JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 20, Number 9, 2013

Pp. 643–659
DOI: 10.1089/cmb.2013.0057

The Duplication-Loss Small Phylogeny Problem: From Cherries to Trees

SANDRO ANDREOTTI,[1,2] KNUT REINERT,[1] and STEFAN CANZAR[3,*]

ABSTRACT

The reconstruction of the history of evolutionary genome-wide events among a set of related organisms is of great biological interest since it can help to reveal the genomic basis of phenotypes. The sequencing of whole genomes faciliates the study of gene families that vary in size through duplication and loss events, like transfer RNA. However, a high sequence similarity often does not allow one to distinguish between orthologs and paralogs. Previous methods have addressed this difficulty by taking into account flanking regions of members of a family independently. We go one step further by inferring the *order* of genes of (a set of) families for ancestral genomes by considering the order of these genes on sequenced genomes. We present a novel branch-and-cut algorithm to solve the *two species small phylogeny* problem in the evolutionary model of duplications and losses. On average, our implementation, DupLoCut, improves the running time of a recently proposed method in the experiments on six *Vibrionaceae* lineages by a factor of ∼200. Besides the mere improvement in running time, the efficiency of our approach allows us to extend our model from cherries of a species tree, that is, subtrees with two leaves, to the *median of three species* setting. Being able to determine the median of three species is of key importance to one of the most common approaches to ancestral reconstruction, and our experiments show that its repeated computation considerably reduces the number of duplications and losses along the tree both on simulated instances comprising 128 leaves and a set of *Bacillus* genomes. Furthermore, in our simulations we show that a reduction in cost goes hand in hand with an improvement of the predicted ancestral genomes. Finally, we prove that the small phylogeny problem in the duplication-loss model is NP-complete already for two species.

1. INTRODUCTION

In the course of evolution, genome-wide changes either (i) rearrange the order of the genes or (ii) modify the content. The former class of changes results from inversions, transpositions, and translocations. The latter have an effect on the number of gene copies that are either inserted, lost, or duplicated. In particular, gene duplication plays a decisive role as a source of raw genetic material for the creation of

[1]Department of Mathematics and Computer Science, Institute of Computer Science, Freie Universität Berlin, Berlin, Germany.
[2]International Max Planck Research School for Computational Biology and Scientific Computing, Berlin, Germany.
[3]Center for Computational Biology, McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins University School of Medicine, Baltimore, Maryland.
*Corresponding author.

evolutionary novelties (Ohno, 1970). Although prevalent in all three domains of life (Zhang, 2003), an estimated (Lynch and Conery, 2000) rate of 0.01 per gene per million years underlines its importance for eukaryotes such as *Homo sapiens, Mus musculus, D. melanogaster, C. elegans,* and *Arabidopsis thaliana*. The reconstruction of the history of such events among a set of (related) organisms is of great biological interest since it can help to reveal the genomic basis of phenotypes.

The sequencing of whole genomes allows us to study macro-evolutionary processes like gene duplications and losses (through deletion or pseudogenization) at an unprecedented resolution. Indeed, the size of gene families has been shown to largely vary, even among 12 *Drosophila* species (Hahn et al., 2007). A prominent example of a gene family that is continuously duplicated and lost is transfer RNA (tRNA) (Rogers et al., 2010; Swenson et al., 2008; Withers et al., 2006). In *Escherichia coli*, for example, the rate at which tRNA genes evolve by duplication and loss events has been estimated to be on the order of one per million years (Withers et al., 2006). Since tRNA as an adapter molecule between a codon in the mRNA and the amino acids is an essential element in the translation of RNA into proteins, reconstructing their evolutionary history among species might lead to new insights into the translationary machinery.

At the same time, studying the evolution of tRNA is particularly challenging, since the high sequence similarity of functionally equivalent tRNAs often does not allow one to distinguish between paralogs and orthologs (Withers et al., 2006). The approach developed in Rogers et al. (2010) tries to address this difficulty by considering the environment of tRNAs, more specifically, by taking into account the order of genes in tRNA flanking regions. Roughly speaking, tRNA genes that map to the same region of *D. melanogaster* form orthologous sets. Duplications and losses are then distributed on the branches of the drosophila tree by a maximum parsimony criteria, that is, by minimizing the number of these evolutionary events.

Inferring the *order* of genes of (a set of) families for ancestral genomes by taking into account the order of these genes on sequenced genomes goes one step further. It implicitly declares genes as orthologs or paralogs and respects the order of *all* genes among all studied families simultaneously. Furthermore, it specifies the direction of duplication events, a problem that was separately addressed in Han and Hahn (2009) based on synteny.

Most of the methods for phylogenetic inference solve one of two optimality criteria, the *maximum parsimony* (MP) or the *maximum likelihood* criterion. Although potentially more powerful, the latter makes explicit assumptions about the evolutionary process that produced the sequences. However, little is known about the mechanisms controlling the evolution of gene families like tRNA (Rogers et al., 2010). In contrast, the MP criterion does not rely on such assumptions and was the criterion according to which duplications and losses were distributed along the tree in Rogers et al. (2010) and Han and Hahn (2012). In a recent work by Holloway et al. (2012), the authors have observed that the problem of inferring the most parsimonious ancestor of two genomes in the duplication-loss model can be cast into an alignment problem, since the order of genes is preserved by these content-modifying operations of type (ii). Although an alignment is generally favorable from a computational perspective, we show that it is NP-hard to solve. Being theoretically hard for cherries, i.e., trees with two leaves, we cannot hope for an exact solution for large sets of species and more complex trees. However, we show that we can find an ''almost'' exact median of three species in reasonable time. The *median of three* is of key importance to one of the most common approaches to ancestral reconstruction, and our experiments show that its repeated computation considerably reduces the number of duplications and losses and thus increases the number of orthologs along the tree.

A second favorable property of the considered evolutionary model is that duplications and losses are asymmetric events that can be uniquely assigned to one of the two genomes and thus an ancestral genome can immediately be obtained from a duplication/loss scenario.

1.1. Related work and our contribution

The duplication-loss alignment problem considered in this work is naturally related to the classical (multiple) sequence alignment problem (MSA) (Althaus et al., 2005). Various methods proposed for MSA, in particular the branch-and-cut algorithm by Althaus et al. (Althaus et al., 2005), are based on a graph-theoretic model of the problem. This *gapped alignment graph* represents the correspondence of genes via alignment edges and losses (insertions) through gap arcs, but it does not capture duplication events. Duplications, as pairs of an origin and a target, lead to hyperedges in the gapped alignment graph. As it is

the case for most problems on graphs, the pairwise alignment problem becomes harder to solve through this generalization to hypergraphs. While the classical pairwise alignment problem (without duplications) is efficiently solvable in polynomial time, we show that the extension to duplication events makes the pairwise alignment problem NP-hard.

Holloway et al. (2012) proposed an approach for the comparison of two ordered gene sequences under the duplication-and-loss model that is based on an integer linear programming (ILP) formulation of the problem. While the method in Holloway et al. (2012) iteratively adds cycle constraints to the ILP, we have developed this idea further into a cutting plane algorithm. One class of cuts, the maximal clique inequalities, comprises solely the targets of duplications and can therefore be separated analogously to the MSA case. Additionally, we introduce cuts that exploit insights into the combinatorial structure of duplication events, the *lifted duplication cycles* and the *duplication islands*. We show that they can be separated efficiently and lead to a branch-and-cut algorithm that outperforms the previous method (Holloway et al., 2012) by several orders of magnitude, while also guaranteeing the solution to be provably optimal.

This boost in efficiency allows us to extend the two-species comparison to the inference of ancestral genomes of a set of species along a given phylogenetic tree. A widely used practice for this so-called *small phylogeny problem* is the *steinerization* method (Sankoff and Blanchette, 1997; Blanchette et al., 1997). Starting with some initial ancestral genomes, ancestral genomes are repeatedly replaced by the *median* of the neighboring genomes in the tree until a local optimum is reached. Therefore, we generalize the pairwise genome comparison model introduced by Holloway et al. to the median of a threen-species problem. A limitation of our median model is that it does not capture the subsequent modification of the target of a duplication from the ancestor of a median to *both* its descendants simultaneously. Nevertheless, as our experiments show, the median computed in our model provides a means of reducing the number of duplications and losses induced by the tree both on simulated and real-world instances and improves the prediction of ancestral genomes in our simulations. Furthermore, comparing the three neighboring genomes simultaneously allows us to relax a restriction of the model proposed in Holloway et al. (2012) (see Section 4). In Holloway et al. (2012), the two-species variant was applied in a bottom-up fashion to pairs of nodes in the tree, which can provide initial ancestral genomes for the steinerization method. Alternatively, in each step, the iterative local optimization method in Kovác et al. (2011) proposes a candidate set for ancestral genomes and computes the best one by dynamic programming. The related problem of labeling a given alignment of two genomes by duplications and losses was recently shown to be APX-hard (Dondi and El-Mabrouk, 2012). We show that the problem of finding a maximum parsimony ancestral genome of two given ordered gene sequences under the duplication-and-loss model is NP-hard.

2. PRELIMINARIES

We start with some basic definitions that are adopted from Holloway et al. (2012). A *genome* is a string over an alphabet Σ, whose characters represent specific gene families. Given two genomes, G^1 and G^2, and a set of allowed evolutionary operations, $\mathcal{O}$, a sequence of n operations $O \in \mathcal{O}$ that transforms G^1 into G^2 is called an *evolutionary history*, $O_{G^1 \to G^2}$. Let $c(O_i)$ define the cost of the i-th operation in $O_{G^1 \to G^2}$. Then the cost of $O_{G^1 \to G^2}$ is defined as $\sum_{i=1}^{n} c(O_i)$. If G^1 is a *potential ancestor* of G^2, that is, there exists an evolutionary history from G^1 to G^2, we define $c(G^1 \to G^2)$, the cost to transform G^1 into G^2, to be the minimal cost over all possible histories transforming G^1 into G^2. Given a tree $T = (V, E)$ with its leaves representing different genomes, the *small phylogeny problem* asks to find an assignment of genomes to the internal nodes of T such that the total cost $\sum_{(u,v)\in E} c(G(u) \to G(v))$ is minimized, where $G(u)$ is the genome assigned to node u and u is the parent of v. In Section 3, we study the small phylogeny problem restricted to two species, the problem to which the work in Holloway et al. (2012) is devoted.

Definition 1. (Two-Species Small Phylogeny Problem (2-SPP)). Given two genomes, G^1 and G^2, and a set of evolutionary operations $\mathcal{O}$, determine the potential common ancestor G^* minimizing the cost $c(G^* \to G^1) + c(G^* \to G^2)$.

While, in general, set $\mathcal{O}$ can include operations that change the order of genes, such as reversals and transpositions, as well as operations that modify the content, such as losses and insertions, the evolutionary model proposed by Holloway et al. only allows for the following two operations:

– A *duplication* of size $k+1$ on genome $G = G_1 \dots G_n$ copies a substring $G[i, \dots, i+k]$, called the *origin*, to a location j of G outside the interval $[i, i+k]$, called the *target* of the duplication.
– A *loss* of size $k+1$ removes a substring $G[i, \dots, i+k]$ from genome G.

Furthermore, the model requires that origin and target of a duplication must be present as contiguous blocks in the genomes:

Definition 2. An evolutionary history $O_{G^1 \to G^2}$ is called a visible history and G^1 a visible ancestor of G^2 if no duplication in $O_{G^1 \to G^2}$ is modified subsequently by inserting (by duplication) or deleting (by loss) genes from its origin or target.

With a slight abuse of notation, we also call pairs of evolutionary histories $(O_{G^* \to G^1}, O_{G^* \to G^2})$ from a potential common ancestor G^* of two genomes G^1 and G^2 a *visible history* if *both* histories $O_{G^* \to G^1}$ and $O_{G^* \to G^2}$ are visible. G^* is then called the *visible ancestor* of G^1 and G^2.

Since duplications and losses preserve the order of the genes and evolutionary histories are visible, Holloway et al. suggest to pose the two-species small phylogeny problem for the duplication and loss model as an alignment problem. An *alignment* of two genomes is obtained by introducing "$-$" $\notin \Sigma$ into the genomes such that the resulting strings are of same length and can be interpreted as rows of a two-dimensional alignment matrix $\mathcal{A}$. A column of $\mathcal{A}$ either *matches* two genes from Σ or aligns a gene $G[i]$ with "$-$". In the latter case, gene $G[i]$ is either the product of a duplication or part of a loss in the other genome. An interpretation of $\mathcal{A}$ as a sequence of duplications and losses is called a *labeling*.

In Holloway et al. (2012), it is shown that in the duplication-and-loss model of evolution, the Two Species Small Phylogeny Problem of visible ancestors is equivalent to the duplication-loss alignment problem.

Definition 3. Duplication-loss alignment problem: Let the cost of a labeled alignment be the sum of the costs of the operations implied by the labeling. Given two genomes, G^1 and G^2, compute a labeled alignment of G^1 and G^2 with minimum cost.

In Supplementary Material C (available online at www.liebertonline.com/cmb), we show that finding such a minimum cost solution is NP-hard.

Theorem 1. *For an instance of the duplication-loss alignment problem and an integer k, it is NP-complete to decide whether there exists an alignment of cost at most k.*

3. PROBLEM FORMULATION AND VALID INEQUALITIES

In this section, we describe a graph-theoretic representation of the duplication-loss alignment problem, which naturally leads to an ILP formulation similar to the one introduced in Holloway et al. (2012). However, we strengthen the *linear programming* (LP) relaxation by introducing three classes of valid inequalities.

Given two genomes, G^1 and G^2, the alignment graph $\mathcal{G} = (V^1 \cup V^2, E)$ is a complete bipartite graph with $V^i = \{v_l^i : 1 \le l \le |G^i|\}$. The nodes in V represent the genes in the genomes, with the j-th gene in G^i corresponding to node $v_j^i \in V$. Undirected edges $\{v_i^1, v_j^2\} \in E$ represent the *alignment* of a gene at position i in G^1 and the gene at position j in G^2 and implies orthology of the involved genes. If some alignment, $\mathcal{A}$, of G^1 and G^2 aligns gene $G^1[i]$ with $G^2[j]$, the corresponding edge $\{v_i^1, v_j^2\}$ is said to be realized by $\mathcal{A}$. Although an alignment of two genes, $G^1[i]$ and $G^2[j]$, does not imply any evolutionary operation, we assign a cost c_{ij} to the corresponding edge. This allows for a more general and powerful scoring scheme that might take into account gene conversions. The evolutionary model introduced in Section 2 can then be modeled by assigning cost ∞ to all edges representing the alignment of nonidentical genes, which causes the resulting alignment graph to be sparse.

Additionally, we define the sets of all possible duplications D^i, and losses L^i, for genomes G^i. For a duplication $d \in D^i$ with origin $G^i[\ell, \dots, \ell+k]$ and target $G^i[j, \dots, j+k]$ we define the functions $origin(d) := [\ell, \dots, \ell+k]$ and $target(d) := [j, \dots, j+k]$. Genes in origin and target are paralogous. Similarly, for a loss $l \in L^i$ that removes substring $G^i[j, \dots, j+k]$ from G^i, we define the function $span(l) := [j, \dots, j+k]$. Every duplication $d \in D^1 \cup D^2$ and every loss $l \in L^1 \cup L^2$ is assigned a cost c_d and c_l, respectively, that reflects the cost of the corresponding evolutionary operation.

A valid alignment has to satisfy three conditions. First, not all pairs of alignment edges can be realized simultaneously. Two crossing edges represent an ordering conflict of the involved genes. Therefore, we define two alignment edges $e_1 = \{v_i^1, v_j^2\}$ and $e_2 = \{v_k^1, v_l^2\}$ to be *incompatible* iff either $i \le k$ and $l \le j$ or $i \ge k$ and $l \ge j$. We let set $\mathcal{I}$ contain all pairs of incompatible alignment edges. Second, the biological interpretation of duplications induces a partial order "$\le$" (by chronology) on the set of duplication events: For a duplication d_1 whose target overlaps the origin of a duplication d_2, it must hold $d_1 \le d_2$, that is, d_1 occurred before d_2. Therefore, due to antisymmetry of partial orders, a sequence of duplications overlapping in their targets and origins must not form a cycle:

Definition 4. Duplication Cycle: A set of duplication events $D \subseteq D^i$ forms a duplication cycle iff there exists a permutation $d_1, d_2, \ldots, d_k$ of the elements in D such that

(a) $\text{origin}(d_i) \cap \text{target}(d_{i-1}) \neq \emptyset \quad \forall 2 \le i \le k$
(b) $\text{origin}(d_1) \cap \text{target}(d_k) \neq \emptyset$

And third, every gene in genome G^i must either be aligned to a gene in the other genome, be contained in the span of a loss $l \in L^i$, or lie in the target of a duplication $d \in D^i$. These three conditions are captured by three classes of constraints in the ILP formulation introduced in the next section.

3.1. An initial ILP formulation

In the remainder of the article, we will introduce and employ a model that assigns losses a linear cost, meaning that a loss of size x has cost $k \cdot x$. This allows us to simplify notation such that for every node $v \in V^i$, the loss event $l_v \in L^i$ denotes the loss of the single gene from G^i that is represented by node v. All our results can be generalized to arbitrary loss costs in a straightforward way by introducing, analogously to *gap arcs* in Althaus et al. (2005), a variable for each of the $\mathcal{O}(n^2)$ potential loss events to which we can assign affine, convex, and even position-dependent costs in the objective function. Occurrences of a variable for a loss event l_v have to be replaced then with the sum of variables for all loss events that contain node v.

Now, the three conditions formulated in the previous section naturally lead to the following ILP formulation. We have binary variables x_{ij} for every alignment edge $\{v_i^1, v_j^2\} \in E$, z_v for every possible loss event $l_v \in L^1 \cup L^2$, and y_d for every possible duplication $d \in D^1 \cup D^2$. Let D^* denote the set of all duplication cycles in D^1 and D^2. Then, the duplication-loss alignment problem is captured by the ILP formulation (1)–(5) (Fig. 1).

Constraints (2)–(4) capture the three conditions stated in the previous section, and thus, a solution to the ILP corresponds one-to-one to a solution for the duplication-loss alignment problem. But solving the ILP formulation directly by a standard ILP solver is not feasible for realistic instance sizes, as the number of

$$\min \quad \sum_{\{v_i^1, v_j^2\} \in E} c_{ij} x_{ij} + \sum_{v \in V} c_v z_v + \sum_{d \in D^1 \cup D^2} c_d y_d \tag{1}$$

$$\text{s.t.} \quad x_{ij} + x_{kl} \le 1 \qquad \forall \{\{v_i^1, v_j^2\}, \{v_k^1, v_l^2\}\} \in \mathcal{I} \tag{2}$$

$$\sum_{d \in D} y_d \le |D| - 1 \qquad \forall D \in D^* \tag{3}$$

$$z_{v_i^\ell} + \sum_{\{v_i^\ell, v_j^{\bar{\ell}}\} \in E} x_{ij} + \sum_{\substack{d \in D^\ell \\ i \in target(d)}} y_d = 1 \qquad \forall 1 \le i \le |G^\ell|,\ \ell \in \{1,2\},\ \bar{\ell} := 3 - \ell \tag{4}$$

$$x, y, z \in \{0, 1\} \tag{5}$$

FIG. 1. Initial ILP formulation.

possible duplication cycles, and thus the number of constraints (3), grows exponentially with the length of the genomes. Therefore, instead of enumerating all inequalities of class (3) in advance, we initially drop them completely.

Although the approach of Holloway et al. follows a similar strategy, the difference is significant. In Holloway et al. (2012), the problem is solved by iteratively solving the ILP formulation [initially without constraints (3)], adding violated duplication cycle inequalities to the ILP and re-solving until the original ILP is feasible, that is, no violated constraint (3) can be found.

In contrast, we embed a lifted version of constraint (3) into a cutting plane approach explained in Section 3.3. In short, in every node of the branch-and-bound tree, we try to cut off as much as possible from the set of feasible solutions to the LP relaxation without losing any feasible integral solution to (2)–(5). For this, we introduce in the next section a variant of the duplication cycle constraints that dominate (3). Furthermore, we identified other classes of valid inequalities that lead to a stronger LP relaxation, and therefore, to larger parts of the branch-and-bound tree to be pruned. In other words, we iteratively solve a strengthened version of the LP relaxation instead of the original ILP. As we will see in Section 5.1, this results in a dramatic improvement in performance. In Section 3.2, we define the classes of valid inequalities, and in Section 3.3, we show how to efficiently separate them. That is, given a (fractional) solution to the LP relaxation, how to identify a violated valid inequality.

3.2. Valid inequalities

In the following, we let $\mathcal{P}$ be the convex hull of the feasible solutions to the above ILP. Strong cuts for binary integer programs, that is, valid inequalities for $\mathcal{P}$, are often derived (Padberg, 1973) by studying the *incompatibility graph H*. Graph H contains a vertex for every binary variable and an edge for each pair of incompatible variables, that is variables that cannot have both value 1 in a feasible solution. Here, incompatibility of alignment edges (x-variables) and duplications (y-variables) follows directly from constraints (2)–(4), and graph H has node set $E \cup D$ and an edge between all pairs of incompatible alignment edges and duplications. Similar to the multiple sequence alignment approach (Althaus et al., 2005), we introduce maximal clique inequalities, which generalize (2).

Maximal clique inequalities Sets $K = K_E \cup K_D$ of pairwise incompatible alignment edges and duplications correspond precisely to the cliques of the incompatibility graph H, with $K_E \subseteq E$ and $K_D \subseteq D^1 \cup D^2$. If there is no alignment edge or duplication that is incompatible with all alignment edges and duplications in K, the corresponding clique is maximal. The following *maximal clique inequality* is valid for $\mathcal{P}$:

$$\sum_{e \in K_E} x_e + \sum_{d \in K_D} y_d \leq 1. \tag{6}$$

Since the incompatibility of duplications is determined by their target alone, maximal clique inequalities can be characterized similarly to the multiple sequence alignment problem (Althaus et al., 2005), which allows for their efficient separation (see Supplementary Material).

Duplication island inequalities The next class of constraints is motivated by the following observation. If a subset of genes is produced exclusively by duplications that also originate within this set, a duplication cycle, according to Definition 4, cannot be avoided. Therefore, Theorem 2 requires at least one of the genes in every set to be aligned, to be labeled as a loss, or to be the product of a duplication originating outside the set. Figure 2 gives an example of a fractional solution that is cut off by a duplication island inequality.

In the following, we consider duplications in G^1 and define $D := D^1$ and $V := V^1$. For duplications in G^2, the same holds. Consider the graph $\mathcal{G}'$ obtained by augmenting the alignment graph $\mathcal{G}$ with directed arcs A as follows: For every duplication $d \in D$, we add an arc from the node representing the ith element in *origin*(d) to the node representing the ith element in *target*(d), for all $i = 1 \ldots |origin(d)|$. Furthermore, for every $(u, v) \in A$, we let $\mathcal{D}((u, v))$ be the set of duplications d in D such that there exists an i with u representing the ith element in *origin*(d) and v representing the ith element in *target*(d). Then for any set $S \subseteq V$, $\mathcal{D}(V \setminus S, S)$ denotes the set of duplications inducing arcs in the cut-set of $(V \setminus S, S)$, that is,

$$\mathcal{D}(V \setminus S, S) = \bigcup_{\substack{(u,v) \in A: \\ u \in V \setminus S,\, v \in S}} \mathcal{D}((u, v)).$$

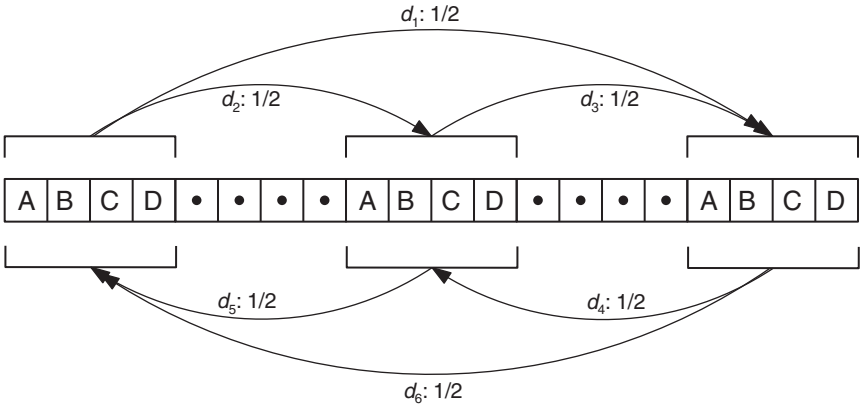

FIG. 2. A duplication island: Fractional values on the edges representing duplications $d_1, \ldots, d_6$ denote a feasible solution to the LP relaxation. In particular, no (lifted) duplication cycle inequality is violated. This solution is cut-off by the duplication island constraints.

Theorem 2. *For every set $S \subseteq V$, the following inequality is valid for $\mathcal{P}$:*

$$\sum_{v \in S} z_v + \sum_{v \in S} \sum_{k=1}^{|G^2|} x_{\{v, v_k^2\}} + \sum_{d \in \mathcal{D}(V \setminus S, S)} y_d \geq 1 \tag{7}$$

Proof. Assume, to the contrary, that the sum on the left-hand side of inequality (7) is 0. Let graph $\mathcal{G}''$ be obtained from graph $\mathcal{G}'$ by removing all alignment edges whose corresponding x-variable is 0 and all arcs $(u, v) \in A$ with $y_d = 0$ for all $d \in \mathcal{D}((u, v))$. Since every position in the genome must be covered [constraint (4)], and since $\sum_{v \in S} z_v + \sum_{v \in S} \sum_{k=1}^{|G^2|} x_{\{v, v_k^2\}} = 0$, exactly one incoming arc in A must be incident to every node $v \in S$. As $\sum_{d \in \mathcal{D}(V \setminus S, S)} y_d = 0$, these arcs must originate at a node in S. Thus, if we repeatedly traverse, starting at an arbitrary node in S, the unique incoming arc backward in $\mathcal{G}''$, we will never leave node set S and hence, ultimately close a cycle. Due to constraint (3), the corresponding solution is infeasible. ■

Lifted duplication cycle inequalities Again, we consider duplications in G^1 and define $D := D^1$ and $V := V^1$. For G^2, a symmetric argument applies. In this section, we introduce the *lifted duplication cycle inequalities*, a class of constraints that dominate (3). The high-level idea that this class of constraints is based on is similar to the one underlying the *lifted mixed cycle inequalities* introduced in Althaus et al. (2005). Consider a set of duplications $C \subseteq D$, which is partitioned into sets $C^1, \ldots, C^t$. If C satisfies

(C1) for $r = 1, \ldots, t$, all duplications in C^r are pairwise incompatible
(C2) every set $\{d_1, \ldots, d_t\}$, where d_r is chosen arbitrarily from C^r for $r = 1, \ldots, t$, forms a cycle according to Definition 4 then the inequality

$$\sum_{d \in C} y_d \leq t - 1 \tag{8}$$

is valid for $\mathcal{P}$. Inequalities (3) are a special case of (8) in which every set C^r has cardinality one. If, additionally,

(C3) C is maximal with respect to properties (C1) and (C2), that is, C cannot be extended without violating (C1) or (C2).

we call (8) a *lifted duplication cycle inequality*. Figure 3 gives an example of a fractional solution that is cut off by a lifted duplication cycle inequality.

Note that although the general definition of lifted duplication inequalities by (C1)–(C3) is analog to the general definition of lifted mixed cycles in Althaus et al. (2005), their specific characteristics differ significantly. The latter involves alignment edges among *three or more* sequences, whereas the former

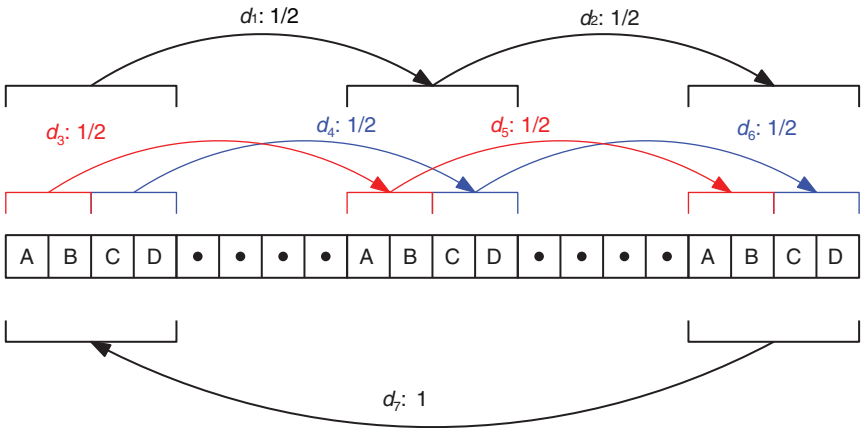

FIG. 3. A feasible fractional solution to the LP relaxation that is cut off by the lifted duplication cycle constraint. Edges d_1, d_3, d_2, d_5, d_7 together form a violated constraint, partitioned into $C^1 = \{d_1,d_3\}$, $C^1 = \{d_2, d_5\}$, and $C^3 = \{d_7\}$.

comprises duplications as hyperedges, that is, pairs of origin and target, in the *pairwise* setting. Consequently, their characterization in Proposition 1 is more involved and exhibits ''less structure'' that can be directly exploited in their separation.

3.3. A branch-and-cut approach

A *branch-and-cut* approach combines the ideas of *branch-and-bound* and *cutting planes*. First, the linear programming (LP) relaxation of the ILP is solved. If the optimal solution to the LP relaxation is not integral, cutting planes are used to cut off the current fractional solution without losing any of the feasible integral solutions. If no further cutting planes can be found, and the current solution is still not integral, the problem is split into subproblems, and the algorithm proceeds along the branches, provided the best feasible solution found so far does not allow one to prune them. In the next three theorems, we show that the cutting planes captured by the inequalities described in the previous section, respectively relaxations of them, can be found efficiently, which is crucial for the overall performance of the branch-and-cut algorithm.

Maximal clique inequalities can be separated similarly to the multiple sequence alignment problem (Althaus et al., 2005). The proof of the following theorem can be found in the Supplementary Material.

Theorem 3. *Let n be the length of the longer of the two genomes. Then for a given point* $(x^*, y^*, z^*) \in \mathbb{R}_+^{|E|+|D|+|V|}$, *it can be determined in time* $\mathcal{O}(n^3)$ *whether a maximal clique inequality* (6) *is violated.*

Next, we will show that a slightly relaxed version of constraint (7) can be separated efficiently. We define the multiplicity $\alpha(d, S)$ of a duplication d in the cutset of a cut $(V \setminus S, S)$:

$$\alpha(d, S) := |\{(u, v) \in A : u \in V \setminus S, v \in S \wedge \mathcal{D}(u, v) \ni d\}|. \tag{9}$$

Theorem 4. *For* $\ell \in \{1, 2\}$, *let* $D := D^\ell$, $V := V^\ell$, $n := |V|$, *and* $m = |V^{\bar{\ell}}|$, *where* $\bar{\ell}$ *is the complement of* ℓ *in* $\{1, 2\}$. *For a given point* $(x^*, y^*, z^*) \in \mathbb{R}_+^{|E|+|D|+|V|}$, *it can be determined in time* $\mathcal{O}(n^{3.5}\sqrt{|D|})$ *whether the following relaxation of a duplication island constraint* (7) *is violated.*

$$\sum_{v \in S} z_v^* + \sum_{v \in S} \sum_{k=1}^{m} x^*_{\{v,\, v_k^{\bar{\ell}}\}} + \sum_{d \in \mathcal{D}(V \setminus S,\, S)} \alpha(d, S) \cdot y_d^* \geq 1 \tag{10}$$

Proof. For an arbitrary node $s \in V$, we let graph $G_s(V, A, w)$ contain a node v_i for every gene $G^\ell[i]$ in genome G^ℓ. Arc set $A = A_1 \cup A_2$, where A_1 contains an arc (u, v) of weight $w(u, v) := \sum_{d \in \mathcal{D}((u,\, v))} y_d^*$ for every pair of vertices $(u, v) \in V \times V$ with $\mathcal{D}(u,v) \neq \emptyset$. A_2 contains an arc (s, v) of weight

$w(s, v) := z^*_v + \sum_{k=1}^{m} x^*_{\{v, v^{\bar{r}}_k\}}$ for every $v \in V$ with $v \neq s$. Then, for every $S \subset V$ with $s \in V - S$, the sum on the left-hand side of inequality (10) equals the weight of the cut $(V \setminus S, S)$ in G_s:

$$\sum_{\substack{(u,v)\in A_1 \cup A_2:\\ u\in V\setminus S,\, v\in S}} w(a) = \sum_{\substack{(u,v)\in A_1:\\ u\in V\setminus S,\, v\in S}} w(u,v) + \sum_{(s,v)\in A_2 : v\in S} w(s,v)$$

$$= \sum_{\substack{(u,v)\in A_1:\\ u\in V\setminus S,\, v\in S}} \sum_{d\in\mathcal{D}((u,v))} y^*_d + \sum_{v\in S}\left(z^*_v + \sum_{k=1}^{m} x^*_{\{v, v^{\bar{r}}_k\}}\right)$$

$$= \sum_{d\in\mathcal{D}(V\setminus S, S)} \alpha(d,S)\cdot y^*_d + \sum_{v\in S} z^*_v + \sum_{v\in S}\sum_{k=1}^{m} x^*_{\{v, v^{\bar{r}}_k\}}.$$

The last step follows directly from the definition of $\alpha(d, S)$ [see (9)]. Determining set S^* that minimizes the left-hand side of inequality (10) is thus equivalent to computing the minimum $s - t$ cut in G_s over all $s \in V$. This can be reduced to $2n - 2$ maximum flow problems, that is, from an arbitrary node s to all $t \neq s$ and from all $t \neq s$ to s, each taking time $\mathcal{O}(n^2\sqrt{|A|})$ using Goldberg-Tarjan's preflow push-relabel algorithm. ■

We show next how to separate a certain relaxation of the lifted duplication cycle constraints efficiently. The separation algorithm exploits the following characterization of this class of constraints:

Proposition 1. *An inequality of the form* (8) *with* $C = \bigcup_{i=1}^{t} C^i$, $C \subseteq D^j$, *is a lifted duplication cycle inequality if and only if there exists a sequence of non-empty intervals* $[a_1, b_1], [a_2, b_2], [a_t, b_t]$ *such that for* $i = 1, \ldots, t$, *it holds*

(P1) $\bigcap_{d\in C^i} target(d) = [a_{i+1}, b_{i+1}]$
(P2) $\forall d \in C^i : origin(d) \cap [a_i, b_i] \neq \emptyset$
(P3) $\forall d \in D^j \setminus C : target(d) \cap [a_{i+1}, b_{i+1}] \neq \emptyset \rightarrow$

$$origin(d) \cap [a_i, b_i] = \emptyset \vee \exists d' \in C^{i+1} : target(d) \cap origin(d') = \emptyset$$

where $[a_{t+1}, b_{t+1}] := [a_1, b_1]$ *and* $C^{i+1} := C^1$.

Intuitively, property (P1) captures condition (C1), property (P2) ensures that (C2) is satisfied, and (P3) implies maximality. Notice that condition (P3) is not equivalent to requiring C to be maximal with respect to (P1) and (P2), since a duplication satisfying (P3) might intersect interval $[a_{i+1}, b_{i+1}]$ only partially. A formal proof of Proposition 1 follows.

Proof. To prove sufficiency, assume set C has the claimed structure (P1)–(P3). For $i = 1, \ldots, t$, any two duplications in C^i contain at least one common vertex in their target violating constraint (4) and are thus incompatible. Furthermore, any set of duplications $\{d_{l_1}, \ldots, d_{l_t}\}$ with $d_{l_i} \in C^i$, $i = 1, \ldots, t$ forms a cycle according to Definition 4 since, due to properties (P1) and (P2), $origin(d_{i+1}) \cap target(d_i) \neq \emptyset$. Finally, assume C is not maximal with respect to (C1) and (C2). Consider a duplication $d \notin C$ such that $C \cup \{d\}$ satisfies (C1) and (C2). In particular, there exists $1 \leq r \leq i$ such that d is incompatible with all duplications in C^i and thus, $target(d) \cap [a_{i+1}, b_{i+1}] \neq \emptyset$. From condition (P3), it follows that either $origin(d) \cap [a_i, b_i] = \emptyset$, in which case d does not lie on a common cycle with any duplication from C^{i-1}, or $\exists d' \in C^{i+1} : target(d) \cap origin(d') = \emptyset$, which implies that there exists a duplication $d' \in C^{i+1}$ such that d and d' do not lie on a common cycle, violating in both cases condition (C2).

To prove necessity, we show that every set C that satisfies (C1), (C2), and (C3) exhibits the claimed structure (P1)–(P3). For $i = 1, \ldots, t$, let $[a_{i+1}, b_{i+1}] := \bigcap_{d\in C^i} target(d)$ where $[a_{t+1}, b_{t+1}] := [a_1, b_1]$. Due to condition (C1), $[a_i, b_i] \neq \emptyset$ and thus, property (P1) is satisfied. By the definition of cycles, (C2) implies that the origin of every duplication in C^i intersects the target of every duplication in C^{i-1}, $i = 1, \ldots, t$, where $C^0 := C^t$. For intervals $target(d')$ with non-empty intersection, $d' \in C^{i-1}$, this is equivalent to $\bigcap_{d'\in C^{i-1}} target(d') \cap origin(d) \neq \emptyset$, for all $d \in C^i$, which satisfies (P2). Finally, assume (P3) does not hold, that is, there exists a duplication $d \in D \setminus C$ with (i) $target(d) \cap [a_{i+1}, b_{i+1}] \neq \emptyset$, (ii) $origin(d) \cap [a_i, b_i] \neq \emptyset$, and (iii) $\forall d' \in C^{i+1} : target(d) \cap origin(d') \neq \emptyset$. Due to constraint (4), (i) causes d to be incompatible with all duplications in C^i. Properties (i) and (ii) imply, by the definition of cycles (see

Definition 4), that for every cycle C that contains an arbitrary duplication $d' \in C^i$, replacing d' by d results in a cycle C'. Therefore, $C \cup \{d\}$ satisfies (C1) and (C2), which is in contradiction to (C3). ■

The high-level idea of the separation algorithm is to construct a graph whose nodes represent elements that satisfy (P1) and whose edges connect intervals that satisfy (P2). Similar to the separation of lifted mixed cycles in the multiple sequence alignment problem (Althaus et al., 2005), a potential violation of a slightly relaxed variant of a lifted duplication cycle can be detected by a shortest path computation.

Theorem 5. *For $\ell \in \{1, 2\}$, let $D := D^{\ell}$, $V := V^{\ell}$, $n := |V|$ and $m = |V^{\bar{\ell}}|$, where $\bar{\ell}$ is the complement of ℓ in $\{1, 2\}$. For a given point $(x^*, y^*, z^*) \in \mathbb{R}_+^{|E|+|D|+|V|}$, it can be determined in time $\mathcal{O}(n^3 + |D|n^2)$ whether the relaxation of a lifted duplication cycles* (8)*, in which for every interval $[a_i, b_i]$ in Proposition 1 $a_i = b_i$, $i = 1, \ldots, t$, is violated.*

Proof. We construct an arc-weighted graph $G = (V, A, w)$ in the following way. Similar to the alignment graph, we have one node for every gene in the given genome. With a slight abuse of notation, we compute, for every pair of nodes v_i and v_j, the set of duplications $\mathcal{D}(i, j)$ whose origin contain v_i and whose target contain v_j, that is, $\mathcal{D}(i, j) := \{d \in D : i \in \mathit{origin}(d) \wedge j \in \mathit{target}(d)\}$. For every non-empty set $\mathcal{D}(i, j)$, we add an arc from node v_i to node v_j. We define the weight of an arc as

$$w((v_i, v_j)) := 1 - \sum_{d \in \mathcal{D}(i,j)} y_d^*.$$

The violation of a lifted duplication cycle having the claimed structure, given by the sequence of nodes $v_{i_1}, v_{i_2}, \ldots, v_{i_t}$, with $v_{i_j} \in [a_j, b_j]$, is

$$\sum_{j=1}^{t} \sum_{d \in C^j} y_d^* - t + 1 = 1 - \sum_{j=1}^{t} \left(1 - \sum_{d \in C^j} y_d^*\right) = 1 - \sum_{j=1}^{t} w((v_{i_j}, v_{i_{j+1}})),$$

where $v_{it+1} := v_{i1}$. Note that sets $\mathcal{D}(i_j, i_{j+1})$, $j = 1, \ldots, t$, satisfy (P1)–(P3) and thus the last equality follows. The most violated lifted duplication cycle of the relaxed kind can therefore be obtained by computing the shortest arc-weighted path in G from every node v to itself (if it exists).

Implemented naïvely, the weight of the arcs in A can be determined in $\mathcal{O}(|D|n^2)$. Note that due to constraint (4), the arc weights are all non-negative and we can compute the shortest paths by Dijkstra's algorithm. Since graph G has $\mathcal{O}(n^2)$ arcs and Dijkstra's algorithm is called n times, the shortest cycle in G can be found in time $\mathcal{O}(n^3)$. ■

4. MEDIAN OF THREE SPECIES

The alignment model introduced in Holloway et al. (2012) is restricted to cherries of a species tree, that is, to subtrees with two leaves. In this section, we formulate a generalization of the alignment model that captures the *median-of-three problem*, which is of key importance in the reconstruction of ancestral genomes in a phylogenetic tree. The *steinerization* method (Blanchette et al., 1997) starts with an initial assignment of genomes to ancestral nodes. In each iteration, until convergence, it traverses the tree and improves for every internal node the ancestral genome assigned to it by computing the median of its three neighbors, its immediate ancestor, and its two descendants. Convergence to the global optimum is not guaranteed. In contrast, applying the method proposed by Holloway et al. to pairs of nodes in a phylogenetic tree can only be used as the initialization step in this context, as convergence would be reached after the first iteration.

Definition 5. (Median of Three Problem). Let x be a node in the phylogenetic tree whose parent is u and whose two children are v and w. Then, the median genome $G(x)$ minimizes

$$c(G(u) \to G(x)) + c(G(x) \to G(v)) + c(G(x) \to G(w)).$$

For ease of notation, we refer to $G(u)$, $G(v)$, $G(w)$, and $G(x)$ as G^1, G^2, G^3, and G^m, respectively. Figure 4 depicts the phylogenetic scenario with (unknown) median genome G^m, its immediate ancestor G^1, and the

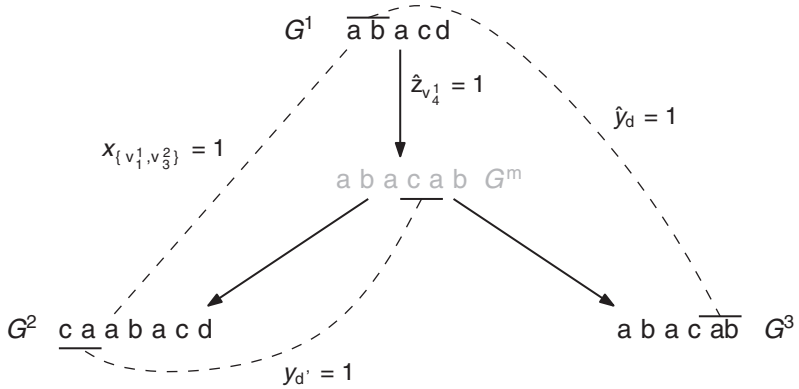

FIG. 4. The figure illustrates the four different types of variables used to express the evolutionary history along two generations. Leftmost gene a in G^1 is retained in G^m, G^2, and G^3 [scenario (i)]. Genes ab are duplicated from G^1 to G^m, modeled by duplication d with origin in G^1 and target in G^3 [Scenario (ii)]. The rightmost gene a in G^m is lost subsequently in G^2. Genes ca are duplicated from G^m to G^2 [Scenario (iii)]. Gene d is lost from G^1 to G^m.

two descendants G^2 and G^3. Compared to the Two-Species Small Phylogeny Problem introduced in Section 2, an alignment of G^1, G^2, and G^3 has to express the evolutionary history transforming G^1 into G^2 and G^3 along *two* generations. Compared to the classical sequence alignment problem, the alignment of three sequences in the duplication-loss scenario is not equivalent to the cycle-free combination of all pairwise alignments. It requires additional variables and constraints that ensure the consistency of duplication events and the fulfillment of the visible history condition. The restriction of the median problem to the inference of the evolutionary history from G^m to G^2 and G^3 represents an instance to the 2-SPP and can be modeled analogously using alignment variables $x_{\{v_i^2, v_j^3\}}$, loss variables $z_{v_j^i}$, $i \in \{2, 3\}$, and duplication variables $d \in D^2 \cup D^3$. Since the alignment of the three genomes G^1, G^2, and G^3 also has to express the evolutionary history that transforms G^1 into G^m, we introduce alignment variables $x_{\{v_i^1, v_j^2\}}$ and $x_{\{v_i^1, v_j^3\}}$, loss variables $\hat{z}_{v_j^1}$, and duplication variables $\hat{y}_d$, $d \in D^m$ (Fig. 4). A loss variable $\hat{z}_{v_j^1}$ denotes the loss of gene $G^1[j]$ from G^1 to G^m. Since G^m is unknown, we model a duplication $d \in D^m$ from G^1 to G^m as being composed of an origin in G^1 and a target in G^2 or G^3 (see duplication d in Fig. 4). Genes in G^2 and G^3 that are aligned to genes in the origin of a duplication $d \in D^m$ are paralogous to the genes in the target of d. It is worth noting that this model allows us to relax the visible history condition (see Definition 2) in that we also consider duplications $d \in D^m$ whose origin is subsequently modified arbitrarily, including its complete loss (relocation). The reason is that the origin of such a duplication is fully preserved in G^1, even if it is modified subsequently. The strict visible history condition as formulated in Definition 2 can be enforced through additional constraints (Supplementary Material D).

4.1. Additional constraints

Concerning the pairwise alignment of genomes G^2 and G^3, constraints (2)–(5) remain valid. Additionally, we have to model the initial occurrence of each gene in G^2 and G^3. A gene i in G^2 (G^3) either

(i) has been present as gene j in G^1 already,
(ii) arose from a duplication event $d \in D^m$ from G^1 to G^m, or
(iii) results from a duplication event $d' \in D^2$ ($d' \in D^3$) from G^m to G^2 (G^3).

Scenario (*i*) is captured by $x_{\{v_j^1, v_i^2\}} = 1$ ($x_{\{v_j^1, v_i^3\}} = 1$), scenario (*ii*) by $\hat{y}_d = 1$, and scenario (*iii*) by $y_{d'} = 1$.

To avoid inconsistencies between duplications d_1, $d_2 \in D^m$ that represent the same duplication from G^1 to G^m but whose target might have been modified from G^m to G^2 or G^3, we allow only one of the two variables $\hat{y}_{d_1}$, $\hat{y}_{d_2}$ to be active. Therefore, technically case (*ii*) must be captured by either $\hat{y}_{d_1} = 1$ or by i corresponding to a gene j in G^3 (G^2), that is, $x_{\{v_i^2, v_j^3\}} = 1$, and gene j lying in the target of a duplication $d_2 \in D^m$ with $\hat{y}_{d_2} = 1$. For example, in Figure 4, the duplication of genes ab from G^1 to G^m is subsequently modified from G^m to G^2 by a loss of gene a. Therefore, the rightmost gene b in G^2 is aligned to the rightmost gene b in G^3 (not shown), that is, $x_{v_7^2, v_6^3} = 1$. Let such a relationship be indicated by variable

$\xi_{(v_i^2, v_j^3)}$. Since exactly one of the three scenarios applies, we require for each gene i in G^2 (analog for genes in G^3),

$$\sum_{j=1}^{|G^1|} x_{\{v_j^1, v_i^2\}} + \sum_{\substack{d \in D^m \\ v_i^2 \in target(d)}} \hat{y}_d + \sum_{\substack{d \in D^2 \\ v_i^2 \in target(d)}} y_d + \sum_{j=1}^{|G^3|} \xi_{(v_i^2, v_j^3)} = 1. \tag{11}$$

Variable $\xi_{(v_i^2, v_j^3)}$ corresponds to the product of $x_{\{v_i^2, v_j^3\}}$ and $\sum_{d \in D^m, j \in target(d)} \hat{y}_d$, which can be expressed by linear constraints in a straightforward way (Nemhauser and Wolsey, 1988). Similarly, each gene in G^1 must have been lost from G^1 to G^m or appear (aligned) in at least one of G^2 or G^3. Therefore, for every gene i in G^1, we introduce constraints

$$\hat{z}_{v_i^1} + \sum_{j=1}^{|G^2|} x_{\{v_i^1, v_j^2\}} + \sum_{j=1}^{|G^3|} x_{\{v_i^1, v_j^3\}} \geq 1 \tag{12}$$

$$\hat{z}_{v_i^1} + \sum_{j=1}^{|G^2|} x_{\{v_i^1, v_j^2\}} \leq 1, \quad \hat{z}_{v_i^1} + \sum_{j=1}^{|G^3|} x_{\{v_i^1, v_j^3\}} \leq 1.$$

Note that this model does not allow a gene from G^1 to be lost both from G^m to G^2 and from G^m to G^3. However, for a median minimizing (5), it is always preferable (i.e., has smaller cost) to not contain such a gene. That is, it is better to imply a single loss of the gene from G^1 to G^m as opposed to two losses from G^m to G^2 and G^3.

Furthermore, we have to ensure that alignments between different pairs of genomes in the multiple alignment are consistent. That is, we have to generalize constraint (2) to sets of incompatible alignment edges containing edges between all three pairs of genomes. In particular, cycles being composed of both alignment edges and arcs representing the order of genes on the genomes are introduced in the multiple sequence alignment context as *mixed cycles* (Reinert, 1999). Mixed cycles represent a contradictory ordering of the genes in the alignment. We introduce *lifted mixed cycle* inequalities into our model that are shown in Althaus et al. (2005) to dominate the mixed cycle inequalities and can be separated in time $\mathcal{O}(n^3)$, where n is the total number of genes in the three genomes.

Finally, we have to ensure consistency of duplications $d \in D^m$ and alignment edges between G^1 and G^3 (G^2 analog). A duplication $d \in D^m$ with target region $[k, \ldots, l]$ in G^2 and an alignment edge $x_{\{v_i^1, v_j^3\}}$ are not consistent if $G^3[j]$ lies between two genes $G^3[j']$ and $G^3[j'']$ that are aligned to the interval $[k, \ldots, l]$ in G^2 (Fig. 5). This scenario would imply that duplication $d \in D^m$ from G^1 to G^m has a nonconsecutive target region as it encloses a gene already present in G^1. We exclude such solutions by adding, for every pair of duplication $d \in D^m$ and alignment edge $x_{\{v_i^1, v_j^3\}}$, the following constraints:

$$\sum_{n=1}^{j-1} x_{\{v_s^2, v_n^3\}} + \sum_{n=j+1}^{|G^3|} x_{\{v_t^2, v_n^3\}} + \hat{y}_d + x_{\{v_i^1, v_j^3\}} \leq 3 \qquad \forall k \leq s < t \leq l$$

A limitation of our model is that it does not capture the subsequent modification of the target of a duplication $d \in D^m$ from G^m to *both* G^2 or G^3. Since, in either G^2 or G^3, the target of a duplication in D^m

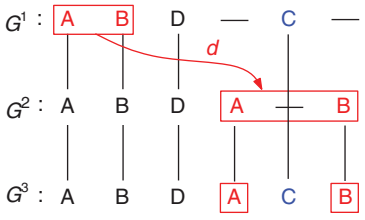

FIG. 5. This figure illustrates a pair of inconsistent duplication $d \in D^m$ and alignment edge $\{v_4^1, v_5^3\}$. The presented solution corresponds to the median genome G^m = ABDACB, which implies a nonconsecutive duplication of AB enclosing the conserved gene C. Transitive alignment edges between G^1 and G^3 not shown.

has to be covered by the corresponding $\hat{y}$-variable [inequality (11)], and since we only consider duplications with identical origin and target, modifications (duplications or losses) are valid only in the branch where the genes in the target are covered indirectly by the ξ-variables. Notice that this limitation does not cause our model to be infeasible. However, the cost of our solution in this case might not be optimal.

5. EXPERIMENTAL RESULTS

In the following two sections, we evaluate our branch-and-cut approach for the TWO-SPECIES SMALL PHYLOGENY problem as well as its extension to the MEDIAN OF THREE problem.

5.1. Two-species alignment

In this section, we compare our preliminary implementation of the branch-and-cut approach outlined in Section 3.3, referred to as DUPLOCUT, to the iterative ILP method by Holloway et al. (2012). The implementation of the latter method was provided by the authors of Holloway et al. We implemented DUPLOCUT in C++ and used CPLEX version 12.4 as ILP solver for both tools. We ran both methods single threaded, except for the *Vibrionaceae* dataset in which we allowed (only) the implementation by Holloway et al. to use up to 10 threads. We used the same scoring scheme as Holloway et al., which assigns alignments of homologous genes a cost of 0, a loss of length x a cost of x (i.e., $k = 1$), and every duplication event is charged with cost 1. For benchmarking, we used two sources of data, real-world data and simulated data.

Real-world instances We compared the two approaches on two sets of real-world instances that were also used in Holloway et al. (2012). The sets contain the stable tRNA and rRNA contents of 12 *Bacillus* and 6 *Vibrionaceae* lineages that were preprocessed as discussed in Holloway et al. (2012). They are linearized according to their origin of replication and inverted segments are manually reinverted. The average number of stable tRNA and rRNA genes is $\approx$120 for the bacillus and $\approx$140 for the vibrio genomes. For both sets we ran both algorithms for all pairs of genomes leading to 66 pairs for *Bacillus* and 15 pairs for *Vibrionaceae*. The average run time of the tool by Holloway et al. on the *Bacillus* instances was around 67 seconds, while DUPLOCUT took less than 1.5 seconds.

The superiority of our approach however becomes apparent when comparing the two methods on the *Vibrionaceae* pairs. On several instances, the method by Holloway et al. ran for a couple of hours whereas DUPLOCUT is able to find the optimal solution within seconds. For two instances that were not solved within 2 weeks of computation by the iterative ILP method, we found the optimal ancestral genome in 33 min and 49 min, respectively. On average, neglecting instances that could not be solved by Holloway et al., we achieve an improvement of a factor of $\sim$200. On none of the instances did DUPLOCUT require more than 1 hour.

Simulated instances The simulation of artificial instances follows the strategy of Holloway et al. (2012) and tries to capture characteristics of the real-world instances. The simulation is performed in the following steps. First, a random sequence R of length n and alphabet size α is simulated where the alphabet symbols at each position are independent and identically distributed. In the second step, l moves (single gene loss or duplication event) are applied to R where the length of a duplication follows a Gaussian distribution with mean 5 and standard deviation 2, and the start position of every move is uniformly distributed. This sequence is then used as the ancestor genome X, and two extant genomes are generated by applying l moves to X for each of them. In Table 1, we present running-time results for several settings of parameters n, l, and α where we simulated 50 instances for each setting. For the selected parameter settings, the ratios l/n and α/n are based on the values observed in *Bacillus* data ($l/n \approx 0.1$, $\alpha/n \approx 0.5$) (Holloway et al., 2012) except for the second setting where we doubled the number of moves. For the remaining three parameter settings, we examined the impact of different genome lengths between 100 and 400 genes. The results in Table 1 show that running time increases with increasing genome length and increasing distance between the two genomes. Nevertheless, even for 400 genes, the average running time of DUPLOCUT is still within seconds while the implementation of Holloway et al. requires almost 1 hour.

The enormous improvement in running time is a prerequisite to a generalization of the pairwise case to the median of three-species setting. In the next section, we show that the repeated reoptimization of medians of three neighbors leads to an improvement in the prediction.

Table 1. Runtime Comparison of DupLoCut and the Implementation by Holloway et al. (2012) for 2-SPP on Simulated Data (See Text for Description of Settings)

Setting (n,l,α)	*Average run time in seconds*	
	DupLoCut	*Holloway et al.*
(100,10,50)	0.4	22.0
(100,20,50)	1.7	491.2
(200,20,100)	1.4	239.7
(400,40,200)	22.0	3112.8

5.2. Median of three

In this section, we demonstrate that our median-of-three model can improve the quality of predicted ancestral genomes in a phylogenetic tree. All results in this section refer to the relaxed variant of the visible history condition (see Section 4). Imposing the stricter requirement formulated in Definition 2 yielded very similar results (not shown).

We generated five balanced phylogenetic trees, each with 128 extant species. For each internal node, the two descendant genomes were generated as in the pairwise alignment benchmark. We generate a random genome of length $n = 100$ and alphabet size $\alpha = 50$ at the root node. All other genomes along the tree were generated by applying five random moves (single gene loss or duplication event) to their direct ancestor.

Following the *steinerization* method, we first initialized the genomes of all internal nodes bottom up by solving the two-species alignment problem for their two immediate descendants. This initial genome assignment corresponds to the solution output by the SPP-heuristic introduced in Holloway et al. (2012). After initialization, we applied four rounds of reoptimization and replaced the genome of every internal node by the median of its three immediate neighbors. The root genome was re-estimated by computing the pairwise alignment of its two children. We traversed the tree bottom-up, and up to 16 independent nodes (number of CPU threads) at the same depth were processed in parallel. The running time for each tree was between 60 and 180 minutes on an 8-core Intel Xeon machine using 16 threads.

Deviating from the standard procedure, we update a genome even if it yields a cost increase because of the overestimation of the true cost of the median (see Section 1.1). In our experiments we obtained more parsimony solutions and a higher prediction quality using this strategy. A possible explanation for this behavior is that by this process the risk of getting stuck in a local minimum decreases. The total cost of the complete tree after each round with respect to the cost after initialization is shown in Figure 6. For all five trees the total cost was reduced by $\approx 25\%$ after four rounds of reoptimization.

To analyze whether the cost reduction also implies a more accurate reconstruction of ancestral genomes, we compared all predicted genomes to the true genomes after each round. The number of duplication and loss events determined by the pairwise alignment serves as a measure of distance. The average distance of all ancestral genomes after each round is depicted in Figure 7. Note that the total distance at round *init* is with respect to the (final) solution of the SPP-heuristic based on the pairwise model of Holloway et al. (2012). In all cases, the average distance of the predicted genomes to the true genomes could be reduced by more than 60%.

In Supplementary Figure S3, we show the same data grouped by node depth. The highest improvement is achieved for nodes that are closer to the extant genomes and decreases toward the root. Nevertheless, even for the direct descendants of the root the average distance to the true genomes decreased by $\approx 50\%$.

We also applied our method to real-world data. We took the National Center for Biotechnology Information (NCBI) taxonomy of 12 *Bacillus* strains of four species (amyloliquefaciens, subtilis, thuringiensis, and cereus) and extracted stable RNA genes. The ancestral genomes predicted after initialization implied a total of 151 duplication and loss events. After four rounds of reoptimization, our method converged to a prediction of ancestral genomes requiring only 121 duplication and loss events. Although we cannot guarantee that the predicted genomes are really closer to the true sequences, the results from the simulation benchmark encourage this assumption. A plot summarizing the results and the phylogenetic tree together with the strains can be found in Supplementary Material A.

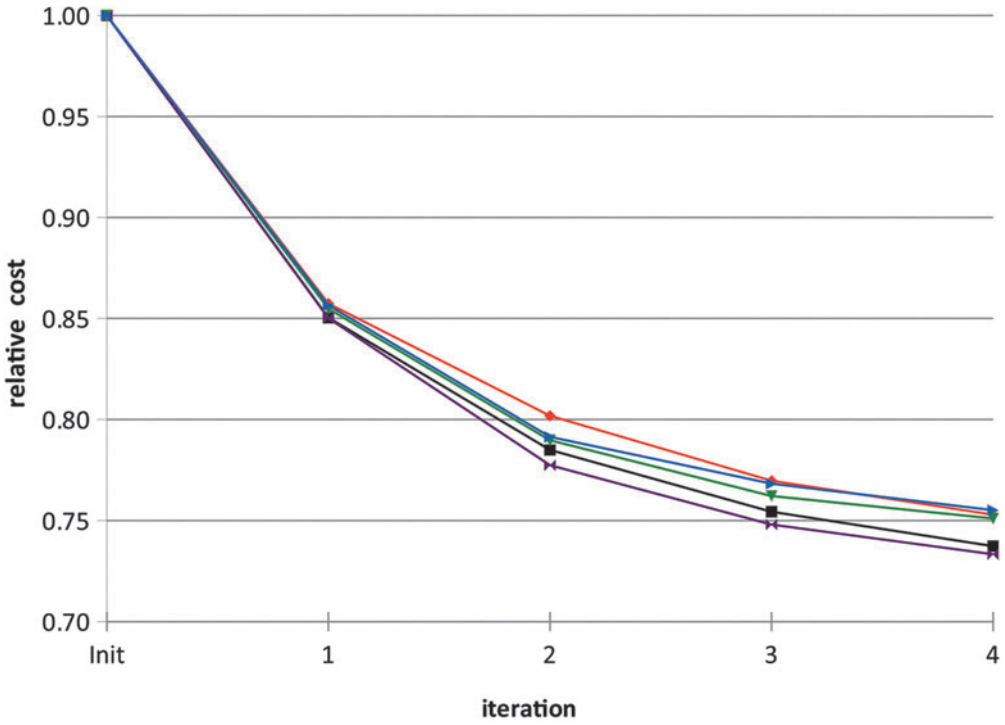

FIG. 6. Total cost (number of duplication/loss operations) for five simulated phylogenetic trees after one to four steps of reoptimization normalized by the total cost after initialization.

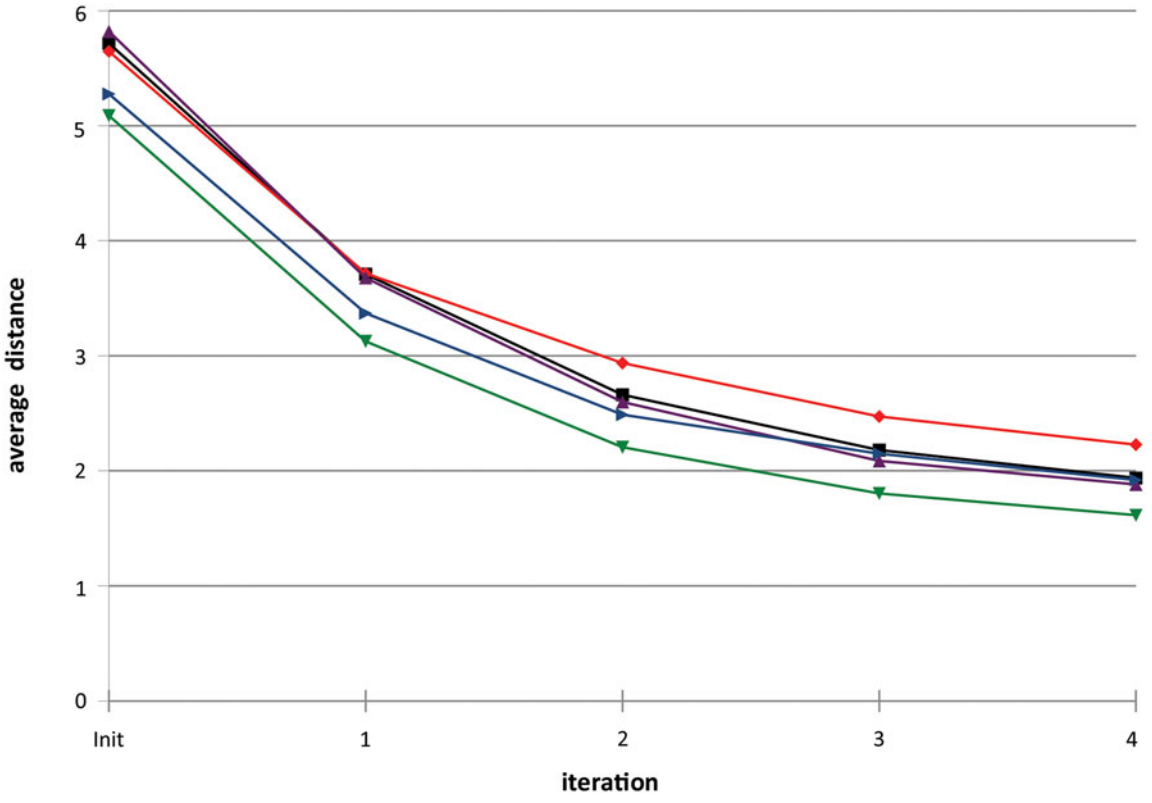

FIG. 7. Average distance of all inferred genomes from the true sequences after one to four steps of reoptimization with the median of three algorithms for five simulated phylogenetic trees.

6. CONCLUSION

The results from the runtime comparison show that our novel branch-and-cut algorithm is superior to the previously proposed ILP-based approach. For larger instances (bigger n) or pairs of rather distant genomes like in the *Vibrionaceae* dataset, the improvement is even more dramatic. Therefore, our branch-and-cut algorithm allows us to solve instances that are more complex than the pairs of *Bacillus* genomes on a desktop PC and does not require compute clusters to find the optimal ancestral genome in a reasonable amount of time. Even more important than the mere improvement in runtime is that this efficiency allowed us to extend the model to three species. This generalized model is of great importance for the widely used *steinerization* method. For both simulated instances comprising 128 leaves and a set of *Bacillus* genomes, we observed, after a few steps of reoptimization, a significant cost reduction for the complete tree compared to the pairwise initialization proposed by Holloway et al. (2012). In our simulations, this reduction in cost goes hand in hand with an improved prediction of ancestral genomes. The model we have proposed readily captures inverted duplications. Although nonoverlapping inversions can easily be incorporated into our formulation by additional variables, they might invalidate our separation procedures. Our software DuplocCut is freely available online.

ACKNOWLEDGMENTS

We thank the authors of Holloway et al. (2012) for providing us with the genome datasets and their software. This work is supported in part by the National Institutes of Health under grant R01 HG006677.

AUTHOR DISCLOSURE STATEMENT

The authors declare that no competing financial interests exist.

REFERENCES

Althaus, E., Caprara, A., Lenhof, H.-P., and Reinert, K., 2005. A branch-and-cut algorithm for multiple sequence alignment. *Mathematical Programming* 105, 387–425.

Blanchette, M., Bourque, G., and Sankoff, D., 1997. Breakpoint phylogenies. In *Genome Informatics*, 25–34. Univ. Academy Press.

Dondi, R. and El-Mabrouk, N., 2012. On the complexity of minimum labeling alignment of two genomes. *CoRR* abs/1206.1877.

Hahn, M.W., Han, M.V., and Han, S.G., 2007. Gene Family Evolution across 12 Drosophila Genomes. *PLoS Genetics* 3, e197+.

Han, M.V. and Hahn, M.W., 2009. Identifying parent-daughter relationships among duplicated genes. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing* 114–125.

Han, M. V. and Hahn, M. W., 2012. Inferring the history of interchromosomal gene transposition in drosophila using n-dimensional parsimony. *Genetics* 190, 813–25.

Holloway, P., Swenson, K., Ardell, D.H., and El-Mabrouk, N., 2012. Evolution of genome organization by duplication and loss: An alignment approach. In Chor, B., ed., *RECOMB*, *Lecture Notes in Computer Science*, volume 7262, 94–112. Springer.

Kovác, J., Brejová, B., and Vinar, T., 2011. A practical algorithm for ancestral rearrangement reconstruction. In Przytycka, T.M. and Sagot, M.-F., eds., *WABI*, *Lecture Notes in Computer Science*, volume 6833, 163–174. Springer.

Lynch, M. and Conery, J. S., 2000. The evolutionary fate and consequences of duplicate genes. *Science* 290, 1151–1155.

Nemhauser, G.L. and Wolsey, L.A., 1988. *Integer and combinatorial optimization*. Wiley-Interscience, New York, NY, USA.

Ohno, S., 1970. *Evolution by gene duplication*. Springer-Verlag, Berlin, New York,.

Padberg, M.W., 1973. On the Facial Structure of Set Packing Polyhedra. *Mathematical Programming* 5, 199–215.

Reinert, K., 1999. *A polyhedral approach to sequence alignment problems*. Ph.D. thesis, Saarlndische Universitts- und Landesbibliothek, Postfach 151141, 66041 Saarbrcken.

Reinert, K., Lenhof, H.-P., Mutzel, P., Mehlhorn, K., and Kececioglu, J. D., 1997. A branch-and-cut algorithm for multiple sequence alignment. In *Proceedings of the 1st Annual international conference on Research in Computational Molecular Biology*, RECOMB'97, 241–249. ACM Press.

Rogers, H.H., Bergman, C.M., and Griffiths-Jones, S., 2010. The evolution of tRNA genes in Drosophila. *Genome biology and evolution* 2, 467–77.

Sankoff, D. and Blanchette, M., 1997. The median problem for breakpoints in comparative genomics. In Jiang, T. and Lee, D.T., eds., *COCOON*, *Lecture Notes in Computer Science*, volume 1276, 251–264. Springer.

Swenson, K.M., Marron, M., Earnest-Deyoung, J.V., and Moret, B.M.E., 2008. Approximating the true evolutionary distance between two genomes. *Journal of Experimental Algorithmics* 12, 1.

Withers, M., Wernisch, L., and dos Reis, M., 2006. Archaeology and evolution of transfer RNA genes in the Escherichia coli genome. *RNA (New York, N.Y.)* 12, 933–42.

Zhang, J., 2003. Evolution by gene duplication: an update. *Trends in Ecology & Evolution* 18, 292–298.

Address correspondence to:
Dr. Stefan Canzar
McKusick-Nathans Institute of Genetic Medicine
Johns Hopkins University School of Medicine
773 N. Broadway, MRB 423
Baltimore, MD 21205

E-mail: canzar@jhu.edu