Algorithms for exploring the space of gene tree/species tree reconciliations

Jean-Philippe Doyon[2], Cedric Chauve[1], and Sylvie Hamel[2]

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

[2] DIRO, Université de Montréal, CP6128, succ. Centre-Ville, H3C 3J7, Montréal (QC),Canada, [hamelsyl,doyonjea]@iro.umontreal.ca

Abstract. We describe algorithms to explore the space of all possible reconciliations between a gene tree and a species tree. We propose an algorithm for generating a random reconciliation, and combinatorial operators and algorithms to explore the space of all possible reconciliations between a gene tree and a species tree in optimal time. We apply these algorithms to simulated data.

Note. This extended abstract appeared in the proceedings of the 6th RECOMB Comparative Genomics Satellite Workshop (RCG 2008), volume 5267 of *Lecture Notes in Computer Scence*, pages 1-13, Springer, 2008.

1 Introduction

Genomes of contemporary species, especially eukaryotes, are the result of an evolutionary history, that started with a common ancestor from which new species evolved through evolutionary events called speciations. One of the main objectives of molecular biology is the reconstruction of this evolutionary history, that can be depicted with a rooted binary tree, called a *species tree*, where the root represents the common ancestor, the internal nodes the ancestral species and speciation events, and the leaves the extant species. Other events than speciation can happen, that do not result immediately in the creation of new species but are essential in eukaryotic genes evolution, such as gene duplication and loss [12]. 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 preserves its original role. A gene is said to be lost when it has no function or is fully deleted from the genome. (See [12] for example). Other genomic events such as lateral gene transfer, that occurs mostly in bacterial genomes, will not be considered here. Genes of contemporary species that evolved from a common ancestor, through speciations and duplications, are said to be homologs [9] and are grouped into a gene family. The evolution of a gene family can be depicted with a rooted binary tree, called a *gene tree*, where the leaves represent the homologous contemporary genes, the root their common ancestral gene and the internal nodes represent ancestral genes that have evolved through speciations and duplications.

Given a gene tree G and the species tree of the corresponding genomes S, an important question is to locate in S the evolutionary events of speciations and duplications. A *reconciliation* between G and S is a mapping of the genes (extant and ancestral) of G onto the nodes of S that induces an evolutionary scenario, in terms of speciations, duplications and losses, for the gene family described by G. In this perspective, the notion of reconciliation was first introduced in the pioneering work of [10] and a first formal definition was given in [17] to explain the discrepancies between genes and species trees. The 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, as it depicts a parsimonious evolutionary process according to the number of duplications or duplications and losses it induces. It is widely accepted that parsimony is a pertinent criterion in evolutionary biology, but that it does not always reflects the true evolutionary history. This lead to the definition of more general notions of reconciliations between a gene tree and a species tree [2, 11, 1] and the natural problem of exploring non-optimal (for a given criterion) reconciliations, and then alternative evolutionary scenarios for gene families.

The main concern of our work is the development of algorithms for exploring the *space of the reconciliations* between a gene tree and a species tree. After introducing a very general notion of reconciliation (Section 2), we describe in Section 3 an algorithm that generates a random reconciliation under the uniform distribution, in Section 4.1 combinatorial operators that are sufficient to explore the complete space of reconciliations between a gene tree and a species tree, and in Section 4.2 an algorithm that explores exhaustively this space and computes in optimal time the distribution of reconciliation scores in the duplication, loss, and mutation (duplication + loss) cost models. (All proofs will be given in a future technical report [6]). There are several applications of our algorithms in functional and evolutionary genomics, such as inferring orthologs and paralogs [8, 14], the gene content of an ancestral genome [16], or in the context of Markov Chain Monte Carlo analysis of gene families [1]. We illustrate our algorithms with experiments on simulated gene families in

Section 5 computed using duplication and loss rates taken from [13]. Our experiments suggest that, at least for some real datasets, the use of a parsimony model may be justified.

2 Preliminaries

Let T be a binary tree with vertices $V(T)$ and edges $E(T)$, and such that only its leaves are labeled. Let $r(T)$, $L(T)$, and $\Lambda(T)$ respectively denote its root, the set of its leaves, and the set of the labels of its leaves. We will adopt the convention that the root is at the top of the tree and the leaves at the bottom. A *species tree* S is a binary 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 *gene tree* G is a binary tree. From now on, we consider a species tree S, with $|V(S)| = n$ and a gene tree G such that $\Lambda(G) \subseteq \Lambda(S)$ and $|V(G)| = m$. Let $\sigma : L(G) \to L(S)$ be the function that maps each leaf of G to the unique leaf of S with the same label.

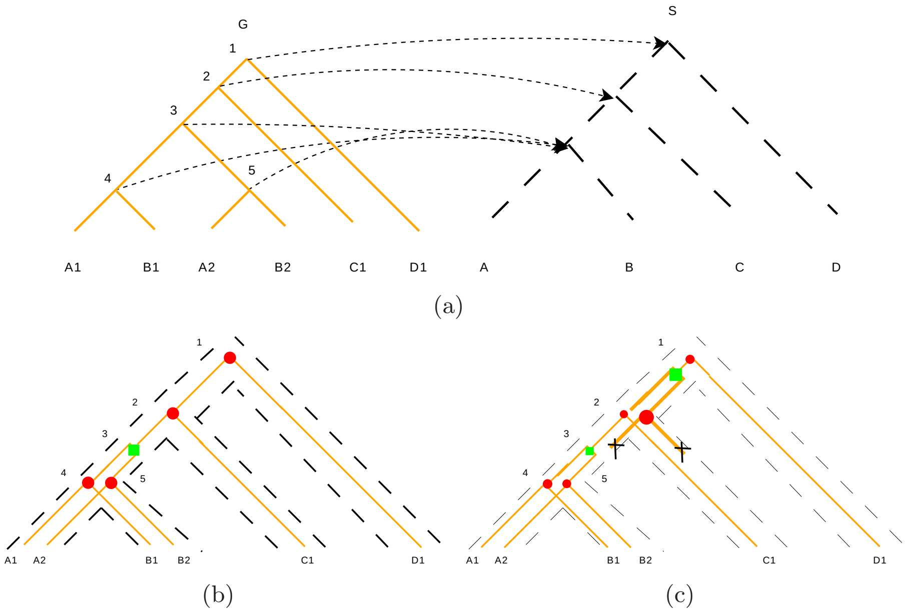
For a vertex u of T, we denote by u_1 and u_2 its children and by T_u the subtree of T rooted at u. For a vertex $u \in V(T) \setminus \{r(T)\}$, we denote by $p(u)$ its parent. A *cell* of a tree T is either a vertex of T or an edge of T. Given two cells c and c' of T, $c' \leq_T c$ (resp. $c' <_T c$) if and only if c is on the unique path from c' to $r(T)$ (resp. and $c \neq c'$); in such a case, c' is said to be a *descendant* of c. The *LCA-mapping* $M : V(G) \to V(S)$ maps each vertex u of G to the unique vertex $M(u)$ of S such that $\Lambda(S_{M(u)})$ is the smallest cluster of S containing $\Lambda(G_u)$.

Definition 1. A reconciliation between a gene tree G and a species tree S is a mapping $\alpha : V(G) \to V(S) \cup E(S)$ such that

1. (Base constraint) $\forall u \in L(G), \alpha(u) = M(u) = \sigma(u)$.
2. (Tree Mapping Constraint) For any vertex $u \in V(G) \setminus L(G)$,
   (a) if $\alpha(u) \in V(S)$, then $\alpha(u) = M(u)$.
   (b) If $\alpha(u) \in E(S)$, then $M(u) <_S \alpha(u)$.
3. (Ancestor Consistency Constraint) For any two vertices $u, v \in V(G)$, such that $v <_G u$,
   (a) if $\alpha(u), \alpha(v) \in E(S)$, then $\alpha(v) \leq_S \alpha(u)$,
   (b) otherwise, $\alpha(v) <_S \alpha(u)$.

Remark 1. This definition of reconciliation differs slightly from the classical ones as vertices of G can be mapped onto edges of S, in order to represent duplication events (see explanations below). However, it is equivalent to the definitions given in [1, 11], that are the most complete ones known so far, and it is more general than the Inclusion-Preserving mapping of [2].

The whole set of reconciliations between a gene tree G and a species tree S is denoted $\Psi(G, S)$. A reconciliation α of $\Psi(G, S)$ implies an evolutionary scenario for the genes of G in terms of gene duplications, gene losses, and speciations. A vertex u of G that is mapped onto an edge (x, y) of S (where $x = p(y)$) represents a gene of the ancestral species $p(y)$ that has been duplicated in y. If u is mapped onto an internal vertex x of S, then this represents a gene that will be present in a single copy in the two genomes x_1 and x_2 following a speciation event that happened to x. It is important to point out that the number of reconciliations is finite. Briefly, a reconciliation α between G and S represents any birth-and-death scenario along S such that the resulting gene tree is consistent with G and each duplication event that implies an internal node u of G is consistent with the mapping $\alpha(u)$. (See Figure 1).



(a)

(b)

(c)

Fig. 1. (a) Left: gene tree G. Right: species tree S. The arrows represent the LCA-mapping between G and S. (b) A reconciliation between G and S. The red circles represent speciation events, and the green squares, duplications. (c) A birth-and-death scenario that is consistent with the reconciliation. A cross represents a gene loss. The right lineage of the first duplication has no extant gene that descents from it, as opposite to its left lineage. We then say that this duplication is hypothetical, because it is not a useful information for the evolutionary scenario of the extant genes of G along S. Hence, such duplication is not depicted by the reconciliation.

We denote by $dup(\alpha)$ and $los(\alpha)$ respectively the number of duplications and losses induced by a reconciliation α. $dup(\alpha)$ is the number of vertices of G that are mapped onto an edge of S (see below [3]). Given two cells $c, c' \in V(S) \cup E(S)$, where $c' <_S c$, $D(c, c')$ is the number of vertices $x \in V(S)$ such that $c' <_S x <_S c$. Also, if $c = c'$, then $D(c, c') = 0$. The number of losses associated to a vertex $u \in V(G) \setminus L(G)$ is noted l_u and equal to $D\big(\alpha(u), \alpha(u_1)\big) + D\big(\alpha(u), \alpha(u_2)\big)$ (see [15] for example). $los(\alpha)$ is then the sum of l_u over all internal vertices u. The third constraint of Definition 1 leads to the notion of *forced duplication*, that corresponds to vertices of G that can only be mapped onto an edge of S: an internal vertex $u \in V(G) \setminus L(G)$ is said to be a forced duplication if and only if $M(u) = M(u_1)$ or $M(u) = M(u_2)$.

For a vertex $u \in V(G)$, a cell of S *covers* it if u can be mapped onto this cell according to Definition 1. The set of cells that can cover it is denoted by $A(u)$ and is defined below.

$$A(u) = \begin{cases} \big\{M(u)\big\} & \text{if } u \in L(G) \text{ or } u = r(G) \\ \big\{c \in E(S) : M(u) <_S c\big\} & \text{if } u \text{ is a forced duplication} \\ \big\{c \in E(S) : M(u) <_S c\big\} \cup \big\{M(u)\big\} & \text{otherwise} \end{cases}$$

It is important to point out that there is three mappings that are considered here: $M(u)$, $\alpha(u)$, and $A(u)$. From now on, except when indicated, the term mapping will refer to the reconciliation mapping $\alpha(u)$ of Definition 1.

Finally, combinatorial and probabilistic criteria can be used to compare the different possible reconciliations and pick one that is supposed to reflect the most the true evolution of G according to S. Three parsimonious cost models, that aim to minimize the number of genomic events, have been proposed so far: duplication [15], loss [4], and mutation (duplication+loss) [15]. Arvestad *et al.* also introduced a notion of likelihood of a reconciliation in the framework of birth-and-death processes [1].

3 Counting and uniform random generation

In this section, we describe an efficient algorithm that computes a random reconciliation between G and S following the uniform distribution. This

[3] To consider duplication that preceds the first speciation event represented by $r(S)$, we can insert in S an "artificial" cell c such that $r(S) <_S c$. For space reason, we assume here that no duplication occurs in the most ancestral species. The details to account for such early duplications will be described in the full version of this paper.

problem is important in the context of MCMC analysis for gene families, as a major issue is to analyze if the Markov chain converges to the true posterior probabilities. One of the most popular and simple tests of convergence is to run several Markov chains, each starting at a different state in the space, which motivates our random generation algorithm.

As usual in uniform random generation, it is based on a preprocessing that computes the cardinality of $\Psi(G,S)$ [5]. We first address this problem, then describe the random generation algorithm.

For every node $u \in V(G)$ and cell $c \in A(u)$, we denote by $Nb(u,c)$ the number of reconciliations of G_u and S_c for which u is mapped on c. It follows immediately that $|\Psi(G,S)| = Nb\left(r\left(G\right), r\left(S\right)\right)$.

Lemma 1. *Let $u \in V(G)$ and $c \in A(u)$ be a cell that covers u. Then $Nb(u,c) = 1$ if $u \in L(G)$, and otherwise*

$$Nb(u,c) = \sum_{c_1 \in A(u_1),\ c_1 \leq_S c} Nb(u_1,c_1) \sum_{c_2 \in A(u_2),\ c_2 \leq_S c} Nb(u_2,c_2). \qquad (1)$$

Proposition 1. *$|\Psi(G,S)|$ can be computed in $O(mn)$ time and space.*

It follows from the work [4] that there is a single optimal reconciliation for the loss and mutation costs, but that there can be several ones for the duplication cost. Building on Lemma 1, we also get the following result, that is of interest with respect to this point.

Proposition 2. *The number of reconciliations of $\Psi(G,S)$ that minimizes the duplication cost can be computed in $O(mn)$ time and space.*

The algorithm 1.1 below computes a random reconciliation between G and S. For a node $u \in V(G)$ and a cell $c \in A(u)$, let $f(c)$ ($d(c)$) be the ancestor (resp. descendant) cell of c in $A(u)$, that is the cell of $A(u)$ that is the closest one to c and above (resp. below) it. The lowest cell of $A(u)$ is the one that has no descendant cell in $A(u)$.

Theorem 1. *Given a reconciliation $\alpha \in \Psi(G,S)$, Algorithm 1.1 returns α with probability $\frac{1}{|\Psi(G,S)|}$. Given the table Nb and the sets $A(u)$ for every node u of G, it can be implemented to run in $O(mn)$ space and $\Theta(mn)$ time in the worst case and $\Theta(m)$ time in the best case.*

Hence, the preprocessing time of our algorithm (computing the table Nb and the sets $A(u)$) requires $O(mn)$ time and space. However, it needs to be done once and can be used for generating several random reconciliations.

Algorithm 1.1 Uniform random generation in $\varPsi(G,S)$.

```
Let $\alpha$ be an empty reconciliation.
Perform a prefix traversal of $G$, and let $u \in V(G)$ be the current node.
    if $u = r(G)$ or $u \in L(G)$ then $\alpha(u) \leftarrow M(u)$
    else
        Let $\hat{c} \leftarrow \alpha(p(u))$.
        {Choose randomly a cell $c \in A(u)$ such that $c \leq_S \hat{c}$}
        Let $k \leftarrow \sum_{c \in A(u), c \leq_S \hat{c}} Nb(u,c)$
        Generate randomly and uniformly an integer $n \in \{1, \ldots, k\}$.
        $c \leftarrow$ lowest cell in $A(u)$ {If $u$ is a forced duplication, then $M(u) \notin A(u)$}
        $l \leftarrow Nb(u,c)$
        while $l < n$ do $c \leftarrow f(c)$, $l \leftarrow l + Nb(u,c)$
        $\alpha(u) \leftarrow c$
return $\alpha$
```

Our algorithm is useful for sampling the space of reconciliations, but not for exhaustive enumeration of that space. Therefore, in the next section, an algorithm for enumeration is introduced.

4 Exhaustive exploration of the whole space $\varPsi(G,S)$

We first define combinatorial operators used to explore the space of all possible reconciliations, and then give an algorithm, based on these operators, that explores exhaustively this space.

4.1 Space exploration operators

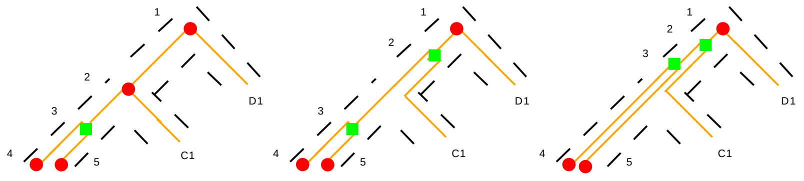
We present in this section a type of operator, called *Nearest Mapping Change* (NMC), acting on a reconciliation between a gene tree G and a species tree S. This movement is similar to the ones described in [11]. We show that this operator is sufficient to explore the space of all possible reconciliations.

Definition 2. Let $\alpha : V(G) \to V(S) \cup E(S)$ be a given reconciliation between G and S, and u a vertex of $V(G) \setminus L(G)$ such that $u \neq r(G)$. Let $\hat{c}, c, c_1$, and c_2 respectively denote $\alpha(p(u))$, $\alpha(u)$, $\alpha(u_1)$, and $\alpha(u_2)$.

1. An *upward* NMC (uNMC) can be applied to u if $c <_S \hat{c}$, and if $\hat{c} \in V(S)$ and $c \in E(S)$, then $D(\hat{c}, c) > 0$. It changes $\alpha(u)$ into its ancestor cell $f(\alpha(u))$ of $A(u)$.
2. A *downward* NMC (dNMC) can be applied to u if $c_1 <_S c$, $M(u) <_S c$, and if $c_1 \in V(S)$ and $c \in E(S)$, then $D(c, c_1) > 0$ (idem for c_2). It changes $\alpha(u)$ into its descendant cell $d(\alpha(u))$ of $A(u)$.

It follows immediately from the definition of NMC operators that, given $\alpha \in \Psi(G,S)$, applying an NMC operator to a vertex u of G results in a reconciliation α' between G and S. More precisely, it can induce the following changes in the evolutionary scenario for the gene family (see Figure 2).

- Changing a speciation by a duplication (uNMC, $\alpha(u) = M(u)$).
- Changing a duplication by a speciation (dNMC, $\alpha'(u) = M(u)$).
- Moving a duplication upward (uNMC, $\alpha(u) \neq M(u)$).
- Moving a duplication downward (dNMC, $\alpha'(u) \neq M(u)$).



(a) (b) (c)

Fig. 2. (a) A section of the reconciliation depicted in figure 1. Here, the mapping of node 2 forbids to move up node 3. (b) The node 2 changes from a speciation to a duplication by moving it up. (c) Then, node 3 can be moved up and still is a duplication.

For $u \in V(G)$, and $c, c' \in A(u)$, $d_u\left(c, c'\right)$ is the number of cells of $A(u)$ between c and c', where $d_u\left(c, c'\right) = 0$ if and only if $c = c'$. For two reconciliations α and α', $D_{NMC}(\alpha, \alpha') = \sum_{u \in V(G)} d_u\left(\alpha(u), \alpha'(u)\right)$. We call $D_{NMC}(\alpha, \alpha')$ the *NMC distance* between α and α'. A valid (according to Definition 2) NMC application to α can be encoded by a pair (u, c), where $u \in V(G)$ is the node being moved and $c \in V(S) \cup E(S)$ is its new mapping. We denote by $NMC(\alpha)$ the set of such pairs for a given reconciliation α.

Theorem 2. *Let α and α' two reconciliations of $\Psi(G,S)$. There exists a sequence of $D_{NMC}(\alpha, \alpha')$ NMC that transforms α into α'. No shorter sequence of NMC can transform α into α'.*

We denote by $\mathcal{G}_{NMC}(G,S)$ the graph with vertex set $\Psi(G,S)$ and where two reconciliations are linked by an edge if they differ by a single NMC. Let α_{min} be the unique reconciliation where, for each vertex u of G, $\alpha_{min}(u)$ is the unique cell of $A(u)$ that has no descendant in $A(u)$, and α_{max} be the unique reconciliation where, for each vertex u of G, $\alpha_{max}(u)$

is the unique cell of $A(u)$ that has no ancestor in $A(u)$. The following results shows that although $\Psi(G,S)$ can have an exponential size, NMC operators are sufficient to define a structure on this space of polynomial diameter.

Corollary 1. *The diameter of* $\mathcal{G}_{NMC}(G,S)$ *is equal to* $D_{NMC}(\alpha_{min},\alpha_{max})$ *and is in* $O(nm)$*.*

Finally, as our NMC operators are intended to explore the space of reconciliations between a gene tree and a species tree, we address now the issue of updating the classical combinatorial criteria used to evaluate a reconciliation: the following observation implies that they can be easily updated in constant time.

Property 1. Let α and α' be two reconciliations of $\Psi(G,S)$ such that α' can be obtained from α by a single NMC. Then, $|dup(\alpha) - dup(\alpha')| \in \{0,1\}$ and $|los(\alpha) - los(\alpha')| \in \{1,2\}$.

4.2 Algorithm for the exhaustive exploration

We present in this section a simple algorithm, based on the NMC operator, that computes the set of all possible reconciliations between a gene tree G (with $|V(G)| = m$) and a species tree S (with $|V(S)| = n$) in time $\Theta(|\Psi(G,S)|)$ (see Theorem 3), which gives a CAT (Constant Amortized Time) algorithm to generate $\Psi(G,S)$.

For a node $u \in V(G)$, let $id(u)$ be the number of nodes that precede u according to the prefix traversal of G, where the left child u_1 of a node $u \in V(G) \setminus L(G)$ is visited before the right child u_2. Let $\mathcal{T}_{NMC}(G,S)$ be the tree defined as follows (see Figure 3):

- the root is the reconciliation α_{min} and its children are the reconciliations that can be obtained from α_{min} by applying a single uNMC from $NMC(\alpha_{min})$,
- given a reconciliation α, that differs from its parent by an uNMC (u_i, c), its children are the reconciliations that can be obtained from α by applying a single uNMC (u_j, c') from $NMC(\alpha)$ such that $id(u_j) \geq id(u_i)$.

Proposition 3. $\mathcal{T}_{NMC}(G,S)$ *is a spanning tree of* $\mathcal{G}_{NMC}(G,S)$*.*

The exhaustive exploration algorithm of the whole space $\Psi(G,S)$ is based on the tree $\mathcal{T}_{NMC}(G,S)$. It follows immediately from the definition of $\mathcal{T}_{NMC}(G,S)$ that the main tasks for a given reconciliation α is 1) to know the list of allowed uNMC operators that can be applied to obtain the children of α, and 2) to keep in order its nodes according to the increasing value of their indexes *id*. We denote by $P(\alpha)$ this ordered list. The key to achieve this efficiently is the Property 2 below, that follows easily from the definitions of NMC operators and of $\mathcal{T}_{NMC}(G,S)$.

Property 2. Let α and α' be two reconciliations of $\Psi(G,S)$ such that α' is a child of α in $\mathcal{T}_{NMC}(G,S)$, and differs from α by an uNMC (u,c). Then $P(\alpha)$ and $P(\alpha')$ differ by at most three uNMC, that involve u, u_1 and u_2.

Based on this property, we describe below an algorithm that performs a prefix traversal of $\mathcal{T}_{NMC}(G,S)$, where the children of a reconciliation α are visited according to the ordered list $P(\alpha)$, in such a way that each time an edge from α to a reconciliation α' is followed, $P(\alpha)$ is updated into $P(\alpha')$. To perform this update in constant time, we encode P using two disjoint lists P_ℓ and P_r and two cursors u_ℓ and u_r on these lists, in such a way that a node u is in P if and only if u is in the sublist of P_ℓ (or P_r) that starts at u_ℓ (resp. u_r).

Algorithm 1.2 below describes the general recursive function, where the main tasks for the current reconciliaton α are i) select the next node $u \in P(\alpha)$ with the smallest *id* from the sublists of P_ℓ or P_r (lines 4,5); ii) define the child reconciliation α' by moving u upward (line 6); iii) define $P(\alpha')$ from $P(\alpha)$ by updating P_ℓ, P_r, u_ℓ, and u_r (lines 7-12). The function is first called with $\alpha = \alpha_{min}$, $P_\ell = NMC(\alpha)$, $P_r = \emptyset$, $u_\ell = first(P_\ell)$, and $u_r = end(P_r)$, that are computed during a preprocessing phase. Here, $first()$ and $end()$ respectively represents the first cursor of the considered list and a null one located at the end of the list. For a node $u \in V(G)$ and a cell $c \in A(u)$, recall that $f(c)$ and $d(c)$ respectively are its ancestor and descendant cells in $A(u)$.

Theorem 3. *Algorithm 1.2 visits all reconciliations of $\Psi(G,S)$. Given α_{min}, and $P_\ell = NMC(\alpha_{min})$, it can be implemented to run in time $\Theta(|\Psi(G,S)|)$ and space $O(nm)$.*

Together with Property 1, that implies that updating the number of duplications and/or losses after a single NMC can be done in constant time, this algorithm allows to compute efficiently the exact distribution of the duplication, loss and mutation costs in optimal time $\Theta(|\Psi(G,S)|)$ (see Section 5).

Algorithm 1.2 Exhaustive exploration algorithm of the space $\Psi(G, S)$

```
1:  RecurExplore $(\alpha, u_\ell, u_r)$
2:      Let $u'_\ell \leftarrow u_\ell$ and $u'_r \leftarrow u_r$
3:      **while** $u'_\ell \neq end(P_\ell)$ or $u'_r \neq end(P_r)$ **do**
4:          **if** $u'_\ell = end(P_\ell)$ **then** $u \leftarrow u'_r$
5:          **else if** $u'_r = end(P_r)$ or $id(u'_\ell) < id(u'_r)$ **then** $u \leftarrow u'_\ell$, **else** $u \leftarrow u'_r$
6:          $\alpha'(u) \leftarrow f(\alpha(u))$
7:          **if** $u_1 \notin P_\ell$ and $u = u'_\ell$ **then** insert $u_1$ in $P_\ell$ after $u'_\ell$
8:          **else if** $u_1 \notin P_\ell$ and $u = u'_r$ **then** insert $u_1$ in $P_\ell$ before $u'_\ell$, $u'_\ell \leftarrow u_1$
9:          **if** $u_2 \notin P_\ell \cup P_r$ and $u = u'_r$ **then** insert $u_2$ in $P_r$ after $u'_r$
10:         **else if** $u_2 \notin P_\ell \cup P_r$ and $u = u'_\ell$ **then** insert $u_2$ in $P_r$ before $u'_r$, $u'_r \leftarrow u_2$
11:         **if** $(u, f(\alpha'(u))) \notin NMC(\alpha')$ and $u = u'_\ell$ **then** $u'_\ell \leftarrow succ(u'_\ell, P_\ell)$
12:         **if** $(u, f(\alpha'(u))) \notin NMC(\alpha')$ and $u = u'_r$ **then** $u'_r \leftarrow succ(u'_r, P_r)$
13:         RecurExplore $(\alpha', u'_\ell, u'_r)$
14:         Retrieve old values of $P_\ell, P_r, u'_\ell, u'_r$ by performing the inverse operations of
            lines 7 to 12.
15:         $\alpha(u) \leftarrow d(\alpha'(u))$ {Backtrack}
16:         **if** $u = u'_\ell$ **then** $u'_\ell \leftarrow succ(u'_\ell, P_\ell)$ **else** $u'_r \leftarrow succ(u'_r, P_r)$
```

5 Experimental results

We considered the phylogenetic tree of 12 *Drosophila* species and the branch lengths, and gene gain/loss rates that are given in [13, Figure 1]. We generated 1000 synthetic gene trees according to the birth-and-death process (with a single ancestral gene) along this species tree, and removed multiple copies of each gene tree. This resulted in 249 unique gene trees having from 6 to 22 leaves (Figure 4). Figure 5 describes the cardinality and diameter of $\Psi(G, S)$ for these 249 gene trees.

For each of the 249 unique gene trees, we used the algorithm 1.2 to explore the whole space $\Psi(G, S)$ focusing on the duplication cost (for the loss and mutation criteria, the results are similar). For the duplication criterion, 237 gene trees have a unique global minimum, and 12 have two. In each of these 12 cases, the NMC distance between the two global minimums is one. Over all the 249 gene trees, the LCA reconciliation α_{min}, that is a global minimum, is either identical or, in the worst case, at a distance of a single NMC to the true evolutionary scenario induced by the birth-and-death and noted α_{real}. However, it is important to point out that this is probably due to the low duplication and loss rates given in [13].

For a reconciliation $\alpha \in \Psi(G, S)$, let $d_{\text{cost}}(\alpha) = dup(\alpha) - dup(\alpha^*)$, where α^* is a global minimum, according to the duplication cost, that minimizes $D_{NMC}(\alpha, \alpha^*)$. We denote by $N(k)$ the number of reconciliations $\alpha \in \Psi(G, S)$ such that $d_{\text{cost}}(\alpha) = k$, for a given $k \in \mathbb{N}$. Figure 6

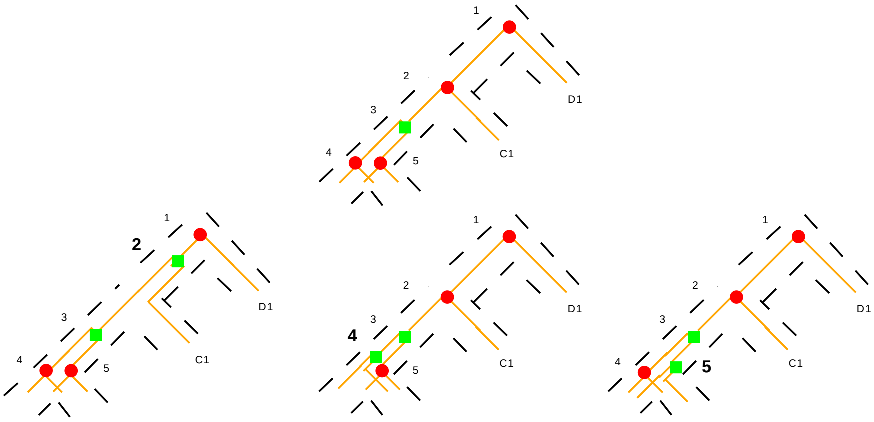


Fig. 3. The subtree of $\mathcal{T}_{NMC}(G,S)$ rooted at α_{min} for the trees G and S depicted in Figure 1. α_{min} and its children respectively are at the top and bottom of the figure. For each child, the node that has been moved upward is in boldface.

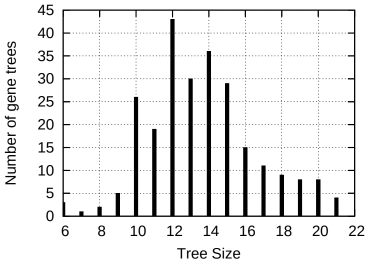


Fig. 4. Distribution of the 249 gene trees according to their number of leaves.

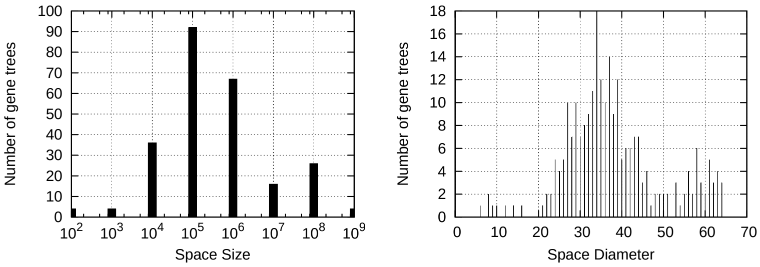


Fig. 5. Left. Distribution of the number of gene trees (y axis) according to the reconciliation space size (x axis). A gene tree G is counted in the bar 10^i iff $10^{i-1} \leq \left|\Psi(G,S)\right| < 10^i$. Right. Distribution of the 249 gene trees according to the diameter of $\Psi(G,S)$.

shows that, on average over all gene trees, $N(k)$ is proportional to k from $k = 0$ to $k = 13$ and inversely proportional from $k = 13$ to $k = 18$. This can be explained by the following facts: the maximum value of d_{cost} is equal to the number of internal nodes u of G that can be mapped on $M(u)$, and the average number of such nodes is 13. All this suggests that, for a given gene tree, $N(k)$ is maximized at this maximum value of $d_{\text{cost}} = k$.

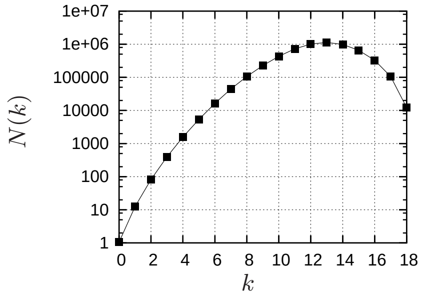


Fig. 6. Over all 249 gene trees, average distribution of the number $N(k)$ (y axis) of reconciliations α such that $d_{\text{cost}}(\alpha) = dup(\alpha) - dup(\alpha^*) = k$, for $k \in \mathbb{N}$ (x axis). α^* is a global minimum that minimizes the NMC distance $D_{NMC}(\alpha, \alpha^*)$.

We analyzed the relationship between the NMC and cost distances using the average value of $D_{NMC}(\alpha, \alpha^*)$ over all gene trees G and all reconciliations $\alpha \in \Psi(G, S)$ such that $d_{\text{cost}}(\alpha) = k$, for a given $k \in \mathbb{N}$. We also computed the number of nodes $u \in V(G)$ such that $\alpha(u) \neq \alpha_{\text{real}}(u)$. According to Figure 7, we observe that the cost distance of a reconciliation α is proportional both to the NMC distance with the closest optimal reconciliation α^* and to how much α differs from the real reconciliation α_{real}.

6 Conclusion

We described in this work several algorithms related to exploring the space of all reconciliations between a gene tree and a species tree. From an algorithmic point of view, our exhaustive exploration algorithm is optimal as it requires an (amortized) constant time between successive reconciliations. Our experiments on a realistic simulated dataset with low duplication/loss rates (we will consider simulated datasets with higher

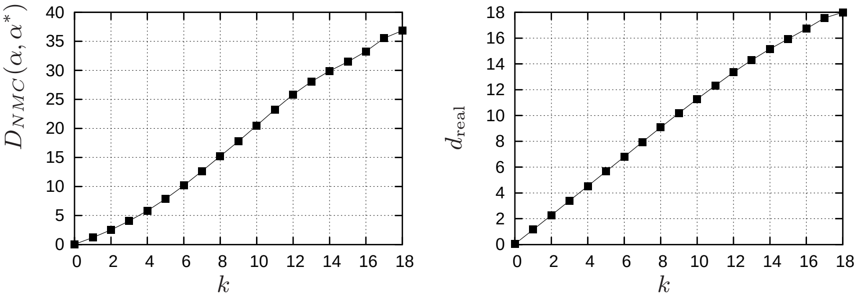


Fig. 7. Left: over all gene trees G, average value of $D_{NMC}(\alpha, \alpha^*)$ (y axis) for all reconciliations $\alpha \in \Psi(G, S)$ such that $d_{\mathrm{cost}}(\alpha) = k$, for a given $k \in \mathbb{N}$ (x axis). Right: the same distribution with the real distance d_{real}, that is the number of nodes u of G such that $\alpha(u) \neq \alpha_{\mathrm{real}}(u)$.

duplication/loss rates in the full version of this paper) show that even in this case the number of reconciliations can be very large, but that for all three combinatorial criterion considered there are relatively few optimal or near-optimal reconciliations, always located close (in terms of NMC distance) to the LCA reconciliation. It is known that for the loss and mutation costs, this LCA reconciliation is the only possible minimum. However, for the duplication cost (as well as for the maximum likelihood cost), it can happen that several optimal reconciliations exist and our exhaustive exploration algorithm was able to locate them. This motivates our current work to modify our algorithm to handle dNMC operators in order to explore efficiently alternative but close evolutionary scenarios (in terms of NMC) of a given reconciliation (work in progress). Our algorithm can already be applied to this task when the starting reconciliation is α_{min} by visiting only the reconciliations that are at a fixed distance (in terms of NMC) from α_{min}. Natural generalizations of the algorithms we described in the present work include handling either non-binary gene or species trees [3, 18] (or both) and attacking the more difficult problem of multiple gene duplications [7]. Moreover, we are now developing our exhaustive exploration algorithm for the maximum likelihood cost.

Acknowledgements. Cedric Chauve akcnowledges the support of NSERC through an Individual Discovery Grant and of Simon Fraser University through a Startup Grant.

References

1. L. Arvestad, A.-C. Berglund, J. Lagergren and B. Sennblad. Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution. In *RECOMB 2004*, pp. 326–335, 2004.
2. P. Bonizzoni, G. Della Vedova and R. Dondi. Reconciling a gene tree to a species tree under the duplication cost model. *Theoret. Comput. Sci.*, 347:36–53, 2005.
3. W.-C. Chang, O. Eulenstein. Reconciling gene trees with apparent polytomies. In *COCOON 2006*, *LNCS* 4112, pp. 235-244. 2006
4. C. Chauve, J.-P. Doyon and N. El-Mabrouk. Gene family evolution by duplication, speciation and loss. To appear in J. Comput. Biol., 2008.
5. A. Denise and P. Zimmermann. Uniform random generation of decomposable structures using floating-point arithmetic. *Theoret. Comput. Sci.*, 218:233–248, 1999.
6. J.-P. Doyon, C. Chauve and S. Hamel. Algorithms for exploring the space of gene tree/species tree reconciliations. IRO Technical Report # 1323, 2008.
7. M.R. Fellows, M.T. Hallett and U. Stege. On the multiple gene duplication problem. In *ISAAC 1998*, *LNCS* 1553, pp. 347–356. 1998.
8. W.M. Fitch. Distinguishing homologous from analogous proteins. *Syst. Zool.*, 19:99–113, 1970.
9. W.M. Fitch. Homology - a personal view on some of the problems. *Trends Genet.*, 16:227 – 231, 2000.
10. 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. *Syst. Zool.*, 28:132–163, 1979.
11. P. Górecki and J. Tiuryn. DLS-trees: a model of evolutionary scenarios. *Theoret. Comput. Sci.*, 359:378–399, 2006.
12. D. Graur and W.-H. Li. *Fundamentals of Molecular Evolution second edition.* Sinauer Associates, Sunderland, MA., 1999.
13. M.W. Hahn, M.V. Han and S.-G. Han. 2007. Gene family evolution across 12 *Drosophilia* genomes. *PLoS Genet.*, 3:e197, 2007.
14. R. Jensen. Orthologs and paralogs - we need to get it right. *Genome Biology*, 2:interactions1002.1–interactions1002.3, 2001.
15. B. Ma, M. Li and L. Zhang. From gene trees to species trees. *SIAM J. Comput.*, 30:729–752, 2001.
16. J. Ma, A. Ratan, L. Zhang, W. Miller and D. Haussler A heuristic algorithm for reconstructing ancestral gene orders with duplications. In *RCG 2007*, *LNCS* 4751, pp. 122–135. 2007.
17. R. D. Page. Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas. *Syst. Biol.*, 43:58–77, 1994.
18. B. Vernot, M. Stolzer, A. Goldman and D. Durand. Reconciliation with non-binary species trees In *CSB2007* pp. 441–452. 2007.