also computed the average distance $RF(S_0,S)$, and according to Figure 3 (right), the loss cost is clearly correlated with the R.F. distance to the tree S_0, which is not as clear with the duplication cost. Also note that, due to the fact that losses are more numerous than duplications, the properties of the mutation cost are very similar to the loss cost. Finally, we

	$c(G, S_{min})$	$c(G, S_{max})$	$c(G, S_0)$	$RF(S_0, S_{min})$
Loss	3577	35072	5226 (0.05)	3
Dup.	2229	6313	2425 (0.04)	1
Mut.	5812	41355	7651 (0.05)	3

Table 1. Minimum (col. 1) and maximum (col 2.) costs for the loss, duplication, and mutation criteria. Col. 3: costs of S_0, both absolute and normalized. Col. 4: R.F. distance between the optimal solution and S_0.

can notice that S_0 is close to the optimal species tree, both in terms of duplication and/or loss events, and in the RF distance.

Study of the whole 29 taxa dataset. Next, we attacked the problem of computing a parsimonious species tree for the 29 considered genomes. There are about 10^{36} possible species trees, and the Branch-and-Bound starting from the root of $\mathcal{T}^n$ was not completed after a few days of computation. There are two reasons for this problem: first, during the traversal of $\mathcal{T}^n$, the CLB of the newly visited forest is computed in linear time; second, the subtree of $\mathcal{T}^n$ induced by the pruned forests is not small enough for an exhaustive exploration.

We then decided to reduce the number of considered species trees by integrating prior information on the seeked species tree. For our experiments, we defined such prior information from the species tree S_0 as follows. A species subset denoted $\pi \subseteq \Lambda(S_0)$ is said to be consistent with a given gene tree G if and only if its intersection π' with $\Lambda(G)$ is consistent with each vertex $u \in V(G)$, that is either $\pi' \subseteq \Lambda(G_u)$, $\Lambda(G_u) \subseteq \pi'$, or $\Lambda(G_u) \cap \pi' = \emptyset$. Hence, the more a clade is respected among the considered gene trees, the more probable it is present in an optimal solution of the GTP problem. We found 19 clades, which can either be disjoint or included one into the other, that are consistent with a majority of the 1111 gene trees, and defined four species trees, denoted S_i for $i \in \{1, 2, 3, 4\}$, as follows: S_1 is consistent with all and only all the 19 clades, and for $i = 2, 3, 4$, one clade is removed from S_{i-1} to define S_i (see Appendix, Figures 6 and 7). Recall that $\mathcal{K}^n(S_i)$ denotes the subset of species trees that are consistent with a given tree S_i (see Section 3), then the five subsets of $\mathcal{K}^n$ are embedded as follows: $\mathcal{K}^n(S_i) \subset \mathcal{K}^n(S_{i+1})$, for $0 \leq i \leq 3$. Hence, $\mathcal{K}^n(S_0)$ (resp. $\mathcal{K}^n(S_4)$) is the smallest (resp. largest) set of species trees, and similarly is the induced space tree defined at the end of Section 3, which is used to solve the GTP problem on the considered reduced set of species trees. For S_1, S_2, S_3, and S_4, the number of possible species trees is respectively 127575, 893025, 9823275 and 29469825. For the three usual criteria, we applied the Branch-and-Bound to solve the GTP problem first on the smallest set $\mathcal{K}^n(S_0)$, and then the optimal solution for $\mathcal{K}^n(S_i)$, with increasing index i from 0 to 3, was then used as the first upper bound for the Branch-and-Bound applied on $\mathcal{K}^n(S_{i+1})$. For each criterion and each constrained species tree, the result are summarized in Table 2 below.

For the duplication criterion, the best solution found in $\mathcal{K}^n(S_1)$ and $\mathcal{K}^n(S_2)$ is the same and is depicted in Figure 8 (see Appendix A), and the Branch-and-Bound applied on $\mathcal{K}^n(S_3)$ (resp. $\mathcal{K}^n(S_4)$) was not completed after 4 days CPU time, and this is caused (as previously explained) by the slow rise of the CLB for the number of duplications according to the one for the losses. Moreover, among all the 20942 internal vertices, there is 3693 apparent duplications. For the loss and mutation criteria and the four sets $\mathcal{K}^n(S_i)$, the optimal solution is the same and is depicted in Figure 8. For both criteria, this means that the optimal solution for the GTP applied on the largest set $\mathcal{K}^n(S_4)$ can be found solely by applying the Branch-and-Bound on the smallest set $\mathcal{K}^n(S_1)$, although that its optimality status requires the use of $\mathcal{K}^n(S_4)$. The Robinson and Foulds distance between the two optimal solutions for loss (i.e. and mutation) (in $\mathcal{K}^n(S_4)$) and duplication (in

	Optimal cost in $\mathcal{K}^n(S_i)$		CPU time (in seconds)				
	S_0	$S_1 \ldots S_4$	S_0	S_1	S_2	S_3	S_4
Loss	22464	21257	54	3147	7897	26444	59997
Mut	27691	26328	49	3944	10697	39742	94429
Dup	5140	4941	50	7296	32117	?	?

Table 2. For each criterion and each constrained species tree S_i, the **optimal cost** for the GTP problem applied on the set of allowed solutions $\mathcal{K}^n(S_i)$ and the **CPU time** used by the Branch-and-bound. For each of the three criteria, the optimal cost in $\mathcal{K}^n(S_i)$, for $i = 1, 2, 3,$ and 4, is the same. The '?' character indicates that the Branch-and-bound process was not terminated after 4 days, where the optimal solution of $K^n(S_2)$ was the best solution found so far for both process.

$\mathcal{K}^n(S_2)$) criteria is 4, while their distance with S_0 (i.e. the reference species tree) respectively is 5 and 3.

We applied Duptree [21], which is based on a Randomized hill climbing heuristic, on our 1111 gene trees to solve the GTP problem for the duplication criterion. First, when the topological constraints of S_4 are used to reduce the search space (that is $\mathcal{K}^n(S_4)$ as for our Branch-and-Bound), duptree found the same solution as our approach applied on $\mathcal{K}^n(S_2)$ (see Table 2) within solely 2 seconds and 6 rSPRs. Second, when the program is applied without prior knowledge on the species phylogeny, the same solution is found within 3 seconds and 8 rSPRs.

5 Conclusion

In the current work, we described a Branch-and-Bound algorithm that seeks a parsimonious species tree, given a set of gene trees, according to the number of duplications and/or losses. Up to now, two (resp. one) Fixed-Parameter Tractable algorithms exists for the duplication (resp. mutation) criterion [13, 20] (resp. [13]), and there is no exact approach for the loss criterion, which appears to be relevant for phylogenetic inference according to [4]. Whereas these FPT algorithms are not suitable for phylogenetic inference problems with several genomes, we demonstrated the applicability of our Branch-and-Bound on a large dataset of 29 eukaryotic genomes to obtain the optimal species tree given prior constraints on the species phylogeny. For both loss and mutation criteria, our approach found the optimal species tree on a real and large dataset. Moreover, as Duptree does not consider the number of duplications and losses, it can not be used to validate the optimal solution (for the mutation criterion) proposed in this work.

Our results show that the species phylogeny in TreeFam is very close (but different) to the optimal species tree. This suggests that near-optimal species trees are worth to be considered when looking for a species tree from gene families, and that methods that explore the neighborhood of a given species tree (here the optimal one) are pertinent. Local-search heuristics such as the ones described in [1, 2] are natural candidates. However, it is important to recall that such approaches, in order to assess the computational quality of the produced species trees, need to be complemented by methods that compute an optimal species tree (or an optimal among a large set of considered species tree), which we described here.

A possible direction for further research is to consider multiple gene duplication episodes, during which a large portion of an organism's genome is duplicated. Given a species tree S and a set of gene trees, [3] proposed the first exact and efficient algorithm that locates in S gene duplication events in such a way that the number of multiple gene duplication episodes is minimized. However, when the species phylogeny is unknown, inferring the most parsimonious species tree according to the multiple gene duplication criterion is still an open problem.

References

1. M.S. Bansal, J.G. Burleigh, O. Eulenstein, and A. Wehe. Heuristics for the gene-duplication problem: A $\theta(n)$ speed-up for the local search. In *Research in Computational Molecular Biology, 11th Annual International Conference, RECOMB 2007, Oakland, CA, USA, April 21-25, 2007, Proceedings*, volume 4453 of *Lecture Notes in Computer Science*, pages 238–252. Springer, 2007.
2. M.S. Bansal, O. Eulenstein, and A. Wehe. The gene-duplication problem: Near-linear time algorithms for nni-based local searches. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(2):221–231, 2009.
3. Mukul S. Bansal and Oliver Eulenstein. The multiple gene duplication problem revisited. *Bioinformatics*, 24(13):i132–i138, 2008.
4. C. Chauve, J.P. Doyon, and N. ElMabrouk. Gene family evolution by duplication, speciation and loss. *Journal of Computational Biology*, 15(8):1043–1062, 2008.
5. C. Chauve and N. ElMabrouk. New perspectives on gene family evolution: Losses in reconciliation and a link with supertrees. In *Research in Computational Molecular Biology, 13th Annual International Conference, RECOMB 2009, Tucson, AZ, USA, May 18-21, 2009. Proceedings*, volume 5541 of *Lecture Notes in Computer Science*, pages 46–58. Springer, 2009.
6. J. Felsenstein. *Inferring Phylogenies*. Sinauer Associates, 2003.
7. W.M. Fitch. Homology a personal view on some of the problems. *Trends in Genetics*, 16:227–231, 2000.
8. M. Goodman, J. Czelusniak, G.W. Moore, R.A. Herrera, and G. Matsuda. Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences. *Systematic Zoology*, 28:132–163, 1979.
9. D. Graur and W.-H. L. *Fundamentals of Molecular Evolution*. Sinauer Associates, second edition edition, 1999.
10. M.D. Hendy and D. Penny. Branch and bound algorithms to determine minimal evolutionary trees. *Mathematical Biosciences*, 59, 1982.
11. H. Li and *et al.* Treefam: a curated database of phylogenetic trees of animal gene families. *Nucleic Acids Research*, 34(Database-Issue):572–580, 2006.
12. B. Ma, M. Li, and L. Zhang. From gene trees to species trees. *SIAM Journal on Computing*, 30(3):729–752, 2000.
13. Hallett M.T. and Lagergren J. New algorithms for the duplication-loss model. In *Proceedings of the Fourth Annual International Conference on Computational Molecular Biology, RECOMB 2000, April 8-11, 2000, Tokyo, Japan*, pages 138–146. ACM Press, 2000.
14. R.D.M. Page. GeneTree: comparing gene and species phylogenies using reconciled trees. *Bioinformatics*, 14(9):819–820, 1998.
15. N.D. Patengale, E.J. Gottlieb, and B.M.E. Moret. Efficiently computing the robinson-foulds metric. *Journal of Computational Biology*, 14(6):724–735, 2007.
16. D.F. Robinson and L.R. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53:131–147, 1981.
17. J. Ruan and *et al.* TreeFam: 2008 Update. *Nucleic Acids Research*, 36(Database issue):D735–D740, 2008.
18. M. Sanderson and M. Mcmahon. Inferring angiosperm phylogeny from est data with widespread gene duplication. *BMC Evolutionary Biology*, 7(Suppl 1):S3, 2007.
19. E.W. Sayers and *et al.* Database resources of the national center for biotechnology information. *Nucleic Acids Research*, 37(Database issue):D5–D15, 2009.
20. Ulrike Stege. Gene trees and species trees: The gene-duplication problem is fixed-parameter tractable. In *Proceedings of the 6th International Workshop on Algorithms and Data Structures (WADS'99*, pages 166–3, 1999.
21. A Wehe, M.S. Bansal, J.G Burleigh, and O. Eulenstein. DupTree: a program for large-scale phylogenetic analyses using gene tree parsimony. *Bioinformatics*, 24(13):1540–1541, 2008.
22. L. Zhang. On a mirkin-muchnik-smith conjecture for comparing molecular phylogenies. *Journal of Computational Biology*, 4(2):177–187, 1997.

A Appendix. Additional tables and figures

Arath	Arabidopsis thaliana	Orysa	Oryza sativa
Schpo	Schizosaccharomyces pombe	Yeast	Saccharomyces cerevisiae
Schma	Schistosoma mansoni	Caebr	Caenorhabditis briggsae
Caeel	Caenorhabditis elegans	Caere	Caenorhabditis remanei
Aedae	Aedes aegypti	Anoga	Anopheles gambiae
Drome	Drosophila melanogaster	Drops	Drosophila pseudoobscura
Cioin	Ciona intestinalis	Ciosa	Ciona savignyi
Brare	Danio rerio	Fugru	Fugu rubripes
Tetng	Tetraodon nigroviridis	Gasac	Gasterosteus aculeatus
Oryla	Oryzias latipes	Xentr	Xenopus tropicalis
Chick	Gallus gallus	Mondo	Monodelphis domestica
Bovin	Bos taurus	Canfa	Canis familiaris
Mouse	Mus musculus	Rat	Rattus norvegicus
Macmu	Macaca mulatta	Human	Homo sapiens
Pantr	Pan troglodytes		

Table 3. The list of the 29 considered species, with their names (second and fourth columns) and their abbreviations (first and third columns).

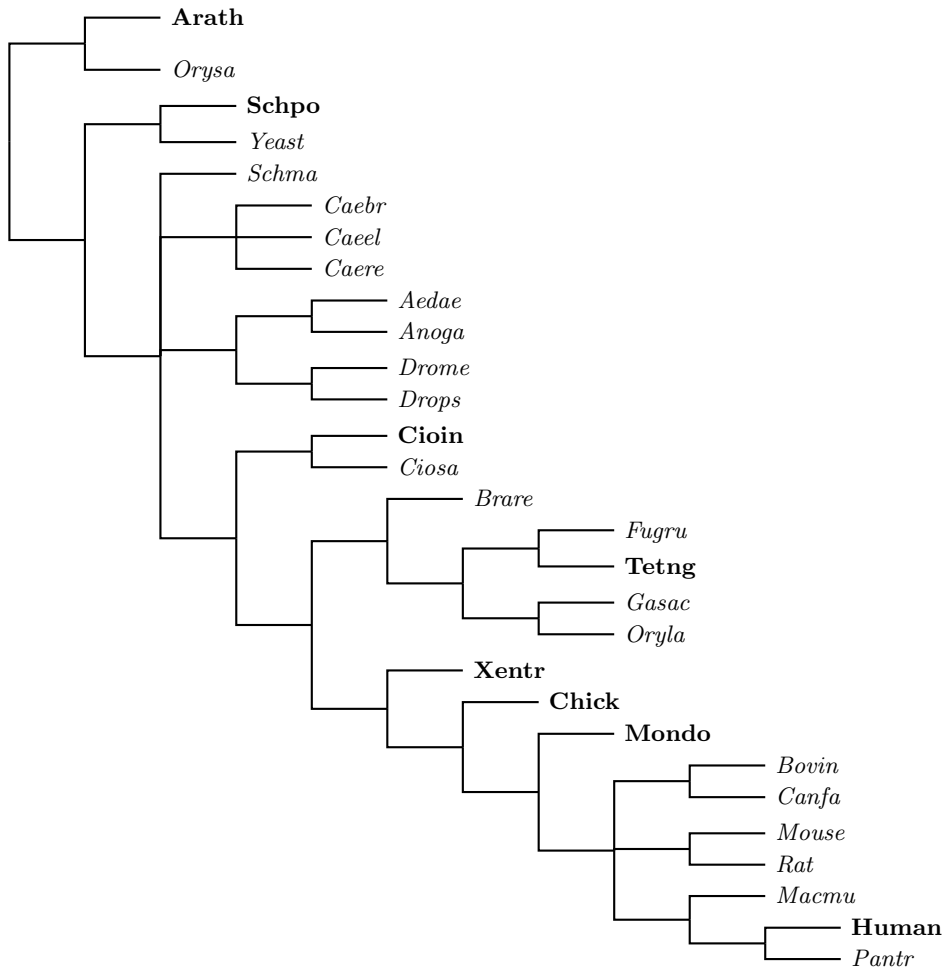

Fig. 4. The species tree for the 29 animals considered in the experiments. Species in boldface corresponds to the eight species used for an exhaustive exploration of the space $\mathcal{K}^n$, where $n = 8$ (See Section 4).

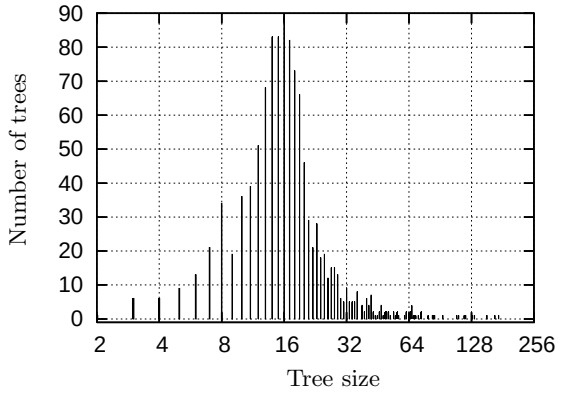

Fig. 5. Distribution of the 1111 gene trees (y axis) according to their number of leaves (x axis).

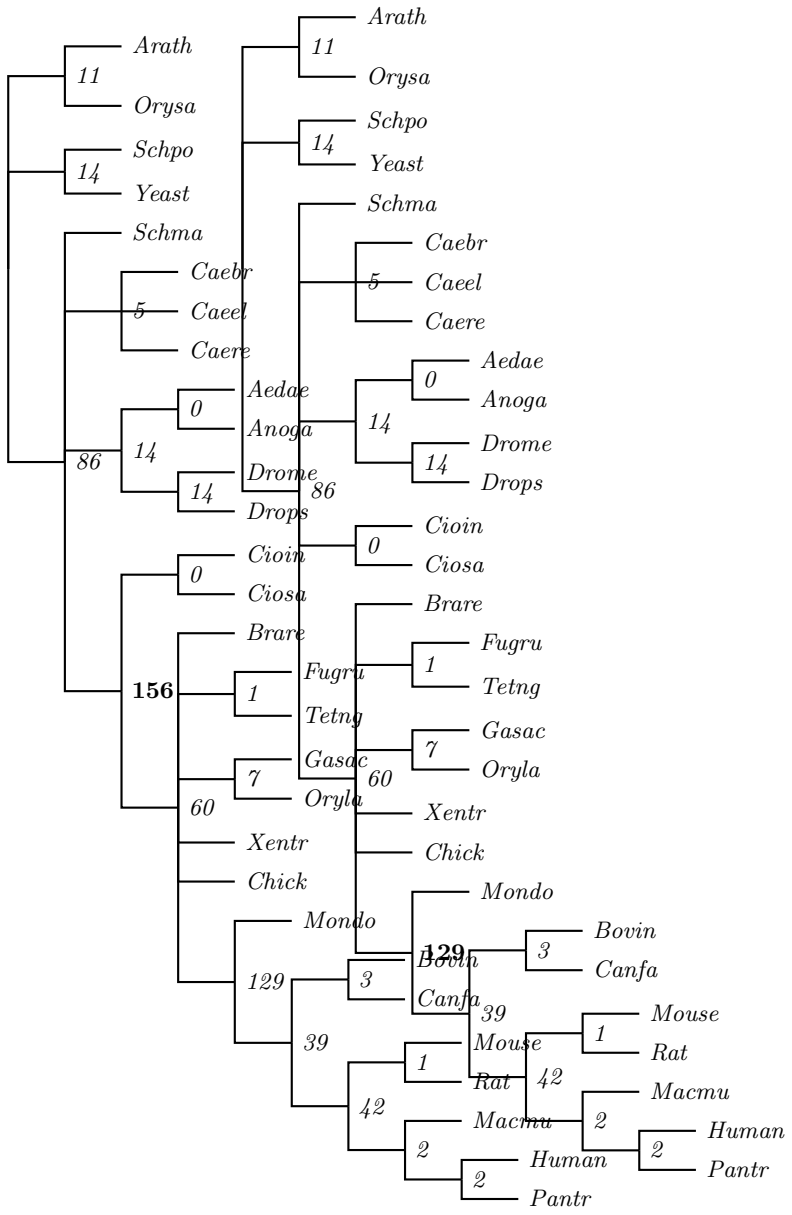

Fig. 6. Constrained species trees S_1 (**left**) and S_2 (**right**), where the integer beside each vertex indicates the number of trees among the 1111 considered ones that are not consistents with the corresponding clade and the integer in boldface in S_1 (resp. S_2) corresonds to the clade that is not present in S_2 (resp. S_3).

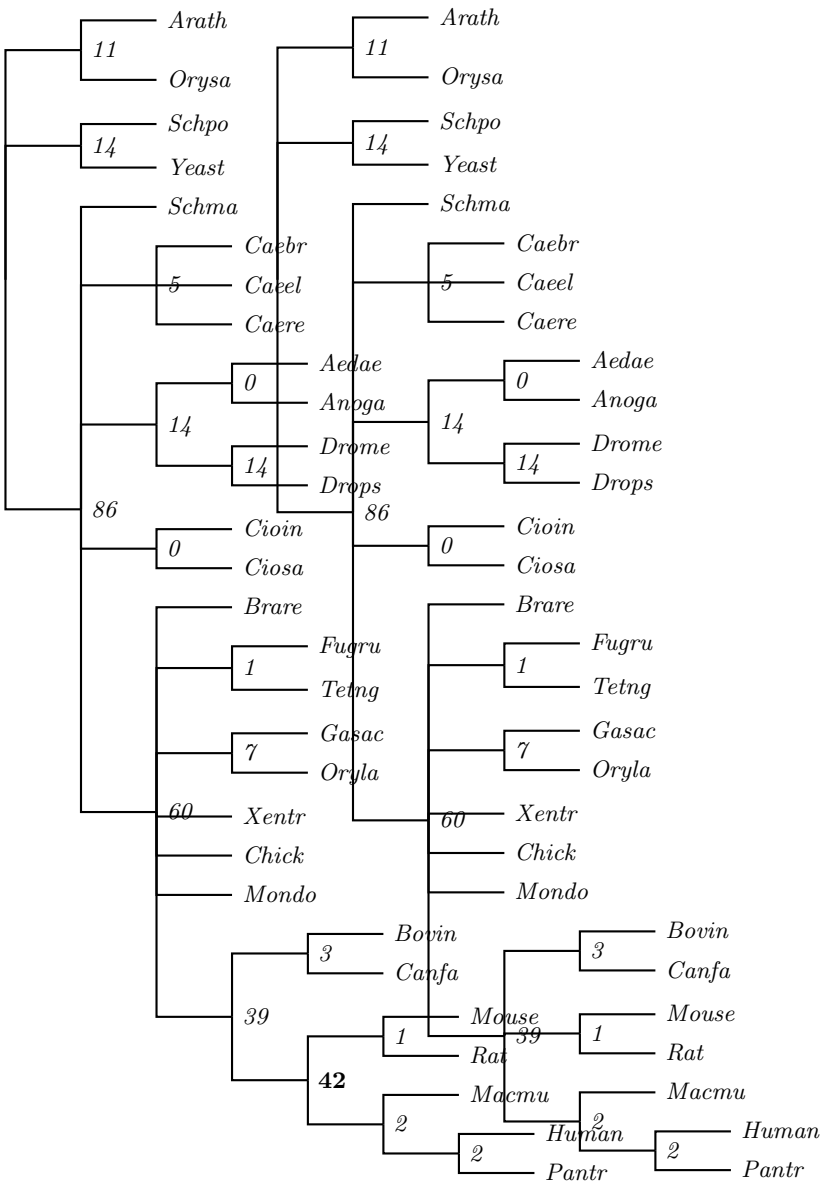

Fig. 7. Constrained species trees S_3 (**left**) and S_4 (**right**).

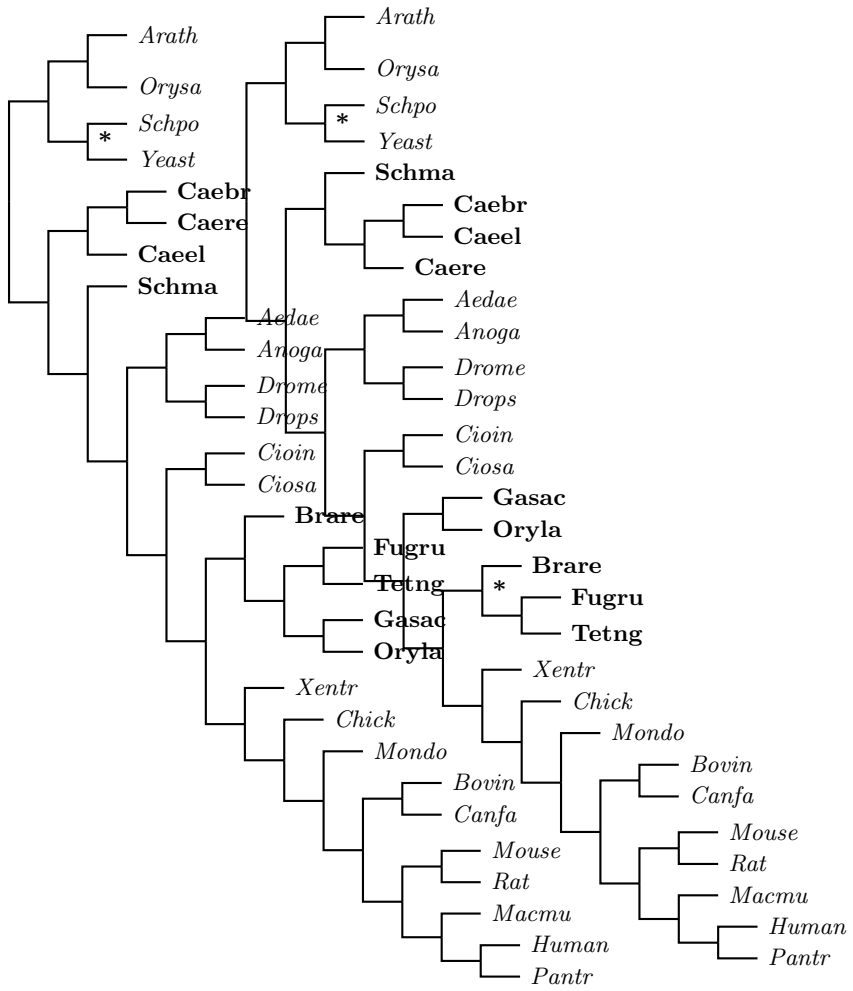

Fig. 8. Left (resp. right): optimal solution for the duplication (resp. loss and mutation) criterion. The character '*' indicates a clade that is misplaced according to the proposed phylogeny (Figure 4) and the taxa in boldface point out the disagreements between the two optimal solutions.