ANALYZING AND RECONSTRUCTING RETICULATION NETWORKS UNDER TIMING CONSTRAINTS

Simone Linz, Charles Semple and Tanja Stadler [†]

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

Report Number: UCDMS2009/9 MAY 2009

[†] Corresponding author
All authors contributed equally

Simone Linz
Department of Computer Science, University of California, Davis, USA, email linzs@cs.ucdavis.edu

Charles Semple
Biomathematics Research Centre, Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand, email c.semple@math.canterbury.ac.nz

Tanja Stadler
Institute of Integrative Biology, ETH Zürich, Switzerland, email tanja.stadler@env.ethz.ch
phone +41 44 632 4548, Fax +41 44 632 1271

Noname manuscript No.
(will be inserted by the editor)

Analyzing and reconstructing reticulation networks under timing constraints

Simone Linz · Charles Semple · Tanja Stadler †

the date of receipt and acceptance should be inserted later

Abstract Reticulation networks are now frequently used to model the history of life for various groups of organisms whose evolutionary past is likely to include reticulation events like horizontal gene transfer or hybridization. However, the reconstructed networks are rarely guaranteed to be temporal. If a reticulation network is temporal, then it satisfies the two biologically motivated timing constraints of instantaneously occurring reticulation events and successively occurring speciation events. On the other hand, if a reticulation network is not temporal, it is always possible to resolve this issue by adding a number of additional unsampled or extinct taxa. In the first half of the paper, we show that deciding whether a given number of additional taxa is sufficient to transform a non-temporal reticulation network into a temporal one is an NP-complete problem. As one is often given a set of gene trees instead of a network in the context of hybridization, this motivates the second half of the paper which provides an algorithm for reconstructing a temporal hybridization network that simultaneously explains the ancestral history of two trees or indicates that no such network exists. We highlight two practical applications of this algorithm and illustrate the second application on a grass data set.

† Corresponding author
All authors contributed equally

Simone Linz
Department of Computer Science
University of California, Davis
USA
E-mail: linzs@cs.ucdavis.edu

Charles Semple
Biomathematics Research Centre, Department of Mathematics and Statistics
University of Canterbury, Christchurch
New Zealand
E-mail: c.semple@math.canterbury.ac.nz

Tanja Stadler
Institute of Integrative Biology
ETH Zürich
Switzerland
E-mail: tanja.stadler@env.ethz.ch
Phone: +41 44 632 4548
Fax: +41 44 632 1271

1 Introduction

Evolution is often regarded as a tree-like process in which an ancestral species evolves to a set of present-day species via a sequence of speciation events. This approach is well-suitable to tackle various questions arising from evolutionary studies. However, reticulation has now been accepted as a common force in the evolution of various species. Two major reticulation scenarios that are discussed in this paper are horizontal gene transfer (HGT) and hybridization. In the case of the latter, two ancestral lineages combine their DNA to create a new species. This process is common in certain groups of plants and fish [14]. On the other hand, in the case of HGT, which is widely observed among bacteria [17], a piece of DNA (e.g. a gene) is transferred from one organism to another coexisting species. Consequently, if the genome of a species is chimeric as a result of one or more HGT or hybridization events, its evolutionary history can often be better represented by using a reticulation network rather than a phylogenetic tree.

Recently, a lot of effort has been put into the development of algorithms to reconstruct reticulation networks for a set of present-day species (for example, see [13] and references therein). However, as pointed out in [12], although a reticulation network might explain several conflicting signals in a data set, there may be no process of instantaneously occurring reticulation events that realizes this network. Consequently, the two (resp. three) lineages that are involved in an HGT (resp. a hybridization) event are not guaranteed to exist contemporaneously. Roughly speaking, we say that a reticulation network is *temporal* if each reticulation event can be realized between coexisting species, while speciation events occur successively. Beside the reconstruction of possibly non-temporal reticulation networks, there exists a number of algorithms that do calculations on networks and implicitly assume that the input consists of a temporal reticulation network. For example, Jin *et al.* [10] have developed an algorithm for computing the parsimony score of a temporal reticulation network.

A reticulation network that is not temporal does not necessarily imply that the network is incorrect. By allowing for additional taxa that for instance correspond to unsampled or extinct taxa one can always resolve this issue without introducing any new reticulation events [2,16]. For example, consider the reticulation network shown in Figure 1, where the non-arrowed arcs are directed down the page. If one ignores the dashed arc and its end vertices, then the network is not temporal. However, by allowing for this dashed arc and the taxon x, then the resulting network is temporal. Given this, a natural task in the study of biologically meaningful reticulation networks is to calculate the minimum number of additional taxa that must be allowed for so that the resulting network is temporal. We call the analogous decision problem AddTaxa and show in the first half of this paper that this problem is NP-complete.

In the second half of this paper, we focus our attention on hybridization and consider a task that is one step closer to the initial biological data. Instead of being given a reticulation network, one frequently starts with a set of gene trees. For example, due to hybridization, these gene trees—reconstructed for different genetic loci—often reveal inconsistencies. Here, a fundamental problem is to calculate the smallest number of hybridization events needed to simultaneously explain the the set of gene trees. While this problem is NP-hard, Bordewich *et al.* [3] and, more recently, Collins *et al.* [7] have implemented a fixed-parameter algorithm for solving it when the initial set consists of two gene trees, $\mathcal{T}$ and $\mathcal{T}'$ say. This algorithm is dependent on finding an associated optimal agreement forest. For the purposes of the introduction, simply think of the forest as a smallest collection of (disjoint) subtrees common to $\mathcal{T}$ and $\mathcal{T}'$. From this forest, the algorithm HybridPhylogeny [2] can be applied to reconstruct a hybridization network that explains $\mathcal{T}$ and $\mathcal{T}'$, and in which the number of hybridization events equates to the size of the forest. However, despite its practical application, HybridPhylogeny does not guarantee that the resulting network is temporal. In the second half of the paper, we provide an algorithm, called TemporalHybrid, that constructs temporal hybridization networks from agreement forests. It is worth noting here that there is no guarantee that such networks exist. Furthermore, two practical applications of TemporalHybrid are given with the second application applied to a grass data set.

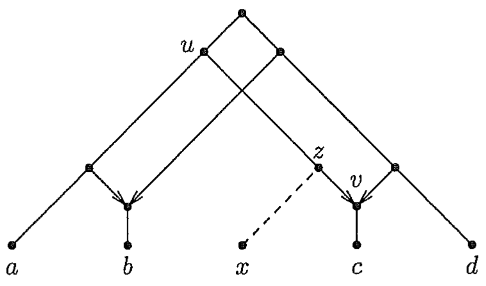

Fig. 1 A temporal hybridization network with one additional taxa x.

The paper is organized as follows. The next section contains some notation and terminology that is used throughout the paper and formally defines the decision problem AddTaxa. In Section 3, we show that AddTaxa is NP-complete. Despite this negative result, we end the section by showing how fixed-parameter algorithms for another NP-complete problem can be used to solve it. In Section 4, we present the algorithm TemporalHybrid and two practical applications of it, one of which is applied to a grass data set. Section 5 summarizes the paper. Notation and terminology typically follows [18].

2 Modeling reticulate evolution

In this section, we give some preliminary definitions for directed acyclic graphs that are commonly used to model reticulate evolution, and formally state the decision problem AddTaxa.

A *reticulation network* $\mathcal{N}$ (on X) is a rooted acyclic digraph with the following properties:

(i) the outdegree of the root is 2,
(ii) X is the set of vertices of outdegree zero (and all such vertices have indegree 1),
(iii) all remaining vertices either have indegree 1 and outdegree 2, or have indegree 2 and outdegree 1, and
(iv) for every vertex $u \notin X$, there is a vertex v such that (u, v) is an arc in $\mathcal{N}$ and v has indegree 1.

The set X represents a collection of present-day taxa. Furthermore, vertices of $\mathcal{N}$ with indegree 1 are referred to as *tree vertices*, while vertices of $\mathcal{N}$ with indegree 2 are referred to as *reticulation vertices.* Property (iv) in the definition of a reticulation network guarantees that every species that arises from either a speciation or a reticulation event exists for a certain time before going extinct. For example, a species does not yield two new hybrid species and simultaneously becomes extinct. This is biologically well-motivated since, although hybridization can result in the extinction of one or both hybridizing species, this process takes at least a few generations, and hybridization and extinction are often only locally observed [15,19]. Ignoring the dashed arc and its end vertices for the moment, Figure 1 shows a reticulation network, where $X = \{a, b, c, d\}$. Here, as in all figures in the paper, the direction of any non-arrowed arc is down the page.

Let $\mathcal{N}$ be a reticulation network with vertex set V and arc set A, and let $u, v \in V$. If there is a directed path from u to v and $u \neq v$, we write $u < v$ and refer to u as an *ancestor* of v and to v as a *descendant* of u. If (u, v) is an arc of $\mathcal{N}$, we call (u, v) a *parent arc* of v, and say that u is a *parent* of v and that v is a *child* of u.

A reticulation vertex due to hybridization is called a *hybridization vertex*, while the parent arcs of such a vertex are called *hybridization arcs.* Similarly, a reticulation vertex due to HGT is called an *HGT*

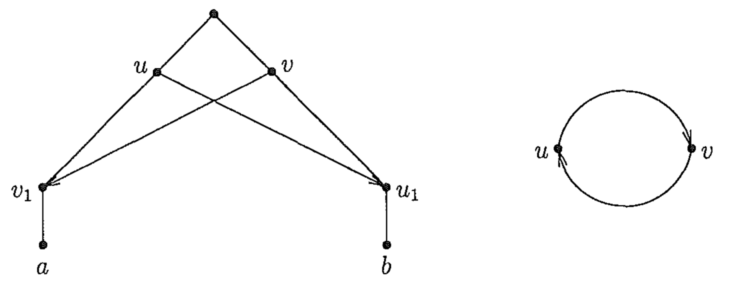

Fig. 2 A non-temporal HGT network (left) and its associated critical graph (right).

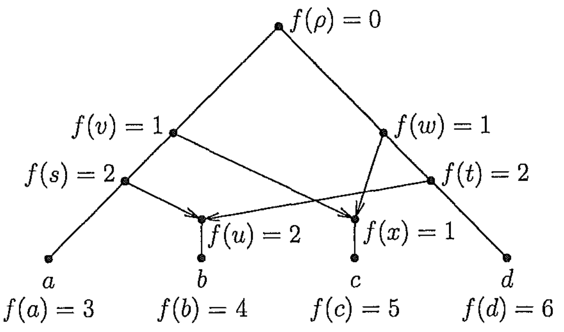

Fig. 3 A temporal hybridization network for the 4 taxa a, b, c, and d, with a temporal labeling.

vertex. The arc pointing to an HGT vertex which contributes some DNA to another lineage is called an *HGT arc*. Note that for a hybridization vertex, both parent arcs are hybridization arcs, whereas, for an HGT vertex, only one parent arc is an HGT arc. Hybridization and HGT arcs are collectively referred to as *reticulation arcs*. The remaining arcs of a reticulation network are called *tree arcs*. A *hybridization network* is a reticulation network where all reticulation events are due to hybridization, while an *HGT network* is a reticulation network where all reticulation events are due to HGT. Throughout the paper, whenever we refer to a reticulation network we mean that the network is either a hybridization network or an HGT network. Furthermore, in the context of figures, reticulation arcs are always drawn with an arrow while tree arcs are drawn without an arrow. Again ignoring the dashed arc and its end vertices, Figure 1 and the left figure in Figure 2 show a hybridization network and an HGT network, respectively.

We next formalize the notion of assigning dates to the vertices of a reticulation network. Let $\mathcal{N}$ be a reticulation network with vertex set V and arc set A, and let V' be a subset of V. Let $f : V' \rightarrow \mathbb{N} = \{0, 1, 2, \ldots\}$ be a map such that, for all $s, t \in V'$, we have $f(s) = f(t)$ whenever (s, t) is a reticulation arc, and $f(s) < f(t)$ whenever there is a directed path from s to t separated by at least one tree arc. Then f is a *partial temporal labeling* of $\mathcal{N}$. If $V' = V$, then f is a *temporal labeling* of $\mathcal{N}$, and we say that $\mathcal{N}$ is *temporal* or has a *temporal representation*. As an example, consider the hybridization network shown in Figure 3 which illustrates a temporal labeling f of this network.

Not all reticulation networks have a temporal representation. However, as noted in [2], one can resolve this issue without introducing new reticulation vertices by allowing for additional taxa. Such taxa may correspond to unsampled or extinct taxa. This can be done as follows. Let $\mathcal{N}$ be a reticulation network, and let $e = (u, v)$ be an arc of $\mathcal{N}$. Consider the reticulation network obtained from $\mathcal{N}$ by replacing e with a 2-arc directed path consisting of (u, z) and (z, v), and then adjoining a new taxa x via the new arc (z, x). We say that the resulting reticulation network has been obtained from $\mathcal{N}$ by *adding a new taxa x across* (u, v). As an example, consider Figure 1. The hybridization network shown (including the dashed arc) has been obtained from the underlying solid-arc hybridization network by adding x across

(u, v). We will soon see that by adding new taxa in this way to $\mathcal{N}$ it is always possible to produce a reticulation network that has a temporal representation. This motivates the following decision problem which is the main focus of the first part of this paper.

Problem: AddTaxa($\mathcal{N}, k$)
Instance: A reticulation network $\mathcal{N}$ and a positive integer k.
Question: Is there a temporal reticulation network that can be obtained from $\mathcal{N}$ by adding at most k new taxa?

3 AddTaxa is NP-complete

In this section, we show that AddTaxa is NP-complete. We begin with a lemma which in turn requires some further definitions. Let $\mathcal{N}$ be a reticulation network, and let (v, v_1) be a reticulation arc. If there exists a reticulation vertex w with $w \neq v_1$ such that $v < w$, but $v_1 \not< w$, then v is said to be *critical*, in which case, v_1 is a *critical reticulation vertex*. To illustrate, consider Figure 2. In the left-hand figure, u is a critical vertex as the reticulation vertex v_1 is a descendant of u, but is not a descendant of u_1.

Lemma 31 *Let $\mathcal{N}$ be a reticulation network. Then $\mathcal{N}$ has a temporal representation if and only if there exists a partial temporal labeling for the set consisting of the critical vertices and critical reticulation vertices of $\mathcal{N}$.*

Proof Evidently, if $\mathcal{N}$ has a temporal representation, then, by restricting the labeling of such a representation to the critical vertices and critical reticulation vertices, we have a partial temporal labeling of these vertices.

Now suppose that we have a partial temporal labeling f_c of the critical vertices and critical reticulation vertices of $\mathcal{N}$. By assigning values to the other vertices of $\mathcal{N}$, we extend the relative ordering of the labeling of these vertices under f_c to a temporal labeling of $\mathcal{N}$. We do this in two steps; we first extend to all reticulation vertices and their parents, and then, second, extend to all remaining tree vertices. For convenience, we will assume that the labeling is over the non-negative rationals. No generality is lost here as it is the relative ordering that is important and not the actual values assigned to the vertices.

If $\mathcal{N}$ is a hybridization network and (v, v_1) is a hybridization arc in which v is critical, then it is possible that the other parent of v_1, say v', is not critical. In this case, extend f_c by assigning v' the same value as v and v_1. Since v' is not critical, every hybridization vertex that is a descendant of v' is also a descendant of v_1. It is now easily seen that this extension is a partial temporal labeling of $\mathcal{N}$. Continuing in this way, we eventually assign appropriate values to all parents of critical hybridization vertices.

Together with the assignment given in the last paragraph in the case that $\mathcal{N}$ is a hybridization network, we now extend f_c to all remaining reticulation vertices and their respective parents so that the resulting extension is a partial temporal labeling of $\mathcal{N}$. We do this using induction on the number k of reticulation vertices not currently assigned a temporal label. If $k = 0$, then this extension is vacuous. Now suppose that we can extend f_c to include an additional $k - 1$ reticulation vertices and their respective parents, where $k \geq 1$. Let v_1 be a reticulation vertex without a label, and suppose that the parent vertices of v_1 are v and v'. Let t_a denote the maximum value of a temporal label assigned to an ancestor of v_1. If no ancestor of v_1 has been assigned such a label, set $t_a = 0$. Let t_d denote the minimum value of a temporal label assigned to a descendant of v_1. If no descendant of v_1 has been assigned such a label, set $t_d = \infty$. Note that, since v_1 is unlabeled, neither v nor v' is critical, and so every reticulation vertex that is a descendant of v or v' is a descendant of v_1. Therefore, t_d is also the minimum value of a temporal label assigned to a descendant of either v or v'. Now $t_a < t_d$; otherwise we contradict the induction assumption that we can extend f_c to $k-1$ additional reticulation vertices and their respective parents. It now follows that any rational in the interval (t_a, t_d) can be assigned to v_1, v, and v' to obtain an appropriate extension of f_c to k reticulation vertices and their parents. Thus, by induction, f_c can be extended to all such reticulation vertices and their parents.

Now that all of the reticulation vertices and their parents are labeled appropriately, we next label the remaining tree vertices of $\mathcal{N}$. We first partition the set of tree vertices (including the root) of $\mathcal{N}$ as follows. Let $C_1, C_2, \ldots, C_k$ denote the maximal connected subgraphs of $\mathcal{N}$ whose vertex sets consist entirely of tree vertices. For all i, the subgraph C_i is a rooted tree as each vertex is a tree vertex. Without loss of generality, we may assume that $C_1, C_2, \ldots, C_k$ is ordered so that $i < j$ if and only if either the root of C_i is an ancestor of the root of C_j or neither the root of C_i nor the root of C_j is an ancestor of the other. Thus C_1 contains the root of $\mathcal{N}$. Beginning with the vertices in C_1 and using this ordering, we can systematically label the remaining unlabeled tree vertices of $\mathcal{N}$ as follows. For each i, label the root with a rational bigger than that of any of its ancestors, but smaller than that of any descendant that is labeled and, for each of its leaves, label with a rational bigger than that of any currently labeled ancestor, but smaller than that of any descendant that is labeled. Now label the rest of the vertices of C_i appropriately. Note that C_i may contain a vertex v that has already been assigned a label; in which case v is a parent of a reticulation vertex. However, this is not problematic as it simply means that each ancestral vertex of v must be labeled with a rational smaller than the label assigned to v, and each descendant vertex of v must be labeled with a rational bigger than the label assigned to v. As C_i is a rooted tree, this is always possible. The resulting labeling is a temporal labeling of $\mathcal{N}$. This completes the proof of the lemma.

Remarks.

1. We remarked prior to the formal description of AddTaxa that a reticulation network $\mathcal{N}$ can always be made temporal by adding new taxa in a certain way. Indeed, because of Lemma 31, we can do this as follows. Suppose that (v, v_1) is a reticulation arc in $\mathcal{N}$ such that v is critical, and consider the reticulation network $\mathcal{N}'$ obtained from $\mathcal{N}$ by adjoining a new taxa x across (v, v_1). Let (z, x) denote the adjoining arc. Since (v, z) is not a reticulation arc, v is not critical in $\mathcal{N}'$. Furthermore, z is not critical in $\mathcal{N}'$. So $\mathcal{N}'$ has strictly less critical vertices than $\mathcal{N}$. Continuing in this way, we eventually obtain a reticulation network with no critical vertices and thus, by Lemma 31, a network that is temporal.
2. In the context of AddTaxa, we can view an instance consisting of an HGT network $\mathcal{N}$ and k as an instance consisting of a hybridization network and k. To see this, let $\mathcal{N}'$ be the hybridization network that is obtained from $\mathcal{N}$ by adding a new taxa across the parent tree arc of each HGT event and viewing each such event as a hybridization event. Clearly, $\mathcal{N}$ is a temporal HGT network if and only if $\mathcal{N}'$ is a temporal hybridization network. Moreover, if $\mathcal{N}$ is not temporal, it is easily seen that the minimum number of new taxa to add to $\mathcal{N}$ so that the resulting HGT network is temporal is equal to the minimum number of new taxa to add to $\mathcal{N}'$ so that the resulting hybridization network is temporal.

Because of the second remark, to show that AddTaxa is NP-complete, it is sufficient to show that it is NP-complete when restricted to HGT networks.

For an HGT network $\mathcal{N}$, let $C_{\mathcal{N}}$ denote the graph whose vertex set is the set of critical vertices of $\mathcal{N}$ and whose arc set is

$$\{(u, v) : u < v_1, \text{ where } (v, v_1) \text{ is an HGT arc in } \mathcal{N}\}.$$

The graph $\mathcal{C}_{\mathcal{N}}$ is called the *critical graph* of $\mathcal{N}$. As an example, the critical graph of the HGT network shown in Figure 2 is shown in the right of that figure.

Lemma 32 *Let $\mathcal{N}$ be an HGT network, and let (v, v_1) be an HGT arc of $\mathcal{N}$ such that v is a vertex of $\mathcal{C}_{\mathcal{N}}$. Let $\mathcal{N}'$ be the HGT network obtained from $\mathcal{N}$ by adding a new taxa across (v, v_1). Then the graphs $\mathcal{C}_{\mathcal{N}'}$ and $\mathcal{C}_{\mathcal{N}}\backslash v$ are equal.*

Proof First observe that, as v is not incident with an HGT arc in $\mathcal{N}'$, it is not critical. Moreover, the parent vertex, z say, of the new taxa in $\mathcal{N}'$ is also not critical since, except for the new taxa, z and v_1 have the same descendants. It now follows that the vertex sets of $\mathcal{C}_{\mathcal{N}'}$ and $\mathcal{C}_{\mathcal{N}}\backslash v$ are equal. Furthermore, for all vertices u and w in $\mathcal{C}_{\mathcal{N}'}$, we have that (u, w) is an arc in $\mathcal{C}_{\mathcal{N}'}$ if and only if it is an arc in $\mathcal{C}_{\mathcal{N}}\backslash v$. Thus the graphs $\mathcal{C}_{\mathcal{N}'}$ and $\mathcal{C}_{\mathcal{N}}\backslash v$ are equal.

Proposition 33 *Let $\mathcal{N}$ be an HGT network. Then $\mathcal{N}$ has a temporal representation if and only if $\mathcal{C}_{\mathcal{N}}$ is acyclic.*

Proof First suppose that $\mathcal{C}_{\mathcal{N}}$ is acyclic. To show that $\mathcal{N}$ has a temporal representation, it suffices to show by Lemma 31 that there is partial temporal labeling f_c of the critical vertices and critical HGT vertices of $\mathcal{N}$. We define such a labeling as follows. It is well-known and easily proved that, as $\mathcal{C}_{\mathcal{N}}$ is acyclic, it has a vertex v with indegree zero. Let (v, v_1) be the HGT arc incident with v in $\mathcal{N}$. Since v has indegree zero, no vertex in $\mathcal{C}_{\mathcal{N}}$ is an ancestor of v or v_1 in $\mathcal{N}$. Set $f_c(v) = f_c(v_1) = 1$. Now delete v in $\mathcal{C}_{\mathcal{N}}$ and consider the resulting graph. Since this graph is acyclic, it has a vertex of indegree zero. Repeat the above process for this vertex, but assign it and its child HGT vertex value 2 under f_c. By deleting this vertex and continuing in this way, we eventually assign all critical vertices and critical HGT vertices of $\mathcal{N}$ a value under f_c. Moreover, f_c is a partial temporal labeling of the critical vertices and critical HGT vertices of $\mathcal{N}$ and so, by Lemma 31, $\mathcal{N}$ has a temporal representation.

Now suppose that $\mathcal{C}_{\mathcal{N}}$ has a directed cycle $v_0, v_1, \ldots, v_{m-1}, v_0$. For all i, let v_{i1} denote the child HGT vertex of v_i in $\mathcal{N}$. Then, for all i modulo m, there is a directed path from v_{i-1} to v_{i1} containing at least one tree arc. It follows that $\mathcal{N}$ has no temporal representation. This completes the proof of the proposition.

Corollary 34 *Let $\mathcal{N}$ be an HGT network, and let V_c be a subset of the vertex set of $\mathcal{C}_{\mathcal{N}}$. Then the HGT network obtained from $\mathcal{N}$ by adding, for each $v \in V_c$, a new taxa across the HGT arc incident with v has a temporal representation if and only if $\mathcal{C}_{\mathcal{N}}\backslash V_c$ is acyclic.*

Proof Combining Lemma 32 and Proposition 33 gives the desired result.

We next show that AddTaxa is NP-complete. The NP-complete problem that we use for the polynomial-time reduction is FeedbackVertexSet [11]:

Problem: FeedbackVertexSet(G, m)
Instance: Directed graph $G = (V, E)$ and a positive integer $m \leq |V|$.
Question: Is there a subset $V' \subseteq V$ with $|V'| \leq m$ such that $G\backslash V'$ is acyclic?

Observe that if $\mathcal{N}$ is an HGT network, then, by Corollary 34, the answer to AddTaxa$(\mathcal{N}, k)$ is *yes* if and only if the answer to FeedbackVertexSet$(\mathcal{C}_{\mathcal{N}}, k)$ is *yes*. It is this observation that is exploited in the proof of the next theorem.

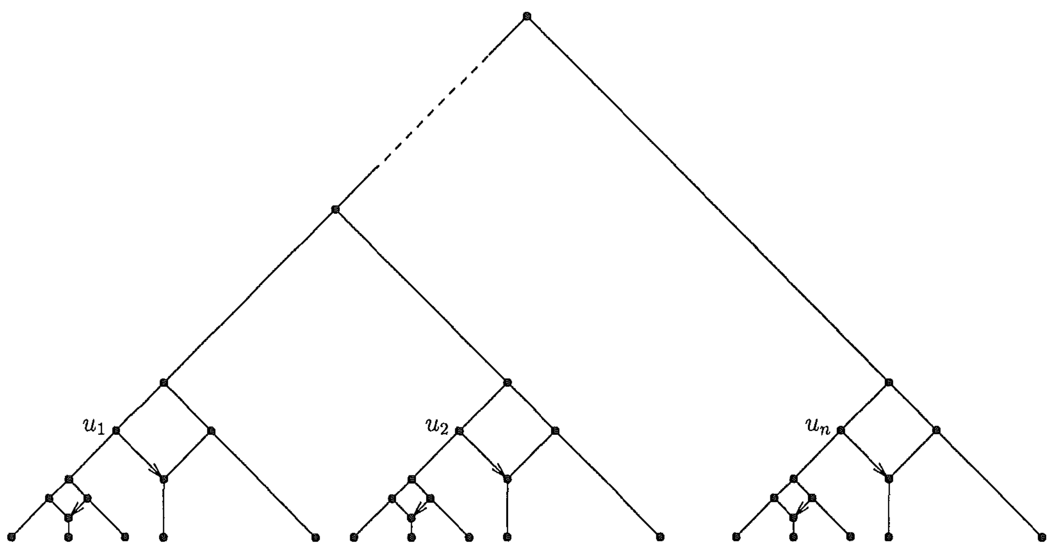

Fig. 4 The base case for the proof of Theorem 35.

Theorem 35 *The decision problem* ADDTAXA *is NP-complete.*

Proof Baroni *et al.* [2] showed that, given a reticulation network, there is a polynomial-time algorithm for deciding if this network has a temporal representation. Therefore, ADDTAXA is in NP. To show that it is NP-complete, recall from the the second remark following Lemma 31 that it suffices to show that ADDTAXA is NP-complete when restricted to HGT networks. Furthermore, by the observation prior to the statement of the theorem, it sufficient to show that, given an instance G of FEEDBACKVERTEXSET, we can construct in polynomial time an HGT network $\mathcal{N}$ whose size is polynomial in the size of G and for which $\mathcal{C}_{\mathcal{N}}$ and G are the same. The proof that such an HGT network always exists is by induction on the number k of arcs of G.

If $k = 0$, then G consists of isolated vertices, $u_1, u_2, \ldots, u_n$ say, and the HGT network shown in Figure 4 has the desired properties. Now assume that, if an instance of FEEDBACKVERTEXSET has $k-1$ arcs, where $k \geq 1$, then there is an HGT network with the desired properties.

Let G' denote the directed graph obtained from G by deleting an arbitrary arc (u, v). By the induction assumption, there is an HGT network $\mathcal{N}'$ that can be constructed in polynomial time and is polynomial in the size of G', and has the property that $\mathcal{C}_{\mathcal{N}'}$ and G' are the same. A generic picture of $\mathcal{N}'$ is shown in Figure 5. Referring to this figure, u_1 and v_1 are the child HGT vertices of u and v, respectively. Note that u is not an ancestor of v_1 since (u, v) is not an arc in G'. We complete the proof by modifying $\mathcal{N}'$ to obtain an HGT network $\mathcal{N}$ such that $\mathcal{C}_{\mathcal{N}}$ and G are the same.

Let $\mathcal{N}$ denote the HGT network shown in Figure 6, where the part of the network enclosed by the solid triangle is the same as that shown in Figure 5 except that vertices u, u_1, v, and v_1 have been renamed as y, y_1, z, and z_1, respectively. In addition to the placement of u, u_1, v, and v_1, the rest of $\mathcal{N}$ is obtained by adding non-critical HGT events. For clarity, each of these non-critical HGT events is indicated by a dashed line. Each dashed line represents a 2-arc directed path (with the middle vertex omitted) and the attachment of a new taxa (also omitted) adjoined to the middle vertex on the path. Observe that each middle vertex is non-critical in $\mathcal{N}$. If (v, u) is not an arc in G, then the dashed line e_w is not included in the construction. Clearly the construction of $\mathcal{N}$ from $\mathcal{N}'$ takes polynomial time and so, by the induction assumption the construction of $\mathcal{N}$ from G also takes polynomial time. Moreover,

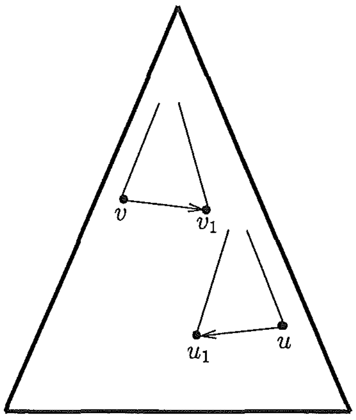

Fig. 5 The HGT network $\mathcal{N}'$ in the proof of Theorem 35. Note that u is not an ancestor of v_1, however, v is an ancestor of u_1 if (v, u) is an arc in G'.

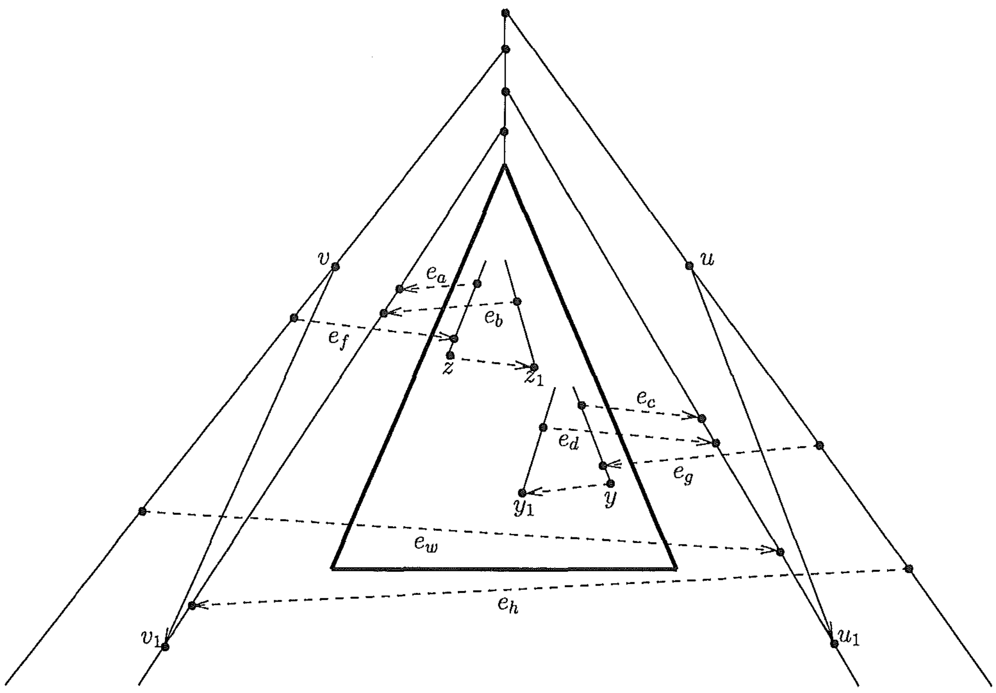

Fig. 6 The HGT network $\mathcal{N}$ constructed from $\mathcal{N}'$ in the proof of Theorem 35.

the size of $\mathcal{N}$ is polynomial in the size of $\mathcal{N}'$ and so, again by the induction assumption, the size of $\mathcal{N}$ is polynomial in the size of G. In particular, at each step of the induction process, the number of additional arcs and vertices is constant.

Because of the way in which $\mathcal{N}$ is constructed from $\mathcal{N}'$ and noting that neither y nor z is critical in $\mathcal{N}$, the set of critical vertices of $\mathcal{N}$ is exactly the same as the set of critical vertices of $\mathcal{N}'$. Thus the vertex sets of $\mathcal{C}_{\mathcal{N}}$ and G are the same. Furthermore, by construction, if w and x are vertices of $\mathcal{N}$ and $\{w, x\} \cap \{u, v\} = \emptyset$, then (w, x) is an arc in $\mathcal{C}_{\mathcal{N}}$ precisely if (w, x) is an arc in G. Thus to show that $\mathcal{C}_{\mathcal{N}}$ and G are the same, it remains to check, for each $j \in \{u, v\}$, that (w, j) is an arc in $\mathcal{C}_{\mathcal{N}}$ precisely if (w, j)

is an arc in G and that (j, w) is an arc in $\mathcal{C}_{\mathcal{N}}$ precisely if (j, w) is an arc in G. First observe that because of the path indicated by e_h, we have (u, v) in $\mathcal{C}_{\mathcal{N}}$. Furthermore, if (v, u) is in G, then the 2-arc directed path of $\mathcal{N}$ indicated by e_w means that (v, u) is in $\mathcal{C}_{\mathcal{N}}$. By symmetry, it is now sufficient to complete the check for when $j = v$.

Since any ancestor of v in $\mathcal{N}'$ remains an ancestor of v_1 in $\mathcal{N}$ and any ancestor of v_1 in $\mathcal{N}'$ remains an ancestor of v_1 in $\mathcal{N}$ because of the paths indicated by e_a and e_b, respectively, it follows by the induction assumption that (w, v) is an arc in $\mathcal{C}_{\mathcal{N}}$ precisely if (w, v) is an arc in G. Furthermore, since any descendant of v in $\mathcal{N}'$ remains a descendant of v in $\mathcal{N}$ because of the path indicated by e_f, it follows by the induction assumption that (v, w) is an arc in $\mathcal{C}_{\mathcal{N}}$ precisely if (v, w) is an arc in G. It now follows that the two graphs $\mathcal{C}_{\mathcal{N}}$ and G are equal. This completes the proof of the theorem.

We end this section with a brief discussion on solving AddTaxa in reasonable time despite its NP-completeness. For an HGT network $\mathcal{N}$, the NP-completeness proof provided an algorithm for solving AddTaxa$(\mathcal{N}, k)$. In particular, construct the critical graph of $\mathcal{N}$ and use an exact algorithm for FeedbackVertexSet$(\mathcal{C}_{\mathcal{N}}, k)$. For hybridization networks, we can use the same approach as the analogous results for HGT networks also hold in this setting if we define the *critical graph* of a hybridization network $\mathcal{N}$, again denoted by $\mathcal{C}_{\mathcal{N}}$, as follows. The vertex set of $\mathcal{C}_{\mathcal{N}}$ is the set of critical vertices of $\mathcal{N}$ and the arc set of $\mathcal{C}_{\mathcal{N}}$ is

$$\{(u, v) : u < v_1, \text{ where } (v, v_1) \text{ is a hybridization arc in } \mathcal{N}\}.$$

With this definition of the critical graph for hybridization networks, it is straightforward to check that Lemma 32 and Proposition 33 hold for hybridization networks. For the analogous proof of Proposition 33, note that, in the context of hybridization networks, if v_1 is a hybridization vertex with parents v and v', and both v and v' are in the critical graph, then (u, v) is an arc in $\mathcal{C}_{\mathcal{N}}$ if and only if (u, v') is an arc in $\mathcal{C}_{\mathcal{N}}$. Thus v and v' can be labeled appropriately, similar to the procedure described in the first part of the proof of this proposition for labeling an HGT network. Since Lemma 32 and Proposition 33 hold for hybridization networks, Corollary 34 holds for hybridization networks. Hence, one can also solve AddTaxa$(\mathcal{N}, k)$ for when $\mathcal{N}$ is a hybridization network by constructing $\mathcal{C}_{\mathcal{N}}$ and using an exact algorithm for FeedbackVertexSet$(\mathcal{C}_{\mathcal{N}}, k)$.

It is shown in [5] that FeedbackVertexSet for directed graphs is fixed-parameter tractable. The authors provide an algorithm which solves FeedbackVertexSet(G, k) in $O(4^k k! n^{O(1)})$ where n is the number of vertices in G. Thus if k is relatively small, this algorithm will work reasonably quickly in practice regardless of the size of G.

For a reticulation network $\mathcal{N}$, the size of $\mathcal{C}_{\mathcal{N}}$ is determined by the number of critical vertices of $\mathcal{N}$. This number is less than the number of reticulation vertices and we would expect this latter number to be much less than the size of $\mathcal{N}$. Consequently, k may typically be relatively small and so the algorithm in [5] should work well in helping to find the solution for many instances of AddTaxa$(\mathcal{N}, k)$.

4 Constructing temporal hybridization networks for two trees

In the previous section, we determined the smallest number of taxa that must be added to a reticulation network so that the resulting network is temporal. However, for many evolutionary studies, we are initially given a set of gene trees rather than a reticulation network. In such a case, in particular, when there are no unsampled taxa, it is of importance to decide whether a temporal reticulation network exists that simultaneously explains the evolutionary histories of the gene trees under consideration. An example showing that the existence of such a network is not guaranteed is given after some preliminaries.

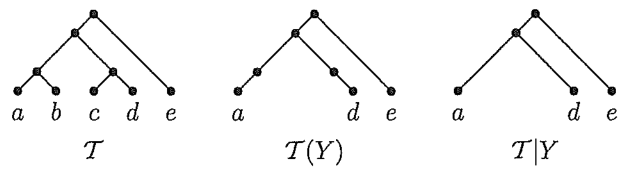

Fig. 7 A rooted binary phylogenetic X-tree $\mathcal{T}$, and the two subtrees $\mathcal{T}(Y)$ and $\mathcal{T}|Y$ with $Y = \{a, d, e\}$.

In this section, we analyze the construction of temporal hybridization networks from so-called agreement forests—such forests are frequently used to model reticulate evolution for when two rooted binary phylogenetic trees are given, see for example [1,4,9]. Our analysis centers around a new algorithm called TEMPORALHYBRID which reconstructs a temporal hybridization network of two rooted binary phylogenetic trees if one exists. Two applications of the algorithm are given at the end of this section. Throughout this section, we restrict our attention to hybridization networks.

4.1 The hybridization number

Before detailing TEMPORALHYBRID, we need some additional definitions.

Hybrid phylogenies and the hybridization number. A *rooted binary phylogenetic X-tree* $\mathcal{T}$ is a rooted tree whose root has degree two and all other internal vertices have degree three, and whose leaf set is X. The set X is called the *label set* of $\mathcal{T}$ and is denoted by $\mathcal{L}(\mathcal{T})$. For a subset A of X, the minimal rooted subtree of $\mathcal{T}$ that connects all the elements in A is denoted by $\mathcal{T}(A)$. Furthermore, the *restriction of* $\mathcal{T}$ *to* A, denoted by $\mathcal{T}|A$, is the rooted phylogenetic tree obtained from $\mathcal{T}(A)$ by contracting every vertex of degree two apart from the root. An example of both types of subtrees is shown in Figure 7.

Let $\mathcal{T}$ be a rooted binary phylogenetic X'-tree, and let $\mathcal{N}$ be a hybridization network with label set X, where $X' \subseteq X$. We say that $\mathcal{N}$ *displays* $\mathcal{T}$ if all of the ancestral relationships described in $\mathcal{T}$ are preserved by $\mathcal{N}$. Formally, $\mathcal{N}$ displays $\mathcal{T}$ if $\mathcal{T}$ can be obtained from $\mathcal{N}$ by deleting a subset of arcs and vertices of $\mathcal{N}$, and contracting any resulting degree-2 vertices apart from the root.

A fundamental problem for biologists studying a set of present-day species whose evolutionary history includes hybridization is to determine the extent to which hybridization has influenced their past. For two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$, a common way to quantify this extent is by determining the value

$$h(\mathcal{T}, \mathcal{T}') = \min\{h(\mathcal{N}) : \mathcal{N} \text{ is a hybridization network on } X \text{ that displays } \mathcal{T} \text{ and } \mathcal{T}'\},$$

where $h(\mathcal{N})$ is the number of hybridization vertices of $\mathcal{N}$. However, Bordewich and Semple [4] showed that this determination is an NP-hard problem.

We next give an example of two trees which cannot be displayed in a temporal hybridization network. Consider the two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$ shown at the top of Figure 8. It is easily seen that every hybridization network that displays $\mathcal{T}$ and $\mathcal{T}'$ has at least two reticulation vertices. There are exactly nine hybridization networks on X, each with two hybridization vertices, displaying $\mathcal{T}$ and $\mathcal{T}'$. These networks are shown in the bottom part of Figure 8. None of these networks is temporal. Moreover, a straightforward check shows that any hybridization network that displays $\mathcal{T}$ and $\mathcal{T}'$ with more than two hybridization vertices is not temporal either. Thus, there is no temporal hybridization network on X that displays $\mathcal{T}$ and $\mathcal{T}'$.

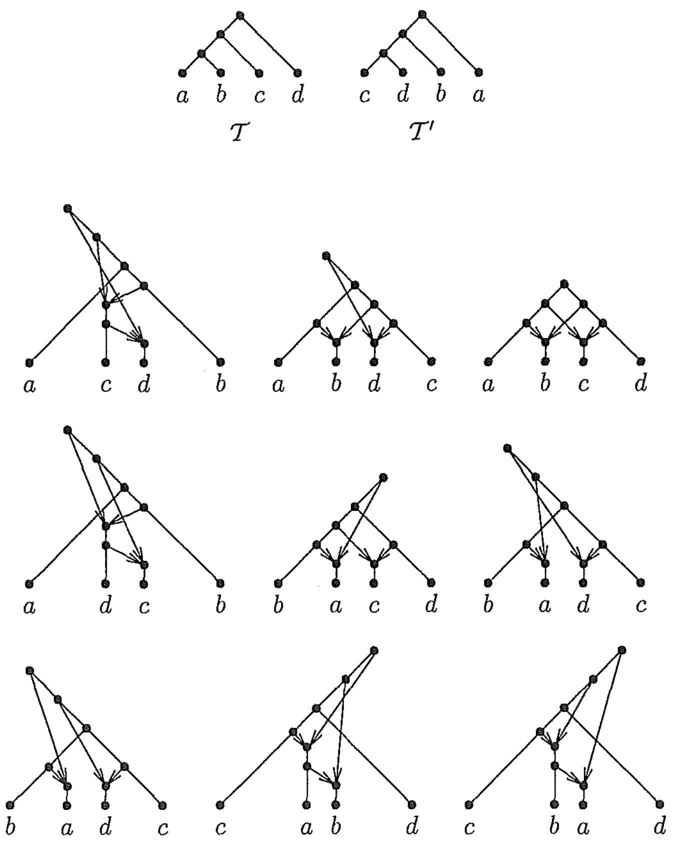

Fig. 8 Two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$, and all nine hybridization networks on X displaying both trees with two hybridization vertices. None of the hybridization networks is temporal.

Agreement Forests. The approach taken by TEMPORALHYBRID (see Section 4.2) is based on the following characterization. Loosely speaking, for two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$, this characterization equates $h(\mathcal{T}, \mathcal{T}')$ with the smallest size of a so-called acyclic-agreement forest for $\mathcal{T}$ and $\mathcal{T}'$. We next introduce agreement forests and make this characterization precise.

Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees. For the upcoming definitions, we regard the roots of both $\mathcal{T}$ and $\mathcal{T}'$ as an extra vertex ρ adjoined to the original root by an additional edge and $\mathcal{L}(\mathcal{T}) = \mathcal{L}(\mathcal{T}') = X \cup \{\rho\}$. An *agreement forest* for $\mathcal{T}$ and $\mathcal{T}'$ is a collection $\mathcal{F} = \{\mathcal{S}_\rho, \mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k\}$ of rooted leaf-labeled trees, where $\mathcal{S}_\rho$ is a rooted tree whose label set $\mathcal{L}_\rho$ contains ρ and $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k$ are rooted binary phylogenetic trees with label sets $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_k$, respectively, such that the following properties hold:

(i) The label sets $\mathcal{L}_\rho, \mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_k$ partition $X \cup \{\rho\}$.
(ii) For all $i \in \{\rho, 1, 2, \ldots, k\}$, $\mathcal{S}_i \cong \mathcal{T}|\mathcal{L}_i$ and $\mathcal{S}_i \cong \mathcal{T}'|\mathcal{L}_i$.
(iii) The trees in $\{\mathcal{T}(\mathcal{L}_i) : i \in \{\rho, 1, 2, \ldots, k\}\}$ and $\{\mathcal{T}'(\mathcal{L}_i) : i \in \{\rho, 1, 2, \ldots, k\}\}$ are vertex-disjoint rooted subtrees of $\mathcal{T}$ and $\mathcal{T}'$, respectively.

We sometimes refer to the component $\mathcal{S}_\rho$ of $\mathcal{F}$ as the *root component* of $\mathcal{F}$. To illustrate, consider the two trees $\mathcal{T}$ and $\mathcal{T}'$ in Figure 8. Viewing the root of each of $\mathcal{T}$ and $\mathcal{T}'$ as an extra vertex ρ adjoined to the original root as described above, the restrictions of $\mathcal{T}$ (and $\mathcal{T}'$) to the label sets $\{\rho, a, d\}$, $\{b\}$, and $\{c\}$, respectively, are the components of an agreement forest of $\mathcal{T}$ and $\mathcal{T}'$.

To make the connection between hybridization networks and agreement forests, we need to extend the definition of an agreement forest. This extension allows for the biological constraint that species cannot inherit genetic material from their own descendants. Let $G_{\mathcal{F}}$ be the directed graph whose vertex set is $\mathcal{F}$ and for which $(\mathcal{S}_i, \mathcal{S}_j)$ is an arc precisely if $i \neq j$ and

(I) the root of $\mathcal{T}(\mathcal{L}_i)$ is an ancestor of the root of $\mathcal{T}(\mathcal{L}_j)$ or
(II) the root of $\mathcal{T}'(\mathcal{L}_i)$ is an ancestor of the root of $\mathcal{T}'(\mathcal{L}_j)$.

We say that $\mathcal{F}$ is *acyclic* precisely if $G_{\mathcal{F}}$ is acyclic. If $\mathcal{F}$ is acyclic and it has the smallest number of components amongst all such forests of $\mathcal{T}$ and $\mathcal{T}'$, then $\mathcal{F}$ is a *maximum-acyclic-agreement forest* of $\mathcal{T}$ and $\mathcal{T}'$, in which case we denote $|\mathcal{F}| - 1$ by $m_a(\mathcal{T}, \mathcal{T}')$.

The minimum number $h(\mathcal{T}, \mathcal{T}')$ of hybridization events and the size of a maximum-acyclic-agreement forest of $\mathcal{T}$ and $\mathcal{T}'$ are related through the following theorem [1].

Theorem 41 *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees. Then*

$$h(\mathcal{T}, \mathcal{T}') = m_a(\mathcal{T}, \mathcal{T}').$$

For an acyclic-agreement forest $\mathcal{F}$, a tuple $\mathcal{O} = (\mathcal{S}_\rho, \mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k)$ is an *ordering* of the components of $\mathcal{F}$ if, for each i, the vertex $\mathcal{S}_i$ has indegree 0 in the graph obtained from $G_{\mathcal{F}}$ by deleting the vertices $\mathcal{S}_\rho, \mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_{i-1}$ and their incident arcs. Since $\mathcal{F}$ is acyclic, such an ordering always exists.

Although not explicitly stated in [1], one direction of Theorem 41 is essentially established by proving that, if $\mathcal{N}$ is a hybridization network on X that displays $\mathcal{T}$ and $\mathcal{T}'$, then there is an acyclic-agreement forest $\mathcal{F}$ for $\mathcal{T}$ and $\mathcal{T}'$ such that $|\mathcal{F}| \leq h(\mathcal{N}) + 1$. Intuitively, one takes $\mathcal{N}$ and iteratively cuts off rooted subtrees by deleting hybridization vertices and their three incident arcs. By viewing the root of $\mathcal{N}$ as a vertex at the end of a pendant arc adjoined to the original root, we obtain an acyclic-agreement forest $\mathcal{F}$ of $\mathcal{T}$ and $\mathcal{T}'$, and so $|\mathcal{F}| - 1 \leq h(\mathcal{N})$. Now, if we extend their argument and suppose that $\mathcal{N}$ is temporal, then we can obtain an acyclic-agreement forest for $\mathcal{T}$ and $\mathcal{T}'$ as follows. First, select a vertex v that either (i) is a hybridization vertex and, except for v itself, its parents have no hybridization vertex as a descendant or (ii) is a tree vertex whose parent is a tree vertex that has no hybridization vertex as a descendant. Now delete v and its three incident arcs if v is a hybridization vertex or delete the parent arc of v if v is a tree vertex, and contract any non-root degree-2 vertex. Repeating this process so that every hybridization vertex is selected, and then reversing the order of the selections, we eventually obtain an ordering of an acyclic-agreement forest for $\mathcal{T}$ and $\mathcal{T}'$. Given an ordering $\mathcal{O}$ of an arbitrary acyclic-agreement forest $\mathcal{F}$ of two rooted binary phylogenetic X-trees, we say that a temporal hybridization network $\mathcal{N}$ *preserves* $\mathcal{O}$ if $\mathcal{O}$ can be obtained from $\mathcal{N}$ in the above way.

4.2 The algorithm TemporalHybrid

In this section, we present the algorithm TemporalHybrid. This algorithm takes two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$, and an ordering $\mathcal{O}$ of an acyclic-agreement forest $\mathcal{F}$ of $\mathcal{T}$ and $\mathcal{T}'$ as input, and outputs either

(i) a temporal hybridization network on X that displays $\mathcal{T}$ and $\mathcal{T}'$, and preserves $\mathcal{O}$, or
(ii) a statement indicating that there is no such network.

Baroni *et al.* [2], have previously presented a similar algorithm. Called HybridPhylogeny, this algorithm has the same input as TemporalHybrid but without an ordering $\mathcal{O}$ of $\mathcal{F}$. The task for HybridPhylogeny is simply to construct one of potentially many hybridization networks that display $\mathcal{T}$ and $\mathcal{T}'$. However, in doing so, there is no guarantee that the resulting network is temporal. The correctness of TemporalHybrid is established after some remarks following the description of the algorithm.

Algorithm: TemporalHybrid$(\mathcal{T}, \mathcal{T}', \mathcal{O})$
Input: Two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$, and an ordering $\mathcal{O} = (\mathcal{S}_\rho, \mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_k)$ of an acyclic-agreement forest $\mathcal{F}$ for $\mathcal{T}$ and $\mathcal{T}'$.
Output: A temporal hybridization network on X with at most k hybridization vertices that displays $\mathcal{T}$ and $\mathcal{T}'$ and preserves $\mathcal{O}$, or a statement indicating that there is no such network.

```
1. For each $i \in \{1, \dots, k\}$, regard $\rho_i$ as the label of an extra vertex adjoined to the original root of $\mathcal{S}_i$
   by an additional edge.
2. Set $\mathcal{N}_0 = \mathcal{S}_\rho$, and set $i = 1$.
3. Attach $\mathcal{S}_i$ to $\mathcal{N}_{i-1}$ via at most two new arcs. Each new arc joins the vertex labeled $\rho_i$ to a new vertex
   which subdivides an arc of $\mathcal{N}_{i-1}$. The subdivided arcs $e$ and $e'$ are chosen so that
   (i) the resulting network displays $\mathcal{T}|(\mathcal{L}(\mathcal{N}_{i-1}) \cup \mathcal{L}(\mathcal{S}_i))$ and $\mathcal{T}'|(\mathcal{L}(\mathcal{N}_{i-1}) \cup \mathcal{L}(\mathcal{S}_i))$, where $\mathcal{L}(\mathcal{N}_{i-1}) =$
       $\mathcal{L}(\mathcal{S}_\rho) \cup \mathcal{L}(\mathcal{S}_1) \cup \dots \cup \mathcal{L}(\mathcal{S}_{i-1})$, and
   (ii) no element of $\{\rho_1, \rho_2, \dots, \rho_{i-1}\}$ is a descendant of either $e$ or $e'$ and, if $e \neq e'$, then $e$ and $e'$ are
       not on the same path from the root of $\mathcal{N}_{i-1}$ to one of its leaves.
   Set $\mathcal{N}_i$ to be the resulting network. If there is no such attachment for $\mathcal{S}_i$, then stop and return *there
   is no temporal hybridization network on $X$ that displays $\mathcal{T}$ and $\mathcal{T}'$, and preserves $\mathcal{O}$.*
4. If $i = k$, remove the arc incident with the vertex labeled $\rho$ and remove all labels in $\{\rho, \rho_1, \rho_2, \dots, \rho_k\}$
   from $\mathcal{N}_k$, contract any resulting degree-0 and degree-2 vertices apart from the root, and return the
   obtained network. If $i < k$, increment $i$ by 1 and return to Step 3.
```

Remarks.

1. If $\mathcal{N}_0$ consists of an isolated vertex labeled ρ, a slight complication arises in Step 3 of the algorithm since no arc can be subdivided by a new vertex. In this case, $\mathcal{S}_1$ is adjoined to $\mathcal{N}_0$ by adding precisely one new arc that joins the vertex labeled ρ_1 with the vertex labeled ρ.
2. For all $\mathcal{S}_i \in \mathcal{F}$, the algorithm TemporalHybrid potentially checks each arc of $\mathcal{N}_{i-1}$ to decide whether $\mathcal{S}_i$ can be appropriately attached to $\mathcal{N}_{i-1}$. Thus, the running time of the algorithm is at most $O(kn)$, where $n = |X|$.
3. If $\mathcal{F}$ is a maximum-acyclic-agreement forest, then the attachment of a new component $\mathcal{S}_i$ to $\mathcal{N}_{i-1}$ always requires two new arcs in Step 3 of the algorithm; otherwise, $\mathcal{F}$ is not optimal.
4. If TemporalHybrid returns a temporal hybridization network, then this is the unique such network on X that displays $\mathcal{T}$ and $\mathcal{T}'$, and preserves $\mathcal{O}$ (see Theorem 42).

We next prove the correctness of TemporalHybrid. In doing this, we additionally show that if the algorithm returns a hybridization network, then this network is unique relative to the ordering of $\mathcal{O}$.

Theorem 42 *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees, and let $\mathcal{O}$ be an ordering of an acyclic-agreement forest $\mathcal{F}$ for $\mathcal{T}$ and $\mathcal{T}'$ with $|\mathcal{F}| - 1 = k$. Then*

(i) TemporalHybrid *returns a temporal hybridization network $\mathcal{N}$ that displays $\mathcal{T}$ and $\mathcal{T}'$ with $h(\mathcal{N}) \leq k$ and that preserves $\mathcal{O}$ if there exists such a network, or*

(ii) TEMPORALHYBRID *returns a statement indicating that there is no such network.*

Moreover, if TEMPORALHYBRID *returns a temporal hybridization network, then, up to isomorphism, this is the unique network with the properties in (i).*

Proof Without loss of generality, let $\mathcal{O} = (\mathcal{S}_\rho, \mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_k)$. The proof is by induction on k. If $k = 0$, then $\mathcal{S}_\rho \cong \mathcal{T} \cong \mathcal{T}'$ and the theorem trivially holds. Now assume that $k > 0$ and that the theorem holds for all orderings of all acyclic-agreement forests with at most k components of two rooted binary phylogenetic trees. Let $\mathcal{T}_1$ be the rooted binary phylogenetic tree $\mathcal{T}|(X - \mathcal{L}(\mathcal{S}_k))$, and let $\mathcal{T}_1'$ be the rooted binary phylogenetic tree $\mathcal{T}'|(X - \mathcal{L}(\mathcal{S}_k))$. Since $\mathcal{S}_k$ is the last coordinate in $\mathcal{O}$, the trees $\mathcal{T}_1$ and $\mathcal{T}_1'$ can also be obtained from $\mathcal{T}$ and $\mathcal{T}'$, respectively, by deleting a single edge and contracting the resulting degree-2 vertex. Let $\mathcal{F}_1 = \mathcal{F} - \{\mathcal{S}_k\}$. As $\mathcal{O}$ is an ordering of $\mathcal{F}$, it follows that $\mathcal{O}_1 = (\mathcal{S}_\rho, \mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_{k-1})$ is an ordering of $\mathcal{F}_1$. Now observe that the workings of TEMPORALHYBRID applied to $(\mathcal{T}_1, \mathcal{T}_1', \mathcal{O}_1)$ and applied to $(\mathcal{T}, \mathcal{T}', \mathcal{O})$ up to considering $\mathcal{S}_k$ are identical. This observation is used in the rest of the proof.

Since $|\mathcal{O}_1| < |\mathcal{O}|$, it follows by the induction assumption that TEMPORALHYBRID$(\mathcal{T}_1, \mathcal{T}_1', \mathcal{O}_1)$ either returns, up to isomorphism, a unique temporal hybridization network, $\mathcal{N}_{k-1}$ say, that displays $\mathcal{T}_1$ and $\mathcal{T}_1'$, and preserves $\mathcal{O}_1$ or returns a statement indicating that there is no such network. First suppose that TEMPORALHYBRID$(\mathcal{T}_1, \mathcal{T}_1', \mathcal{O}_1)$ returns the latter. If there is a temporal hybridization network that displays $\mathcal{T}$ and $\mathcal{T}'$, and preserves $\mathcal{O}$, let v be the vertex of $\mathcal{N}$ that corresponds to the root of $\mathcal{S}_k$. Then by deleting v and its incident arcs or the parent arc of v, depending on whether v is a hybridization vertex or a tree vertex, respectively, and contracting any resulting degree-2 vertices apart from the root, we have a temporal hybridization network that displays $\mathcal{T}_1$ and $\mathcal{T}_1'$, and preserves $\mathcal{O}_1$; a contradiction. Thus, if TEMPORALHYBRID$(\mathcal{T}, \mathcal{T}', \mathcal{O})$ returns a statement that there is no such network prior to considering $\mathcal{S}_k$, then it returns correctly. Therefore, we may suppose that TEMPORALHYBRID$(\mathcal{T}_1, \mathcal{T}_1', \mathcal{O}_1)$ returns $\mathcal{N}_{k-1}$. By the observation at the end of the last paragraph, this means that $\mathcal{N}_{k-1}$ is constructed immediately prior to considering $\mathcal{S}_k$ in TEMPORALHYBRID$(\mathcal{T}, \mathcal{T}', \mathcal{O})$.

If TEMPORALHYBRID$(\mathcal{T}, \mathcal{T}', \mathcal{O})$ returns a statement indicating that there is no appropriate network, then, because of the uniqueness of $\mathcal{N}_{k-1}$ and the fact that (i) and (ii) in Step 3 of the algorithm are necessary conditions for the placement of $\mathcal{S}_k$, the algorithm performs correctly. On the other hand, if TEMPORALHYBRID$(\mathcal{T}, \mathcal{T}', \mathcal{O})$ returns a network $\mathcal{N}$, it follows by (i) and (ii) in Step 3 that $\mathcal{N}$ displays $\mathcal{T}$ and $\mathcal{T}'$, and $\mathcal{N}$ is temporal and preserves $\mathcal{O}$. Furthermore, let e and e' denote the arcs of $\mathcal{N}_{k-1}$ that are subdivided in the attachment of $\mathcal{S}_k$. By (ii) in Step 3, no hybridization vertex is a descendant of either e or e', so the choice of e and e' is unique. By the uniqueness of $\mathcal{N}_{k-1}$, it follows that $\mathcal{N}$ is the unique temporal hybridization network that displays $\mathcal{T}$ and $\mathcal{T}'$, and preserves $\mathcal{O}$. Note that this argument also holds if $e = e'$ in which case $\mathcal{S}_k$ is attached by a single arc. This completes the proof of the theorem.

We saw earlier that, for two rooted binary phylogenetic X-trees, there may not exist a temporal hybridization network on X that displays both trees. In general, how does one decide such an outcome for two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$? Of course, by considering all orderings of all acyclic-agreement forests for $\mathcal{T}$ and $\mathcal{T}'$, we can repeatedly use TEMPORALHYBRID to decide whether or not such a hybridization network exists. While still exponential in time, we can do much better than this as shown by the next proposition.

For two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$, an acyclic-agreement forest $\mathcal{F} = \{\mathcal{S}_\rho, \mathcal{S}_1, \dots, \mathcal{S}_k\}$ of $\mathcal{T}$ and $\mathcal{T}'$ is *trivial* if $|\mathcal{L}(\mathcal{S}_\rho)| = 3$ and each of $\mathcal{S}_1, \dots, \mathcal{S}_k$ consists of an isolated vertex. Provided $\mathcal{S}_\rho$ is the first coordinate, any $(k+1)$−tuple of $\mathcal{F}$ is an ordering of $\mathcal{F}$. The next proposition shows that it is sufficient to consider each such ordering of $\mathcal{F}$ to decide whether or not there exists a temporal hybridization network that displays $\mathcal{T}$ and $\mathcal{T}'$.

Proposition 43 *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees, and let $|X| = n$. Deciding whether there is a temporal hybridization network on X that displays $\mathcal{T}$ and $\mathcal{T}'$ takes at most time $O(n! \cdot p(a))$, where $p(a)$ is the polynomial running time of the algorithm* TEMPORALHYBRID.

Proof Suppose there is a temporal hybridization network $\mathcal{N}$ on X that displays $\mathcal{T}$ and $\mathcal{T}'$. Let $X = \{a_1, a_2, \ldots, a_n\}$. We next construct an ordering of a trivial forest of $\mathcal{T}$ and $\mathcal{T}'$, that is preserved by $\mathcal{N}$. Select an element of X so that its parent is a tree vertex and does not have a hybridization vertex as a descendant. If this is not possible, then, as $\mathcal{N}$ is acyclic, there exists a hybridization vertex whose child is an element of X and whose parents have no hybridization vertex as a descendant. Without loss of generality, we may assume that the selected element is a_n. If the parent of a_n is a tree vertex, then delete the parent arc of a_n and contract the resulting degree-2 vertex. Since the parent of a_n has no hybridization vertex as a descendant, the resulting network is again a hybridization network. If the parent of a_n is a hybridization vertex, then delete the parent vertex of a_n and its three incident arcs, and contract the resulting degree-2 vertices. By property (iv) in the definition of a reticulation network, it is easily checked that the arcs incident with the contracted degree-2 vertices are tree arcs, and thus, the resulting network is again a hybridization network. Furthermore, in both cases, the resulting hybridization network is temporal as $\mathcal{N}$ is temporal. Selecting another vertex and continuing in this way, we eventually select (in order) the vertices $a_n, a_{n-1}, \ldots, a_3$ say. This gives a trivial forest $\mathcal{F}$ of $\mathcal{T}$ and $\mathcal{T}'$, where the label set of the root component is $\{\rho, a_1, a_2\}$ and the remaining components consist of isolated vertices labeled $a_3, a_4, \ldots, a_n$. Furthermore, the tuple beginning with the root component and followed (in order) by the components whose label sets are $\{a_3\}, \{a_4\}, \ldots, \{a_n\}$ is an ordering $\mathcal{O}$ of $\mathcal{F}$ and, by construction, $\mathcal{N}$ preserves $\mathcal{O}$. Now, by Theorem 42, calling TEMPORALHYBRID$(\mathcal{T}, \mathcal{T}', \mathcal{O})$ returns $\mathcal{N}$. Thus to decide whether or not there is a temporal hybridization network on X that displays $\mathcal{T}$ and $\mathcal{T}'$, it suffices to consider all possible acyclic-agreement forests of $\mathcal{T}$ and $\mathcal{T}'$ that are trivial. Since there are $\binom{n}{2} \cdot (n-2)! = \frac{1}{2}n!$ such forests, the proposition now follows.

While the above approach is not fast because of the number of orderings to consider, we can do much better in practice if we restrict our attention to maximum-acyclic-agreement forests. We do this in the next section.

4.3 Minimal temporal hybridization networks

To provide an indication of the significance of hybridization, biologists are often interested in reconstructing (temporal) hybridization networks that explain the ancestral history of the species under consideration and simultaneously minimize the number of hybridization events. This minimum number provides a lower bound on the number of such events, thus it gives an indication of the role that hybridization has had on the evolution of the present-day species. In this section, we consider an approach to this task with the view of constructing hybridization networks that are temporal.

Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted phylogenetic X-trees. A (temporal) hybridization network $\mathcal{N}$ that displays $\mathcal{T}$ and $\mathcal{T}'$ is *minimal* if $h(\mathcal{N}) = m_a(\mathcal{T}, \mathcal{T}')$. Since we are using a combinatorial framework to calculate the number of hybridization events, it is likely that there exist several maximum-acyclic-agreement forests for $\mathcal{T}$ and $\mathcal{T}'$. For example, for the grass (*Poaceae*) data set that has been analyzed in [3], the associated gene loci for 12 gene tree pairs, the minimum number of hybridization events, and the number of maximum-acyclic-agreement forests are given in Table 1. This data set was originally provided by [8] and contains DNA sequences for the six genetic loci *ndhF*, *phyB*, *rbcL*, *rpoC2*, *waxy*, and *ITS*. For each locus, up to 65 taxa were sequenced and a maximum likelihood gene tree was reconstructed (for details, see [3] and references therein).

Table 1 Results for the *Poaceae* data set.

Pairwise combination		Hybridization number $h(\mathcal{T}, \mathcal{T}')$	# MAAFs[a]	#MAAFs with a proper root component
ndhF	*phyB*	14	2268	0
ndhF	*rbcL*	13	48	0
ndhF	*rpoC2*	12	27	0
ndhF	*waxy*	9	396	18
phyB	*rbcL*	4	4	4
phyB	*rpoC2*	7	1	0
phyB	*waxy*	3	6	6
phyB	*ITS*	8	9	9
rbcL	*rpoC2*	13	9	0
rbcL	*waxy*	7	35	0
rpoC2	*waxy*	1	1	1
waxy	*ITS*	8	18	0

[a] Abbreviation for maximum-acyclic-agreement forests.

A natural way to decide whether there exists a minimal temporal hybridization network for two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$ is to apply TEMPORALHYBRID to all orderings of each maximum-acyclic-agreement forest for $\mathcal{T}$ and $\mathcal{T}'$. However, since the number of maximum-acyclic-agreement forests can still be quite large (see Table 1), we next establish a quick test that significantly reduces the number of such forests that one needs to consider.

To describe the test, we need one further definition. Let $\mathcal{F}$ be an acyclic-agreement forest for two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$, and let $\mathcal{S}_\rho$ be the root component of $\mathcal{F}$. If the roots of $\mathcal{T}(\mathcal{L}_\rho - \{\rho\})$ and $\mathcal{T}'(\mathcal{L}_\rho - \{\rho\})$ coincide with the original roots of $\mathcal{T}$ and $\mathcal{T}'$, respectively, $\mathcal{S}_\rho$ is said to be *proper*.

Proposition 44 *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees, and let $\mathcal{F}$ be a maximum-acyclic-agreement forest of $\mathcal{T}$ and $\mathcal{T}'$ with root component $\mathcal{S}_\rho$. Let $\mathcal{O}$ be an ordering of $\mathcal{F}$. If $\mathcal{S}_\rho$ is not proper, then* TEMPORALHYBRID *applied to $(\mathcal{T}, \mathcal{T}', \mathcal{O})$ returns a statement indicating that there is no temporal hybridization network on X that displays $\mathcal{T}$ and $\mathcal{T}'$, and preserves $\mathcal{O}$. In particular, there is no minimal temporal hybridization network on X that displays $\mathcal{T}$ and $\mathcal{T}'$, and preserves $\mathcal{O}$.*

Proof Suppose that $\mathcal{S}_\rho$ is not proper, and consider TEMPORALHYBRID applied to $(\mathcal{T}, \mathcal{T}', \mathcal{O})$. If this application returns a hybridization network $\mathcal{N}$, then, as $\mathcal{F}$ is maximum, $\mathcal{N}$ has exactly $|\mathcal{F}|-1$ hybridization vertices and so, at each iteration i, two (distinct) new arcs are used to adjoin $\mathcal{S}_i$ to $\mathcal{N}_{i-1}$. Now, at some iteration i, the component $\mathcal{S}_i$ gets adjoined to $\mathcal{N}_{i-1}$ via an arc that is incident with a new vertex that subdivides the arc ending in the vertex labeled ρ. But this mean that wherever the second new arc is placed to adjoin $\mathcal{S}_i$ to $\mathcal{N}_{i-1}$ we contradict (ii) in Step 3. Thus, by Theorem 42, there is no temporal hybridization network on X that displays $\mathcal{T}$ and $\mathcal{T}'$, and preserves $\mathcal{O}$. The proposition now follows.

By checking which maximum-acyclic-agreement forests for a given pair of gene trees have a proper root component, the number of such forests that can yield a minimal temporal hybridization network can be reduced significantly. For example, in reference to Table 1, the highest number of maximum-acyclic-agreement forests for a pair of trees is 2268. However, as shown in the last column, none of these forests has a proper root component. Hence, for the first pair of gene trees (*ndhF* and *phyB*) in this table, there is no minimal temporal hybridization network that displays the two trees. Moreover, for the 12 analyzed gene tree pairs of the grass data set, at most 18 maximum-acyclic-agreement forests need to be checked in order to determine whether any associated ordering leads to a minimal temporal hybridization network for the gene trees under consideration. In general, we can check in time $O(rk! \cdot p(a))$ if a minimal temporal hybridization network exists, where r is the number of maximum-acyclic-agreement forests (with a proper root component) for a pair of trees, each such forest consists of $k+1$ components, and $p(a)$ is the polynomial running time of the algorithm TEMPORALHYBRID. Note that $k!$ is an upper bound on the number of orderings that are associated with a maximum-acyclic-agreement forest containing $k+1$ components. Furthermore, we are assuming here that we have the list of maximum-acyclic-agreement forests with a proper root component. Such a list can be found by using the recently implemented extended version (unpublished) of the fixed-parameter algorithm HYBRIDINTERLEAVE [6, 7]. Despite the theoretical exponential time of calculating this list, the practical running times presented in [7] essentially show that computing maximum-acyclic-agreement forests is remarkably quick for many biological instances.

5 Summary

In the first part of this paper, we showed that ADDTAXA—the decision problem associated with determining the minimum number of taxa to add to a reticulation network so that the resulting network has a temporal representation—is an NP-complete problem. However, in establishing the result, this determination comes down to finding the minimum number of vertices to delete so that the associated critical graph is acyclic. In practice, we expect this graph to be relatively small for many biological instances and thus even brute-force algorithms might be feasible.

In the second part of this paper, we presented the polynomial-time algorithm TEMPORALHYBRID. This algorithm takes as input two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$ and an ordering $\mathcal{O}$ of an associated acyclic-agreement forest, and outputs a temporal hybridization network that displays $\mathcal{T}$ and $\mathcal{T}'$ and preserves $\mathcal{O}$, or the statement that no such network exists. As many biological studies consider the reconstruction of minimal hybridization networks to provide an indication of the significance of hybridization in evolution, we focused our attention to the potentially exponential-time task of finding a temporal hybridization network whose number of hybridization vertices is minimized in the last section. By using a simple and quick check prior to the application of TEMPORALHYBRID, we showed that—applied to a grass data set—the number of maximum-acyclic-agreement forests that need to be considered for finding a minimal temporal hybridization network is significantly reduced and that most inferred minimal hybridization networks are not temporal.

6 Acknowledgements

S.L. was supported by NSF grants SEI-BIO 0513910 and IIS-0803564, and the New Zealand Marsden Fund. C.S. thanks the New Zealand Marsden Fund for supporting this work. T.S. was funded by the Deutsche Forschungsgemeinschaft through the graduate program "Angewandte Algorithmische Mathematik" at the Munich University of Technology. All authors thank the Allan Wilson Centre for Molecular Ecology and Evolution for its support.

References

1. M. Baroni, S. Grünewald, V. Moulton, and C. Semple. Bounding the number of hybridization events for a consistent evolutionary history. Journal of Mathematical Biology, 51:171–182, 2005.

2. M. Baroni, C. Semple, and M. Steel. Hybrids in real time. Systematic Biology, 44:46–56, 2006.
3. M. Bordewich, S. Linz, K. John, and C. Semple. A reduction algorithm for computing the hybridization number of two trees. Evolutionary Bioinformatics, 3:86–98, 2007.
4. M. Bordewich and C. Semple. Computing the minimum number of hybridization events for a consistent evolutionary history. Discrete Applied Mathematics, 155:914–928, 2007.
5. J. Chen, Y. Liu, and S. Lu. A fixed-parameter algorithm for the directed feedback vertex set problem. In: Proceedings of the Fourtieth Annual ACM Symposium on Theory of Computing, pages 177–186, 2008.
6. J. Collins. Rekernelisation Algorithms in Hybrid Phylogenies. MSc Thesis, University of Canterbury, Christchurch, New Zealand, 2009.
7. J. Collins, S. Linz, and C. Semple. Quantifying hybridization in realistic time, submitted.
8. Grass Phylogeny Working Group. Phylogeny and subfamilial classification of the grasses *Poaceae*. Annals of the Missouri Botanical Garden, 88:373–457, 2001.
9. J. Hein, T. Jiang, L. Wang, and K. Zhang. On the complexity of comparing evolutionary trees. Discrete Applied Mathematics, 71:153–169, 1996.
10. G. Jin, L. Nakhleh, S. Snir, and T. Tuller. Efficient parsimony-based methods for phylogenetic network reconstruction. Bioinformatics, 23:e123–e128, 2007.
11. R. M. Karp. Reducibility among combinatorial problems. In: Complexity of Computer Computations, pages 85–103. Plenum Press, 1972.
12. W. Maddison. Gene trees in species trees. Systematic Biology, 46:523–536, 1997.
13. V. Makarenkov, D. Kevorkov, and P. Legendre. Phylogenetic Network Construction Approaches. In: Applied Mycology and Biotechnology. International Elsevier Series 6, Bioinformatics, pages 61–97. 2006.
14. J. Mallet. Hybridization as an invasion of the genome. Trends in Ecology and Evolution, 20:229–237, 2005.
15. G. Martinsen, T. Whitham, R. Turek, and P. Keim. Hybrid populations selectively filter gene introgression between species. Evolution, 55:1325–1335, 2001.
16. B. M. E. Moret, L. Nakhleh, T. Warnow, C. R. Linder, A. Tholse, A. Padolina, J. Sun, and R. Timme. Phylogenetic networks: modeling, reconstructibility, and accuracy. Transactions on Computational Biology and Bioinformatics, 1:13–23.
17. H. Ochman, J. Lawrence, and E. Groisman. Lateral gene transfer and the nature of bacterial innovation. Nature, 405:299–304, 2000.
18. C. Semple and M. Steel. Phylogenetics. Oxford University Press, 2003.
19. D. Wolf, N. Takebayashi, and L. Rieseberg. Predicting the risk of extinction through hybridization. Conservation Biology, 15:1039–1053, 2001.