

ResearchSpace@Auckland

Version

This is the Accepted Manuscript version. This version is defined in the NISO recommended practice RP-8-2008 <http://www.niso.org/publications/rp/>

Suggested Reference

van Iersel, L., & Linz, S. (2013). A quadratic kernel for computing the hybridization number of multiple trees. *Information Processing Letters*, 113(9), 318-323. doi: [10.1016/j.ipl.2013.02.010](https://doi.org/10.1016/j.ipl.2013.02.010)

Copyright

Items in ResearchSpace are protected by copyright, with all rights reserved, unless otherwise indicated. Previously published items are made available in accordance with the copyright policy of the publisher.

NOTICE: this is the author's version of a work that was accepted for publication in *Information Processing Letters*. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in *Information Processing Letters*, Vol. 113, No. 9, 2013 DOI: [10.1016/j.ipl.2013.02.010](https://doi.org/10.1016/j.ipl.2013.02.010)

<http://www.elsevier.com/about/open-access/open-access-policies/article-posting-policy#accepted-author-manuscript>

<http://www.sherpa.ac.uk/romeo/issn/0020-0190/>

<https://researchspace.auckland.ac.nz/docs/uoa-docs/rights.htm>

A quadratic kernel for computing the hybridization number of multiple trees

Leo van Iersel^a, Simone Linz^b

^a*Centrum Wiskunde & Informatica (CWI), P.O. Box 94079, 1090 GB, Amsterdam, Netherlands.*

^b*Center for Bioinformatics, University of Tübingen, Sand 14, 72076 Tübingen, Germany. Present Address: Department of Mathematics and Statistics, University of Canterbury, 8140 Christchurch, New Zealand.*

Abstract

It has recently been shown that the NP-hard problem of calculating the minimum number of hybridization events that is needed to explain a set of rooted binary phylogenetic trees by means of a hybridization network is fixed-parameter tractable if an instance of the problem consists of precisely two such trees. In this paper, we show that this problem remains fixed-parameter tractable for an arbitrarily large set of rooted binary phylogenetic trees. In particular, we present a quadratic kernel.

Keywords: fixed-parameter tractability, generator, hybridization, kernel, phylogenetic network

1. Introduction

Phylogenetic trees are a commonly used tool for representing evolutionary relationships. Let X be a finite set representing for example biological species or, more generally, *taxa*. A rooted *phylogenetic X -tree* is a rooted tree that has no vertices of outdegree 1 and whose leaves are bijectively labeled by the elements of X . Recently, rooted phylogenetic networks have become increasingly important in analyzing evolutionary histories of sets of taxa whose past may include reticulate evolutionary events such as horizontal gene transfer, hybridization, or recombination. Rooted phylogenetic networks are a generalization of rooted phylogenetic X -trees to directed acyclic graphs. In particular, vertices of indegree at least two are called *reticulation vertices* and represent events in which, in the context of hybridization, two distinct ancestral species combine their genomes and form a new species. The *number of reticulations* specified by a reticulation vertex is defined as its indegree minus one while the *number of reticulations* specified by a phylogenetic network N is defined as the sum of the number of reticulations over all reticulation vertices in N . To quantify the extent to which hybridization events have had an impact on the evolutionary history of a set of present-day species, the following optimization problem has attracted much interest. Let \mathcal{T} be a set of rooted phylogenetic trees on the same set of taxa. What is the minimum number of reticulations specified by any phylogenetic network that explains each of the trees in \mathcal{T} ? The decision variant of this problem, called HYBRIDIZATION NUMBER, as well as precise definitions are stated in Section 2. Since most of the research that is concerned with this question has been done in the context of hybridization, we henceforth refer to a phylogenetic network as a *hybridization network* and to reticulations specified by a network as *hybridizations*.

Since HYBRIDIZATION NUMBER is APX-hard and, thus, NP-hard even for sets of rooted phylogenetic trees consisting of precisely two binary such trees [6], many theoretical results as well as practical algorithms

Email addresses: l.j.v.iersel@gmail.com (Leo van Iersel), linz@informatik.uni-tuebingen.de (Simone Linz)

have been developed for this restricted case. In particular, it has been shown that the two-tree case is fixed-parameter tractable (FPT), regardless of whether the two rooted phylogenetic trees are binary or not [5, 15]. Roughly speaking, to establish these results, the authors used several reduction rules that shrink each problem instance to a reduced (weighted) instance whose size is linear in the value of an optimal solution. Subsequent to these results, practical algorithms have been developed that solve HYBRIDIZATION NUMBER for two rooted binary phylogenetic trees [1, 8, 9, 16, 18]. Instead of calculating an optimal hybridization network directly, all these algorithms make use of the concept of so-called *agreement forests*. Without going into details, an agreement-forest for two rooted binary phylogenetic trees T and T' on the same set of taxa is a collection of disjoint subtrees that are common to T and T' . If such a collection is, in a certain sense, acyclic and of minimum size, then its number of elements minus one equates to the solution of HYBRIDIZATION NUMBER for $\mathcal{T} = \{T, T'\}$ [2]. However, this framework of agreement forests can only be applied to more than two phylogenetic trees if one is solely interested in the minimum number of hybridization *vertices*, but not the actual minimum number of hybridizations specified by any hybridization network that explains the set of trees under consideration. These two numbers are equal in the two-tree case since each hybridization vertex has exactly two parents [14]. Given this difficulty and the computational hardness of HYBRIDIZATION NUMBER, it does not come as a surprise that, prior to this paper, there were no exact algorithms that can solve HYBRIDIZATION NUMBER for more than two trees. The only available algorithms are described in [7, 17, 8] and, in fact, are heuristics that compute lower and upper bounds for a given instance.

In this paper, we show that HYBRIDIZATION NUMBER remains fixed-parameter tractable if the input to this problem consists of arbitrarily many rooted binary phylogenetic trees on the same set of taxa. This generalization is of significant relevance for applications in (for example) evolutionary biology since biologists usually construct phylogenetic trees for more than two different genes and are interested in the number of hybridizations necessary to explain all reconstructed gene trees simultaneously. Our result shows that, as in the two-tree case, this problem can be solved by using an FPT-algorithm. We hope that this result will facilitate the development of practical algorithms in the same way as it has been the case for the restricted two-tree version of the problem.

The paper is organized as follows. The next section contains some notation and terminology that is used throughout this paper and formally states the decision problem HYBRIDIZATION NUMBER. Section 3 establishes the main result of this paper; thus showing that HYBRIDIZATION NUMBER is fixed-parameter tractable by providing a quadratic kernel. We end this paper with some concluding remarks in Section 4.

2. Preliminaries

This section provides preliminary definitions that are used throughout this paper and formally states the decision problem HYBRIDIZATION NUMBER for a set of rooted binary phylogenetic trees. Let X be a finite set. We refer to the elements of X as *taxa*.

Phylogenetic trees. A *rooted binary phylogenetic X -tree* T is a rooted tree whose root has degree two while all other interior vertices have degree three, and whose leaves are bijectively labeled by the elements of X . We identify each leaf with its label and thus refer to X as the leaf set of T . Furthermore, we regard the edges of T as being directed away from the root. Now let T be a rooted phylogenetic X -tree, and let X' be a subset of X . We define $T|X'$ to be the rooted phylogenetic X' -tree obtained from T by taking the minimal subtree of T containing all leaves in X' and suppressing all vertices with indegree 1 and outdegree 1.

Hybridization networks. A *hybridization network* N on X is a rooted acyclic digraph which has a single root of indegree 0 and outdegree at least 2, has no vertex with indegree and outdegree both 1, and in which the vertices of outdegree 0 are bijectively labeled with the elements of X . A vertex whose indegree is at least 2 is called a *hybridization vertex*. A hybridization network is *binary* if all vertices have indegree and

outdegree at most 2 and each hybridization vertex has outdegree 1. Note that a rooted binary phylogenetic X -tree is a binary hybridization network on X with no hybridization vertices.

Let N be a hybridization network on X . Furthermore, let X' be a subset of X , and let T' be a rooted phylogenetic X' -tree. Then T' is said to be a *pendant subtree* of N if it is a subtree that can be detached from N by deleting a single edge. Furthermore, if (u, v) is an edge of N , we say that u is a parent of v and v is a child of u . Note that these definitions hold in particular for rooted phylogenetic trees.

To quantify the number of hybridizations in a hybridization network N , the *hybridization number* of N is given by

$$h(N) = \sum_{v \neq \rho} (d^-(v) - 1),$$

where $d^-(v)$ is the indegree of v and ρ is the root of N .

Let N again be a hybridization network on X , and let T be a rooted binary phylogenetic X' -tree, with $X' \subseteq X$. We say that T is *displayed* by N if T can be obtained from N by deleting a subset of the edges and vertices of N and suppressing vertices with indegree and outdegree both 1. In other words, N displays T if there exist a subgraph of N that is a subdivision of T . Intuitively, if N displays T , then all of the ancestral relationships of T are visualized by N . Furthermore, for a set \mathcal{T} of rooted binary phylogenetic X' -trees, we say that N displays \mathcal{T} if N displays each tree in \mathcal{T} .

The problem HYBRIDIZATION NUMBER is to compute the minimum hybridization number of a set \mathcal{T} of rooted binary phylogenetic X -trees, which is defined as follows.

$$h(\mathcal{T}) = \min\{h(N) : N \text{ is a hybridization network that displays } \mathcal{T}\}.$$

This problem can formally be stated as the following decision problem.

Problem: HYBRIDIZATION NUMBER

Instance: A set \mathcal{T} of rooted binary phylogenetic X -trees and a positive integer k .

Question: Is $h(\mathcal{T}) \leq k$?

In the remainder of this paper, we will exclusively focus our attention on binary hybridization networks. To see that this is sufficient, we need the following lemma [10, Lemma 3].

Lemma 1. *Let N be a hybridization network on X that displays a set of rooted binary phylogenetic X -trees. Then there exists a binary hybridization network N' on X that displays \mathcal{T} such that $h(N') = h(N)$.*

Let (\mathcal{T}, k) be an instance of HYBRIDIZATION NUMBER. We will show that two reduction rules described below transform (\mathcal{T}, k) into an equivalent instance (\mathcal{T}', k) with a quadratic number of taxa. More precisely, \mathcal{T}' is a collection of rooted binary phylogenetic X' -trees such that $h(\mathcal{T}') \leq k$ if and only if $h(\mathcal{T}) \leq k$ and $|X'| \leq 20k^2$.

To describe the reduction rules, we need some additional definitions. Let \mathcal{T} be a set of rooted binary phylogenetic X -trees and let $X' \subseteq X$. A rooted phylogenetic X' -tree is a *common pendant subtree* of \mathcal{T} if it is a pendant subtree of each element in \mathcal{T} . Now, let $T \in \mathcal{T}$ and let (x_1, x_2, \dots, x_n) be a tuple of elements of X with $n \geq 2$, and let p_i be the parent of the leaf labeled x_i in T , for each $i \in \{1, 2, \dots, n\}$. Then, (x_1, x_2, \dots, x_n) is called a *chain* of T if either $(p_n, p_{n-1}, \dots, p_1)$ is a directed path in T , or $(p_n, p_{n-1}, \dots, p_2)$ is a directed path in T and $p_1 = p_2$. Furthermore, (x_1, x_2, \dots, x_n) is a *common chain* of \mathcal{T} if it is a chain of each element in \mathcal{T} .

Let (\mathcal{T}, k) be an instance of HYBRIDIZATION NUMBER. We are now in a position to state two reduction rules.

Subtree Reduction. For a common pendant subtree T of \mathcal{T} with at least two leaves, replace, in each element of \mathcal{T} , the pendant subtree T by a single leaf labeled by a new taxon (that is not yet in X).

Chain Reduction. For a common chain (x_1, x_2, \dots, x_n) of \mathcal{T} with $n > 5k$, delete, in each element of \mathcal{T} , the leaves labeled with a member of $\{x_{5k+1}, x_{5k+2}, \dots, x_n\}$ and suppress all vertices with indegree and outdegree both 1.

We remark that similar reductions have been published in the context of calculating the minimum hybridization number as well as the so-called subtree prune and regraft distance for two phylogenies and proven to be important to develop ‘efficient’ algorithms despite the NP-hardness of the underlying problems [3, 4, 5].

To obtain a proof of the kernelization for more than two trees, we need the following notion of *generators*. A *binary k -reticulation generator* (with $k \in \mathbb{N}^+$) is an acyclic directed multigraph with a single root (with indegree 0 and outdegree 1), exactly k hybridization vertices (with indegree 2 and outdegree at most 1) and all other vertices have indegree 1 and outdegree 2. Let N be a binary hybridization network, with $h(N) = k$, that has no pendant subtrees with two or more leaves. Then, a binary k -reticulation generator is said to be the *generator underlying N* if it can be obtained from N in the following way. First, delete all leaves of N and suppress each resulting vertex with indegree and outdegree both 1. Second, if the root has outdegree 2, add a new root with an edge to the old root. For a formal proof showing that the resulting directed multigraph is indeed a binary k -reticulation generator, we refer the reader to [12, Lemma 4]. Reversely, N can be reconstructed from its underlying generator by subdividing edges, adjoining a leaf to each vertex that subdivides an edge, or has indegree 2 and outdegree 0 via a new edge, and deleting the outdegree-1 root. The *sides* of a generator are its edges (the *edge sides*) and its vertices with indegree 2 and outdegree 0 (the *vertex sides*). Thus, each leaf of N is on a certain side of its underlying generator. To be more formal, let x be a leaf of N and let p be the parent of x . If p is a hybridization vertex, then p is a vertex side of the underlying generator and we say that x is on side p . If, on the other hand, p has indegree 1 and outdegree 2, then p is used to subdivide an edge side e of the underlying generator (because N has no pendant subtrees with two or more leaves) and we say that x is on side e .

Let \mathcal{T} be a set of rooted binary phylogenetic X -trees with no common pendant subtrees with two or more leaves, and let N be a binary hybridization network on X that displays \mathcal{T} . Then, clearly, N has no pendant subtrees with two or more leaves. Let G be the generator underlying N . A common chain $C = (x_1, x_2, \dots, x_n)$ of \mathcal{T} is said to *survive* in N if all elements of $\{x_1, x_2, \dots, x_n\}$ are on the same edge side of G , and C is said to be *atomized* in N if no two elements of $\{x_1, x_2, \dots, x_n\}$ are on the same side of G .

Kernels and fixed-parameter tractability. A *kernelization* of a parameterized problem is a polynomial-time algorithm that maps an instance x with parameter k to an instance x' with parameter k' such that (1) (x', k') is a yes-instance if and only if (x, k) is a yes-instance, (2) the size of x' is bounded by a function f of k , and (3) the size of k' is bounded by a function of k . A kernelization is usually referred to as a *kernel* and the function f as the *size* of the kernel. Thus, a parameterized problem admits a quadratic kernel if there exists a kernelization with f being a quadratic function. A parameterized problem is *fixed-parameter tractable* if there exists an algorithm that solves the problem in time $O(g(k)|x|^{O(1)})$, with g being some function of k and $|x|$ the size of x . Such an algorithm is called an *FPT-algorithm*. It is well known that a parameterized problem is fixed-parameter tractable if and only if it admits a kernelization and is decidable. However, not for every fixed-parameter tractable problem a kernel of polynomial size is known. Kernels are of particular interest because they can be used as a polynomial-time preprocessing which can be combined with any algorithm solving the problem.

3. Fixed-parameter tractability of HYBRIDIZATION NUMBER

In this section, we establish the following theorem which is the main result of this paper.

Theorem 1. *Let \mathcal{T} be a set of rooted binary phylogenetic X -trees, let \mathcal{T}' be the set of rooted binary phylogenetic X' -trees obtained from \mathcal{T} by applying the subtree reduction as often as possible and subsequently the chain reduction as often as possible, and let $k \in \mathbb{N}^+$. Then, $h(\mathcal{T}') \leq k$ if and only if $h(\mathcal{T}) \leq k$ and $|X'| \leq 20k^2$. In particular, HYBRIDIZATION NUMBER, parameterized by k , is fixed-parameter tractable.*

To establish Theorem 1, we need several lemmas. We start by showing that the subtree reduction does not affect the solution of any instance (\mathcal{T}, k) of HYBRIDIZATION NUMBER.

Lemma 2. *Let \mathcal{T}_0 be a set of rooted binary phylogenetic X -trees and $k \in \mathbb{N}^+$. Furthermore, let \mathcal{T}^s be the set of trees that results from a single application of the subtree reduction to \mathcal{T}_0 . Then $h(\mathcal{T}_0) \leq k$ if and only if $h(\mathcal{T}^s) \leq k$.*

PROOF. Suppose that an application of the subtree reduction to \mathcal{T}_0 replaces a common pendant subtree S by a leaf labeled s .

First, suppose that $h(\mathcal{T}_0) \leq k$. Let N_0 be a hybridization network that displays \mathcal{T}_0 such that $h(N_0) \leq k$. Furthermore, let (s_1, s_2, \dots, s_n) be an ordering on the elements of the leaf set of S . Obtain a network N_1 from N_0 by deleting the leaf labeled s_1 , and repeatedly deleting vertices with indegree 2 and outdegree 0, suppressing vertices with indegree and outdegree both 1, replacing multiple edges with a single edge and deleting vertices with indegree 0 and outdegree 1 until the resulting network is a hybridization network. Note that the operation of deleting and suppressing vertices as described in the last sentence stops eventually. Let \mathcal{T}_1 be the set of trees obtained from \mathcal{T}_0 by deleting the leaf labeled s_1 and suppressing the resulting vertex with indegree and outdegree both 1 for each tree in \mathcal{T}_0 . Now, as N_0 displays \mathcal{T}_0 , it is easily checked that N_1 displays \mathcal{T}_1 . Furthermore, we have $h(N_1) \leq h(N_0) \leq k$. We next repeat this process in order for s_2, s_3, \dots, s_{n-1} ; thereby obtaining a hybridization network N_{n-1} . Finally, obtain a hybridization network N^s from N_{n-1} by relabeling the leaf labeled s_n with s . Clearly, we have $h(N^s) \leq k$. Furthermore, since each tree in \mathcal{T}^s can be obtained from its associated tree in \mathcal{T}_0 by deleting all vertices labeled with an element in $\{s_1, s_2, \dots, s_{n-1}\}$, suppressing vertices with indegree and outdegree both 1, and relabeling the vertex labeled s_n with s , it now follows that N^s displays \mathcal{T}^s .

Second, suppose that $h(\mathcal{T}^s) \leq k$. Then there exists a hybridization network N^s that displays \mathcal{T}^s such that $h(N^s) \leq k$. Obtain a hybridization network N_0 from N^s by replacing the leaf labeled s with the tree S . Since S is pendant in each element of \mathcal{T}_0 , it follows that N_0 displays \mathcal{T}_0 . Moreover, as $h(N_0) = h(N^s) \leq k$ this establishes the lemma. \square

As a result of this lemma, we can assume throughout the remainder of this paper that the set of input trees \mathcal{T} to HYBRIDIZATION NUMBER has no common pendant subtree. To establish a similar result for the chain reduction (Lemma 5), we need two additional lemmas and some definitions.

Given two vertices u and v of a hybridization network, we say that u is an *ancestor* of v if there is a directed path from u to v . Furthermore, a vertex of a directed path P is called *internal* if it is not the first or the last vertex of P . Lastly, two directed paths P_1 and P_2 are called *internally vertex-disjoint* if there is no vertex of P_1 and P_2 that is an internal vertex of P_1 and P_2 .

Lemma 3. *Let \mathcal{T} be a set of rooted binary phylogenetic X -trees with no common pendant subtrees with at least two leaves. Then there exists a binary hybridization network N on X with $h(N) = h(\mathcal{T})$ that displays \mathcal{T} such that each common chain of \mathcal{T} either survives or is atomized in N .*

PROOF. Let N_0 be a binary hybridization network that displays \mathcal{T} such that $h(N_0) = h(\mathcal{T})$. Note that such a network exists by Lemma 1. We will construct a network N from N_0 that satisfies the statement of the lemma by considering each common chain c of \mathcal{T} that neither survives nor is atomized in N_0 and making changes to the network so that c survives in N .

Let $c = (x_1, x_2, \dots, x_n)$ be a common chain of \mathcal{T} that neither survives nor is atomized in N_0 . Note that $n \geq 3$, since any chain of two taxa that does not survive is, by definition, atomized, and that N_0 has no pendant subtrees of at least two leaves since \mathcal{T} has no common pendant subtrees of at least two leaves.

Let G_0 be the generator underlying N_0 and, for convenience, let $C = \{x_1, x_2, \dots, x_n\}$. Since c is not atomized, there exist taxa x and x' in C that are on the same side s of G_0 . Note that s can only be an edge side. Let p and p' be the parents of x and x' , respectively, and assume without loss of generality that p is an ancestor of p' . Let $e = (g, p)$ be the unique edge entering p . Hence, e is an edge of the path in N_0 corresponding to side s .

We move all taxa of C to side s ; thereby creating a network in which c survives. More precisely, we construct networks N_1 and N_2 from N_0 as follows. First, delete all taxa of C and clean up the resulting network by repeatedly deleting unlabeled outdegree-0 vertices, deleting vertices with indegree 0 and outdegree 1, suppressing vertices with indegree 1 and outdegree 1, and replacing edges that are incident with the same pair of vertices by a single edge until none of these operations is possible (and one has thus obtained a valid hybridization network). Call this intermediate network N_1 . We remark that we delete unlabeled outdegree-0 vertices because these arise whenever a leaf is deleted that is on a vertex side of G_0 . However, by moving such a leaf to an edge side, we reduce the hybridization number of the resulting network which would lead to a contradiction at the end of the proof. Thus, no taxon of C is on a vertex side of G_0 . Now, let e' be the edge of N_1 corresponding to edge e of N_0 . Subdivide e' by n vertices p_1, p_2, \dots, p_n , creating a directed path p_n, p_{n-1}, \dots, p_1 , and introduce a leaf labeled x_i and an edge (p_i, x_i) for each $i \in \{1, 2, \dots, n\}$. Call the obtained network N_2 .

It remains to show that N_2 displays \mathcal{T} . Consider any tree $T \in \mathcal{T}$. Since N_0 displays T , there exists a subtree T_0 of N_0 that is a subdivision of T . Since c is a chain of T , T_0 contains a subdivision of a caterpillar on C . In other words, there exist a directed path B in N_0 and directed paths L_n, L_{n-1}, \dots, L_1 in N_0 that start on B (in that order) and lead to x_n, x_{n-1}, \dots, x_1 respectively, such that the directed paths B, L_1, L_2, \dots, L_n are pairwise internally vertex-disjoint. Moreover, B is chosen such that the first vertex r_c of B is the first vertex of L_n and the last vertex of B is the first vertex of L_1 and the first vertex of L_2 . We next argue that p is a vertex of B . Since x and x' are on the same side of G_0 (and p is an ancestor of p'), there is a unique directed path from p to p' . Hence, any path from r_c to p' passes through p . Thus, B passes through p and it follows that p is a vertex of B . If edge $e = (g, p)$ is not an edge of B (i.e. if $x = x_n$), add g to B . Now, recall that N_1 was obtained from N_0 by deleting and suppressing vertices. By deleting or suppressing each vertex in T_0 that has been deleted or suppressed in N_0 to obtain N_1 , we obtain a subtree T_1 of N_1 that contains a subdivision of $T|(X \setminus C)$. Hence, N_1 displays $T|(X \setminus C)$. Moreover, note that e' is an edge of T_1 . Recall that N_2 was obtained from N_1 by subdividing e' and hanging leaves labeled by elements of C below the vertices subdividing e' , and observe that T can be obtained from $T|(X \setminus C)$ by applying the same operations. Therefore, we consider the subtree T_2 of N_2 obtained by applying the same operations to T_1 , and conclude that T_2 contains a subdivision of T . It follows that N_2 displays T . Since the above arguments hold for all $T \in \mathcal{T}$, it follows that N_2 displays \mathcal{T} .

By repeating the above construction for each common chain of \mathcal{T} that does not survive and is not atomized in N_0 , we obtain a network N that displays \mathcal{T} such that each common chain of \mathcal{T} either survives

or is atomized. Moreover, the changes that turned N_0 into N did not increase the reticulation number. Hence, $h(N) \leq h(N_0) = h(\mathcal{T})$. If $h(N) < h(\mathcal{T})$, we would obtain a contradiction. Therefore, we conclude that $h(N) = h(\mathcal{T})$. \square

The following lemma is implicit in [12, Theorem 3.2]. We include it here for reasons of completeness.

Lemma 4. *Let N be a binary hybridization network with $h(N) = k$, and let G be its underlying generator. Then G has at most $4k - 1$ edge sides and at most k vertex sides. In particular, G has at most $5k - 1$ sides.*

PROOF. Let n_0 be the number of vertices in G with indegree 2 and outdegree 0, let n_1 be the number of vertices in G with indegree 2 and outdegree 1, and let n_2 be the number of vertices in G with indegree 1 and outdegree 2. Then, the total indegree of G is $n_2 + 2n_1 + 2n_0$ while, considering the root vertex with indegree 0 and outdegree 1, the total outdegree of G is $1 + 2n_2 + n_1$. Hence, by the Handshaking Lemma, we have $n_2 + 2n_1 + 2n_0 = 1 + 2n_2 + n_1$ and, therefore, $n_2 = n_1 + 2n_0 - 1$. Since the number of edge sides of G , denoted $|E(G)|$, is equal to the total indegree of G and noting that $n_0 + n_1 = k$, we have

$$|E(G)| = n_2 + 2n_1 + 2n_0 = 3n_1 + 4n_0 - 1 \leq 4k - 1.$$

Furthermore, since each vertex side of G is a vertex with indegree 2, G has at most k such sides; thereby establishing the lemma. \square

Lemma 5. *Let \mathcal{T} be a set of rooted binary phylogenetic X -trees and $k \in \mathbb{N}^+$. Furthermore, let \mathcal{T}^c be the set of trees that results from a single application of the chain reduction to \mathcal{T} . Then $h(\mathcal{T}) \leq k$ if and only if $h(\mathcal{T}^c) \leq k$.*

PROOF. Let $c = (x_1, x_2, \dots, x_n)$ be a common chain of \mathcal{T} which has been reduced by a chain reduction to a common chain $c' = (x_1, x_2, \dots, x_{5k})$ of \mathcal{T}^c . Thus, $n > 5k$.

First, suppose that $h(\mathcal{T}) \leq k$. Then, by Lemma 3, there exists a binary hybridization network N , with $h(N) \leq k$, that displays \mathcal{T} such that any common chain of \mathcal{T} either survives or is atomized in N . Furthermore, by Lemma 4, the generator underlying N has at most $5k - 1$ sides. Hence, by the pigeonhole principle, c cannot be atomized in N and, therefore, survives in N . Now, let N' be the network obtained from N by replacing c with c' . More precisely, delete all leaves labeled by taxa in $\{x_{5k+1}, x_{5k+2}, \dots, x_n\}$ and suppress all resulting vertices of indegree and outdegree both 1. Then, as N displays \mathcal{T} , it is easily checked that N' displays \mathcal{T}^c and $h(N') \leq k$. Thus, $h(\mathcal{T}^c) \leq k$.

To show the other direction, suppose that $h(\mathcal{T}^c) \leq k$. Then, by Lemma 3, there exists a binary hybridization network N' with $h(N') \leq k$, that displays \mathcal{T}^c such that any common chain of \mathcal{T}^c either survives or is atomized in N' . By again using the pigeonhole principle, c' cannot be atomized in N' since it has $5k$ taxa while the generator underlying N' has at most $5k - 1$ sides. Hence, c' survives in N' . Now, let N be the network obtained from N' by replacing c' with c . To be precise, let e be the edge entering the parent, say p_{5k} , of the vertex labeled x_{5k} in N' . Since c' survives in N' , note that e is unique. Subdivide e by $n - 5k$ new vertices $p_{5k+1}, p_{5k+2}, \dots, p_n$, creating a directed path $p_n, p_{n-1}, \dots, p_{5k+1}$, and add a leaf labeled x_i and an edge (p_i, x_i) for each $i \in \{5k + 1, 5k + 2, \dots, n\}$. Then, as N' displays \mathcal{T}^c , it is easily checked that N displays \mathcal{T} and has $h(N) \leq k$. Thus, $h(\mathcal{T}) \leq k$. \square

We next show that the subtree and chain reduction can be applied to a collection of rooted binary phylogenetic X -trees until the leaf set of the resulting collection of trees has size bounded by a quadratic function of $h(\mathcal{T})$. For the proof, we follow an approach similar to the one taken by Kelk et al. [12, Lemma 3.2].

Lemma 6. *Let \mathcal{T} be a set of rooted binary phylogenetic X -trees, let \mathcal{T}' be the set of rooted binary phylogenetic X' -trees obtained from \mathcal{T} by applying the subtree and chain reduction until no further reduction is possible, and let $k \in \mathbb{N}^+$. If $h(\mathcal{T}) \leq k$, then $|X'| \leq 20k^2$.*

PROOF. As $h(\mathcal{T}) \leq k$, it follows from Lemmas 2 and 5 that $h(\mathcal{T}') \leq k$. Let N be a binary hybridization network that displays \mathcal{T}' such that $h(N) \leq k$. Furthermore, let G be its underlying binary $h(N)$ -reticulation generator.

Observe that N has no pendant subtrees of size at least 2, since otherwise \mathcal{T}' would have a common pendant subtree; thereby contradicting that the subtree reduction has been applied as often as possible. Furthermore, N does not have more than $5k$ leaves that are on the same side of G , since otherwise \mathcal{T}' would have a common chain of size greater than $5k$, thereby contradicting that the chain reduction has been applied as often as possible.

Thus, N has one leaf per vertex side of G and at most $5k$ leaves per edge side of G . By Lemma 4 (and because $h(N) \leq k$), G has at most $4k - 1$ edge sides and at most k vertex sides. Thus, the total number of leaves is at most $5k \cdot (4k - 1) + k = 20k^2 - 4k < 20k^2$. It now follows that $|X'| \leq 20k^2$. \square

Now, Theorem 1 follows from Lemmas 2, 5, and 6.

4. Concluding remarks

We remark that if one allows weighted chains, as in [5], then a slightly modified chain reduction can be used to obtain a linear kernel for a modified problem, where each common chain is associated with a weight.

While Theorem 1 proves the existence of an FPT-algorithm to solve HYBRIDIZATION NUMBER, it does not describe an explicit algorithm to do so. In order to obtain such an algorithm, one needs an exponential-time exact algorithm to solve an instance of HYBRIDIZATION NUMBER after it has been kernelized. One possible way to design an FPT-algorithm for HYBRIDIZATION NUMBER is the following. Theorem 2 of [13] establishes an algorithm—called CLUSTISTIC—that, given a set of rooted binary phylogenetic trees and an integer k , finds all binary hybridization networks that represent all clusters of the trees (in the so-called *softwired sense*, see e.g. [10]) and have hybridization number at most k . Since any network that displays a given set of rooted binary phylogenetic trees also represents all clusters of those trees, CLUSTISTIC finds it. Thus, an exponential-time exact algorithm for HYBRIDIZATION NUMBER can be obtained by using CLUSTISTIC and checking for each returned network if it displays the input trees (e.g. using the algorithm in [11], which is exponential in the number of hybridizations of a given hybridization network). In combination with the presented kernelization, this leads to an FPT-algorithm for HYBRIDIZATION NUMBER. We omit the details of this algorithm as its theoretical worst-case running time is not necessarily the best and we expect that methods are possible that are much faster in practice. Indeed, an important challenge is the development of an exponential-time exact algorithm that can be applied to reasonable-size instances.

Another major open problem is to show whether or not it is also fixed-parameter tractable to compute the minimum hybridization number of a set of arbitrary rooted phylogenetic trees; thus allowing for trees that are nonbinary.

Acknowledgements. We thank Steven Kelk and Charles Semple for many helpful discussions on the topic of this paper, and two anonymous reviewers for their comments. Leo van Iersel was funded by a Veni grant of The Netherlands Organization for Scientific Research (NWO) and Simone Linz by a Marie Curie International Outgoing Fellowship within the 7th European Community Framework Programme.

References

- [1] B. Albrecht, C. Scornavacca, A. Cenci, and D.H. Huson. Fast computation of minimum hybridization networks. *Bioinformatics*, 28(2):191–197, 2012.
- [2] M. Baroni, S. Grünewald, V. Moulton, and C. Semple. Bounding the number of hybridisation events for a consistent evolutionary history. *J. of Math. Biol.*, 51:171–182, 2005.
- [3] M.L. Bonet and K. St. John. On the complexity of uSPR distance. *IEEE/ACM Trans. Comput. Biol. Bioinf.*, 7(3):572–576, 2010.
- [4] M. Bordewich and C. Semple. On the computational complexity of the rooted subtree prune and regraft distance. *Ann. Comb.*, 8(4):409–423, 2005.
- [5] M. Bordewich and C. Semple. Computing the hybridization number of two phylogenetic trees is fixed-parameter tractable. *IEEE/ACM Trans. Comput. Biol. Bioinf.*, 4:458–466, 2007.
- [6] M. Bordewich and C. Semple. Computing the minimum number of hybridization events for a consistent evolutionary history. *Discrete Appl. Math.*, 155(8):914–928, 2007.
- [7] Z.Z. Chen and L. Wang. Algorithms for reticulate networks of multiple phylogenetic trees. *IEEE/ACM Trans. Comput. Biol. Bioinf.*, 9(2):1545–5963, 2012.
- [8] Z.Z. Chen and L. Wang. An ultrafast tool for minimum reticulate networks. *J. Comput. Biol.*, 20(1):38–41, 2013.
- [9] J. Collins, S. Linz, and C. Semple. Quantifying hybridization in realistic time. *J. Comput. Biol.*, 18:1305–1318, 2011.
- [10] L.J.J. van Iersel and S.M. Kelk. When two trees go to war. *J. Theo. Biol.*, 269(1):245–255, 2011.
- [11] I.A. Kanj, L. Nakhleh, C. Than, and G. Xia. Seeing the trees and their branches in the network is hard. *Theo. Comp. Science*, 401:153–164, 2008.
- [12] S. Kelk, L. van Iersel, N. Lekic, S. Linz, C. Scornavacca, and L. Stougie. Cycle killer... qu’est-ce que c’est? on the comparative approximability of hybridization number and directed feedback vertex set. *SIAM J. Discrete Math.*, 26(4):1635–1656, 2012.
- [13] S.M. Kelk, C. Scornavacca, and L.J.J. van Iersel. On the elusiveness of clusters. *IEEE/ACM Trans. Comput. Biol. Bioinf.*, 9(2):517–534, 2012.
- [14] S. Linz. Reticulation in evolution. *PhD thesis, Heinrich-Heine Universität, Düsseldorf, Germany*, 2008.
- [15] S. Linz and C. Semple. Hybridization in nonbinary trees. *IEEE/ACM Trans. Comput. Biol. Bioinf.*, 6(1):30–45, 2009.
- [16] C. Whidden, R.G. Beiko, and N. Zeh. Fixed-parameter and approximation algorithms for maximum agreement forests, 2011. arXiv:1108.2664v1 [q-bio.PE].
- [17] Y. Wu. Close lower and upper bounds for the minimum reticulate network of multiple phylogenetic trees. *Bioinformatics*, 26:i140–i148, 2010. Special issue: Proceedings of Intelligent Systems for Molecular Biology 2010 (ISMB2010), 10th-13th September 2010, Boston USA.
- [18] Y. Wu and W. Jiayin. Fast computation of the exact hybridization number of two phylogenetic trees. In *Bioinformatics Research and Applications (ISBRA)*, volume 6053, pages 203–214, 2010.