Polynomial-Time Algorithms for Phylogenetic Inference Problems Involving Duplication and Reticulation

Van Iersel, Leo; Janssen, Remie; Jones, Mark; Murakami, Yukihiro; Zeh, Norbert

DOI
10.1109/TCBB.2019.2934957

Publication date
2020

Document Version
Final published version

Published in
IEEE/ACM Transactions on Computational Biology and Bioinformatics

Citation (APA)
Van Iersel, L., Janssen, R., Jones, M., Murakami, Y., & Zeh, N. (2020). Polynomial-Time Algorithms for Phylogenetic Inference Problems Involving Duplication and Reticulation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *17*(1), 14-26. [8798653]. https://doi.org/10.1109/TCBB.2019.2934957

Polynomial-Time Algorithms for Phylogenetic Inference Problems Involving Duplication and Reticulation

Leo van Iersel, Remie Janssen, Mark Jones, Yukihiro Murakami, and Norbert Zeh

Abstract—A common problem in phylogenetics is to try to infer a species phylogeny from gene trees. We consider different variants of this problem. The first variant, called Unrestricted Minimal Episodes Inference, aims at inferring a species tree based on a model with speciation and duplication where duplications are clustered in duplication episodes. The goal is to minimize the number of such episodes. The second variant, Parental Hybridization, aims at inferring a species *network* based on a model with speciation and reticulation. The goal is to minimize the number of reticulation events. It is a variant of the well-studied Hybridization Number problem with a more generous view on which gene trees are consistent with a given species network. We show that these seemingly different problems are in fact closely related and can, surprisingly, both be solved in polynomial time, using a structure we call “beaded trees”. However, we also show that methods based on these problems have to be used with care because the optimal species phylogenies always have a restricted form. To mitigate this problem, we introduce a new variant of Unrestricted Minimal Episodes Inference that minimizes the duplication episode depth. We prove that this new variant of the problem can also be solved in polynomial time.

Index Terms—Phylogenetics, duplication, Minimum Episodes problem, Hybridization Number problem, gene trees, inference, polynomial-time algorithm

1 Introduction

Phylogenetic trees are commonly used to represent the evolutionary history of a set of taxa. The leaves represent extant taxa; internal nodes represent speciation events that caused lineages to diverge. If we assume that the only process is speciation and that no incomplete lineage sorting occurs, then any gene will have a gene tree that is consistent with the species phylogeny. There are, however, evolutionary processes beyond vertical inheritance of genetic material and speciation events that make it more challenging to reconstruct the real evolutionary history. Examples of such processes are hybridization, horizontal gene transfer, and duplication. Each of these processes can result in discordance between gene trees.

This leads to a number of problems in which the task is to minimize the number of such complicating events. In *reconciliation problems*, we are given the gene trees together with the species phylogeny, and the task is to find optimal embeddings of the gene trees into the species phylogeny. Such methods are for example used to estimate dates of duplications, to discover relationships between duplicate genes [1], [2], and to reconstruct the infection history of parasites [3]. In *inference problems*, only the gene trees are given and we aim to find a species phylogeny that minimizes the discordance with the gene trees. Such problems are relevant when the species phylogeny is not yet known with certainty.

1.1 Duplication Minimization Problems

Gene duplications happen as a consequence of errors in the DNA replication process. This leads to a species having multiple copies of the same gene. There exist many types of gene duplication, which depend on the positions of errors within the replication process [4], [5]. The scale of gene duplications is determined by the number of genes that get duplicated. An extreme example of a large-scale duplication is *Whole Genome Duplication (WGD)*, in which every gene in the genome is duplicated. This process, also known as polyploidization, occurs as a result of an error in separation of chromosomes during gamete production. It is most common in plants (see, e.g., Fig. 1 in [11]) but has also occurred in animals [6], and there are two WGD events even in the evolutionary history leading to humans [7], [8]. Large-scale duplications provide species with diversification potential, giving them the ability to quickly adapt to a changing environment [6], [9], [10].

In their seminal paper [12], Goodman et al. pioneered the parsimony approach to reconciling gene trees with species trees. This has motivated researchers to explore reconciliation through different models whilst optimizing some measure of the number of duplication events.

- *L. van Iersel, R. Janssen, M. Jones, and Y. Murakami are with the Delft Institute of Applied Mathematics, Delft University of Technology, Van Mourik Broekmanweg 6, Delft 2628 XE, the Netherlands.*
  E-mail: {L.J.J.vanIersel, R.Janssen-2, M.E.L.Jones, Y.Murakami}@tudelft.nl.
- *N. Zeh is with the Faculty of Computer Science, Dalhousie University, 6050 University Ave, Halifax, NS B3H 1W5, Canada.*
  E-mail: nzeh@cs.dal.ca.

Manuscript received 26 Sept. 2018; revised 5 July 2019; accepted 9 Aug. 2019. Date of publication 14 Aug. 2019; date of current version 4 Feb. 2020.
(Corresponding author: Yukihiro Murakami.)
Digital Object Identifier no. 10.1109/TCBB.2019.2934957

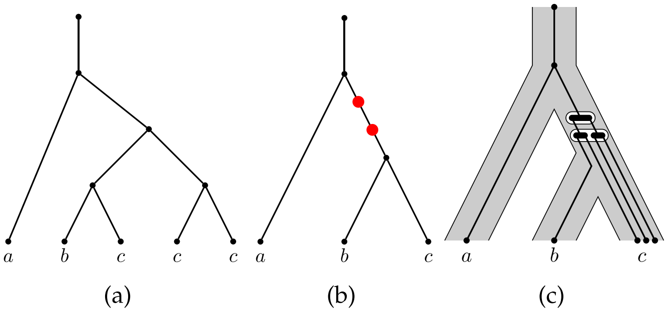


Fig. 1. (a) A MUL-tree T on $X = \{a, b, c\}$. (b) A duplication tree D that is consistent with T. (c) An illustration showing how T can be drawn inside D. This shows how two or more incoming branches may duplicate simultaneously at a duplication node (according to Minimal Episodes clustering).

The studied problems can be categorized according to how duplication events are clustered to form duplication episodes and which restrictions are put on the possible locations of duplications [13]. We focus on *minimal episodes (ME)* clustering where duplications can be clustered if they occur on the same branch of the species phylogeny and have no ancestor-descendant relationship in any gene tree. We believe this way of clustering to be most relevant since it can cluster duplications that can be part of a single (large-scale) duplication event. We consider *unrestricted ME* (called the FHS-model in [13]), which does not put any restrictions on the locations of gene duplications.

Reconciliation problems have been studied intensively, especially without clustering of duplication events [3], [14], [15]. Several reconciliation problems with clustering of duplication events have been proven to be computationally intractable [16], [17], whereas for others there are polynomial-time [18], [19] or even linear-time [13], [20], [21] algorithms. For unrestricted ME reconciliation, which was recently shown to be NP-hard [23], there only exists an exponential-time algorithm [13].

It has also been attempted to use reconciliation as a basis for inferring species phylogenies. For the unrestricted ME inference problem, [22] used a brute-force approach on all possible species phylogenies. It was observed that unrestricted ME fails to rank the true species tree among the top third of all topologies (for real data with a well accepted species phylogeny). It was suggested that a possible reason for this anomaly is that duplication episodes near the root are overly powerful under this criterion. A similar observation was made in a more recent reconciliation study [13]. However, neither article gives a mathematical explanation for this phenomenon. It should also be noted that, since the number of possible species phylogenies grows extremely quickly with the number of species, brute-force approaches are only feasible for very small data sets.

Inference problems are generally assumed to be computationally intractable. However, NP-hardness has been proven only for some restricted inference problem without clustering of duplication events [17]. For an inference problem with restricted clustering (called gene duplication (GD) clustering in [13]), NP-hardness was suggested in [16] but not proven. Because of the suspected intractability of these problems, some heuristic inference approaches have been attempted using efficient algorithms for reconciliation (see, e.g., [24]).

1.2 Reticulation Minimization Problems

Another possible cause of discordance between gene trees is *reticulate evolution*, such as hybridization or horizontal gene transfer. In such cases, the evolutionary history is represented by a *phylogenetic network* rather than a tree.

Reticulate evolution can occur in nature when genetic material from one species is transmitted to some other species. In asexual species, such transfers are called *horizontal gene transfers (HGT)*. In bacteria, for example, this happens in nature by transformation (take-up from the environment) or conjugation (transmission from another bacterium). In sexual species, a cause for such transmissions can be *hybridization*, where individuals from different but related taxa mate. There is also evidence that horizontal gene transfers occur between multicellular sexual species. An example is the transfer of a phototropin gene from Hornworts to Ferns (see [25], [26]). HGT can even happen between more distant species.

Gene trees that appear to be inconsistent may in fact simply take different paths through the network. This leads to a family of inference problems in which the aim is to find a phylogenetic network that is consistent with the gene trees and has the minimum number of *reticulation events* (nodes in the network with two ancestral branches). A phylogenetic network is often taken to be consistent with a gene tree if that tree is *displayed* by the network, which, roughly speaking, means that the gene tree can be drawn inside the network in such a way that each network branch contains at most one lineage of the gene tree. A more generous definition is to count a network as consistent with a gene tree if the tree is *weakly displayed* by the network [27], [28]. Roughly speaking, this means that different lineages of the gene tree may "travel down" the same branch of the network, as long as any branching node in the tree coincides with a branching node in the network. In this case, the tree is also called a *parental tree* of the network. This models situations where a species has individuals carrying multiple homologous copies of a gene.

The Hybridization Number problem, in which we seek a network with the minimum number of reticulations displaying all input trees, has been well-studied. It has been shown that Hybridization Number is NP-hard already when the input consists of only two gene trees [29]. Furthermore, there are theoretical FPT algorithms for any fixed number of gene trees, but there are no practical algorithms that can handle instances with more than two input trees unless the number of taxa is extremely small [30], [31].

In contrast, the Parental Hybridization problem, in which we seek a network with the minimum number of reticulations that weakly displays each input tree, was introduced only recently [28] and its computational complexity was open prior to this article. Our motivation for studying this problem is threefold:

(i) Since Hybridization Number is NP-hard, it is interesting whether relaxing the notion of a tree displayed by a network leads to an easier problem.
(ii) Since reticulation can lead to multiple homologous copies of a gene in a species, requiring that each gene tree is displayed by the network may lead us to overestimate the number of reticulations.

(iii) The problem of finding an optimal network that weakly displays a set of phylogenies arises as a crucial subproblem in a recent heuristic approach for constructing phylogenetic networks in the presence of hybridization and incomplete lineage sorting [28].

1.3 Structural Assumptions

In this paper, as is common in the literature, we assume that all networks and trees are *binary*, that is, every node except the root and the leaves has total degree exactly 3. Our results should easily generalize to nonbinary trees and networks, but we do not verify this here.

We note that, unlike many papers in this area, we allow a network to contain *parallel arcs*, that is, pairs of arcs that join the same pair of nodes. Parallel arcs are normally omitted because, for most problems, it can either be shown that there exists an optimal solution without parallel arcs or it can be assumed that a realistic solution contains no parallel arcs. For example, any set of gene trees is displayed by an optimal hybridization network without parallel arcs. For the problems studied in this paper, however, an optimal solution may require parallel arcs. Considering this problem with the added restriction that parallel arcs are forbidden may be an interesting mathematical challenge; however, we do not believe it is biologically meaningful.

Explicit reasons to allow parallel arcs in networks are abundant. We give three: First, if one restricts a large network to a subset of the taxa, the natural restriction could have parallel arcs. Second, phylogenetic Markov models for character evolution behave differently if parallel arcs are suppressed. Third, polyploidization events often result from a sort of interspecific or intraspecific hybridization [32]; an intraspecific hybridization is most naturally represented by parallel arcs in the network.

Throughout this paper, we allow input trees to be multilabelled, that is, each species may appear as a label of multiple leaves in a tree. This is natural for the problems we study, as gene duplication and reticulation can both lead to multiple homologous genes appearing in the genome of a single species.

1.4 Our Contributions

We show that both Unrestricted Minimal Episodes Inference and Parental Hybridization reduce to the problem Beaded Tree, which we introduce in this paper. Using this reduction, we show that both problems can be solved in polynomial time by adapting Aho et al.'s classic algorithm for testing gene tree consistency [33]. Thereby, we provide the first polynomial-time algorithm for an inference problem with duplication clustering. Furthermore, we provide the first polynomial-time algorithm for constructing a phylogenetic *network* with a minimum number of reticulations from gene trees.

We also show that optimal solutions to Beaded Tree have a restricted structure and this has corresponding implications for the optimal solutions to Unrestricted Minimal Episodes Inference and Parental Hybridization that our algorithms produce. Moreover, we show that, in fact, *all* optimal solutions to Unrestricted Minimal Episodes Inference have a particular structure. Therefore, this problem should be used with care. For this reason, we introduce a variation of Unrestricted Minimal Episodes Inference, in which the aim is not to minimize the total number of duplication episodes but to minimize instead the maximum number of duplication episodes on any path from the root to a leaf in the output tree. We show that this problem can also be solved in polynomial time via reduction to a variant of Beaded Tree, which we call Beaded Tree Depth.

1.5 Structure of the Paper

In Section 2, we introduce the main definitions, including formal problem definitions. In Section 3, we show that both Unrestricted Minimal Episodes Inference and Parental Hybridization reduce to the problem Beaded Tree. In Section 4, we prove structural properties of optimal solutions to Beaded Tree. In Section 5, we provide a polynomial-time algorithm for Beaded Tree and prove its correctness and running time. In Section 6, we provide a polynomial-time algorithm for Beaded Tree Depth. Finally, in Section 7, we discuss our results and possibilities for further research.

2 Preliminaries and Definitions

We begin by defining *multi-labelled trees*, which form the input for all problems considered in this paper.

Definition 1. *Let X be a set of species. A* multi-labelled tree (MUL-tree) *on X is a directed acyclic graph with one node of in-degree 0 and out-degree 1 (the* root*) and with all other nodes having either in-degree 1 and out-degree 2 (*tree nodes*) or in-degree 1 and out-degree 0 (*leaves*). Each leaf is labelled with an element of X. If each element of X labels at most one leaf, we call the MUL-tree a* tree.

Note that we will often refer to a labelled node by its label; for example, we may say that $x \in X$ is a leaf in a MUL-tree T if one of the leaves of T is labelled with x.

The notation introduced in the next definition is common to all structures considered in this paper, that is, not just to MUL-trees but also to duplication trees, phylogenetic networks, and beaded trees, defined later in this section.

Definition 2. *Given a directed acyclic graph G, let $V(G)$ denote the nodes, and $E(G)$ the edges of G. Let $L(G)$ denote the leaves (i.e., nodes of out-degree 0) of G. We refer to the non-leaf nodes of G as the* internal nodes *of G. Given an edge xy in G, we say that x is a* parent *of y and y is a* child *of x. We say a node x is an* ancestor *of a node y (and y is a* descendant *of x) if there is a path from x to y in G (including if $x = y$). If in addition $x \neq y$, we say x is a* strict *ancestor of y (and y is a* strict *descendant of x). A node x is a* least common ancestor *of two nodes y and z if it is an ancestor of both y and z and no strict descendant of x is an ancestor of both y and z. If G is a tree, then the LCA of any two nodes is unique; otherwise, it may not be unique.*

2.1 Duplication Episodes

The evolutionary history of a set of species, including points at which duplication events occurred, can be modelled by a duplication tree, defined as follows:

Definition 3. *Let X be a set of species. A* duplication tree *on X is a directed acyclic graph D with one node of in-degree 0 and* *out-degree 1 (the* root*), $|X|$ nodes of in-degree 1 and out-degree 0 (*leaves*), and all other nodes having either in-degree 1 and out-degree 2 (*tree nodes*) or in-degree 1 and out-degree 1 (*duplication nodes*). The leaves are bijectively labelled with the elements of X. The* duplication number *of D is the number of duplication nodes it contains.*

We note that, in contrast to MUL-trees, each species in X appears as the label of exactly one leaf in a duplication tree. Informally, a MUL-tree T is *consistent* with a duplication tree D if T can be drawn inside D so that branches of T duplicate only at duplication nodes of D, in the sense that both out-edges of a node of T may follow the same out-edge of the duplication node. We formalize this as follows:

Definition 4. *Given a MUL-tree T on X and a duplication tree D on X, a* duplication mapping *from T to D is a function $M : V(T) \to V(D)$ such that*

- *For each leaf $l \in L(T)$, $M(l)$ is a leaf of D labelled with the same species as l,*
- *For each edge $uv \in E(T)$, $M(u)$ is a strict ancestor of $M(v)$, and*
- *For each internal node u of T with children v, v', either $M(u)$ is the least common ancestor of $M(v)$ and $M(v')$, or $M(u)$ is a duplication node.*

This is illustrated in Fig. 1. We say that D is consistent with *T if there is a duplication mapping from T to D.*

Let S be the species tree derived from D by suppressing duplication nodes. Then a duplication mapping from T to D represents a reconciliation of T with S with Minimal Episodes clustering. Each duplication node in D represents a cluster of duplications, which is called a *duplication episode*. Internal nodes in T are treated as duplications if they are mapped to duplication nodes of D, and as speciations otherwise. Duplications are clustered together if they are mapped to the same duplication node of D. The properties of a duplication tree and duplication mapping ensure that duplications that are clustered occur on the same branch of the species phylogeny and have no ancestor-descendant relationship in a gene tree, as required by Minimal Episodes clustering. We are now ready to define the following problem:

Unrestricted Minimal Episodes Inference

Input. A set $\mathcal{T} = \{T_1, \ldots, T_t\}$ of MUL-trees with label sets $X_1, \ldots, X_t \subseteq X$.

Output. A duplication tree D on X with minimum duplication number such that D is consistent with each tree in $\mathcal{T}$.

For this and other optimization problems, we use the term *solution* to refer to an object that satisfies the requirements specified in the description of the output except that it does not necessarily need to optimize the optimization criterion. An *optimal solution* is a solution that optimizes the optimization criterion. For example, for Unrestricted Minimal Episodes Inference, a solution is a duplication tree on X that is consistent with each tree in $\mathcal{T}$. It is an optimal solution if, in addition, it has minimum duplication number over all such duplication trees.

We note that for any MUL-tree T on X and any duplication tree D on X that has at least $|V(T)|$ duplication nodes as ancestors of every tree node, D is consistent with T. It follows that every instance of Unrestricted Minimal Episodes Inference has a solution (and therefore an optimal solution).

2.2 Parental Hybridization

Phylogenetic networks are an appropriate mathematical model used for describing evolutionary histories that include reticulation events and are central to the problem Parental Hybridization, defined below.

Definition 5. *Let X be a set of species. A* (rooted binary) phylogenetic network *N on X is a directed acyclic multigraph with one node of in-degree 0 and out-degree 1 (the* root*), $|X|$ nodes of in-degree 1 and out-degree 0 (*leaves*), and all other nodes having either in-degree 1 and out-degree 2 (*tree nodes*) or in-degree 2 and out-degree 1 (*reticulation nodes*). The leaves are bijectively labelled with the elements of X. The* reticulation number *of N is the number of reticulation nodes it contains. If N contains no reticulation nodes, then N is a* tree.

We note that the key distinctions between a phylogenetic network and a MUL-tree are that a phylogenetic network may contain reticulation nodes but each label in X may appear only once, whereas a MUL-tree has no reticulations but each label can appear multiple times. Also note that, due to the degree restrictions, there can be at most two edges between any pair of nodes in a phylogenetic network, and there are no loops.

Definition 6. *Given a set X of species, let N be a phylogenetic network, and T a MUL-tree on X. A* weak embedding *of T into N is a function h that maps every node of T to a node of N, and every edge in T to a directed path in N such that*

- *For each leaf $l \in L(T)$, $h(l)$ is a leaf of N labelled with the same species,*
- *For each edge $xy \in E(T)$, the path $h(xy)$ is a path from $h(x)$ to $h(y)$ in N, and*
- *For each internal node x in T with children y, y', the paths $h(xy)$ and $h(xy')$ start with different out-edges of $h(x)$.*

This is illustrated in Fig. 2. We say that N weakly displays *T if there is a weak embedding of T into N.*

We note that N weakly displays T if and only if T is a *parental tree inside N* as defined in [28], hence the name Parental Hybridization. The notion of a tree *weakly displayed* by a network was first introduced in [27], where it was shown that T is weakly displayed by N if and only if there exists a *locally separated reconciliation* from T to N, which is equivalent to our definition of a weak embedding.

We now define the Parental Hybridization problem:

Parental Hybridization

Input. A set $\mathcal{T} = \{T_1, \ldots, T_t\}$ of MUL-trees with label sets $X_1, \ldots, X_t \subseteq X$.

Output. A phylogenetic network N on X with minimum reticulation number and such that N weakly displays all MUL-trees in $\mathcal{T}$.

Even though we do not use it in this paper, it is worth noting the relationship between weak embeddings, weakly displayed trees, and Parental Hybridization on one hand and embeddings, displayed trees, and Hybridization Number on the other hand. An *embedding* of a tree T into a network N is a weak embedding h of T into N with the added condition that the paths $h(e)$ and $h(e')$ are edge-disjoint for every pair of edges $e \neq e' \in T$. (Note that this also implies that the paths are node-disjoint unless e and e' have a node in common.) If such an embedding exists, then N *displays* T.

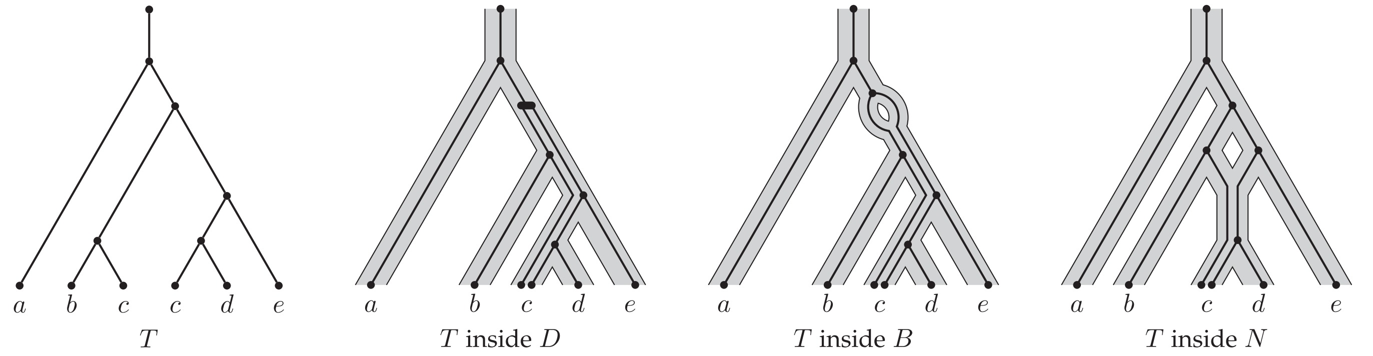


Fig. 2. A MUL-tree T and illustrations of a duplication mapping from T to a duplication tree D, and of weak embeddings of T into a beaded tree B and into a phylogenetic network N that is not a beaded tree.

Similarly to Parental Hybridization, the Hybridization Number problem for a set of phylogenetic trees $\mathcal{T}$ asks for a phylogenetic network N with the minimum reticulation number and such that N displays all trees in $\mathcal{T}$.

2.3 Beaded Trees

The key to solving both Unrestricted Minimal Episodes Inference and Parental Hybridization is the equivalence between these two problems and the following Beaded Tree problem, which we establish in this paper.

Definition 7. *A* bead *in a phylogenetic network N is a pair of nodes (u, v) such that there are two parallel edges from u to v. A* beaded tree *is a phylogenetic network B in which every reticulation node is part of a bead (see Fig. 2).*

The Beaded Tree problem is defined as follows:

Beaded Tree

Input. A set $\mathcal{T} = \{T_1, \dots, T_t\}$ of MUL-trees with label sets $X_1, \dots, X_t \subseteq X$.

Output. A beaded tree B on X with minimum reticulation number that weakly displays all MUL-trees in $\mathcal{T}$.

3 Reductions to Beaded Tree

In this section, we show that the two problems Unrestricted Minimal Episodes Inference and Parental Hybridization are both reducible to Beaded Tree, which will allow us to focus on the latter problem in the rest of the paper. We begin with the proof for Unrestricted Minimal Episodes Inference.

Lemma 8. *Let X be a set of species and $\mathcal{T} = \{T_1, \dots, T_t\}$ a set of MUL-trees on X. For any integer k, there exists a solution to Unrestricted Minimal Episodes Inference on $\mathcal{T}$ with k duplications if and only if there exists a solution to Beaded Tree on $\mathcal{T}$ with k beads.*

Proof. Let the duplication tree D be a solution to Unrestricted Minimal Episodes Inference on $\mathcal{T}$ with k duplications. Then construct a beaded tree B from D as follows: Replace each duplication node d in D with a bead (u_d, v_d). If p is d's parent in D, then u_d's parent in B is p or, if p is itself a duplication node, v_p; v_d's child in B is d's child c in D or, if c is itself a duplication node, u_c.

It is easy to observe that B is a beaded tree with k beads. To see that B is a solution to Beaded Tree on $\mathcal{T}$, consider any tree $T \in \mathcal{T}$ and let M be a duplication mapping from T to D. Then we can construct a weak embedding h from T into B as follows. For each node x in T, if $M(x)$ is a duplication node d, then let $h(x)$ be the tree node u_d (i.e., the top node of the bead (u_d, v_d)). Otherwise, let $h(x) = M(x)$. For any edge xy, the node $h(y)$ is by construction a strict descendant of $h(x)$, so there exists a path from $h(x)$ to $h(y)$ in B. We choose $h(xy)$ to be any such path but ensure that the two paths $h(xy)$ and $h(xy')$ start with different edges in the bead (u_d, v_d) if $M(x)$ is a duplication node d in D and y and y' are x's children in T. This guarantees that the paths $h(xy)$ and $h(xy')$ start with different out-edges of $h(x)$ if $M(x)$ is a duplication node. If $M(x)$ is not a duplication node, then $M(x)$ is the least common ancestor of $M(y)$ and $M(y')$, so the paths $h(xy)$ and $h(xy')$ are edge-disjoint and again start with different out-edges of $h(x)$. Thus, h is a weak embedding of T into B.

Conversely, let the beaded tree B be a solution to Beaded Tree on $\mathcal{T}$ with k beads. Then construct a duplication tree D from B by replacing each bead (u, v) with a duplication node $d_{(u,v)}$. $d_{(u,v)}$'s parent in D is u's parent p in B or, if p is itself part of a bead (x, p), the duplication node $d_{(x,p)}$; $d_{(u,v)}$'s child in D is v's child c in B or, if c is itself part of a bead (c, y), the duplication node $d_{(c,y)}$.

It is easy to observe that D is a duplication tree with k duplications. To see that D is a solution to Unrestricted Minimal Episodes Inference on $\mathcal{T}$, consider any tree $T \in \mathcal{T}$ and let h be a weak embedding of T into B. Then we can construct a duplication mapping from T to D as follows. For any node x in T, if $h(x)$ is not in a bead, then set $M(x) = h(x)$. If $h(x)$ is the top node u of a bead (u, v), then let $M(x) = d_{(u,v)}$. (Note that $h(x)$ cannot be the bottom node of a bead, because x is either a leaf or has out-degree 2.) By the requirements of a weak embedding, $M(x)$ is a strict ancestor of $M(y)$ for any edge xy in T. Furthermore, for any internal node x with children y and y', there are paths from $h(x)$ to $h(y)$ and from $h(x)$ to $h(y')$ that start with different out-edges of $h(x)$. It follows that either $M(x) = h(x)$ is the least common ancestor of $M(y)$ and $M(y')$ or $M(x)$ is a duplication node. □

The next lemma shows that any instance $\mathcal{T}$ of Parental Hybridization has an optimal solution that is a beaded tree, that is, Parental Hybridization can be reduced to Beaded Tree.

Lemma 9. *For any set $\mathcal{T}$ of MUL-trees on X, there exists a phylogenetic network N with k reticulations that weakly displays all MUL-trees in $\mathcal{T}$ if and only if there exists a beaded tree B with k reticulations that weakly displays the MUL-trees in $\mathcal{T}$.*

Proof. The if-direction is trivial because every beaded tree is a phylogenetic network. For the only-if-direction, consider a network N with the maximum number of beads among all solutions of Parental Hybridization on $\mathcal{T}$ with k reticulations. If N is a beaded tree, the lemma holds. Otherwise, there is some reticulation node r in N that has two different parents c_s and d_t. Let q be the unique child of r. Let u be a least common ancestor of c_s and d_t in N, let c_1 and d_1 be the children of u, let $c_1, \dots, c_s$ be the nodes on a path from c_1 to c_s, and let $d_1, \dots, d_t$ be the nodes on a path from d_1 to d_t. Note that, by construction, there is no directed path from d_j to c_i, for any $1 \leq i \leq s$ and $1 \leq j \leq t$.

We obtain a phylogenetic network N' from N as follows: Delete r and any edges incident to it, as well as the edges uc_1 and ud_1. Now add a new node v, a pair of parallel edges from u to v, and edges vc_1, c_sd_1, and d_tq. (Note that this construction assumes that $s, t \geq 1$; if this is not the case, then we can produce N' by introducing a "dummy node" c_1 or d_1 and suppressing it after the construction is complete.)

Observe that (as there is no path from any node d_j to any node c_i in N) N' is still an acyclic graph. It follows that N' is a phylogenetic network, and it is easy to see that N' has the same number of reticulations as N but one more bead than N. We show now that any MUL-tree T weakly displayed by N is also weakly displayed by N', from which it follows that N' is also a solution to Parental Hybridization on $\mathcal{T}$ with k reticulations. Since N' has one more bead than N, this contradicts the choice of N, that is N must be a beaded tree.

Let h be a weak embedding of T into N. Then we define a weak embedding h' of T into N' as follows. Since $h(x) \neq r$ for every node $x \in T$ and $V(N) \setminus V(N') = \{r\}$, we have $h(x) \in V(N')$ for all $x \in T$. Thus, we can define $h'(x) = h(x)$ for all $x \in T$. Next observe that, for any two nodes $u', v' \in V(N) \setminus \{r\}$, there exists a path from u' to v' in N' if there exists such a path in N. Thus, since there exists a path $h(xy)$ from $h(x)$ to $h(y)$ in N, for every edge $xy \in T$, there also exists a path $h'(xy)$ from $h'(x)$ to $h'(y)$ in N' for every edge $xy \in T$. We need to show that we can choose these paths such that, for every node $x \in V(T)$ with children y and y', the paths $h'(xy)$ and $h'(xy')$ begin with different out-edges of $h'(x)$.

So consider a node x and its two children y and y' in T. If no out-edges of $h'(x)$ were deleted in the construction of N', then the children of $h'(x)$ are the same in N' as in N, and these children are still ancestors of $h'(y)$ and $h'(y')$. Thus, the required paths exist. Now assume that at least one out-edge of $h'(x)$ was deleted, from which it follows that $h'(x) \in \{u, c_s, d_t\}$. If $h'(x) = u$, then there are two paths from $h'(x)$ to $h'(y)$ and from $h'(x)$ to $h'(y')$ that use different out-edges of $h'(x)$, as each path can use a different parallel edge from u to v. If $h'(x) = c_s$, then one of $\{h'(y), h'(y')\}$ is a descendant of r (and therefore a descendant of q), and the other is a descendant of the other child of c_s. Therefore, in N', one of $\{h'(y), h'(y')\}$ is descended from q, and the other is descended from the child of c_s that is not d_1. Thus, the required paths still exist. A similar argument applies when $h'(x) = d_t$. This finishes the proof. □

4 Structural Properties of Optimal Beaded Trees

In this section, we prove some of the properties of an optimal solution to an instance of Beaded Tree. These properties will both be used in Section 5 as a basis for our algorithm for finding an optimal beaded tree for any given instance and highlight that in fact *every* optimal solution to an instance of Beaded Tree has a very restrictive structure.

Definition 10. *Given a phylogenetic network N on X and a subset $S \subseteq X$, let $N \setminus S$ denote the network derived from N by deleting every leaf in S, and then exhaustively deleting unlabelled nodes of out-degree 0 and suppressing nodes of in-degree 1 and out-degree 1. Let $N|_S$ denote the network $N \setminus (X \setminus S)$.*

For a set of MUL-trees $\mathcal{T}$, let $F_1(\mathcal{T})$ denote the set of trees derived by, roughly speaking, deleting the topmost tree node from every tree. We make this notion more precise in the following definition.

Definition 11. *Given a MUL-tree T with more than one leaf, let r denote the root, x the child of r and y_l and y_r the children of x. Let T_l be derived from T by deleting y_r and all its descendants, and suppressing x. Similarly let T_r be derived from T by deleting y_l and all its descendants, and suppressing x. Then we call $\{T_l, T_r\}$ the* depth-1 forest *of T, denoted $F_1(T)$. For a set of MUL-trees $\mathcal{T}$, we define*

$$F_1(\mathcal{T}) = \bigcup_{T \in \mathcal{T}} F_1(T).$$

In what follows, we say that a beaded tree B has a *bead at the root* if the child u of the root node is part of a bead (u, v).

Lemma 12. *Given an instance $\mathcal{T}$ of Beaded Tree, there exists a solution B with a bead at the root and reticulation number k if and only if $F_1(\mathcal{T})$ has a solution B' with reticulation number $k-1$.*

Proof. Suppose first that $F_1(\mathcal{T})$ has a solution B' with reticulation number $k-1$. Let r be the root of B' and a its child. Construct a beaded tree B from B' by deleting the edge ra, adding a new bead (u, v), and adding edges ru and va. By construction, B is a beaded tree with k beads, and it has a bead at the root.

To see that B is a solution for $\mathcal{T}$, consider any tree T in $\mathcal{T}$, and let $\{T_l, T_r\} = F_1(T)$. Let r_T be the root of T, x its child, and y_l and y_r the children of x, with $y_l \in V(T_l)$ and $y_r \in V(T_r)$. Since B' is a solution for $F_1(\mathcal{T})$, there exist weak embeddings h_l and h_r of T_l and T_r, respectively, into B'. Construct a weak embedding h of T into B as follows: Let $h(r_T) = r$, $h(x) = u$, and for all other nodes $x' \in V(T)$, $h(x') = h_l(x')$ if $x' \in V(T_l)$, and $h(x') = h_r(x')$ if $x' \in V(T_r)$. Let $h(r_Tx)$ be the path from r to u, $h(x'y') = h_l(x'y')$ if $x'y' \in E(T_l)$, and $h(x'y') = h_r(x'y')$ if $x'y' \in E(T_r)$. Finally, let $h(xy_l)$ be a path from u to $h(y_l)$, and $h(xy_r)$ a path from u to $h(y_r)$, letting those two paths start with different out-edges of u. It is easy to see that h is a weak embedding of T into B, so B is a solution for $\mathcal{T}$.

Conversely, suppose that $\mathcal{T}$ has a solution B with a bead (u, v) at the root and reticulation number k. Let r be the root of B and z the child of v. Let B' be the network derived from B by deleting u and v and adding an edge

rz. By construction, B' is a beaded tree with reticulation number $k-1$.

To see that B' is a solution for $F_1(\mathcal{T})$, consider any tree T in $\mathcal{T}$, and let $\{T_l, T_r\} = F_1(T)$. Let r_T be the root of T, x its child, and y_l and y_r the children of x, with $y_l \in V(T_l)$ and $y_r \in V(T_r)$. Since B is a solution for $\mathcal{T}$, there exists a weak embedding h of T into B. Observe that $h(x')$ must be a strict descendant of v for any strict descendant x' of x (indeed, u is the earliest node that x could be mapped to and any strict descendant of x must be mapped to a tree node strictly descended from this point). So we can define a weak embedding h_l of T_l into B' by letting $h_l(r_T) = r$ and $h_l(x') = h(x')$ for every node $x' \neq r_T \in T_l$, letting $h_l(r_T y_l)$ be a path in B' from r to $h_l(y_l)$, and letting $h_l(e) = h(e)$ for any other edge $e \in T_l$. By a similar method, we can define a weak embedding h_r of T_r into B'. Thus, B' is a solution for $F_1(\mathcal{T})$, as required. □

In the same way that Lemma 9 establishes that Parental Hybridization always has an optimal solution that is a beaded tree, the following lemma shows that there always exists an optimal solution to Beaded Tree of an even more restrictive structure.

Lemma 13. *Every instance $\mathcal{T}$ of Beaded Tree has an optimal solution B such that all reticulations are on the same path.*

Proof. Consider an optimal solution B for $\mathcal{T}$. For each reticulation node $z \in B$, let $\lambda_B(z)$ be the number of reticulation nodes strictly descended from z. Let $\lambda(B)$ be the sum of $\lambda_B(z)$ over all reticulation nodes z in B. Choose B such that $\lambda(B)$ is maximized. Since all optimal solutions for $\mathcal{T}$ have the same number b of beads and $\lambda(B') \leq \binom{b-1}{2}$ for any beaded tree B' with b beads, an optimal solution B for $\mathcal{T}$ that maximizes $\lambda(B)$ exists.

If all reticulations in B are on the same path, the lemma holds. So assume that not all reticulations are on the same path. Then there is some tree node b in B that is not in a bead and such that both children of b are ancestors of a bead. Let (u_L, v_L) be an earliest bead descended from one child of b, and (u_R, v_R) an earliest bead descended from the other child of b. If u_L is not a child of b, then let $c_1, \dots c_l$ be the nodes on the path from b to u_L. Similarly, if u_R is not a child of b, then let $d_1, \dots d_r$ be the nodes on the path from b to u_R. Note that $c_1, \dots c_l$ and $d_1, \dots d_r$ are all tree nodes. Finally let w_L be the single child of v_L, and w_R the single child of v_R.

Construct a new beaded tree B' from B as follows: Delete the nodes u_L, v_L, u_R, v_R and any edges incident to them, as well as the edges bc_1 and bd_1. Now add new nodes q, u, v, w and add a pair of parallel arcs from b to q and from u to v, as well as arcs $qc_1, c_l d_1, d_r u, vw, ww_L$, and ww_R. (Note that this construction assumes that $l, r \geq 1$; if this is not the case, then we may produce B' by introducing "dummy nodes" c_1 and d_1 and suppressing them after the construction is complete.) Observe that this construction ensures that every node $u' \in V' = V(B) \setminus \{u_L, v_L, u_R, v_R\}$ is an ancestor of a node $v' \in V'$ in B' if this is the case in B, that every node $v' \in V'$ that is a descendant of u_L or u_R in B is a descendant of u in B', and that every node $v' \in V'$ that is an ancestor of u_L or u_R in B is an ancestor of u in B'.

To show that any MUL-tree weakly displayed by B is also weakly displayed by B', let T be a MUL-tree weakly displayed by B and let h be a weak embedding of T into B. We define a weak embedding h' of T into B' as follows: For any node $x \in V(T)$, let

$$h'(x) = \begin{cases} u & \text{if } h(x) \in \{u_L, u_R\} \\ h(x) & \text{otherwise} \end{cases}.$$

Note that this ensures that $h'(x) \in V' \cup \{u\}$ because $h(x)$ is a tree node for all $x \in V(T)$, that is, $h(x) \notin \{v_L, v_R\}$. This definition of h' ensures that there exists a path $h'(xy)$ from $h'(x)$ to $h'(y)$ for every edge xy of T. Indeed, there exists a path $h(xy)$ from $h(x)$ to $h(y)$ in B because h is a weak embedding of T into B. If $h(x), h(y) \in V'$, then $h'(x) = h(x)$, $h'(y) = h(y)$, and we observed above that every descendant of $h'(x)$ in B that belongs to V' is also a descendant of $h'(x)$ in B', that is, there exists a path $h'(xy)$ from $h'(x)$ to $h'(y)$. If $h(x) \in \{u_L, u_R\}$, then $h(y) \in V'$, $h'(x) = u$, and $h'(y) = h(y)$. As observed above, every descendant of u_L or u_R in B that belongs to V' is a descendant of u in B'. Thus, there exists a path $h'(xy)$ from $h'(x)$ to $h'(y)$ also in this case. Finally, if $h(y) \in \{u_L, u_R\}$, then $h(x) \in V'$, $h'(x) = h(x)$, and $h'(y) = u$. As observed above, every ancestor of u_L or u_R in B that belongs to V' is an ancestor of u in B'. Thus, there exists a path $h'(xy)$ from $h'(x)$ to $h'(y)$ once again. It remains to show that these paths can be chosen so that the two paths $h'(xy)$ and $h'(xy')$ corresponding to the edges xy and xy' between a node $x \in V(T)$ and its two children y and y' in T begin with different out-edges of $h'(x)$.

So consider any node x of T and its two children y and y'. If $h'(x)$ is the top node of a bead, then the two paths $h'(xy)$ and $h'(xy')$ can be chosen to start with different edges of this bead. If $h'(x)$ is not the top node of a bead, then $h'(x) = h(x) \in V'$ and $h(x)$ is not the top node of a bead in B either. Since h is a weak embedding of T into B, $h(y)$ is a descendant of one child z of $h(x)$ and $h(y')$ is a descendant of the other child z' of $h(x)$. Moreover, one of these two children, say z, is also a child of $h'(x)$ in B'. As observed above, since $h(y)$ is a descendant of z in B, $h'(y)$ is also a descendant of z in B', so we can choose the path $h'(xy)$ to start with the edge $h'(x)z$. If z' is a child of $h'(x)$ in B', then, by an analogous argument, we can choose the path $h'(xy')$ to start with the edge $h'(x)z'$, so the two paths $h'(xy)$ and $h'(xy')$ start with different out-edges of $h'(x)$. If z' is not a child of $h'(x)$, then $h(x) \in \{c_l, d_r\}$, $z' \in \{u_L, u_R\}$, and z is the child of $h(x)$ not on the path from $h(x)$ to u_L or u_R. In this case, u is a descendant of $h'(x)$ and $h'(y')$ is a descendant of u. Thus, we can choose $h'(xy')$ to be the concatenation of two paths from $h'(x)$ to u and from u to $h'(y')$. Since z does not belong to this path, the two paths $h'(xy)$ and $h'(xy')$ once again start with different edges.

Since we have just shown that any MUL-tree weakly displayed by B is also weakly displayed by B', B' is a solution for $\mathcal{T}$. Moreover, B' has the same number of beads as B and, since $\lambda_{B'}(v) = \lambda_B(v_L) + \lambda_B(v_R)$, $\lambda_{B'}(q) = \lambda_B(v_L) + \lambda_B(v_R) + 1$, and $\lambda_{B'}(z) = \lambda_B(z)$ for any reticulation node $z \in V'$, $\lambda(B') > \lambda(B)$. This contradicts the choice of B, so B has all its beads on a single path. □

In what follows, we use $\mathcal{T}|_S$ and $\mathcal{T} \setminus S$ to denote the sets $\{T_1|_S, \dots, T_t|_S\}$ and $\{T_1 \setminus S, \dots, T_t \setminus S\}$, respectively, for any set of trees $\mathcal{T} = \{T_1, \dots, T_t\}$ and any label set S. If any tree in $\mathcal{T}|_S$ or $\mathcal{T} \setminus S$ is empty, it is removed from the set.

The following definitions and lemmas describe the structure of an optimal solution for $\mathcal{T}$ in terms of optimal solutions for $\mathcal{T}|_S$ and $\mathcal{T} \setminus S$. These structural results will make it easy to design an algorithm for Beaded Tree.

Definition 14. *Given a set of MUL-trees $\mathcal{T} = \{T_1, \dots, T_t\}$, with each MUL-tree T_i having label set $X_i \subseteq X$, the* split partition*$\mathcal{S} = \{S_1, \dots, S_s\}$ of $\{T_1, \dots, T_t\}$ is the partition of X into minimal sets such that any two labels of the same MUL-tree in $F_1(\mathcal{T})$ belong to the same set in $\mathcal{S}$.*

Definition 15. *Given two phylogenetic networks N_1 on X_1 and N_2 on X_2 with $X_1 \cap X_2 = \emptyset$, the process of* joining *N_1 with N_2 consists of identifying the root r_1 of N_1 and the root r_2 of N_2 into a single node u and making u the child of a new root node r.*

Observation 16. If N is obtained by joining N_1 and N_2, then any MUL-tree weakly displayed by N_1 or N_2 is also weakly displayed by N.

The following lemma immediately suggests a strategy for constructing an optimal beaded tree for a collection $\mathcal{T}$ of MUL-trees.

Lemma 17. *Given an instance $\mathcal{T}$ of Beaded Tree, if $|X| = 1$ and $\max_{1 \le i \le t} |L(T_i)| = 1$, then the optimal solution is the tree with a single leaf on X. Otherwise, let $\mathcal{S} = \{S_1, \dots, S_s\}$ be the split partition of $\mathcal{T}$. If for some S_i, there exists a tree T weakly displaying the MUL-trees in $\mathcal{T}|_{S_i}$, then there exists an optimal solution B that is obtained by joining T with an optimal solution to $\mathcal{T} \setminus S_i$. If no such tree T exists, there exists an optimal solution B with a bead (u, v) at the root and such that v is the root of an optimal solution for $F_1(\mathcal{T})$.*

Proof. If $|X| = 1$ and $\max_{1 \le i \le t} |L(T_i)| = 1$, then the optimal solution clearly is the tree with a single leaf on X. So suppose that $|X| > 1$ and assume first that there exists a set $S_i \in \mathcal{S}$ such that the MUL-trees in $\mathcal{T}|_{S_i}$ are weakly displayed by some tree T. If some tree T' weakly displays the MUL-trees in $\mathcal{T}$, then $s \ge 2$ (since any tree in $F_1(\mathcal{T})$ has its leaf set contained within the leaf set of one of the trees in $F_1(T')$ and we can assume w.l.o.g. that not all trees in $F_1(\mathcal{T})$ are displayed by the same tree in $F_1(T')$). In particular, $S_i \ne X$. If no such tree T' exists, then $S_i \ne X$ because T weakly displays all MUL-trees in $\mathcal{T}|_{S_i}$. Since $S_i \ne X$ in both cases, it follows that $X \setminus S_i \ne \emptyset$. Now consider any optimal solution B' for $\mathcal{T}$. Observe that $B' \setminus S_i$ weakly displays all MUL-trees in $\mathcal{T} \setminus S_i$. Moreover, $B' \setminus S_i$ has reticulation number at most that of B'.

Construct a network B by joining $B' \setminus S_i$ with T. Any MUL-tree $T_j \in \mathcal{T}$ with no leaves in S_i is weakly displayed by $B' \setminus S_i$ and therefore by B. Similarly, if every leaf of T_j is in S_i, then T_j is weakly displayed by T and therefore by B. So suppose T_j has leaves in both S_i and $X \setminus S_i$. Since $F_1(T_j)$ consists of two MUL-trees and $\mathcal{S}$ is a split partition of $\mathcal{T}$, we must have $F_1(T_j) = \{T_j|_{S_i}, T_j \setminus S_i\}$. Since T weakly displays $T_j|_{S_i}$ and $B' \setminus S_i$ weakly displays $T_j \setminus S_i$, it follows that B weakly displays T_j. This shows that B displays all MUL-trees in $\mathcal{T}$. Since B has reticulation number at most that of B', B is therefore an optimal solution for $\mathcal{T}$.

It remains to observe that $B' \setminus S_i$ is an optimal solution to $\mathcal{T} \setminus S_i$, as otherwise we could obtain a solution for $\mathcal{T}$ that is better than B by joining T with an optimal solution for $\mathcal{T} \setminus S_i$. Thus, the lemma holds for the case when there exists a tree T weakly displaying all MUL-trees in $\mathcal{T}|_{S_i}$ for some $S_i \in \mathcal{S}$.

Now suppose that there is no tree weakly displaying the MUL-trees in $\mathcal{T}|_{S_i}$ for any $S_i \in \mathcal{S}$. By Lemma 13, there exists an optimal solution B with all reticulations on one path. Suppose that B does not have a bead at the root. Then the child a of the root is a tree node which is the root of two otherwise disjoint beaded trees, and at least one of these beaded trees is a tree T (without beads). Let S be the leaves of this tree T. Since we can assume that at least one MUL-tree in $\mathcal{T}$ has a leaf in S, there exists a set $S_i \in \mathcal{S}$ such that $S_i \cap S \ne \emptyset$. Any such set S_i must be a subset of S because otherwise there exists a MUL-tree $T' \in F_1(\mathcal{T})$ that has leaves in both S and $X \setminus S$; since a is a tree node that is not part of a bead, T' would have to be weakly displayed by either T or $B \setminus S$, which is impossible.

So consider such a set $S_i \subseteq S$ in $\mathcal{S}$. $B|_{S_i}$ weakly displays the MUL-trees in $\mathcal{T}|_{S_i}$ and is a tree because $B|_{S_i} = T|_{S_i}$. Since we assumed that no tree displaying all MUL-trees in $\mathcal{T}|_{S_i}$ exists, B must in fact have a bead at the root, as claimed. By Lemma 12, we also have that the bottom part of the bead is the root of a solution B' for $F_1(\mathcal{T})$ with reticulation number $k-1$, where k is the reticulation number of B. Moreover, B' must be an optimal solution for $F_1(\mathcal{T})$ because otherwise we could obtain a solution for $\mathcal{T}$ that is better than B by adding a bead at the root of an optimal solution for $F_1(\mathcal{T})$. This proves the lemma for the case when there is no tree T displaying all MUL-trees in $\mathcal{T}|_{S_i}$ for any $S_i \in \mathcal{S}$. □

The next two lemmas show that not only does there exist an optimal solution to Beaded Tree with all reticulations on one path, but in fact *any* optimal solution must be quite close to such a structure.

Lemma 18. *Given two beads in any optimal solution to an instance $\mathcal{T}$ of Beaded Tree such that neither bead is a descendant of the other, at least one of these beads has no beads strictly descended from it.*

Proof. The proof is similar to the proof of Lemma 13.

Consider an optimal solution B and suppose for the sake of contradiction that the claim does not hold for B. Then let (p_L, q_L), (p_R, q_R), (u_L, v_L) and (u_R, v_R) be four distinct beads such that (p_L, q_L) is not an ancestor of (p_R, q_R) and (p_R, q_R) is not an ancestor of (p_L, q_L) , but (p_L, q_L) is an ancestor of (u_L, v_L) and (p_R, q_R) is an ancestor of (u_R, v_R). Moreover, assume that (p_L, q_L), (p_R, q_R), (u_L, v_L) and (u_R, v_R) are the earliest such beads, that is, the condition is not satisfied if we replace any one of these beads with one of its strict ancestors. This implies that there are no beads on the path between q_L and u_L, on the path between q_R and u_R or on the path from the least common ancestor of p_L and p_R to either p_L or p_R.

Let x be the least common ancestor of p_L and p_R. If p_L is not a child of x, then let $a_1, \dots a_s$ be the nodes on the path from x to p_L. Similarly, if p_R is not a child of x, then let $b_1, \dots b_t$ be the nodes on the path from x to p_R. If u_L is not a child of q_L, then let $c_1, \dots c_l$ be the nodes on the path from q_L to u_L. Similarly, if u_R is not a child of q_R, then let $d_1, \dots d_r$ be the nodes on the path from q_R to u_R. Note that $a_1, \dots a_s, b_1, \dots b_t, c_1, \dots c_l, d_1, \dots d_r$ are all tree nodes. Finally let w_L be the single child of v_L and w_R the single child of v_R.

Construct B' from B as follows: Delete the nodes p_L, q_L, u_L, v_L, p_R, q_R, u_R, and v_R and any edges incident to them, as well as the edges xa_1 and xb_1. Now add new nodes y, p, q, u, v, and w, and add a pair of parallel arcs from x to y, from p to q, and from u to v, as well as arcs ya_1, a_sb_1, b_tp, qc_1, c_ld_1, d_ru, vw, ww_L, and ww_R (Note that this construction assumes that $s, t, l, r \geq 1$; if this is not the case, then we can produce B' by introducing "dummy nodes" a_1, b_1, c_1, and d_1 and suppressing them after the construction is complete.)

We now show that any MUL-tree weakly displayed by B is also weakly displayed by B'. Let T be a MUL-tree weakly displayed by B, and let h be a weak embedding of T into B. Then we define a weak embedding h' of T into B' as follows: For any node $z \in V(T)$, we set

$$h'(z) = \begin{cases} p & \text{if } h(z) \in \{p_L, p_R\} \\ u & \text{if } h(z) \in \{u_L, u_R\} \\ h(z) & \text{otherwise} \end{cases}.$$

Note that $h(z) \notin \{q_L, q_R, v_L, v_R\}$ because $h(z)$ is a tree node for all $z \in V(T)$. Observe that, if there is a path from u' to v' in B, for any two nodes $u', v' \in V(B) \setminus \{p_L, q_L, u_L, v_L, p_R, q_R, u_R, v_R\}$, then there is a path from u' to v' in B'. Moreover, if there is a path in B from $h(x')$ to $h(y')$, for any two nodes $x', y' \in V(T)$, then there is a path $h'(x'y')$ in B' from $h'(x')$ to $h'(y')$. It remains to verify that these paths can be chosen such that, for any tree node $x' \in V(T)$ with children y' and z', the two paths $h'(x'y')$ and $h'(x'z')$ start with different out-edges of $h'(x')$.

So consider any tree node $x' \in V(T)$ and its two children y' and z'. Since h is a weak embedding, $h(x')$ is a tree node and, by construction, so is $h'(x')$. If $h'(x')$ is the top node of a bead, then $h'(x'y')$ and $h'(x'z')$ can be chosen to start with different parallel arcs of this bead. So assume $h'(x')$ is not the top of a bead in B'. Then, by construction, $h(x')$ is not the top part of a bead in B and $h(y')$ and $h(z')$ are descendend from different children of $h(x')$ in B. If no out-arcs of $h(x')$ were deleted in the construction of B', then the children of $h'(x')$ are the same as the children of $h(x')$, and these children are still ancestors of $h'(y')$ and $h'(z')$. Thus paths $h'(x'y')$ and $h'(x'z')$ can still be chosen to start with different out-edges of $h'(x')$. The final case is when $h'(x')$ is not the top of a bead and at least one out-arc of $h(x')$ was deleted in the construction of B'. In this case, $h'(x') \in \{a_s, b_t, c_l, d_r\}$. It is easy to check that in each of these cases, $h'(y')$ and $h'(z')$ are still descendants of different children of $h'(x')$.

This completes the proof that any MUL-tree weakly displayed by B is also weakly displayed by B'. Moreover, B' has fewer beads than B (as we replaced the four beads $(p_L, q_L), (p_R, q_R), (u_L, v_L), (u_R, v_R)$ with the three beads $(x, y), (p, q), (u, v)$), contradicting the optimality of B. □

Using Lemmas 13 and 18, we can show the following lemma. Intuitively speaking, it says that any optimal solution to Beaded Tree must have "almost all reticulations on one path", in the sense that most reticulations exist on a single path, and any branch coming off of this path leads to at most one reticulation.

Lemma 19. *Given any optimal solution B to an instance $\mathcal{T}$ of Beaded Tree, there exists a path from the root to a leaf of B such that any node not on this path has at most one strict descendant that is a reticulation.*

Proof. Suppose for the sake of contradiction that the claim does not hold, that is, for any path P in B, there exists a node u not in P that has at least two reticulations among its strict descendants. In particular, this implies that there exist two nodes a, b such that a is not an ancestor of b, b is not an ancestor of a, and each of a and b is a strict ancestor of at least two reticulations. Let B_a be the part of B descended from a and let B_b be the part of B descended from b. By Lemma 13, there exist beaded trees B'_a and B'_b such that B'_a weakly displays every MUL-tree weakly displayed by B_a, B'_b weakly displays every MUL-tree weakly displayed by B_b, B'_a has no more reticulations than B_a, B'_b has no more reticulations than B_b, and both B'_a and B'_b have all their reticulations on a single path. By replacing B_a and B_b with B'_a and B'_b, respectively, in B, we obtain a beaded tree B' that weakly displays all MUL-trees in $\mathcal{T}$ and has no more reticulations than B. If B'_a or B'_b has only one bead, then B' has fewer reticulations than B, contradicting B''s optimality. Thus, B'_a has a bead (p_a, q_a) that is an ancestor of another bead (u_a, v_a) in B'_a and B'_b has a bead (p_b, q_b) that is an ancestor of another bead (u_b, v_b) in B'_b. Since neither (p_a, q_a) nor (p_b, q_b) is an ancestor of the other, B' cannot be an optimal solution for $\mathcal{T}$, by Lemma 18. Thus, since B' has no more reticulations than B, B is not an optimal solution for $\mathcal{T}$ either, a contradiction. □

Algorithm 1. Algorithm Beaded-Tree($\mathcal{T}$)

```
Input: A set of MUL-trees $\mathcal{T} = \{T_1, \dots, T_t\}$
Output: A beaded tree $B$ with the minimum number of
reticulations that weakly displays the MUL-trees in $\mathcal{T}$
if $|X| = 1$ and $\max_{1 \leq i \leq t} |L(T_i)| = 1$ then
  return a tree with 1 leaf on $X$;
else
  Calculate the split partition $\{S_1, \dots, S_s\}$ of $\mathcal{T}$;
  for $i \leftarrow 1$ to $s$ do
    $T \leftarrow$ Supertree $(\mathcal{T}|_{S_i})$;
    if $T \neq$ None then
      $B' \leftarrow$ Beaded-Tree $(\mathcal{T} \setminus S_i)$;
      Construct $B$ by joining $B'$ and $T$;
      return $B$;
    end
  end
  $B' \leftarrow$ Beaded-Tree $(F_1(\mathcal{T}))$;
  Construct $B$ by adding a bead whose child is the root of $B'$;
  return $B$;
end
```

5 Beaded Tree Algorithm

In what follows, we let Supertree denote an algorithm that takes as input a set of MUL-trees $\mathcal{T}$, and returns either a tree T weakly displaying all MUL-trees in $\mathcal{T}$ or the value NONE if no such tree exists. The algorithm of [33] achieves this in $O(|X|n)$ time, where $n = \sum_{i=1}^{t} |T_i|$ and $|T_i|$ is the total number of nodes in T_i. We note that the algorithm of [33] is designed only for MUL-trees with at most one copy of each label, for the simple reason that there is no tree weakly displaying a MUL-tree with multiple copies of some label. Fortunately, the fix for this is straightforward: we just let SUPERTREE return NONE whenever $\mathcal{T}$ contains a MUL-tree with two or more copies of some label. By the following lemma, an optimal solution for any instance of BEADED TREE can be found in polynomial time using Algorithm 1.

Lemma 20. *Let $\mathcal{T} = \{T_1, \ldots, T_t\}$ be an instance of BEADED TREE, let $n = \sum_{i=1}^{t} |T_i|$, and let k be the reticulation number of an optimal solution for $\mathcal{T}$. Algorithm 1 finds an optimal solution for $\mathcal{T}$ in $O((|X|^2 + |X|k)n)$ time.*

Proof. The correctness of the algorithm follows from Lemma 17. To analyze the running time, observe that each recursive call of BEADED-TREE acts on an instance $\mathcal{T}'$ on leaf set X' such that either $|X'| < |X|$ and an optimal solution for $\mathcal{T}'$ has at most as many reticulations as an optimal solution for $\mathcal{T}$, or $X' = X$ and an optimal solution for $\mathcal{T}'$ has fewer reticulations than an optimal solution for $\mathcal{T}$. It follows that the algorithm makes at most $k + |X| + 1$ recursive calls of BEADED-TREE, where k is the reticulation number of an optimal solution to $\mathcal{T}$.

To determine the cost of a single invocation of BEADED-TREE, observe that line 14 clearly takes constant time and line 4 takes $O(n)$ time. Indeed, it takes $O(|X|) = O(n)$ time to construct a graph $G = (X, \emptyset)$. Then, for each tree T_i, we compute the connected components of its depth-1 forest in $O(|T_i|)$ time. For each such component C, we choose one of its leaves as the "representative leaf" ℓ of the component and add an edge (ℓ, x) to G for every leaf x in C. This also takes $O(|T_i|)$ time. Doing this for all trees in $\mathcal{T}$ takes $O(\sum_{i=1}^{t} |T_i|) = O(n)$ time. The split partition of $\mathcal{T}$ is now easily seen to be the partition of X into the vertex sets of the connected components of G, which can be computed in $O(n)$ time. Each iteration of the for-loop in lines 5–12, excluding lines 6 and 8 takes constant time. In line 6, the construction of $\mathcal{T}|_{S_i}$ is easily accomplished in $O(|S_i|n)$ time and the call to SUPERTREE takes $O(|S_i|n)$ time. Thus, excluding the cost of line 8, the total cost of all iterations of the for-loop is $O(\sum_{i=1}^{s} |S_i|n) = O(|X|n)$ and the entire invocation of BEADED-TREE takes $O(|X|n)$ time.

Since the algorithm makes at most $k + |X| + 1$ invocations, its total cost is thus $O((k + |X| + 1)|X|n) = O((|X|^2 + |X|k)n)$. □

6 MINIMIZING BEAD DEPTH

Lemma 19 implies that any optimal solution to an instance of BEADED TREE has a very restrictive structure. Informally speaking, there is a single path in the beaded tree that may contain any number of reticulations, and any "branch" coming off this path can contain at most one reticulation. Because of the close relationship between BEADED TREE and UNRESTRICTED MINIMAL EPISODES INFERENCE (described in Lemma 8), the same structural properties apply to optimal solutions for the latter problem: there is one main path containing any number of duplication episodes, and any path branching off from the main path contains at most one duplication episode.

This structure is quite unusual. Furthermore, it is not clear why, from a biological perspective, it should be the case that most duplications occur on a single path. For this reason, we now consider the problems UNRESTRICTED MINIMAL EPISODES DEPTH INFERENCE and BEADED TREE DEPTH.

UNRESTRICTED MINIMAL EPISODES DEPTH INFERENCE

Input. A set $\mathcal{T} = \{T_1, \ldots, T_t\}$ of MUL-trees with label sets $X_1, \ldots, X_t \subseteq X$.

Output. A duplication tree D on X with the minimum number of duplication nodes on any path from the root to a leaf and such that D is consistent with each of $T_1, \ldots, T_t$.

BEADED TREE DEPTH

Input. A set $\mathcal{T} = \{T_1, \ldots, T_t\}$ of MUL-trees with label sets $X_1, \ldots, X_t \subseteq X$.

Output. A beaded tree B on X with the minimum number of beads on any directed path and such that B weakly displays each of $T_1, \ldots, T_t$.

By a similar argument to the proof of Lemma 8, these two problems are equivalent.

UNRESTRICTED MINIMAL EPISODES DEPTH INFERENCE is loosely based on the following two assumptions: separate lineages accumulate duplications independently; there is a maximal duplication rate that does not vary too much between lineages. Given that d duplication episodes happened on one path, these assumptions make it reasonable to expect at most d duplication episodes on any other path disjoint from it (with same evolutionary length). In particular, this holds for all paths (lineages) starting at the root, which justifies the maximum depth formulation. These assumptions seem close to those of evolutionary models. However, this does not make the UNRESTRICTED MINIMAL EPISODES DEPTH INFERENCE problem model-based. The problem is still one of parsimony: we *minimize* the maximum depth or, equivalently, the duplication rate.

Note that solutions to UNRESTRICTED MINIMAL EPISODES DEPTH INFERENCE explicitly do not contain unnecessarily highly placed duplications: where the proof of Lemma 13 "zipped" duplication episodes at much as possible, we now "unzip" them to avoid "stacking" duplications as in the proof of Lemma 13. Hence, this new problem is biologically motivated and it has more reasonable solutions than UNRESTRICTED MINIMAL EPISODES INFERENCE.

Fortunately, it turns out that a similar algorithm to that for BEADED TREE can be used to solve BEADED TREE DEPTH. The difference between the two algorithms may be summed up as follows: Both algorithms begin by considering the split partition of the set of MUL-trees under consideration. If any set in this partition can be "solved" using a tree, then for both problems it is always optimal to assume that the solution does not start with a bead, but instead includes such a tree as a child of the top tree node. If the split partition consists of a single set (and there is more than one leaf), then any possible solution (even a non-optimal solution) must begin with a bead. For the remaining cases, we essentially have a choice; there exist solutions that begin with a bead and solutions that don't. The algorithm for BEADED TREE always introduces a bead in these cases; the algorithm for BEADED TREE DEPTH never does. The following lemma is the basis for our algorithm to solve Beaded Tree Depth and establishes its correctness.

Lemma 21. *Let $\mathcal{T}$ be an instance of Beaded Tree Depth, and let $\{S_1, \ldots, S_s\}$ be the split partition of $\mathcal{T}$. If $|X| = 1$ and $\max_{1 \le i \le t} |L(T_i)| = 1$, then the optimal solution is the tree with a single leaf on X. Otherwise, if $s = 1$, then every optimal solution B has a bead (u, v) at the root and the child of v is the root of an optimal solution for $F_1(\mathcal{T})$. If $s > 1$, then any network B obtained by joining an optimal solution for $\mathcal{T}|_{S_s}$ with an optimal solution for $\mathcal{T} \setminus S_s$ is optimal for $\mathcal{T}$. Such a network B always exists.*

Proof. If $|X| = 1$ and $\max_{1 \le i \le t} |L(T_i)| = 1$, then the optimal solution clearly is the tree with a single leaf on X. So assume that $|X| > 1$ and assume first that the split partition of $\mathcal{T}$ is trivial ($s = 1$). Consider an optimal solution B. We prove first that B must have a bead at the root. Assume the contrary. Since either $|X| > 1$ or $\max_{1 \le i \le t}| L(T_i)| > 1$, B is not a tree with a single leaf. Therefore, the child of the root of B is a *split node* a, that is, a tree node that is not in a bead. Let b_1 and b_2 be the children of a and let S and $X \setminus S$ be the disjoint leaf sets descended from b_1 and b_2, respectively. Since both b_1 and b_2 have non-empty sets of descendant leaves, S is a non-empty proper subset of X.

Since the split partition is trivial, there exists at least one MUL-tree $T \in \mathcal{T}$ such that some MUL-trees $T' \in F_1(T)$ has a leaf $\ell_1 \in S$ and a leaf $\ell_2 \in X \setminus S$. Let r_T be the root of T, x the child of r_T, and y_l and y_r the children of x. Without loss of generality, T' is the tree obtained from T by deleting y_r and all its descendants, and suppressing x. We show that B does not weakly display T, which is the desired contradiction. So consider any weak embedding h of T into B. If $h(y_l)$ is a proper descendant of a, then either $h(\ell_1)$ or $h(\ell_2)$ is not a descendant of $h(y_l)$, a contradiction because y_l is an ancestor of both ℓ_1 and ℓ_2 in T. Thus, $h(y_l) \in \{r, a\}$. $h(y_l) = r$ is impossible because $h(x)$ must be a proper ancestor of $h(y_l)$. Thus, $h(y_l) = a$ and $h(x) = r$. Since a is the only child of r, this implies that both paths $h(xy_l)$ and $h(xy_r)$ start with the edge ra, again a contradiction. This proves that B must have a bead at the root.

The part of B descended from this bead must be an optimal solution to $F_1(\mathcal{T})$ because otherwise we could obtain a solution for $\mathcal{T}$ that is better than B by constructing an optimal solution for $F_1(\mathcal{T})$ and adding a bead at its root. This proves the lemma for the case when $s = 1$.

Finally, assume that $\mathcal{T}$ does not have a trivial split partition, that is, $s > 1$. For any collection $\mathcal{T}'$ of MUL-trees, let $\mathrm{OPT}(\mathcal{T}')$ denote an optimal solution to $\mathcal{T}'$. For any beaded tree B, let $d(B)$ be the maximum number of beads along any root-to-leaf path in B. We show first that the beaded tree B defined in the lemma weakly displays all trees in $\mathcal{T}$.

Any MUL-tree $T \in \mathcal{T}$ with no leaves in S_s is weakly displayed by $\mathrm{OPT}(\mathcal{T} \setminus S_s)$ and therefore by B. Similarly, if all leaves of T belong to S_s, then T is weakly displayed by $\mathrm{OPT}(\mathcal{T}|_{S_s})$ and therefore by B. So suppose T has leaves in both S_s and $X \setminus S_s$. Since $F_1(T)$ consists of two MUL-trees and $\{S_1, \ldots, S_s\}$ is a split partition of $\mathcal{T}$, we must have $F_1(T) = \{T|_{S_s}, T \setminus S_s\}$. Since $T|_{S_s} \in \mathcal{T}_{S_s}$ and $T \setminus S_s \in \mathcal{T} \setminus S_s$, the former is weakly displayed by OPT $(\mathcal{T}_{S_s})$ and the latter is weakly displayed by $\mathrm{OPT}(\mathcal{T} \setminus S_s)$. Thus, T is once again weakly displayed by B. This shows that B weakly displays all trees in $\mathcal{T}$.

Now, since $\mathrm{OPT}(\mathcal{T})$ weakly displays all MUL-trees in $\mathcal{T}|_{S_s}$ and $\mathcal{T} \setminus S_s$ and B is obtained by joining $\mathrm{OPT}(\mathcal{T}|_{S_s})$ and $\mathrm{OPT}(\mathcal{T} \setminus S_s)$, we have $d(B) = \max(d(\mathrm{OPT}(\mathcal{T}|_{S_s})), d(\mathrm{OPT}\ (\mathcal{T} \setminus S_s))) \le d(\mathrm{OPT}(\mathcal{T}))$, that is, B is an optimal solution for $\mathcal{T}$. □

Algorithm 2. Algorithm Bead-Depth($\mathcal{T}$)

```
   Input: A set of MUL-trees $\mathcal{T} = \{T_1, \ldots, T_t\}$
   Output: A beaded tree $B$ with the minimum bead depth
   that weakly displays all MUL-trees in $\mathcal{T}$
 1 if $|X| = 1$ and $\max_{1 \le i \le t} |L(T_i)| = 1$ then
 2   return a tree with 1 leaf on $X$;
 3 else
 4   Calculate the split partition $\{S_1, \ldots, S_s\}$ of $\mathcal{T}$;
 5   if $s = 1$ then
 6     $B' \leftarrow$ Bead-Depth $(F_1(\mathcal{T}))$;
 7     Construct $B$ by adding a bead whose child is the
       root of $B'$;
 8   else
 9     $B' \leftarrow$ Bead-Depth $(\mathcal{T}|_{S_s})$;
10     $B'' \leftarrow$ Bead-Depth $(\mathcal{T} \setminus S_s)$;
11     Construct $B$ by joining $B'$ and $B''$;
12   end
13   return $B$;
14 end
```

Lemma 22. *Let $\mathcal{T} = \{T_1, \ldots, T_t\}$ be an instance of Beaded Tree Depth. Algorithm 2 finds an optimal solution for $\mathcal{T}$ in $O((|X|^2 + |X|k)n)$ time, where $n = \sum_{i=1,}^{t} |T_i|$ and k is the reticulation number of the computed solution.*

Proof. The correctness of the algorithm follows from Lemma 21. To analyze the running time, the cost per invocation of Bead-Depth is $O(|X|n)$, by the same analysis as in the proof of Lemma 20. To bound the number of recursive calls, observe that the input to the recursive call in line 6 has label set X and has an optimal solution with $k - 1$ reticulations. The inputs to the recursive calls in lines 9 and 10 have label sets S_s and $X \setminus S_s$ and have optimal solutions with k_1 and k_2 reticulations, respectively, where $k_1 + k_2 = k$. Thus, if $S(x, k)$ is the number of recursive calls made on an input with $x = |X|$ and having k reticulations in the optimal solution, we have $S(x, k) = 1 + \min(S(x, k - 1), S(x_1, k_1) + S(x_2, k_2))$, where $x_1 + x_2 = x$ and $k_1 + k_2 = k$. This recurrence has the solution $S(x, k) = 2(x + k) - 1$. Thus, the running time of the algorithm is $O((2|X| + 2k - 1) \cdot |X|n) = O((|X|^2 + |X|k)n)$. □

7 Concluding Remarks

Although we have shown that the Unrestricted Minimal Episodes Inference and Parental Hybridization problems are solvable in polynomial time, we have also shown that the phylogenies produced by solving these problems have a severely restricted structure.

The optimal phylogenetic network that our algorithm produces for the Parental Hybridization problem is always a phylogenetic tree with “beads”, where a bead consists of a speciation directly followed by a reticulation. Such solutions are not necessarily the most realistic or likely ones since they contain a lot of “extra lineages”, that is, multiple lineages of an input tree travelling through the same branch of the phylogenetic network. Minimizing the total number of extra lineages, the *XL-score* [34], irrespective of the reticulation number, is also not ideal, since there always exists a solution with zero extra lineages and possibly a very high reticulation number. Therefore, the most relevant open problem that needs to be solved is to find a phylogenetic network that minimizes a weighted sum of the XL-score and the reticulation number of the network. Another alternative problem formulation that seems reasonable is to minimize the total number of parental trees that the constructed phylogenetic network has in addition to the input trees.

Another option would be to completely exclude beads in the solutions. However, although this is an interesting theoretical open problem, we do not see a reason why the resulting optimal solutions would by any more realistic, or why it would be reasonable to assume that a speciation cannot be followed by a reticulation.

Regarding Unrestricted Minimal Episodes Inference, the situation is in some sense even worse. We have shown that *all* optimal solutions have a very specific structure: there is one main path from the root to a taxon containing potentially many duplication episodes, while each path branching off this main path contains at most one duplication episode. Although such scenarios are not to be excluded (for example see the eukaryotic species phylogeny from [24]), it is unrealistic to expect all phylogenies to look like this (see for example Fig. 1 in [11] for a phylogeny where the duplication episodes are significantly more spread out). Therefore, we have proposed an alternative problem that minimizes the “duplication depth”: the maximum number of duplication episodes that lie on any directed path. This problem can also be solved in polynomial time and we expect it to produce more realistic solutions. Moreover, note that, although the problem definition does not exclude unnecessary duplication episodes as long as they do not increase the duplication depth, our algorithm will not create such redundant duplication episodes. Nevertheless, to properly assess the two algorithms, it is necessary to implement both algorithms and extensively test them on simulated and real biological data sets.

Finally, it would be interesting to study more general problem variants, which simultaneously take different processes into account, such as duplication episodes, hybridization, and gene loss and transfers. Although such problems have been studied in a reconciliation setting where the species tree is (assumed to be) known, there has been less work on variants where the species tree or network needs to be inferred. Although such problems seem daunting, we have shown here that not knowing the species tree can actually make computational problems easier.

Acknowledgments

L. van Iersel, R. Janssen, M. Jones, and Y. Murakami were supported by the Netherlands Organization for Scientific Research (NWO), including Vidi grant 639.072.602, and L van Iersel also by the 4TU Applied Mathematics Institute. N. Zeh was supported by the Natural Sciences and Engineering Research Council of Canada.

References

[1] Y.-B. Chan, V. Ranwez, and C. Scornavacca, “Reconciliation-based detection of co-evolving gene families,” *BMC Bioinf.*, vol. 14, no. 1, 2013, Art. no. 332.

[2] B. Vernot, M. Stolzer, A. Goldman, and D. Durand, “Reconciliation with non-binary species trees,” *J. Comput. Biol.*, vol. 15, no. 8, pp. 981–1006, 2008.

[3] R. D. Page, “Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas,” *Systematic Biol.*, vol. 43, no. 1, pp. 58–77, 1994.

[4] N. Panchy, M. Lehti-Shiu, and S.-H. Shiu, “Evolution of gene duplication in plants,” *Plant Physiology*, vol. 171, no. 4, pp. 2294–2316, 2016.

[5] A. B. Reams and J. R. Roth, “Mechanisms of gene duplication and amplification,” *Cold Spring Harbor Perspectives Biol.*, vol. 7, no. 2, 2015, Art. no. a016592.

[6] J. Zhang, “Evolution by gene duplication: An update,” *Trends Ecology Evolution*, vol. 18, no. 6, pp. 292–298, 2003.

[7] P. Dehal and J. L. Boore, “Two rounds of whole Genome duplication in the ancestral vertebrate,” *PLoS Biol.*, vol. 3, no. 10, 2005, Art. no. e314.

[8] S. Ohno, U. Wolf, and N. B. Atkin, “Evolution from fish to mammals by gene duplication,” *Hereditas*, vol. 59, no. 1, pp. 169–187, 1968.

[9] S. Ohno, *Evolution by Gene Duplication*. Berlin, Germany: Springer, 1970.

[10] S. M. Glasauer and S. C. Neuhauss, “Whole-Genome duplication in teleost fishes and its evolutionary consequences,” *Mol. Genetics Genomics*, vol. 289, no. 6, pp. 1045–1060, 2014.

[11] K. L. Adams and J. F. Wendel, “Polyploidy and Genome evolution in plants,” *Current Opinion Plant Biol.*, vol. 8, no. 2, pp. 135–141, 2005.

[12] M. Goodman, J. Czelusniak, G. W. Moore, A. E. Romero-Herrera, and G. Matsuda, “Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences,” *Systematic Biol.*, vol. 28, no. 2, pp. 132–163, 1979.

[13] J. Paszek and P. Gorecki, “Efficient algorithms for Genomic duplication models,” *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 15, no. 5, pp. 1515–1524, Sep./Oct. 2018.

[14] R. D. Page and M. A. Charleston, “From gene to organismal phylogeny: Reconciled trees and the gene tree/species tree problem,” *Mol. Phylogenetics Evolution*, vol. 7, no. 2, pp. 231–240, 1997.

[15] J.-P. Doyon, C. Chauve, and S. Hamel, “Space of gene/species trees reconciliations and parsimonious models,” *J. Comput. Biol.*, vol. 16, no. 10, pp. 1399–1418, 2009.

[16] M. Fellows, M. Hallett, and U. Stege, “On the multiple gene duplication problem,” in *Proc. Int. Symp. Algorithms Comput.*, 1998, pp. 348–357.

[17] B. Ma, M. Li, and L. Zhang, “From gene trees to species trees,” *SIAM J. Comput.*, vol. 30, no. 3, pp. 729–752, 2000.

[18] M. S. Bansal and O. Eulenstein, “The multiple gene duplication problem revisited,” *Bioinf.*, vol. 24, no. 13, pp. i132–i138, 2008.

[19] J. G. Burleigh, M. S. Bansal, A. Wehe, and O. Eulenstein, “Locating multiple gene duplications through reconciled trees,” in *Proc. Annu. Int. Conf. Res. Comput. Mol. Biol.*, 2008, pp. 273–284.

[20] V. Mettanant and J. Fakcharoenphol, “A linear-time algorithm for the multiple gene duplication problem,” in *Proc. Nat. Comput. Sci. Eng. Conf. (Thailand)*, 2008, pp. 198–203.

[21] C.-W. Luo, M.-C. Chen, Y.-C. Chen, R. W. Yang, H.-F. Liu, and K.-M. Chao, “Linear-time algorithms for the multiple gene duplication problems,” *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 8, no. 1, pp. 260–265, Jan./Feb. 2011.

[22] J. G. Burleigh, M. S. Bansal, O. Eulenstein, and T. J. Vision, “Inferring species trees from gene duplication episodes,” in *Proc. 1st ACM Int. Conf. Bioinf. Comput. Biol.*, 2010, pp. 198–203.

[23] R. Dondi, M. Lafond, and C. Scornavacca, “Reconciling multiple genes trees via segmental duplications and losses,” *Algorithms Mol. Biol.*, vol. 14, no. 1, 2019, Art. no. 7.

[24] R. Guigo, I. Muchnik, and T. F. Smith, “Reconstruction of ancient molecular phylogeny,” *Mol. Phylogenetics Evolution*, vol. 6, no. 2, pp. 189–213, 1996.

[25] F.-W. Li, J. C. Villarreal, S. Kelly, C. J. Rothfels, M. Melkonian, E. Frangedakis, M. Ruhsam, E. M. Sigel, J. P. Der, J. Pittermann, et al., “Horizontal transfer of an adaptive chimeric photoreceptor from bryophytes to ferns,” *Proc. Nat. Academy Sci. United States America*, vol. 111, no. 18, pp. 6672–6677, 2014.

[26] F.-W. Li, C. J. Rothfels, M. Melkonian, J. C. Villarreal, D. W. Stevenson, S. W. Graham, G. K.-S. Wong, S. Mathews, and K. M. Pryer, "The origin and evolution of phototropins," *Frontiers Plant Sci.*, vol. 6, 2015, Art. no. 637.

[27] K. T. Huber, V. Moulton, M. Steel, and T. Wu, "Folding and unfolding phylogenetic trees and networks," *J. Math. Biol.*, vol. 73, no. 6/7, pp. 1761–1780, 2016.

[28] J. Zhu, Y. Yu, and L. Nakhleh, "In the light of deep coalescence: Revisiting trees within networks," *BMC Bioinf.*, vol. 17, no. 14, 2016, Art. no. 415.

[29] M. Bordewich and C. Semple, "Computing the minimum number of hybridization events for a consistent evolutionary history," *Discrete Appl. Math.*, vol. 155, no. 8, pp. 914–928, 2007.

[30] M. Bordewich, S. Linz, K. S. John, and C. Semple, "A reduction algorithm for computing the hybridization number of two trees," *Evol. Bioinf. Online*, vol. 3, 2007, Art. no. 86.

[31] L. van Iersel, S. Kelk, N. Lekic, C. Whidden, and N. Zeh, "Hybridization number on three rooted binary trees is EPT," *SIAM J. Discrete Math.*, vol. 30, no. 3, pp. 1607–1631, 2016.

[32] W. Albertin and P. Marullo, "Polyploidy in fungi: Evolution after whole-Genome duplication," *Proc. Roy. Soc. London B: Biological Sci.*, vol. 279, no. 1738, pp. 2497–2509, 2012.

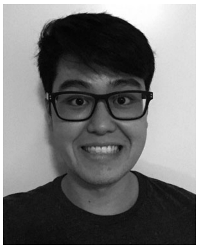
[33] A. V. Aho, Y. Sagiv, T. G. Szymanski, and J. D. Ullman, "Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions," *SIAM J. Comput.*, vol. 10, pp. 405–421, 1981.

[34] Y. Yu, C. Than, J. H. Degnan, and L. Nakhleh, "Coalescent histories on phylogenetic networks and detection of hybridization despite incomplete lineage sorting," *Systematic Biol.*, vol. 60, no. 2, pp. 138–149, 2011.

Leo van Iersel received the MSc degree in applied mathematics from Twente University, the Netherlands, in 2004, and the PhD degree from the Eindhoven University of Technology, in 2009. He is an assistant professor with the Delft University of Technology. After receiving the PhD degree, he worked as a postdoc with the University of Canterbury in New Zealand, as a teacher in different schools in Tanzania and Kenya and as a researcher with Centrum Wiskunde & Informatica (CWI) in Amsterdam.

Remie Janssen received the bachelor's degree in biology, in 2014, and the master's degree in mathematics (specializing in geometry and topology), in 2017 from Utrecht University. He is working toward the PhD degree in the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology. He started his research in phylogenetics after receiving the bachelor's degree. His research focuses on combinatorial problems in phylogenetics.

Mark Jones received the master's degree in mathematics and philosophy from Balliol College, Oxford, in 2008, and the PhD degree in computer science from Royal Holloway, University of London, in 2013. He is a postdoc researcher with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology. His research interests include parameterized complexity and phylogenetic networks.



Yukihiro Murakami received the bachelor's and master's degrees in mathematics from Corpus Christi College, Oxford, in 2017. He is working toward the PhD degree in the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology. His research interests currently lie in phylogenetics.

Norbert Zeh received the diploma degree in computer science from Friedrich-Schiller-University Jena, in 1998, and the PhD degree in computer science from Carleton University, in 2002. He is a professor of computer science with Dalhousie University. He was a postdoctoral fellow with Duke University, in 2002 and has been a faculty member with the Faculty of Computer Science, Dalhousie University since 2003.