

# Gene repertoire evolution minimizing episodes of gains and losses

Mathieu Gascon<sup>[0009–0003–8654–8375]</sup>, Mattéo Delabre<sup>[0000–0003–4561–683X]</sup>, and  
Nadia El-Mabrouk<sup>\*[0000–0002–5385–1015]</sup>

Département d’informatique et de recherche opérationnelle, Université de Montréal,  
Québec, Canada

mathieu.gascon.1@umontreal.ca, matteo.delabre@umontreal.ca,  
mabrouk@iro.umontreal.ca

**Abstract.** Given a tree topology and an assignment of character states to its leaves, the Small Parsimony Problem (SPP) consists in assigning character states to the internal nodes in a way maximizing a certain parsimony or probabilistic criterion. In the genome rearrangement field, tree leaves are permutations of gene sets, and the problem is to infer permutations at internal nodes minimizing a rearrangement distance. Almost all genome rearrangement models lead to intractable problems for the SPP. Considering only the numerical profiles on a phylogeny, the Count package (Csűrös, 2010) can be used to predict the size of gene families on internal nodes under a parsimony or probabilistic model. Here, we present a tractable version of the SPP: given a tree topology leaf-labeled by unordered gene sets, infer gene sets at internal nodes in a way minimizing the number of gain and loss episodes on the edges of the tree, while having a single gain point for each gene (i.e. under Dollo’s law). We show that the entire solution space is covered by testing four possible cases on each internal node’s content, leading to a linear-time dynamic programming algorithm for obtaining an optimal solution. We apply our *InOutParsimony* software to clusters of orthologous mitochondrial protein-coding genes (MitoCOGs) in both the mitochondrial and nuclear genomes of 11 land plant species. The results are discussed considering the Endosymbiotic Gene Transfer events shaping the mitochondria and nucleus contents and compared with Count’s returned numerical profiles.

**Keywords:** Genome evolution · Gain and loss · Gene clusters · Synteny · Small parsimony · Dollo parsimony · Dynamic programming

## Introduction

Understanding how a gene family, a set of genomic regions, or a set of related genomes have evolved from a common ancestor is an important step towards elucidating a large number of biological questions. If a rooted phylogenetic tree is already available for the considered biological units represented at the leaves

---

\* Corresponding author.

of the tree, the problem reduces to the Small Parsimony Problem (SPP) which consists in inferring the content of ancestral nodes in a way maximizing a certain parsimony (or minimizing a certain cost) or probabilistic criterion.

Inferring ancestral gene sequences by minimizing the alignment score on a tree leaf-labeled by the sequences of extant genes is a well-known SPP handled by the Fitch [14] and Sankoff [21] algorithms. On a genomic level, an algorithm implemented in the software package Count [8] has been developed to predict the size of whole gene families on internal nodes of a phylogenetic tree, under the Dollo parsimony [10], Wagner parsimony [7], or under a probabilistic model [9]. On the other hand, in the genome rearrangement field, biological units are permutations of gene sets. In this case, minimization can be done on the number of order disruptions (number of breakpoints), on rearrangement events remodeling the gene order (inversions, transpositions, translocations, etc), or on mathematical abstractions such as Double Cut-and-Join (DCJ) or Single-Cut-or-Join (SCJ) operations. Except for the latter case [13,19], almost all genome rearrangement models lead to intractable problems [5,12], even when restricted to the median problem [6,18,20,22]. The problem becomes even harder in the presence of gene duplicates [1,4].

Finally, if the biological units are homologous genomic regions containing genes, also called syntenies, evolving through content-modifying operations (such as gains, losses, duplications, transfers, etc), the problem is to infer a history minimizing such operations. In a previous set of papers [2,11,15], we presented a generalization of the gene/species reconciliation [16] and developed *Synesth* taking as input a species tree and a synteny tree (tree topology leaf-labeled by unordered gene sets representing homologous genomic regions), and giving as output one or many possible histories (gene sets and events at the internal nodes of the tree) minimizing a certain function (minimum cost, Pareto-optimal vectors of event counts, etc). However, to simplify the problem, gains were excluded from the optimization function. More precisely, contrary to losses, taking advantage of simultaneous gains of multiple genes to reduce the overall number of individual gain events was disregarded.

In this paper, we present a new and tractable version of the SPP restricting the evolutionary model to gains and losses of groups of genes. More precisely, the problem is to infer gene sets at internal nodes of a synteny tree in a way minimizing the number of gain and loss episodes on the edges of the tree, further considering an extension of Dollo’s law, i.e. requesting to have a single gain point for each gene. This is the first attempt to address the above mentioned shortcoming of *Synesth* by accounting for the gain of multiple genes as a single event. We show that the entire solution space is covered by testing four cases on each possible internal node’s content, leading to a linear-time dynamic programming algorithm for obtaining an optimal solution.

After setting our notations in the next section, the model is formally defined in Section 2 and the algorithm is described in Section 3. We then, in Section 4, apply the implemented software *InOutParsimony* to a dataset of mitochondrial-encoded protein families (called MitoCOGs) taken from [17], in order to study

the exchange of those genes between the mitochondrial and nuclear eukaryotic genomes. For this purpose, we selected 11 land plant species containing a diversified MitoCOG's content in both their nuclear and mitochondrial genomes. The results are discussed in light of the Endosymbiotic Gene Transfer events shaping those genomes and numerical profiles of ancestral genomes are compared with those obtained using Count with both parsimony and probabilistic models. These results illustrate the potential of our method to account for the simultaneous loss of entire biological complexes, but also reflect its limitations related to overly restrictive cost models. We conclude by suggesting avenues for future extensions<sup>1</sup>.

## 1 Notations

All trees considered in this paper are rooted. Given a tree  $T$ , we denote by  $r(T)$  its root, by  $V(T)$  its node set and by  $L(T) \subseteq V(T)$  its leafset. A node  $v'$  is an *ancestor* of  $v$  if  $v'$  is on the (inclusive) path between  $v$  and the root, and we then call  $v$  a *descendant* of  $v'$ . The ancestor–descendant relation is denoted  $\leq$  and forms a partial order on nodes, in which the root is minimal and the leaves are maximal. The node  $v' = p(v)$  immediately preceding  $v \neq r(T)$  on this path is the *parent* of  $v$ , and then  $v$  is a *child* of  $v'$ . The set of children of a node  $v$  is denoted by  $\text{ch}(v)$ ; if  $|\text{ch}(v)| = 1$ , then  $v$  is said to be *unary*; if  $|\text{ch}(v)| = 2$ , then it is said to be *binary* and, unless specified otherwise, we denote its children by  $v_\ell$  and  $v_r$  in an arbitrary order. A tree is *binary* if all its internal (non-leaf) nodes are binary.

For any node  $v$  of  $T$ , we denote by  $T_v$  the subtree of  $T$  rooted at  $v$ , i.e. obtained from  $T$  by removing all the nodes which are not descendant of  $v$ . We denote by  $E(T)$  the edge set of  $T$ , where each edge is represented by a pair of nodes  $(p(v), v)$ .

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

A *synteny* is a genomic region containing a set of genes from a set of gene families  $\mathcal{F}$ , where the genes of a given synteny all belong to different gene families (i.e. repeated gene copies inside a synteny are ignored). Therefore, from now on, a gene is simply identified by the family  $g \in \mathcal{F}$  it belongs to. We call a *synteny tree* on  $\mathcal{F}$  a tuple  $\mathcal{T} = \langle T, \tilde{x} \rangle$  where  $T$  is a binary tree and  $\tilde{x} : L(T) \rightarrow \mathcal{P}(\mathcal{F}) - \{\emptyset\}$  ( $\mathcal{P}(\mathcal{F})$  is the powerset of  $\mathcal{F}$ ) is the function mapping each leaf  $v \in L(T)$  (element of  $\mathcal{X}$ ) to its *synteny content*. We will say that  $v$  is *labeled* by  $\tilde{x}(v)$ .

Finally, the restriction of any function  $f$  to a subset  $A$  of its domain is denoted by  $f|_A$ .

<sup>1</sup> For the sake of clarity, the proofs are given in Appendix. The proof of Theorem 4 is omitted for space constraints. It will be included in a future journal version.

## 2 The Small Gain-Loss Parsimony Problem

We consider a tree-like evolutionary model involving gain and loss episodes, starting with a unary root with an empty gene set, and giving rise to a set  $\mathcal{F}$  of gene families and to an extant set of syntenies on  $\mathcal{F}$ . Those syntenies may represent genomic regions as well as entire genomes, identified by their gene content (with no duplicates, i.e. with at most one copy per gene family).

Except for the root, all other internal nodes of a history are binary. Binary nodes may correspond to speciations, as well as duplications or transfers in the case of genomic regions. However, as the only events considered in our optimization function are gains and losses, we ignore the nature of those binary nodes.

Since we do not consider the order of genes inside the syntenies, the relative order of gain and loss events on a given edge of a history is irrelevant. Moreover, as we will aim to minimize the overall number of events, any sequence of consecutive gains or losses can be collapsed into a single event. Therefore at most one gain and/or one loss may occur on an edge, thus instead of representing gains and losses as unary nodes, we rather represent them as edge labels.

In the following, for a binary tree  $T$ , we will denote by  $\dot{T}$  the tree obtained from  $T$  by rerooting it to a new unary root, i.e. by adding a new unary node  $u$  with a new edge  $(u, r(T))$ . This is required in order to account for an initial gain event on an initial edge connecting a hypothetical ancestor with an empty content to the root of the input tree with a non-empty content.

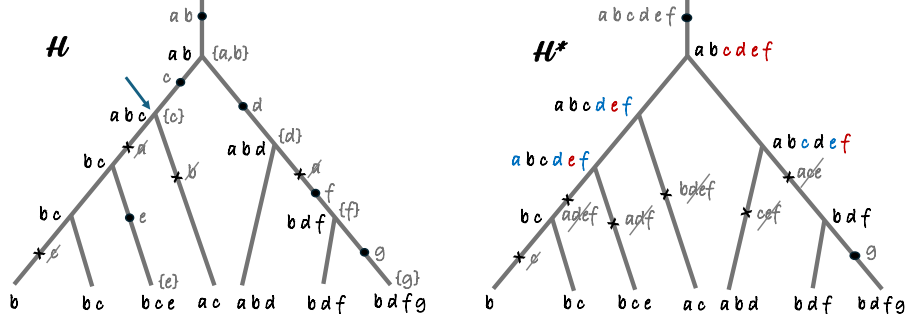
**Definition 1 (Explanatory history).** Let  $\mathcal{T} = \langle T, \tilde{x} \rangle$  be a synteny tree on  $\mathcal{F}$ . A history  $\mathcal{H}$  explaining  $\mathcal{T}$  is a tuple  $\langle \dot{T}, \epsilon, x \rangle$ , where each node  $v \in V(\dot{T})$  is labeled with a synteny content  $x(v) \in \mathcal{P}(\mathcal{F})$ , and each edge  $e \in E(\dot{T})$  is labeled with an event subset  $\epsilon(e) \subseteq \{\text{Gain}, \text{Loss}\}$  satisfying

1. For each  $l \in L(\dot{T})$ ,  $x(l) = \tilde{x}(l)$
2. For each edge  $(v, v_c) \in E(\dot{T})$ :
  - (a) Loss  $\in \epsilon((v, v_c))$  iff  $x(v) \not\subseteq x(v_c)$
  - (b) Gain  $\in \epsilon((v, v_c))$  iff  $x(v_c) \not\subseteq x(v)$
3. For each gene family  $g \in \mathcal{F}$ , there is a unique node  $v \in V(\dot{T}) - \{r(\dot{T})\}$  such that  $g \in x(v) - x(p(v))$ . We say that  $v$  is the gain point for  $g$  and denote it as  $\text{Gain}_{\mathcal{H}}(g)$ .
4.  $x(r(\dot{T})) = \emptyset$  and for all  $v \in V(\dot{T}) - \{r(\dot{T})\}$ ,  $x(v) \neq \emptyset$ .

The set of all histories explaining a synteny tree  $\mathcal{T}$  is denoted by  $\mathbb{H}(\mathcal{T})$ .

Given a history  $\mathcal{H}$  and a node  $v \in V(\dot{T})$ , we call subhistory of  $\mathcal{H}$  rooted at  $v$  the history  $\mathcal{H}_v = \langle T_v, \epsilon|_{E(T_v)}, x|_{V(T_v)} \rangle$ , and hanging subhistory of  $\mathcal{H}$  rooted at  $v$  the same subhistory but also including the edge of  $\mathcal{H}$  from  $p(v)$  to  $v$  (or, said differently, the subhistory of  $\mathcal{H}$  rooted at  $p(v)$  but only keeping the subhistory of its child  $v$ ).

Finally, for a given history  $\mathcal{H}$  explaining  $\mathcal{T}$ , let  $c_{\text{gain}}(\mathcal{H})$  (respec.  $c_{\text{loss}}(\mathcal{H})$ ) be the number of gains (respec. losses) of  $\mathcal{H}$ , and let  $\delta_{\text{gain}} > 0$  (respec.  $\delta_{\text{loss}} > 0$ ) be the gain (respec. loss) cost. Then the cost of  $\mathcal{H}$  is  $c(\mathcal{H}, \delta_{\text{gain}}, \delta_{\text{loss}}) = \delta_{\text{gain}} \cdot$



**Fig. 1.** Two different histories explaining the same synteny tree  $\langle T, \tilde{x} \rangle$  on  $\mathcal{F} = \{a, b, c, d, e, f, g\}$ . In other words,  $T$  is simply the underlying binary tree (the same on the left and on the right), and  $\tilde{x}$  is the synteny labeling of its leaves. In both histories, gains are represented by dots and losses by crosses. The gray characters represent the set of gained or lost characters. Although written as sequences, the syntenies are unordered sets of genes. In the left figure, the set of characters in curly brackets at a node  $v$  represents  $x_{\text{lca}}(v)$  (see Definition 2). In both trees, each node  $v$  is labeled by  $x(v)$  for the considered history. Three pairwise disjoint covering subsets of  $x(v)$  are introduced in Definition 3; here, we color each gene  $g \in x(v)$  black if  $g \in x_{\text{min}}(v)$ , red if  $g \in x_{\text{in}}(v)$  and blue if  $g \in x_{\text{out}}(v)$ . (Left) A suboptimal history of cost 10 (6 gains and 4 losses) for a unit cost obtained by assigning  $x_{\text{min}}(v)$  to each node  $v$  of  $T$ . (Right) An optimal history of cost 8 (2 gains and 6 losses), for the unit cost  $\delta_{\text{loss}} = \delta_{\text{gain}} = 1$ .

$c_{\text{gain}}(\mathcal{H}) + \delta_{\text{loss}} \cdot c_{\text{loss}}(\mathcal{H})$ . We are now ready to state the optimization problem. See Figure 1 for an example.

**SMALL GAIN-LOSS PARSIMONY PROBLEM.**

**Input:** A synteny tree  $\mathcal{T} = \langle T, \tilde{x} \rangle$  on  $\mathcal{F}$ ,  $\delta_{\text{gain}}$  and  $\delta_{\text{loss}}$ .

**Output:**  $\mathcal{H}^* \in \text{argmin}_{\mathcal{H} \in \mathbb{H}(\mathcal{T})} \{c(\mathcal{H}, \delta_{\text{gain}}, \delta_{\text{loss}})\}$ .

For conciseness, in the rest of this paper, we omit  $\delta_{\text{gain}}$  and  $\delta_{\text{loss}}$  in the input of the cost  $c(\mathcal{H}, \delta_{\text{gain}}, \delta_{\text{loss}})$  and simply write  $c(\mathcal{H})$ .

### 3 Algorithm

In this section, we are given a synteny tree  $\mathcal{T} = \langle T, \tilde{x} \rangle$  on a set  $\mathcal{F}$  of gene families. By definition, the only difference between  $T$  and  $\dot{T}$  is an additional unary root in  $\dot{T}$ . We will alternate between  $T$  and  $\dot{T}$  depending on whether a unary root needs to be considered or not.

We first define the lowest possible gain point for each gene family.

**Definition 2 (Set of LCAs at a node).** *The least common ancestor of a gene family  $g \in \mathcal{F}$  is defined as:*

$$\text{lca}(g) = \text{lca}_T(\{l \in L(T) \mid g \in \tilde{x}(l)\})$$

For any  $v \in V(T)$ , we define  $x_{\text{lca}}(v) = \{g \in \mathcal{F} \mid \text{lca}(g) = v\}$ .

We can then define the minimum and extra contents at a node  $v$  of  $T$ .

**Definition 3 (Minimum and extra contents at a node).** Let  $v$  be a node of  $T$ . We define:

$$\begin{aligned} x_{\min}(v) &= \{g \in \mathcal{F} \mid \text{lca}(g) \leq v \wedge g \in \cup_{l \in L(T_v)} \tilde{x}(l)\} \\ x_{\text{out}}(v) &= \{g \in \mathcal{F} \mid g \notin \cup_{l \in L(T_v)} \tilde{x}(l)\} \\ x_{\text{in}}(v) &= \{g \in \mathcal{F} \mid \text{lca}(g) > v\} \end{aligned}$$

For example, for the node  $v$  pointed by the blue arrow on the left tree of Figure 1,  $x_{\text{lca}}(v) = \{c\}$ ,  $x_{\min}(v) = \{a, b, c\}$ ,  $x_{\text{out}}(v) = \{d, f, g\}$  and  $x_{\text{in}}(v) = \{e\}$ .

Notice that, for  $v \in T$ ,  $x_{\min}(v)$  corresponds to the set of gene families with lca above  $v$  and present in at least one leaf descendant of  $v$ ,  $x_{\text{out}}(v)$  corresponds to the gene families that are not present in the leaves descendant of  $v$  and  $x_{\text{in}}(v)$  corresponds to the set of gene families with lca below  $v$ . Therefore,  $x_{\min}(v) \cup x_{\text{out}}(v) \cup x_{\text{in}}(v) = \mathcal{F}$  and the three sets are pairwise disjoint. The next lemma trivially follows from this fact and from the fact that a gene cannot be gained twice in a history.

**Lemma 1 (Content of a node).** Let  $\mathcal{H} = \langle \dot{T}, \epsilon, x \rangle \in \mathbb{H}(\mathcal{T})$  and let  $v \in V(T)$ . Then,  $x(v) = x_{\min}(v) \cup O \cup I$  for some sets  $O \subseteq x_{\text{out}}(v)$  and  $I \subseteq x_{\text{in}}(v)$ .

According to Lemma 1, each node  $v$  of a history must be in one of the following four states: “min” if its content is exactly  $x_{\min}(v)$ , “out” if it additionally contains at least one gene from  $x_{\text{out}}(v)$  but none from  $x_{\text{in}}(v)$ , “in” in the opposite case, or “in out” if it contains at least one gene from both  $x_{\text{out}}(v)$  and  $x_{\text{in}}(v)$ . In the following, we will build recurrence equations to compute the minimum possible cost for each subhistory of an history  $\mathcal{H}$  explaining  $\mathcal{T} = \langle T, \tilde{x} \rangle$ , distinguishing between these four possible states for each node.

**Definition 4 (Cost of a subhistory).** Let  $v$  be a node of  $T$ . For a state  $\sigma \in \{\text{min}, \text{out}, \text{in}, \text{in out}\}$ , define  $c_{\sigma}(v)$  as the minimum cost of an explanatory subhistory rooted at  $v$  such that  $v$  is in state  $\sigma$ , i.e.,

- for  $c_{\min}(v)$ , such that  $x(v) = x_{\min}(v)$ ;
- for  $c_{\text{out}}(v)$ , such that  $x(v) = x_{\min}(v) \cup O$  with  $\emptyset \neq O \subseteq x_{\text{out}}(v)$ ;
- for  $c_{\text{in}}(v)$ , such that  $x(v) = x_{\min}(v) \cup I$  with  $\emptyset \neq I \subseteq x_{\text{in}}(v)$ ;
- for  $c_{\text{in out}}(v)$ , such that  $x(v) = x_{\min}(v) \cup O \cup I$  with  $O$  and  $I$  as above.

If  $v$  has a child  $v_c$ , define  $\Delta_{\sigma}(v_c)$  as the minimum cost of an explanatory hanging subhistory rooted at  $v$  such that  $v$  is in state  $\sigma$  relative to  $v_c$ , i.e.,

- for  $\Delta_{\min}(v_c)$ , such that  $x(v) = x_{\min}(v)$ ;
- for  $\Delta_{\text{out}}(v_c)$ , such that  $x(v) = x_{\min}(v) \cup O$  with  $\emptyset \neq O \subseteq x_{\text{out}}(v_c) - x_{\min}(v)$ ;
- for  $\Delta_{\text{in}}(v_c)$ , such that  $x(v) = x_{\min}(v) \cup I$  with  $\emptyset \neq I \subseteq x_{\text{in}}(v_c) \cup x_{\text{lca}}(v_c)$ ;
- for  $\Delta_{\text{in out}}(v_c)$ , such that  $x(v) = x_{\min}(v) \cup O \cup I$  with  $O$  and  $I$  as above.

**Lemma 2 (Partial recurrences including the root edge).** *For any edge  $(v, v_c) \in E(T)$ , assuming the values for  $c_\sigma(v_c)$ ,  $x_{\min}(v)$ , and  $x_{\min}(v_c)$  are known, the values of  $\Delta_\sigma(v_c)$  may be computed as follows:*

$$\Delta_{\min}(v_c) = \min \begin{cases} c_{\min}(v_c) + l(v, v_c) \cdot \delta_{\text{loss}} + g(v, v_c) \cdot \delta_{\text{gain}} & (1a) \\ c_{\text{out}}(v_c) + g(v, v_c) \cdot \delta_{\text{gain}} & \text{if } l(v, v_c) = 1 \quad (1b) \\ c_{\text{in}}(v_c) + l(v, v_c) \cdot \delta_{\text{loss}} + \delta_{\text{gain}} & (1c) \\ c_{\text{in out}}(v_c) + \delta_{\text{gain}} & \text{if } l(v, v_c) = 1 \quad (1d) \end{cases}$$

$$\Delta_{\text{out}}(v_c) = \min \begin{cases} c_{\min}(v_c) + \delta_{\text{loss}} + g(v, v_c) \cdot \delta_{\text{gain}} & (2a) \\ c_{\text{out}}(v_c) + g(v, v_c) \cdot \delta_{\text{gain}} & (2b) \\ c_{\text{in}}(v_c) + \delta_{\text{loss}} + \delta_{\text{gain}} & (2c) \\ c_{\text{in out}}(v_c) + \delta_{\text{gain}} & (2d) \end{cases}$$

$$\Delta_{\text{in}}(v_c) = \min \begin{cases} c_{\min}(v_c) + l(v, v_c) \cdot \delta_{\text{loss}} & \text{if } g(v, v_c) = 1 \quad (3a) \\ c_{\text{out}}(v_c) & \text{if } g(v, v_c) = 1 \text{ and } l(v, v_c) = 1 \quad (3b) \\ c_{\text{in}}(v_c) + l(v, v_c) \cdot \delta_{\text{loss}} & (3c) \\ c_{\text{in out}}(v_c) & \text{if } l(v, v_c) = 1 \quad (3d) \end{cases}$$

$$\Delta_{\text{in out}}(v_c) = \min \begin{cases} c_{\min}(v_c) + \delta_{\text{loss}} & \text{if } g(v, v_c) = 1 \quad (4a) \\ c_{\text{out}}(v_c) & \text{if } g(v, v_c) = 1 \quad (4b) \\ c_{\text{in}}(v_c) + \delta_{\text{loss}} & (4c) \\ c_{\text{in out}}(v_c), & (4d) \end{cases}$$

where  $l(v, v_c)$  and  $g(v, v_c)$  are indicator functions defined as

$$l(v, v_c) = \begin{cases} 1 & \text{if } x_{\min}(v) \not\subseteq x_{\min}(v_c) \\ 0 & \text{otherwise} \end{cases} \quad g(v, v_c) = \begin{cases} 1 & \text{if } x_{\min}(v_c) \not\subseteq x_{\min}(v) \\ 0 & \text{otherwise.} \end{cases}$$

See an illustration for the  $\Delta_{\min}(v_c)$  case in Figure 2, for  $\Delta_{\text{out}}(v_c)$  in Figure 6 (in Appendix) and for  $\Delta_{\text{in}}(v_c)$  in Figure 7.

We are now ready to state the main recurrence equations allowing us to solve the SMALL GAIN-LOSS PARSIMONY PROBLEM.

**Theorem 1 (Main recurrences).** *For any node  $v \in V(T)$ , the values of  $c_\sigma(v)$  may be computed as follows. If  $v$  is a leaf,  $c_{\min}(v) = 0$  and  $c_{\text{out}}(v) = c_{\text{in}}(v) =$*

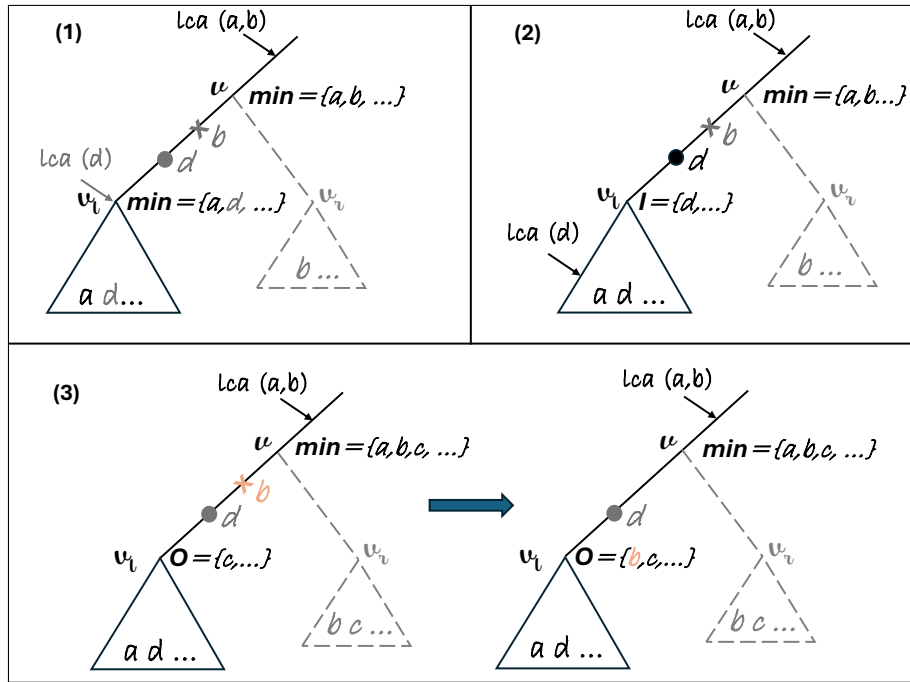
$c_{\text{in out}}(v) = \infty$ . Otherwise,

$$c_{\text{min}}(v) = \begin{cases} \Delta_{\text{min}}(v_\ell) + \Delta_{\text{min}}(v_r) & \text{if } x_{\text{min}}(v) \neq \emptyset \\ \infty & \text{otherwise} \end{cases}$$

$$c_{\text{out}}(v) = \Delta_{\text{out}}(v_\ell) + \Delta_{\text{out}}(v_r)$$

$$c_{\text{in}}(v) = \min \begin{cases} \Delta_{\text{in}}(v_\ell) + \Delta_{\text{out}}(v_r) \\ \Delta_{\text{out}}(v_\ell) + \Delta_{\text{in}}(v_r) \\ \Delta_{\text{in out}}(v_\ell) + \Delta_{\text{in out}}(v_r) \end{cases}$$

$$c_{\text{in out}}(v) = \min \begin{cases} \Delta_{\text{in out}}(v_\ell) + \Delta_{\text{out}}(v_r) \\ \Delta_{\text{out}}(v_\ell) + \Delta_{\text{in out}}(v_r) \\ \Delta_{\text{in out}}(v_\ell) + \Delta_{\text{in out}}(v_r). \end{cases}$$



**Fig. 2.** An illustration of a left scenario (of cost  $\Delta_{\text{min}}(v_\ell)$ ) leading to  $c_{\text{min}}(v)$  (a similar scenario can be drawn for the right part, i.e. for  $\Delta_{\text{min}}(v_r)$ ); the characters in a triangle represent the gene families in the leaves of the corresponding subtree; the events and characters in gray can be present or absent, depending on the value of  $l(v, v_\ell)$  and  $g(v, v_\ell)$ . (1): A scenario with  $x(v_\ell) = x_{\text{min}}(v_\ell)$  (line 1a). (2): A scenario with  $x(v_\ell) = x_{\text{min}}(v_\ell) \cup I$  where  $I \subseteq x_{\text{in}}(v_\ell)$ ,  $I \neq \emptyset$  (line 1c). There must be a gain on the edge  $(v, v_\ell)$  because  $I \subseteq x(v_\ell) - x(v)$  and therefore  $x(v_\ell) \not\subseteq x(v)$ . (3): Two scenarios with  $x(v_\ell) = x_{\text{min}}(v_\ell) \cup O$  where  $O \subseteq x_{\text{out}}(v_\ell)$ ,  $O \neq \emptyset$  (line 1b). The left scenario is not optimal with a loss on the edge  $(v, v_\ell)$  as the lost gene (here  $b$ ) can always be postponed to be lost later in the same losses of the syntenies containing the extra gene  $c$ .



In order to compute  $c_{\min}(v)$ ,  $c_{\text{out}}(v)$ ,  $c_{\text{in}}(v)$  and  $c_{\text{in out}}(v)$  using Theorem 1, we need to determine the value of  $x_{\min}(v)$ , which is the purpose of Algorithm 2. This, in turn, requires computing  $x_{\text{lca}}$ , which is done using Algorithm 1. The correctness of both algorithms is proven in Lemma 3.

<hr/> <b>Algorithm 1:</b> LCA-Content( $\mathcal{T} = \langle T, \tilde{x} \rangle$ ) <hr/> <pre> 1 <b>for each</b> <math>v \in V(T)</math> <i>in post-order</i>   <b>do</b> 2   <b>if</b> <math>v \in L(T)</math> <b>then</b> 3     <math>A(v) \leftarrow \tilde{x}(v)</math> 4   <b>else</b> 5     <math>A(v) \leftarrow A(v_\ell) \cup A(v_r)</math> 6 <math>B(r(T)) \leftarrow A(r(T))</math> 7 <b>for each</b> <math>v \in V(T)</math> <i>in pre-order</i>   <b>do</b> 8   <b>if</b> <math>v \in L(T)</math> <b>then</b> 9     <math>x_{\text{lca}}(v) \leftarrow B(v)</math> 10  <b>else</b> 11    <math>B(v_\ell) \leftarrow B(v) - A(v_r)</math> 12    <math>B(v_r) \leftarrow B(v) - A(v_\ell)</math> 13    <math>x_{\text{lca}}(v) \leftarrow</math> 14      <math>B(v) - B(v_\ell) - B(v_r)</math> 14 <b>return</b> <math>x_{\text{lca}}</math>                 </pre> <hr/>	<hr/> <b>Algorithm 2:</b> Min-Content( $\mathcal{T} = \langle T, \tilde{x} \rangle, x_{\text{lca}}$ ) <hr/> <pre> 1 <b>for each</b> <math>v \in V(T)</math> <i>in post-order</i>   <b>do</b> 2   <b>if</b> <math>v \in L(T)</math> <b>then</b> 3     <math>x_{\min}(v) \leftarrow \tilde{x}(v)</math> 4   <b>else</b> 5     <math>x_{\min}(v) \leftarrow</math> 6       <math>(x_{\min}(v_\ell) - x_{\text{lca}}(v_\ell)) \cup</math> 7       <math>(x_{\min}(v_r) - x_{\text{lca}}(v_r))</math> 6 <b>return</b> <math>x_{\min}</math>                 </pre> <hr/>
--	--

**Lemma 3 (Computing  $x_{\text{lca}}$  and  $x_{\min}$ ).** *For all  $v \in V(T)$ , Algorithm 1 returns  $x_{\text{lca}}(v)$  and Algorithm 2 returns  $x_{\min}(v)$ .*

Using the equations from Theorem 1, Algorithm 3 returns the cost of an optimal history explaining  $\mathcal{T}$ .

<hr/> <b>Algorithm 3:</b> InOutParsimonyCost( $\mathcal{T} = \langle T, \tilde{x} \rangle, x_{\min}, \delta_{\text{loss}}, \delta_{\text{gain}}$ ) <hr/> <pre> 1 <b>for each</b> <math>v \in V(T)</math> <i>in post-order</i> <b>do</b> 2   <math>\mid</math> compute <math>c_{\min}(v)</math>, <math>c_{\text{out}}(v)</math>, <math>c_{\text{in}}(v)</math>, <math>c_{\text{in out}}(v)</math> according to Theorem 1 3 <b>return</b> <math>\delta_{\text{gain}} + \min\{c_{\min}(r(T)), c_{\text{in}}(r(T))\}</math>                 </pre> <hr/>
--

**Corollary 1.** *Algorithm 3 returns  $\min_{\mathcal{H}=\langle \dot{T}, \epsilon, x \rangle \in \mathbb{H}(\mathcal{T})} \{c(\mathcal{H})\}$ .*

From the result of Algorithm 3 and through a standard backtracking procedure, we can output  $\epsilon$ , i.e. the positions of gain and loss events in an optimal history. The remaining information, i.e. the content  $x$  of ancestral syntenies, is then returned by Algorithm 4.

**Algorithm 4:** InOutParsimonyContent( $\mathcal{T} = \langle T, \tilde{x} \rangle, \epsilon, x_{\min}$ )

---

```

1 for each  $v \in V(T)$  in post-order do
2    $x(v) \leftarrow x_{\min}(v)$ 
3   if  $v \notin L(T)$  then
4     if Gain  $\notin \epsilon((v, v_\ell))$  then
5        $x(v) \leftarrow x(v) \cup x(v_\ell)$ 
6     if Gain  $\notin \epsilon((v, v_r))$  then
7        $x(v) \leftarrow x(v) \cup x(v_r)$ 
8 for each  $v \in V(T) - \{r(T)\}$  in pre-order do
9   if Loss  $\notin \epsilon((p(v), v))$  then
10     $x(v) \leftarrow x(v) \cup x(p(v))$ 
11  $x(r(\dot{T})) \leftarrow \emptyset$ 
12 return  $\langle \dot{T}, \epsilon, x \rangle$ 

```

---

**Theorem 2 (Correctness of Algorithm 4).** *If  $\epsilon$  is such that there exists  $\langle \dot{T}, \epsilon, x^* \rangle \in \operatorname{argmin}_{\mathcal{H} \in \mathbb{H}(\mathcal{T})} \{c(\mathcal{H})\}$ , then Algorithm 4 returns a history  $\langle \dot{T}, \epsilon, x \rangle \in \operatorname{argmin}_{\mathcal{H} \in \mathbb{H}(\mathcal{T})} \{c(\mathcal{H})\}$  on input  $(\mathcal{T} = \langle T, \tilde{x} \rangle, \epsilon, x_{\min})$ .*

As stated in the next theorem, the history returned by Algorithm 4 is unique for a given input. More precisely, given  $\epsilon$  leading to an optimal history for  $\mathcal{T}$ , there is a unique mapping  $x$  such that  $\langle \dot{T}, \epsilon, x \rangle \in \operatorname{argmin}_{\mathcal{H} \in \mathbb{H}(\mathcal{T})} \{c(\mathcal{H})\}$ .

**Theorem 3 (Uniqueness of  $x$  given  $\epsilon$ ).** *Let  $\langle \dot{T}, \epsilon, x \rangle \in \operatorname{argmin}_{\mathcal{H} \in \mathbb{H}(\mathcal{T})} \{c(\mathcal{H})\}$ . There exists no history  $\langle \dot{T}, \epsilon, x' \rangle \in \mathbb{H}(\mathcal{T})$  such that  $x' \neq x$ .*

We are now ready to state our main InOutParsimony algorithm (Algorithm 5 below) solving the SMALL GAIN-LOSS PARSIMONY PROBLEM. Its correctness directly follows from all previous results of this section. See Figure 3 for an example of the execution of the algorithm.

**Algorithm 5:** InOutParsimony( $\mathcal{T} = \langle T, \tilde{x} \rangle, \delta_{\text{loss}}, \delta_{\text{gain}}$ )

---

```

1  $x_{\text{lca}} \leftarrow \text{LCA-Content}(\mathcal{T} = \langle T, \tilde{x} \rangle)$ 
2  $x_{\min} \leftarrow \text{Min-Content}(\mathcal{T} = \langle T, \tilde{x} \rangle, x_{\text{lca}})$ 
3  $\epsilon \leftarrow$  backtracking procedure from the result of
   InOutParsimonyCost( $\mathcal{T} = \langle T, \tilde{x} \rangle, x_{\min}, \delta_{\text{loss}}, \delta_{\text{gain}}$ )
4 return InOutParsimonyContent( $\mathcal{T} = \langle T, \tilde{x} \rangle, \epsilon, x_{\min}$ )

```

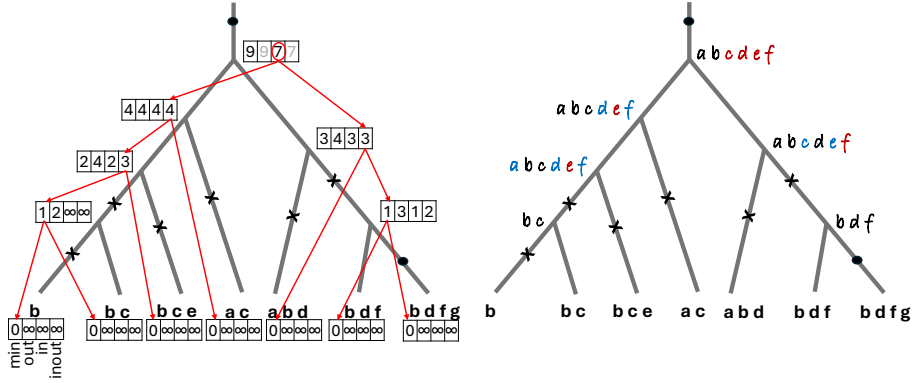
---

Notice that the standard backtracking procedure mentioned above for Algorithm 3 does not necessarily lead to a unique  $\epsilon$ . Consider breaking the ties in the minimums (see Theorem 1) by selecting the first (from top to bottom) line leading to the minimum. Using this backtracking, that we call *First-Line-Backtracking*, we obtain an optimal history of minimum total size in terms of gene contents, as shown in the next theorem.

**Theorem 4 (A solution minimizing gene contents).** *If  $x_{\min}(v) \neq \emptyset$  for all  $v \in V(T)$ , then Algorithm InOutParsimony (Algorithm 5) using First-Line-Backtracking returns the solution  $\langle \hat{T}, \epsilon, x \rangle$  minimizing  $\sum_{v \in V(T)} |x(v)|$  among all possible solutions  $\operatorname{argmin}_{\mathcal{H} = \langle \hat{T}, \epsilon, x \rangle \in \mathbb{H}(\mathcal{T})} \{c(\mathcal{H})\}$ .*

We finally state the time complexity of Algorithm 5, assuming operations on sets can be done in constant time (by encoding the gene family sets as binary vectors fitting into a constant number of 64-bit words).

**Theorem 5 (Time complexity).** *Assuming operations on sets can be performed in constant time, the SMALL GAIN-LOSS PARSIMONY PROBLEM can be solved in  $O(n)$  time where  $n = |V(T)|$ .*



**Fig. 3.** (Left) the synteny tree  $\langle T, \tilde{x} \rangle$  from Figure 1 with each node  $v$  labeled with a vector of size four containing the values (respectively from left to right) of  $c_{\min}(v)$ ,  $c_{\text{out}}(v)$ ,  $c_{\text{in}}(v)$  and  $c_{\text{in out}}(v)$ . Those values are computed by Algorithm 3, considering a unit cost  $\delta_{\text{loss}} = \delta_{\text{gain}} = 1$ . Notice that the  $c_{\text{out}}(v)$  and  $c_{\text{in out}}(v)$  positions in the vector of  $r(T)$  (shown in light gray) do not lead to a valid solution. From Line 4 of Algorithm 3, as  $\min\{c_{\min}(r(T)), c_{\text{in}}(r(T))\} = 7$  and  $\delta_{\text{gain}} = 1$ , the cost of a minimum solution returned by the algorithm is 8. The *First-Line-Backtracking* (see Theorem 4) is illustrated by the red pointers. It leads to the optimal labeling  $\epsilon$  (positions of the 2 gains and 6 losses). (Right) The optimal history output by Algorithm 4 when executed on the tree and events on the left. The gene families in black and red are added to the content of internal nodes during the bottom-up part of Algorithm 4, while those in blue are added during the top-down part of Algorithm 4. This optimal history corresponds to  $\mathcal{H}^*$  from Figure 1.

## 4 Application

We implemented InOutParsimony in Python. Given a binary tree, a content labeling for the tree leaves, and the costs for gains and losses, the program returns

an optimal history as a fully labeled tree in Newick format. The source code can be accessed at <https://github.com/UdeM-LBIT/InOutParsimony/tree/recombcg2026>.

We applied InOutParsimony to a dataset of mitochondrial-encoded protein families, called MitoCOGs, taken from [17]. This dataset was considered in a previous paper of our group [3] for studying the exchange of genes between the mitochondrial and nuclear eukaryotic genomes. In fact, it is largely established that all extant mitochondria originated from a unique endosymbiotic event integrating an  $\alpha$ -proteobacterial genome into an eukaryotic cell, creating an ancestral mitochondrial genome. Subsequently, eukaryote evolution was marked by episodes of Endosymbiotic Gene Transfers (EGT), meaning the transfer of genes between the mitochondrial and nuclear genomes of the same species, mainly from the mitochondria to the nucleus, eventually leading to the disappearance of the mitochondria. However, a high variability of gene repertoire distribution indicating an ongoing EGT process in both directions is still observable in some lineages such as in land plants.

We took the same dataset of 11 plants considered in [3] chosen to have MitoCOGs in both their nuclear and mitochondrial genomes. This set includes nine *Viridiplantae*, one *Rhodophyta* (*Cyanidioschyzon merolae*) and one *Glaucophyta* (*Cyanophora paradoxa*). The selected MitoCOGs were those appearing in either the mitochondrial or nuclear genomes of those 11 plants, yielding 81 MitoCOGs.

We ran InOutParsimony with two models: (1) the *unitary-cost model* with  $\delta_{\text{gain}} = \delta_{\text{loss}}$  minimizing the number of events, and (2) a *gains-at-LCA model*, based on the previous model of *Synesth* [11], in which the gain point of each gene family is positioned at its LCA and gains are excluded from the optimization function, only minimizing losses. Figure 4 for the unitary-cost and Figure 5 for the gains-at-LCA model illustrate the predicted mitochondrial and nuclear contents for the seven multisubunits complexes given in Kannan’s paper [17, Table S8].

The scenario with the unit-cost model involves two large gains, one at the root and the other at one of its children, while those gains are rather split into smaller ones closer to the leaves with the gains-at-LCA model. The first scenario avoids splitting the genes of a given complex into different gain episodes, which may be considered more biologically relevant. For example, the SDH complex in *Cyanidioschyzon merolae* is gained in two episodes with the unit-cost model, while it is gained in three episodes with the gains-at-LCA model (i.e. with an additional gain on the terminal edge). In addition, the disappearance of this complex in the mitochondrial genome of *Chlamydomonas reinhardtii*, or similarly in the nuclear genome of *Cyanidioschyzon merolae*, is explained by a single loss.

We wanted to see how well the two scenarios reflect the EGTs predicted on the same dataset with EndoRex [3]. This software takes as input a species tree and a gene tree where each leaf is additionally labeled 0 or 1 depending on whether the corresponding gene appears in the mitochondrial or nuclear genome, and returns a reconciliation involving EGT events. Figure 4 in [3] shows such predicted reconciliations obtained with different cost functions. The most sup-

ported events are an EGT of MitoCOG0014 (corresponding to a gene family of the ATP complex) from mitochondria to nucleus in *Chlamydomonas reinhardtii*, and another EGT of MitoCOG0072 (belonging to the RPS complex) from nucleus to mitochondria in the LCA of *Ostreococcus tauri* and *Micromonas sp. RCC299*. These two events are more consistent with the scenario obtained with the gains-at-LCA model as, in each case, the loss in one genome (mitochondria or nucleus) is accompanied with a parallel gain in the other genome, while on the scenario obtained with the unit-cost model the complex it is gained earlier in the tree.

We also compared the output of our algorithm with the output of Count [8] under parsimony and probabilistic models. For the latter, similarly to Kannan [17], because gene gain in mitochondrial genome is rare, a pure-loss model architecture was assumed and the prior distribution at the root was assumed to be Poisson. InOutParsimony’s results for the unit-cost model are comparable to Count’s results for the probabilistic pure-loss model (see Figure 4 and Figure 8 in Appendix), while InOutParsimony’s results for the gains-at-LCA model are comparable to Count’s results under Dollo parsimony (see Figure 5 and Figure 9 in Appendix). For both models, Count’s inferred sizes are almost always slightly lower than those inferred with InOutParsimony. This is due to the fact that each gene family is considered independently one from the other by Count, while InOutParsimony is able to consider them together and group individual losses in unique loss episodes placing them lower in the tree, thus keeping more genes at internal nodes.

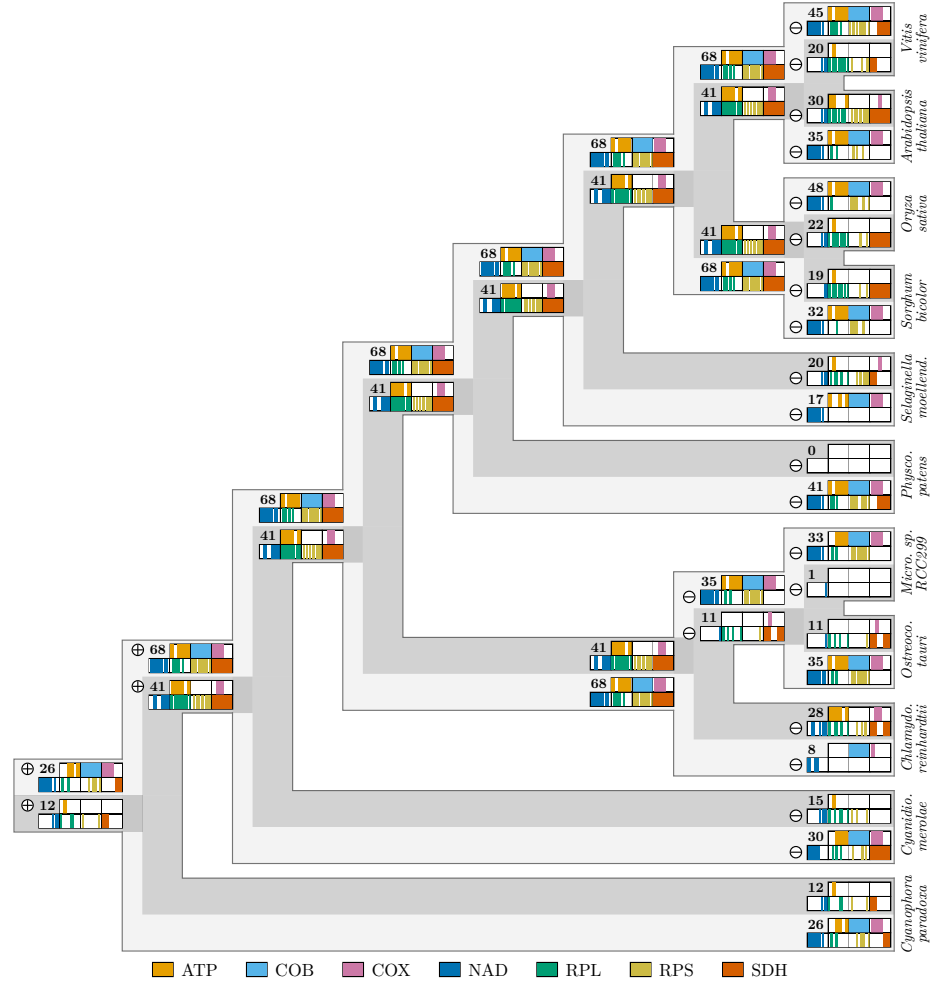
While grouping losses can more appropriately reflect the unique transfer of a full gene complex, as illustrated above with the SDH complex, it can also lead to artificially large genomic sizes on internal nodes. This argues for an intermediate model, going beyond Count’s single loss model (a loss per gene family) but penalizing each segmental loss according to its size.

Finally, prohibiting parallel gains (Dollo’s parsimony) can also artificially increase genomic sizes, in addition to losing the possibility of identifying multiple EGT events. For example, gene ATP9 (MitoCOG0014), only found in the nuclear genomes of *Arabidopsis* and *Chlamydomonas*, is explained by the EndoRex software [3] with two EGT events copying the gene family from the mitochondria to the nucleus in both species. This gene is also identified as two separate gains with Count under the Wagner parsimony (see Figure 10 in Appendix, where ATP9 is the last vertical strip in the ATP complex rectangles). However, these two EGT events are rather interpreted as a single gain with InOutParsimony.

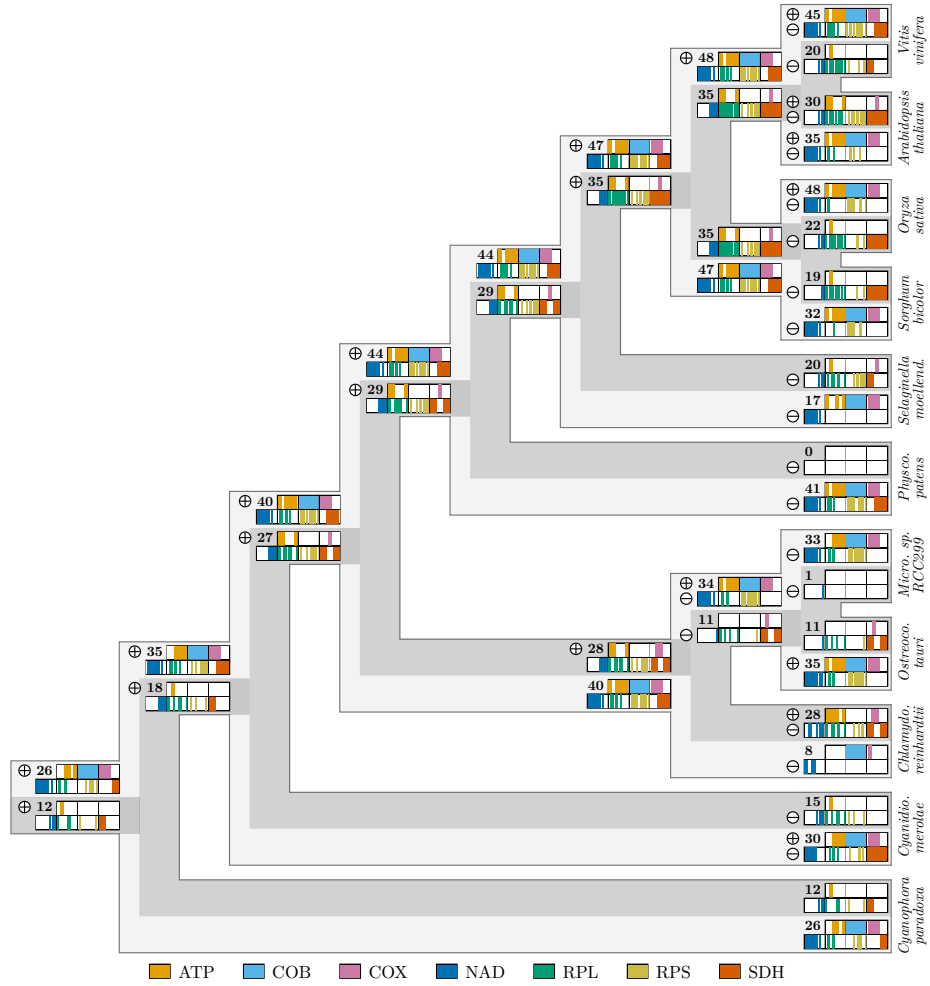
## 5 Conclusion

We developed the first linear-time algorithm that predicts a most parsimonious history scenario of segmental gains and losses for a given phylogenetic tree leaf-labeled by sets of genes without duplicates. Linearity is achieved by testing only four cases at internal nodes rather than all possible genetic contents. The algorithm can be used with arbitrary, but constant, costs for gains and losses.

It will be interesting to extend the model to more general weights, such as affine or convex weight function accounting for the size of a gain or a loss. This could help strike a balance between individual loss and gain scenarios, as inferred



**Fig. 4.** Results obtained with InOutParsimony for the *unitary-cost* model on the species tree of the considered 11 plants. The topology of the tree is based on [17]. Light gray edges (the outer structure) represent the mitochondrial tree, while dark gray edges (the inner structure) represent the nuclear tree. Each internal node of the mitochondrial/nuclear tree contains seven rectangles representing, in order, the seven multisubunits complexes given by Table S8 in [17] and represented on the bottom of this figure. Each rectangle is divided into strips corresponding to the genes of the complex and each strip is colored when the gene is present at that node. Gains and losses are signaled by  $\oplus$  and  $\ominus$  symbols respectively.



**Fig. 5.** Results obtained with InOutParsimony for the *gains-at-LCA* model on the species tree of the considered 11 plants. Representation is the same as in Figure 4.

by Count, and overly broad events leading to redundant segmental losses in the terminal edges of the tree.

It will be also interesting to generalize InOutParsimony to enable exploring the whole solution space through algebraic dynamic programming. For now, our method outputs a single solution among all optimal ones, the one minimizing the total size of gene contents.

Another promising extension of this work would be to allow multiple gain points for a given gene family, extending the model from Dollo parsimony to Wagner parsimony. In our MitoCOGs case study, this would lead to a more precise and robust identification of EGTs.

Finally, the results presented in this paper pave the way to the integration of segmental gain minimization into the Synesth reconciliation framework, enabling the study of gene repertoire's evolution through a comprehensive model involving duplications and horizontal gene transfers.

**Acknowledgments.** We would like to thank Corélie Godefroid for her important work in clearing up the problem. We also thank the community of the “Novel Mathematical Paradigm for Phylogenomics” workshop (hosted by the Banff International Research Station) for fruitful discussions. This research was funded by the *Fonds de recherche du Québec – Nature et technologies* [grant numbers 335135 (MG) and 335893 (MD)] and the Natural Sciences and Engineering Research Council of Canada [grant number RN000743 (NEM)].

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## Appendix

### Additional Notation

Let  $T$  be a tree. For any two nodes  $v$  and  $v'$  of  $T$ , there exists a unique path from  $v$  to  $v'$  that we denote  $P_T(v, v') \subseteq E(T)$ .

Let  $v \in V(T)$ . The depth of  $v$  is defined as  $|P_T(r(T), v)|$  and the height of  $v$  is defined as  $\max_{l \in L(T_v)} |P_T(v, l)|$ .

### Additional Definitions

The following definition is used in the proofs in Appendix.

**Definition 5 (Cost of a subhistory with content  $X$ ).** For any  $X \subseteq \mathcal{F}$ , let  $c(v, X)$  be the minimum cost of an explanatory subhistory rooted at  $v$  such that  $x(v) = X$ , i.e  $c(v, X) = \min\{c(\mathcal{H}_v) \mid \mathcal{H} = \langle \dot{T}, \epsilon, x \rangle \in \mathbb{H}(\mathcal{T}) \wedge x(v) = X\}$ .

### Additional Lemmas

The following lemmas are used in the proofs of the results stated in the paper.

**Lemma 4.** Let  $v$  be an internal node of  $T$  and let  $I \subseteq x_{\text{in}}(v)$  be non empty. For any non empty  $O \subseteq x_{\text{out}}(v)$ :

- There exists a subhistory  $\mathcal{H}_v$  of a history  $\mathcal{H} = \langle \dot{T}, \epsilon, x \rangle$  explaining  $\mathcal{T}$  such that  $x(v) = x_{\text{min}}(v) \cup O$  and  $c(\mathcal{H}_v) = c_{\text{out}}(v)$ .
- There exists a subhistory  $\mathcal{H}_v$  of a history  $\mathcal{H} = \langle \dot{T}, \epsilon, x \rangle$  explaining  $\mathcal{T}$  such that  $x(v) = x_{\text{min}}(v) \cup O \cup I$  and  $c(\mathcal{H}_v) = \min_{\substack{O' \subseteq x_{\text{out}}(v) \\ O' \neq \emptyset}} c(v, x_{\text{min}}(v) \cup O' \cup I)$ .



*Proof.* Let  $O \subseteq x_{\text{out}}(v)$  be non empty. First notice that there exists a history  $\mathcal{H} = \langle \dot{T}, \epsilon, x \rangle$  explaining  $\mathcal{T}$  such that  $x(v) \cap x_{\text{out}}(v) = O$ . In fact, it is possible to gain all content in  $x_{\text{out}}(v)$  at the root of  $T$  and then not lose any content from  $O$  and lose all content from  $x_{\text{out}}(v) - O$  on the path from the root to  $v$ .

Now, let  $\mathcal{H}'_v$  be a subhistory of cost  $c_{\text{out}}(v)$  (reps.  $\min_{\substack{O' \subseteq x_{\text{out}}(v) \\ O' \neq \emptyset}} c(v, x_{\text{min}}(v)) \cup O' \cup I$ ) of a history  $\mathcal{H}' = \langle \dot{T}, \epsilon', x' \rangle$  explaining  $\mathcal{T}$ . Notice that there must be a loss event on every path from  $v$  the leaves in  $L(\dot{T}_v)$  as otherwise  $\mathcal{H}$  would not be explaining  $\mathcal{T}$ . Therefore, we can add any given subset of  $x_{\text{out}}(v)$  to  $x'(v)$  and simply lose this new content at those loss events without changing the cost of the subhistory. The result follows.  $\square$

**Lemma 5.** *Let  $\mathcal{H} = \langle \dot{T}, \epsilon, x \rangle \in \mathbb{H}(\mathcal{T})$ . If there is an edge  $(p(v), v) \in E(\dot{T})$  such that  $\text{Gain} \in \epsilon((p(v), v))$  and  $x(p(v)) \cap (x_{\text{in}}(v) \cup x_{\text{lca}}(v)) \neq \emptyset$ , then*

$$\mathcal{H} \notin \text{argmin}_{\mathcal{H}' \in \mathbb{H}(\mathcal{T})} \{c(\mathcal{H}')\}$$

and

$$c(\mathcal{H}_{p(v)}) \notin \min_{I \subseteq x_{\text{in}}(v), I \neq \emptyset} \{c(p(v), x(p(v)) \cup I)\}.$$

*Proof.* Let  $\mathcal{H} = \langle \dot{T}, \epsilon, x \rangle \in \mathbb{H}(\mathcal{T})$  and let's assume that there exists an edge  $(p(v), v) \in E(\dot{T})$  such that  $\text{Gain} \in \epsilon((p(v), v))$  and  $x(p(v)) \cap (x_{\text{in}}(v) \cup x_{\text{lca}}(v)) \neq \emptyset$ . As  $x(p(v)) \cap (x_{\text{in}}(v) \cup x_{\text{lca}}(v)) \neq \emptyset$ , there is at least one gene family  $g$  in  $x(p(v))$  such that  $\text{lca}(g) \geq v$  and  $\text{Gain}_{\mathcal{H}}(g)$  is an ancestor of  $p(v)$ . As  $\text{lca}(g) \geq v$ ,  $g \notin \cup_{l \in L(T) - L(T_v)} \tilde{x}(l)$ . Therefore,  $g$  must be lost at some point between  $\text{Gain}_{\mathcal{H}}(g)$  and each leaves in  $L(T) - L(T_v)$  descending from  $\text{Gain}_{\mathcal{H}}(g)$  because otherwise  $\mathcal{H}$  would not be explaining  $\mathcal{T}$ . We can thus obtain another history  $\mathcal{H}^*$  explaining  $\mathcal{T}$  from  $\mathcal{H}$  by removing  $\text{Gain}$  from  $\epsilon((p(v), v))$  and adding the content that was gained in this event in the content of every descendant node of  $\text{Gain}_{\mathcal{H}}(g)$  containing the gene family  $g$  that is not a descendant of  $v$ . Therefore,  $c(\mathcal{H}^*) = c(\mathcal{H}) - \delta_{\text{gain}}$  and  $\mathcal{H} \notin \text{argmin}_{\mathcal{H}' \in \mathbb{H}(\mathcal{T})} \{c(\mathcal{H}')\}$ . Furthermore,  $c(\mathcal{H}_{p(v)}^*) = c(\mathcal{H}_{p(v)}) - \delta_{\text{gain}}$  and  $x^*(p(v)) = x(p(v)) \cup I$  from some non empty set  $I \subseteq x_{\text{in}}(v)$  by construction and therefore

$$c(\mathcal{H}_{p(v)}) \notin \min_{I \subseteq x_{\text{in}}(v), I \neq \emptyset} \{c(p(v), x(p(v)) \cup I)\}.$$

$\square$

**Lemma 6.** *Let  $\mathcal{H} = \langle \dot{T}, \epsilon, x \rangle \in \mathbb{H}(\mathcal{T})$ . If there is an edge  $(p(v), v) \in E(\dot{T})$  such that  $\text{Loss} \in \epsilon((p(v), v))$  and  $x(v) \cap x_{\text{out}}(v) \neq \emptyset$ , then*

$$\mathcal{H} \notin \text{argmin}_{\mathcal{H}' \in \mathbb{H}(\mathcal{T})} \{c(\mathcal{H}')\} \text{ and } c(\mathcal{H}_{p(v)}) \notin c(p(v), x(p(v))).$$

*Proof.* Let  $\mathcal{H} = \langle \dot{T}, \epsilon, x \rangle \in \mathbb{H}(\mathcal{T})$  and let's assume that there exists an edge  $(p(v), v) \in E(\dot{T})$  such that  $\text{Loss} \in \epsilon((p(v), v))$  and  $x(v) \cap x_{\text{out}}(v) \neq \emptyset$ . As  $x(v) \cap x_{\text{out}}(v) \neq \emptyset$ , there is at least one gene family  $g$  in  $x(v)$  such that  $g \notin$

$\cup_{l \in L(T_v)} \tilde{x}(l)$  and therefore this gene family must be lost at some point between  $v$  and each leaves in  $L(T_v)$  because otherwise  $\mathcal{H}$  would not be explaining  $\mathcal{T}$ . We can thus obtain another history  $\mathcal{H}^*$  explaining  $\mathcal{T}$  from  $\mathcal{H}$  by removing Loss from  $\epsilon((p(v), v))$  and adding the content that was lost in this event in the content of every descendant node of  $v$  containing the gene family  $g$ . Therefore,  $c(\mathcal{H}^*) = c(\mathcal{H}) - \delta_{\text{loss}}$  and  $\mathcal{H} \notin \text{argmin}_{\mathcal{H} = \langle \dot{T}, \epsilon, x \rangle \in \mathbb{H}(\mathcal{T})} \{c(\mathcal{H})\}$ . Furthermore,  $c(\mathcal{H}_{p(v)}^*) = c(\mathcal{H}_{p(v)}) - \delta_{\text{loss}}$  and  $x^*(p(v)) = x(p(v))$  by construction and thus  $c(\mathcal{H}_{p(v)}) \notin c(p(v), x(p(v)))$ .  $\square$

## Proofs

### Proof of Lemma 1

*Proof.* Let  $\mathcal{H} = \langle \dot{T}, \epsilon, x \rangle \in \mathbb{H}(\mathcal{T})$  and let  $v$  be a node of  $T$ . Let  $g \in \mathcal{F}$ .

If  $\text{lca}(g) \leq v$ , then  $\text{Gain}_{\mathcal{H}}(g) \leq v$ , and if  $g \in \tilde{x}(l)$  for a given  $l \in L(T_v)$ , then it cannot be lost on the path  $P_T(\text{Gain}_{\mathcal{H}}(g), l)$  containing  $v$  by condition (3) of Definition 1. Therefore, for any  $g \in x_{\min}(v)$ ,  $g \in x(v)$ .

The result follows from the fact that for any  $g \notin x_{\min}(v)$ , either  $\text{lca}(g) > v$  in which case  $g \in x_{\text{in}}(v)$ , or  $g \notin \cup_{l \in L(T_v)} \tilde{x}(l)$  in which case  $g \in x_{\text{out}}(v)$ .  $\square$

### Proof of Lemma 2

*Proof.* Let  $v$  be an internal node of  $T$  and let  $v_c$  be a child of  $v$ . Then for any history  $\mathcal{H} = \langle \dot{T}, \epsilon, x \rangle$  explaining  $\mathcal{T}$ ,  $x(v_c) = x_{\min}(v_c) \cup O_c \cup I_c$  for some sets  $O_c \subseteq x_{\text{out}}(v_c)$  and  $I_c \subseteq x_{\text{in}}(v_c)$  by Lemma 1. For each  $\Delta_{\sigma}(v_c)$  ( $\sigma \in \{\text{min}, \text{out}, \text{in}, \text{in out}\}$ ), we will test each possible case of the sets  $O_c$  and  $I_c$  being empty or not:

- $\Delta_{\min}(v)$ : In this case,  $x(v) = x_{\min}(v)$ . Then,
  - If  $O_c = \emptyset$  and  $I_c = \emptyset$ : In that case,  $x(v_c) = x_{\min}(v_c)$ . The cost of an optimal subhistory rooted at  $v_c$  is therefore  $c_{\min}(v_c)$  by Definition 4. By Definition 1, Loss  $\in (v, v_c)$  if and only if  $x_{\min}(v) \not\subseteq x_{\min}(v_c)$  and Gain  $\in (v, v_c)$  if and only if  $x_{\min}(v_c) \not\subseteq x_{\min}(v)$ . This leads to a hanging subhistory of cost  $c_{\min}(v_c) + l(v, v_c) \cdot \delta_{\text{loss}} + g(v, v_c) \cdot \delta_{\text{gain}}$  (line 1a). See Figure 2 (1) for an illustration.
  - If  $O_c \neq \emptyset$  and  $I_c = \emptyset$ : This case is valid if and only if  $x_{\min}(v) \not\subseteq x_{\min}(v_c)$  as otherwise there exists no history explaining  $\mathcal{T}$  such that  $x(v) = x_{\min}(v)$  and  $x(v_c) = x_{\min}(v_c) \cup O_c$  for a non empty set  $O_c \subseteq x_{\text{out}}(v_c)$ . If this is the case, the cost of an optimal subhistory rooted at  $v_c$  is  $c_{\text{out}}(v_c)$  by Lemma 4. Loss  $\notin (v, v_c)$  by Lemma 6 and Gain  $\in (v, v_c)$  if and only if  $x_{\min}(v_c) \not\subseteq x_{\min}(v)$  by Definition 1. This leads to a hanging subhistory of cost  $c_{\text{out}}(v_c) + g(v, v_c) \cdot \delta_{\text{gain}}$  (line 1b). See Figure 2 (3) for an illustration.
  - If  $O_c = \emptyset$  and  $I_c \neq \emptyset$ : In that case, the cost of an optimal subhistory rooted at  $v_c$  is  $c_{\text{in}}(v_c)$  by Definition 4. Gain  $\in (v, v_c)$  because the content in  $I_c \subseteq x_{\text{in}}(v_c)$  is not in  $x(v) = x_{\min}(v)$  by definition

and therefore  $x(v_c) \not\subseteq x(v)$ . Loss  $\in (v, v_c)$  if and only if  $x_{\min}(v) \not\subseteq x_{\min}(v_c)$  by Definition 1. This leads to a hanging subhistory of cost  $c_{\text{in}}(v_c) + l(v, v_c) \cdot \delta_{\text{loss}} + \delta_{\text{gain}}$  (line 1c). See Figure 2 (2) for an illustration.

- If  $O_c \neq \emptyset$  and  $I_c \neq \emptyset$ : This case is valid if and only if  $x_{\min}(v) \not\subseteq x_{\min}(v_c)$  as otherwise there exists no history explaining  $\mathcal{T}$  such that  $x(v) = x_{\min}(v)$  and  $x(v_c) = x_{\min}(v_c) \cup O_c \cup I_c$  for some non empty sets  $O_c \subseteq x_{\text{out}}(v_c)$  and  $I_c \subseteq x_{\text{in}}(v_c)$ . If this is the case, the cost of the optimal subhistory rooted at  $v_c$  is  $c_{\text{in out}}(v_c)$  by Lemma 4. Gain  $\in (v, v_c)$  because the content in  $I_c \subseteq x_{\text{in}}(v_c)$  is not in  $x(v) = x_{\min}(v)$  by definition and therefore  $x(v_c) \not\subseteq x(v)$ . Loss  $\notin (v, v_c)$  by Lemma 6. This leads to a hanging subhistory of cost  $c_{\text{in out}}(v_c) + \delta_{\text{gain}}$  (line 1d).
- $\Delta_{\text{out}}(v)$ : In this case,  $x(v) = x_{\min}(v) \cup O$  with  $\emptyset \neq O \subseteq x_{\text{out}}(v_c) - x_{\min}(v)$ . This case is similar to the previous case ( $\Delta_{\text{in}}(v)$ ) with the difference that here we are sure that there is some content at  $v$  that is not in  $x_{\min}(v_c) \cup x_{\text{in}}(v_c)$  (Definition 3). Thus, everywhere in the equation for  $\Delta_{\text{in}}(v)$  that we test if there is a need to lose some content between  $v$  and  $v_c$ , we know this is the case here and therefore the result is obtained by replacing  $l(v, v_c)$  by 1 in the equation for  $\Delta_{\text{min}}(v)$ . See Figure 6 for an illustration of some of the cases.
- $\Delta_{\text{in}}(v)$ : In this case,  $x(v) = x_{\min}(v) \cup I$  with  $\emptyset \neq I \subseteq x_{\text{in}}(v_c) \cup x_{\text{lca}}(v_c)$ . This case is similar to the case  $\Delta_{\text{min}}(v)$ , with the difference that Gain  $\notin (v, v_c)$  by Lemma 5 and that the case  $I_c = \emptyset$  is only allowed if  $x_{\min}(v_c) \not\subseteq x_{\min}(v)$  as otherwise it would be impossible that  $x(v) = x_{\min}(v) \cup I$  where  $I \subseteq x_{\text{in}}(v_c) \cup x_{\text{lca}}(v_c)$  is non empty. See Figure 7 for an illustration of some of the cases.
- $\Delta_{\text{in out}}(v)$ : In this case,  $x(v) = x_{\min}(v) \cup O \cup I$  with  $\emptyset \neq O \subseteq x_{\text{out}}(v_c) - x_{\min}(v)$  and  $\emptyset \neq I \subseteq x_{\text{in}}(v_c) \cup x_{\text{lca}}(v_c)$ . This case is similar to the previous case ( $\Delta_{\text{in}}(v)$ ) with the difference that here we are sure that there is some content at  $v$  that is not in  $x_{\min}(v_c) \cup x_{\text{in}}(v_c)$  (Definition 3). Thus, everywhere in the equation for  $\Delta_{\text{in}}(v)$  that we test if there is a need to lose some content between  $v$  and  $v_c$ , we know this is the case here and therefore the result is obtained by replacing  $l(v, v_c)$  by 1 in the equation for  $\Delta_{\text{in}}(v)$ .

□

### Proof of Theorem 1

*Proof.* Let  $v \in V(T)$ .

If  $v$  is a leaf, then the results follows from Definitions 1 and 4. In fact, there is no history  $\mathcal{H} = \langle \dot{T}, \epsilon, x \rangle$  explaining  $\mathcal{T}$  such that  $x(v) \neq x_{\min}(v)$  and thus  $c_{\text{in}}(v) = c_{\text{out}}(v) = c_{\text{in out}}(v) = \infty$ .

Let  $v$  be an internal node and let show the result for every possible state.

For  $c_{\text{min}}(v)$ ,  $x(v) = x_{\min}(v)$  by definition. Then, either  $c_{\text{min}}(v) = \infty$  if  $x_{\min}(v) = \emptyset$  by Definition 1 or  $c_{\text{min}}(v) = \Delta_{\text{min}}(v_\ell) + \Delta_{\text{min}}(v_r)$  by Definition 4.

For  $c_{\text{out}}(v)$ ,  $x(v) = x_{\min}(v) \cup O$  with  $\emptyset \neq O \subseteq x_{\text{out}}(v)$  by definition. In this case,  $O \subseteq x_{\text{out}}(v_c) - x_{\min}(v)$  for  $c \in \{\ell, r\}$  by definition and thus  $c_{\text{out}}(v) = \Delta_{\text{out}}(v_\ell) + \Delta_{\text{out}}(v_r)$ .

For  $c_{\text{in}}(v)$ ,  $x(v) = x_{\min}(v) \cup I$  with  $\emptyset \neq I \subseteq x_{\text{in}}(v)$  by definition. In this case, either  $I \subseteq x_{\text{in}}(v_c) \cup x_{\text{lca}}(v_c)$  and  $I \subseteq x_{\text{out}}(v_{c'}) - x_{\min}(v)$  for  $c \neq c'$ ,  $c, c' \in \{\ell, r\}$  or  $I \cap (x_{\text{in}}(v_c) \cup x_{\text{lca}}(v_c)) \neq \emptyset$  and  $I \cap (x_{\text{out}}(v_c) - x_{\min}(v)) \neq \emptyset$  for both  $c = \ell$  and  $c = r$ . Therefore,  $c_{\text{in}}(v) = \min\{\Delta_{\text{in}}(v_\ell) + \Delta_{\text{out}}(v_r), \Delta_{\text{out}}(v_\ell) + \Delta_{\text{in}}(v_r), \Delta_{\text{in out}}(v_\ell) + \Delta_{\text{in out}}(v_r)\}$ .

For  $c_{\text{in out}}(v)$ ,  $x(v) = x_{\min}(v) \cup O \cup I$  with  $\emptyset \neq O \subseteq x_{\text{out}}(v)$  and  $\emptyset \neq I \subseteq x_{\text{in}}(v)$  by definition. Here,  $O \subseteq x_{\text{out}}(v_c) - x_{\min}(v)$  for  $c \in \{\ell, r\}$  by definition and either  $I \subseteq x_{\text{out}}(v_c) - x_{\min}(v)$  and  $I \subseteq x_{\text{in}}(v_{c'}) \cup x_{\text{lca}}(v_{c'})$  for  $c \neq c'$ ,  $c, c' \in \{\ell, r\}$  or  $I \cap (x_{\text{in}}(v_c) \cup x_{\text{lca}}(v_c)) \neq \emptyset$  for both  $c = \ell$  and  $c = r$ . Therefore,  $c_{\text{in}}(v) = \min\{\Delta_{\text{in out}}(v_\ell) + \Delta_{\text{out}}(v_r), \Delta_{\text{out}}(v_\ell) + \Delta_{\text{in out}}(v_r), \Delta_{\text{in out}}(v_\ell) + \Delta_{\text{in out}}(v_r)\}$ .  $\square$

### Proof of Lemma 3

*Proof.* We first show that Algorithm 1 returns  $x_{\text{lca}}(v)$  for all  $v \in V(T)$ . Notice that after the first **for** loop,  $A(v) = \cup_{l \in L(T_v)} \tilde{x}(l)$  for all  $v \in L(T)$ . We now show that in the second **for** loop,  $B(v)$  is set to  $\{g \in \mathcal{F} \mid \text{lca}(g) \geq v\}$  for all  $v \in V(T)$  by induction on the depth of  $v$ . If  $v = r(T)$ , then  $\{g \in \mathcal{F} \mid \text{lca}(g) \geq v\} = \cup_{l \in L(T_v)} \tilde{x}(l) = A(v)$ . Hence,  $B(r(T)) = A(r(T))$ . Now, we suppose that  $B(v) = \{g \in \mathcal{F} \mid \text{lca}(g) \geq v\}$  for  $v \in V(T) - L(T)$  by induction hypothesis (IH) and we want to show that  $B(v_c) = \{g \in \mathcal{F} \mid \text{lca}(g) \geq v_c\}$  for  $v_c \in \text{ch}(v)$ . Without loss of generality, we show the result for  $v_\ell$ . Notice that if  $g \in \{g \in \mathcal{F} \mid \text{lca}(g) \geq v_r \vee \text{lca}(g) = v\}$ , then  $g \in \cup_{l \in L(T_{v_r})} \tilde{x}(l)$  because if  $\text{lca}(g)$  is a descendant of  $v_r$ , or  $\text{lca}(g) = v$  there must be at least one leaf  $l$  in  $T_{v_r}$  such that  $g \in \tilde{x}(l)$  by definition of  $\text{lca}(g)$ . Therefore,

$$\{g \in \mathcal{F} \mid \text{lca}(g) \geq v_r \vee \text{lca}(g) = v\} - \cup_{l \in L(T_{v_r})} \tilde{x}(l) = \emptyset \quad (5)$$

Also notice that

$$\{g \in \mathcal{F} \mid \text{lca}(g) \geq v_\ell\} \cap \cup_{l \in L(T_{v_r})} \tilde{x}(l) = \emptyset \quad (6)$$

because if  $g \in \cup_{l \in L(T_{v_r})} \tilde{x}(l)$ , then  $\text{lca}(g)$  cannot be a descendant of  $v_\ell$ . Then,

$$\begin{aligned} B(v) - A(v_r) &= \{g \in \mathcal{F} \mid \text{lca}(g) \geq v\} - \cup_{l \in L(T_{v_r})} \tilde{x}(l) && \text{by IH} \\ &= (\{g \in \mathcal{F} \mid \text{lca}(g) \geq v_\ell\} \\ &\quad \cup \{g \in \mathcal{F} \mid \text{lca}(g) \geq v_r \vee \text{lca}(g) = v\}) - \cup_{l \in L(T_{v_r})} \tilde{x}(l) \\ &= \{g \in \mathcal{F} \mid \text{lca}(g) \geq v_\ell\} && \text{by (5) and (6)} \end{aligned}$$

Thus,  $B(v_\ell) = \{g \in \mathcal{F} \mid \text{lca}(g) \geq v_\ell\}$ . By definition of  $x_{\text{lca}}$ ,  $x_{\text{lca}}(v) = B(v)$  if  $v$  is a leaf and  $x_{\text{lca}}(v) = B(v) - B(v_\ell) - B(v_r)$  otherwise. The result follows.

We now show that Algorithm 2 returns  $x_{\min}(v)$  for all  $v \in V(T)$ . Let  $v \in V(T)$ . If  $v \in L(T)$ , then for any  $g \in \tilde{x}(v)$ ,  $\text{lca}(g) \leq v$  thus  $g \in x_{\min}(v)$ , and conversely for any  $g \in x_{\min}(v)$ ,  $g$  should appear in the subtree rooted at  $v$  which is  $v$  itself and thus  $g \in \tilde{x}(v)$ . Now suppose  $v \notin L(T)$ . Let  $g \in x_{\min}(v)$ , thus  $\text{lca}(g) \leq v$  and  $g \in \cup_{l \in L(T_{v_\ell})} \tilde{x}(l)$  or  $g \in \cup_{l \in L(T_{v_r})} \tilde{x}(l)$ . In the first case,

$g \in x_{\min}(v_\ell)$  but as  $lca(g) < v_\ell$ ,  $g \in x_{\min}(v_\ell) - x_{lca}(v_\ell)$ . In the second case,  $g \in x_{\min}(v_r)$  but as  $lca(g) < v_r$ ,  $g \in x_{\min}(v_r) - x_{lca}(v_r)$ . Conversely, for any  $c \in \{\ell, r\}$  and  $g \in x_{\min}(v_c) - x_{lca}(v_c)$ , we have  $lca(g) \leq v$  and  $g \in \cup_{l \in L(T_v)} \tilde{x}(l)$ , thus  $g \in x_{\min}(v)$ .  $\square$

### Proof of Corollary 1

*Proof.* Notice that for any history  $\mathcal{H} = \langle \dot{T}, \epsilon, x \rangle$ ,  $\text{Gain} \in \epsilon((r(\dot{T}), r(\dot{T})_c))$  because otherwise point (4) of Definition 1 would not be respected. Also notice that  $x_{\text{out}}(r(T)) = \emptyset$  by definition and therefore the content of the node  $r(T)$  in any history explaining  $T$  is  $x_{\min}(r(T)) \cup I$  for some set  $I \subseteq x_{\text{in}}(r(T))$  by Lemma 1. Therefore, the cost of the optimal subhistory rooted at  $r(T)$  explaining  $\mathcal{T}$  is either  $c_{\min}(r(T))$  or  $c_{\text{in}}(r(T))$  and the result follows.  $\square$

### Proof of Theorem 2

*Proof.* As  $\epsilon$  is given in input and lead to an optimal cost by hypothesis, it is sufficient to show that  $\langle \dot{T}, \epsilon, x \rangle \in \mathbb{H}(\mathcal{T})$ . Notice that  $x(v) \in \mathcal{P}(\mathcal{F})$  for  $v \in V(\dot{T})$  by construction because the only gene families appearing in the history are the genes families appearing at the leaves of  $T$ . We now need to show that  $\langle \dot{T}, \epsilon, x \rangle$  respects the conditions of Definition 1.

- Condition (1): Notice that for each leaf  $l \in L(T)$ ,  $x(l)$  is set to  $x_{\min}(l)$  on line 2. By Lemma 3,  $x_{\min}(l) = \tilde{x}(l)$  because  $l$  is leaf. We now show that  $x(l)$  is not modified afterward on line 10. Let us assume, for the sake of contradiction, that there is a leaf  $l \in L(T)$  such that  $x(l)$  is modified on line 10. In that case,  $\text{Loss} \notin \epsilon((p(l), l))$  and  $x(p(l)) \not\subseteq \tilde{x}(l)$ . Thus, there exists a gene family  $g \in x(p(l))$  such that  $g \notin \tilde{x}(l)$ . There are three possible cases as to why  $g$  is in  $x(p(l))$ :
  - Case (1):  $g \in x_{\min}(p(l))$ . In that case, for any history  $\mathcal{H} = \langle \dot{T}, \epsilon, x \rangle$  explaining  $\mathcal{T}$ ,  $g \in x(p(l))$  by Lemma 1. As  $\text{Loss} \notin \epsilon((p(l), l))$ ,  $g \in x(l)$  by Definition 1. But  $g \notin \tilde{x}(l)$  and we conclude that there exists no history  $\mathcal{H} = \langle \dot{T}, \epsilon, x \rangle$  explaining  $\mathcal{T}$  which is a contradiction.
  - Case (2):  $g$  was added to  $x(p(l))$  on either line 5 or line 7. In that case, notice that for any history  $\mathcal{H} = \langle \dot{T}, \epsilon, x \rangle$  explaining  $\mathcal{T}$ ,  $g$  must be in  $x(p(l))$  because in that case there is some leaf  $l'$  descendant of  $p(l)$  such that  $g \in x(l')$  and there is no gain event in the path from  $p(l)$  to  $l'$  allowing to gain  $g$ . As  $\text{Loss} \notin \epsilon((p(l), l))$ ,  $g \in x(l)$ . But  $g \notin \tilde{x}(l)$  and we conclude that there exists no history  $\mathcal{H} = \langle \dot{T}, \epsilon, x \rangle$  explaining  $\mathcal{T}$  which is a contradiction.
  - Case (3):  $g$  was added to  $x(p(l))$  on line 10. Here,  $\text{Loss} \notin \epsilon((p(p(l)), p(l)))$  and we can repeat the same argument as to why  $g$  is in  $p(p(l))$  (and thus in  $p(l)$  and in  $l$ ) until we get to the root to obtain a contradiction. If  $p(p(l)) = r(T)$ , then only cases (1) and (2) apply, leading to a contradiction.

Therefore, for each leaf  $l \in L(T)$ ,  $x(l)$  equals  $x_{\min}(l)$ . By Lemma 3,  $x_{\min}(l) = \tilde{x}(l)$  because  $l$  is a leaf and thus, condition (1) is respected.

- Condition (2): Let  $(v, v_c) \in E(\dot{T})$ . If  $\text{Loss} \notin \epsilon((v, v_c))$ , then all the content of  $v$  is added to  $v_c$  on line 10 and afterward the contents of  $v$  and  $v_c$  are not modified. Therefore,  $x(v) \subseteq x(v_c)$ . Otherwise, if  $\text{Loss} \in \epsilon((v, v_c))$ , the fact that  $x(v) \not\subseteq x(v_c)$  comes from the fact that  $\epsilon$  leads to an optimal cost by hypothesis. In fact, if  $x(v) \subseteq x(v_c)$  and  $\text{Loss} \in \epsilon((v, v_c))$  in the output of Algorithm 4, we can obtain a new history  $\langle \dot{T}, \epsilon', x' \rangle$  explaining  $\mathcal{T}$  by simply removing the useless loss. This new history has a lower cost than any history  $\langle \dot{T}, \epsilon, x \rangle$  and thus  $\epsilon$  does not lead to an optimal cost, which is a contradiction. We conclude that condition 2.(a) is respected. If  $\text{Gain} \notin \epsilon((v, v_c))$ , then all the content of  $v_c$  is added to  $v$  on either line 5 or line 7 and afterward the only possible modification to the content of  $v$  is to add some new content on line 10 and then the only possible modification to the content of  $v_c$  is to add some new content on line 10, but this content is already in  $p(v)$ . Thus,  $x(v_c) \subseteq x(v)$ . Otherwise, if  $\text{Gain} \in \epsilon((v, v_c))$ , the fact that  $x(v_c) \not\subseteq x(v)$  comes from the fact that  $\epsilon$  leads to an optimal cost by hypothesis, similarly to the previous case. Therefore, condition 2.(b) is respected.
- Condition (3): Let  $g \in \mathcal{F}$ . Let's first show that there is a node  $v \in V(\dot{T}) - \{r(\dot{T})\}$  such that  $g \in x(v) - x(p(v))$ . By construction, the first node  $v$  on the path from  $\text{lca}(g)$  to  $r(T)$  such that  $\text{Gain} \in \epsilon(p(v), v)$  is such that  $g \in x(v) - x(p(v))$ . If there is no such node, then there exists no history  $\langle \dot{T}, \epsilon, x \rangle$  explaining  $\mathcal{T}$  because there is then no gain event on the edges of the path from  $\text{lca}(g)$  to  $r(T)$  and thus  $\text{Gain} \notin \epsilon(r(T), r(T)_c)$  and then it is impossible for Conditions (2) and (4) of Definition 1 to be respected simultaneously. This is a contradiction and we conclude that such a node must exist. We now show that this node  $v$  is unique. Let  $v' \in V(\dot{T}) - \{r(\dot{T})\}$  different from  $v$ . If  $v'$  is a strict ancestor of  $v$ , then  $g \notin x(v')$  and  $g \notin x(p(v'))$  by construction. Otherwise, if  $v'$  is on the path from  $\text{lca}(g)$  (included) to  $v$  (excluded), then there is no gain on the edges of this path and thus  $g \in x(v')$  and  $g \in x(p(v'))$  by construction. Otherwise, if  $v'$  is strict descendant of  $\text{lca}(g)$  such that  $g \in \cup_{l \in L(T_{v'})} x(l)$ , then  $g \in x_{\min}(v')$  and  $g \in x_{\min}(p(v'))$  by definition and thus  $g \in x(v')$  and  $g \in x(p(v'))$  by construction. Finally, if  $v'$  is such that  $g \notin \cup_{l \in L(T_{v'})} x(l)$ , then  $g \in x(v')$  if and only if  $g \in x(p(v'))$  by construction. In all cases, we conclude that  $g \notin x(v') - x(p(v'))$ .
- Condition (4): Notice that  $x(r(\dot{T}))$  is set to  $\emptyset$  on line 11. Let us assume, for the sake of contradiction, that there is a node in  $V(\dot{T}) - \{r(\dot{T})\}$  with an empty synteny content. Let  $v$  be the highest node in  $\dot{T}$  different from the root such that this is the case (i.e.  $x(v) = \emptyset$ ). First notice that  $x_{\min}(v) = \emptyset$  as otherwise  $x(v)$  would not be empty by construction. Also notice that for  $v_c \in \{v_\ell, v_r\}$ , either  $x(v_c) = \emptyset$  or  $\text{Gain} \in \epsilon((v, v_c))$  as otherwise  $x(v)$  would not be empty by construction. If  $\text{Gain} \in \epsilon((v, v_c))$ , then  $x(v) \cap (x_{\text{in}}(v_c) \cup x_{\text{lca}}(v_c)) = \emptyset$  by Lemma 5 otherwise  $\epsilon$  would not lead to an optimal history. If  $x(v_c) = \emptyset$ , then we can repeat the same argument (i.e. for  $v_{c_c} \in \{v_{c_\ell}, v_{c_r}\}$  either  $x(v_{c_c}) = \emptyset$  or  $\text{Gain} \in \epsilon((v_c, v_{c_c}))$ ) until we get to a leaf  $l$  for which we know that  $\text{Gain} \in \epsilon((p(l), l))$  because  $x(l) \neq \emptyset$  by construction. In both case, we conclude that

$x(v) \cap (x_{\text{in}}(v_c) \cup x_{\text{lca}}(v_c)) = \emptyset$  and we can deduce that  $x(v) \cap x_{\text{in}}(v) = \emptyset$  for any history explaining  $\mathcal{T}$  with events  $\epsilon$  (given that  $\epsilon$  leads to an optimal history). As  $v$  is the highest node in  $\dot{T}$  different from the root such that  $x(v) = \emptyset$ , either  $v$  is the child of the root or  $\text{Loss} \in \epsilon((p(v), v))$  as otherwise  $x(v)$  would not be empty by construction. Thus,  $x(v) \cap x_{\text{out}}(v) = \emptyset$  by Lemma 6 otherwise  $\epsilon$  would not lead to an optimal history. But it is impossible that simultaneously  $x_{\text{min}}(v) = \emptyset$ ,  $x(v) \cap x_{\text{out}}(v) = \emptyset$  and  $x(v) \cap x_{\text{in}}(v) = \emptyset$ . Therefore,  $\epsilon$  does not lead to an optimal history, which is a contradiction. Thus, we conclude that condition (4) is respected.

□

### Proof of Theorem 3

*Proof.* Let  $\mathcal{H} = \langle \dot{T}, \epsilon, x \rangle \in \text{argmin}_{\mathcal{H} \in \mathbb{H}(\mathcal{T})} \{c(\hat{\mathcal{H}})\}$ . Let us assume, for the sake of contradiction, that there exists a history  $\mathcal{H}' = \langle \dot{T}, \epsilon, x' \rangle \in \mathbb{H}(\mathcal{T})$  such that  $x' \neq x$ . Let  $v$  be the highest node in  $\dot{T}$  such that  $x(v) \neq x'(v)$ . Notice that  $v \neq \text{r}(\dot{T})$  because  $x(\text{r}(\dot{T})) = \emptyset$  and  $x'(\text{r}(\dot{T})) = \emptyset$  by Definition 1. We will show that a contradiction arises in each possible cases for the set  $\epsilon((p(v), v))$ :

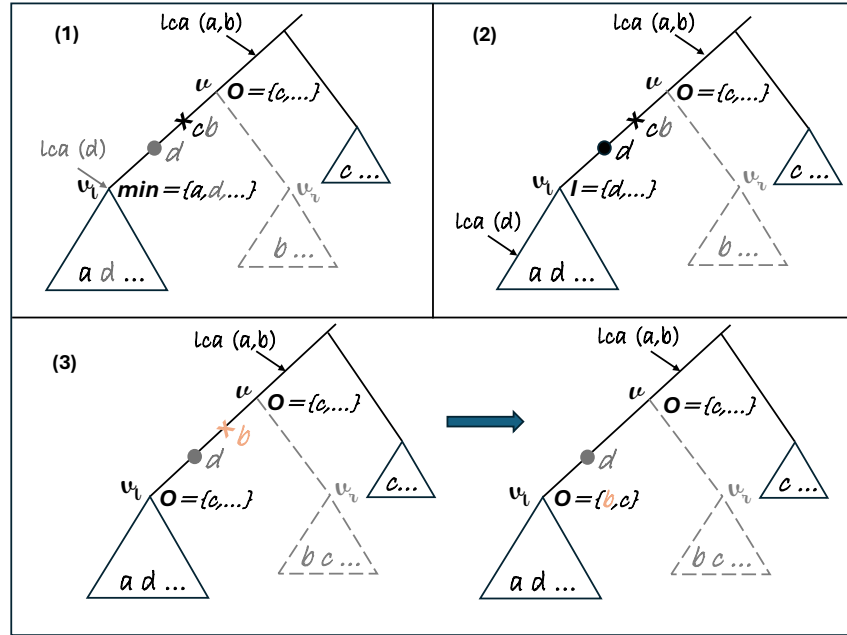
- $\epsilon((p(v), v)) = \emptyset$ : In this case,  $x(v) = x(p(v))$  and  $x'(v) = x'(p(v))$  by Definition 1. But as  $x(p(v)) = x'(p(v))$ , this implies that  $x(v) = x'(v)$  which is a contradiction.
- $\epsilon((p(v), v)) = \{\text{Loss}\}$ : In this case,  $x(v) \subsetneq x(p(v))$  and  $x'(v) \subsetneq x'(p(v))$  by Definition 1. By lemma 6,  $x(v) \cap x_{\text{out}}(v) = \emptyset$  and  $x'(v) \cap x_{\text{out}}(v) = \emptyset$ . Notice that any gene family  $g \in x_{\text{min}}(v) \cup x_{\text{in}}(v)$  such that  $g \in x(p(v))$  cannot be lost by the event on the edge  $(p(v), v)$  because each gene family is gained only once in a history by Definition 1. As  $x(p(v)) = x'(p(v))$  and as for each  $g \in \mathcal{F}$ , either  $g \in x_{\text{min}}(v) \cup x_{\text{in}}(v)$  or  $g \in x_{\text{out}}(v)$  by definition,  $x(v) = x'(v)$  which is a contradiction.
- $\epsilon((p(v), v)) = \{\text{Gain}\}$ : As  $x(v) \neq x'(v)$ , there is at least one gene family in either  $x(v)$  or  $x'(v)$  that is not in the other set. Without loss of generality, we will assume that there exists a gene family  $g \in x(v)$  such that  $g \notin x'(v)$ . As  $x(p(v)) = x'(p(v))$  this implies that  $\text{Gain}_{\mathcal{H}}(g) = v$ . Thus,  $\text{lca}(g) \geq v$  because each gene family is gained only once in a history by Definition 1. Therefore,  $\text{Gain}_{\mathcal{H}'}(g) > v$  because  $g \notin x'(v)$ . Let  $w = \text{Gain}_{\mathcal{H}'}(g)$ . This implies that  $\text{lca}(g) \geq w$  and thus there is at least one loss event in  $\mathcal{H}'$  (and  $\mathcal{H}$ ) in each of the paths from  $v$  to a leaf  $l \in L(\dot{T}_v) - L(\dot{T}_w)$ . Therefore, we can construct a new history  $\mathcal{H}^*$  explaining  $\mathcal{T}$  from  $\mathcal{H}$  by removing the gain event on the edge  $(p(w), w)$  and adding the content that was gained in this event in the content of every node containing the gene family  $g$  that is not a descendant of  $w$ . Thus,  $c(\mathcal{H}^*) = c(\mathcal{H}) - \delta_{\text{gain}}$  and  $\mathcal{H} \notin \text{argmin}_{\mathcal{H} \in \mathbb{H}(\mathcal{T})}$  which is a contradiction.
- $\epsilon((p(v), v)) = \{\text{Gain}, \text{Loss}\}$ : As shown in the case  $\epsilon((p(v), v)) = \{\text{Loss}\}$ , any gene family  $g \in x_{\text{out}}(v)$  must be lost in the loss event on the edge  $(p(v), v)$  and any gene family not in  $x_{\text{out}}(v)$  cannot be lost in the loss event on the edge  $(p(v), v)$ . Therefore, if  $x(v) \neq x'(v)$ , there must be at least one gene

family in either  $x(v)$  or  $x'(v)$  that is not in the other set and that is gained at  $v$ . We can thus use the same argument as in the case  $\epsilon((p(v), v)) = \{\text{Gain}\}$ .  $\square$

### Proof of Theorem 5

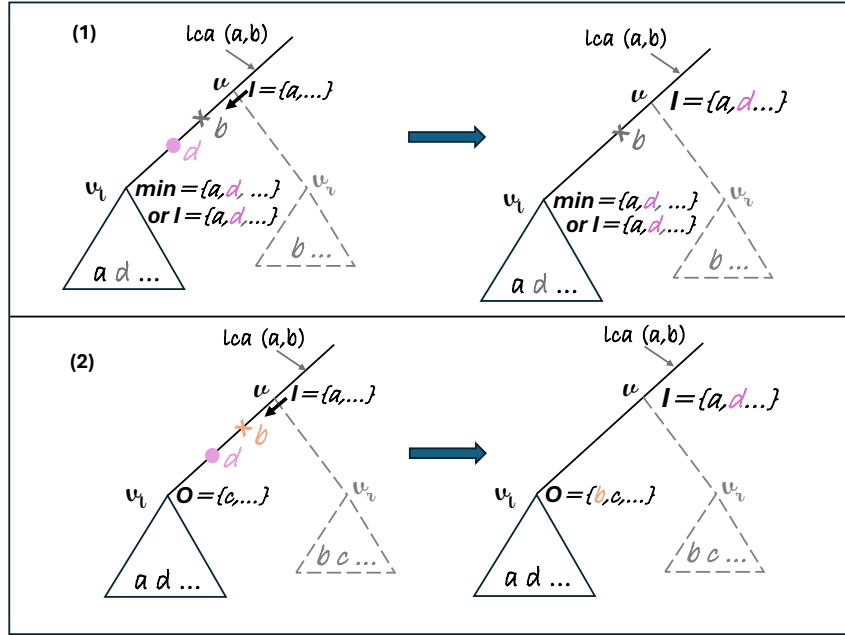
*Proof.* This result follows from Lemma 3, Theorems 1 and 2, Corollary 1 and from the fact that each step of the algorithm consists in a constant number of traversal of  $T$  in which each node is processed in  $O(1)$  time because we suppose by hypothesis that operations on sets of gene families can be done in  $O(1)$  time.  $\square$

### Additional Figures

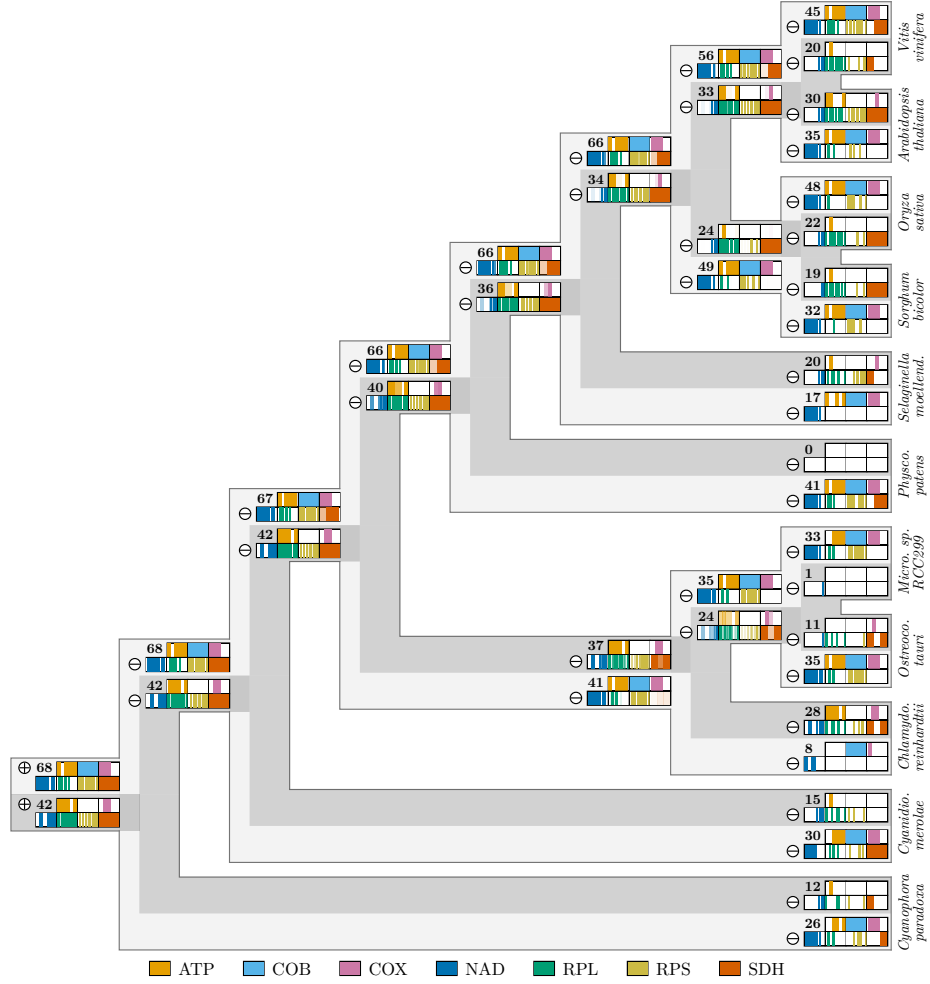


**Fig. 6.** An illustration of a left scenario (of cost  $\Delta_{\text{out}}(v_\ell)$ ) leading to  $c_{\text{out}}(v)$  (a similar scenario can be drawn for the right part, i.e. for  $\Delta_{\text{out}}(v_r)$ ); the characters in a triangle represent the gene families in the leaves of the corresponding subtree; the events and characters in gray can be present or absent, depending on the value of  $g(v, v_\ell)$ . (1): A scenario with  $x(v_\ell) = x_{\min}(v_\ell)$  (line 2a) (2): A scenario with  $x(v_\ell) = x_{\min}(v_\ell) \cup I$  where  $I \subseteq x_{\text{in}}(v_\ell)$  (line 2c) (3): Two scenarios with  $x(v_\ell) = x_{\min}(v_\ell) \cup O$  where  $O \subseteq x_{\text{out}}(v_\ell)$ ,  $O \neq \emptyset$  (line 2b). The left scenario is not optimal with a loss on the edge  $(v, v_\ell)$  as the lost gene (here  $b$ ) can always be postponed to be lost later in the same losses of the syntenies containing the extra gene  $c$ .

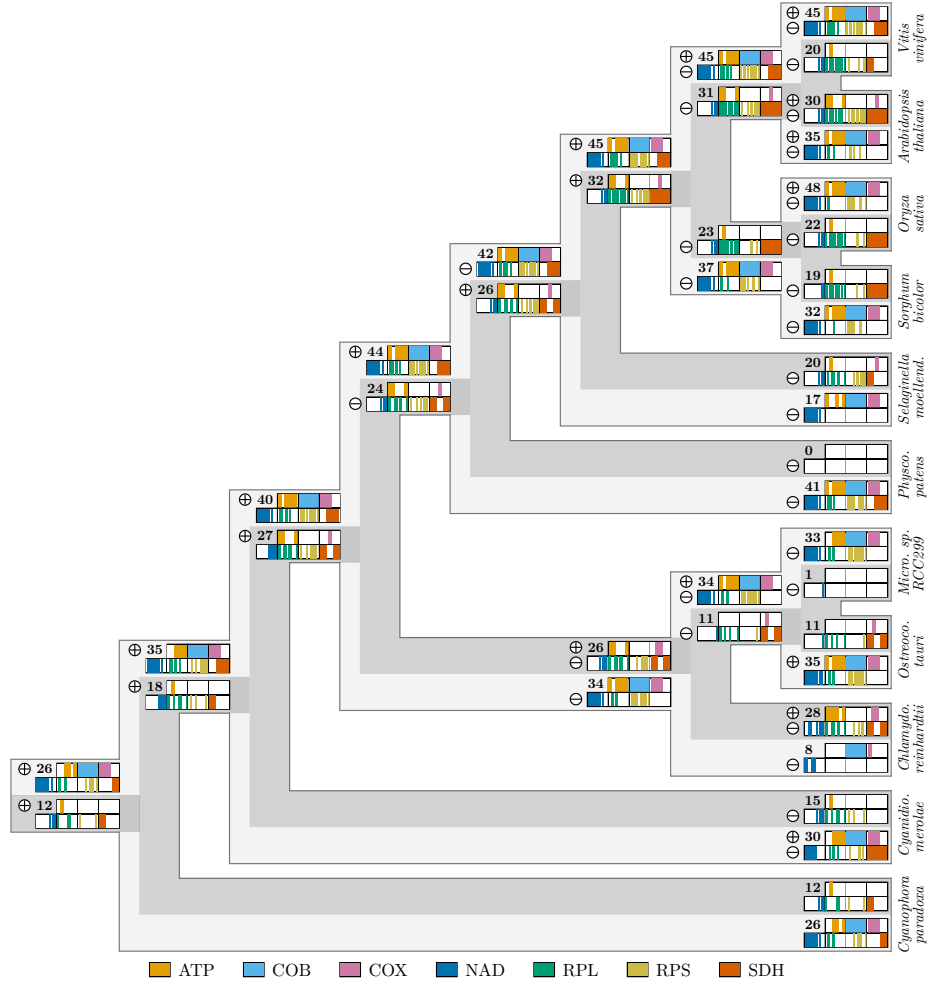




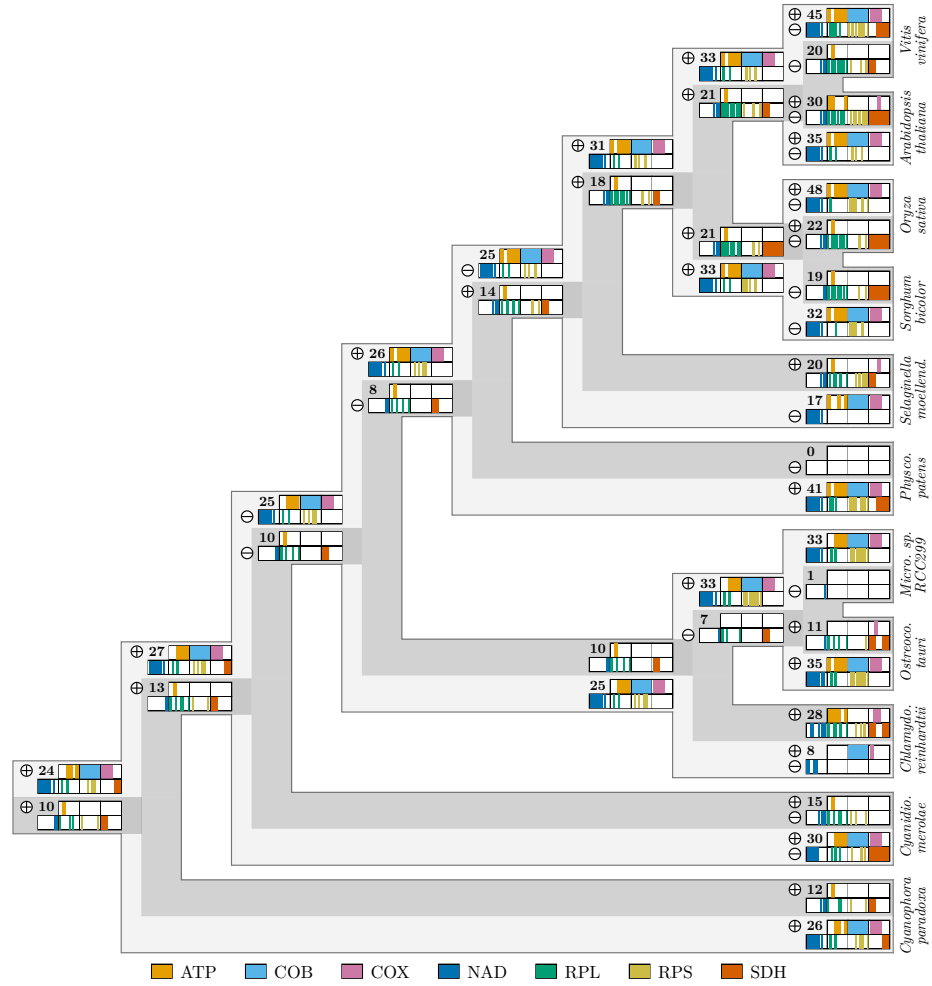
**Fig. 7.** An illustration of a left scenario (of cost  $\Delta_{in}(v_\ell)$ ) leading to  $c_{in}(v)$  in the case where the extra content  $I$  comes from the left side (a scenario similar to those of Figure 6 can be drawn for the right part, i.e. for  $\Delta_{out}(v_r)$ ); the characters in a triangle represent the gene families in the leaves of the corresponding subtree; the events and characters in gray can be present or absent, depending on the value of  $l(v, v_\ell)$ . (1): Two scenarios with either  $x(v_\ell) = x_{min}(v_\ell)$  or  $x(v_\ell) = x_{min}(v_\ell) \cup I$  where  $I \subseteq x_{in}(v_\ell)$  (line 3a or 3c). The left scenario is not optimal with a gain on the edge  $(v, v_\ell)$  as the gained gene (here  $d$ ) can always be gained earlier in this history at the same gain point as the gene  $a$ . (2): Two scenarios with  $x(v_\ell) = x_{min}(v_\ell) \cup O$  where  $O \subseteq x_{out}(v_\ell)$ ,  $O \neq \emptyset$  (line 3b). The left scenario is not optimal with a loss and a gain on the edge  $(v, v_\ell)$  as the lost gene (here  $b$ ) can always be postponed to be lost later in the same losses of the synteny containing the extra gene  $c$  and the gained gene (here  $d$ ) can always be gained earlier in this history at the same gain point as the gene  $a$ .



**Fig. 8.** Results obtained with Count for a probabilistic pure-loss model on the species tree of the considered 11 plants. Representation is the same as in Figure 4, except that vertical strips are represented with a level of transparency illustrating their inferred probability of presence.



**Fig. 9.** Results obtained with Count for the Dollo parsimony model on the species tree of the considered 11 plants. Representation is the same as in Figure 4.



**Fig. 10.** Results obtained with Count for the Wagner parsimony model with unitary costs on the species tree of the considered 11 plants. Representation is the same as in Figure 4.

## References

1. Angibaud, S., Fertin, G., Rusu, I., Thévenin, A., Vialette, S.: On the approximability of comparing genomes with duplicates. *Journal of Graph Algorithms and Applications* **13**(1), 19–53 (2009)
2. Anselmetti, Y., Delabre, M., El-Mabrouk, N.: Reconciliation with segmental duplication, transfer, loss and gain. In: Jin, L., Durand, D. (eds.) *Comparative Genomics*. pp. 124–145. Cham (2022)

3. Anselmetti, Y., El-Mabrouk, N., Lafond, M., Ouangraoua, A.: Gene tree and species tree reconciliation with endosymbiotic gene transfer. *Bioinformatics* **37**(SI-1), i120–i132 (2021)
4. Blin, G., Chauve, C., Fertin, G., Rizzi, R., Vialette, S.: Comparing genomes with duplications: A computational complexity point of view. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **4**(4), 523–534 (2007)
5. Bohnenkamper, L., Stoye, J., Doerr, D.: Reconstructing rearrangement phylogenies of natural genomes. *Algorithms for Molecular Biology* **20**(10) (2025)
6. Caprara, A.: The reversal median problem. *INFORMS Journal on Computing* **15**(1), 93–113 (2003)
7. Csűrös, M.: Ancestral reconstruction by asymmetric Wagner parsimony over continuous characters and squared parsimony over distributions. In: *Proceedings of the Sixth RECOMB Comparative Genomics Satellite Workshop. Lecture Notes in Bioinformatics*, vol. 5267, pp. 72–86 (2008)
8. Csűrös, M.: Count: evolutionary analysis of phylogenetic profiles with parsimony and likelihood. *Bioinformatics* **26**(15), 1910–2 (2010)
9. Csűrös, M.: Gain-loss-duplication models for copy number evolution on a phylogeny: Exact algorithms for computing the likelihood and its gradient. *Theoretical Population Biology* **145**, 80–94 (2022)
10. Csűrös, M., Miklós, I.: A probabilistic model for gene content evolution with duplication, loss, and horizontal transfer. In: *Proceedings of the Tenth Annual International Conference on Research in Computational Molecular Biology (RECOMB). Lecture Notes in Bioinformatics*, vol. 14899, pp. 206–220 (2006)
11. Delabre, M., El-Mabrouk, N.: Synesth: Comprehensive syntenic reconciliation with unsampled lineages. *Algorithms* **17**(5) (2024)
12. Doerr, D., Chauve, C.: Small parsimony for natural genomes in the DCJ-indel model. *Journal of Bioinformatics and Computational Biology* **19**(6) (2021)
13. Feijao, P., Meidanis, J.: SCJ: A breakpoint-like distance that simplifies several rearrangement problems. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **8**(5), 1318–1329 (2011)
14. Fitch, W.M.: Toward defining the course of evolution: minimum change for a specific tree topology. *Syst. Zool.* **20**(4), 406–416 (1971)
15. Gascon, M., Delabre, M., El-Mabrouk, N.: Fullsynesth: Syntenic reconciliation of a set of consistent gene trees. *Theory of Computing Systems* **70**(8) (2026)
16. Goodman, M., Czelusniak, J., Moore, G.W., Romero-Herrera, A.E., Matsuda, G.: Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences. *Systematic Zoology* **28**, 132–163 (1979)
17. Kannan, S., Rogozin, I., Koonin, E.: MitoCOGs: Clusters of orthologous genes from mitochondria and implications for the evolution of eukaryotes. *BMC Evolutionary Biology* **14**(11), 1–16 (2014)
18. Kovács, J.: On the complexity of rearrangement problems under the breakpoint distance. *Journal of Computational Biology* **21**(1), 1–15 (2014)
19. Luhmann, N., Lafond, M., Thévenin, A., Ouangraoua, A., Wittler, R., Chauve, C.: The SCJ small parsimony problem for weighted gene adjacencies. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **16**(4), 1364–1373 (2019)
20. Pe’er, I., Shamir, R.: The median problems for breakpoints are NP-complete. In: *Proceedings of the Electronic Colloquium on Computational Complexity. ECCC’98*, vol. 5 (1998)

21. Sankoff, D.: Minimal parsimony is not enough. *Syst. Zool.* **24**(2), 158–164 (1971)
22. Tannier, E., E., C.Z., Sankoff, D.: Multichromosomal median and halving problems under different genomic distances. *BMC Bioinformatics* **10**, 120 (2009)