QUANTIFYING HYBRIDIZATION IN REALISTIC TIME

Joshua Collins[1], Simone Linz [2] [3], and Charles Semple [4]

Department of Mathematics and Statistics
University of Canterbury
Private Bag 4800
Christchurch, New Zealand

Report Number: UCDMS2009/3 MARCH 2009

[1] *Biomathematics Research Centre, Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand*
Email: j.collins@math.canterbury.ac.nz

[2] *Department of Computer Science, University of California, Davis, USA*
Email: linzs@cs.ucdavis.edu

[3] *Corresponding author: Telephone +15307523785, Fax: +15307524767*

[4] *Biomathematics Research Centre, Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand*
Email: c.semple@math.canterbury.ac.nz

Abstract

Recently, numerous practical and theoretical studies in evolutionary biology aim at calculating the extent to which reticulation—for example horizontal gene transfer, hybridization, or recombination—has influenced the evolution for a set of present-day species. It has been shown that inferring the minimum number of hybridization events that is needed to simultaneously explain the evolutionary history for a set of trees is an NP-hard and also fixed-parameter tractable problem. In this paper, we give a new fixed-parameter algorithm for computing the minimum number of hybridization events for when two rooted binary phylogenetic trees are given. This newly developed algorithm is based on interleaving—a technique using repeated kernelization steps that are applied throughout the exhaustive search part of a fixed-parameter algorithm. To show that our algorithm runs efficiently to be applicable to a wide range of practical problem instances, we apply it to a grass data set and highlight the significant improvements in terms of running times in comparison to an algorithm which has previously been implemented.

Key words: reticulate evolution, hybridization, agreement forests, interleaving, fixed-parameter tractability.

Molecular evolution (phylogenetics) is a lively field of research that is affected by a variety of scientific disciplines. Viewing it from the perspective of computer science, the NP-hardness of many fundamental problems in phylogenetics makes it a challenging subject to study. Prominent examples of such problems are the theoretically well-analyzed and widely-applied tree reconstruction methods of maximum parsimony and maximum likelihood (Foulds and Graham, 1982; Chor and Tuller, 2005; Roch, 2006). In this paper, we present a new algorithm for the following NP-hard optimization problem which arises from various phylogenetic studies. Evolutionary biologists often observe inconsistencies amongst phylogenetic trees that represent the evolution of different parts of present-day species genomes. Such inconsistencies can essentially be caused either by reticulation events like horizontal gene transfer, hybridization, and recombination, or by non-biological processes like sequencing errors or signals in the data that may yield trees whose branching patterns do not always represent the correct evolutionary history. Here, we assume that hybridization (as a representative of reticulation) has led to the observed inconsistencies. In this case, it may be more appropriate to represent the evolutionary history of a set of species by a phylogenetic network rather than a phylogenetic tree since the parents of a hybrid taxa belong to two different species. It is consequently desirable to calculate a hybridization network that simultaneously explains the evolutionary histories for a given set of trees and minimizes the number of hybridization events. The reason for the latter is that it quantifies the significance of hybridization for the evolution of the species under consideration. However, computing this minimum number is NP-hard even for two trees (Bordewich and Semple, 2007a). This two-tree problem is known as MINIMUM HYBRIDIZATION.

To overcome the computational burden of NP-hard problems, one frequently resorts to approximation algorithms, heuristics, or solving polynomial-time restrictions of the problem. However, the solutions obtained from these approaches are not always acceptable; for example, this may be due to complex and expensive processes that were needed to generate certain data sets. In such cases, fixed-parameter algorithms have proven to be a valuable tool to calculate the exact solution of a computationally-hard problem. Mathematically speaking, a problem of size n, parameterized by k, is fixed-parameter tractable if it can be computed in $O(f(k) + n^c)$, where f is an arbitrary function which only depends on k, and c is a constant which is independent of both n and k. The success of such algorithms in solving many practical problem instances can be seen in the separation of the variables k and n. Loosely speaking, the running time only depends on k and not n. Thus, if k is small, the problem may be tractable in reasonable time despite the size of the problem instance. For a more detailed description of fixed-parameter tractability (FPT), we refer the interested reader to Downey and Fellows (1998) and Flum and Grohe (2006).

Minimum Hybridization and various other problems in computational biology are known to be fixed-parameter tractable (for example, Ávila *et al.*, 2006; Bordewich and Semple, 2007b; Gramm *et al.*, 2008). However, practical fixed-parameter algorithms that have been applied to biological data sets rarely exist. Recently, Bordewich *et al.* (2007) implemented a fixed-parameter algorithm for Minimum Hybridization. By applying this algorithm to a grass data set, the authors subsequently showed that many problem instances were computable within a couple of minutes. However, there were several instances to which the algorithm did not return the exact answer in reasonable time. In particular, for three tree pairs, the running time to calculate the exact solution was at least two days. Other studies in computational biology that have introduced fixed-parameter algorithms and applied them to biological or synthetic data sets are for example described in Dehne *et al.* (2006), and Gramm and Niedermeier (2002, 2003).

To keep up with the constant progress in molecular biology, which primarily originates from the development of efficient DNA sequencing technologies, it is of importance to develop new and to further improve existing fixed-parameter algorithms such that they can cope with an increasing data set size. Beside data reduction by the so-called kernelization and bounded search tree techniques, *interleaving* has been introduced as a new method in the design of fixed-parameter algorithms (Niedermeier and Rossmanith, 2000). Interleaving refers to repeated kernelization steps while one systematically processes the bounded search tree. Apart from Abu-Khzam *et al.* (2006) and Dehne *et al.* (2006), where the authors showed that interleaving has a positive impact on the overall running time of a fixed-parameter algorithm, this technique has so far attracted more attention in theoretical analyses concerned with FPT than in practical studies.

Making use of interleaving, we present a greatly improved fixed-parameter algorithm for solving Minimum Hybridization. This improvement is highlighted by the fact that all instances of the grass data set described above can be solved in better than reasonable time. For example, an instance for which the previously implemented algorithm did not return the solution within two days, can now be calculated in less than a minute.

The new algorithm—called HybridInterleave—has been implemented in Java and is available for application at http://www.math.canterbury.ac.nz/~c.semple/software.shtml or http://wwwcsif.cs.ucdavis.edu/~linzs/. To start a calculation with HybridInterleave, the program requires the two input trees to be given in Newick format and that taxa names have been replaced with integer values. As output, the program provides the user with the minimum number of hybridization events that explain the two input trees.

This paper is organized as follows. The next section contains some preliminaries and formally states the decision problem Minimum Hybridization for which a previously established fixed-parameter algorithm is summarized in the following section. Then the new algorithm HybridInterleave and its proof of correctness are given. The subsequent section analyzes the running times of HybridInterleave when applied to a grass data set and compares it with the running times of the recently implemented algorithm HybridNumber (Bordewich *et al.*, 2007). We end this paper with some concluding remarks. Unless otherwise stated, the notation and terminology follows Semple and Steel (2003).

Preliminaries

This section provides preliminary definitions which are used throughout the rest of this paper and formally states the decision problem Minimum Hybridization.

Phylogenetic Trees

A *rooted binary phylogenetic X-tree* $\mathcal{T}$ is a rooted tree whose root has degree two while all other interior vertices have degree three. The leaf set X is the label set of $\mathcal{T}$ and frequently denoted by $\mathcal{L}(\mathcal{T})$. Furthermore, a subset A of X is a *cluster* of $\mathcal{T}$ if there is a vertex v whose set of descendants is precisely A. We view v as an ancestor and descendant of itself.

In the course of this paper, two types of subtrees play an important role. Let X' be a subset of X, and let $\mathcal{T}$ be a rooted phylogenetic X-tree. The *minimal rooted subtree* of $\mathcal{T}$ that connects all leaves in X' is denoted by $\mathcal{T}(X')$. Furthermore, the subtree obtained from $\mathcal{T}(X')$ by contracting all non-root degree-2 vertices is the *restriction of* $\mathcal{T}$ *to* X' and is denoted by $\mathcal{T}|X'$. Lastly, a subtree is *pendant* if it can be detached from $\mathcal{T}$ by deleting a single edge.

Hybridization Networks

A *hybridization network* $\mathcal{H}$ on a set X is a rooted acyclic digraph with root ρ such that the following properties are satisfied:

(i) X is the set of vertices of out-degree 0,

(ii) the out-degree of ρ is at least 2, and

(iii) for all vertices v with $d^+(v) = 1$, we have $d^-(v) \geq 2$,

where $d^+(v)$ and $d^-(v)$ denote the out-degree and in-degree of v, respectively. To quantify the number of hybridization events, the *hybridization number* of $\mathcal{H}$ with root ρ is defined as

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

where v is a vertex of $\mathcal{H}$. Since every non-root vertex has at least one parent, $d^{-}(v) - 1$ is the number of additional parents of v. Observe that if $\mathcal{H}$ is a rooted binary phylogenetic tree, then $h(\mathcal{H}) = 0$.

Now let $\mathcal{H}$ be a hybridization network on X, and let $\mathcal{T}$ be a rooted binary phylogenetic X'-tree with $X' \subseteq X$. We say that $\mathcal{T}$ is *displayed* by $\mathcal{H}$ if $\mathcal{T}$ can be obtained from $\mathcal{H}$ by deleting a subset of its edges and any resulting degree-0 vertices, and then contracting edges. Intuitively, if $\mathcal{H}$ displays $\mathcal{T}$, then all of the ancestral relationships visualized by in $\mathcal{T}$ are visualized by $\mathcal{H}$. Extending the definition of the hybridization number to two rooted binary phylogenetic X-trees $\mathcal{S}$ and $\mathcal{T}$, we set

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

With the above definition, we now formally state MINIMUM HYBRIDIZATION.

Problem: MINIMUM HYBRIDIZATION($\mathcal{S}, \mathcal{T}, k$)
Instance: Two rooted binary phylogenetic X-trees $\mathcal{S}$ and $\mathcal{T}$, and an integer k.
Question: Is $h(\mathcal{S}, \mathcal{T}) < k$?

Agreement Forests

Originating from an idea in Hein *et al.* (1996), Bordewich and Semple (2007a) showed that MINIMUM HYBRIDIZATION is NP-complete by using a characterization of the problem in terms of agreement forests. Such forests play a fundamental role in this paper. For the purpose of the upcoming definitions, we regard the root of a rooted binary phylogenetic X-tree $\mathcal{T}$ as a vertex ρ at the end of a pendant edge adjoined to the original root. For an example of two such trees, see Figure 1. Furthermore, we view ρ as an element of the label set of $\mathcal{T}$; thus $\mathcal{L}(\mathcal{T}) = X \cup \{\rho\}$. Now, let $\mathcal{S}$ and $\mathcal{T}$ be two rooted binary phylogenetic X-trees. An *agreement forest* $\mathcal{F} = \{\mathcal{L}_\rho, \mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_k\}$ for $\mathcal{S}$ and $\mathcal{T}$ is a partition of $\mathcal{L}(\mathcal{S})$ such that $\rho \in \mathcal{L}_\rho$ and the following conditions are fulfilled:

(1) For all $i \in \{\rho, 1, 2, \ldots, k\}$, we have $\mathcal{S}|\mathcal{L}_i \cong \mathcal{T}|\mathcal{L}_i$.

(2) The trees in $\{\mathcal{S}(\mathcal{L}_i) : i \in \{\rho, 1, 2, \ldots, k\}\}$ and $\{\mathcal{T}(\mathcal{L}_i) : i \in \{\rho, 1, 2, \ldots, k\}\}$ are vertex-disjoint subtrees of $\mathcal{S}$ and $\mathcal{T}$, respectively.

As an example, two agreement forests for the two rooted binary phylogenetic trees $\mathcal{S}$ and $\mathcal{T}$ depicted in Figure 1 are $\mathcal{F} = \{\{\rho, 7\}, \{1, 2, 3\}, \{4, 5, 6\}\}$ and $\mathcal{F}' = \{\{\rho, 1, 2, 3, 7\}, \{4\}, \{5\}$

A characterization of the minimum number $h(\mathcal{S}, \mathcal{T})$ of hybridization events in terms of agreement forests requires an additional condition. Without going into details, this condition avoids the possibility of species inheriting genetic material from their own descendants. Let $\mathcal{F} = \{\mathcal{L}_\rho, \mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_k\}$ be an agreement forest for $\mathcal{S}$ and $\mathcal{T}$. Let $G_\mathcal{F}$ be the directed graph that has vertex set $\mathcal{F}$ and an arc from $\mathcal{L}_i$ to $\mathcal{L}_j$ precisely if $i \neq j$, and either

(1) the root of $\mathcal{S}(\mathcal{L}_i)$ is an ancestor of the root of $\mathcal{S}(\mathcal{L}_j)$ in $\mathcal{S}$ or

(2) the root of $\mathcal{T}(\mathcal{L}_i)$ is an ancestor of the root of $\mathcal{T}(\mathcal{L}_j)$ in $\mathcal{T}$.

We call $\mathcal{F}$ an *acyclic-agreement forest* for $\mathcal{S}$ and $\mathcal{T}$ if $G_\mathcal{F}$ has no directed cycle. Moreover, if $\mathcal{F}$ contains the smallest number of parts over all acyclic-agreement forests for $\mathcal{S}$ and $\mathcal{T}$, we say that $\mathcal{F}$ is a *maximum-acyclic-agreement forest* for $\mathcal{S}$ and $\mathcal{T}$, in which case, we denote this number minus one by $m_a(\mathcal{S}, \mathcal{T})$. Figure 2 shows the two digraphs $G_\mathcal{F}$ and $G_{\mathcal{F}'}$ that are associated with the agreement forests $\mathcal{F}$ and $\mathcal{F}'$, respectively, for the two rooted binary phylogenetic X-trees $\mathcal{S}$ and $\mathcal{T}$ depicted in Figure 1. Note that, as $G_{\mathcal{F}'}$ is acyclic, $\mathcal{F}'$ is an acyclic-agreement forest for $\mathcal{S}$ and $\mathcal{T}$ while $\mathcal{F}$ is no such forest. Indeed, $\mathcal{F}'$ is a maximum-acyclic-agreement forest for $\mathcal{S}$ and $\mathcal{T}$. Baroni *et al.* (2005) established the following characterization.

Theorem 1. *Let $\mathcal{S}$ and $\mathcal{T}$ be two rooted binary phylogenetic X-trees. Then*

$$h(\mathcal{S}, \mathcal{T}) = m_a(\mathcal{S}, \mathcal{T}).$$

It is this characterization that was used to show that Minimum Hybridization is NP-complete.

Overview of a Known Fixed-Parameter Algorithm for Minimum Hybridization

In this section, we summarize the basic ideas of the first fixed-parameter algorithm for Minimum Hybridization. This algorithm is based on an earlier result that showed, for a pair of rooted binary phylogenetic X-trees $\mathcal{S}$ and $\mathcal{T}$, Minimum Hybridization is fixed-parameter tractable with $h(\mathcal{S}, \mathcal{T})$ being the parameter (Bordewich and Semple, 2007b). In establishing this result, the authors used two reductions—called the subtree and chain reduction—that kernelize $\mathcal{S}$ and $\mathcal{T}$ to two smaller trees whose number of leaves is linear in $h(\mathcal{S}, \mathcal{T})$.

Before detailing these reductions, we need some additional definitions. Let $\mathcal{T}$ be a rooted binary phylogenetic X-trees. An *n-chain* of $\mathcal{T}$ is an ordered tuple $(a_1, a_2, \ldots, a_n)$ of elements in X such that the parent of a_1 is either the same as the parent of a_2 or a child of the parent of a_2 and, for all $i \in \{2, 3, \ldots, n-1\}$, the parent of a_i is a child of the parent of a_{i+1}. Now, let $\mathcal{S}$ and $\mathcal{T}$ be two rooted binary phylogenetic X-trees, and let P be a disjoint collection of 2-element subsets of X such that each pair $\{a, b\} \in P$ is a common 2-chain of $\mathcal{S}$ and $\mathcal{T}$. Furthermore, let the weight function $w : P \rightarrow \mathbb{Z}^+$ assign each element of P a positive integer weight. We refer to $\mathcal{S}$ and $\mathcal{T}$ with an associated weight function w as a *pair of weighted phylogenetic trees on X*.

Let $\mathcal{S}$ and $\mathcal{T}$ be two weighted rooted binary phylogenetic X-trees with an associated set P.

Subtree reduction. Replace a maximal pendant subtree that is common to $\mathcal{S}$ and $\mathcal{T}$ by a single leaf with a new label. Furthermore, delete all members in P whose elements label leaves of the pendant subtree under consideration.

Chain reduction. Replace a maximal n-chain $(a_1, a_2, \ldots, a_n)$ with $n > 2$ that occurs identically in $\mathcal{S}$ and $\mathcal{T}$ by a 2-chain with new labels a and b. Furthermore, add the 2-element set $\{a, b\}$ to P with an associated weight of

$$w(\{a,b\}) = n - 2 + \sum_{\substack{\{a_i, a_j\} \in P \text{ and} \\ a_i, a_j \in \{a_1, \ldots, a_n\}}} w(\{a_i, a_j\}),$$

and delete all members in P whose elements are in $\{a_1, a_2, \ldots, a_n\}$. An explicit example of the chain reduction is shown in Figure 3, where the two rooted binary phylogenetic trees $\mathcal{S}'$ and $\mathcal{T}'$ have been obtained from $\mathcal{S}$ and $\mathcal{T}$, which are shown in Figure 1, by replacing the 3-chain $(1, 2, 3)$ with the 2-chain (a, b), and similarly, the 3-chain $(4, 5, 6)$ with (c, d). Note that $w(\{a, b\}) = w(\{c, d\}) = 1$.

The correspondence between the trees resulting from repeated applications of the subtree and chain reductions, and the initial two trees is given in the next lemma. This correspondence is done via a notion of agreement forests that extends acyclic-agreement forests. An agreement forest $\mathcal{F}$ for two rooted binary phylogenetic X-trees $\mathcal{S}$ and $\mathcal{T}$ is called *legitimate* if it is acyclic and the following property holds:

(P) For each $\{a, b\} \in P$, either $\{a\}$ and $\{b\}$ are elements in $\mathcal{F}$, or there exists an element of $\mathcal{F}$, say $\mathcal{L}$, such that $\{a, b\} \subseteq \mathcal{L}$.

Now, let $\mathcal{F}$ be a legitimate-agreement forest for two weighted rooted binary phylogenetic X-trees $\mathcal{S}$ and $\mathcal{T}$ with an associated set P of 2-element subsets of X. With

$$w_c(\mathcal{F}, P) = \sum_{\substack{\{a,b\} \in P;\ a \text{ and } b \\ \text{are singletons in } \mathcal{F}}} w(\{a, b\}),$$

we define the *weight* of $\mathcal{F}$ as

$$w(\mathcal{F}) = |\mathcal{F}| - 1 + w_c(\mathcal{F}, P)$$

and set $f(\mathcal{S}, \mathcal{T})$ to be the minimum weight of a legitimate-agreement forest for $\mathcal{S}$ and $\mathcal{T}$. Note that we always have $f(\mathcal{S}, \mathcal{T}) \geq h(\mathcal{S}, \mathcal{T})$, and $f(\mathcal{S}, \mathcal{T}) = h(\mathcal{S}, \mathcal{T})$ whenever P is empty. The next two lemmas are central to showing that MINIMUM HYBRIDIZATION is fixed-parameter tractable (Bordewich and Semple, 2007b).

Lemma 2. *Let $\mathcal{S}$ and $\mathcal{T}$ be two weighted rooted binary phylogenetic X-trees, and let $\mathcal{S}'$ and $\mathcal{T}'$ be two weighted rooted binary phylogenetic X'-trees that have been obtained from $\mathcal{S}$ and $\mathcal{T}$, respectively, by applying the subtree or chain reduction. Then $f(\mathcal{S}, \mathcal{T}) = f(\mathcal{S}', \mathcal{T}')$.*

Lemma 3. *Let $\mathcal{S}$ and $\mathcal{T}$ be two weighted rooted binary phylogenetic X-trees, and let the associated set P of 2-element subsets of X be empty. Furthermore, let $\mathcal{S}'$ and $\mathcal{T}'$ be two weighted rooted binary phylogenetic X'-trees that have been obtained from $\mathcal{S}$ and $\mathcal{T}$, respectively, by repeatedly applying the subtree and chain reduction until no further reduction is possible. Then $|X'| \leq 14h(\mathcal{S}, \mathcal{T})$.*

Cluster reduction. Besides repeatedly applying the subtree and chain reductions to kernelize a problem instance of MINIMUM HYBRIDIZATION before exhaustively calculating a legitimate-agreement forest of minimum weight, we can use a third reduction that breaks a problem instance of MINIMUM HYBRIDIZATION into two smaller subproblems. This reduction is depicted in Figure 4 and can be repeatedly intertwined with the other two reductions before the inevitable exhaustive search part of the algorithm. How the two smaller problem instances relate to the original instance is described in the next corollary. Due to Linz (2008), this corollary generalizes the unweighted version given by Baroni *et al.* (2006).

Corollary 4. *Let $\mathcal{S}$ and $\mathcal{T}$ be two weighted rooted binary phylogenetic X-trees with an associated set P, and let A be a common minimal cluster of both $\mathcal{S}$ and $\mathcal{T}$ with $|A| \geq 2$. Then,*

$$f(\mathcal{S}, \mathcal{T}) = f(\mathcal{S}|A, \mathcal{T}|A) + f(\mathcal{S}_a, \mathcal{T}_a),$$

where $\mathcal{S}_a$ and $\mathcal{T}_a$ are the trees obtained from $\mathcal{S}$ and $\mathcal{T}$, respectively, by replacing the pendant subtree whose label set is precisely A with a new leaf labeled a.

In the last corollary, the sets, P_A and P_a say, associated with $\mathcal{S}|A$ and $\mathcal{T}|A$, and $\mathcal{S}_a$ and $\mathcal{T}_a$, respectively, are

$$P_A = \{\{\ell, \ell'\} : \{\ell, \ell'\} \subseteq P \text{ and } \ell, \ell' \in A\}$$

and

$$P_a = \{\{\ell, \ell'\} : \{\ell, \ell'\} \subseteq P \text{ and } \ell, \ell' \notin A\}.$$

Remarks.

(i) Note that the cluster reduction can repeatedly be applied to break $\mathcal{S}$ and $\mathcal{T}$ into as many smaller tree pairs as possible by setting A to be a minimal common cluster of $\mathcal{S}$ and $\mathcal{T}$ with $|A| \geq 2$, and resetting $\mathcal{S}$ and $\mathcal{T}$ to be $\mathcal{S}_a$ and $\mathcal{T}_a$, respectively, before applying this reduction again until $\mathcal{S} \cong \mathcal{T}$.

(ii) We impose *maximality* on a common pendant subtree and a common n-chain and *minimality* on a common cluster to guarantee that the corresponding label set of any such common feature intersects each member of P in either both elements or neither.

A New Algorithm for Minimum Hybridization

In this section, we present our fixed-parameter algorithm HybridInterleave for Minimum Hybridization. It makes use of the subtree, chain, and cluster reductions, but importantly, in terms of obtaining significantly decreased running times (see the results section), it additionally uses interleaving.

Before outlining HybridInterleave and giving its pseudocode, we state two lemmas that are central to its correctness and description and whose proofs are given in the appendix. Let $\mathcal{T}$ be a rooted binary phylogenetic X-tree, and let ℓ and ℓ' be elements of X. To ease reading in this section, we use $\mathcal{T}[-\ell]$ to denote $\mathcal{T}|(\mathcal{L}(\mathcal{T}) - \{\ell\})$, and use

$\mathcal{T}[-\ell, \ell']$ to denote $\mathcal{T}|(\mathcal{L}(\mathcal{T}) - \{\ell, \ell'\})$. Furthermore, let P be a collection of 2-element subsets associated with two weighted rooted binary phylogenetic X-trees $\mathcal{S}$ and $\mathcal{T}$. If ℓ is contained in a member of P, we say that ℓ *crosses* P.

Lemma 5. *Let $\mathcal{S}$ and $\mathcal{T}$ be two weighted rooted binary phylogenetic X-trees with no common pendant subtree whose leaf set size is at least 2, and let P be the disjoint collection of 2-element subsets of X associated with $\mathcal{S}$ and $\mathcal{T}$. Then, for each $\ell \in X$, we have*

$$f(\mathcal{S}, \mathcal{T}) \leq f(\mathcal{S}[-\ell, \ell'], \mathcal{T}[-\ell, \ell']) + 2 + w(\{\ell, \ell'\})$$

if ℓ crosses P with $\{\ell, \ell'\} \in P$, and

$$f(\mathcal{S}, \mathcal{T}) \leq f(\mathcal{S}[-\ell], \mathcal{T}[-\ell]) + 1$$

otherwise.

Lemma 6. *Let $\mathcal{S}$ and $\mathcal{T}$ be two weighted rooted binary phylogenetic X-trees with no common pendant subtree whose leaf set size is at least 2, and let P be the disjoint collection of 2-element subsets of X associated with $\mathcal{S}$ and $\mathcal{T}$. Then there exists an element $\ell \in X$ such that either ℓ crosses P with $\{\ell, \ell'\} \in P$ and*

$$f(\mathcal{S}, \mathcal{T}) = f(\mathcal{S}[-\ell, \ell'], \mathcal{T}[-\ell, \ell']) + 2 + w(\{\ell, \ell'\}),$$

or ℓ does not cross P and

$$f(\mathcal{S}, \mathcal{T}) = f(\mathcal{S}[-\ell], \mathcal{T}[-\ell]) + 1.$$

We give a brief outline of the algorithm HybridInterleave before detailing its pseudocode. The algorithm takes as input two rooted binary phylogenetic X-trees $\mathcal{S}$ and $\mathcal{T}$, and an integer k, and outputs $h(\mathcal{S}, \mathcal{T})$ precisely if $h(\mathcal{S}, \mathcal{T}) < k$. It starts with initializing the variable P which represents the collection of 2-chains that have previously been obtained by applying a chain reduction to $\mathcal{S}$ and $\mathcal{T}$. Recall that w is the weight function associated with P. HybridInterleave then directly calls the subroutine Interleave which contains the key features of this algorithm.

If $k > 0$, INTERLEAVE initially finds all maximal pendant subtrees that are common to $\mathcal{S}$ and $\mathcal{T}$. If the resulting two trees have a label set size of at most 3, then, as $\rho \in \mathcal{L}(\mathcal{S})$, they are identical. Consequently, INTERLEAVE directly returns 0 as the minimum weight for a legitimate-agreement forest of $\mathcal{S}$ and $\mathcal{T}$. Otherwise, the algorithm proceeds with replacing each maximal common n-chain, where $n \geq 3$, with a 2-chain. Resetting $\mathcal{S}$ and $\mathcal{T}$ to be the reduced weighted trees, they always have a cluster A with $2 \leq |A| < |\mathcal{L}(\mathcal{S})|$ in common which allows for an application of the cluster reduction. This reduction returns two new tree pairs. The second pair $\mathcal{S}''$ and $\mathcal{T}''$ has been obtained from $\mathcal{S}$ and $\mathcal{T}$ by replacing $\mathcal{S}(A)$ and $\mathcal{T}(A)$, respectively, by a new leaf while the first pair is $\mathcal{S}' = \mathcal{S}|A$ and $\mathcal{T}' = \mathcal{T}|A$ (viewing the root of $\mathcal{S}'$ and $\mathcal{T}'$, respectively, as a vertex ρ' adjoined to the original root by a pendant edge). With $P' = \{\{\ell, \ell'\} \in P : \{\ell, \ell'\} \subseteq A\}$ whose associated weight function is w', the algorithm next checks whether there exists a legitimate-agreement forest for $\mathcal{S}'$ and $\mathcal{T}'$ with $f(\mathcal{S}', \mathcal{T}') < k$, where w' is obtained from w by restricting its domain to members that are subsets of A. To this end, the subroutine branches into $|A|$ computational paths, where each path corresponds to an element of A and a call to INTERLEAVE. This guarantees that an element is found for which Lemma 6 holds. Furthermore, for each $\ell \in A$, the algorithm successively resets the variable h, which was originally initialized with k, to the minimum of the current value of h and the return value of the associated recursive call to INTERLEAVE increased by $2 + w'(\{\ell, \ell'\})$ if ℓ crosses P' with $\{\ell, \ell'\}$, or increased by 1 otherwise. Thus, at each step, h equals k or it contains the minimum weight over all legitimate-agreement forests for $\mathcal{S}'$ and $\mathcal{T}'$ that have previously been considered. After at most k iterations, INTERLEAVE$(\mathcal{S}, \mathcal{T}, k)$ declares h+INTERLEAVE$(\mathcal{S}'', \mathcal{T}'', w'', k - h)$, where w'' is obtained from w by restricting its domain to members that are not subsets of A. Eventually, HYBRIDINTERLEAVE either returns $h(\mathcal{S}, \mathcal{T})$ if $h(\mathcal{S}', \mathcal{T}') = h < k$ and $h(\mathcal{S}'', \mathcal{T}'') < k - h$, or it returns k.

The pseudocode for HYBRIDINTERLEAVE is given below. The pseudocodes for the subtree, chain, and cluster reductions are given in Bordewich *et al.* (2007). Because of this and the description given earlier, we have omitted their respective pseudocodes.

Algorithm: HybridInterleave($\mathcal{S}, \mathcal{T}, k$)

```
procedure Interleave($\mathcal{S}, \mathcal{T}, w, k$)
  if $k \leq 0$
    then return ($k$)
  $(\mathcal{S}, \mathcal{T}, w) \leftarrow$ SubtreeReduction($\mathcal{S}, \mathcal{T}, w$)
  if $|\mathcal{L}(\mathcal{S})| \leq 3$
    then return (0)
  $(\mathcal{S}, \mathcal{T}, w) \leftarrow$ ChainReduction($\mathcal{S}, \mathcal{T}, w$)
  $(\mathcal{S}', \mathcal{T}', w', \mathcal{S}'', \mathcal{T}'', w'') \leftarrow$ ClusterReduction($\mathcal{S}, \mathcal{T}, w$)
  $h \leftarrow k$
  for each $\ell \in \mathcal{L}(\mathcal{S}') - \{\rho'\}$
    do {
      if $\exists \ell' \in \mathcal{L}(\mathcal{S}') - \{\rho', \ell\}$ such that $\{\ell, \ell'\} \in \mathrm{domain}\, w'$
        then {
          $\mathcal{S}' \leftarrow \mathcal{S}'[-\ell, \ell']$
          $\mathcal{T}' \leftarrow \mathcal{T}'[-\ell, \ell']$
          $h \leftarrow \min\{h,$ Interleave($\mathcal{S}', \mathcal{T}', w', h-$
              $w'(\{\ell, \ell'\}) - 2) + 2 + w'(\{\ell, \ell'\})\}$
        }
        else {
          $\mathcal{S}' \leftarrow \mathcal{S}'[-\ell]$
          $\mathcal{T}' \leftarrow \mathcal{T}'[-\ell]$
          $h \leftarrow \min\{h,$ Interleave($\mathcal{S}', \mathcal{T}', w', h - 1) + 1\}$
        }
    }
  return ($h +$ Interleave($\mathcal{S}'', \mathcal{T}'', w'', k - h$))

main
  $P \leftarrow \emptyset$
  $w : P \rightarrow \mathbb{Z}^+$
  $k \leftarrow$ Interleave($\mathcal{S}, \mathcal{T}, w, k$)
  return ($k$)
```

Remarks.

(i) Let X' be the label set of the two rooted binary phylogenetic trees that result from repeated applications of the subtree and chain reduction in the *first* call to INTERLEAVE. By Lemma 3, we freely assume for the the rest of the paper that the algorithm directly returns k if $|X'| > 14k$.

(ii) The actual implementation of HYBRIDINTERLEAVE contains several improvements compared to the above given pseudocode. They do not affect the theoretical worst-case running time, but have a significant impact on the algorithm's speed in practice. For example, if the subtree and chain reduction cannot be applied in some call to INTERLEAVE, the algorithm makes use of the numerical ordering on the leaf labels, which is required for any input to HYBRIDINTERLEAVE, by exclusively branching into a computational path if it considers a leaf whose label is greater than the label that has been considered in the previous call to INTERLEAVE. Without going into details, the reason for this is that, if some computational path that is not considered by the algorithm yields a solution, then, due to symmetry in the search tree, this solution is also found along some path considered by HYBRIDINTERLEAVE (Collins, 2009).

We next establish the correctness of HYBRIDINTERLEAVE.

Theorem 7. *Let $\mathcal{S}$ and $\mathcal{T}$ be a pair of rooted binary phylogenetic X-trees. Then the output of* HYBRIDINTERLEAVE$(\mathcal{S}, \mathcal{T}, k)$ *is $h(\mathcal{S}, \mathcal{T})$ if and only if $h(\mathcal{S}, \mathcal{T}) < k$; otherwise it is k.*

Proof. The proof is by induction on k. If $k = 0$, then INTERLEAVE immediately returns 0, and so the theorem holds. Now suppose that $k \geq 1$ and that theorem holds whenever the input parameter is at most $k - 1$. Because of the structure of HYBRIDINTERLEAVE and Corollary 4, to establish this part of the induction, it suffices to show that the first call to the **for each** loop correctly returns h+INTERLEAVE$(\mathcal{S}'', \mathcal{T}'', w'', k-h)$ with $h = f(\mathcal{S}', \mathcal{T}')$ if and only if $f(\mathcal{S}', \mathcal{T}') < k$, otherwise with $h = k$.

By Lemma 6, there is an $\ell \in \mathcal{L}(\mathcal{S}') - \{\rho'\}$ such that one of the following holds:

(a) If ℓ does not cross P, then

$$f(\mathcal{S}'[-\ell], \mathcal{T}'[-\ell]) = f(\mathcal{S}', \mathcal{T}') - 1.$$

(b) If ℓ crosses P with $\{\ell, \ell'\} \in P$, then

$$f(\mathcal{S}'[-\ell, \ell'], \mathcal{T}'[-\ell, \ell']) = f(\mathcal{S}', \mathcal{T}') - 2 - w(\{\ell, \ell'\}).$$

Moreover, by Lemma 5, for all other $\ell \in \mathcal{L}(\mathcal{S}') - \{\rho'\}$, we have that

$$f(\mathcal{S}'[-\ell], \mathcal{T}'[-\ell]) \geq f(\mathcal{S}', \mathcal{T}') - 1$$

if ℓ does not cross P and

$$f(\mathcal{S}'[-\ell, \ell']), \mathcal{T}'[-\ell, \ell']) \geq f(\mathcal{S}', \mathcal{T}') - 2 - w(\{\ell, \ell'\})$$

if ℓ crosses P with $\{\ell, \ell'\} \in P$. It now follows by the induction assumption and Lemma 5 that if $f(\mathcal{S}', \mathcal{T}') \geq k$, then the first call to the **for each** loop correctly returns k+INTERLEAVE$(\mathcal{S}'', \mathcal{T}'', w'', 0)$. Furthermore, by the induction assumption and Lemma 6, if $f(\mathcal{S}', \mathcal{T}') < k$, then the first call to the **for each** loop correctly returns h+INTERLEAVE$(\mathcal{S}'', \mathcal{T}'', w'', k-h)$, where $h = f(\mathcal{S}', \mathcal{T}')$. This completes the proof of the theorem. □

We end this section by analyzing the running time of HYBRIDINTERLEAVE and comparing it with the time complexity of a previous implemented algorithm to solve MINIMUM HYBRIDIZATION.

Proposition 8. *Let $\mathcal{S}$ and $\mathcal{T}$ be two rooted binary phylogenetic X-trees, and let P be an empty collection of* 2*-element subsets of X. Furthermore, let k be an integer. Then the running time of* HYBRIDINTERLEAVE$(\mathcal{S}, \mathcal{T}, k)$ *is $O((14k)^k n^3)$, where $n = |X|$.*

Proof. By repeatedly applying the subtree and chain reductions to $\mathcal{S}$ and $\mathcal{T}$ until no further reductions are possible, it follows from Lemma 3 that the leaf set size of $\mathcal{S}'$ and $\mathcal{T}'$ is at most $14h(\mathcal{S}, \mathcal{T})$. Furthermore, while the subtree and chain reduction can be computed in $O(n^3)$ (Bordewich and Semple, 2007b), a single application of the cluster reduction results in an $O(2n)$ algorithm. Thus, calling all three reductions takes time $O(n^3)$.

Since HybridInterleave directly returns k if $|X'| > 14k$, we may assume that $|X'| \leq 14k$. The remaining part of this proof is by induction on k. If $k = 0$, then the algorithm returns k in constant time. Now suppose that the running time of HybridInterleave is $O((14k')^{k'}n^3)$ for all $0 \leq k' < k$. Let A be a minimal common cluster of $\mathcal{S}'$ and $\mathcal{T}'$. As $14k \geq |A|$, the algorithm makes at most $14k$ calls to Interleave for the tree pair $\mathcal{S}'|A$ and $\mathcal{T}'|A$ with parameter of at most $k - 1$. Thus the running time is $O(n^3 + 14k(14(k-1))^{k-1}n^3)$ which is $O((14k)^k n^3)$ as claimed. □

Remark. Comparing the result of Proposition 8 with the running time of HybridNumber (Bordewich *et al.*, 2007) which is $O((2 \cdot 14k)^k + n^3)$ (Bordewich and Semple, 2007b), it is easily seen that the increase in the theoretical worst-case running time of HybridInterleave is due to repeated applications of the subtree, chain, and cluster reduction. However, this slight increase is negligible because the number of possibilities one needs to consider in the exhaustive search part is significantly decreased by 50 %. Indeed, the impact of repeated kernelizations becomes more beneficial with an increased k.

Experimental Results

To evaluate the performance of HybridInterleave, we applied it to a grass (*Poaceae*) data set that has previously been used for running-time analyses in the context of calculating the hybridization number (Bordewich *et al.*, 2007) and the rooted subtree prune and regraft distance (for details, see the last section) which is frequently used to calculate the dissimilarities between two phylogenies for when reticulation is not assumed to be its major cause. The *Poaceae* data set was originally provided by the Grass Phylogeny Working Group (2001) and contains DNA sequences for six genetic loci, each with up to 65 taxa. Details about this data set and how a gene tree was reconstructed for each locus can be found in Bordewich *et al.* (2007) (and references therein). Species of the *Poaceae* family are subject to numerous natural hybridization events (Ellstrand *et al.*, 1996). Therefore, the conflicting signals in this data set are more likely to be due to hybridization than to other processes causing inconsistencies.

For each of the 15 tree pairs, we restricted the two associated phylogenies to taxa that are common to both (second column of Table 1) and calculated the hybridization number of the resulting trees. The results are summarized in Table 1, where—beside the hybridization numbers—the running times for HybridNumber and HybridInterleave are compared for each tree pair. A detailed description of the former algorithm, is given by Bordewich *et al.* (2007). Note that we reran HybridNumber to guarantee consistency among the obtained running times for both algorithms. While HybridNumber computes the hybridization number for eight tree pairs within a couple of minutes, HybridInterleave does so for all instances of the *Poaceae* data set and performs significantly faster. The latter algorithm successfully completes each program run in less than 8 minutes and calculates hybridization numbers as high as 19 for gene tree pairs with up to 46 taxa. This seems remarkably quick since HybridNumber cannot calculate the exact solution for three tree pairs (*ndhF* and *ITS*, *rbcL* and *ITS*, and *rpoC2* and *ITS*) within 48 hours. The running time of HybridInterleave mostly depends on the exhaustive search part of this algorithm since the reductions can be computed in polynomial-time. Clearly, the running time primarily decreases with an increase in the number of taxa that can be reduced by any of the three reductions. On the other hand, if the reductions have little effect because the trees only share a limited amount of common features such as subtrees, chains, or clusters, then the running time greatly increases with the hybridization number.

Concluding Remarks

In this paper, we presented the new algorithm HybridInterleave that exactly calculates the hybridization number for two rooted binary phylogenetic trees. The algorithm can be applied to answer questions that consider the extent to which hybridization has influenced evolution and, therefore, shaped the current diversity of species. However, from a biological point of view, the results should carefully be interpreted since the algorithm is based on the assumption that hybridization is the only cause of gene tree inconsistencies. Moreover, it is possible that the real number of hybridization events for two trees is underestimated because HybridInterleave minimizes this number and the true biological scenario might be less parsimonious. Nevertheless, HybridInterleave provides a first step towards analyzing the occurrence of hybridization within a data set and, additionally, is remarkably quick.

We have shown that interleaving is an advantageous technique to speed-up the previously implemented fixed-parameter algorithm HybridNumber. Referring back to the running time results summarized in Table 1, it is likely that HybridInterleave can also compute the exact hybridization number in a reasonable short amount of time for problem instances that either contain bigger trees or have a greater hybridization number than those of the *Poaceae* data set. In conclusion, interleaving has proven to be most effective for our purpose of providing an exact algorithm to compute the hybridization number for two phylogenies of biologically relevant size, and we look forward to seeing whether interleaving has the same positive impact when applied to other fixed-parameter tractable problems.

We end this paper with a remark on how interleaving can also be applied to calculate the rooted subtree prune and regraft (rSPR) distance. Loosely speaking, the graph-theoretic operation of rSPR cuts (prunes) a subtree and reattaches (regrafts) it to another part of the tree. The *rSPR distance* between two arbitrary rooted binary phylogenetic X-trees $\mathcal{S}$ and $\mathcal{T}$ is the smallest number of rSPR operations that transforms $\mathcal{S}$ into $\mathcal{T}$. We denote this distance by $d_{\mathrm{rSPR}}(\mathcal{S}, \mathcal{T})$ and note that it is well-defined since one can always transform $\mathcal{S}$ into $\mathcal{T}$ via a sequence of single rSPR operations. Like Minimum Hybridization, calculating $d_{\mathrm{rSPR}}(\mathcal{S}, \mathcal{T})$ is NP-hard and fixed-parameter tractable (Bordewich and Semple, 2004). Furthermore, the following theorem was central to obtaining these results.

Theorem 9. *Let $\mathcal{S}$ and $\mathcal{T}$ be two rooted binary phylogenetic X-trees, and let $m(\mathcal{S}, \mathcal{T})$ denote the smallest number of elements among all agreement forests for $\mathcal{S}$ and $\mathcal{T}$ minus one. Then*

$$d_{\mathrm{rSPR}}(\mathcal{S}, \mathcal{T}) = m(\mathcal{S}, \mathcal{T}).$$

Given the strong similarities between the characterizations of $h(\mathcal{S}, \mathcal{T})$ and $d_{\mathrm{rSPR}}(\mathcal{S}, \mathcal{T})$ (see Theorems 1 and 9), it is not surprising that interleaving can also be applied to calculate the latter distance. However, while it is sufficient to exclusively consider 1-element subsets in the **for each** loop of HybridInterleave, for calculating the rSPR distance, we need to iterate through all proper subsets of the label set under consideration, and thus, subsequently apply analogous subtree, chain, and cluster reductions to possibly more than one tree pair. This is due to the missing acyclic property in the context of calculating $d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}')$. A detailed description of how interleaving can be applied in order to speed-up the computation of $d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}')$ is given by Collins (2009).

Acknowledgement

We thank Michael Langston for introducing and highlighting to us the use of interleaving in fixed-parameter algorithms. We thank the New Zealand Marsden Fund and the Department of Mathematics and Statistics at the University of Canterbury for their financial support. This research was partially supported by NSF grants SEI-BIO 0513910 and IIS-0803564.

References

Abu-Khzam, F .N., M. A. Langston, P. Shanbhag, and C. T. Symons. 2006. Scalable parallel algorithms for FPT problems. Algorithmica 45:269–284.

Ávila, L. F., G. García, M. Serna, and D. M. Thilikos. 2006. A list of parameterized problems in bioinformatics, Technical report LSI-06-24-R, Technical University of Catalonia.

Baroni, M., S. Grünewald, V. Moulton, and C. Semple. 2005. Bounding the number of hybridisation events for a consistent evolutionary history. J. Math. Biol. 51:171–182.

Baroni, M., C. Semple, and M. Steel. 2006. Hybrids in real time. Syst. Biol. 55:46–56.

Bordewich, M. and C. Semple. 2004. On the computational complexity of the rooted subtree prune and regraft distance. Ann. Combin. 8:409–423.

Bordewich, M. and C. Semple. 2007a. Computing the minimum number of hybridization events for a consistent evolutionary history. Discrete Appl. Math. 155:914–928.

Bordewich, M. and C. Semple. 2007b. Computing the hybridization number of two phylogenetic trees is fixed-parameter tractable. IEEE Trans. Comput. Biol Bioinf. 4:458–466.

Bordewich, M., S. Linz, K. St. John, and C. Semple. 2007. A reduction algorithm for computing the hybridization number of two trees. Evol. Bioinform. Online 3:86–98.

Chor, B. and T. Tuller. 2005. Maximum likelihood of evolutionary trees is hard. In: Proceedings of the 9th International Conference on Computational Molecular Biology (RECOMB 2005), Vol. 53, 722–744.

Collins, J. 2009. Rekernelisation algorithms in hybrid phylogenies. MSc Thesis, University of Canterbury, Christchurch, New Zealand.

Dehne, F., M. A. Langston, X. Luo, S. Pitre, P. Shaw, and Y. Zhang. 2006. The cluster editing problem: Implementations and experiments. In: Proceedings of the International Workshop on Parameterized and Exact Computation, Vol. 4169, 13–24.

Downey, R. and M. Fellows. 1998. Parameterized Complexity (Monographs in Computer Science). Springer Publishing.

Ellstrand, N. C., R. Whitkus, and L. H. Rieseberg. 1996. Distribution of spontaneous plant hybrids. P. Natl. Acad. Sci. USA 93:5090-5093.

Flum, J. and G. Grohe. 2006. Parameterized Complexity Theory. Springer Publishing.

Foulds, L. R. and R. L. Graham. 1982. The Steiner problem in phylogeny is NP-complete. Adv. Appl. Math. 3:43–49.

Gramm, J. and R. Niedermeier. 2002. Breakpoint medians and breakpoint phylogenies: A fixed-parameter approach. Bioinformatics 18:S128–S139.

Gramm, J. and R. Niedermeier. 2003. A fixed-parameter algorithm for minimum quartet inconsistency. J. Comput. Syst. Sci. 67:723–741.

Gramm, J., A. Nickelsen, and T. Tantau. 2008. Fixed-parameter algorithms in phylogenetics, The Computer Journal 51:79–101.

Grass Phylogeny Working Group. 2001. Phylogeny and subfamilial classification of the grasses (*Poaceae*). Ann. Mo. Bot. Gard. 88:373–457.

Hein, J., T. Jing, L. Wang, and K. Zhang. 1996. On the complexity of comparing evolutionary trees. Discrete Appl. Math. 71:153–169.

Linz, S. 2008. Reticulation in evolution. PhD Thesis, Heinrich-Heine-University, Düsseldorf, Germany.

Niedermeier, R. and P. Rossmanith. 2000. A general method to speed up fixed-parameter-tractable algorithms. Inform. Process. Lett. 73:125–129.

Roch, S. 2006. A short proof that phylogenetic tree reconstruction by maximum likelihood is hard. IEEE Trans. Comput. Biol Bioinf. 3:1545–5963.

Semple, C. and M. Steel. 2003. Phylogenetics. Oxford University Press.

Appendix

In this section, we restate Lemmas 5 and 6, and give their proofs of correctness.

Lemma 5. *Let $\mathcal{S}$ and $\mathcal{T}$ be two weighted rooted binary phylogenetic X-trees with no common pendant subtree whose leaf set size is at least 2, and let P be the disjoint collection of 2-element subsets of X associated with $\mathcal{S}$ and $\mathcal{T}$. Then, for each $\ell \in X$, we have*

$$f(\mathcal{S},\mathcal{T}) \leq f(\mathcal{S}[-\ell,\ell'],\mathcal{T}[-\ell,\ell']) + 2 + w(\{\ell,\ell'\})$$

if ℓ crosses P with $\{\ell,\ell'\} \in P$, and

$$f(\mathcal{S},\mathcal{T}) \leq f(\mathcal{S}[-\ell],\mathcal{T}[-\ell]) + 1$$

otherwise.

Proof. First assume that ℓ crosses P in an element $\{\ell,\ell'\}$. Let $\mathcal{F}_\ell$ be a legitimate-agreement forest for $\mathcal{S}[-\ell,\ell']$ and $\mathcal{T}[-\ell,\ell']$ of minimum weight. Then it is easily checked that

$$\mathcal{F} = \mathcal{F}_\ell \cup \{\{\ell\},\{\ell'\}\}$$

is a legitimate-agreement forest for $\mathcal{S}$ and $\mathcal{T}$. Moreover, we have $|\mathcal{F}| = |\mathcal{F}_\ell| + 2$ and $w_c(\mathcal{F}_\ell, P - \{\ell,\ell'\}) = w_c(\mathcal{F},P) - w(\{\ell,\ell'\})$. Hence,

$$\begin{aligned} & f(\mathcal{S}[-\ell,\ell'],\mathcal{T}[-\ell,\ell']) + 2 + w(\{\ell,\ell'\}) && (1)\\ &\quad = |\mathcal{F}_\ell| - 1 + w_c(\mathcal{F}_\ell, P - \{\ell,\ell'\}) + 2 + w(\{\ell,\ell'\}) \\ &\quad = |\mathcal{F}| - 1 + w_c(\mathcal{F},P) \\ &\quad \geq f(\mathcal{S},\mathcal{T}). \end{aligned}$$

Now assume that ℓ does not cross P. Let $\mathcal{F}_\ell$ be a legitimate-agreement forest for $\mathcal{S}[-\ell]$ and $\mathcal{T}[-\ell]$ of minimum weight. Again, it is clear that

$$\mathcal{F} = \mathcal{F}_\ell \cup \{\ell\}$$

is a legitimate-agreement forest for $\mathcal{S}$ and $\mathcal{T}$. Moreover, we have $|\mathcal{F}| = |\mathcal{F}_\ell| + 1$ and $w_c(\mathcal{F}_\ell, P) = w_c(\mathcal{F}, P)$. Thus

$$\begin{aligned} f(\mathcal{S}[-\ell], \mathcal{T}[-\ell]) + 1 &= |\mathcal{F}_\ell| - 1 + w_c(\mathcal{F}_\ell, P) + 1 \qquad (2) \\ &= |\mathcal{F}| - 1 + w_c(\mathcal{F}, P) \\ &\geq f(\mathcal{S}, \mathcal{T}). \end{aligned}$$

Inequalities (1) and (2) establish the lemma. □

Lemma 6. *Let $\mathcal{S}$ and $\mathcal{T}$ be two weighted rooted binary phylogenetic X-trees with no common pendant subtree whose leaf set size is at least* 2, *and let P be the disjoint collection of* 2*-element subsets of X associated with $\mathcal{S}$ and $\mathcal{T}$. Then there exists an element $\ell \in X$ such that either ℓ crosses P with $\{\ell, \ell'\} \in P$ and*

$$f(\mathcal{S}, \mathcal{T}) = f(\mathcal{S}[-\ell, \ell'], \mathcal{T}[-\ell, \ell']) + 2 + w(\{\ell, \ell'\}),$$

or ℓ does not cross P and

$$f(\mathcal{S}, \mathcal{T}) = f(\mathcal{S}[-\ell], \mathcal{T}[-\ell]) + 1.$$

Proof. Let $\mathcal{F} = \{\mathcal{L}_\rho, \mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_k\}$ be a legitimate-agreement forest for $\mathcal{S}$ and $\mathcal{T}$ of minimum weight. First observe that, as $G_\mathcal{F}$ is acyclic, it has a vertex $\mathcal{L}_i$ with $i \in \{\rho, 1, 2, \ldots, k\}$ whose out-degree is zero. Furthermore, since $\mathcal{S}$ and $\mathcal{T}$ have no common pendant subtree whose leaf set size is at least 2, $\mathcal{L}_i$ is a singleton in $\mathcal{F}$. Since ρ is never a singleton in $\mathcal{F}$ by Lemma 1 of Baroni *et al.* (2005), we may assume that $\mathcal{L}_i = \{\ell\}$, where $\ell \in X$.

First assume that ℓ crosses P in an element $\{\ell, \ell'\}$. Since $\mathcal{F}$ is legitimate, $\{\ell'\} \in \mathcal{F}$ as $\{\ell\} \in \mathcal{F}$, and so

$$\mathcal{F}_\ell = \mathcal{F} - \{\{\ell\}, \{\ell'\}\}$$

is a legitimate-agreement forest for $\mathcal{S}[-\ell,\ell']$ and $\mathcal{T}[-\ell,\ell']$. Furthermore, we have $|\mathcal{F}| = |\mathcal{F}_\ell| + 2$ and $w_c(\mathcal{F},P) = w_c(\mathcal{F}_\ell, P - \{\ell,\ell'\}) + w(\{\ell,\ell'\})$. It now follows that

$$\begin{aligned} f(\mathcal{S},\mathcal{T}) &= |\mathcal{F}| - 1 + w_c(\mathcal{F},P) \qquad (3)\\ &= |\mathcal{F}_\ell| + 2 - 1 + w_c(\mathcal{F}_\ell, P - \{\ell,\ell'\}) + w(\{\ell,\ell'\}) \\ &\geq f(\mathcal{S}[-\ell,\ell'],\mathcal{T}[-\ell,\ell']) + 2 + w(\{\ell,\ell'\}). \end{aligned}$$

Second assume that ℓ does not cross P. Since $\mathcal{F}$ is a legitimate-agreement forest for $\mathcal{S}$ and $\mathcal{T}$,

$$\mathcal{F}_\ell = \mathcal{F} - \{\ell\}$$

is such a forest for $\mathcal{S}[-\ell]$ and $\mathcal{T}[-\ell]$. Furthermore, we have $|\mathcal{F}| = |\mathcal{F}_\ell|+1$ and $w_c(\mathcal{F},P) = w_c(\mathcal{F}_\ell, P)$. It now follows that

$$\begin{aligned} f(\mathcal{S},\mathcal{T}) &= |\mathcal{F}| - 1 + w_c(\mathcal{F},P) \qquad (4)\\ &= |\mathcal{F}_\ell| + 1 - 1 + w_c(\mathcal{F}_\ell, P) \\ &\geq f(\mathcal{S}[-\ell],\mathcal{T}[-\ell]) + 1. \end{aligned}$$

Combining (3) and (4) with Lemma 5 gives the lemma. □

Table 1: Running time comparison of HybridInterleave with HybridNumber (Bordewich *et al.*, 2007) for the *Poaceae* data set (Grass Phylogeny Working Group, 2001).

Pairwise combination		#Taxa	Hybridization number	RT[a] of HybridNumber	RT[a] of HybridInterleave
ndhF	*phyB*	40	14	5.9 h	23 s
ndhF	*rbcL*	36	13	5.3 h	3 s
ndhF	*rpoC2*	34	12	13 h	6 s
ndhF	*waxy*	19	9	150 s	$<$ 1 s
ndhF	*ITS*	46	19	$>$ 48 h	258 s
phyB	*rbcL*	21	4	$<$ 1 s	$<$ 1 s
phyB	*rpoC2*	21	7	90 s	$<$ 1 s
phyB	*waxy*	14	3	$<$ 1 s	$<$ 1 s
phyB	*ITS*	30	8	10 s	$<$ 1 s
rbcL	*rpoC2*	26	13	15.2 h	8 s
rbcL	*waxy*	12	7	132 s	$<$ 1 s
rbcL	*ITS*	29	14	$>$ 48 h	612 s
rpoC2	*waxy*	10	1	$<$ 1 s	$<$ 1 s
rpoC2	*ITS*	31	15	$>$ 48 h	57 s
waxy	*ITS*	15	8	330 s	$<$ 1 s

[a] Running time (RT) on a 2.66 GHz CPU, 2 GB RAM machine measured in seconds (s) and hours (h), respectively.

Figure Captions

Figure 1. Two rooted binary phylogenetic X-trees $\mathcal{S}$ and $\mathcal{T}$ with their roots labeled ρ.

Figure 2. Two agreement forests $\mathcal{F}$ and $\mathcal{F}'$ and their associated digraphs $G_{\mathcal{F}}$ and $G_{\mathcal{F}'}$, respectively, for $\mathcal{S}$ and $\mathcal{T}$ shown in Figure 1.

Figure 3. Two rooted binary phylogenetic X-trees $\mathcal{S}'$ and $\mathcal{T}'$ that have been obtained from $\mathcal{S}$ and $\mathcal{T}$ depicted in Figure 1 by repeated applications of the chain reduction.

Figure 4. A cluster reduction applied to the two rooted binary phylogenetic trees $\mathcal{S}$ and $\mathcal{T}$, where $\mathcal{S}_a$ and $\mathcal{T}_a$ have been obtained from $\mathcal{S}$ and $\mathcal{T}$, respectively, by replacing the pendant subtree whose label set is A with a new leaf labeled a.

FIGURES

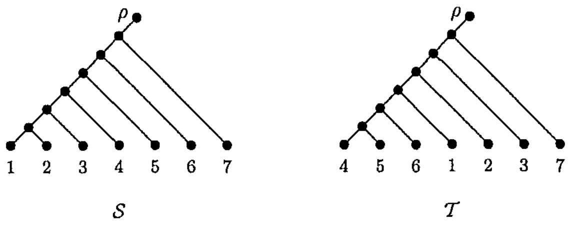

Figure 1:

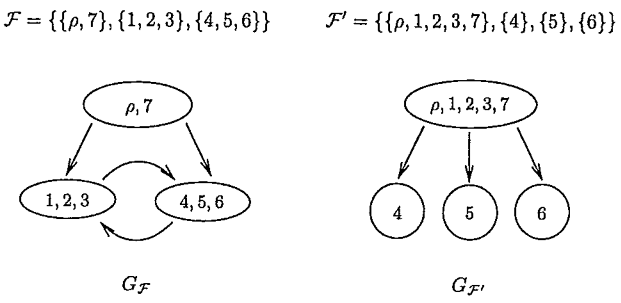
$\mathcal{F} = \{\{\rho, 7\}, \{1,2,3\}, \{4,5,6\}\}$
$\mathcal{F}' = \{\{\rho, 1,2,3,7\}, \{4\}, \{5\}, \{6\}\}$
$\rho, 7$
1, 2, 3
4, 5, 6
$\rho, 1, 2, 3, 7$
4
5
6
$G_{\mathcal{F}}$
$G_{\mathcal{F}'}$

Figure 2:

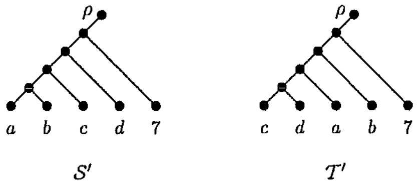

Figure 3:

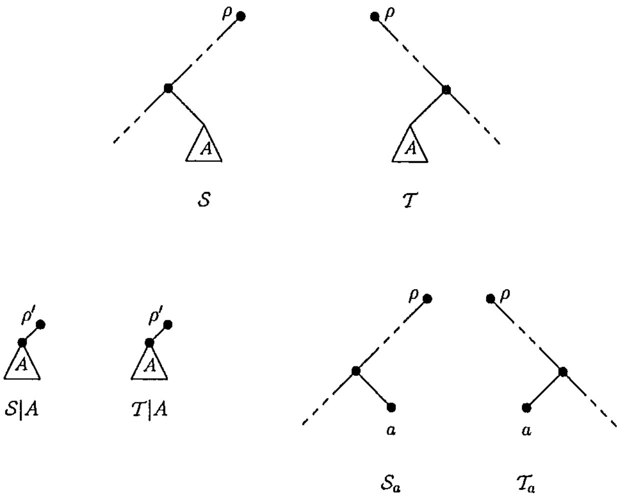
ρ
ρ
A
A
$\mathcal{S}$
$\mathcal{T}$
ρ'
ρ'
A
A
$\mathcal{S}|A$
$\mathcal{T}|A$
ρ
ρ
a
a
$\mathcal{S}_a$
$\mathcal{T}_a$

Figure 4: