Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Genome analysis

Gene Tree and Species Tree Reconciliation with Endosymbiotic Gene Transfer

Yoann Anselmetti [1], Nadia El-Mabrouk [2,*], Manuel Lafond [1] and Aïda Ouangraoua [1]

[1] Département d'informatique, Université de Sherbrooke, Québec, Canada.
[2] Département d'informatique et de recherche opérationnelle, Université de Montréal, Québec, Canada.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: It is largely established that all extant mitochondria originated from a unique endosymbiotic event integrating an $\alpha-$proteobacterial genome into an eukaryotic cell. Subsequently, eukaryote evolution has been marked by episodes of gene transfer, mainly from the mitochondria to the nucleus, resulting in a significant reduction of the mitochondrial genome, eventually completely disappearing in some lineages. However, in other lineages such as in land plants, a high variability in gene repertoire distribution, including genes encoded in both the nuclear and mitochondrial genome, is an indication of an ongoing process of Endosymbiotic Gene Transfer (EGT). Understanding how both nuclear and mitochondrial genomes have been shaped by gene loss, duplication and transfer is expected to shed light on a number of open questions regarding the evolution of eukaryotes, including rooting of the eukaryotic tree.

Results: We address the problem of inferring the evolution of a gene family through duplication, loss and EGT events, the latter considered as a special case of horizontal gene transfer occurring between the mitochondrial and nuclear genomes of the same species (in one direction or the other). We consider both EGT events resulting in maintaining or removing the gene copy in the source genome. We present a linear-time algorithm for computing the DEL (Duplication, EGT and Loss) distance, as well as an optimal reconciled tree, for the unitary cost, and a dynamic programming algorithm allowing to output all optimal reconciliations for an arbitrary cost of operations. We illustrate the application of our EndoRex software and analyse different costs settings parameters on a plant dataset and discuss the resulting reconciled trees.

Availability: EndoRex implementation and supporting data are available on the GitHub repository via https://github.com/AEVO-lab/EndoRex

1 Introduction

Genomics and cell biology investigations have revealed that all known eukaryotes descend from a common ancestral mitochondrial-containing cell that originated from the integration of an endosymbiotic $\alpha-$proteobacterium into a host cell [1]. After this early event, eukaryotic gene contents have been shaped by duplications, losses and Horizontal Gene Transfers (HGT) from one species to another, but also by Endosymbiotic Gene Transfers (EGT), mainly from the mitochondrion to the nucleus, in some cases leading to the total disapearance of the mitochondrion [2, 3].

Many questions regarding the ancestral mitochondrial proteome and gene content evolution remain open [4]. One of the reasons is that, to date, comparative genomics studies have largely focused on multicellular eukaryotes, mainly animals and plants. While imprints of global evolutionary events at the genomic level are hardly visible on multicellular eukaryotes that have diverged too much from the Last Eukaryotic Common Ancestor (LECA), protists, known to have emerged close to the eukaryotic origin, are better candidates for such a comprehensive evolutionary study. Interestingly, a recent sequencing

effort on jakobids [5] and malawimonads [6] protist genomes have been undertaken by a consortium of protistologists (DeepEuk), suggesting that soon enough data will be available to allow further investigations on early-eukaryotic evolution.

In addition to having the appropriate datasets, understanding the concerted evolution of the eukaryotic mitochondrial and nuclear genomes also requires having the appropriate algorithmic tools. This problem can be seen as related to the host-parasite coevolution inference problem [22]. Given a host tree and a parasite tree, cophylogenetic analysis consist in inferring a history of codivergence, parasite duplication, host switch or extinction events explaining the coevolution of hosts and parasites. However, nuclear and mitochondrial genomes can hardly be treated by the same kind of approach, as they evolve, through a different evolutionary model, together in the same species, and thus are related through the same species tree. Rather, inferring an endosymbiotic evolutionary history requires focusing on gene families and studying the movement of genes between the mitochondrial and nuclear genomes.

Inferring the evolution of gene families is the purpose of the gene-tree-species-tree-reconciliation field, seeking for a most parsimonious [7, 8], or a most probable [9, 10] evolutionary scenario of gene gain and loss explaining the incongruence between a gene tree and a species tree. A most parsimonious reconciliation minimizing the number of Duplications (the D-distance) or the number of Duplications and Losses (the DL-distance) can be found in linear time using the LCA (Last Common Ancestor) mapping [11]. Such an algorithm can actually be used to solve the cophylogenetic problem if operations are restricted to coevolution, duplication and extinction. Including HGT events (i.e. finding the DTL-distance) leads to an NP-hard problem if time-consistency is required, remaining polynomial otherwise [12, 13].

In this paper, we introduce the reconciliation model accounting for EGT events, i.e. the special case of HGT events where genes are exchanged between the mitochondrial and nuclear genomes of the same species. Although integration of the mitochondrial content into the nucleus is the most frequent event in the course of evolution of eukaryotes, the transfer from the nucleus to the mitochondrion has also been observed [21]. Here, we consider the exchange of genes in both directions. Moreover, we consider EGT events resulting in maintaining a gene copy in the source genome (we call them Transfers, or EGTr), as well as those resulting in the removal or loss of function of the gene in the source genome (we call them Transpositions, or EGTr).

Formally, given a gene tree for a gene family with a known mitochondrial or nuclear location for each gene copy, we seek for a most parsimonious sequence of Duplication, EGT and Loss (DEL) events explaining the tree given a known species tree. First, based on the DL-distance and on the Fitch algorithm for weighted parsimony, we present, in Section 3, a linear-time algorithm for computing the DEL-Distance, as well as an optimal reconciled tree for the unitary cost. We then develop, in Section 4, a general dynamic programming algorithm that can be used to output all optimal reconciliations, for an arbitrary cost of operations, including possibly a different cost for an EGT from the mitochondria to the nucleus, or conversely.We finally illustrate, in Section 5, the application of our EndoRex software on clusters of orthologous mitochondrial protein-coding genes (MitoCOGs) [14] of plants, analyse different costs settings parameters and discuss the obtained reconciled trees.

For space reasons, some of the proofs are given in Appendix.

2 Preliminaries

All trees are considered rooted. Given a tree T, we denote by $r(T)$ its root, by $V(T)$ its set of nodes and by $\mathcal{L}(T) \subseteq V(T)$ its leafset. A node x is a *descendant* of x' if x is on the path from x' to a leaf of T and an *ancestor* of x' if x is on the path from $r(T)$ to x'; x is a *strict descendant* (respect. *strict ancestor*) of x' if it is a descendant (respec. ancestor) of x' different from x'. Moreover, x is the *parent* of $x' \neq r(T)$ if it directly precedes x' on the path from x' to $r(T)$. In this latter case, x' is a *child* of x. We denote by $E(T)$ the set of edges of T, where an edge is represented by its two terminal nodes (x, x'), with x being the parent of x'. An internal node (a node which is not a leaf) is said to be *unary* if it has a single child and *binary* if it has two children. If not stated differently, the children of a binary node x are denoted x_l and x_r. Given a node x of T, the subtree of T rooted at x is denoted $T[x]$.

A *binary tree* is a tree with all internal nodes being binary. If internal nodes have one or two children, then the tree is said *partially binary*.

The *lowest common ancestor* (LCA) in T of a subset L' of $\mathcal{L}(T)$, denoted $lca_T(L')$, is the ancestor common to all the nodes in L' that is the most distant from the root.

A tree R is *an extension* of a tree T if it is obtained from T by *grafting* unary or binary nodes in T, where grafting a unary node x on an edge (u, v) consists in creating a new node x, removing the edge (u, v) and creating two edges (u, x) and (x, v), and in the case of grafting a binary node, also creating a new leaf y and a an edge (x, y). In the latter case, we say that y is a grafted leaf.

Species and gene trees: The *species tree* S for a set Σ of species represents a partially ordered set of speciation events that have led to Σ. In this paper, we consider that each species of $\sigma \in \Sigma$ has two genomes: σ_0 corresponding to its mitochondrial genome and σ_1 corresponding to its nuclear genome.

A *gene family* is a set Γ of genes where each gene x belongs to a given species $s(x)$ of Σ. A tree T is a *gene tree* for a gene family Γ if its leafset is in bijection with Γ. We will make no distinction between a leaf of T and the gene of Γ it corresponds to. We call $s(x)$ the *species labeling* of the leaf x. For a subset $G \subseteq \Gamma$ of genes, we write $s(G) = \{s(g) : g \in G\}$ as the set of species containing the genes of G.

Moreover, we assign to each gene x of Γ a boolean value corresponding to the genome it belongs to. More precisely, $b(x) = 0$ if x belongs to $s(x)_0$ and $b(x) = 1$ if x belongs to $s(x)_1$. In this paper, we assume that the mitochondrial or nuclear location of each extant gene is known. We call $b(x)$ the *genome labeling* of the leaf representing x.

An evolutionary history is represented by an *event labeled* tree, where the event label $\tilde{e}(x)$ of an internal node x is its corresponding event. The event labeling of the internal nodes of a gene tree is obtained through reconciliation.

2.1 Reconciliation

Inside the species' genomes, genes undergo *Speciation* (Spe) when the species to which they belong do, but also *Duplication* (Dup) i.e. the creation of a new gene copy, *Loss* of a gene copy, and *Horizontal Gene Transfer* (HGT) when a gene is transmitted from a source to a target genome. In this paper, we consider special cases of HGTs, called EGTs, only allowing the transmission of genes from the mitochondrial genome to the nuclear genome of the same species, or vice-versa. Moreover we consider two types of EGTs: *EGTransfers*, or simply *Transfers*, denoted EGTr, and *EGTranspositions*, or simply *Transpositions*, denoted EGTp, defined as follows (see Figure 1):

- A gene x belonging to σ_i is *Transposed* (by an EGTp event) to σ_j for $\{i, j\} = \{0, 1\}$ if it is inserted in σ_j and removed from σ_i;
- A gene x belonging to σ_i is *Transferred* (by an EGTr event) to σ_j for $\{i, j\} = \{0, 1\}$ if it is inserted in σ_j without being removed from σ_i.

picture(0,0)(-35,0)(1,0)30 (0,35)(0,-1)30 picture

picture(0,0)(35,0)(-1,0)30 (0,35)(0,-1)30 picture

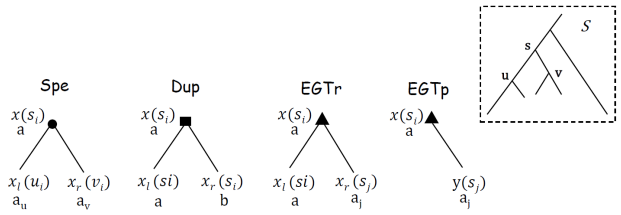

Fig. 1. The effect of an event on a node x of a gene tree representing the gene a belonging to the genome s_i, where s is a species and $i \in \{0, 1\}$ (for a species s, s_o is the mitochondrial genome and s_1 the nuclear genome of s). The tree S up-right is the species tree, where u and v are the two species arising from the speciation of s. (Spe): Gives rise to a copy a_u in u_i and a_v in v_i; (Dup): Preserves the copy a in s_i and gives rise to a new copy b in s_i; (EGTr): Represents a Transfer event from s_i to s_j preserving the copy a in s_i and giving rise to a new copy a_j in s_j; (EGTp): Represents a Transposition event from s_i to s_j removing the copy a in s_i and creating a copy a_j in s_j.

Thus, in this paper, the set of considered events is:

$$DEL = \{Spe, Dup, Loss, EGTr, EGTp\}$$

To define a DEL-Reconciliation, we need to define an extension of the genome labeling b and the species labeling s to the internal nodes of a tree. Given a tree T, an extension R of T (R can be equal to T), and a mapping s from $\mathcal{L}(T)$ to $V(S)$, *an extension of* s is a function $\tilde{s}$ from $V(R)$ to $V(S)$ such that, for each leaf x of T, $\tilde{s}(x) = s(x)$. Moreover, given an assignment b of the leaves of T, an *extension of* b is a function $\tilde{b}$ from $V(R)$ to $\{0, 1\}$ such that, for each leaf x of T, $\tilde{b}(x) = b(x)$.

Definition 1 (DEL-Reconciliation). *Let Γ be a gene family where each $x \in \Gamma$ belongs to the genome $b(x)$ of a species $s(x)$ of Σ. Let T be a binary gene tree for Γ and S be a binary species tree for Σ. A* DEL-Reconciliation *is a quadruplet $\langle R, \tilde{s}, \tilde{b}, \tilde{e}\rangle$ where R is a partially binary extension of T, $\tilde{s}$ is an extension of s and $\tilde{b}$ is an extension of b such that:*

1. *Each unary node x with a single child y is such that $\tilde{e}(x) = EGTp$, $\tilde{s}(x) = \tilde{s}(y) = \sigma$ and $\tilde{b}(x) \neq \tilde{b}(y)$; x represents a transposition event with source genome $\sigma_{\tilde{b}(x)}$ and target genome $\sigma_{\tilde{b}(y)}$.*
2. *For each binary node x of R with two children x_l and x_r, one of the following cases holds:*
   a. *$\tilde{s}(x_l)$ and $\tilde{s}(x_r)$ are the two children of $\tilde{s}(x)$ in S and $\tilde{b}(x_l) = \tilde{b}(x_r) = \tilde{b}(x)$, in which case $\tilde{e}(x) = Spe$;*
   b. *$\tilde{s}(x_l) = \tilde{s}(x_r) = \tilde{s}(x) = \sigma$ and $\tilde{b}(x_l) = \tilde{b}(x_r) = \tilde{b}(x)$ in which case $\tilde{e}(x) = Dup$ representing a duplication in $\sigma_{\tilde{b}(x)}$;*
   c. *$\tilde{s}(x_l) = \tilde{s}(x_r) = \tilde{s}(x) = \sigma$ and $\tilde{b}(x_l) \neq \tilde{b}(x_r)$ in which case $\tilde{e}(x) = EGTr$; let y be the element of $\{x_l, x_r\}$ such that $\tilde{b}(x) \neq \tilde{b}(y)$, then $\tilde{e}(x)$ is a transfer with source genome $\sigma_{\tilde{b}(x)}$ and target genome $\sigma_{\tilde{b}(y)}$.*

A grafted leaf on a newly created node x corresponds to a loss in $\tilde{s}(x)$.

As R is as an extension of T, each node in T has a corresponding node in R. In other words, we can consider that $V(T) \subseteq V(R)$. In particular the species labeling on R induces a species labeling on T.

Given a cost function c on DEL and a reconciliation $\mathcal{R} = \langle R, \tilde{s}, \tilde{b}, \tilde{e}\rangle$, the cost $c(\mathcal{R})$ is the sum of costs of the induced events. In this paper, we assume a 0 cost for speciations and positive costs for all the other events.

We are now ready to formally define the considered optimization problem.

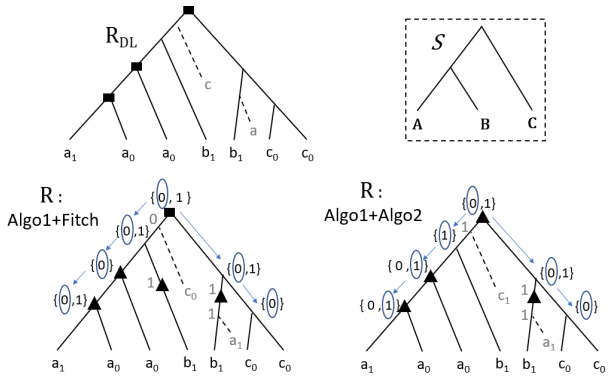

Fig. 2. The tree R_{DL} up left, together with its node labeling, is the optimal DL-Reconciliation for the gene tree T represented by the plain edges of R_{DL} and the species tree S up right. The two down trees are obtained by Algorithm 1 for two different $\tilde{b}$ labeling of internal nodes: the left labeling is obtained by the Fitch algorithm for weighted parsimony, while the right labeling is obtained by applying Algorithm 2. The left labeling gives rise to a non-optimal reconciliation with seven operations (two losses, one duplication, two EGTr and two EGTp), while the right labeling gives rise to the DEL-Distance which is equal to six (two losses, three EGTr and one EGTp). Rectangles represent duplications; triangles represent either EGTr or EGTp events depending whether the labeled node is binary or unary; dotted lines represent losses; A leaf x_i represent a gene x belonging to the genome i (0 for mitochondrial and 1 for nuclear) of species X.

DEL-Reconciliation Problem:
Input: A species tree S for a set of species Σ, a gene family Γ on Σ, a gene tree T for Γ, a species labeling s and a genome labeling b of $\mathcal{L}(T)$, and a cost function c on DEL.
Output: A most *parsimonious DEL-Reconciliation*, i.e. a DEL-Reconciliation $\langle R, \tilde{s}, \tilde{b}, \tilde{e}\rangle$ of minimum cost.

In the next section, we first consider the case of a unitary cost, thus reducing the problem to minimizing the number of operations induced by a reconciliation. The cost $DEL(T, S)$ of the most parsimonious DEL-Reconciliation for T and S in the case of a unitary cost c is called the *DEL-Distance*. We then extend the algorithmic developments to arbitrary costs, allowing in particular to consider an EGTr or an EGTp event transferring a gene from the mitochondria to the nucleus differently from a similar event transferring a gene from the nucleus to the mitochondria.

In the following section, we will refer to the DL-Reconciliation of T and S. Recall that it is a triplet $\langle R_{DL}, \tilde{s}, \tilde{e}\rangle$ defined by only considering the cases of speciations, duplications and losses in Definition 1, and ignoring the binary assignment of genes. We denote by $DL(T, S)$ the DL-Distance, i.e. the minimum number of duplications and losses induced by a DL-reconciliation. The DL-Reconciliation $\langle R_{DL}, \tilde{s}, \tilde{e}\rangle$ of cost $DL(T, S)$ is unique and verifies, for any internal node x of $V(R_{DL}) \cap V(T)$:

1. $\tilde{s}(x) = lca_S(s(\mathcal{L}(T[x])))$;
2. if $\tilde{s}(x) \neq \tilde{s}(x_l)$ and $\tilde{s}(x) \neq \tilde{s}(x_r)$ then v is a Speciation; otherwise x is a Duplication.

We finally need to make the link between the species labeling $\tilde{s}$ of an optimal reconciliation and the well-known LCA-Mapping. This is formally stated in the following lemma.

Lemma 1 (LCA-Mapping). *Let $\langle R, \tilde{s}, \tilde{b}, \tilde{e}\rangle$ be a DEL-Reconciliation of minimum cost between T and S. Then, for each $x \in V(R) \cap V(T)$, $\tilde{s}(x) = lca_S(s(\mathcal{L}(T[x])))$.*

picture(0,0)(-35,0)(1,0)30 (0,-35)(0,1)30 picture

picture(0,0)(35,0)(-1,0)30 (0,-35)(0,1)30 picture

3 A linear-time algorithm for the DEL-Distance

In this section, we consider a unitary cost c on DEL.

Consider a *given* extension $\tilde{b}_T$ of b to the internal nodes of T. We first present an algorithm for computing a DEL-Reconciliation $\langle R, \tilde{s}, \tilde{b}, \tilde{e}\rangle$ of minimum cost, under the condition that $\tilde{b}(x) = \tilde{b}_T(x)$ for each $x \in V(T) \cap V(R)$. We will then show how a $\tilde{b}_T$ minimizing the DEL-Distance can be obtained.

Algorithm 1 computes the DEL-Reconciliation $\langle R, \tilde{s}, \tilde{b}, \tilde{e}\rangle$ from the DL-Reconciliation $\langle R_{DL}, \tilde{s}_{DL}, \tilde{e}_{DL}\rangle$ (see Figure 2 for an example).

Lemma 2 (Optimality of Algorithm 1). *Given a binary assignment $\tilde{b}_T$ of the nodes of T, Algorithm 1 outputs a DEL-Reconciliation $\langle R, \tilde{s}, \tilde{b}, \tilde{e}\rangle$ of minimum cost with the constraint that $\tilde{b}(x) = \tilde{b}_T(x)$ for $x \in V(R) \cap V(T)$.*

It follows from Lemma 2 that if $\tilde{b}$ is known in advance for the nodes of T, a DEL-Reconciliation of minimum cost is obtained from Algorithm 1 with $\tilde{b}$ as input. We now focus on finding such a labeling $\tilde{b}$.

Lemma 3 (Necessary condition for $\tilde{b}$). *There exists a DEL-Reconciliation $\langle R, \tilde{s}, \tilde{b}, \tilde{e}\rangle$ of minimum cost $DEL(T, S)$ such that, for any node x of T and its children x_l and x_r in T, $\tilde{b}(x) = \tilde{b}(x_l)$ or $\tilde{b}(x) = \tilde{b}(x_r)$.*

Proof. Assume $\langle R, \tilde{s}, \tilde{b}, \tilde{e}\rangle$ is a most parsimonious DEL-Reconciliation with a lowest node x not satisfying condition (1): $\tilde{b}(x) = \tilde{b}(x_l)$ or $\tilde{b}(x) = \tilde{b}(x_r)$. Thus we should have $\tilde{b}(x) \neq \tilde{b}(x_l) = \tilde{b}(x_r)$. Note that an EGTp event must be present on at least one of the (x, x_l) or (x, x_r) branches. A reconciliation of lower or equal cost can be obtained by assigning $\tilde{b}(x) = \tilde{b}(x_l) = \tilde{b}(x_r)$ and removing this EGTp event, reducing the cost by one. Let p_x be the parent of x in R (note that if x is the root, p_x might not exist, in which case there is nothing else to do). If $\tilde{b}(x)$ is now different from $\tilde{b}(p_x)$, we add an EGTp event between p_x and x, yielding an alternate reconciliation of equal or lower cost.

We can reproduce the same transformation iteratively in a bottom-up fashion until condition (1) is satisfied for every node. □

For a node $x \in V(T)$, define $d(x) = 1$ if x is a duplication in the DL-Reconciliation of minimum cost, and $d(x) = 0$ otherwise. Let $\tilde{b}$ be a binary labeling of $V(T)$. For any node x of T, denote $\Delta_{\tilde{b}}(x) = 0$ if $x \in \mathcal{L}(T)$, otherwise

$$\Delta_{\tilde{b}}(x) = \max(0, |\tilde{b}(x) - \tilde{b}(x_l)| + |\tilde{b}(x) - \tilde{b}(x_r)| - d(x))$$

and define :

$$cost(T, S, \tilde{b}) = \sum_{x \in V(T)} \Delta_{\tilde{b}}(x)$$

Roughly speaking, $\Delta_{\tilde{b}}(x)$ reflects the number of label changes between x and its children x_l and x_r in T, with the exception that a duplication is allowed a "free" change since it can be turned into an EGTr node. For example, in Figure 2, $cost(T, S, \tilde{b}) = 2$ for the labeling $\tilde{b}$ of T consistent with that of the left tree R (Algo1+Fitch), and $cost(T, S, \tilde{b}) = 1$ for the labeling $\tilde{b}$ of T consistent with that of the right tree R (Algo1+Algo2), reflecting, for each one, the number of requested EGTp.

Lemma 4. *The minimum cost of a DEL-Reconciliation between a gene tree T and a species tree S is*

$$DEL(T, S) = DL(T, S) + \min_{\tilde{b}} cost(T, S, \tilde{b})$$

Proof. By Lemma 2, Algorithm 1 correctly infers a minimum cost DEL-Reconciliation for a given $\tilde{b}$. Note that this DEL-Reconciliation is obtained from a DL-Reconciliation by changing some duplication nodes to EGTr nodes (which do not change the cost), and by grafting some EGTp nodes. Thus the latter are responsible for any possible change in cost from $DL(T, S)$ to $DEL(T, S)$. It follows that the cost of the returned DEL-Reconciliation is $DL(T, S)$, plus the number of grafted EGTp nodes.

Let $\tilde{b}$ be a binary assignment of T that minimizes $DEL(T, S)$ when $\tilde{b}$ is passed to Algorithm 1. By Lemma 3, we may assume that for any node x and its children x_l and x_r, $\tilde{b}(x) = \tilde{b}(x_l)$ or $\tilde{b}(x) = \tilde{b}(x_r)$. Thus $\Delta_{\tilde{b}}(x) \in \{0, 1\}$ for every x. Furthermore, $\Delta_{\tilde{b}}(x) = 1$ if and only if x is a speciation node and an EGTp node is grafted on the edge (x, x_l) (if $\tilde{b}(x) \neq \tilde{b}(x_l)$) or on the edge (x, x_r) (if $\tilde{b}(x) \neq \tilde{b}(x_r)$). In consequence, $cost(T, S, \tilde{b})$ counts exactly the number of graftings of EGTp nodes. □

Algorithm 1 $MinDELrec$

Input: $T, S, s, \tilde{b}_T$; *Output:* $R, \tilde{s}, \tilde{b}, \tilde{e}$.

```
Let ⟨R_DL, s̃_DL, ẽ_DL⟩ be the reconciliation satisfying the DL-Distance;
R = R_DL; s̃ = s̃_DL; ẽ = ẽ_DL;
for each node x of R in a bottom-up traversal do
    if x ∈ V(T) then
        b̃(x) = b̃_T(x)
    else
        Let y be the highest node of T with the parent of x being an ancestor of y;
        b̃(x) = b̃(y);
for each node x of T in a top-down traversal do
    Let x_l and x_r be the two children of x in T;
    if b̃(x_l) = b̃(x) and b̃(x_r) ≠ b̃(x) then
        if ẽ_DL(x) = Dup then
            ẽ(x) = EGTr
        else
            ẽ(x) = Spe;
            Graft a node v on the edge (x, v_1) where v_1 is an ancestor of x_r in R;
            ẽ(v) = EGTp; b̃(v) = b̃(x_r); s̃(v) = s̃(x);
    else if b̃(x_l) ≠ b̃(x) and b̃(x_r) = b̃(x) then
        if ẽ_DL(x) = Dup then
            ẽ(x) = EGTr
        else
            ẽ(x) = Spe;
            Graft a node v on the edge (x, v_1) where v_1 is an ancestor of x_l in R;
            ẽ(v) = EGTp; b̃(v) = b̃(x_l); s̃(v) = s̃(x);
    else if b̃(x_l) ≠ b̃(x) and b̃(x_r) ≠ b̃(x) then
        if ẽ_DL(x) = Dup then
            ẽ(x) = EGTr;
            Graft a node v on the edge (x, v_1) where v_1 is an ancestor of x_r in R;
            ẽ(v) = EGTp; b̃(v) = b̃(x_r); s̃(v) = s̃(x);
        else
            ẽ(x) = Spe;
            Graft a node v_l on the edge (x, v_1l) where v_1l is an ancestor of x_l in R;
            Graft a node v_r on the edge (x, v_1r) where v_1r is an ancestor of x_r in R;
            ẽ(v_l) = ẽ(v_r) = EGTp; b̃(v_l) = b̃(v_l) = b̃(x_l); ;
            s̃(v_l) = s̃(x_l); s̃(v_r) = s̃(x_r);
```

picture(0,0)(-35,0)(1,0)30 (0,35)(0,-1)30 picture

picture(0,0)(35,0)(-1,0)30 (0,35)(0,-1)30 picture

Since the most-parsimonious DL-Reconciliation is unique, the $DL(T,S)$ term in the above lemma is an invariant. Our goal is therefore to find the labeling $\tilde{b}$ that minimizes $cost(T,S,\tilde{b})$.

This can be achieved by a slight modification of the Fitch (1971) algorithm [19] computing, for a given tree with leaf labels, all possible label assignments of internal nodes minimizing the number of label changes along the edges of the tree. We first need to recall some concepts on parsimony. Given a tree T on a leafset L of residues (generally nucleotides or amino-acids, but in this paper $L=\{0,1\}$ corresponding to the possible $\tilde{b}$ labeling), the *weighted parsimony* problem consists in assigning a residue $\tilde{b}(u)\in L$ to each internal node u of T in a way minimizing the total weight of the tree. More precisely, given a cost matrix M on residues, the weight of T is the sum of weights $M(\tilde{b}(u),\tilde{b}(v))$ for all $(u,v)\in E(T)$. An *assignment of* T refers to the assignment of a residue to each internal node of T.

The Sankoff & Cedergren (1983) algorithm [20] allows computing, in quadratic time, the minimum cost $\min(T)$ of an assignment of T. Moreover, it allows finding all the assignments $\tilde{T}$ of T leading to $\min(T)$. When $M(a,a)=0$ for all $a\in L$ and $M(a,b)=1$ for $a\neq b$, weighted parsimony can be computed in linear time using the Fitch algorithm.

The Fitch algorithm consists of two phases. The first phase is recursive and reconstructs possible ancestral labels $L(x)$ for each node x of T and the overall minimum number of label changes required as follows: For each node x of T in a bottom-up traversal, (1) if x is a leaf, then $L(x)=\{\tilde{b}(x)\}$ and $cost(T[x])=0$. (2) Else, let x_l and x_r be the children of x. If $L(x_l)\cap L(x_r)=\emptyset$, then $L(x)=L(x_l)\cup L(x_r)$ and $cost(T[x])=cost(T[x_l])+cost(T[x_r])+1$; else $L(x)=L(x_l)\cap L(x_r)$ and $cost(T[x])=cost(T[x_l])+cost(T[x_r])$. The second phase of the algorithm reconstructs an assignment $\tilde{b}$ of T that has a minimum cost, by computing $\tilde{b}(x)$ as follows: For each node x of T in a top-down traversal, (1) if x is the root, assign $\tilde{b}(x)$ to any label in $L(x)$. (2) Else, let x_p be the parent of x. If $\tilde{b}(x_p)\in L(x)$, then assign $\tilde{b}(x)=\tilde{b}(x_p)$, else assign $\tilde{b}(x)$ to any label in $L(x)$.

The Fitch algorithm does not always find an optimal $\tilde{b}$ assignment because of duplications that can be turned into EGTr events. Algorithm 2 modifies the first phase of the Fitch algorithm to compute the DEL-Distance and an assignment $\tilde{b}$ of T that leads to the DEL-Distance. The modification reflects the fact that a duplication node is allowed a "free" change since it can be turned into an EGTr node (see Figure 2 for an illustration).

Algorithm 2 $MinDEL$

Input: T,S,s,b; *Output:* $DEL(T,S),\tilde{b}$.

```
1: Let ⟨R_DL, s̃_DL, ẽ_DL⟩ be the reconciliation satisfying the DL-Distance;
2: for each node x of T in a bottom-up traversal do
3:   if x is a leaf then
4:     L(x) = {b(x)};
5:     DEL(T[x], S) = 0
6:   else
7:     Let x_l and x_r be the children of x;
8:     if ẽ_DL(x) = Dup then
9:       L(x) = L(x_l) ∪ L(x_r);
10:      DEL(T[x], S) = DEL(T[x_l], S) + DEL(T[x_r], S) + 1
11:    else if L(x_l) ∩ L(x_r) = ∅ then
12:      L(x) = L(x_l) ∪ L(x_r);
13:      DEL(T[x], S) = DEL(T[x_l], S) + DEL(T[x_r], S) + 1
14:    else
15:      L(x) = L(x_l) ∩ L(x_r);
16:      DEL(T[x], S) = DEL(T[x_l], S) + DEL(T[x_r], S)
17: for each node x of T in a top-down traversal do
18:   if x is the root then
19:     b̃(x) = any label in L(x);
20:   else
21:     Let x_p be the parent of x;
22:     if b̃(x_p) ∈ L(x) then
23:       b̃(x) = b̃(x_p);
24:     else
25:       b̃(x) = any label in L(x);
```

Lemma 5. *Algorithm 2 outputs, in linear time, the DEL-Distance* $DEL(T,S)$ *and a binary assignment* $\tilde{b}$ *of* T *that leads to a most parsimonious DEL-Reconciliation.*

Proof. It suffices to prove that the following statement holds for any node x of T: for any label β in $L(x)$, there exists a binary assignment $\tilde{b}$ of $T[x]$ such that $\tilde{b}(x)=\beta$ and $\tilde{b}$ minimizes $cost(T[x],S,\tilde{b})$.

1. If x is a leaf (Lines 3-5), then $L(x)=\{b(x)\}$. For $\tilde{b}(x)=b(x)$, $cost(T[x],S,\tilde{b})=0$.
2. If x is not a leaf (Lines 6-20). Let x_l and x_r be the children of x, and assume that the statement holds for x_l and x_r. Let $\beta\in L(x)$. Let $\tilde{b}_l$ and $\tilde{b}_r$ be two binary assignments of $T[x_l]$ and $T[x_r]$ that minimize $cost(T[x_l],S,\tilde{b}_l)$ and $cost(T[x_l],S,\tilde{b}_r)$, respectively, and such that $\tilde{b}_l(x_l)=\beta$ if $\beta\in L(x_l)$ and $\tilde{b}_r(x_r)=\beta$ if $\beta\in L(x_r)$. Let $\tilde{b}$ be the binary assignment of $T[x]$ obtained by merging $\tilde{b}_l$ and $\tilde{b}_r$ and extending it with $\tilde{b}(x)=\beta$.

   a. If x is a duplication node in the DL-reconciliation (Lines 8-10), then $L(x)=L(x_l)\cup L(x_r)$.

      (1) If $\beta\in L(x_l)\cap L(x_r)$, then $\tilde{b}(x_l)=\tilde{b}(x_r)=\tilde{b}(x)=\beta$, and $\Delta_{\tilde{b}}(x)=0$. Thus $cost(T[x],S,\tilde{b})=cost(T[x_l],S,\tilde{b}_l)+cost(T[x_r],S,\tilde{b}_r)$, without any increment.

      (2) If $\beta\notin L(x_l)\cap L(x_r)$, then $\beta\in L(x_l)$ or $\beta\in L(x_r)$, and $\tilde{b}(x_l)=\tilde{b}(x)=\beta$ or $\tilde{b}(x_r)=\tilde{b}(x)=\beta$, and $\Delta_{\tilde{b}}(x)=0$. Thus $cost(T[x],S,\tilde{b})=cost(T[x_l],S,\tilde{b}_l)+cost(T[x_r],S,\tilde{b}_r)$, without any increment.

      In both cases, Algorithm 1 computes a DEL-Reconciliation with minimum cost $DEL(T[x_l],S)+DEL(T[x_r],S)+1$ with a minimum increment of 1 for a Dup node in case (1), or by making x an EGTr node in case (2), but no additional EGTp node is required.

   b. If x is a speciation node in the DL-reconciliation.

      (1) If $L(x)\neq L(x_l)\cap L(x_r)$, then $L(x_l)\cap L(x_r)=\emptyset$, and $\beta\in L(x_l)$ or $\beta\in L(x_r)$. So $\tilde{b}(x_l)=\tilde{b}(x)=\beta$ or $\tilde{b}(x_r)=\tilde{b}(x)=\beta$, and $\Delta_{\tilde{b}}(x)=1$. Thus $cost(T[x],S,\tilde{b})=cost(T[x_l],S,\tilde{b}_l)+cost(T[x_r],S,\tilde{b}_r)+1$,g with a minimum increment of 1 by grafting an EGTp node on one of the (x,x_l) or (x,x_r) branches. In this case, Algorithm 1 computes a DEL-Reconciliation with minimum cost $DEL(T[x_l],S)+DEL(T[x_r],S)+1$.

      (2) If $L(x)=L(x_l)\cap L(x_r)$, then $\beta\in L(x_l)$ and $\beta\in L(x_r)$. So $\tilde{b}(x_l)=\tilde{b}(x_r)=\tilde{b}(x)=\beta$, and $\Delta_{\tilde{b}}(x)=0$. Thus $cost(T[x],S,\tilde{b})=cost(T[x_l],S,\tilde{b}_l)+cost(T[x_r],S,\tilde{b}_r)$ without any additional cost. Algorithm 1 computes a DEL-Reconciliation with minimum cost $DEL(T[x_l],S)+DEL(T[x_r],S)$ when given $\tilde{b}$.

It is easy to see that both the first and the second phases of the algorithm have linear time complexity, thus the overall algorithm has a linear time complexity. □

picture(0,0)(-35,0)(1,0)30 (0,-35)(0,1)30 picture

picture(0,0)(35,0)(-1,0)30 (0,-35)(0,1)30 picture

As for the Fitch Algorithm, Algorithm 2 does not allow to output all the solutions of the DEL-Reconciliation problem leading to the DEL-Distance. However, this can be achieved by adapting the Sankoff and Cedergren's dynamic programming algorithm. Rather, we choose to introduce, in the next section, a more general dynamic programming algorithm allowing to output all optimal solutions for an arbitrary cost of the DEL events, not only for the unitary cost.

4 Solving the DEL-Reconciliation Problem with arbitrary DEL costs

In this section, we assume that each possible event has a different cost. We use δ and λ to denote the cost of a duplication and a loss, respectively. We use ρ_0 (resp. τ_0) for the cost of a transposition (resp. transfer) from the mitochondrial genome to the nuclear genome, and ρ_1 (resp. τ_1) for the cost of a transposition (resp. transfer) from the nuclear genome to the mitochondrial genome. Note that the subscripts of the EGT costs indicate the source of the switch.

Also denote

$$\rho_0^* = \min(\rho_0, \tau_0 + \lambda) \qquad \rho_1^* = \min(\rho_1, \tau_1 + \lambda)$$

Roughly speaking, ρ_0^* represents the minimum cost required to switch from mitochondrial to nuclear genome inside a branch of T, and ρ_1^* the minimum cost required in the other direction. The purpose of ρ_0^* and ρ_1^* is that a switch can be accomplished by a transposition event, but also by a transfer event followed by a loss.

Let $x \in V(T)$ and let $b_x \in \{0, 1\}$. We denote by $D[x, b_x]$ the minimum cost of a DEL-Reconciliation $\langle R, \tilde{s}, \tilde{b}, \tilde{e}\rangle$ of $T[x]$ with S in which $\tilde{b}(x) = b_x$ (or ∞ if no such reconciliation exists). Note that $\tilde{s}(x)$ does not need to be inferred, since by Lemma 1, we may assume that $\tilde{s}(x) = lca_S(s(\mathcal{L}(T[x])))$.

Trivially, if x is a leaf of T, we have

$$D[x, b_x] = \begin{cases} 0 & \text{if } b_x = b(x) \\ \infty & \text{otherwise} \end{cases}$$

Assume now that x is an internal node of T. Let x_l, x_r be the children of x. For $s_1, s_2 \in V(S)$, let $path(s_1, s_2)$ denote the number of vertices on the path between s_1 and s_2 in S, *including* s_1 and s_2. Then define

$$l_x = path(\tilde{s}(x), \tilde{s}(x_l)) + path(\tilde{s}(x), \tilde{s}(x_r))$$

which counts the number of mandatory losses on the child branches of a node x of T.

To compute $D[x, b_x]$, we use three auxiliary values $D[x, b_x, e_x]$, where $e_x \in \{Spe, Dup, EGTr\}$ represents the event label of x (note that e_x cannot be a transposition event, since x has two children).

If $\tilde{s}(x) = \tilde{s}(x_l)$ or $\tilde{s}(x) = \tilde{s}(x_r)$, then $D[x, b_x, Spe] = \infty$. Assuming this check has been performed, we have

$$D[x, b_x, Spe] = \lambda(l_x - 4) + \sum_{x' \in \{x_l, x_r\}} \min(D[x', b_x], \rho_{b_x}^* + D[x', 1 - b_x])$$

$$D[x, b_x, Dup] = \delta + \lambda(l_x - 2) + \sum_{x' \in \{x_l, x_r\}} \min(D[x', b_x], \rho_{b_x}^* + D[x', 1 - b_x])$$

$$D[x, b_x, EGTr] = \tau_{b_x} + \lambda(l_x - 2) + \min \begin{cases} D[x_l, b_x] + D[x_r, 1 - b_x] \\ D[x_l, 1 - b_x] + D[x_r, b_x] \\ \rho_{1-b_x}^* + D[x_l, b_x] + D[x_r, b_x] \\ \rho_{b_x}^* + D[x_l, 1 - b_x] + D[x_r, 1 - b_x] \end{cases}$$

Put $D[x, b_x] = \min(D[x, b_x, Spe], D[x, b_x, Dup], D[x, b_x, EGTr])$. The value of interest is $\min(D[r(T), 0], D[r(T), 1])$.

Theorem 1. *For any $x \in V(T)$ and $b_x \in \{0, 1\}$, the value of $D[x, b_x]$, as defined above, is equal to the minimum cost of a DEL-Reconciliation $\langle R, \tilde{s}, \tilde{b}, \tilde{e}\rangle$ of $T[x]$ with S satisfying $\tilde{b}(x) = b_x$.*

Moreover, the minimum cost $\min(D[r(T), 0], D[r(T), 1])$ of a reconciliation of T with S can be computed in time $O(|V(T)| + |V(S)|)$.

Let us note that once the D table is computed, a standard backtracking procedure allow to reconstruct every optimal DEL-Reconciliation.

5 Experimental results

We implemented the above dynamic programming procedure in python in a software called EndoRex, which supports arbitrary costs as input and returns a reconciled gene tree in Newick format. The python source can be accessed at https://github.com/AEVO-lab/EndoRex. We then performed a variety of experiments on a dataset obtained from [14], as described in the following.

5.1 Kannan et al., (2014) dataset

For the reconstruction of evolutionary histories with EGT events, we used a dataset from Kannan et al., 2014 [14] available at ftp://ftp.ncbi.nih.gov/pub/koonin/MitoCOGs. The dataset consists of 140 clusters of orthologous mitochondrial protein-coding genes (MitoCOGs) extended with paralogs and nuclear protein-coding homologs from 2,486 eukaryotes with complete mitochondrial genomes. Statistics on MitoCOGs of the Kannan *et al.* dataset are described in Table 1.

Table 1. Statistics of the Kannan *et al.*, 2014 dataset.

Gene set	Nb of MitoCOGs	Nb of species	Nb of genes
Mitochondrial-encoded	140	2,486	34,755
Nuclear-encoded	45	52	1,317
Whole set	140	2,486	36,072

5.2 Dataset preprocessing

Among these 140 MitoCOGs of the initial Kannan *et al.* dataset, we first selected the 45 clusters with nuclear-encoded protein sequences. Within

these MitoCOGs, 52 eukaryotes are represented including 28 *Opisthokonta* (10 *Fungi*, 17 *Metazoa* and 1 *Choanoflagellata*), 9 *Viridiplantae*, 1 *Rodophyta*, 1 *Glaucophyta*), 5 *Alveolata*, 1 *Amoebozoa*, 2 *Euglenozoa*, 1 *Heterolobosea*, 1 *Rhizaria* and 3 *Stramenopiles*. Based on the Figure 1 of [14] and analysis of the dataset, we selected the 11 plants species clade, including the 9 *Viridiplantae*, *Cyanidioschyzon merolae* (*Rodophyta*) and *Cyanophora paradoxa* (*Glaucophyta*), for EGT evolutionary history inference with EndoRex as gene-content location is more diversified among this species group.

The 11 plants species are represented in 68 MitoCOGs with mitochondrial-encoded proteins and 41 MitoCOGs with nuclear-encoded proteins. We selected the clusters for which there were mitochondrial and nuclear encoded genes, yielding 28 MitoCOGS containing 326 protein-coding genes, including 184 encoded in the mitochondria and 142 in the nucleus.

Table 2 gives information about the 28 MitoCOGs of the 11 plants dataset specifying the gene name, the protein metabolic pathway, and the number of genes and species for each MitoCOG.

Table 2. Statistics on the 28 MitoCOGs of the 11 plants dataset. For the "Nb of gene" column, the number of mitochondria-encoded (mito) and nucleus-encoded (nuc) gene are specified.

MitoCOG ID	Gene name	Metabolic pathway	Nb of genes (mito+nuc)	Nb of species
MitoCOG0006	nad3	Complex I	11 (10+1)	11
MitoCOG0007	nad4L	Complex I	13 (12+1)	11
MitoCOG0031	nad7	Complex I	11 (9+2)	11
MitoCOG0043	nad9	Complex I	11 (9+2)	11
MitoCOG0029	nad10	Complex I	13 (1+12)	10
MitoCOG0052	sdh2	Complex II	22 (1+21)	10
MitoCOG0051	sdh3	Complex II	8 (3+5)	6
MitoCOG0075	sdh4	Complex II	9 (4+5)	9
MitoCOG0003	cox2	Complex IV	13 (10+3)	11
MitoCOG0005	cox3	Complex IV	13 (10+3)	11
MitoCOG0059	atp1	Complex V	9 (7+2)	8
MitoCOG0076	atp4	Complex V	12 (11+1)	10
MitoCOG0004	atp6	Complex V	13 (12+1)	11
MitoCOG0014	atp9	Complex V	13 (10+3)	11
MitoCOG0027	rpl2	Translation	14 (5+9)	10
MitoCOG0053	rpl6	Translation	10 (4+6)	8
MitoCOG0092	rpl10	Translation	5 (2+3)	5
MitoCOG0048	rpl14	Translation	15 (5+10)	11
MitoCOG0039	rpl16	Translation	12 (8+4)	11
MitoCOG0070	rpl20	Translation	11 (2+9)	8
MitoCOG0080	rps2	Translation	9 (5+4)	9
MitoCOG0067	rps4	Translation	8 (7+1)	7
MitoCOG0061	rps7	Translation	12 (8+4)	11
MitoCOG0072	rps10	Translation	12 (3+9)	8
MitoCOG0054	rps11	Translation	12 (6+6)	10
MitoCOG0064	rps13	Translation	10 (7+3)	10
MitoCOG0055	rps14	Translation	9 (5+4)	8
MitoCOG0026	rps19	Translation	16 (8+8)	8

For each MitoCOG, we applied a pipeline to infer the evolutionary history of EGTs with DEL-Reconciliation along the 11 plants species tree. Topology of the species tree was taken from [14]. We added the species *Micromonas sp. RCC299* as the sister species of *Ostreococcus tauri* as only these 2 among the 11 plants species belong to the *Mamiellophyceae* class (Figure 3).

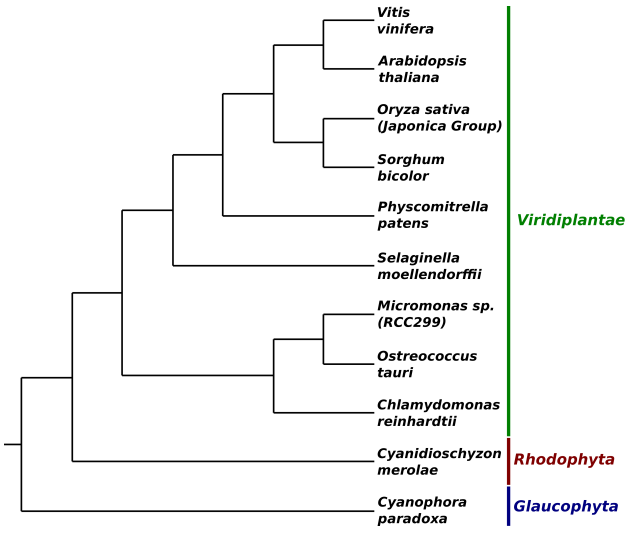

Fig. 3. Species tree of the 11 plants considered in our experimental analysis. Topology of the tree is based on [14].

As for constructing gene trees, the first step of the pipeline was to align the protein sequences with MUSCLE [16]. In the second step, a maximum likelihood protein tree was infered using RAxML (v8.2.4) with the PROTGAMMAAUTO option that automatically choose the best fitting amino acid substitution model [17]. NOTUNG (v.2.9.1.5) was then used to root the trees by minimizing the cost of a duplication-loss reconciliation with default parameter (loss cost: 1.0 and duplication cost: 1.5) [18].

The rooted protein trees obtained with this pipeline and the 11 plants species tree were given as input of the EndoRex software to infer a most parsimonious DEL-Reconciliation allowing for arbitrary costs for duplications, losses and EGTs.

5.3 EndoRex evolutionary events cost setting

As a reminder, we consider six parameters corresponding to the different evolutionary event costs: δ and λ the cost of, respectively, a gene duplication and loss; ρ_0 (resp. τ_0) the cost of a transposition (resp. transfer) from the mitochondrial genome to the nuclear genome, and ρ_1 (resp. τ_1) the cost of a transposition (resp. transfer) from the nuclear genome to the mitochondrial genome.

We test six different costs settings for the application of EndoRex on the 11 plants dataset. The setting $S1$ corresponds to the default values for parameters, with a unitary cost for evolutionary events (allowing to compute the DEL-Distance). For setting $S2$, the gene loss and duplication costs are those used in NOTUNG for rooting the protein trees, and transfers and transpositions are given the same cost as losses: $\lambda = \rho_0 = \rho_1 = \tau_0 = \tau_1 = 1.0$ and $\delta = 1.5$. For setting $S3$, transposition and transfer costs are increased to reflect the fact that these evolutionary events are less frequent than gene duplications: $\lambda = 1.0$, $\delta = 1.5$ and $\rho_0 = \rho_1 = \tau_0 = \tau_1 = 2.0$. In setting $S4$, we consider transfers as less frequent than transpositions: $\lambda = 1.0$, $\delta = 1.5$, $\rho_0 = \rho_1 = 2.0$ and $\tau_0 = \tau_1 = 3.0$. For setting $S5$, we differentiate the cost of the mitochondria to the nucleus from the nucleus to the mitochondria gene move. It is well established that mitochondrial genes are integrated into the nuclear genome, and the reversal process is an extremely rare event: $\lambda = 1.0$, $\delta = 1.5$, $\rho_0 = 2.0$, $\rho_1 = 3.0$, $\tau_0 = 3.0$ and $\tau_1 = 4.0$. Then, setting $S6$ is as setting $S5$ except we make no difference between transfer and transposition costs: $\lambda = 1.0$, $\delta = 1.5$, $\rho_0 = 2.0$, $\rho_1 = 3.0$, $\tau_0 = 2.0$ and $\tau_1 = 3.0$.

Applied to the 28 MitoCOGs trees, EndoRex infer the same DEL-Reconciliation with the six different settings for the five MitoCOGs: MitoCOG0029, MitoCOG0048, MitoCOG0067, MitoCOG0080 and MitoCOG0092.

We first observed that setting $S2$ is the parameterization leading to the most different DEL-Reconciliation solutions compared to the five other settings. Actually, this cost setting is not realistic as EGTs are much less frequent than gene duplications and should logically be penalized more. Therefore, from now, we discard this setting from the analysis. Comparison of the five remaining settings yields a unique DEL-Reconciliation for 21 of the 28 MitoCOGs.

Among the seven MitoCOGs with different DEL-Reconciliations depending on the considered setting (among the five left), six of them (MitoCOG0005, MitoCOG0014, MitoCOG0027, MitoCOG0051, MitoCOG0053, MitoCOG0072) lead to two different DEL-Reconciliations and one (MitoCOG0039) to three different DEL-Reconciliations.

We analysed the three DEL-Reconciliations of MitoCOG0039, corresponding to the *rpl16* gene, to illustrate the dynamic of the score settings (see Figure 4). According to this case study, it seems that setting $S1$ is inappropriate as it leads to the prediction of numerous EGTs from the nucleus to the mitochondria, which is very unrealistic as a very few number of gene movements from the nucleus to the mitochondria have been described in the literature. DEL-Reconciliation predicted with setting $S5$ is the scenario most in line with the literature as it only infers EGTs from the mitochondria to the nucleus, with transpositions located close to the leaves of the tree, indicating an ongoing process of endosymbiotic gene transfer in plants for this gene family. The scenario inferred with settings $S3$, $S4$ and $S6$ is an intermediate between those inferred with setting $S1$ and $S5$. The main difference with scenario of $S1$ is the inference of an EGTp followed by a duplication instead of an EGTr for $S1$ at the lowest ancestor of *S. bicolor*, *O. sativa*, *V. vinifera* and *A. thaliana* replacing two EGTp from nucleus to mitochondria, in $S1$, by one EGTp from mitochondria to nucleus, in settings $S3$, $S4$ and $S6$. But in this DEL-Reconciliation, there is still one EGTp and one EGTr from the nucleus to the mitochondria that may be considered unlikely.

6 Conclusion

Investigating the origin, evolution and characteristics of gene coding capacity of eukaryotes has been among the central themes in the Life Sciences. In this context, the endosymbiotic origin of mitochondrial genomes and the gradual integration of the mitochondrial gene content to the nucleus are important evolutionary parameters expected to shed light on features of eukaryotic gene evolution and function.

From a computational point of view, detecting the footprint of endosymbiosis in the gene repertoires of the mitochondrial and nuclear genomes of eukaryotes require new evolutionary prediction methods. This paper is a first effort towards developing the appropriate algorithmic tools for analysing the movement of genes inside a gene family between the mitochondrial and nuclear genome of the same species. More precisely, we have introduced a new reconciliation model accounting for endosymbiotic gene transfer (EGT), a special case of horizontal gene transfer, with the purpose of reconstructing the evolutionary history of genes accounting for their movement between the mitochondria and nucleus. We presented a linear-time algorithm, based on the Duplication-Loss (DL) reconciliation distance and the Fitch algorithm for weighted parsimony, to compute a most parsimonious history of Duplication, EGT and Loss (DEL) events explaining a gene tree with leaves identified as mitochondrial or nuclear genes. We also presented a general dynamic programming algorithm, implemented in the EndoRex software, to compute all optimal DEL-Reconciliations for any arbitrary cost scheme of operations.

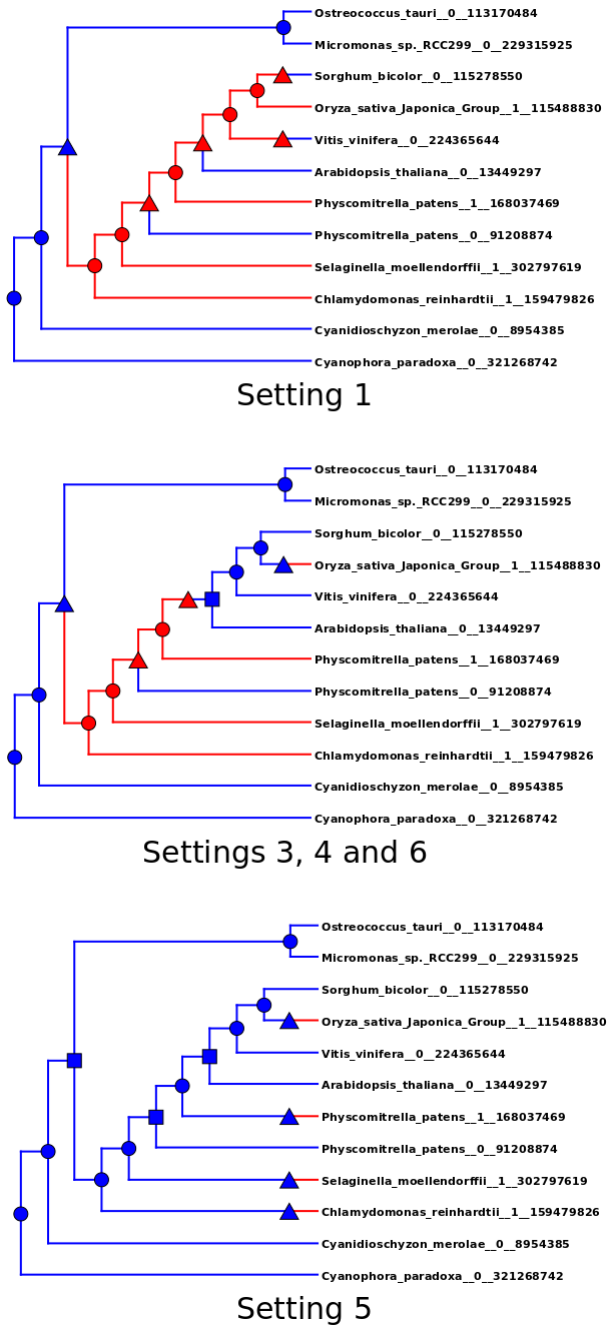

Fig. 4. DEL-Reconciliations obtained for MitoCOG0039 with the EndoRex scores settings $S1$, $S3$, $S4$, $S5$ and $S6$. The blue part of the tree indicates that the genetic material is located in the mitochondria, while the red part indicates location in the nucleus. Shapes at the internal nodes represent the evolutionary event associated according to the Figure 1. Gene are formatted as follow: [species_name]__[gene-encoding location]__[gene id]. With gene-encoding location annotation: 0→mitochondria and 1→nucleus. Loss events are not represented.

By applying EndoRex to a plants dataset, we showed that it is well-designed to infer the evolutionary histories of EGT events, considering a variety of cost settings. Some reconciled trees (not shown) of the 11 plants dataset produced evolutionary histories that could be considered unrealistic as leading to an unexpected high number of gene duplications and losses. As our algorithm is exact and thus guaranteed to infer the minimum number of events given a gene tree, this is likely due to errors

picture(0,0)(-35,0)(1,0)30 (0,35)(0,-1)30 picture

picture(0,0)(35,0)(-1,0)30 (0,35)(0,-1)30 picture

in protein sequence alignment and/or gene tree inference, leading to erroneous gene trees [24]. A better gene tree inference pipeline should be designed to get more accurate gene trees, probably allowing for more realistic DEL-Reconciliations.

Future applications will concern a thorough analysis of protein-coding genes involved in common metabolic pathways. As an example, the oxydative phophorylation (OXPHOS) is a series of protein complexes (I, II, III, IV and V) leading to an electrochemical proton gradient activating the ATP synthase (complex V) that produces ATP, an important molecule involved in the intracellular energy tranfer. These protein-coding genes involved in OXPHOS are expected to share common mitochondrial-nuclear movement, as nucleus and mitochondria are two compartments with different biological dynamics.

The purpose of inferring common episodes of EGT events for a set of gene families opens new avenues towards algorithmic extensions seeking for a most parsimonious joint DEL-Reconciliation of multiple gene trees with a species tree. This may be handled by generalizing the Super-Reconciliation [23] model to account for segmental DEL events.

Finally, the recent sequencing effort conducted towards jakobids and malawimonads protists genomes known to have emerged close to the eukaryotic origin will provide a valuable dataset that can be analysed with the new developed algorithms, helping to shed light on a number of important biological questions, among them resolving the root of the eukaryote tree. In fact, as EGTs are rare events, candidate topologies for which DEL-Reconciliations infer the lowest number of EGT events, may provide evidence of a correct rooting.

Acknowledgements

We thank B. Franz Lang (Biochemistry Department, University of Montreal) for his insights and clever advices on the algorithmic needs and open questions regarding eukaryotes' evolution.

References

[1]Dyall, S.D. and Johnson, P.J. (2000) Origins of hydrogenosomes and mitochondria: evolution and organelle biogenesis. *Curr Opin Microbiol*, **3**(4), 404-411.

[2]Roger, A.J., Munoz-Gomez, S.A. and Kamikawa, R. (2017) The Origin and Diversification of Mitochondria, *Current Biology*, **27**, R1177–R1192. doi:10.1016/j.cub.2017.09.015.

[3]Sloan, D.B., Warren, J.M., Williams, A.M., Wu, Z, Abdel-Ghany, S.E.,Chicco, A.J. and Havird, J.C. (2018) Cytonuclear integration and co-evolution, *Nature Reviews Genetics*, **19**, 635-648. doi:10.1038/s41576-018-0035-9.

[4]Lang B.F. and Burger, G. (2012) Mitochondrial and Eukaryotic Origins: A Critical Review, *Botanical Research*, Elsevier. **63**, 1-20.

[5]Gray, M.W., Burger, G., Derelle, R., Klimes, V., Leger, M.M, Sarrasin, M., Vlcek C., Roger A.J, Elias M. and Lang B.F. (2020) The draft nuclear genome sequence and predicted mitochondrial proteome of *Andalucia godoyi*, a protist with the most gene-rich and bacteria-like mitochondrial genome, *BMC Biology*, **18**(22).

[6]Derelle, R. *et al*. (2015) Bacterial proteins pinpoint a single eukaryotic root, *Proc Natl Acad Sci U S A*, **112**(7), E693-E699.

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

[8]El-Mabrouk, N. and Noutahi, E. (2019) Gene Family Evolution—An Algorithmic Framework. In Warnow, T. (ed.), *B*ioinformatics and Phylogenetics: Seminal Contributions of Bernard Moret, Springer International Publishing, 87–119.

[9]Akerborg, O., Sennblad, B., Arvestad, L. and Lagergren, J. (2009) Simultaneous Bayesian gene tree reconstruction and reconciliation analysis, *Proceedings of the National Academy of Sciences*, **106**(Num. 14), 5714–5719. doi:10.1073/pnas.0806251106.

[10]Szollosi, G.J., Tannier, E., Daubin, V. and Boussau, B. (2015) The Inference of Gene Trees with Species Trees, *Systematic Biology*, **64**(Num.1), e42–e62. doi:10.1093/sysbio/syu048.

[11]Zmasek, C.M. and Eddy, S.R. (2001) A simple algorithm to infer gene duplication and speciation events on a gene tree., *Bioinformatics*, **17**(Num.9), 821–828. doi:10.1093/bioinformatics/17.9.821.

[12]Bansal, M.S., Alm, E.J. and Kellis, M. (2012) Efficient algorithms for the reconciliation problem with gene duplication, horizontal transfer and loss, *Bioinformatics*, **28**, 283–291. doi:10.1093/bioinformatics/bts225.

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

[14]Kannan, S., Rogozin, I. and Koonin, E. (2014) MitoCOGs: clusters of orthologous genes from mitochondria and implications for the evolution of eukaryotes, *BMC Evolutionary Biology*, **14**(Num. 11), 1–16. doi:10.1186/s12862-014-0237-5

[15]Kristensen, D., Kannan, L., Coleman, M., Wolf, Y., Sorokin, A., Koonin, E. and Mushegian, A. (2010) A low-polynomial algorithm for assembling clusters of orthologous groups from intergenomic symmetric best matches, *Bioinformatics*, **26**(Num. 12), 1481–1487. doi:10.1093/bioinformatics/btq229

[16]Edgar, R. (2004) MUSCLE: Multiple sequence alignment with high accuracy and high throughput, *Nucleic Acids Research*, **32**(Num. 5), 1792–1797. doi:10.1093/nar/gkh340

[17]Stamatakis, A. (2014) RAxML version 8: A tool for phylogenetic analysis and post-analysis of large phylogenies, *Bioinformatics*, **30**(Num. 9), 1312–1313. doi:10.1093/bioinformatics/btu033

[18]Stolzer, M., Lai, Ha., Xu, M., Sathaye, D., Vernot, B. and Durand, D. (2012) Inferring duplications, losses, transfers and incomplete lineage sorting with nonbinary species trees, *Bioinformatics*, **28**(Num. 18), 409–415. doi:10.1093/bioinformatics/bts386

[19]Fitch, W.A. (1971) Minimum Change for a Specific Tree Topology, *Systematic Biology*, **20**(Num. 4), 406–416. doi:10.1093/bioinformatics/bts386

[20]Sankoff, D. and Cedergren, R.J. (1983) Simultaneous comparison of three or more sequences related by a tree. In Sankoff, D. and Kruskal, J.B. (eds.), *T*ime warps, string edits and macromolecules: the theory and practice of sequence comparison, Addison-Wesley. Chapter 9, 253–264.

[21]Adams, K.L. and Palmer, J.D. (2003) Evolution of mitochondrial gene content: gene loss and transfer to the nucleus. *M*olecular Phylogenetics and Evolution, *Plant Molecular Evolution* **29**(Num. 3), 380–395. doi:10.1016/S1055-7903(03)00194-5

[22]Charleston, M.A. and Perkins, S.L. (2006) Traversing the tangle: Algorithms and applications for cophylogenetic studies. *J*ournal of Biomedical Informatics, **39**(Num. 1), 62–71. doi:10.1016/j.jbi.2005.08.006

[23]Delabre, M., El-Mabrouk, N., Huber, K.T., Lafond,M., Moulton, V., Noutahi, E. Sautie Castellanos, M. (2020) Evolution through segmental duplications and losses: a Super-Reconciliation approach. *A*lgorithms. Mol. Biol., **15**(Num. 12). doi:10.1186/s13015-020-00171-4

[24]Hahn, M.W. (2007) Bias in phylogenetic tree reconciliation methods: implications for vertebrate genome evolution *G*enome Biology, **8**(Num. 7). doi:10.1186/gb-2007-8-7-r141

picture(0,0)(-35,0)(1,0)30 (0,-35)(0,1)30 picture

picture(0,0)(35,0)(-1,0)30 (0,-35)(0,1)30 picture

Appendix

Proof of Lemma 1

Let $\langle R, \tilde{s}, \tilde{b}, \tilde{e}\rangle$ be a DEL-Reconciliation of minimum cost between T and S. Let λ be the cost of a loss event. Let us first make an observation. Let $v \in V(R)$ and let $l \in \mathcal{L}(R[v]) \cap \mathcal{L}(T)$, assuming that l exists. Let $P = (v = p_1, p_2, \ldots, p_k = l)$ be the path from v to l in R. It is easy to see from the definition of reconciliation that $\tilde{s}(v) = \tilde{s}(p_1), \tilde{s}(p_2), \ldots, \tilde{s}(p_k) = \tilde{s}(l)$ is a path of S, but with some vertices possibly being repeated (i.e. $\tilde{s}(p_i) = \tilde{s}(p_{i+1})$ is possible, but otherwise $\tilde{s}(p_{i+1})$ is a child of $\tilde{s}(p_i)$). It follows that $\tilde{s}(v)$ must be an ancestor of $s(l)$. Since v and l were chosen arbitrarily, we have that for any $v \in V(R)$, $\tilde{s}(v)$ is an ancestor of $s(l)$ for every leaf $l \in \mathcal{L}(R[v]) \cap \mathcal{L}(T)$.

Now suppose that, for some $x \in V(R) \cap V(T)$, $\tilde{s}(x) \neq lca_S(s(\mathcal{L}(T[x])))$. Moreover, choose x as a lowest node of $V(R) \cap V(T)$ with this property (i.e. $\tilde{s}(x') = lca_S(s(\mathcal{L}(T[x'])))$ for all descendants $x' \in V(R) \cap V(T)$ of x in R). Note that x is an internal node of T since $\tilde{s}(x) = s(x)$ for every leaf x of T.

As we argued, $\tilde{s}(x)$ is an ancestor of $s(l)$ for every leaf $l \in \mathcal{L}(T[x])$. Since $\tilde{s}(x) \neq lca_S(s(\mathcal{L}(T[x])))$, it follows that $\tilde{s}(x)$ is a strict ancestor of $lca_S(s(\mathcal{L}(T[x])))$. We first argue that x cannot be a speciation. Assume this is the case and let x'_l, x'_r be the children of x in R (but not necessarily in T). We use x_l and x_r to denote the children of x in T. By the definition of speciation, $\tilde{s}(x'_l)$ and $\tilde{s}(x'_r)$ are the two children of $\tilde{s}(x)$. Because $\tilde{s}(x)$ is a strict ancestor of $lca_S(s(\mathcal{L}(T[x])))$, only one of $\tilde{s}(x'_l)$ or $\tilde{s}(x'_r)$ has descendants in $\{s(l) : l \in \mathcal{L}(T[x])\}$. Assume without loss of generality that only $\tilde{s}(x'_l)$ has such descendants. But then, $\tilde{s}(x'_r)$ is not an ancestor of any member of $s(\mathcal{L}(T[x]))$. In particular, $\tilde{s}(x'_r)$ is not an ancestor of any member of $s(\mathcal{L}(R[x'_r]) \cap \mathcal{L}(T))$, and the latter is easily seen to be non-empty (this is because x'_r is an ancestor of x_r and $T[x_r]$ has leaves from T). As we argued before, this is not possible, since there should be a path from $\tilde{s}(x'_r)$ to any $s(l)$ with $l \in \mathcal{L}(T[x'_r]) \cap \mathcal{L}(T)$.

Assume that x is a duplication or transfer event (x cannot be a transposition event because it is binary). As before, let x_l and x_r be the children of x in T (but not necessarily in R). By the choice of x, $\tilde{s}(x_l) = lca_S(s(\mathcal{L}(T[x_l])))$ and $\tilde{s}(x_r) = lca_S(s(\mathcal{L}(T[x_r])))$. Thus $\tilde{s}(x)$ must be a strict ancestor of both $\tilde{s}(x_l)$ and $\tilde{s}(x_r)$. Let s' be the child of $\tilde{s}(x)$ that is on the path from $\tilde{s}(x)$ to $lca_S(s(\mathcal{L}(T[x])))$. We obtain an alternate reconciliation by modifying R to obtain another extension R' of T. We do not change any event labeling. We map x to s' and graft a loss in $\tilde{s}(x)$ on the edge between x and its parent in R (if any). In that manner, the parent of x in R still has a child mapped to $\tilde{s}(x)$ in R'. This increases the cost by λ, the cost of one loss.

Now let $x_1, x_2, \ldots, x_k$ be the nodes on the path from x to x_l in R (excluding x and x_l). Note that since x is a duplication or EGTr, $\tilde{s}(x) = \tilde{s}(x_1)$. Moreover, at most one node among $x_1, \ldots, x_k$ can be an EGTr or an EGTp, since there is no point in making more than one switch within an edge.

If present, we may assume without loss of generality that such an event occurs at x_k, the parent of x_l in R, since the timing of the switch does not affect the reconciliation cost. In this case, $\tilde{s}(x_k) = \tilde{s}(x_l) = lca_S(s(\mathcal{L}(T[x_l])))$. On the other hand, $\tilde{s}(x_1) = \tilde{s}(x) \neq lca_S(s(\mathcal{L}(T[x])))$. This implies that $x_1 \neq x_k$, and thus x_1 is not an EGTr or an EGTp. It follows that x_1 is a node inserted because of a grafted loss, and $\tilde{s}(x_2) = s'$. In R', we can remove x_1 and its loss leaf, and by doing so, the left child of x becomes x_2. This preserves all properties of a valid reconciliation because both x and x_2 are mapped to s'. We can apply the same procedure on the path from x to x_r.

In R', we have created one loss above x, but have removed two losses on both sides of x. No other event labeling has changed. Since we assume that losses have a non-zero cost, R' has a strictly lower cost than R, a contradiction. □

Proof of Lemma 2

We first show that the reconciliation $\langle R, \tilde{s}, \tilde{b}, \tilde{e}\rangle$ obtained from Algorithm 1 is a valid DEL-Reconciliation. Note that the tree R returned by the algorithm is the same as R_{DL}, but with some grafted unary nodes for EGTp events where needed. Consider some $x \in V(R_{DL})$. In R, we put $\tilde{e}(x) = Spe$ if $\tilde{e}_{DL}(x) = Spe$, and $\tilde{e}(x) \in \{Dup, ETTT\}$ if $\tilde{e}_{DL}(x) = Dup$. If no additional node was grafted as a new child of x, all properties of reconciliation would be preserved since we keep $\tilde{s}$ as in $\tilde{s}_{DL}$. If some node x' was grafted as a new child of x, we ensure that $\tilde{s}(x')$ is the same as the previous child of x, which ensures that we satisfy the properties of reconciliation. Therefore, we only need to check whether the tree R_{DL} is modified in an appropriate way in the case of a different $\tilde{b}$ value for a node x of T and one of its two children x_l or x_r.

Lines 2-8 first ensure that the starting tree R is such that, for each node x of T, $\tilde{b}(x) = \tilde{b}_T(x)$, and for any edge (x, y) in T such that $\tilde{b}_T(x) \neq \tilde{b}_T(y)$, the corresponding path $(x, v_1, v_2, \cdots v_n, y)$ on R is such that for all i, $\tilde{b}(v_i) = \tilde{b}(y)$. Subsequently, in the case of a different $\tilde{b}$ value for a node x of T and its child y, the node x is either modified to an EGTr node, ensuring that the switch between $\tilde{b}(x)$ and $\tilde{b}(v_1)$ is correctly explained by this EGTr, or a new EGTp node v is grafted on the edge (x, v_1), also correctly explaining the switch between $\tilde{b}(x)$ and $\tilde{b}(v_1)$.

We now show that the DEL-Reconciliation output by Algorithm 1 is of minimum cost. First Note that, from the initialization done in Line 8, for each leaf x which is on R_{DL} but not in T (lost gene), the algorithm ensures that $\tilde{b}(x) = \tilde{b}(p_x)$ were p_x is x's parent. Thus, grafted loss leaves never require an extra EGTr event on an "inserted edge" of R_{DL}.

Assume another reconciliation $\langle R', \tilde{s}', \tilde{b}', \tilde{e}'\rangle$ has a strictly lower cost than $\langle R, \tilde{s}, \tilde{b}, \tilde{e}\rangle$ output by Algorithm 1. We first show that, for any node of T, the corresponding node in R and R' have the same event label. Assume this is not the case. Let x be the lowest node of T such that $\tilde{e}'(x) \neq \tilde{e}(x)$. Let x_l and x_r be its two children in T and v_l and v_r be the two non-unary descendant of x in R' the closest from x. Note that x_l and x_r do not necessarily correspond to v_l and v_r in R'. Rather, they may be strict descendants of these nodes in R'.

1. If $\tilde{e}_{DL}(x) = Dup$, then from Algorithm 1, $\tilde{e}(x) = Dup$ if $\tilde{b}(x_l) = \tilde{b}(x)$ and $\tilde{b}(x_r) = \tilde{b}(x)$, and $\tilde{e}(x) = EGTr$ otherwise. As $\tilde{e}'(x) \neq \tilde{e}(x)$, we should have $\tilde{e}'(x) \in \{Spe, EGTr\}$ in the first case, or $\tilde{e}'(x) \in \{Spe, Dup\}$ in the second case.

   Assume $\tilde{e}'(x) = Spe$. From Lemma 1, as $\langle R', \tilde{s}', \tilde{b}', \tilde{e}'\rangle$ is a reconciliation of minimum cost, $\tilde{s}'(x) = lca_S(s(\mathcal{L}(T[x])))$, and as x is a speciation node in R', one of v_l and v_r should be mapped to $\tilde{s}(x)_l$ and the other to $\tilde{s}(x)_r$. Assume w.l.o.g. that $\tilde{s}'(v_l) = \tilde{s}(x)_l$ and $\tilde{s}'(v_r) = \tilde{s}(x)_r$. Now, as x is a duplication node in R_{DL}, then $\tilde{s}(x_l) = \tilde{s}(x)$ or $\tilde{s}(x_r) = \tilde{s}(x)$. Assume w.l.o.g. that $\tilde{s}(x_l) = \tilde{s}(x)$. As x_l is a node of the subtree of R' rooted at v_l, by definition of a reconciliation, $\tilde{s}'(x_l)$ should be a descendant of $\tilde{s}(v_l)$, which is not the case as $\tilde{s}'(v_l) = \tilde{s}(x)_l$ is rather a strict descendant of $\tilde{s}(x) = \tilde{s}(x_l) = \tilde{s}'(x_l)$. Therefore, x cannot be a speciation node in $\langle R', \tilde{s}', \tilde{b}', \tilde{e}'\rangle$. We deduce that $\tilde{e}'(x) \in \{Dup, EGTr\}$.

   Now assume that $\tilde{b}(x_l) \neq \tilde{b}(x)$ or $\tilde{b}(x_r) \neq \tilde{b}(x)$. In this case, the algorithm puts $\tilde{e}(x) = EGTr$ and, as x is not a speciation, it should be a duplication node in $\langle R', \tilde{s}', \tilde{b}', \tilde{e}'\rangle$. But then an a unary EGTp node v should be present in one of the two paths from x to x_l or from x to x_r in R', contradicting the fact that $\langle R', \tilde{s}', \tilde{b}', \tilde{e}'\rangle$ is a reconciliation of minimum cost, since labeling x as an EGTr node and removing v would reduce the cost of the reconciliation by one.

   Finally, assume that $\tilde{b}(x_l) = \tilde{b}(x)$ and $\tilde{b}(x_r) = \tilde{b}(x)$. In this case, the algorithm puts $\tilde{e}(x) = Dup$ and, as x is not a speciation, it should be an EGTr node in $\langle R', \tilde{s}', \tilde{b}', \tilde{e}'\rangle$, which induces, by definition of an EGTr event, that one of the two children y of x in R' is such that $\tilde{b}(y) \neq \tilde{b}(x)$. Now, as $\tilde{b}(x) = \tilde{b}(x_l) = \tilde{b}(x_r)$,

picture(0,0)(-35,0)(1,0)30 (0,35)(0,-1)30 picture

picture(0,0)(35,0)(-1,0)30 (0,35)(0,-1)30 picture

one unary EGTp node v should change the $\tilde{b}$ labeling of y to the $\tilde{b}$ labeling of its descendant in $\{x_l, x_r\}$. But then relabeling x as a duplication node would allow removing v and thus reducing the cost of the reconciliation by one, contradicting the fact that $\langle R', \tilde{s}', \tilde{b}', \tilde{e}' \rangle$ is a reconciliation of minimum cost.

2. If $\tilde{e}_{DL}(x) = Spe$, then from the properties of a DL-Reconciliation, we should have $\tilde{s}(x_l) \neq \tilde{s}(x)$ and $\tilde{s}(x_r) \neq \tilde{s}(x)$. From Algorithm 1, x remains a speciation node in $\langle R, \tilde{s}, \tilde{b}, \tilde{e} \rangle$.

   As $\tilde{e}'(x) \neq \tilde{e}(x)$, we should have $\tilde{e}'(x) = Dup$ or $\tilde{e}'(x) = EGTr$. In both cases, $\tilde{s}(v_l) = \tilde{s}(v_r) = s(x)$. This implies that $x_l \neq v_l$ and $x_r \neq v_r$, and thus v_l and v_r are grafted because of losses. Since R' uses the LCA-mapping by Lemma 1, we can remove v_l, v_r and their corresponding grafted loss leaves and make x a speciation, while preserving a valid reconciliation. This saves a cost of three (two losses and a Dup or $EGTr$ event). In the worst case, we had $\tilde{e}'(x) = EGTr$, in which case we can add an EGTp event on the appropriate branch to encorce the same switch.

   Thus replacing the Dup or EGTr label of x by a speciation reduces the cost of R' by at least two, contradicting the fact that R' is a reconciliation of minimum cost.

Since we have the same number of Dup and ETTr events as R', it remains to show that we cannot graft less nodes than those induced by Algorithm 1. The grafted nodes are either binary nodes corresponding to losses, or EGTp unary nodes. Suppose R' has less grafted nodes than R. Then there is an edge (x, y) in T such that the corresponding path $P'_{x,y} = (x, v'_1, v'_2, \cdots v'_{n'}, y)$ in R' is shorter than the corresponding path $P_{x,y} = (x, v_1, v_2, \cdots v_n, y)$ in R. We consider a lowest edge (x, y) of T verifying this condition, and we assume, without loss of generality, that $y = x_l$. Recall that by Lemma 1, $\tilde{s}(x) = \tilde{s}'(x)$ and $\tilde{s}(y) = \tilde{s}'(y)$.

- If $\tilde{e}_{DL}(x) = Dup$, then x is a duplication or an EGTr node in both R and R'. Then, by definition of a reconciliation, $\tilde{s}(v_1) = \tilde{s}(x)$. Moreover, from the fact that R is obtained from R_{DL}, Algorithm 1 leads to a path $P_{x,y}$ with as many nodes as the path from $\tilde{s}(x)$ to $\tilde{s}(x_l)$ in S if x is a duplication node, and an additional EGTp node if $b_T(x) \neq b_T(x_l) = b_T(x_r)$. Moreover, it is easy to see that the number of losses crafted on (x, y) must be equal to the number of nodes on the path from $\tilde{s}(x)$ and $\tilde{s}(y)$, excluding $\tilde{s}(y)$, either in R or R', and that the EGTp event added by the algorithm cannot be avoided. And thus, the path $P'_{x,y}$ should be at least as long as $P_{x,y}$, contradicting the hypothesis that $P'_{x,y}$ is shorter than $P_{x,y}$.
- If $\tilde{e}_{DL}(x) = Spe$, then x is a speciation node in both R and R'. Then, by definition of a reconciliation, $\tilde{s}(v_1) = \tilde{s}'(v_1) = \tilde{s}(x)_l$. Thus, from the fact that R is obtained from R_{DL}, Algorithm 1 leads to a path $P_{v_1,y}$ with as many nodes as the path from $\tilde{s}(x)_l$ to $\tilde{s}(x_l)$ in S, with an additional EGTp node if $\tilde{b}(x) \neq \tilde{b}(x_l)$. Moreover, it is easy to see that no other operation (Spe, Dup, RGT or EGTp) can allow making less losses or avoid the EGTp event. And thus, the path $P'_{v_1,y}$ should be at least as long as $P_{v_1,y}$, contradicting the hypothesis that $P'_{x,y}$ is shorter than $P_{x,y}$. □

Proof of Theorem 1

Let us first argue on the complexity of computing $D[x, b_x]$ for every $x \in V(T)$ and every $b_x \in \{0, 1\}$ (including $D[r(T), 0]$ and $D[r(T), 1]$), our values of interest). The LCA-mapping $\tilde{s}$ can be computed in time $O(|V(T)| + |V(S)|)$ using classical approaches from DL-reconciliation. We can compute $D[x, 0]$ and $D[x, 1]$ for every $x \in V(T)$ in a post-order traversal of T (because their value only depends on x_l and x_r), and thus there are $O(|V(T)|)$ values to compute. If we assume that if we have access to l_x for each x, it is clear from the recurrences that $D[x, b_x, Spe]$, $D[x, b_x, Dup]$ and $D[x, b_x, EGTr]$ can be computed in $O(1)$ time. To access l_x in time $O(1)$ for any x, we can preprocess S by labeling each $v \in V(S)$ by its depth (i.e. its distance to the root). Then, $path(\tilde{s}(x), \tilde{s}(x_l))$ is simply the difference in depth between $\tilde{s}(x)$ and $\tilde{s}(x_l)$ (because $\tilde{s}(x_l)$ must be a descendant of $\tilde{s}(x)$). This difference can be obtained in constant time, and it follows that l_x can be obtained in $O(1)$. Therefore, each $D[x, b_x]$ entry takes $O(1)$ time to compute. Including the time to compute the preprocessing and the LCA-mapping, the total time of the algorithm is $O(|V(T)| + |V(S)|)$.

Let us now argue that the algorithm is correct. Let $x \in V(T)$, let $b_x \in \{0, 1\}$, and let $\mathcal{R} = \langle R, \tilde{s}, \tilde{b}, \tilde{e} \rangle$ be a DEL-Reconciliation of minimum cost between $T[x]$ and S that satisfies $\tilde{b}(x) = b_x$. The proof is by induction on the height of $T[x]$. If x is a leaf, it is easy to see that $D[x, b_x]$ is correct. Assume that x is an internal node with children x_l and x_r. We may inductively assume that $D[x_l, b_l]$ and $D[x_r, b_r]$ are computed correctly for $b_l, b_r \in \{0, 1\}$.

In what follows, let $\mathcal{R}_l = \langle R_l, \tilde{s}_l, \tilde{b}_l, \tilde{e}_l \rangle$ be the reconciliation between $T[x_l]$ and S obtained by taking $R[x_l]$, and restricting $\tilde{s}, \tilde{b}$ and $\tilde{e}$ to $V(R[x_l])$. Similarly, let $\mathcal{R}_r$ be the reconciliation of $T[x_r]$ with S obtained by taking $R[x_r]$ and restricting $\tilde{s}, \tilde{b}$ and $\tilde{e}$ to $R[x_r]$.

We show two useful claims, the first being that these sub-reconciliations must be optimal with respect to their subtrees.

Claim 1.1. *$c(\mathcal{R}_l) = D[x_l, \tilde{b}(x_l)]$ and $c(\mathcal{R}_r) = D[x_r, \tilde{b}(x_r)]$.*

Proof. By induction and by the definition of D, we have $D[x_l, \tilde{b}(x_l)] \leq c(\mathcal{R}_l)$. Moreover, in $\mathcal{R}$ we may replace the $R[x_l]$ subtree by $\mathcal{R}_l$ (more precisely, replace $R[x_l]$ by R_l, and use $\tilde{s}_l, \tilde{b}_l$ and $\tilde{e}_l$ for the vertices of R_l). Since $\tilde{s}_l(x_l) = \tilde{s}(x_l)$ and $\tilde{b}_l(x_l) = \tilde{b}(x_l)$, all conditions of a valid reconciliation are met after such a replacement. Furthermore, no additional loss, transfer or transposition is required on the path between x to x_l. If $D[x_l, \tilde{b}(x_l)] < c(\mathcal{R})$ held, this transformation would yield a lower cost reconciliation and contradict the optimality of $\mathcal{R}$. Therefore, $D[x_l, b(x_l)] \geq c(\mathcal{R})$. It follows that $D[x_l, b(x_l)] = c(\mathcal{R}_l])$. By a symmetric argument, $D[x_r, \tilde{b}(x_r)] = c(\mathcal{R})$.

Claim 1.2. *If $\tilde{e}(x) = Spe$, then there are at least $l_x - 4$ losses grafted on the (x, x_l) and (x, x_r) branches, and otherwise, there are at least $l_x - 2$ such grafted losses.*

Proof. If $\tilde{e}(x) = Spe$, in R there must be a loss grafted on the (x, x_l) (resp. (x, x_r)) branch for each node of $path(\tilde{s}(x), \tilde{s}(x_l))$ (resp. $path(\tilde{s}(x), \tilde{s}(x_r))$), excluding $\tilde{s}(x)$ and $\tilde{s}(x_l)$ (resp. $\tilde{s}(x_r)$). The number of such losses is $l_x - 4$ and induce a cost of $\lambda(l_x - 4)$. If $\tilde{e}(x) \in \{Dup, Etrf\}$, the required losses are the same, except that we do not exclude x from both paths, and thus $l_x - 2$ losses are required for a cost of $\lambda(l_x - 2)$.

We now argue that $D[x, b_x] \leq c(\mathcal{R})$. First assume that $\tilde{e}(x) \in \{Spe, Dup\}$. We then consider the four possible $\tilde{b}$ labelings of x_l and x_r.

- If $\tilde{b}(x) = \tilde{b}(x_l) = \tilde{b}(x_r)$, then no cost other than the losses is required on the (x, x_l) and (x, x_r) branches. Thus using claims 1.1 and 1.2,

$$c(\mathcal{R}) \geq \begin{cases} \lambda(l_x - 4) + c(\mathcal{R}_l) + c(\mathcal{R}_r) & \text{if } \tilde{e}(x) = Spe \\ \delta + \lambda(l_x - 2) + c(\mathcal{R}_l) + c(\mathcal{R}_r) & \text{if } \tilde{e}(x) = Dup \end{cases}$$

$$= \begin{cases} \lambda(l_x - 4) + D[x_l, b_x] + D[x_r, b_x] & \text{if } \tilde{e}(x) = Spe \\ \delta + \lambda(l_x - 2) + D[x_l, b_x] + D[x_r, b_x] & \text{if } \tilde{e}(x) = Dup \end{cases}$$

  Since for both $\tilde{e}(x) \in \{Spe, Dup\}$, $D[x, b_x, \tilde{e}(x)]$ adds the losses, plus the minimum of $D[x', b_x]$ and $\rho^*_{b_x} + D[x', 1 - b_x]$ for each child $x' \in \{x_l, x_r\}$, we see that $D[x, b_x] \leq D[x, b_x, \tilde{e}(x)] \leq c(\mathcal{R})$.
- If $\tilde{b}(x) = \tilde{b}(x_l)$ and $\tilde{b}(x) = 1 - \tilde{b}(x_r)$, then no additional cost is required on the (x, x_l) branch, but a switch is required on (x, x_r).

picture(0,0)(-35,0)(1,0)30 (0,-35)(0,1)30 picture

picture(0,0)(35,0)(-1,0)30 (0,-35)(0,1)30 picture

The minimum possible cost of such a switch is $\rho^*_{b_x}$, and thus using the two claims as the previous case (we omit the step replacing $c(\mathcal{R}_l)$ by $D[x_l, b_x]$ and $c(\mathcal{R}_r)$ by $D[x_r, 1-b_x]$, which is implicit by claim 1.1), if $\tilde{e}(x) = Spe$, we have

$$c(\mathcal{R}) \geq \lambda(l_x - 4) + D[x_l, b_x] + \rho^*_{b_x} + D[x_r, 1 - b_x]$$

and if $\tilde{e}(x) = Dup$, we have

$$c(\mathcal{R}) \geq \delta + \lambda(l_x - 2) + D[x_l, b_x] + \rho^*_{b_x} + D[x_r, 1 - b_x]$$

Again, the above expressions are considered by the minimisation of $D[x, b_x, \tilde{e}(x)]$, and so $D[x, b_x] \leq D[x, b_x, \tilde{e}(x)] \leq c(\mathcal{R})$.

- If $\tilde{b}(x) = 1 - \tilde{b}(x_l)$ and $\tilde{b}(x) = \tilde{b}(x_r)$, this case is symmetric to the previous one.
- If $\tilde{b}(x) = 1 - \tilde{b}(x_l)$ and If $\tilde{b}(x) = 1 - \tilde{b}(x_r)$, then a switch with host b_x is needed on both branches (x, x_l) and (x, x_r). Thus, if $\tilde{e}(x) = Spe$, we have

$$c(\mathcal{R}) \geq \lambda(l_x - 4) + \rho^*_{b_x} + D[x_l, 1 - b_x] + \rho^*_{b_x} + D[x_r, 1 - b_x]$$

and if $\tilde{e}(x) = Dup$, we have

$$c(\mathcal{R}) \geq \delta + \lambda(l_x - 2) + \rho^*_{b_x} + D[x_l, 1 - b_x] + \rho^*_{b_x} + D[x_r, 1 - b_x]$$

Again, these are considered in $D[x, b_x, \tilde{e}(x)]$, and we get $D[x, b_x] \leq D[x, b_x, \tilde{e}(x)] \leq c(\mathcal{R})$.

In all cases, $D[x, b_x] \leq c(\mathcal{R})$. It remains to show that this holds for $\tilde{e}(x) = Etrf$. In this case, a cost of τ_{b_x} must be counted for the x node, plus the cost for $l_x - 2$ losses by claim 1.2. Next, we consider all values of $\tilde{b}(x_l)$ and $\tilde{b}(x_r)$.

- if $\tilde{b}(x_l) \neq \tilde{b}(x_r)$, then as we argued

$$\begin{aligned} c(\mathcal{R}) &\geq \tau_{b_x} + \lambda(l_x - 2) + c(\mathcal{R}_l) + c(\mathcal{R}_r) \\ &= \tau_{b_x} + \lambda(l_x - 2) + D[x_l, \tilde{b}(x_l)] + D[x_r, \tilde{b}(x_r)] \end{aligned}$$

The latter expression is among the expressions that $D[x, b_x, Etrf]$ minimizes and thus $D[x, b_x] \leq D[x, b_x, Etrf] \leq c(\mathcal{R})$.

- if $b_x = \tilde{b}(x_r) = \tilde{b}(x_r)$, then since x is an Etrf event, one of the (x, x_l) or (x, x_r) branches must switch to $1 - b_x$, then switch back to b_x, implying a transposition from $1 - b_x$ to b_x of cost $\rho^*_{1-b_x}$. In this situation,

$$c(\mathcal{R}) \geq \tau_{b_x} + \lambda(l_x - 2) + \rho^*_{1-b_x} + D[x_l, b_x] + D[x_r, b_x]$$

which is considered among the expressions minimized by $D[x, b_x, Etrf]$. Again, $D[x, b_x] \leq D[x, b_x, Etrf] \leq c(\mathcal{R})$.

- if $b_x = 1 - \tilde{b}(x_l) = 1 - \tilde{b}(x_r)$, then one of the (x, x_l) or (x, x_r) branches stays in b_x, and thus must switch to $1 - b_x$ for a cost of $\rho^*_{b_x}$. In this situation,

$$c(\mathcal{R}) \geq \tau_{b_x} + \lambda(l_x - 2) + \rho^*_{b_x} + D[x_l, b_x] + D[x_r, b_x]$$

which is considered among the expressions minimized by $D[x, b_x, Etrf]$. Again, $D[x, b_x] \leq D[x, b_x, Etrf] \leq c(\mathcal{R})$.

In every possible case, $D[x, b_x] \leq c(\mathcal{R})$.

We must now prove the complementary bound, i.e. that $D[x, b_x] \geq c(\mathcal{R})$. Let $e \in \{Spe, Dup, Etrf\}$ such that $D[x, b_x] = D[x, b_x, e]$. If $e = Spe$, the expression $D[x, b_x, Spe]$ corresponds to making x a speciation (which is possible since we check that neither of $\tilde{s}(x) = \tilde{s}(x_l)$ nor $\tilde{s}(x) = \tilde{s}(x_r)$ holds) and adding the minimum number of mandatory losses on (x, x_l) and (x, x_r). Let $b_l \in \{0, 1\}$ that minimizes $\min(D[x_l, b_x], \rho^*_{b_x} + D[x_l, 1 - b_x])$, and define b_r for x_r analogously. Thus consider the reconciliation $\mathcal{R}'$ in which x is a speciation, on which we graft the $l_x - 4$ mandatory losses on (x, x_l) and (x, x_r) and then, for each of b_l or b_r that differs from b_x, adds a transposition on the corresponding branch. Then, for $T[x_l]$ subtree, take an optimal reconciliation $\mathcal{R}_l$ for $T[x_l]$ and for the $T[x_r]$ subtree, take the optimal reconciliation $\mathcal{R}_r$ for $T[x_r]$. By induction, $\mathcal{R}_l$ and $\mathcal{R}_r$ are of costs $D[x_l, b_l]$ and $D[x_r, b_r]$ respectively. Since all optimal reconciliations use the LCA-mapping, such a reconciliation is valid and its cost is as defined in $D[x, b_x, Spe]$. It follows that $D[x, b_x, Spe] = c(\mathcal{R}') \geq c(\mathcal{R})$ (the latter inequality owing to the optimality of $\mathcal{R}$).

If $e = Dup$, the argument is exactly the same, except that to construct $\mathcal{R}'$, we make x a duplication and add $l_x - 2$ losses instead.

Finally, assume that $e = Etrf$. It is not hard to see that each expression that $D[x, b_x, Etrf]$ may choose when minimizing corresponds to a valid reconciliation. Indeed, consider the reconciliation $\mathcal{R}'$ where $\tilde{e}(x) = Etrf$ for a cost of τ_{b_x}. We add $l_x - 2$ mandatory losses on the (x, x_l) and (x, x_r) branches. Then, the first two cases of the minimization in $D[x, b_x, Etrf]$ correspond to having no additional switch needed, and hence we can use the optimal reconciliation for $T[x_l]$ and $T[x_r]$. The third case corresponds to having both x_l and x_r mapped to b_x, in which case we can chosse to apply the transfer on (x, x_l), but need need to switch back for a cost of $\rho^*_{1-b_x}$. The last case corresponds to having both x_l and x_r mapped to $1 - b_x$, in which case the transfer applies one switch, an we add a transposition for the other switch of cost $\rho^*_{b_x}$.

Since each possible case represents the cost of a valid reconciliation $\mathcal{R}'$, we get $D[x, b_x, Etrf] = c(\mathcal{R}') \geq c(\mathcal{R})$. Thus for every possible value of e, we have $D[x, b_x] = D[x, b_x, e] \geq c(\mathcal{R})$.

To conclude, the two complementary bounds show that $D[x, b_x] = c(\mathcal{R})$. □