algorithms

MDPI

Article

Synesth: Comprehensive Syntenic Reconciliation with Unsampled Lineages

Mattéo Delabre * and Nadia El-Mabrouk *

Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, 2920 Chemin de la Tour, Montréal, QC H3T 1J4, Canada

* Correspondence: matteo.delabre@umontreal.ca (M.D.); mabrouk@iro.umontreal.ca (N.E.-M.)

Abstract: We present Synesth, the most comprehensive and flexible tool for tree reconciliation that allows for events on *syntenies* (i.e., on sets of multiple genes), including duplications, transfers, fissions, and transient events going through unsampled species. This model allows for building *histories* that explicate the inconsistencies between a *synteny tree* and its associated *species tree*. We examine the combinatorial properties of this extended reconciliation model and study various associated parsimony problems. First, the infinite set of explicatory histories is reduced to a finite but exponential set of *Pareto-optimal* histories (in terms of counts of each event type), then to a polynomial set of Pareto-optimal event count vectors, and this eventually ends with minimum event cost histories given an event cost function. An inductive characterization of the solution space using different algebras for each granularity leads to efficient dynamic programming algorithms, ultimately ending with an $O(mn)$ time complexity algorithm for computing the cost of a minimum-cost history (m and n: number of nodes in the input synteny and species trees). This time complexity matches that of the fastest known algorithms for classical gene reconciliation with transfers. We show how Synesth can be applied to infer Pareto-optimal evolutionary scenarios for CRISPR-Cas systems in a set of bacterial genomes.

Keywords: tree reconciliation; algebraic dynamic programming; multi-objective optimization; horizontal gene transfer; synteny; fission; unsampled lineages

Citation: Delabre, M.; El-Mabrouk, N. Synesth: Comprehensive Syntenic Reconciliation with Unsampled Lineages. *Algorithms* **2024**, *17*, 186. https://doi.org/10.3390/a17050186

Academic Editor: Frank Werner

Received: 14 March 2024
Revised: 20 April 2024
Accepted: 25 April 2024
Published: 29 April 2024

1. Introduction

A *gene/species tree reconciliation* is an embedding of a gene tree explicating the topological difference between the two trees through a sequence of events into its associated species tree by shaping the gene family inside its host species. Gene/species tree reconciliation has been widely studied since the 1980s [1]; first, by focusing on duplication and loss events and then extending to horizontal gene transfers [2–4] and other events such as hybridization or incomplete lineage sorting (see review in [5]). One of the major drawbacks of classical reconciliation is that gene families are considered separately from one another, which is not appropriate for genes organized in *syntenies*, i.e., colocalized genes likely to have evolved together through segmental events.

Although some work has been achieved to infer the evolution of adjacencies [6], to group individual events into segmental ones [7] or to minimize "duplication episodes" [8,9], none of these methods are intended to explicitly look for evolutionary scenarios that minimize segmental events. We presented the first algorithm that generalizes reconciliation to synteny trees (i.e., with leaves representing syntenies rather than single genes) and segmental events for the duplication–loss distance in [10], and we extended it to horizontal transfers with *SuperDTL* in [11].

Here, we present Synesth (for *SYNteny Evolution in SegmenTal Histories*), an extended syntenic reconciliation model accounting for fissions, whereby part of a synteny is detached to another locus or species, in addition to losses, gains, duplications, and transfers.

Moreover, as the choice of the species included in a study—or, equivalently, the absence of species that are not chosen, unsampled, or extinct—has a serious impact on the output of a reconciliation algorithm [12–14], we account for the transfers going in species.

Among the infinite space of possible histories explicating the evolution of a synteny tree inside a species tree when both are given as an input, we are interested in selecting the most likely ones. A possible way to achieve this is by assigning costs (or probabilities) to event types and selecting histories minimizing (resp. maximizing) the overall cost (resp. probability). The resulting histories can vary significantly depending on the chosen costs, which are usually challenging to determine and strongly depend on the taxa under study [15,16]. Instead, our goal here is to provide an overview of the optimal histories for all possible cost choices.

Our approach is to progressively subdivide the history space. First, the space is reduced to a finite but exponential size set of Pareto-optimal histories (in terms of counts of each event type), then to a polynomial size set of Pareto-optimal event count vectors, and eventually to the single cost of an optimal history. We develop efficient and exact algorithms to compute the optimal histories for each subdivision level using algebraic dynamic programming. The inductive characterization of Pareto-optimal histories, which is given in Section 5, leads to a polynomial–time dynamic programming algorithm to output all Pareto-optimal vectors (Section 6), and then ultimately to an $O(mn)$ time complexity algorithm (where m and n are number of nodes in the input synteny and the species trees, respectively) to output the cost of a minimum cost history (Section 7), thereby matching the time complexity of the fastest algorithms for the classical duplication–transfer–loss reconciliation [14], an implementation of the algorithm is available at: https://github.com/UdeM-LBIT/superrec2/tree/algo2024 (accessed on 20 April 2024).

In Section 8, we apply Synesth to study the evolution of CRISPR-associated (Cas) gene syntenies. In taking advantage of a previous work [15,17], we present a visualization of the solution landscape as a partition of the space of cost choices into regions of equivalent costs leading to the same set of optimal histories.

We first introduce the required notations in Section 2, the evolutionary model in Section 3, and then we consider the solution space subdivision and optimization problems in Section 4.

2. Preliminary Notation

All trees are considered rooted in this method. Given a tree T, we denote by $\mathrm{r}(T)$ its root, by $\mathrm{V}(T)$ its node set, by $\mathrm{L}(T) \subseteq \mathrm{V}(T)$ its leaf set, and we let $\mathrm{I}(T) = \mathrm{V}(T) - \mathrm{L}(T)$ be the set of its internal nodes. A node v' is an *ancestor* of v if v' is on the path from $\mathrm{r}(T)$ to v, and the *parent* $\mathrm{p}(v)$ of v, of which v is a *child*, directly precedes v on this path. Conversely, v is a *descendant* of v'. This ancestor–descendant relation is denoted as $\leq$ and forms a partial order on the nodes, in which the root is minimal and the leaves are maximal. Any pair of nodes v and v' not ordered by this relation are said to be *separated*, which we denote as $v \parallel v'$. Notice that a node v is both an ancestor and a descendant of itself; whenever this case needs to be excluded, we will talk about strict ancestors and strict descendants.

We denote by $\mathrm{E}(T)$ the set of edges of T, each of which are represented by a pair of nodes $(\mathrm{p}(v), v)$. For any two nodes v and v' of T, there exists a unique path from v to v' that we denote $\mathrm{P}_T(v, v') \subseteq \mathrm{E}(T)$. The distance between v and v' is defined as $\mathrm{D}_T(v, v') = |\mathrm{P}_T(v, v')|$. Given a node v of T, $T[v]$ is the *subtree* of T rooted at v (i.e., containing only the descendants of v). The *lowest common ancestor* (LCA) of a subset V of nodes, denoted as $\mathrm{lca}_T(V)$, is the ancestor of all nodes in V, which is the most distant from the root.

A tree T' is said to be an *extension* of a tree T if T' can be obtained from T by a sequence of operations among the following: (1) *Subdividing* an edge (u, w) by adding a new node v and replacing (u, w) by two edges (u, v) and (v, w); (2) *Grafting* a new node v below an existing node u by adding the edge (u, v); (3) *Rerooting* the tree to a new node u by adding the edge $(u, \mathrm{r}(T))$.

The set of children of any node v is denoted by $\mathrm{ch}(v)$. If $|\mathrm{ch}(v)| = 1$, then v is said to be *unary* and we denote its only child by v_{c}. If $|\mathrm{ch}(v)| = 2$, then it is said to be *binary* and, unless specified otherwise, we denote its children by v_ℓ and v_{r} in no particular order. A *binary tree* is a tree where all internal (non-leaf) nodes are binary. If all internal nodes are unary or binary, then the tree is *partially binary*.

If $\mathcal{F}$ is a set of gene families, then a *synteny* on $\mathcal{F}$ is a subset $X \subseteq \mathcal{F}$, which represents a group of genes assumed to have jointly evolved. Notice that we ignore the relative order of genes in the genomic region, as well as the physical distance between genes and regions. The genes of a synteny are considered to all belong to different gene families (i.e., repeated gene families inside a synteny are forbidden); therefore, a gene is simply identified by the family $\Gamma \in \mathcal{F}$ it belongs to.

A *species tree* S on a set Σ of species is a binary tree with a bijection between $\mathrm{L}(S)$ and Σ. For a set of syntenies $\mathcal{X}$, a *synteny tree* $\langle T, x, s\rangle$ is a tuple where T is a binary tree, $x : \mathrm{L}(T) \to \mathcal{X}$ and $s : \mathrm{L}(T) \to \Sigma$ are two functions, the second indicating the species to which each synteny belongs.

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

3. Evolutionary Histories for Syntenies

We model the evolution of syntenies through the following syntenic events: codivergence with the host species ("Spe"), duplication of a synteny subset ("Dup"), fission of a synteny ("Cut"), transfer of a duplicated or cut subset (resp. "TrDup" or "TrCut"), and the gain or loss of a subset ("Gain", "Loss"). Losses can be partial in the sense that only a subset of genes in a synteny are lost. Evolutionary histories are the sequences of such events, as formally defined below (see example in Figure 1).

Definition 1 (Evolutionary history for syntenies)**.** *A history $\mathcal{H}$ on a species tree S is a tuple $\langle H, e, x, s\rangle$, where H is a partially binary tree. Each node $v \in \mathrm{V}(H)$ is labeled with a species $s(v) \in \mathrm{V}(S)$ and a synteny (i.e., a subset of gene families) $x(v)$. Each internal node is additionally labeled with an event $e(v) \in \mathcal{E} = \{\mathrm{Spe}, \mathrm{Dup}, \mathrm{Cut}, \mathrm{TrDup}, \mathrm{TrCut}, \mathrm{Gain}, \mathrm{Loss}\}$ acting on $x(v)$ and $s(v)$. These labels satisfy the following conditions:*

1. *If $e(v) = \mathrm{Spe}$, with $\mathrm{ch}(v) = \{v', v''\}$ and $\sigma = s(v)$, then $x(v) = x(v') = x(v'')$, $s(v') = \sigma_\ell$, and $s(v'') = \sigma_{\mathrm{r}}$.*
2. *If $e(v) \in \{\mathrm{Dup}, \mathrm{Cut}, \mathrm{TrDup}, \mathrm{TrCut}\}$, with $\mathrm{ch}(v) = \{v_{\mathrm{t}}, v_{\mathrm{k}}\}$:*
    1. *If $e(v) \in \{\mathrm{Dup}, \mathrm{TrDup}\}$, then $x(v_{\mathrm{t}}) \subseteq x(v) = x(v_{\mathrm{k}})$;*
    2. *If $e(v) \in \{\mathrm{Cut}, \mathrm{TrCut}\}$, then $x(v_{\mathrm{t}}) \cup x(v_{\mathrm{k}}) = x(v)$, $x(v_{\mathrm{t}}) \cap x(v_{\mathrm{k}}) = \emptyset$, and $x(v_{\mathrm{t}}) \neq \emptyset$ (but $x(v_{\mathrm{k}}) = \emptyset$ is allowed);*
    3. *If $e(v) \in \{\mathrm{Dup}, \mathrm{Cut}\}$, then $s(v) = s(v_{\mathrm{t}}) = s(v_{\mathrm{k}})$;*
    4. *If $e(v) \in \{\mathrm{TrDup}, \mathrm{TrCut}\}$, then $s(v) \parallel s(v_{\mathrm{t}})$, and $s(v) = s(v_{\mathrm{k}})$.*
3. *If $e(v) \in \{\mathrm{Gain}, \mathrm{Loss}\}$ with $\mathrm{ch}(v) = \{v_{\mathrm{c}}\}$, then $s(v_{\mathrm{c}}) = s(v)$ and the following:*
    1. *If $e(v) = \mathrm{Gain}$, then $x(v_{\mathrm{c}}) \supsetneq x(v)$.*
    2. *If $e(v) = \mathrm{Loss}$, then $x(v_{\mathrm{c}}) \subsetneq x(v)$ (a loss is* full *if $x(v_{\mathrm{c}}) = \emptyset$, and* partial *otherwise).*
4. *For each gene family Γ, exactly one* Gain *event in H involves Γ.*
5. *The only nodes v of the history such that $x(v) = \emptyset$ are its root and its leaves.*

Finally, we denote by Tr *any event in* $\{\mathrm{TrDup}, \mathrm{TrCut}\}$.

Notice that, when the sets of $x(v)$ are restricted to at most one gene family each, this model reduces to the classical reconciliation model of [14], with TrDup corresponding to $\mathbb{T}$ (transfer) and TrCut to $\mathbb{TL}$ (transfer–loss). Additionally, from any syntenic history on a set of gene families $\mathcal{F}$, one can extract a reconciled gene tree for each gene family $\Gamma \in \mathcal{F}$ whose

root is the gain event for Γ and leaves are the loss events where Γ is lost. This root is unique because of Condition (4), which excludes *convergent gains*.

Moreover, as in [14], we will allow for transfers to and from unsampled or extinct species by augmenting the species tree. For example, in Figure 1, Synteny $\{1,2\}$ in Species A is the result of a transfer from an unsampled species. In the following, unless specified otherwise, all histories are on augmented species trees, as formally defined below.

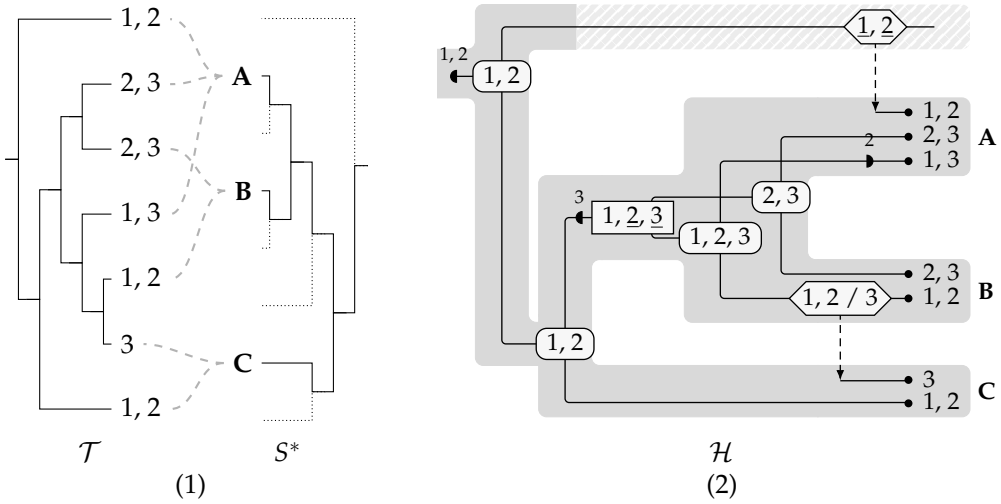


Figure 1. (1) A *synteny tree* $\mathcal{T}$ (on the **left**) with an augmented *species tree* S^* (on the **right**). The numbers represent single gene families and the letters represent species. As indicated by the dashed gray lines connecting the two trees, the Syntenies $\{1,2\}$, $\{2,3\}$, and $\{1,3\}$ belong to Species **A**, $\{2,3\}$, and $\{1,2\}$ to **B** $\{3\}$ and $\{1,2\}$ to **C**. The dotted lines in the augmented species tree represent unsampled edges. (2) An output *history* $\mathcal{H}$ of Synesth when given $\mathcal{T}$ and S^* as the input with costs $(\delta_{\text{Dup}} = 2, \delta_{\text{Cut}} = 2.5, \delta_{\text{TrDup}} = 3, \delta_{\text{TrCut}} = 3.5, \delta_{\text{Loss}} = 1)$. The history tree is represented with black lines on top of the species tree filled in gray. The hatched background represents a part of the history taking place in an unsampled lineage. The events are represented as follows: "Spe" by ovals, "Dup" and "Cut" by rectangles, "TrDup" and "TrCut" by diamonds, "Loss" by right half-circles, and "Gain" by left half-circles. The synteny contents are written inside of each event, while the associated species is represented implicitly by the position of each event on top of the species tree. Duplicated or transferred genes are underlined, while fissions are represented by a separation in the synteny.

Definition 2 (Augmented species tree)**.** *A tree S can be augmented into S^* by adding* unsampled leaves *as follows: (1) Subdivide each edge (v, v') of $\mathrm{E}(S)$ into two edges (v, z) and (z, v') linked to a new node z; (2) Connect each z to a new unsampled leaf; (3) Create a new root $\mathrm{r}(S^*)$ whose two children are $\mathrm{r}(S)$ and a new unsampled leaf. Edges leading to unsampled leaves are called* unsampled edges.

Finally, notice that Definition 1 does not require the species involved in a transfer to be contemporary, nor does it forbid biologically infeasible cyclic histories, such as one resulting from a transfer from a Species A to a Species B, and then back from Species B to an ancestor of Species A (for a more precise definition of acyclicity, see [4]). This limitation is necessary to make the computational problems of the next section tractable.

4. Explicatory Histories and Optimization Problems

The goal of reconciliation is to infer histories that *explicate* the topology of a synteny tree given a tree of the corresponding species. Such histories are extensions of the synteny tree that map all leaves to the appropriate species and syntenies without introducing new *visible* leaves.

Definition 3 (Visible leaves). *A leaf l of a history $\mathcal{H} = \langle H, e, x, s \rangle$ is said to be* visible *if $x(l) \neq \emptyset$ and $s(l) \in \mathrm{V}(S)$, and* invisible *otherwise. The set of visible leaves of $\mathcal{H}$ is denoted as $\mathrm{L}_V(\mathcal{H})$.*

For example, in Figure 1, the history in (2) explicates the trees in (1). In that history, the only invisible leaf is the unnamed leaf in the hatched region (representing an unsampled species). The leaves below full losses would also be invisible, but that example history does not contain any full loss. The following is a formal definition of an explicatory history:

Definition 4 (Explicatory histories). *For a species tree S, a history $\mathcal{H} = \langle H, e, x, s \rangle$ on S^* is said to* explicate *a synteny tree $\mathcal{T} = \langle T, x', s' \rangle$ and S if the following holds true:*

1. *H is an extension of T.*
2. *$x' = x|_{\mathrm{L}(T)}$ and $s' = s|_{\mathrm{L}(T)}$.*
3. *$\mathrm{L}_V(\mathcal{H}) = \mathrm{L}(T)$.*
4. *No $(u, v) \in \mathrm{E}(T)$ is such that $s(v) < s(u)$.*
5. *No gain event is the parent of a node $v \in \mathrm{V}(H) - \mathrm{V}(T)$.*
6. *No partial loss event is a child of a node $v \in \mathrm{V}(H) - \mathrm{V}(T)$.*

The set of all such histories is denoted by $\mathbb{H}(\mathcal{T}, S)$.

Condition (3) disallows introducing new visible leaves. Condition (4) excludes assignments of species that create cycles between adjacent nodes of the synteny tree. This condition is a necessary, but not sufficient, condition for acyclicity. Imposing a full acyclicity condition would lead to computationally intractable problems [4].

As for Conditions (5) and (6), they are introduced to avoid having multiple histories with the same events, but with gains and losses distributed differently between the adjacent nodes of the synteny tree. More precisely, Condition (5) requires sifting gains down and merging them until they are the parent of a synteny tree node; conversely, Condition (6) requires sifting losses up and merging them until they are the child of a synteny tree node.

Note that $\mathbb{H}(\mathcal{T}, S)$ is infinite: given any explicatory history, it is always possible to extend it into a larger one by introducing superfluous duplications or transfers. We next define a way to reduce this space to a finite one.

Definition 5 (Event vector). *Let $\mathcal{H} = \langle H, e, x, s \rangle$ be a history. We define $\mathrm{ev}(\mathcal{H}) = (c_{\mathrm{Dup}}, c_{\mathrm{Cut}}, c_{\mathrm{TrDup}}, c_{\mathrm{TrCut}}, c_{\mathrm{Loss}}) \in \mathbb{N}^5$ as the vector such that c_e is the number of events of type $e \in \mathcal{E}$ in $\mathcal{H}$.*

As usual for reconciliation, our definition of an event vector excludes the number of speciations since they do not allow one to meaningfully distinguish between histories. Notice that the number of gains is also excluded, as needed to make the problem of Section 6 tractable. Consequently, we disregard taking advantage of the simultaneous gains of multiple genes to reduce the overall number of individual gain events.

Definition 6 (Order on vectors and histories). *For two event vectors $(a_i)_{1 \leq i \leq n}$ and $(b_i)_{1 \leq i \leq n}$, $(a_i) \preceq (b_i)$ if for all $i \in \{1, ..., n\}$, $a_i \leq b_i$. This partial order induces another one on the histories, namely $\mathcal{H} \preceq \mathcal{H}'$ if $\mathrm{ev}(\mathcal{H}) \preceq \mathrm{ev}(\mathcal{H}')$.*

Definition 7 (Pareto optimality). *For any set E with a partial order $\preceq$, we define its Pareto subset as $\min_{\preceq} E = \{x \in E \mid \forall y \in E, (x \neq y) \rightarrow \neg(y \preceq x)\}$.*

As an example, in the set of vectors $A = \{(1, 2), (2, 2), (3, 1)\} \subseteq \mathbb{N}^2$, we have $(1, 2) \preceq (2, 2)$ while $(1, 2)$ and $(3, 1)$ are not comparable; hence, $\min_{\preceq} A = \{(1, 2), (3, 1)\}$.

Let us now consider the set $\mathbb{H}^{\min}(\mathcal{T}, S) = \min_{\preceq} \mathbb{H}(\mathcal{T}, S)$ of histories whose event vectors are Pareto-optimal. As opposed to $\mathbb{H}(\mathcal{T}, S)$, $\mathbb{H}^{\min}$ is a finite set, as we will show in Theorem 3. We can thus meaningfully define the problem of computing this set.

Problem 1 (All Pareto-optimal histories).
Input: A synteny tree $\mathcal{T}$ and a species tree S.
Output: The set $\mathbb{H}^{\min}(\mathcal{T}, S)$ of Pareto-optimal histories explicating $\mathcal{T}$ and S.

Even though $\mathbb{H}^{\min}$ is finite, it may contain a number of optimal histories that are exponential in $|\mathrm{V}(T)|$. Rather, consider the set $\mathrm{ev}^{\min}(\mathcal{T}, S) = \min_{\preceq} \{\mathrm{ev}(\mathcal{H}) \mid \mathcal{H} \in \mathbb{H}(\mathcal{T}, S)\}$ of Pareto-optimal event vectors. As we will show later (Theorem 4), the number of optimal event vectors in $\mathrm{ev}^{\min}$ is polynomial. We now consider the following problem of reduced complexity:

Problem 2 (All Pareto-optimal vectors).
Input: A synteny tree $\mathcal{T}$ and a species tree S.
Output: The set $\mathrm{ev}^{\min}(\mathcal{T}, S)$.

Now, given a vector of costs for each event type $\delta = (\delta_{\mathrm{Dup}}, \delta_{\mathrm{Cut}}, \delta_{\mathrm{TrDup}}, \delta_{\mathrm{TrCut}}, \delta_{\mathrm{Loss}}) \in (\mathbb{R}^+ \cup \{\infty\})^5$, we can associate an overall scalar cost $\mathrm{c}(\mathcal{H}) = \delta \cdot \mathrm{ev}(\mathcal{H})$ to each history.

Problem 3 (Minimum cost).
Input: A synteny tree $\mathcal{T}$, a species tree S, and a vector $\delta \in (\mathbb{R}^+ \cup \{\infty\})^5$.
Output: The minimum cost $c^{\min}(\mathcal{T}, S) = \min\{\mathrm{c}(\mathcal{H}) \mid \mathcal{H} \in \mathbb{H}(\mathcal{T}, S)\}$ of any history explicating $\mathcal{T}$ and S.

Finally, we will call Problems 2′ and 3′ the versions of Problems 2 and 3 where we additionally ask for one history corresponding to each returned optimal event vector (for Problem 2′) or one history for the returned minimum cost (for Problem 3′). For example, Figure 1 shows in (2) a possible solution for Problem 2′ for the synteny and species trees in (1) for vector $(c_{\mathrm{Dup}} = 1, c_{\mathrm{Cut}} = 0, c_{\mathrm{TrDup}} = 1, c_{\mathrm{TrCut}} = 1, c_{\mathrm{Loss}} = 1)$, as well as cost $(\delta_{\mathrm{Dup}} = 2, \delta_{\mathrm{Cut}} = 2.5, \delta_{\mathrm{TrDup}} = 3, \delta_{\mathrm{TrCut}} = 3.5, \delta_{\mathrm{Loss}} = 1)$ for Problem 3′.

5. Generating All Pareto-Optimal Histories

We start by addressing Problem 1, which asks to enumerate the $\mathbb{H}^{\min}$ set. This can be conducted inductively by building histories from the leaves of the synteny tree up to its root. This result is the basis for the dynamic programming formulations introduced for the other problems in the upcoming sections.

The next two definitions are used to build histories by composing partial histories (see Figure 2). In the following, a node v of a history such that $x(v) = X$ and $s(v) = \sigma$ is denoted as $v[X, \sigma]$, or $v[e, X, \sigma]$ if the event $e \in \mathcal{E}$ associated to v is known. The node name may be omitted where it is not relevant by simply writing $[X, \sigma]$ or $[e, X, \sigma]$. If A, B, and C are three nodes, then $(A, B)C$ refers to the triplet tree with Root C and Leaves A and B.

Definition 8 (Partial histories). *Let $\mathcal{T} = \langle T, x, s\rangle$ be a synteny tree on a species tree S. Let $v \in \mathrm{V}(T)$, $X \subseteq \mathcal{F}$, and $\sigma \in \mathrm{V}(S^*)$. We define $h(v, X, \sigma)$ as the set of the Pareto-optimal histories explicating the subtree of $\mathcal{T}$ rooted at $v[X, \sigma]$. We also define* $\mathrm{path}([X, \sigma], [Y, \gamma])$ *as the set of Pareto-optimal acyclic histories whose root is $[X, \sigma]$ and whose only visible leaf is $[Y, \gamma]$. Formally,*

$$h(v, X, \sigma) = \{\langle H, e, x, s\rangle \in \mathbb{H}^{\min}(\mathcal{T}[v], S) \mid (\mathrm{r}(H) = v) \wedge (x(v) = X) \wedge (s(v) = \sigma)\}$$

$$\begin{aligned}\mathrm{path}([X, \sigma], [Y, \gamma]) = \min_{\preceq} \{\mathcal{H} = \langle H, e, x, s\rangle \mid\ & (x(\mathrm{r}(H)) = X) \wedge (s(\mathrm{r}(H)) = \sigma) \\ & \wedge (\mathrm{L}_V(\mathcal{H}) = \{l\}) \wedge (x(l) = Y) \wedge (s(l) = \gamma) \\ & \wedge \mathcal{H} \text{ is acyclic}\}.\end{aligned}$$

Definition 9 (History composition). *If $\mathcal{H} = \langle H, e, x, s\rangle$ and $\mathcal{H}' = \langle H', e', x', s'\rangle$ are two histories such that there exists a leaf $l \in \mathrm{L}(H)$ with $x(l) = x(\mathrm{r}(H'))$ and $s(l) = s(\mathrm{r}(H'))$, then we define their* composition *$\mathcal{H} \otimes \mathcal{H}'$ as the history obtained by replacing l with $\mathrm{r}(H')$ and merging e with e', x with x', and s with s'. This operation is only defined if the resulting history is valid (particularly if no gene family would be gained in two separate* Gain *events).*

When applied to two sets *of histories A and B, then $A \otimes B$ is defined by taking the Cartesian product of the two sets and composing each resulting history pair whilst excluding invalid compositions.*

For any node v of a synteny tree $\mathcal{T}$ and any event $e \in \mathcal{E}$, syntenies $X, Y, Y', Z, Z' \subseteq \mathcal{F}$, and species $\sigma, \gamma_\ell, \gamma'_\ell, \gamma_\mathrm{r}, \gamma'_\mathrm{r} \in \mathrm{V}(S^*)$, we define the set of histories starting with $v[e, X, \sigma]$ as followed by two paths from $[Y', \gamma'_\ell]$ to $v_\ell[Y, \gamma_\ell]$ and from $[Z', \gamma'_\mathrm{r}]$ to $v_\mathrm{r}[Z, \gamma_\mathrm{r}]$, thereby leading to two sub-histories as follows:

$$\begin{aligned}&\mathcal{M}(v[e, X, \sigma], [Y', \gamma'_\ell], v_\ell[Y, \gamma_\ell], [Z', \gamma'_\mathrm{r}], v_\mathrm{r}[Z, \gamma_\mathrm{r}])\\ &= ([Y', \gamma'_\ell], [Z', \gamma'_\mathrm{r}])\ v[e, X, \sigma] \otimes (\mathrm{path}([Y', \gamma'_\ell], v_\ell[Y, \gamma_\ell]) \otimes h(v_\ell, Y, \gamma_\ell))\\ &\qquad\qquad \otimes (\mathrm{path}([Z', \gamma'_\mathrm{r}], v_\mathrm{r}[Z, \gamma_\mathrm{r}]) \otimes h(v_\mathrm{r}, Z, \gamma_\mathrm{r})).\end{aligned}$$

For example, in Figure 2, we have $\mathcal{H} = \mathcal{M}(v[\mathrm{Spe}, X, \sigma], [X, \sigma_\ell], v_\ell[Y, \gamma_\ell], [X, \sigma_\mathrm{r}], v_\mathrm{r}[Z, \gamma_\mathrm{r}])$.

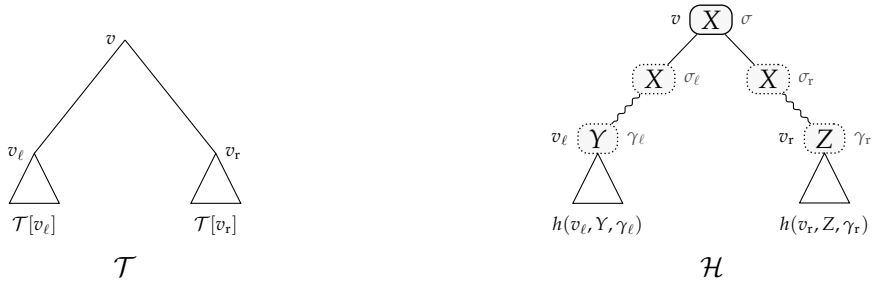


Figure 2. General shape of a history $\mathcal{H}$ for a synteny tree $\mathcal{T}$ starting with $v[\mathrm{Spe}, X, \sigma]$, which is composed of two sub-histories taken from $h(v_\ell, Y, \gamma_\ell)$ and $h(v_\mathrm{r}, Z, \gamma_\mathrm{r})$. These are linked together by two paths (represented by wavy lines) taken from $\mathrm{path}([X, \sigma_\ell], v_\ell[Y, \gamma_\ell])$ and $\mathrm{path}([X, \sigma_\mathrm{r}], v_\mathrm{r}[Z, \gamma_\mathrm{r}])$.

This representation of histories as the compositions of sub-histories and paths will now be used to formulate inductive definitions for $h(v, X, \sigma)$ and $\mathrm{path}(\cdot, \cdot)$, as well as ultimately $\mathbb{H}^{\min}(\mathcal{T}, S)$. In the following, we use $A \oplus B$ to mean $\min_{\preceq}(A \cup B)$.

Theorem 1 (Inductive form of Pareto-optimal histories). *Let $\mathcal{T} = \langle T, x, s\rangle$ be a synteny tree on S, X be any synteny on $\mathcal{F}$, and σ be a node of S^*. If v is a leaf of T, then $h(v, X, \sigma) = \{v[X, \sigma]\}$ if $x(v) = X$ and $s(v) = \sigma$, and $h(v, X, \sigma) = \emptyset$ otherwise. If v is an internal node of T, then $h(v, X, \sigma) = P_\mathrm{Spe} \oplus P_\mathrm{Dup} \oplus P_\mathrm{Cut} \oplus P_\mathrm{TrDup} \oplus P_\mathrm{TrCut}$ with the following:*

$$\begin{aligned}P_\mathrm{Spe} = \bigoplus_{\substack{\gamma_\ell, \gamma_\mathrm{r} \not\preceq \sigma\\ Y, Z \subseteq \mathcal{F}}} &(\mathcal{M}(v[\mathrm{Spe}, X, \sigma], [X, \sigma_\ell], v_\ell[Y, \gamma_\ell], [X, \sigma_\mathrm{r}], v_\mathrm{r}[Z, \gamma_\mathrm{r}])\\ &\oplus \mathcal{M}(v[\mathrm{Spe}, X, \sigma], [X, \sigma_\mathrm{r}], v_\ell[Y, \gamma_\ell], [X, \sigma_\ell], v_\mathrm{r}[Z, \gamma_\mathrm{r}]))\end{aligned}$$

$$\begin{aligned}P_\mathrm{Dup} = \bigoplus_{\substack{\gamma_\ell, \gamma_\mathrm{r} \not\preceq \sigma\\ Y, Z \subseteq \mathcal{F}}} &(\mathcal{M}(v[\mathrm{Dup}, X, \sigma], [X \cap Y, \sigma], v_\ell[Y, \gamma_\ell], [X, \sigma], v_\mathrm{r}[Z, \gamma_\mathrm{r}])\\ &\oplus \mathcal{M}(v[\mathrm{Dup}, X, \sigma], [X, \sigma], v_\ell[Y, \gamma_\ell], [X \cap Z, \sigma], v_\mathrm{r}[Z, \gamma_\mathrm{r}]))\end{aligned}$$

$$\begin{aligned}P_\mathrm{Cut} = \bigoplus_{\substack{\gamma_\ell, \gamma_\mathrm{r} \not\preceq \sigma\\ Y, Z \subseteq \mathcal{F}}} &(\mathcal{M}(v[\mathrm{Cut}, X, \sigma], [X \cap Y, \sigma], v_\ell[Y, \gamma_\ell], [X - Y, \sigma], v_\mathrm{r}[Z, \gamma_\mathrm{r}])\\ &\oplus \mathcal{M}(v[\mathrm{Cut}, X, \sigma], [X - Z, \sigma], v_\ell[Y, \gamma_\ell], [X \cap Z, \sigma], v_\mathrm{r}[Z, \gamma_\mathrm{r}]))\end{aligned}$$

$$P_{\text{TrDup}} = \bigoplus_{\substack{\gamma_i \| \sigma;\ \gamma_t \not< \gamma_i, \sigma \\ \gamma_k \not< \sigma;\ Y, Z \subseteq \mathcal{F}}} \big(\mathcal{M}(v[\text{TrDup}, X, \sigma], [X \cap Y, \gamma_i], v_\ell[Y, \gamma_t], [X, \sigma], v_r[Z, \gamma_k]) \oplus \mathcal{M}(v[\text{TrDup}, X, \sigma], [X, \sigma], v_\ell[Y, \gamma_k], [X \cap Z, \gamma_i], v_r[Z, \gamma_t])\big)$$

$$P_{\text{TrCut}} = \bigoplus_{\substack{\gamma_i \| \sigma;\ \gamma_t \not< \gamma_i, \sigma \\ \gamma_k \not< \sigma;\ Y, Z \subseteq \mathcal{F}}} \big(\mathcal{M}(v[\text{TrCut}, X, \sigma], [X \cap Y, \gamma_i], v_\ell[Y, \gamma_t], [X - Y, \sigma], v_r[Z, \gamma_k]) \oplus \mathcal{M}(v[\text{TrCut}, X, \sigma], [X - Z, \gamma_i], v_\ell[Y, \gamma_t], [X \cap Z, \sigma], v_r[Z, \gamma_k]) \oplus \mathcal{M}(v[\text{TrCut}, X, \sigma], [X \cap Y, \sigma], v_\ell[Y, \gamma_k], [X - Y, \gamma_i], v_r[Z, \gamma_t]) \oplus \mathcal{M}(v[\text{TrCut}, X, \sigma], [X - Z, \sigma], v_\ell[Y, \gamma_k], [X \cap Z, \gamma_i], v_r[Z, \gamma_t])\big)$$

Proof. If v is a leaf of T, then the proposition follows directly from Definitions 1 and 4. Assume now that v is an internal node. First, notice that all histories of the P_e sets—for $e \in \{\text{Spe}, \text{Dup}, \text{Cut}, \text{TrDup}, \text{TrCut}\}$—explicate $\mathcal{T}[v]$ and S, have their root assigned to X and σ and are Pareto-optimal.

Let $\mathcal{H} = \langle H, e, x, s\rangle \in h(v, X, \sigma)$. Since v is an internal node of T, v must be binary in H; hence $e(v) \in \{\text{Spe}, \text{Dup}, \text{Cut}, \text{TrDup}, \text{TrCut}\}$. Denote the children of v in T as v_ℓ and v_r, and the children of v in H as w_ℓ and w_r. If $e(v) \in \{\text{Spe}, \text{Dup}, \text{Cut}\}$, then γ_ℓ and γ_r cannot be strict ancestors of σ, as otherwise $\mathcal{H}$ would not be acyclic. Let $Y = x(v_\ell)$, $Y' = x(w_\ell)$, $Z = x(v_r)$, $Z' = x(w_r)$.

If $e(v) = \text{Spe}$, then $\mathcal{H} \in P_{\text{Spe}}$ by Definition 1 (Item 1).

Otherwise, let us first show that the synteny contents of both children are chosen appropriately. We have that $X \cap Y \subseteq Y'$ (resp. $X \cap Z \subseteq Z'$); since, if any $g \in X \cap Y$ (resp. $g \in X \cap Z$) was not in Y' (resp. Z'), we would have multiple gains of g in H, i.e., one above v (since $g \in X$) and one between v and v_l (since $g \in Y$, resp. v_r since $g \in Z$).

- If $e(v) \in \{\text{Dup}, \text{TrDup}\}$, by Definition 1, at least one of Y' or Z' is equal to X. Without loss of generality (w.l.o.g.), assume that $Z' = X$. We should have $Y' \subseteq X \cap Y$, as otherwise we would have at least one additional event in the path between v and v_l for losing $Y' - (X \cap Y)$, thus contradicting the Pareto-optimality of $\mathcal{H}$. Hence, $Y' = X \cap Y$.
- If $e(v) \in \{\text{Cut}, \text{TrCut}\}$, it cannot be that $Y' \not\subseteq X \cap Y$ and $Z' \not\subseteq X \cap Z$ are both true as such a scenario would lead to a loss on each path between v and v_ℓ and between v and v_r, whereas by choosing $Y' \subseteq X \cap Y$ or $Z' \subseteq X \cap Z$, we save at least one loss. Since Y' and Z' are a partition of X, then $Z' = X - Y$ if $Y' = X \cap Y$, or $Y' = X - Z$ otherwise.

Finally, we show that the species of both children are also chosen appropriately.

- If $e(v) \in \{\text{Dup}, \text{Cut}\}$, then by Definition 1, $s(w_\ell) = s(w_r) = \sigma$.
- If $e(v) \in \{\text{TrDup}, \text{TrCut}\}$, then by Definition 1, either $s(w_\ell) = \sigma$ and $s(w_r) \parallel \sigma$, or $s(w_r) = \sigma$ and $s(w_\ell) \parallel \sigma$.
- If $e(v) = \text{TrDup}$, if $s(w_\ell) = \sigma$ (resp. $s(w_r) = \sigma$), then $x(w_\ell) = X$ (resp. $x(w_r) = Y$). □

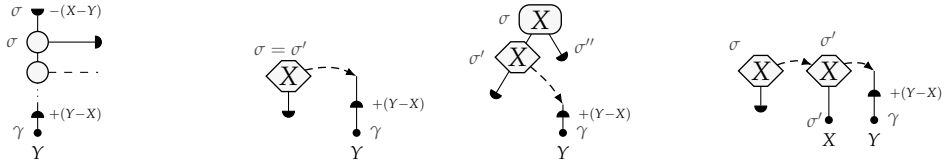
We next consider how to compute the set of histories $\text{path}([X, \sigma], [Y, \gamma])$ given any two syntenies X and Y and species σ and γ.

Theorem 2 (Pareto-optimal paths). *Let X and Y be two syntenies and σ and γ be two species. If γ is a strict ancestor of σ, then* $\text{path}([X, \sigma], [Y, \gamma]) = \emptyset$. *Otherwise,* $\text{path}([X, \sigma], [Y, \gamma]) = P_0 \oplus P_1 \oplus P_2$, *where the following holds:*

- $P_0 = \emptyset$ *if* $\sigma \parallel \gamma$. *Otherwise, P_0 contains the history made up of a chain of* $\text{D}_{S^*}(\sigma, \gamma)$ *speciations and* $|\{(u, v) \in \text{P}_{S^*}(\sigma, \gamma) \mid v \notin \text{V}(S)\}|$ *full losses; preceded by a partial loss event if $X \not\subseteq Y$ and followed by a gain event if $Y \not\subseteq X$. If* $\gamma \in \text{L}(S^*) - \text{L}(S)$, *then P_0 also includes a history where the initial partial loss event is replaced by a terminal* Dup *or* Cut *event and an unsampled leaf.*
- $P_1 = P_{\text{TrDup}} \oplus P^{\text{Left}}_{\text{TrCut}} \oplus P^{\text{Right}}_{\text{TrCut}}$. *If $\sigma \parallel \gamma$, all three sets contain histories starting with a* Tr *event v from $\sigma' = \sigma$ to γ. Otherwise, all three sets contain histories starting with a speciation*

at σ. Letting $\mathrm{ch}(\sigma) = \{\sigma', \sigma''\}$ such that $\sigma' \parallel \gamma$, the speciation is followed by a Tr *event v from σ' to γ and a full loss on the side of σ'' if $\sigma'' \notin \mathrm{L}(S^*) - \mathrm{L}(S)$. In both cases, the histories end with a gain event if $Y \not\subseteq X$. For $\mathcal{H} \in P_{\mathrm{TrDup}}$, $e(v) =$ TrDup and $x(v_\mathrm{t}) = X \cap Y$ with a full loss at v_k if $\sigma' \notin \mathrm{L}(S^*) - \mathrm{L}(S)$. For $\mathcal{H} \in P^{\mathrm{Left}}_{\mathrm{TrCut}}$ or $\mathcal{H} \in P^{\mathrm{Right}}_{\mathrm{TrCut}}$, $e(v) =$ TrCut. For $\mathcal{H} \in P^{\mathrm{Left}}_{\mathrm{TrCut}}$, $x(v_\mathrm{t}) = X \cap Y$ and v_k is a full loss if $X \not\subseteq Y$ and $\sigma' \notin \mathrm{L}(S^*) - \mathrm{L}(S)$. For $\mathcal{H} \in P^{\mathrm{Right}}_{\mathrm{TrCut}}$, $x(v_\mathrm{k}) = \emptyset$ and there is a partial loss at v_t if $X \not\subseteq Y$.*

- *$P_2 = \emptyset$ if $\sigma \parallel \gamma$ and $X \subseteq Y$, or $\sigma \in \mathrm{L}(S^*) - \mathrm{L}(S)$, or $\sigma = \mathrm{r}(S^*)$. Otherwise, $P_2 = P^{\mathrm{TrDup}}_{\mathrm{TrDup}} \oplus P^{\mathrm{TrCut}}_{\mathrm{TrDup}} \oplus P^{\mathrm{TrCut}}_{\mathrm{TrCut}}$. Each of the three sets $P^{e}_{e'}$ contain histories made up of two initial consecutive transfers v and v' such that $e(v) = e$ and $e(v') = e'$. The histories of $P^{\mathrm{TrDup}}_{\mathrm{TrDup}}$ and $P^{\mathrm{TrCut}}_{\mathrm{TrDup}}$ are such that $s(v') = \sigma' \in \mathrm{L}(S^*) - \mathrm{L}(S)$. In $P^{\mathrm{TrDup}}_{\mathrm{TrDup}}$, v is followed by a full loss if $s(v) \notin \mathrm{L}(S^*) - \mathrm{L}(S)$. In $P^{\mathrm{TrCut}}_{\mathrm{TrDup}}$ and $P^{\mathrm{TrCut}}_{\mathrm{TrCut}}$, the first cut v is complete (i.e., with an empty leaf). In $P^{\mathrm{TrCut}}_{\mathrm{TrCut}}$ if $X \not\subseteq Y$, then $s(v) \in \mathrm{L}(S^*) - \mathrm{L}(S)$, otherwise the second cut v' is complete if $s(v) \notin \mathrm{L}(S^*) - \mathrm{L}(S)$. In all cases, the histories end with a gain event if $Y \not\subseteq X$.*



(P_0) (P_1) (P_2)

Proof. If γ is a strict ancestor of σ, there can be no acyclic history leading from $[X, \sigma]$ to $[Y, \gamma]$; therefore, $\mathrm{path}([X, \sigma], [Y, \gamma]) = \emptyset$.

Otherwise, let $\mathcal{H} = \langle H, e, x, s \rangle \in \mathrm{path}([X, \sigma], [Y, \gamma])$, $u = \mathrm{r}(H)$, and let w be the only visible leaf of $\mathcal{H}$. We call $\mathrm{P}(u, w)$ the *main path* of $\mathcal{H}$. We say that a subtree $H[v]$ of H is *invisible* if it contains only invisible leaves.

First notice that all binary nodes of $\mathcal{H}$ must be on the main path. Assume that v is a binary node outside of the main path. Both subtrees of v must be invisible. If $s(v) \notin \mathrm{L}(S^*) - \mathrm{L}(S)$, then at least one subtree contains a full loss; hence, we can completely replace v and its subtrees with a full loss. Otherwise, we can completely replace v and its subtrees with an unsampled leaf. In both cases, we strictly reduce the number of events in $\mathcal{H}$, which contradicts its Pareto-optimality. Also, notice that all binary nodes of the main path have exactly one child whose subtree is invisible.

We show now that $\mathcal{H}$ may only contain at most two Tr nodes. Assume that $\mathcal{H}$ contains at least three consecutive Tr events that we denote v_1, v_2, v_3. Let v_4 be the child of v_3 resulting from the transfer. Consider the history $\mathcal{H}'$ obtained from $\mathcal{H}$ via the following: (1) Removing all the nodes on the path from v_2 to v_4 and their invisible subtrees, excluding v_2 and v_4 themselves but including, in particular, v_3; (2) Connecting v_2 directly to v_4; (3) Remapping $s(v_2)$ to any unsampled species separated both from $s(v_1)$ and $s(v_4)$; (4) Replacing the subtree below v_2 by a single unsampled leaf l with $s(l) = s(v_2)$ and $x(l) = x(v_2)$ if $e(v_2) =$ TrDup, or $x(l) = x(v_2) - x(v_4)$ otherwise. Clearly, $\mathcal{H}' \preceq \mathcal{H}$, $\mathcal{H}' \neq \mathcal{H}$ and $\mathcal{H}' \in \mathrm{path}([X, \sigma], [Y, \gamma])$, which contradicts the Pareto-optimality of $\mathcal{H}$.

As per Definition 4 (Item 5 and Item 6), any partial loss must be placed at the end of the history, and any gain must be placed at the start. Note that $\mathcal{H}$ contains either zero (P_0), one (P_1), or two (P_2) transfers.

- [$\mathcal{H}$ *contains no transfers.*] Note that σ cannot be separated from γ since only transfers can reach separate species; thus, σ is an ancestor of γ. We start by showing that $\mathcal{H}$ can only possibly contain a duplication or a cut if that event is on an unsampled leaf species. If v is a node such that $e(v) \in \{\mathrm{Dup}, \mathrm{Cut}\}$ and $s(v) \notin \mathrm{L}(S^*) - \mathrm{L}(S)$, then the invisible subtree of v contains at least one full loss. In this case, we can remove the invisible subtree and turn v into a partial loss, if needed, otherwise we would replace v with its remaining child. Hence, if $\mathcal{H}$ contains a duplication or a cut, it must be the

last binary event on the main path. Thus, all other binary events must be speciations, and we need exactly $\mathrm{D}_{S^*}(\sigma, \gamma)$ of them to reach σ from γ. Those speciations must lead to one full loss for each species $s \in \mathrm{V}(S)$. Hence, $\mathcal{H} \in P_0$.

- [$\mathcal{H}$ *contains exactly one transfer.*] Let v be the only transfer node. If $\sigma \leq \gamma$, then there must be a speciation above v so that $s(v)$ can be separated from γ since $\mathcal{H}$ cannot contain other transfers. Notice that $\mathcal{H}$ contains no partial losses, duplications, or cuts. In fact, any partial loss can be merged into the transfer. As for duplications and cuts, they can be removed if they do not save any loss, or merged into the transfer otherwise. Apart from the initial speciation if $\sigma \leq \gamma$, $\mathcal{H}$ contains no other speciations as those before (resp. after) the transfer can be removed by redirecting the transfer to start from a higher species (resp. to end at a lower species) that is still separated from γ (resp. σ). If $e(v) = \text{TrDup}$ or $X \not\subseteq Y$, then the invisible subtree of v must contain at least one full loss, unless $s(v) \in \mathrm{L}(S^*) - \mathrm{L}(S)$. If $s(v) \notin \mathrm{L}(S^*) - \mathrm{L}(S)$ and $e(v) = \text{TrCut}$, then either subtree of v must contain at least one full or partial loss if $x(v) \not\subseteq Y$. Hence, $\mathcal{H} \in P_1$.
- [$\mathcal{H}$ *contains exactly two transfers.*] Let v and v' be the only two transfer nodes. If $\sigma = \mathrm{r}(S^*)$, then $\mathcal{H}$ must contain at least one full loss; hence, $\mathcal{H}$ is not Pareto-optimal. If $\sigma \in \mathrm{L}(S^*) - \mathrm{L}(S)$, then, if $\sigma \leq \gamma$, we can remove both transfers, and, if $\sigma \parallel \gamma$, then we can remove v', in both cases without introducing additional events. If $\sigma \parallel \gamma$, then we have the following: If $e(v) = \text{TrDup}$, we can remove v' without adding new losses; If $X \subseteq Y$, then $x(v) \subseteq x(v') \subseteq x(w)$. Hence, we can also remove v'. If either of v or v' is such that $e(v) = \text{TrCut}$ or $e(v') = \text{TrCut}$, then $e(v) = \text{TrCut}$, as otherwise it is $e(v) = \text{TrDup}$, and there is a full loss below v. In addition, we can exchange the transfer types of v and v' so that $e(v) = \text{TrCut}$, $e(v') = \text{TrDup}$ and $s(v') \in \mathrm{L}(S^*)$, so as to save a full loss. If $x(v') \not\subseteq Y$ or $e(v') = \text{TrDup}$, then $s(v') \in \mathrm{L}(S^*) - \mathrm{L}(S)$, as otherwise we can reroute the first transfer toward an unsampled leaf so as to save a full loss. Hence, $\mathcal{H} \in P_2$. □

Finally, the set of all possible histories $\mathbb{H}^{\min}(\mathcal{T}, S)$ can be computed as the union of assignments of the root node of the synteny tree to all possible syntenies and species. This starts with a path from an empty synteny (i.e., an initial gain). In other words,

$$\mathbb{H}^{\min}(\mathcal{T}, S) = \bigoplus_{\substack{X \subseteq \mathcal{F} \\ \sigma \in \mathrm{V}(S^*)}} \mathrm{path}([\emptyset, \sigma], \mathrm{r}(\mathcal{T})[X, \sigma]) \otimes h(\mathrm{r}(\mathcal{T}), X, \sigma).$$

Hence, using Theorems 1 and 2, one can derive a dynamic programming algorithm to solve Problem 1. Due to the exponential size of the set of solutions, the time complexity of that algorithm is also exponential.

Theorem 3 (Number of minimal histories)**.** *Let $\mathcal{T} = \langle T, x, s \rangle$ be a synteny tree on a species tree S with gene family set $\mathcal{F}$. Then,*

$$|\mathbb{H}^{\min}(\mathcal{T}, S)| \in \mathcal{O}\Big((2^{|\mathcal{F}|}|\mathrm{V}(S)|^3)^{|\mathrm{V}(T)|}\Big).$$

Proof. Using Theorem 1, we obtain that $|h(v, X, \sigma)| \in \mathcal{O}(f(v))$ with

$$f(v) = \begin{cases} 1 & \text{if } v \text{ is a leaf,} \\ (2^{|\mathcal{F}|}|\mathrm{V}(S)|^2 p) \times f(v_\ell) \times f(v_\mathrm{r}) & \text{otherwise,} \end{cases}$$

where p is an upper bound on the number of possible paths. This is because, in each case of Theorem 1, up to all subsets $Y, Z \subseteq \mathcal{F}$ are tried along with up to all possible species pairs $\gamma_\ell, \gamma_\mathrm{r} \in \mathrm{V}(S^*)$, which has a number of nodes directly proportional to $|\mathrm{V}(S)|$. Using Theorem 2, we obtain $p \in \mathcal{O}(|\mathrm{V}(S)|)$. We obtain the desired result by solving the recurrence. □

6. Polynomial-Time Computation of Pareto-Optimal Event Vectors

We now address Problem 2. Given a synteny tree $\mathcal{T} = \langle T, x, s \rangle$ on a species tree S, similar to the way $h(v, X, \sigma)$ was previously used to recursively compute $\mathbb{H}^{\min}(\mathcal{T}, S)$, we define $\Lambda(v, X, \sigma)$ for computing $\mathrm{ev}^{\min}(\mathcal{T}, S)$ as $\Lambda(v, X, \sigma) = \{\mathrm{ev}(\mathcal{H}) \mid \mathcal{H} \in h(v, X, \sigma)\}$.

To compute $\Lambda(v, X, \sigma)$, we replace the algebra of Theorem 1 with an algebra where:

- The base cases are $\{(0, 0, 0, 0, 0, 0)\}$ and $\emptyset$;
- $A \oplus B$ is the union of vectors from A and B, retaining only the Pareto-optimal ones;
- $A \otimes B$ sums the pairs of vectors from $A \times B$, retaining only the Pareto-optimal ones.

Additional simplifications can reduce the complexity of computing $\Lambda(v, X, \sigma)$. We start by showing that, when adapted for Problem 2, it is sufficient to try a constant number of syntenies $X, Y, Z \subseteq \mathcal{F}$ at each step of the recurrence of Theorem 1. First, let us show that it is sufficient to place the gain event for each gene family at the lowest common ancestor of the leaves they appear in.

Definition 10 (Gain positions). *Let $\mathcal{T} = \langle T, x, s \rangle$ be a synteny tree on $\mathcal{F}$, $\Gamma \in \mathcal{F}$, $v \in \mathrm{V}(T)$, and $X \subseteq \mathcal{F}$. We thus define*

$$\mathrm{lca}(\Gamma) = \mathrm{lca}_T\{l \in \mathrm{L}(T) \mid \Gamma \in x(l)\},$$
$$f(v, X) = \{\Gamma \in X \mid \mathrm{lca}(\Gamma) \not\succ v\}.$$

Lemma 1 (Gains at the LCA). *If $\mathcal{T} = \langle T, x, s \rangle$ is a synteny tree on $\mathcal{F}$, $v \in \mathrm{V}(T)$, and $X \subseteq \mathcal{F}$, then* $\min_{\preceq}(\Lambda(v, X, \sigma) \cup \Lambda(v, f(v, X), \sigma)) = \Lambda(v, f(v, X), \sigma)$.

Proof. Let $\mathcal{H} \in h(v, X, \sigma)$. Let $\Gamma \in X - f(v, X)$, if there is any. By definition, $\mathrm{lca}(\Gamma) > v$. Assume w.l.o.g. that $\mathrm{lca}(\Gamma) \geq v_\ell$. Consider the history $\mathcal{H}'$ in which Γ is removed from $x(v)$ and all $x(w)$ for $w \in T[v_\mathrm{r}]$, thereby removing any invalid loss event created in the process, and in which the gain event for Γ is moved to be the parent of v_r (potentially merging it with other gains). Then, $\mathrm{ev}(\mathcal{H}') \preceq \mathrm{ev}(\mathcal{H})$, since we only potentially removed losses from $\mathcal{H}$ and the number of gains is not part of the event vector. Let $\mathcal{H}^*$ be the history obtained after repeating this process for each $\Gamma \in X - f(v, X)$. Clearly, $\mathcal{H}^* \in h(v, f(v, X), \sigma)$; hence, $e(\mathcal{H}^*) \in \Lambda(v, f(v, X), \sigma)$ and $\mathrm{ev}(\mathcal{H}^*) \preceq \mathrm{ev}(\mathcal{H})$. □

In a similar way to Lemma 5 in [10], we now show that only two synteny contents have to be tried at each step of the recurrence as any synteny larger than the minimal required gene families (as formally defined below) leads to the same set of optimal event vectors.

Definition 11 (Minimal synteny contents). *For any $v \in \mathrm{V}(T)$, we define*

$$\mathrm{gain}(v) = \{\Gamma \in \mathcal{F} \mid v = \mathrm{lca}(\Gamma)\},$$
$$x^{\min}(v) = \begin{cases} x(v) & \text{if } v \text{ is a leaf,} \\ x^{\min}(v_\ell) \cup x^{\min}(v_\mathrm{r}) - (\mathrm{gain}(v_\ell) \cup \mathrm{gain}(v_\mathrm{r})) & \text{otherwise.} \end{cases}$$

Lemma 2 (Two choices of synteny contents). *Let $\mathcal{T} = \langle T, x, s \rangle$ be a synteny tree on a species tree S and v be a node of T. For any $X, X' \supsetneq x^{\min}(v)$ such that $X = f(v, X)$ and $X' = f(v, X')$, $\Lambda(v, X, \sigma) = \Lambda(v, X', \sigma)$.*

Proof. We proceed by induction on the depth of v in T. If v is a leaf, then no valid history exists; hence ,$\Lambda(v, X, \sigma) = \Lambda(v, X', \sigma) = \emptyset$. Otherwise, let $\mathcal{H} = \langle H, e, x, s \rangle \in h(v, X, \sigma)$, $Y = x(v_\ell)$, $Z = x(v_\mathrm{r})$, and denote $\mathcal{H}_\ell \in h(v_\ell, Y, s(v_\ell))$ (resp. $\mathcal{H}_\mathrm{r}$ for v_r) to be the subhistory of $\mathcal{H}$ below v_ℓ (resp. v_r).

Let $G = X - x^{\min}(v)$, and $G' = X' - x^{\min}(v)$. If $\Gamma \in G$, then $\Gamma \notin x^{\min}(v)$, thus implying that $\Gamma \notin x^{\min}(v_\ell)$ and $\Gamma \notin x^{\min}(v_\mathrm{r})$ since $\Gamma \notin \mathrm{gain}(v_\ell)$ and $\Gamma \notin \mathrm{gain}(v_\mathrm{r})$ because $\mathrm{lca}(\Gamma) \not\succ v$. This implies that $X \not\subseteq Y, Z$. Using the same argument for $\Gamma' \in G'$, we deduce that $X' \not\subseteq Y, Z$.

Suppose that $Y \supsetneq x^{\min}(v_\ell)$. Let $Y' = (Y - G) \cup G'$. Noting that $Y' \supsetneq x^{\min}(v_\ell)$, and using the induction hypothesis, we see that $\Lambda(v_\ell, Y, s(v_\ell)) = \Lambda(v_\ell, Y', s(v_\ell))$. There exists a history $\mathcal{H}'_\ell \in h(v_\ell, Y', s(v_\ell))$ such that $\mathrm{ev}(\mathcal{H}_\ell) = \mathrm{ev}(\mathcal{H}'_\ell)$. The same argument applies to v_r, thereby yielding a history $\mathcal{H}'_\mathrm{r}$ such that $\mathrm{ev}(\mathcal{H}_\mathrm{r}) = \mathrm{ev}(\mathcal{H}'_\mathrm{r})$.

Suppose that $Y = x^{\min}(v_\ell)$. Then, there must be an event to lose $X - Y$ on the path from v to v_ℓ. That event can also be used to lose $X' - Y$. In this case, we let $\mathcal{H}'_\ell = \mathcal{H}_\ell$. The same argument applies to v_r.

Finally, consider the history $\mathcal{H}'$ that is obtained by replacing $\mathcal{H}_\ell$ with $\mathcal{H}'_\ell$, $\mathcal{H}_\mathrm{r}$ by $\mathcal{H}'_\mathrm{r}$, and setting $x(v) = X'$. Clearly, $\mathrm{ev}(\mathcal{H}) = \mathrm{ev}(\mathcal{H}')$ and $\mathcal{H}' \in h(v, X', \sigma)$; hence, $\mathrm{ev}(\mathcal{H}) \in \Lambda(v, X', \sigma)$. □

Using the recurrence from Theorem 1 adapted to use the algebra over Pareto-sets of vectors and simplified, as shown in Lemmas 1 and 2, we can obtain a dynamic programming algorithm for solving Problems 2 and 2′.

Theorem 4. *Problem 2 can be solved in time* $\mathcal{O}((mn)^9 \log(mn))$ *and space* $\mathcal{O}((mn)^5)$*, and Problem 2′ in time* $\mathcal{O}((mn)^9 n \log(mn))$ *and space* $\mathcal{O}((mn)^5 n)$*, where* $m = |\mathrm{V}(\mathcal{T})|$ *and* $n = |\mathrm{V}(S)|$ *are, respectively, the numbers of the nodes in the synteny and species trees.*

Proof. Let us first show that $|\mathrm{ev}^{\min}(\mathcal{T}, S)| \in \mathcal{O}((mn)^4)$. From Theorem 2, the number of events in a history $\mathcal{H} \in \mathrm{path}([X, \sigma], [Y, \gamma])$ is in $\mathcal{O}(n)$ (attained for histories in P_0, while those in P_1 and P_2 have a constant number of events). From Theorem 1, the histories $\mathcal{H} \in \mathbb{H}^{\min}(\mathcal{T}, S)$ are extensions of the synteny tree T, which are obtained by inserting such paths; hence, the number of events in such histories is in $\mathcal{O}(mn)$. As the vectors have five components, we obtain the desired result by adapting the argument from Lemma 3.1 in [15].

We solve Problem 2 using a dynamic programming table for $\Lambda(v, X, \sigma)$. The number of entries in the table is m for v, two for X (as per Lemma 2), and n for σ. Hence, the space complexity result follows from the bound on the size of $\mathrm{ev}^{\min}(\mathcal{T}, S)$ shown above.

As for the time complexity, following the recurrence adapted from Theorem 1, to compute each entry we need to consider four options for Y and Z, as well as the up to n^3 options for γ_ℓ and γ_r (or γ_k and γ_t) and γ_i. However, it is possible to reduce these n^3 species options to a constant number of operations at each step by simultaneously computing three separate tables *in*, *inAlt*, and *out*, as defined by Weiner and Bansal [14] and explained in their Algorithm 1 and the proof of their Theorem 1. For each of these options, the $\oplus$ and $\otimes$ operators need to be used a constant number of times.

The $\oplus$ and $\otimes$ operations on two sets containing k Pareto-vectors can be implemented in time $\mathcal{O}(k \log k)$ [18] and $\mathcal{O}(k^2 \log k)$, respectively [15]. In our case, it follows from the bound on the size of $\mathrm{ev}^{\min}(\mathcal{T}, S)$ that both operators can be implemented in time $\mathcal{O}((mn)^4 \log(mn))$ and $\mathcal{O}((mn)^8 \log(mn))$, respectively.

To solve Problem 2′, we need to be able to reconstruct one of the histories leading to each Pareto-optimal event vector. To that end, we associate additional pieces of information to each vector: the root node of the history, two pointers to two sub-histories, and two paths that lead to those histories (see Figure 2). Since a path can contain, at most, $\mathcal{O}(n)$ events, the time and space complexities of this method to solve Problem 2′ are obtained by adding a factor of n to those of Problem 2. □

7. Efficient Computation of Minimum-Cost Histories

We finally address Problem 3. Let $\mathcal{T} = \langle T, x, s \rangle$ be a synteny tree on a species tree S, and let $\delta \in (\mathbb{R}^+ \cup \{\infty\})^5$ be an event cost vector. Similar to the way through which $h(v, X, \sigma)$ and $\Lambda(v, X, \sigma)$ were previously used to compute $\mathbb{H}^{\min}(\mathcal{T}, S)$ and $\mathrm{ev}^{\min}(\mathcal{T}, S)$, we define $c(v, X, \sigma) = \{\mathrm{c}(\mathcal{H}) \mid \mathcal{H} \in h(v, X, \sigma)\}$ to compute $c^{\min}(\mathcal{T}, S)$.

To compute $c(v, X, \sigma)$, we replace the algebra of Theorem 1 with an algebra where:

- The base cases are 0 and ∞;
- $A \oplus B$ is the minimum of A and B;

- $A \otimes B$ is the sum of A and B.

This is the so-called min-plus or "tropical" semiring. Notice that Lemmas 1 and 2 still apply since if $\Lambda(v, X, \sigma) = \Lambda(v, X', \sigma)$, then $c(v, X, \sigma) = c(v, X', \sigma)$.

Theorem 5. *Problem 3 can be solved in time and space* $\mathcal{O}(mn)$*, and Problem 3′ in time and space* $\mathcal{O}(mn^2)$*, where* $m = |\mathrm{V}(\mathcal{T})|$ *and* $n = |\mathrm{V}(S)|$*.*

Proof. Both results follow from the proof of Theorem 4 due to the fact that the minimum and sum operators can be computed in constant time, thereby removing the $(mn)^8 \log(mn)$ factor from the time complexity, and also due to the fact that only a single value needs to be stored for each entry of the dynamic programming table, thus removing the $(mn)^4$ factor from the space complexity. □

8. Results

The CRISPR–Cas module is an adaptive system that allows prokaryotes to defend against invading viruses and plasmids. Its fame is due to the development of the CRISPR-Cas9 genome editing technology, which is one of the most reliable and accurate "molecular scissors" to date. An important part of any CRISPR-Cas system is the operon of the associated Cas genes playing various roles in the defense machinery. As the microbial function of CRISPR-Cas systems highly depends on the syntenic organization of Cas genes, elucidating the evolution of these syntenies is crucial.

In [11], taking the Makarova et al. [19] CRISPR-Cas classification of Class 1 as the synteny tree—and considering a dataset of 15 bacterial species, as well as the species tree topology inferred in [20]—we recovered an evolutionary history that is broadly consistent with that proposed in [21], with the CRISPR-Cas emergence inferred at the root of Terrabacteria. However, some inconsistencies were observed. For example, in the Proteobacteria subtree, the *SuperDTL* algorithm inferred an unlikely scenario with an ancestral synteny duplication before the LCA of *Shewanella putrefaciens*, *Vibrio crassostreae*, *Yersinia pseudotuberculosis*, and *Escherichia coli*, thereby resulting in a succession of three consecutive full synteny losses along the branch to *E. coli*.

Here, we used Synesth on the same dataset with the same event costs for Loss, Dup, and TrDup events and by choosing intermediate costs for the new Cut and TrCut events. The used event costs were $(\delta_{\mathrm{Dup}} = 2, \delta_{\mathrm{Cut}} = 2.5, \delta_{\mathrm{TrDup}} = 4, \delta_{\mathrm{TrCut}} = 4.5, \delta_{\mathrm{Loss}} = 1)$. Almost the same history was obtained but with the above unlikely scenario replaced by a speciation (on the branch separating the ancestor of *Geobacter sulfurreducens* to the LCA of *Thioalkalivibrio* and *Shewanella putrefaciens*) copying the ancestral synteny to an unsampled species, which was later transferred back to *E. coli* (see Figure 3).

Choosing the appropriate event costs constitutes one of the main challenges of tree reconciliation. Slight changes to the event costs may lead to significantly different history outputs. We use a similar approach to that which was developed by Libeskind-Hadas et al. [15] to display a summary of the solution space over all possible event cost choices. In order to represent the solutions in a 2D plot, we normalized the cost of Loss events to 1 and set an equal cost for Dup and Cut and for TrDup and TrCut. The Pareto-optimal vectors were condensed to three dimensions as follows: $(c_{\mathrm{Dup}} + c_{\mathrm{Cut}}, c_{\mathrm{TrDup}} + c_{\mathrm{TrCut}}, c_{\mathrm{Loss}})$. The resulting plot is given in Figure 4, in which each color-coded region corresponds to a set of event costs that give rise to exactly the same set of Pareto-optimal histories.

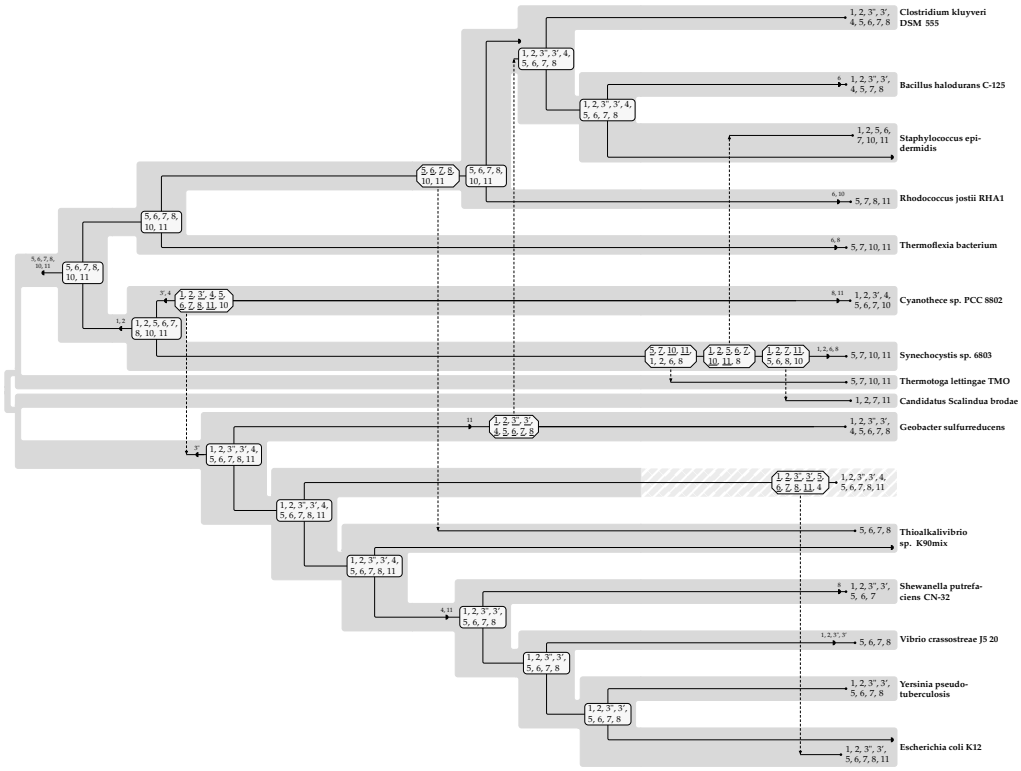


Figure 3. Output of Synesth for the CRISPR-Cas Class 1 dataset when asked for one minimum-cost history with event costs ($\delta_{\text{Dup}} = 2, \delta_{\text{Cut}} = 2.5, \delta_{\text{TrDup}} = 4, \delta_{\text{TrCut}} = 4.5, \delta_{\text{Loss}} = 1$).

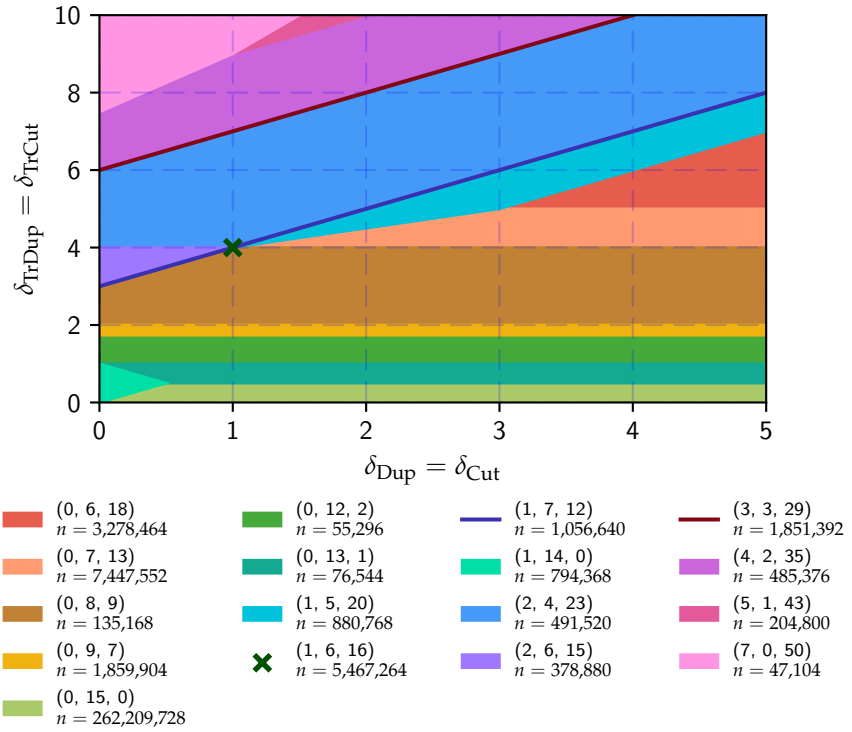


Figure 4. The event cost landscape for the solutions returned by Synesth for the Class 1 Cas gene synteny dataset (see text). The cost of a loss event was fixed to 1. For each color region, the legend shows a condensed event count vector of the form $(c_{\text{Dup}} + c_{\text{Cut}}, c_{\text{TrDup}} + c_{\text{TrCut}}, c_{\text{Loss}})$, where "$n$" indicates the number of distinct Pareto-optimal histories for any set of event costs in that region.

9. Conclusions

Synesth is a flexible tool for tree reconciliation, which allows for a wide range of segmental events that addresses the inevitable incompleteness of the input dataset in terms of unsampled species, as well as offers various optimization and output criteria to the user. Moreover, its time complexity brings it up to the level of the most time-efficient reconciliation algorithms, such as *ecceTERA* [12] and *RANGER-DTLx* [14], but for an evolutionary model with events involving sets of genes rather than single genes.

The inductive characterization of Pareto-optimal histories allows for an exhaustive exploration of the solution space. Taking advantage of this flexibility, future extensions of the computational aspects of this work may address the problem of formally characterizing this space in terms of constructing equivalence classes or normalized histories, thus uniformly sampling the space of histories and assigning confidence values to predicted histories.

Further extensions of the model would also be worth investigating. For example, representing syntenies as sets does not capture the information of gene orders and multiplicities. Accounting for gene orders would require allowing rearrangement events, and this may significantly increase the computational complexity of the problem. Allowing gene repetitions inside syntenies would require representing them as multisets, which would break some of the assumptions required for the algorithmic approach presented in this work. However, probably, the most questionable limitation of the model is the absence of synteny fusions while synteny fissions are allowed; thus, it favors large syntenies up to the root of the tree. Note, however, that including fusions will require adding reticulated nodes. It will be interesting to see whether our dynamic programming scheme can be generalized to such phylogenetic networks or if it makes the problem NP-hard, in which case tree decomposition methods may be explored [22].

Another important challenge not addressed in this paper is how to obtain an input synteny tree. In fact, phylogenetic methods instead output sets of gene trees—one for each gene family. If the individual gene trees are "consistent", i.e., with no contradictory phylogenetic information, then a tree displaying them all can be obtained. However, even in this case, there may be an exponential number of such supertrees. In [10], the suggested solution was to test each possible supertree and retain the one leading to the most parsimonious reconciliation. An alternative would be to simultaneously construct and reconcile a supertree with a given species tree. This opens the door to interesting future investigations.

Author Contributions: Conceptualization, N.E.-M.; methodology, M.D. and N.E.-M.; software, M.D.; writing—original draft, M.D. and N.E.-M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Sciences and Engineering Research Council of Canada (grant number RN000743) and Fonds de recherche du Québec—Nature et technologies (grant number 335893).

Data Availability Statement: Data sharing is not applicable to this article.

Acknowledgments: We would like to thank Arnaud Grandisson for his work on the graphical output and Mathieu Gascon for his help and invaluable comments on the model and algorithm.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. 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. *Syst. Biol.* **1979**, *28*, 132–163. [CrossRef]
2. Bansal, M.S.; Alm, E.J.; Kellis, M. Efficient algorithms for the reconciliation problem with gene duplication, horizontal transfer and loss. *Bioinformatics* **2012**, *28*, i283–i291. [CrossRef]
3. Donati, B.; Baudet, C.; Sinaimeri, B.; Crescenzi, P.; Sagot, M.F. EUCALYPT: Efficient tree reconciliation enumerator. *Algorithms Mol. Biol.* **2015**, *10*, 3. [CrossRef]
4. Tofigh, A.; Hallett, M.; Lagergren, J. Simultaneous identification of duplications and lateral gene transfers. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2011**, *8*, 517–535. [CrossRef] [PubMed]

5. El-Mabrouk, N.; Noutahi, E. Gene family evolution: An algorithmic framework. In *Bioinformatics and Phylogenetics*; Springer International Publishing: Berlin/Heidelberg, Germany, 2019; pp. 87–119. [CrossRef]
6. Duchemin, W.; Anselmetti, Y.; Patterson, M.; Ponty, Y.; Bérard, S.; Chauve, C.; Scornavacca, C.; Daubin, V.; Tannier, E. DeCoSTAR: Reconstructing the ancestral organization of genes or genomes using reconciled phylogenies. *Genome Biol. Evol.* **2017**, *9*, 1312–1319. [CrossRef] [PubMed]
7. Duchemin, W. Phylogeny of Dependencies and Dependencies of Phylogenies in Genes and Genomes. Ph.D. Thesis, Université de Lyon, Lyon, France, 2017.
8. Dondi, R.; Lafond, M.; Scornavacca, C. Reconciling multiple genes trees via segmental duplications and losses. *Algorithms Mol. Biol.* **2019**, *14*, 7. [CrossRef]
9. Paszek, J.; Gorecki, P. Efficient algorithms for genomic duplication models. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2018**, *15*, 1515–1524. [CrossRef] [PubMed]
10. Delabre, M.; El-Mabrouk, N.; Huber, K.T.; Lafond, M.; Moulton, V.; Noutahi, E.; Castellanos, M.S. Evolution through segmental duplications and losses: A super-reconciliation approach. *Algorithms Mol. Biol.* **2020**, *15*, 12. [CrossRef]
11. Anselmetti, Y.; Delabre, M.; El-Mabrouk, N. Reconciliation with segmental duplication, transfer, loss and gain. In *Comparative Genomics*; Springer International Publishing: Berlin/Heidelberg, Germany, 2022; pp. 124–145. [CrossRef]
12. Jacox, E.; Chauve, C.; Szöllősi, G.J.; Ponty, Y.; Scornavacca, C. ecceTERA: Comprehensive gene tree-species tree reconciliation using parsimony. *Bioinformatics* **2016**, *32*, 2056–2058. [CrossRef] [PubMed]
13. Szöllősi, G.J.; Tannier, E.; Lartillot, N.; Daubin, V. Lateral gene transfer from the dead. *Syst. Biol.* **2013**, *62*, 386–397. [CrossRef] [PubMed]
14. Weiner, S.; Bansal, M.S. Improved duplication-transfer-loss reconciliation with extinct and unsampled lineages. *Algorithms* **2021**, *14*, 231. [CrossRef]
15. Libeskind-Hadas, R.; Wu, Y.C.; Bansal, M.S.; Kellis, M. Pareto-optimal phylogenetic tree reconciliation. *Bioinformatics* **2014**, *30*, i87–i95. [CrossRef] [PubMed]
16. David, L.A.; Alm, E.J. Rapid evolutionary innovation during an Archaean genetic expansion. *Nature* **2011**, *469*, 93–96. [CrossRef] [PubMed]
17. Libeskind-Hadas, R. Tree reconciliation methods for host-symbiont cophylogenetic analyses. *Life* **2022**, *12*, 443. [CrossRef] [PubMed]
18. Saule, C.; Giegerich, R. Pareto optimization in algebraic dynamic programming. *Algorithms Mol. Biol.* **2015**, *10*, 22. [CrossRef]
19. Makarova, K.S.; Wolf, Y.I.; Iranzo, J.; Shmakov, S.A.; Alkhnbashi, O.S.; Brouns, S.J.J.; Charpentier, E.; Cheng, D.; Haft, D.H.; Horvath, P.; et al. Evolutionary classification of CRISPR–Cas systems: A burst of class 2 and derived variants. *Nat. Rev. Microbiol.* **2020**, *18*, 67–83. [CrossRef] [PubMed]
20. Coleman, G.A.; Davín, A.A.; Mahendrarajah, T.A.; Szánthó, L.L.; Spang, A.; Hugenholtz, P.; Szöllősi, G.J.; Williams, T.A. A rooted phylogeny resolves early bacterial evolution. *Science* **2021**, *372*, 588. [CrossRef]
21. Koonin, E.V.; Makarova, K.S. Evolutionary plasticity and functional versatility of CRISPR systems. *PLoS Biol.* **2022**, *20*, e3001481. [CrossRef]
22. Scornavacca, C.; Weller, M. Treewidth-based algorithms for the small parsimony problem on networks. *Algorithms Mol. Biol.* **2022**, *17*, 15. [CrossRef]