

HYBRIDIZATION NETWORKS

*Charles Semple***Abstract**

Reticulate evolution is a fundamental process in the evolution of certain groups of taxa. Consequently, conflicting signals in a data set may not be the result of sampling or modelling errors, but due to the fact that reticulation has played a role in the evolutionary history of the species under consideration. Assuming that our initial data set is correct, a fundamental problem is to compute the minimum number of reticulation events that explains this set. This smallest number sets a lower bound on the number of such events and provides an indication of the extent that reticulation has had on the evolutionary history of a collection of present-day species. In this chapter, we focus our attention on this problem for when the initial set consists of two rooted binary phylogenetic trees. This may seem rather special, but there are several reasons for this. Firstly, the problem is NP-hard even when the initial set consists of two such trees. Secondly, we are interested in finding a general solution rather than one that is restricted in some way. Lastly, the problem for when the initial data set consists of binary sequences can be interpreted as a sequence of two-tree problems.

Referring to the problem of when the initial set consists of two trees, this chapter includes the problem's relationship with the rooted subtree prune and regraft distance, mathematical characterizations of the problem based on agreement forests, reduction-based algorithms for solving the problem exactly, and the problem's connection with a variant of it in which the initial data set consists of binary sequences.

11.1 Introduction

Evolutionary (phylogenetic) trees are used to represent the tree-like evolution of a collection of taxa. For many groups of taxa (for example, most mammals) this representation is appropriate. However, non-tree-like evolutionary processes such as hybridization, horizontal gene transfer, and recombination mean that some groups of taxa are not suited to this type of representation. Collectively referred to as reticulation events, these types of processes result in species being a composite of DNA regions derived from different ancestors. Frequently with bacteria, horizontal gene transfer is the transfer of a piece of DNA from one

organism to another which is not its offspring. On the other hand, hybridizations combine two lineages to create a new offspring. Examples of eukaryotes whose ancestral history include hybridization are certain plant and bird species. Recombination is a type of hybridization that has been well-studied within the framework of population genetics. For informative articles on the frequency of hybridization amongst animals and the problem of distinguishing hybridization from other causes of phylogenetic incongruence, see [36] and [37], respectively.

The effect of reticulation in evolution has been recognized for quite some time. Since the 1930's, botanists suggested that the morphological variation in the New Zealand flora is due to hybridization [2]. More recently, in the context of horizontal gene transfer, Doolittle [15] wrote that "molecular phylogeneticists will have failed to find the 'true tree', not because their methods are inadequate or because they have chosen the wrong genes, but because the history of life cannot be properly represented as a tree." Despite this recognition, mathematical investigations into the understanding and analysis of reticulation in evolution are relatively recent.

In a separate chapter, Huson provides an overview of various ways of representing the evolutionary history of a collection of taxa that has undergone reticulate evolution. In this chapter, we focus our attention on a particular problem that is both biologically important and mathematically challenging. A fundamental problem for biologists studying the evolution of species whose past has included reticulation is the following: given a collection of rooted phylogenetic trees on sets of species that correctly represents the tree-like evolution of different parts of their genomes, what is the smallest number of reticulation events needed to explain the evolution of the species under consideration. As well as providing a lower bound on the number of such events, this smallest number also indicates the extent that reticulation has had on the evolutionary history of the collection of present-day species.

The chapter is organized as follows. In Section 11.2, we formalize the above problem and the notion of a hybridization network, the latter is central to this problem. In general, the problem is NP-hard even when the initial collection consists of two trees. However, there is an attractive and particularly useful characterization of it in this case. This characterization is described in Section 11.3, while Section 11.4 contains algorithmic applications of it. In Section 11.5, we consider the variant of the problem for when the initial collection is a set of binary sequences. The material in this section is used in the subsequent two sections. An important biological consideration of the evolutionary history of taxa is that reticulation events occur between taxa that coexist in time. We investigate this consideration in Section 11.6. Lastly, in Section 11.7, we consider some of the computational issues in computing the above smallest number.

For completeness, we end this section with some preliminaries. Unless otherwise stated, the notation and terminology in this chapter follows Semple and Steel [45].

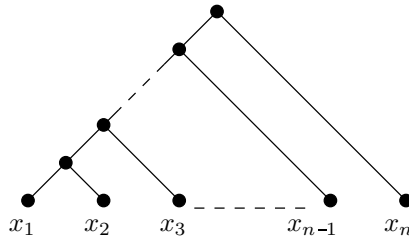


FIG. 11.1. A rooted caterpillar tree.

11.1.1 Preliminaries

A *rooted phylogenetic X -tree* \mathcal{T} is a rooted tree in which no vertex has degree-2 except possibly for the root which has degree at least two, and whose leaf set is X . In addition, \mathcal{T} is *binary* if, apart from the root which has degree two, all interior vertices have degree three. The set X is called the *label set* of \mathcal{T} and we sometimes denote it as $\mathcal{L}(\mathcal{T})$. Examples of rooted binary phylogenetic trees are shown in Fig. 11.1 and at the top of Fig. 11.2.

For convenience, many of the examples that arise in this chapter are based on rooted caterpillar trees. A *rooted caterpillar tree* is a rooted binary phylogenetic tree that has a leaf vertex, x say, such that every other leaf vertex is attached to the path from x to the root via a pendant edge. The rooted binary phylogenetic tree shown in Fig. 11.1 is an example of a rooted caterpillar tree. Without ambiguity, we denote this rooted caterpillar tree by the n -tuple (x_1, x_2, \dots, x_n) as this is the ordering of the label set induced by the path from x_1 to the root. Note that the first two coordinates of this tuple could be interchanged to describe the same rooted caterpillar tree.

Let \mathcal{T} be a rooted phylogenetic X -tree and let v be a vertex of \mathcal{T} . The subset of elements X that are descendants of v is called a *cluster* of \mathcal{T} . We denote this cluster by $C_{\mathcal{T}}(v)$ or simply $C(v)$ if there is no ambiguity. We sometimes say that $C(v)$ is the cluster of \mathcal{T} *corresponding to v in \mathcal{T}* . The set of clusters of \mathcal{T} is denoted by $\mathcal{C}(\mathcal{T})$. Note here that the root of \mathcal{T} gives rise to a cluster.

For a rooted phylogenetic X -tree \mathcal{T} , several different types of rooted subtrees will play a prominent role in this chapter. Let X' be a subset of X . The minimal rooted subtree of \mathcal{T} that connects the leaves in X' is denoted by $\mathcal{T}(X')$. Furthermore, the *restriction* of \mathcal{T} to X' , denoted by $\mathcal{T}|X'$, is the rooted phylogenetic tree obtained from $\mathcal{T}(X')$ by suppressing any non-root vertices of degree 2. Lastly, a rooted subtree of \mathcal{T} is *pendant* if it can be obtained from \mathcal{T} by deleting a single edge. For example, in Fig. 11.1, the minimal rooted subtree that connects the leaves in $\{x_1, x_2, x_3\}$ is a pendant rooted subtree, but the minimal rooted subtree connecting x_2 and x_3 is not a pendant rooted subtree.

11.2 Hybridization networks

In this section, we formalize the optimization problem described in the introduction. We begin with the concept of a hybridization network which is central to

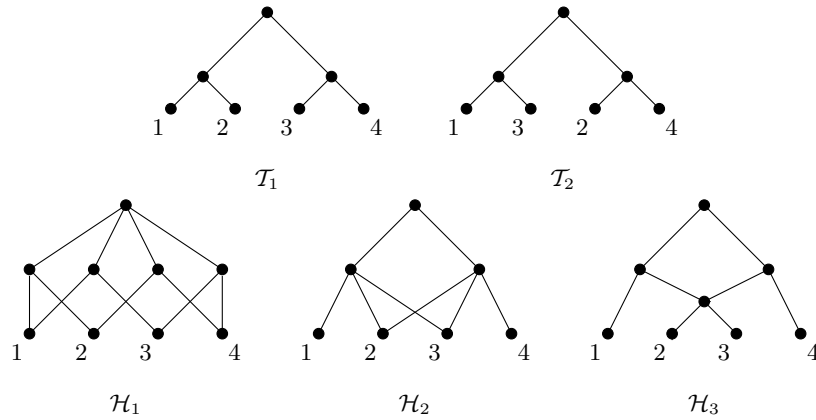


FIG. 11.2. Two rooted binary phylogenetic trees \mathcal{T}_1 and \mathcal{T}_2 , and three hybridization networks \mathcal{H}_1 , \mathcal{H}_2 , and \mathcal{H}_3 . Each of the hybridization networks \mathcal{H}_1 and \mathcal{H}_2 display both \mathcal{T}_1 and \mathcal{T}_2 .

this problem and this chapter. These networks are particular types of digraphs.

A *directed graph* (also known as a *digraph*) consists of a collection of vertices and a collection of directed edges called *arcs*. If an arc is directed from the vertex u to the vertex v , then it is denoted as the ordered pair (u, v) . The *degree* of a vertex v is the number of arcs incident with v . To distinguish between arcs coming into v and arcs coming out of v , we refer to the number of arcs coming into v as the *indegree* of v , while the number of arcs coming out of v is referred to as the *outdegree* of v . This is denoted as $d^-(v)$ and $d^+(v)$, respectively. In evolutionary biology, directed graphs are used to represent the evolutionary history of a collection of present-day species. Vertices may represent species, individuals, or DNA sequences, while arcs represent ancestral relationships. By viewing the edges as arcs directed away from the root, rooted phylogenetic trees are examples of such digraphs.

A *directed path* in a digraph D is an alternating sequence

$$v_0, a_1, v_1, a_2, v_2, \dots, v_{k-1}, a_k, v_k$$

of vertices and arcs in which a_i is directed from v_{i-1} to v_i for all i , and no vertex or arc appears more than once. A *directed cycle* in D is a directed path in which $v_0 = v_k$. We say that D is *acyclic* if it contains no directed cycles. An acyclic digraph D is *rooted* if the underlying graph has no parallel edges, and there is a distinguished vertex ρ with $d^-(\rho) = 0$ and the property that there is a directed path from ρ to every vertex of D .

A *hybridization network* (on X) is a rooted acyclic digraph with root ρ in which

- (i) X is the set of vertices of outdegree zero,
- (ii) $d^+(\rho) \geq 2$, and

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

The set X represents a collection of taxa and is the *label set* of \mathcal{H} . For convenience, it is sometimes denoted as $\mathcal{L}(\mathcal{H})$. Vertices of indegree at least two represent an exchange of genetic information between their parents. Generically, we call these vertices *hybridization vertices*. In the literature, hybridization networks have been referred to as “hybrid phylogenies” (e.g., [7]) and “phylogenetic networks” (e.g., [32, 41]). The latter with the additional property that hybridization vertices have indegree exactly two. Note here that vertices with indegree more than two do not represent a simultaneous exchange of genetic information between several parents but rather an uncertainty of the exact order of “hybridization”. To illustrate the above concepts, in Fig. 11.2, \mathcal{H}_1 , \mathcal{H}_2 , and \mathcal{H}_3 are all examples of hybridization networks in which $X = \{1, 2, 3, 4\}$. Here and in all other figures, it is implicit that arcs are directed downwards. Rooted phylogenetic trees are special examples of hybridization networks in which all vertices, apart from the root, have indegree 1.

Remark In the chapter written by Huson, a “reticulate network” is simply a particular type of hybridization network. Having less restrictions on the indegree and outdegree of vertices allows for uncertainty in the exact order of speciation and hybridization. Furthermore, unlike some authors, we do not impose the condition that the outdegree of a hybridization vertex is one—this is simply for mathematical convenience and has no bearing on the results in this chapter. Lastly, we refer the reader to the figures in Huson’s chapter for the biological interpretation of hybridization networks.

To quantify the number of reticulation events, the *hybridization number* of a hybridization network \mathcal{H} with root ρ is

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

Since $d^-(v)$ is the number of parents of v and since every vertex, apart from the root, has at least one parent, $(d^-(v) - 1)$ is the number of additional parents of v . The hybridization number of a network is at least zero. Indeed, $h(\mathcal{H}) = 0$ if and only if \mathcal{H} is a rooted phylogenetic tree. In Fig. 11.2, $h(\mathcal{H}_1) = 4$, $h(\mathcal{H}_2) = 2$, and $h(\mathcal{H}_3) = 1$.

Let \mathcal{T} be a rooted phylogenetic tree and let \mathcal{H} be a hybridization network. We say that \mathcal{H} *displays* \mathcal{T} if $\mathcal{L}(\mathcal{T}) \subseteq \mathcal{L}(\mathcal{H})$ and there is a rooted subtree of \mathcal{H} that is a refinement of \mathcal{T} . In other words, \mathcal{T} can be obtained from \mathcal{H} by first deleting a subset of the edges of \mathcal{H} and any resulting isolated vertices, and then contracting edges. For example, in Fig. 11.2, \mathcal{H}_1 and \mathcal{H}_2 both display \mathcal{T}_1 and \mathcal{T}_2 , while \mathcal{H}_3 displays neither \mathcal{T}_1 nor \mathcal{T}_2 . We say that \mathcal{H} *displays* a collection \mathcal{P} of rooted phylogenetic trees if each tree in \mathcal{P} is displayed by \mathcal{H} . Furthermore, extending the definition of the hybridization number to a collection \mathcal{P} of rooted phylogenetic trees, we set

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

If $\mathcal{P} = \{\mathcal{T}, \mathcal{T}'\}$, then we denote $h(\mathcal{P})$ by $h(\mathcal{T}, \mathcal{T}')$.

We interpret the fundamental problem for hybridization networks for when the initial collection consists of two rooted binary phylogenetic trees as the following optimization problem:

MINIMUM HYBRIDIZATION

Instance: A finite set X , and two rooted binary phylogenetic X -trees \mathcal{T} and \mathcal{T}' .

Goal: Find a hybridization network \mathcal{H} that displays \mathcal{T} and \mathcal{T}' with minimum hybridization number.

Measure: The value of $h(\mathcal{H})$.

In Fig. 11.2, while \mathcal{H}_1 displays \mathcal{T}_1 and \mathcal{T}_2 , it does not minimize the hybridization number. However, it is easily checked that \mathcal{H}_2 has this property. Thus, in this case, $h(\mathcal{T}_1, \mathcal{T}_2) = 2$.

In its broadest sense, an instance of MINIMUM HYBRIDIZATION would consist of a collection of rooted phylogenetic trees. However, even in this simplest case when it consists of just two rooted binary phylogenetic trees, Bordewich and Semple [13] showed that MINIMUM HYBRIDIZATION is NP-hard (see Section 11.7). Nevertheless, there is an attractive characterization of this problem in the simplest case. This characterization provides valuable insight into the problem and is crucial to many of the results in this chapter. We describe this characterization and some of these results in the next section.

We end this section with several remarks. First, the input in the above problem could equally have been a set of sequences instead of a set of trees, in which case, instead of seeking a ‘minimal’ hybridization network, we look for a “recombination network” that has this property. A number of authors have considered this variant of the problem and we will describe it in Section 11.5. Second, in keeping with the terminology in the chapter written by Huson and elsewhere, we use the term “hybridization networks” as the input is unordered. In contrast, if the input is ordered in some way, as in the case of sequences, then the analogous digraphs are called “recombination networks”. Lastly, as explicitly pointed out by Moret *et al.* [39], one needs to be careful in inferring information about hybridization events and the ancestral species involved in such events. In particular, the absence of unsampled taxa can have important ramifications in interpreting the true evolutionary history of the sampled taxa.

11.3 A characterization of MINIMUM HYBRIDIZATION

Historically, one of the main tools that has been used to understand and model reticulate evolution is a graph-theoretic operation called “rooted subtree prune and regraft”. Informally, this operation prunes a subtree of a rooted tree and then reattaches this subtree to another part of the tree. The use of this tool in evolutionary biology dates back to at least 1990 [24], and has been regularly used since as a way to model reticulate evolution (for example, see [7, 35, 41, 51]). The reason for this is that if two rooted binary phylogenetic X -trees are

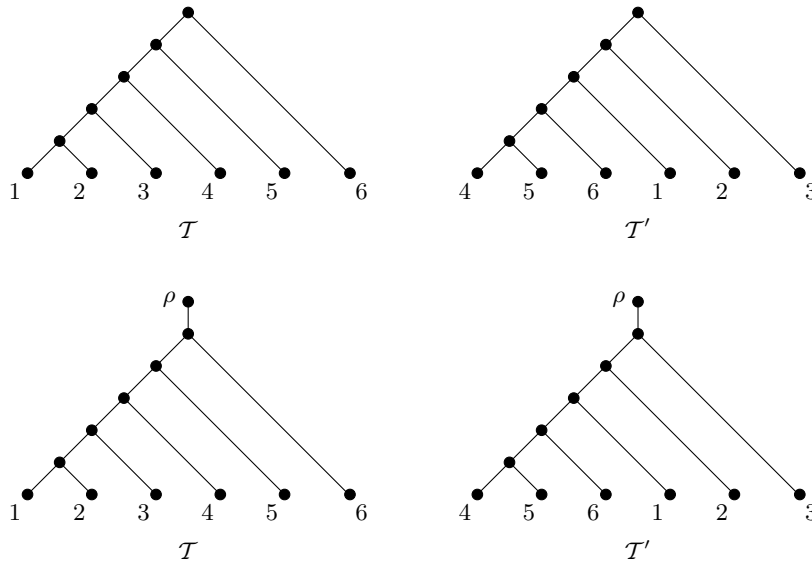


FIG. 11.3. Two rooted binary phylogenetic trees \mathcal{T} and \mathcal{T}' without (above) and with (below) their root labelled ρ .

inconsistent, but this inconsistency can be explained with a single hybridization event, then one tree can be obtained from the other by a single rooted subtree prune and regraft operation. Indeed, given this, it is tempting to conjecture that the minimum number of hybridization events to explain the inconsistency of two rooted binary phylogenetic X -trees is equal to the minimum number of rooted subtree prune and regraft operations to transform one tree into the other. We will make this precise shortly, however, this is not the case. Nevertheless, these two minimum numbers are very closely related as they can both be characterized in terms of “agreement forests”. It is one of these characterizations that is referred to at the end of Section 11.2.

11.3.1 Rooted subtree prune and regraft operation and agreement forests

To make the characterizations work, we regard the root of each of the two rooted binary phylogenetic X -trees \mathcal{T} and \mathcal{T}' in the upcoming definitions as a vertex ρ at the end of a pendant edge (called the *root edge*) adjoined to the original root. Furthermore, we regard ρ as part of the label sets of \mathcal{T} and \mathcal{T}' , and so $\mathcal{L}(\mathcal{T}) = \mathcal{L}(\mathcal{T}') = X \cup \{\rho\}$. To illustrate, consider the two rooted binary phylogenetic trees \mathcal{T} and \mathcal{T}' shown at the top of Fig. 11.3. In the following, we regard \mathcal{T} and \mathcal{T}' as shown at the bottom of Fig. 11.3.

Let $e = \{u, v\}$ be an edge of \mathcal{T} that is not the root edge, where u is the vertex that is on the path from the root of \mathcal{T} to v . Let \mathcal{T}' be the rooted binary phylogenetic tree obtained from \mathcal{T} by deleting e and reattaching the resulting rooted subtree via a new edge, f say, as follows. Create a new vertex u' that

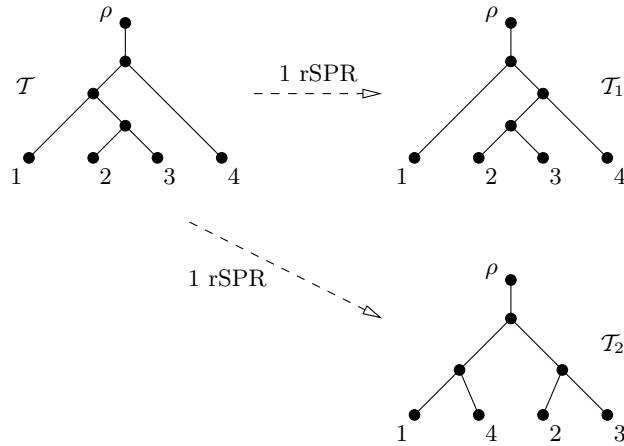


FIG. 11.4. Each of \mathcal{T}_1 and \mathcal{T}_2 are obtained from \mathcal{T} by a single rooted subtree prune and regraft operation.

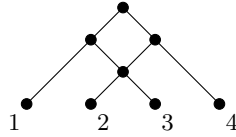


FIG. 11.5. The hybridization network resulting from the single rooted subtree prune and regraft operation that transforms \mathcal{T} into \mathcal{T}_1 in Fig. 11.4.

subdivides an edge of the component that contains ρ and adjoin f between u' and v , then suppress the degree-2 vertex u . We say that \mathcal{T}' has been obtained from \mathcal{T} by a *rooted subtree prune and regraft* (rSPR) operation. To illustrate, consider Fig. 11.4. Each of \mathcal{T}_1 and \mathcal{T}_2 are obtained from \mathcal{T} by a single rSPR operation. Denoted by $d_{\text{rSPR}}(\mathcal{T}, \mathcal{T}')$, we define the *rSPR distance* between \mathcal{T} and \mathcal{T}' to be the minimum number of rooted subtree prune and regraft operations that is required to transform \mathcal{T} into \mathcal{T}' . It is well known that, for any such pair of trees, one can always obtain one tree from the other by a sequence of rSPR operations, and so this distance is well-defined. Moreover, this distance is a metric on the collection of rooted binary phylogenetic X -trees.

To explicitly highlight the connection between rooted subtree prune and regraft operations and hybridization events, consider \mathcal{T} and \mathcal{T}_1 in Fig. 11.4. The evolutionary difference in the two trees can be explained by a single hybridization event; the corresponding hybridization vertex is the root of the pendant subtree that is pruned and regrafted in the rooted subtree prune and regraft operation shown in the figure. The resulting hybridization network is shown in Fig. 11.5.

Analogous to MINIMUM HYBRIDIZATION, we formally state the optimization problem of computing the rSPR distance between two rooted binary phylogenetic

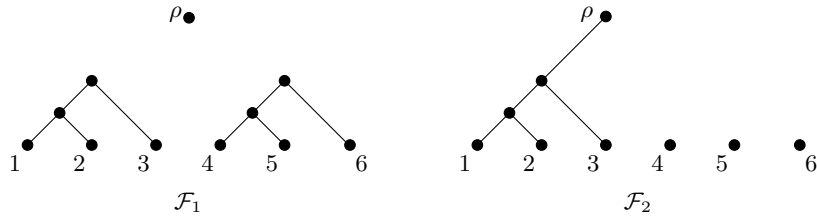


FIG. 11.6. Two possible agreement forests for \mathcal{T} and \mathcal{T}' in Fig. 11.3. \mathcal{F}_1 is a maximum-agreement forest for \mathcal{T} and \mathcal{T}' , while \mathcal{F}_2 is a maximum-acyclic-agreement forest for \mathcal{T} and \mathcal{T}' .

trees as follows.

MINIMUM rSPR

Instance: A finite set X , and two rooted binary phylogenetic X -trees \mathcal{T} and \mathcal{T}' .

Goal: Find a minimum length sequence of single rSPR operations that transforms \mathcal{T} into \mathcal{T}' .

Measure: The length of this sequence.

An *agreement forest* for \mathcal{T} and \mathcal{T}' is a collection $\{\mathcal{T}_\rho, \mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_k\}$ of rooted leaf-labelled trees, where \mathcal{T}_ρ is a rooted tree whose label set \mathcal{L}_ρ contains ρ and $\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_k$ are rooted binary phylogenetic trees with label sets $\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_k$, respectively, such that the following properties are satisfied:

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

It is easily seen that if \mathcal{F} is an agreement forest for \mathcal{T} and \mathcal{T}' , then, up to suppressing non-root vertices of degree two, \mathcal{F} can be obtained from each of \mathcal{T} and \mathcal{T}' by deleting $|\mathcal{F}| - 1$ edges. An agreement forest for \mathcal{T} and \mathcal{T}' is a *maximum-agreement forest* if, amongst all agreement forests for \mathcal{T} and \mathcal{T}' , it has the smallest number of components, in which case we denote this value of k by $m(\mathcal{T}, \mathcal{T}')$. For example, two agreement forests for the two trees \mathcal{T} and \mathcal{T}' in Fig. 11.3 are shown in Fig. 11.6. It is easily checked that the smallest number of components in any such forest is 3, so \mathcal{F}_1 is also a maximum-agreement forest for \mathcal{T} and \mathcal{T}' , and $m(\mathcal{T}, \mathcal{T}') = 2$.

11.3.2 Characterizations of MINIMUM HYBRIDIZATION and MINIMUM rSPR

Intuitively, the edges that are deleted to obtain an agreement forest for \mathcal{T} and \mathcal{T}' are those which disagree in \mathcal{T} and \mathcal{T}' , and correspond to different paths of genetic inheritance; that is hybridization events. Thus, the fewer edges deleted, the smaller the number of hybridization events. Part (i) of the following theorem

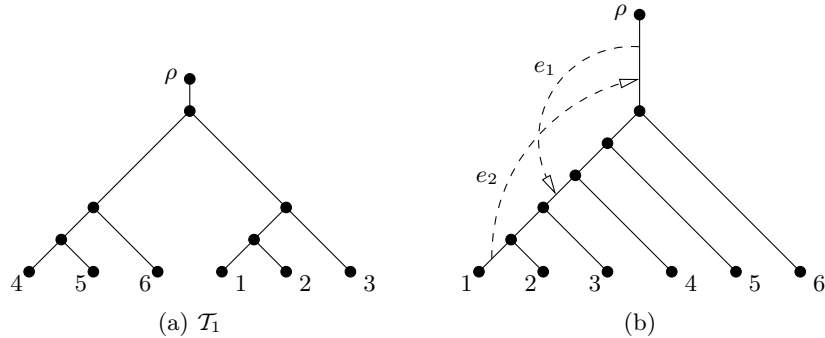


FIG. 11.7. (a) The second tree in the sequence of rSPR operations that transforms \mathcal{T} into \mathcal{T}' , where \mathcal{T} and \mathcal{T}' are as shown in Fig. 11.3. (b) The network induced by the two rSPR operations that transforms \mathcal{T} into \mathcal{T}' .

due to Bordewich and Semple [12] characterizes the rSPR distance between two rooted binary phylogenetic trees in terms of agreement forests.

Theorem 11.1 *Let \mathcal{T} and \mathcal{T}' be two rooted binary phylogenetic X -trees. Then*

- (i) $d_{\text{rSPR}}(\mathcal{T}, \mathcal{T}') = m(\mathcal{T}, \mathcal{T}')$.
- (ii) *If \mathcal{F} is an agreement forest for \mathcal{T} and \mathcal{T}' of size $k + 1$ (i.e. $k \geq m(\mathcal{T}, \mathcal{T}')$), then there is a polynomial-time algorithm for constructing a sequence*

$$\mathcal{T} = \mathcal{T}_0, \mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_k = \mathcal{T}'$$

of rooted binary phylogenetic trees such that, for all i , \mathcal{T}_i is obtained from \mathcal{T}_{i-1} by at most one rooted subtree prune and regraft operation (i.e. $d_{\text{rSPR}}(\mathcal{T}, \mathcal{T}') \leq k$).

Remarks

1. Part (ii) of Theorem 11.1 is not explicitly stated in [12]. However, it is an immediate consequence of the inductive proof of [12, Theorem 2.1]. Although we omit the proof of this result, we will describe the algorithm in (ii) later in this section.
2. For those readers familiar with the tree rearrangement operation “tree bisection and reconnection” (TBR), Allen and Steel [3] describe an analogous characterization for TBR in terms of agreement forests.
3. As we will soon see, agreement forests characterizations have been successfully used in gaining invaluable insights of various measures in phylogenetics. To provide intuition why such a characterization is useful, think how much easier it is to consider deleting edges of \mathcal{T} and \mathcal{T}' to obtain an agreement forest as oppose to keeping track of a sequence of rSPR operations that transforms \mathcal{T} into \mathcal{T}' .

Although it seems plausible that one could repeatedly use a single rooted subtree prune and regraft operation to represent a single hybridization event

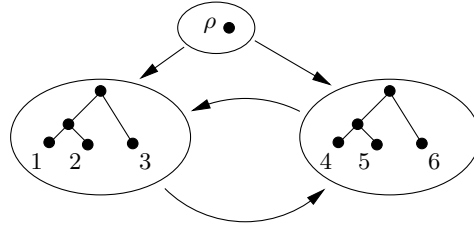


FIG. 11.8. The directed graph $G_{\mathcal{F}_1}$, where \mathcal{F}_1 is the agreement forest in Fig. 11.6.

and thus the number of such events is equal to the number of such operations, the associated hybridization network that one builds in this process may contain a directed cycle. Such a cycle would mean that a vertex in this network inherits genetic information from its own descendants. As an example, consider the two rooted binary phylogenetic trees \mathcal{T} and \mathcal{T}' shown in Fig. 11.3. The tree \mathcal{T}' can be obtained from \mathcal{T} by two rSPR operations by first pruning the pendant subtree with label set $\{1, 2, 3\}$ of \mathcal{T} and regrafting to obtain the tree \mathcal{T}_1 in Fig. 11.7(a), and then pruning the pendant subtree of \mathcal{T}_1 with label set $\{4, 5, 6\}$ and regrafting to obtain \mathcal{T}' . If one keeps each of the edges that are cut and added in this process, one obtains the “hybridization” network shown in Fig. 11.7(b). Here e_1 is the edge that is added in the first rSPR operation and e_2 is the edge that is added in the second rSPR operation. However, by viewing the (solid) edges as arcs directed away from ρ , this network contains a directed cycle. To avoid the construction of such a cycle and, in particular, rooted subtree prune and regraft operations that cause these cycles, we extend the definition of an agreement forest to an acyclic-agreement forest.

Let $\mathcal{F} = \{\mathcal{T}_\rho, \mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_k\}$ be an agreement forest for \mathcal{T} and \mathcal{T}' . Let $G_{\mathcal{F}}$ be the directed graph whose vertex set is \mathcal{F} and for which $(\mathcal{T}_i, \mathcal{T}_j)$ is an arc precisely if $i \neq j$ and either

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

Note that, as \mathcal{F} is an agreement forest, the roots of $\mathcal{T}(\mathcal{L}_i)$ and $\mathcal{T}(\mathcal{L}_j)$, and the roots of $\mathcal{T}'(\mathcal{L}_i)$ and $\mathcal{T}'(\mathcal{L}_j)$ are not the same. We say that \mathcal{F} is *acyclic* if $G_{\mathcal{F}}$ has no directed cycles. If \mathcal{F} is acyclic and it has the smallest number of components over all acyclic-agreement forests for \mathcal{T} and \mathcal{T}' , then \mathcal{F} is a *maximum-acyclic-agreement forest* for \mathcal{T} and \mathcal{T}' , in which case we denote the number k by $m_a(\mathcal{T}, \mathcal{T}')$. Observe that $m_a(\mathcal{T}, \mathcal{T}') = 0$ if and only if, up to isomorphism, \mathcal{T} and \mathcal{T}' are identical. To illustrate these concepts, Fig. 11.8 shows the directed graph $G_{\mathcal{F}_1}$ of the agreement forest \mathcal{F}_1 shown in Fig. 11.6, where large open circles represent the vertices. Since this graph contains a directed cycle, \mathcal{F}_1 is not acyclic. However, it is easily checked that $G_{\mathcal{F}_2}$, where \mathcal{F}_2 is the agreement forest in Fig. 11.6, is acyclic. In fact, one can also check that this is a maximum-acyclic-agreement forest for \mathcal{T} and \mathcal{T}' .

Analogous to Theorem 11.1, Baroni *et al.* [9] characterized the hybridization number of two rooted binary phylogenetic trees in terms of agreement forests.

Theorem 11.2 *Let \mathcal{T} and \mathcal{T}' be two rooted binary phylogenetic X -trees. Then*

- (i) $h(\mathcal{T}, \mathcal{T}') = m_a(\mathcal{T}, \mathcal{T}')$.
- (ii) *If \mathcal{F} is an acyclic-agreement forest for \mathcal{T} and \mathcal{T}' of size $k + 1$ (i.e. $k \geq m_a(\mathcal{T}, \mathcal{T}')$), then there is a polynomial-time algorithm for constructing a hybridization network \mathcal{H} that displays \mathcal{T} and \mathcal{T}' with $h(\mathcal{H}) \leq k$ (i.e. $h(\mathcal{T}, \mathcal{T}') \leq k$).*

Remarks

1. Part (ii) of Theorem 11.2 is not stated in [9], but it is an immediate consequence of its inductive proof [9, Theorem 2]. Like part (ii) of Theorem 11.1, we will describe the algorithm in (ii) at the end of this section.
2. In contrast to the rSPR distance, the hybridization number is not a metric on the collection of rooted binary phylogenetic X -trees. To see this, consider \mathcal{T} and \mathcal{T}' in Fig. 11.3 and \mathcal{T}_1 in Fig. 11.7. We have already noted that $h(\mathcal{T}, \mathcal{T}') = 3$. Furthermore, it is easily checked that $h(\mathcal{T}, \mathcal{T}_1) = h(\mathcal{T}_1, \mathcal{T}') = 1$, and so the hybridization number does not satisfy the triangle inequality.
3. If one is only interested in the number of hybridization vertices (and not what each such vertex contributes to the hybridization number), then Theorem 11.2 is easily generalized to an arbitrary size collection of rooted binary phylogenetic X -trees. Here the notion of an acyclic-agreement forest for two trees is extended in the obvious way. For details, see [34].

Since every acyclic-agreement forest for two rooted binary phylogenetic X -trees \mathcal{T} and \mathcal{T}' is an (ordinary) agreement forest for \mathcal{T} and \mathcal{T}' , it follows from Theorems 11.1 and 11.2 that

$$d_{\text{rSPR}}(\mathcal{T}, \mathcal{T}') \leq h(\mathcal{T}, \mathcal{T}'). \quad (11.1)$$

The fact that this inequality can be strict has been pointed out several times in the literature including [9, 25, 53]. An interesting question is just how strict? We consider this question in Section 11.3.3.

11.3.3 Comparing $d_{\text{rSPR}}(\mathcal{T}, \mathcal{T}')$ and $h(\mathcal{T}, \mathcal{T}')$

Two natural questions arise from the inequality in (11.1).

- (i) Whenever $d_{\text{rSPR}}(\mathcal{T}, \mathcal{T}') = 1$, we have that $h(\mathcal{T}, \mathcal{T}') = 1$, and so $d_{\text{rSPR}}(\mathcal{T}, \mathcal{T}')$ provides a sharp lower bound for $h(\mathcal{T}, \mathcal{T}')$. Can we find a sharp upper bound for $h(\mathcal{T}, \mathcal{T}')$?
- (ii) We have already seen that inequality (11.1) can be strict, so how large can the difference between $d_{\text{rSPR}}(\mathcal{T}, \mathcal{T}')$ and $h(\mathcal{T}, \mathcal{T}')$ be?

Consider (i). Regardless of the topology of \mathcal{T} and \mathcal{T}' , if $X = \{x_1, x_2, \dots, x_n\}$, then, as the forest consisting of $\mathcal{T}|_{\{\rho, x_1, x_2\}}$ and isolated vertices x_3, x_4, \dots, x_n is an acyclic-agreement forest for \mathcal{T} and \mathcal{T}' ,

$$h(\mathcal{T}, \mathcal{T}') \leq n - 2.$$

Using Theorem 11.2, Baroni *et al.* [9] showed that this upper bound is sharp. In particular, if \mathcal{T} and \mathcal{T}' are the two rooted caterpillars (x_1, x_2, \dots, x_n) and $(x_n, x_{n-1}, \dots, x_1)$, then $h(\mathcal{T}, \mathcal{T}') = n - 2$. In the same paper [9] and using Theorems 11.1 and 11.2, the authors also establish the following theorem.

Theorem 11.3 *For all $n \geq 4$, there are rooted binary phylogenetic trees \mathcal{T}_1 , \mathcal{T}_2 , and \mathcal{T}_3 on n leaves such that*

$$\frac{h(\mathcal{T}_1, \mathcal{T}_2)}{d_{\text{rSPR}}(\mathcal{T}_1, \mathcal{T}_2)} = \frac{1}{2} \left\lfloor \frac{n}{2} \right\rfloor$$

and

$$h(\mathcal{T}_1, \mathcal{T}_3) - d_{\text{rSPR}}(\mathcal{T}_1, \mathcal{T}_3) = n - 2\lfloor\sqrt{n}\rfloor - c,$$

where $c = 0$ if n is a square, $c = 1$ if $1 \leq n - \lfloor\sqrt{n}\rfloor^2 < \sqrt{n}$, and $c = 2$ otherwise.

Explicit examples of rooted binary phylogenetic trees that attain the equalities in Theorem 11.3 are given in [9]. For example, let \mathcal{T}_1 be the rooted caterpillar tree $(x_1, x_2, \dots, x_{100})$. Let \mathcal{T}_2 and \mathcal{T}_3 be the rooted caterpillar trees on $\{x_1, x_2, \dots, x_{100}\}$ whose orderings on their leaf sets are

$$(x_{51}, x_{52}, \dots, x_{100}, x_1, x_2, \dots, x_{50})$$

and

$$(x_{91}, x_{92}, \dots, x_{100}, x_{81}, x_{82}, \dots, x_{90}, x_{71}, \dots, x_{19}, x_{20}, x_1, x_2, \dots, x_{10}),$$

respectively. Then

$$\frac{h(\mathcal{T}_1, \mathcal{T}_2)}{d_{\text{rSPR}}(\mathcal{T}_1, \mathcal{T}_2)} = \frac{1}{2} \left\lfloor \frac{100}{2} \right\rfloor = 25$$

and

$$h(\mathcal{T}_1, \mathcal{T}_3) - d_{\text{rSPR}}(\mathcal{T}_1, \mathcal{T}_3) = 100 - 2\lfloor\sqrt{100}\rfloor - 0 = 80$$

An interesting question is determine whether the ratio or difference given in Theorem 11.3 is the best possible.

The answers to (i) and (ii) in [9] both rely on Theorems 11.1 and 11.2. It seems unlikely that, without such characterizations, these results could have been attained as easily. Further applications of these theorems are given in Section 11.4.

11.3.4 Algorithms for constructing rSPR sequences and hybridization networks from agreement forests

Let \mathcal{F} be an arbitrary agreement forest for two rooted binary phylogenetic X -trees \mathcal{T} and \mathcal{T}' . The first algorithm RSPRSEQUENCE constructs a sequence of rooted binary phylogenetic trees beginning with \mathcal{T} and ending with \mathcal{T}' with the property that each tree in the sequence is obtained from its predecessor by a single (possibly trivial) rSPR operation. Provided \mathcal{F} is acyclic, the second algorithm HYBRIDNETWORK constructs a hybridization network \mathcal{H} that displays \mathcal{T} and \mathcal{T}' with $h(\mathcal{H}) \leq |\mathcal{F}| - 1$. Each algorithm is an immediate consequence of the inductive proofs of Theorems 11.1 and 11.2 in [12] and [9], respectively.

Algorithm: RSPRSEQUENCE(\mathcal{F})

Input: An agreement forest \mathcal{F} of size $k + 1$ of two rooted binary phylogenetic X -trees \mathcal{T} and \mathcal{T}' .

Output: A sequence $\mathcal{T}_0, \mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_k$ of rooted binary phylogenetic X -trees with the property that $\mathcal{T}_0 = \mathcal{T}$, $\mathcal{T}_k = \mathcal{T}'$, and, for all i , either \mathcal{T}_i is obtained from \mathcal{T}_{i-1} by a single rSPR operation or $\mathcal{T}_i \cong \mathcal{T}_{i-1}$.

1. Set $\mathcal{T} = \mathcal{T}_0$, $\mathcal{F} = \mathcal{F}_0$, and $i = 1$.
2. Find a tree \mathcal{S}_i in \mathcal{F}_{i-1} such that \mathcal{S}_i is a pendant subtree of \mathcal{T}_{i-1} .
3. In \mathcal{T}' , find the first subtree $\mathcal{T}'(\mathcal{L}(\mathcal{S}_j))$ corresponding to a tree \mathcal{S}_j in \mathcal{F}_{i-1} that is met on the path from the root of $\mathcal{T}'(\mathcal{L}(\mathcal{S}_i))$ to ρ .
4. Set \mathcal{T}_i to be a tree that is obtained from \mathcal{T}_{i-1} by pruning \mathcal{S}_i and regrafting it so that \mathcal{T}_i restricted to $\mathcal{L}(\mathcal{S}_i) \cup \mathcal{L}(\mathcal{S}_j)$ is isomorphic to \mathcal{T}' restricted to $\mathcal{L}(\mathcal{S}_i) \cup \mathcal{L}(\mathcal{S}_j)$.
5. Set \mathcal{F}_i to be the forest obtained from \mathcal{F}_{i-1} by replacing \mathcal{S}_i and \mathcal{S}_j with \mathcal{T}' restricted to $\mathcal{L}(\mathcal{S}_i) \cup \mathcal{L}(\mathcal{S}_j)$.
6. If $i = k$ halt; otherwise, increment i by 1 and return to Step 2.

Remarks The following comments may help the reader.

1. Step 2 is well-defined as there is always at least one tree that has this property.
2. In Step 3, the choice for \mathcal{S}_j is unique because of (iii) in the definition of an agreement forest.
3. In Step 4, \mathcal{F}_i is an agreement forest for \mathcal{T}_i and \mathcal{T}' .

Before stating HYBRIDNETWORK, we need an additional concept. A simple, fast, and well-known way of deciding whether a directed graph G is acyclic is as follows. Find a vertex, v_1 say, of G that has indegree 0. If there is no such vertex, then G contains a directed cycle and so G is acyclic. Otherwise, delete v_1 (and its incident arcs) from G and find a vertex, v_2 say, of G that has indegree 0. Again, if there is no such vertex, then G is not acyclic, otherwise delete v_2 from this last digraph and continue in this way. Eventually, we either decide that G is not acyclic or obtain an ordering v_1, v_2, \dots, v_n of the vertex set of G such that, for all i , the vertex v_i has indegree 0 in the graph obtained from G by deleting

the vertices v_1, v_2, \dots, v_{i-1} . Such an ordering is called an *acyclic ordering* of G and it implies that G is acyclic.

Algorithm: HYBRIDNETWORK(\mathcal{F})

Input: An acyclic-agreement forest \mathcal{F} of size $k + 1$ of two rooted binary phylogenetic X -trees \mathcal{T} and \mathcal{T}' .

Output: A hybridization network \mathcal{H} that displays \mathcal{T} and \mathcal{T}' with $h(\mathcal{H}) \leq k$.

1. Find an acyclic ordering, $\mathcal{S}_\rho, \mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_k$ say, of $G_{\mathcal{F}}$.
2. Set $\mathcal{H}_0 = \mathcal{S}_\rho$ and set $i = 1$.
3. Attach \mathcal{S}_i to \mathcal{H}_{i-1} via two new arcs. Each arc joins the root of \mathcal{S}_i to some (possibly distinct) arc of \mathcal{H}_{i-1} and is directed towards the root of \mathcal{S}_i . These arcs are added so that the resulting network displays both \mathcal{T} restricted to $\mathcal{L}(\mathcal{H}_{i-1}) \cup \mathcal{L}(\mathcal{S}_i)$ and \mathcal{T}' restricted to $\mathcal{L}(\mathcal{H}_{i-1}) \cup \mathcal{L}(\mathcal{S}_i)$. Set \mathcal{H}_i to be the resulting network and return \mathcal{H}_i if $i = k$.
4. Increment i by 1 and return to Step 3.

Remark In Step 3 of the algorithm, it may be possible that only one new edge is required. This implies that \mathcal{F} is not maximum and that a new acyclic-agreement forest for \mathcal{T} and \mathcal{T}' can be obtained by attaching one component \mathcal{S} of \mathcal{F} to another via an edge directed towards the root of \mathcal{S} .

11.4 Algorithmic applications of agreement forests

For two rooted binary phylogenetic trees \mathcal{T} and \mathcal{T}' , agreement forests are a particularly useful tool for analyzing the individual values $d_{\text{tSPR}}(\mathcal{T}, \mathcal{T}')$ and $h(\mathcal{T}, \mathcal{T}')$. In this section, we consider ways that agreement forests can be used for this analysis and the resulting algorithmic implications, while in Section 11.7 we see that this tool provides invaluable leverage in understanding the computation complexity of finding these values.

As we formally state in Section 11.7, both MINIMUM RSPR and MINIMUM HYBRIDIZATION are NP-hard problems. Nevertheless, they are both susceptible to approaches that effectively reduce the size of the problem instance. Interestingly, these approaches are different and it appears that they are unique to the particular problem. For MINIMUM RSPR, we reduce the size of the problem instance while preserving the rooted subtree prune and regraft distance, while, for MINIMUM HYBRIDIZATION, we use a divide-and-conquer type approach, that is we break the problem into a number of smaller problems. To avoid some repetition, the proofs of the first four results in this section rely on either Theorem 11.1 or Theorem 11.2.

11.4.1 Reduction rules

For MINIMUM RSPR, consider the following two reduction rules:

- Rule 1. Replace a pendant subtree that occurs identically in both trees by a single leaf with a new label.

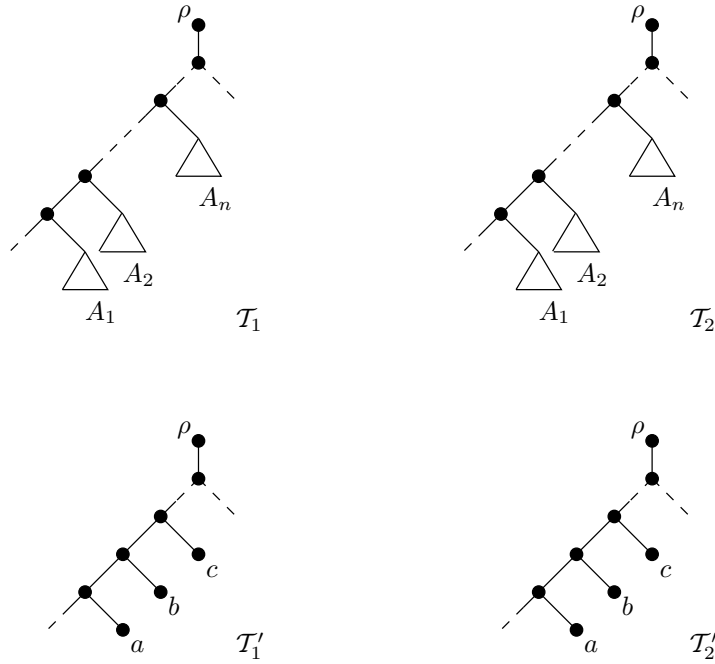


FIG. 11.9. Applying Rule 2 to two rooted binary phylogenetic trees \mathcal{T}_1 and \mathcal{T}_2 , we obtain \mathcal{T}'_1 and \mathcal{T}'_2 , respectively.

Rule 2. Replace a chain of at least three pendant subtrees that occur identically and with the same orientation relative to the root in both trees by three new leaves with new labels correctly orientated to preserve the direction of the chain.

Rule 2 is illustrated in Fig. 11.9, where A_1, A_2, \dots, A_n is the chain of pendant subtrees common to both \mathcal{T}_1 and \mathcal{T}_2 , and $a, b,$ and c are the three new leaf labels orientated appropriately.

The following theorem is due to Bordewich and Semple [12].

Theorem 11.4 *Let \mathcal{T}_1 and \mathcal{T}_2 be two rooted binary phylogenetic X -trees, and let \mathcal{T}'_1 and \mathcal{T}'_2 be the two rooted binary phylogenetic X' -trees obtained from \mathcal{T}_1 and \mathcal{T}_2 , respectively, by applying either Rule 1 or Rule 2. Then*

$$d_{\text{rSPR}}(\mathcal{T}_1, \mathcal{T}_2) = d_{\text{rSPR}}(\mathcal{T}'_1, \mathcal{T}'_2).$$

The proof of Theorem 11.4 relies on Theorem 11.1 and is the basis of showing that MINIMUM RSPR is fixed-parameter tractable in $d_{\text{rSPR}}(\mathcal{T}_1, \mathcal{T}_2)$. Intuitively, this simply means that if the rSPR distance is small, it may be possible to efficiently compute this distance even if X is large. The reason for this is that, for

small rSPR distance, one would expect the problem instance to be significantly reduced by repeatedly applying Rules 1 and 2. Note that, by Theorem 11.4, such repeated applications preserve the rSPR distance. For further details, see Section 11.7.

For MINIMUM HYBRIDIZATION, we have the following theorem due to Baroni [6] (also see Baroni *et al.* [8]), which provides a divide-and-conquer type approach to the problem.

Theorem 11.5 *Let \mathcal{T} and \mathcal{T}' be two rooted binary phylogenetic X -trees, and suppose that $A \subset X$ is a cluster of both \mathcal{T} and \mathcal{T}' . Then*

$$h(\mathcal{T}, \mathcal{T}') = h(\mathcal{T}|A, \mathcal{T}'|A) + h(\mathcal{T}_a, \mathcal{T}'_a),$$

where \mathcal{T}_a and \mathcal{T}'_a are obtained from \mathcal{T} and \mathcal{T}' , respectively, by replacing the pendant subtrees $\mathcal{T}(A)$ and $\mathcal{T}'(A)$ with a single new leaf labelled a . Furthermore, if \mathcal{H}_a is a hybridization network that displays \mathcal{T}_a and \mathcal{T}'_a with $h(\mathcal{H}_a) = h(\mathcal{T}_a, \mathcal{T}'_a)$ and \mathcal{H}_A is a hybridization network that displays $\mathcal{T}|A$ and $\mathcal{T}'|A$ with $h(\mathcal{H}_A) = h(\mathcal{T}|A, \mathcal{T}'|A)$, then the hybridization network obtained from \mathcal{H}_a by identifying the root of \mathcal{H}_A with a displays \mathcal{T} and \mathcal{T}' , and has hybridization number $h(\mathcal{T}, \mathcal{T}')$.

We will discuss the obvious divide-and-conquer algorithm resulting from Theorem 11.5 and highlight its usefulness by applying the algorithm to a biological data set in Section 11.4.2.

Recalling that if, up to isomorphism, two rooted binary phylogenetic trees are identical, then their hybridization number is 0, we get the following corollary as an immediate consequence of Theorem 11.5.

Corollary 11.6 *Let \mathcal{T}_1 and \mathcal{T}_2 be two rooted binary phylogenetic X -trees, and let \mathcal{T}'_1 and \mathcal{T}'_2 be the two rooted binary phylogenetic X' -trees obtained from \mathcal{T}_1 and \mathcal{T}_2 , respectively, by applying Rule 1. Then*

$$h(\mathcal{T}_1, \mathcal{T}_2) = h(\mathcal{T}'_1, \mathcal{T}'_2).$$

Curiously, despite Corollary 11.6, Rule 2 does not preserve the hybridization number of two rooted binary phylogenetic trees. We illustrate with a simple example. The argument used in the example is indicative of the arguments based on agreement forests. Let \mathcal{T}_1 and \mathcal{T}_2 be the rooted caterpillar trees

$$(b_1, b_2, b_3, b_4, b_5, b_6, a_1, a_2, a_3, a_4)$$

and

$$(b_1, a_1, a_2, a_3, a_4, b_2, b_3, b_4, b_5, b_6),$$

respectively. Let \mathcal{T}'_1 and \mathcal{T}'_2 be the rooted caterpillar trees obtained from \mathcal{T}_1 and \mathcal{T}_2 , respectively, by applying Rule 2 to the chain of pendant subtrees corresponding to the labels a_1, a_2, a_3, a_4 . Let a , b , and c denote the resulting new leaves. Thus \mathcal{T}'_1 and \mathcal{T}'_2 are the rooted caterpillar trees

$$(b_1, b_2, b_3, b_4, b_5, b_6, a, b, c)$$

and

$$(b_1, a, b, c, b_2, b_3, b_4, b_5, b_6),$$

respectively. First observe that the agreement forest \mathcal{F} of \mathcal{T}_1 and \mathcal{T}_2 for which the partition of $X \cup \{\rho\}$ induced by the label sets of its trees is

$$\{\{b_1, b_2, b_3, b_4, b_5, b_6, \rho\}, \{a_1\}, \{a_2\}, \{a_3\}, \{a_4\}\}$$

is acyclic. Thus the number of components of a maximum-acyclic-agreement forest of \mathcal{T}_1 and \mathcal{T}_2 is at most 5. We next show that this number is exactly 5 and that \mathcal{F} is the unique maximum-acyclic agreement forest for \mathcal{T}_1 and \mathcal{T}_2 . Let \mathcal{F}' be a maximum-acyclic-agreement forest for \mathcal{T}_1 and \mathcal{T}_2 . If $b_j \in \mathcal{L}_\rho$ for some j , then, by the maximality of \mathcal{F}' , $\{a_1\}, \{a_2\}, \{a_3\}, \{a_4\}$ are label sets of \mathcal{F}' and so, as \mathcal{F}' is maximum, $\mathcal{F}' = \mathcal{F}$. Furthermore, if $a_i \in \mathcal{L}_\rho$ for some i , then $\{b_2\}, \{b_3\}, \{b_4\}, \{b_5\}, \{b_6\}$ are label sets of \mathcal{F}' and so $|\mathcal{F}'| \geq 6$; a contradiction to maximality. Thus $\{\rho\}$ is a label set of \mathcal{F}' , in particular $\mathcal{L}_\rho \cap X$ is empty. But, because of the necessity of being acyclic, $\mathcal{L}_\rho \cap X$ is non-empty in any maximum-acyclic-agreement forest for \mathcal{T}_1 and \mathcal{T}_2 [9]. This last contradiction shows that \mathcal{F} is the unique maximum-acyclic-agreement forest for \mathcal{T}_1 and \mathcal{T}_2 . Using similar arguments, the unique maximum-acyclic-agreement forest for \mathcal{T}'_1 and \mathcal{T}'_2 is the forest for which the partition of $X \cup \{\rho\}$ induced by the label sets of its trees is

$$\{\{b_1, b_2, b_3, b_4, b_5, b_6, \rho\}, \{a\}, \{b\}, \{c\}\}.$$

But then $h(\mathcal{T}_1, \mathcal{T}_2) = 4$, while $h(\mathcal{T}'_1, \mathcal{T}'_2) = 3$. Thus Rule 2 does not preserve the hybridization number of two trees. The main point of the argument above is that, unlike the situation for (ordinary) agreement forests, there is no maximum-acyclic-agreement forest that contains a tree whose label set contains the set $\{a_1, a_2, a_3, a_4\}$, the union of the label sets of the chain of pendant subtrees that are replaced by the three new leaves.

In comparison to the last paragraph, the rSPR distance only satisfies a weaker version of Theorem 11.5. In particular, we have the following result [12].

Proposition 11.7 *Let \mathcal{T} and \mathcal{T}' be two rooted binary phylogenetic X -trees, and suppose that $A \subset X$ is a cluster of both \mathcal{T} and \mathcal{T}' . Then*

$$d_{\text{rSPR}}(\mathcal{T}, \mathcal{T}') \leq d_{\text{rSPR}}(\mathcal{T}|A, \mathcal{T}'|A) + d_{\text{rSPR}}(\mathcal{T}_a, \mathcal{T}'_a) \leq d_{\text{rSPR}}(\mathcal{T}, \mathcal{T}') + 1,$$

where \mathcal{T}_a and \mathcal{T}'_a are obtained from \mathcal{T} and \mathcal{T}' , respectively, by replacing the pendant subtrees $\mathcal{T}(A)$ and $\mathcal{T}'(A)$ with a single new leaf labelled a . Moreover, these bounds are sharp.

To see that the first bound in Proposition 11.7 is sharp, simply choose \mathcal{T} and \mathcal{T}' so that $d_{\text{rSPR}}(\mathcal{T}, \mathcal{T}') = 1$, and choose A to be the cluster of the pendant subtree that is pruned. For the sharpness of the second bound, choose \mathcal{T} and \mathcal{T}'

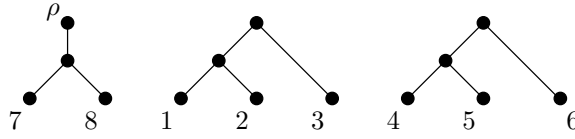


FIG. 11.10. Illustrating strict inequality in Proposition 11.7.

to be the rooted caterpillar trees $(1, 2, 3, 4, 5, 6, 7, 8)$ and $(4, 5, 6, 1, 2, 3, 8, 7)$, and choose A to be the common cluster $\{1, 2, 3, 4, 5, 6\}$. Then $d_{\text{rSPR}}(\mathcal{T}_a, \mathcal{T}'_a) = 1$ and, as we have seen previously, $d_{\text{rSPR}}(\mathcal{T}|A, \mathcal{T}'|A) = 2$, so

$$d_{\text{rSPR}}(\mathcal{T}|A, \mathcal{T}'|A) + d_{\text{rSPR}}(\mathcal{T}_a, \mathcal{T}'_a) = 3.$$

But the forest shown in Fig. 11.10 is an agreement forest for \mathcal{T} and \mathcal{T}' , and therefore $d_{\text{rSPR}}(\mathcal{T}, \mathcal{T}') \leq 2$.

In Sections 11.4.2 and 11.4.3, we describe two applications of Theorem 11.5.

11.4.2 A simple divide-and-conquer algorithm for MINIMUM HYBRIDIZATION

Proposition 11.5 and Corollary 11.6 provides us with the following simple divide-and-conquer approach to MINIMUM HYBRIDIZATION that is somewhat better than the naive approach of exhaustively searching for edges in \mathcal{T} (or \mathcal{T}') whose deletion results in an acyclic-agreement forest. This exact algorithm initially applies Rule 1 to \mathcal{T} and \mathcal{T}' as much as possible, and then locates the smallest pendant subtrees, \mathcal{W} and \mathcal{W}' say, in the resulting trees whose leaf sets are equal. Intuitively, these pendant subtrees localize conflicting signals in the evolutionary history of these parts of \mathcal{T} and \mathcal{T}' . The algorithm finds a maximum-acyclic-agreement forest for these pendant subtrees \mathcal{W} and \mathcal{W}' , and then repeats this process for the rooted binary phylogenetic trees obtained from \mathcal{T} and \mathcal{T}' by replacing the pendant subtrees with a single new vertex. Summing the hybridization number $h(\mathcal{W}, \mathcal{W}')$ at each iteration gives $h(\mathcal{T}, \mathcal{T}')$.

Algorithm: HYBRIDNUMBER($\{\mathcal{T}, \mathcal{T}'\}$)

Input: Two rooted binary phylogenetic X -trees \mathcal{T} and \mathcal{T}' .

Output: The value of $h(\mathcal{T}, \mathcal{T}')$.

1. Set $\mathcal{T}_0 = \mathcal{T}$ and $\mathcal{T}'_0 = \mathcal{T}'$, and set $i = 1$
2. Repeatedly apply Rule 1 to \mathcal{T}_{i-1} and \mathcal{T}'_{i-1} until the rule can no longer be applied, and set \mathcal{S}_{i-1} and \mathcal{S}'_{i-1} to be the resulting rooted binary phylogenetic trees, respectively. If each of \mathcal{S}_{i-1} and \mathcal{S}'_{i-1} consist of a single vertex, then go to Step 7.
3. Find a minimal cluster W_{i-1} in $\mathcal{C}(\mathcal{S}_{i-1}) \cap \mathcal{C}(\mathcal{S}'_{i-1})$ of size at least two.
4. Find a maximum-acyclic-agreement forest \mathcal{F}_{i-1} for $\mathcal{S}_{i-1}|W_{i-1}$ and $\mathcal{S}'_{i-1}|W_{i-1}$.
5. Set \mathcal{T}_i and \mathcal{T}'_i to be the rooted binary phylogenetic trees obtained from \mathcal{S}_{i-1} and \mathcal{S}'_{i-1} , respectively, by replacing $\mathcal{S}_{i-1}|W_{i-1}$ and $\mathcal{S}'_{i-1}|W_{i-1}$ with a single new vertex w_{i-1} .

6. Increment i by 1 and return to Step 2.
7. Output the sum $|\mathcal{F}_0| - 1 + |\mathcal{F}_1| - 1 + \cdots + |\mathcal{F}_{i-1}| - 1$.

Remarks

1. A naive approach to Step 4 is to exhaustively delete edges from one of the trees, \mathcal{T} say, and then see if the resulting forest is an acyclic-agreement forest for \mathcal{T} and \mathcal{T}' .
2. Observe that, if one ignores the task of finding a maximum-acyclic-agreement forest in Step 4, then HYBRIDNUMBER provides a fast lower bound for $h(\mathcal{T}, \mathcal{T}')$. In particular, the number of iterations of the algorithm.

Clearly, Step 4 is the computationally most expensive part of the algorithm. However, although there is no theoretical foundations for the complexity of this algorithm, it will work well in practice provided it breaks the problem into a number of isolated parts for which the associated hybridization number is relatively small. To see whether this proviso is realistic or not, Bordewich *et al.* [11] have carried out an experimental analysis of HYBRIDNUMBER on a particular grass (*Poaceae*) data set that has previously been considered by Schmidt [44]. Because of earlier findings of Ellstrand *et al.* [17], this data set is appropriate for such an analysis as it is more likely that the conflicting signals in the data is due to hybridization rather than other factors. Without going into the details, the analysis involves the running of the algorithm on pairs of trees with up to 40 taxa. The results highlight the usefulness of the reduction rules that underlie HYBRIDNUMBER. We describe one particularly successful example next.

The grass data set consists of sequence data for six loci. The two phylogenetic trees shown in Fig. 11.11 are the result of applying the fastDNAML program [42] to two of the sequences—a nuclear sequence (internal transcribed spacer) and a chloroplast sequence (phytochrome B). For convenience, as this example is simply illustrating how the algorithm works and nothing more, we have replaced the species names with numbers. Taking these two trees as the input to HYBRIDNUMBER, the algorithm initially finds all common subtrees and replaces each such subtree by a single leaf with a new label. The resulting trees are shown in Fig. 11.12 where, for clarity, each common subtree has been replaced by a single leaf whose label is a concatenation of the subtree labels. The next step is to search for a minimal cluster of size at least two that is common to both trees in Fig. 11.12.

One such cluster, as shown by the inside square brackets in Fig. 11.12, is $\{1, 20, 15, 19, 4, 3, 5, 29, 12, 16, 9\}$ and the corresponding subtrees are shown at the top of Fig. 11.13. This essentially completes the first iteration of the algorithm. At the completion of two further iterations, we obtain the two further pairs of subtrees (as indicated by the middle and outside square brackets shown in Fig. 11.12) and these are shown in Fig. 11.13. Again, the trees on the left come from the nuclear sequence, while the trees on the right come from the chloroplast sequence. At this stage the original inputted trees have been reduced to two trees that are identical. We now exhaustively find the hybridization number of each of

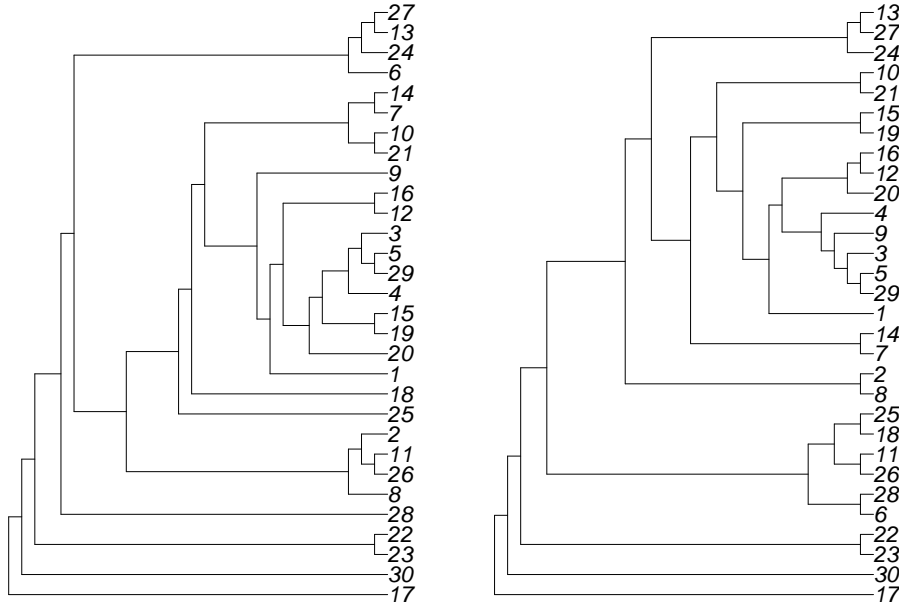


FIG. 11.11. The input to HYBRIDNUMBER. The tree resulting from the nuclear sequence is on the left, while the tree resulting from the chloroplast sequence is on the right.

the three pairs of non-identical trees. The first pair has a hybridization number of 3, while the second and third pairs have hybridization numbers of 1 and 4, respectively. Adding the three numbers together results in the hybridization number of 8 for the phylogenetic trees shown in Fig. 11.11. The running time of an implementation of the algorithm HYBRIDNUMBER applied to the two trees in Fig. 11.11 is 19 seconds. Given that the trees contain 30 taxa and have a hybridization number of 8, this is remarkably quick.

We end this subsection with two further comments. Firstly, Nakhleh *et al.* [40] describe a polynomial-time heuristic for finding $h(\mathcal{T}, \mathcal{T}')$ that is based on an agreement forest type approach. In this heuristic, they obtain a certain agreement forest by repeatedly finding a maximum-agreement subtree of two trees to decompose \mathcal{T} and \mathcal{T}' . For further details and the associated reconstruction algorithm, see [40]. Secondly, although we have not included the details here, it is straightforward to construct a hybridization network associated with HYBRIDNUMBER by combining our earlier algorithm HYBRIDNETWORK (Section 11.3.4) with the second part of Theorem 11.5. However, it is important to note that such a network is not necessarily unique. Typically, there will be a number of possibilities.

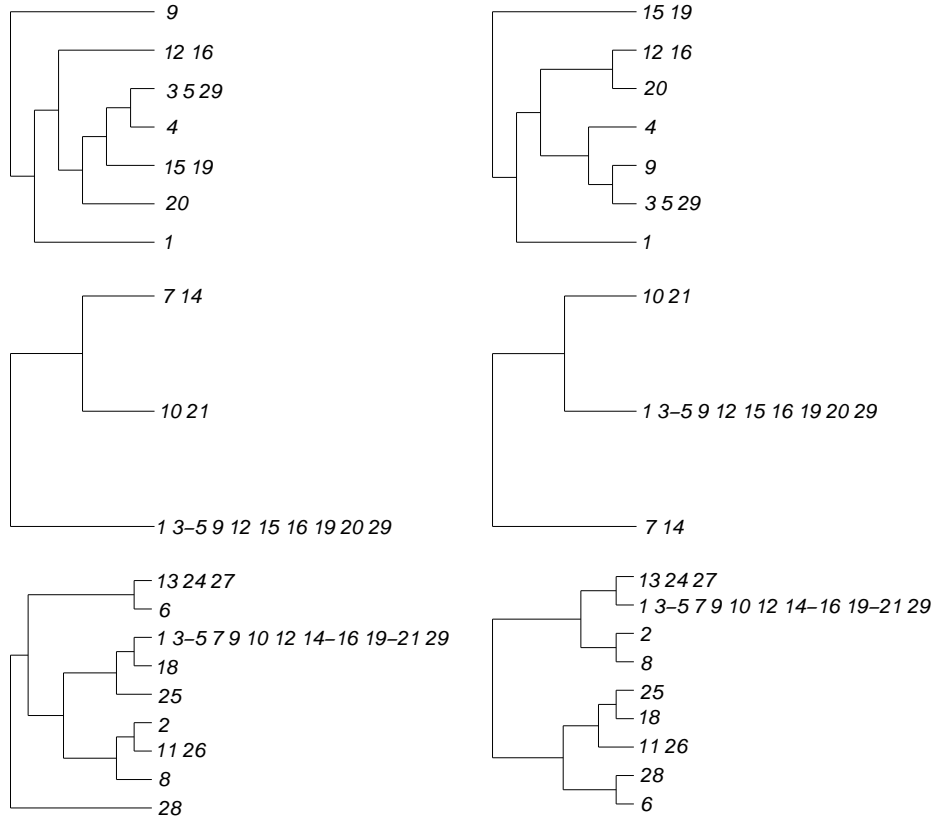


FIG. 11.13. The top pair of trees are the subtrees in Fig. 11.12 corresponding to the common cluster $\{1, 20, 15, 19, 4, 3, 5, 29, 12, 16, 9\}$. The bottom two pairs of trees are the resulting pairs of subtrees after two further iterations of HYBRIDNUMBER.

other galled-trees that display \mathcal{T} and \mathcal{T}' . However, this proviso is not necessary because of the following proposition.

Proposition 11.8 *Let \mathcal{T} and \mathcal{T}' be two rooted binary phylogenetic X -trees, and suppose that there is a galled-tree that displays \mathcal{T} and \mathcal{T}' . Suppose that the smallest number of hybridization vertices in such a network is m . Then $h(\mathcal{T}, \mathcal{T}') = m$.*

Before proving Proposition 11.8, we remark that an alternative, but equivalent, way to say Proposition 11.8 is that if there is a galled-tree that displays \mathcal{T} and \mathcal{T}' , then there is such a galled-tree that minimizes the number of hybridization vertices over all networks that displays \mathcal{T} and \mathcal{T}' . The algorithm in [41] is essentially equivalent to combining HYBRIDNUMBER and HYBRIDNETWORK,

and so one could establish the proposition as a consequence of these algorithms. However, we prove it directly using Theorem 11.5.

Proof of Proposition 11.8 The proof is by induction on m . If $m = 0$, then \mathcal{T} and \mathcal{T}' are isomorphic, so $h(\mathcal{T}, \mathcal{T}') = 0$ and the theorem holds. Now suppose that $m = k + 1$ for some $k \geq 0$ and that the theorem holds whenever the smallest number of hybridization vertices in a galled-tree that displays the two input trees is at most k .

Let \mathcal{H} be a galled-tree that displays \mathcal{T} and \mathcal{T}' , and has the smallest number of hybridization vertices amongst all such networks. Because of the minimality condition, each hybridization vertex has indegree 2. For the purposes of the proof, we will refer to the unique vertex of a gall that is closer to the root than any other vertex of the gall as the *coalescent* vertex of the gall. Let w be the coalescent vertex of a gall Q in \mathcal{H} such that there is no directed path in \mathcal{H} from w to another vertex that is the coalescent vertex of a gall in \mathcal{H} . Before continuing, we make two observations:

- (i) The subset W of X whose elements can be reached from w via a directed path is a cluster of both \mathcal{T} and \mathcal{T}' .
- (ii) The subtree of \mathcal{T} induced by W can be obtained from the subnetwork of \mathcal{H} that consists of all vertices and arcs that lie on a directed path from w by deleting one of the incoming arcs of the hybridization vertex in Q . Similarly, the subtree of \mathcal{T}' induced by W can be obtained by deleting the other incoming arc of the hybridization vertex in Q .

Let \mathcal{T}_w and \mathcal{T}'_w be the rooted binary phylogenetic trees obtained from \mathcal{T} and \mathcal{T}' , respectively, by replacing the subtrees $\mathcal{T}|W$ and $\mathcal{T}'|W$ with a single vertex labelled w , where $w \notin X$. By Theorem 11.5,

$$h(\mathcal{T}, \mathcal{T}') = h(\mathcal{T}|W, \mathcal{T}'|W) + h(\mathcal{T}_w, \mathcal{T}'_w).$$

Since $\mathcal{T}|W$ is not isomorphic to $\mathcal{T}'|W$, we have that $h(\mathcal{T}|W, \mathcal{T}'|W) \geq 1$. But, by (ii), $h(\mathcal{T}|W, \mathcal{T}'|W) \leq 1$ and therefore $h(\mathcal{T}|W, \mathcal{T}'|W) = 1$. Consider $h(\mathcal{T}_w, \mathcal{T}'_w)$. Let \mathcal{H}_w denote the galled-tree obtained from \mathcal{H} by deleting each of the vertices that lie on a directed path from w except w itself. Since \mathcal{H} displays \mathcal{T} and \mathcal{T}' , it follows that \mathcal{H}_w displays \mathcal{T}_w and \mathcal{T}'_w . Now \mathcal{H}_w has k galls. Suppose that there is a galled-tree that displays \mathcal{T}_w and \mathcal{T}'_w , but has less galls than \mathcal{H}_w . Then one could use this network to obtain a galled-tree that displays \mathcal{T} and \mathcal{T}' by adjoining the subnetwork below w in \mathcal{H} to w resulting in a galled-tree with less galls than \mathcal{H} ; a contradiction to the minimality of \mathcal{H} . It now follows that amongst all galled-trees that display \mathcal{T}_w and \mathcal{T}'_w , the galled-tree \mathcal{H}_w has the smallest number of galls. By the induction assumption, this implies that $h(\mathcal{T}_w, \mathcal{T}'_w) = k$ and so

$$\begin{aligned} h(\mathcal{T}, \mathcal{T}') &= h(\mathcal{T}|W, \mathcal{T}'|W) + h(\mathcal{T}_w, \mathcal{T}'_w) \\ &= k + 1. \end{aligned}$$

This completes the proof of the proposition. \square

Nakhleh *et al.* [41] propose a method for inferring hybridization networks that allows for errors in the estimation of the initial two gene trees. In brief, when methods such as maximum likelihood or maximum parsimony infer trees there are a number of equally or close-to-equally good trees that could have been inferred. Thus the strict consensus of each such set of trees is perhaps a better representative of the original data set than one particular tree. However, this representative is typically unresolved, and so an interesting problem is the following. Given two rooted phylogenetic X -trees \mathcal{T}_1 and \mathcal{T}_2 , determine if there is two rooted binary phylogenetic X -trees \mathcal{T}'_1 and \mathcal{T}'_2 such that \mathcal{T}'_i is a refinement of \mathcal{T}_i with the property that there is a galled-tree that displays \mathcal{T}'_1 and \mathcal{T}'_2 . Moreover, if there is such a network, find \mathcal{T}'_1 and \mathcal{T}'_2 that minimizes the number of galls over all galled-trees that display \mathcal{T}'_1 and \mathcal{T}'_2 . In [41], the authors provide a linear-time algorithm for when the galled-tree contains exactly one gall. Huynh *et al.* [32] significantly extend this result by providing a quadratic-time algorithm for this problem with no restrictions on the number of galls in the resulting galled-tree. Moreover, they also show that this algorithm easily extends to an efficient algorithm for an arbitrary number of input trees. For further details, we refer the reader to [32].

Controlling the way in which hybridization events occur in a network is a possible avenue for further polynomial-time algorithms. Indeed, recent positive results by Huson *et al.* [31] suggest that this control could be done in a number of successful ways.

11.5 Recombination networks

The *perfect phylogeny with recombination* is a problem that has a very similar flavour to that of MINIMUM HYBRIDIZATION. Indeed, the two problems are closely related. Instead of inputting a collection of trees, the input for this problem is a collection, B say, of binary sequences. However, the goal is essentially the same. Loosely speaking, this goal is to compute the minimum number of “recombination” events to “explain” B . Introduced by Hein [24, 25], there is now a number of papers on this problem, including [5, 18–21, 49, 51–54]. In this section, we describe this problem and its relationship with MINIMUM HYBRIDIZATION. This relationship will be used in Section 11.7.

An (n, m) -*recombination network* \mathcal{N} is a rooted acyclic digraph with exactly n vertices of outdegree zero in which each vertex other than the root has either one or two incoming arcs, and each vertex of \mathcal{N} is labelled with a binary sequence of length m . The sequence labelling the root is called the *root* or *ancestral* sequence. A vertex with two incoming arcs is called a *recombination* vertex. Each integer in $\{1, 2, \dots, m\}$ is assigned to exactly one arc of \mathcal{N} that is not directed towards a recombination vertex. Beginning with the root and its associated sequence, each of the binary sequences labelling the other vertices is based on the binary sequence of its parent and the incoming arc (in the case it is a non-recombination vertex) or its parents (in the case it is a recombination vertex). In particular, the sequences satisfy the following properties:

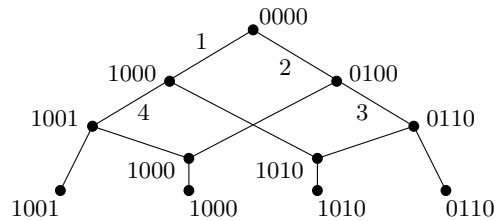


FIG. 11.14. A $(4,4)$ -recombination network in which the root sequence is the all-0 sequence.

- (i) If v is a non-recombination vertex with incoming arc e , then the sequence labelling v is obtained from the sequence labelling its parent by changing the i -th element (site) from 0 to 1 or 1 to 0 appropriately for each integer i assigned to e . If no integer is assigned to e , then the sequence labelling v is the same as its parent.
- (ii) If v is a recombination vertex, then, for some positive integer p strictly between 1 and m (that is, $1 < p < m$), the sequence labelling v is the concatenation of the first p elements of the sequence labelling one of its parents and the last $m - p$ elements of its other parent. To describe the corresponding recombination event one labels the incoming arcs either P or S depending upon which parent contributes the prefix part or the suffix part of the sequence, respectively, and also labels the recombination vertex with an ordered pair indicating the “break-point”.

Biologically speaking, the mutations in (i) are called *point mutations* and, as each site in the sequence mutates exactly once, we are under the so-called *infinite sites model* of mutations. The recombination process in (ii) is called a *single-crossover recombination* as there is exactly one break-point in the resulting sequence. Even though this model of recombination is very simple, it is the basis of most applications of coalescent theory to recombining sequences [27].

As an example, a recombination network is shown in Fig. 11.14, where the root sequence is the all-0 sequence. For each recombination vertex in this example, the first two elements in the associated sequence come from its ‘left’ parent and the second two elements come from its ‘right’ parent. (We have omitted the labelling of the recombination vertices and their incoming arcs as described in (ii) above.) In the literature, a recombination network is commonly referred to as a “phylogenetic network”.

Let B be a collection of n binary sequences of length m . An (n, m) -recombination network \mathcal{N} *explains* B if the n vertices of outdegree zero are bijectively labelled with the elements of B . For example, the recombination network in Fig. 11.14 explains the collection $\{1001, 1000, 1010, 0110\}$ of binary sequences. Over all recombination networks that explain B , we are interested in finding one that has the minimum number of recombination vertices. The perfect phylogeny with recombination problem is formally stated as follows.

PERFECT PHYLOGENY WITH RECOMBINATION

Instance: A set B of n binary sequences of length m .

Goal: Find an (n, m) -recombination network \mathcal{N} that explains B with minimum number of recombination vertices.

Measure: The number of recombination vertices in \mathcal{N} .

Depending upon whether the root sequence of the recombination network is specified or not specified in advance, the problem can be interpreted in one of two ways. If the root sequence is specified in advance, then, from a mathematical perspective, no generality is lost in always choosing the root sequence to be the all-0 sequence. We denote the minimum values for the two problems by $r(B)$ and $r^*(B)$, respectively, and note that $r^*(B) \leq r(B)$. The reason for the wording “perfect phylogeny” is that the classical perfect phylogeny problem can be interpreted as the problem of deciding if there is a recombination network with no recombination vertices that explains B .

Recombination events are one of the primary influences on genetic variation amongst individuals of the same population. Recognizing how many and where in the sequence these events occur is expected to be a contributing factor in answering a number of important problems in genetics including those centred around genetic diseases. Thus the motivation for PERFECT PHYLOGENY WITH RECOMBINATION is similar to that for MINIMUM HYBRIDIZATION except that our input is now a collection of binary sequences. SNP (single nucleotide polymorphism) sequences satisfy this criteria and are now of great interest (for example, see [28]). Each sequence represents an individual of the same population and, in such a sequence, each site represents an allele of the species. In the case the root sequence is specified in advance, a 0 denotes the ancestral allele, while a 1 denotes the derived (mutant) allele. Observe that $0 \rightarrow 1$ is the only allowable transition in this case.

There is a close relationship between MINIMUM HYBRIDIZATION and PERFECT PHYLOGENY WITH RECOMBINATION with the root sequence specified in advance. In particular, the former problem can be interpreted as a particular instance of the latter.

Using the construction in Wang *et al.* [54], let \mathcal{T} and \mathcal{T}' be two rooted binary phylogenetic X -trees and let $|X| = n$. Noting that $|E(\mathcal{T})| = |E(\mathcal{T}')| = 2(n - 1)$, bijectively label the edges of \mathcal{T} and \mathcal{T}' with the elements of $\mathcal{C} = \{\chi_1, \chi_2, \dots, \chi_{2(n-1)}\}$ and $\mathcal{C}' = \{\chi'_1, \chi'_2, \dots, \chi'_{2(n-1)}\}$, respectively. Each of the elements in \mathcal{C} and \mathcal{C}' represent a site. Associated to each vertex v (resp. v') of \mathcal{T} (resp. \mathcal{T}') is the binary sequence of length $2(n - 1)$ in which the i -th element is 1 if and only if χ_i (resp. χ'_i) labels an edge from the root of \mathcal{T} (resp. \mathcal{T}') to v (resp. v'). Now, for each $x \in X$, concatenate the sequences labelling x in \mathcal{T} and \mathcal{T}' with the sequence labelling x in \mathcal{T}' following the sequence labelling x in \mathcal{T} . Let B be the resulting collection of n (concatenated) sequences of length $4(n - 1)$. The following theorem due to Bordewich and Semple [13] provides the above mentioned close relationship.

Theorem 11.9 *Let \mathcal{T} and \mathcal{T}' be two rooted binary phylogenetic X -trees, and let B be the collection of binary sequences that is constructed from \mathcal{T} and \mathcal{T}' as above. Then*

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

The proof of Theorem 11.9 is constructive. In particular, if \mathcal{H} is a minimum hybridization network that displays \mathcal{T} and \mathcal{T}' , then there is a polynomial-time modification of \mathcal{H} that results in a recombination network \mathcal{N} that explains B with the all-0 sequence at the root and has $h(\mathcal{H})$ recombination vertices. On the other hand, if \mathcal{N} is a recombination network explaining B with the all 0-sequence at the root and k recombination vertices, then \mathcal{N} can be modified to produce a hybridization network that displays \mathcal{T} and \mathcal{T}' with k hybridization vertices. Again, this modification can be done in polynomial time.

Remark In this section, we have restricted ourselves to single-crossover recombinations. However, we note here that more general recombinations called *multiple-crossover* recombinations have also been considered (for example, see [18, 19, 31]). Here, if v denotes the recombination vertex, then the sequence labelling v has the weaker property that, for all i , the i -th element in the sequence is the same as the i -th element in at least one of the parent sequences. By specifying, for all i , which parent the i -th element came from, the number of crossover events is equal to the number of pairs $(j, j + 1)$ in which the j -th element comes from one parent while the $(j + 1)$ -th element comes from the other parent. Extending the definition of an (n, m) -recombination network in the obvious way to allow for multiple-crossover events, the “goal” of the optimization problem analogous to PERFECT PHYLOGENY WITH RECOMBINATION could be interpreted in one of two ways. Namely, minimize the number of recombination vertices in a network that explains an initial set B of binary sequences or minimize the number of crossover events in a network that explains B . While the first interpretation has received a reasonable amount of attention, the second interpretation appears to have received little attention.

11.6 Hybridization networks in real time

An important biological requirement of hybridization networks is that hybridization events occur between contemporaneous taxa (past or present). Maddison [35] pointed out this requirement and, from a mathematical perspective, it has been considered in several papers since including [8, 39, 51, 53]. We begin this section by considering the problem of whether a given hybridization network is consistent with this requirement.

11.6.1 Temporal representations

Let \mathcal{H} be a hybridization network with vertex set V , and let $\mathbb{N} = \{0, 1, 2, \dots\}$. We say that \mathcal{H} has a *temporal representation* if there is a map $f : V \rightarrow \mathbb{N}$ that satisfies the following two properties:

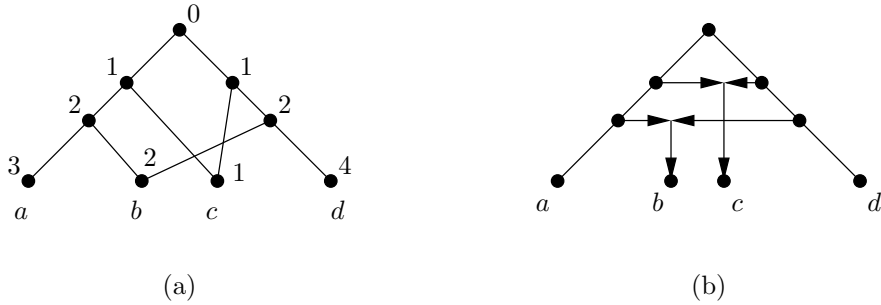


FIG. 11.15. (a) A temporal labelling of a hybridization network and (b) a “real time” realization of this labelling.

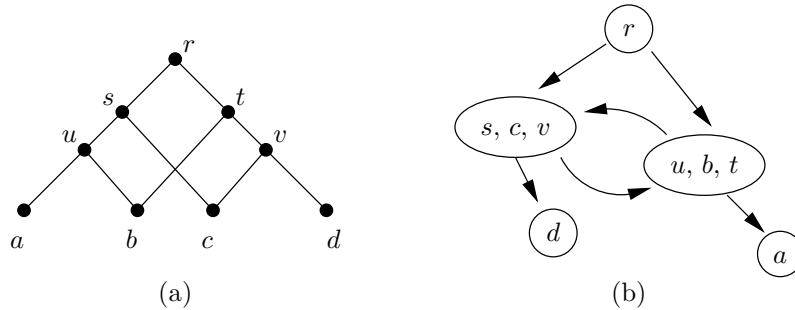


FIG. 11.16. (a) A hybridization network with no temporal representation and (b) its temporal digraph.

- (i) If (u, v) is an arc of \mathcal{H} with $d^-(v) = 1$, then $f(u) < f(v)$.
- (ii) If (u, v) is an arc of \mathcal{H} with $d^-(v) \geq 2$, then $f(u) = f(v)$.

Such a map f is called a *temporal labelling* of \mathcal{H} . The purpose of (ii) is so that hybridization events occur with contemporaneous taxa. A temporal labelling of a hybridization network is shown in Fig. 11.15(a). A “real time” realization of this labelled network is shown in Fig. 11.15(b).

All rooted phylogenetic trees have a temporal representation, but not all hybridization networks have such a representation. For example, the hybridization network in Fig. 11.16(a) has no temporal representation. The reason for this is that u and t , the parents of b , must coexist in time, while s and v , the parents of c , must also coexist in time. By considering the ancestor-descendant relationships of s and u , and t and v this is not possible.

We next describe a simple polynomial-time algorithm for deciding whether a hybridization network has a temporal representation and, if so, constructs such a representation. Due to Baroni *et al.* [8], we begin by defining a particular digraph around which the algorithm is based. Let \mathcal{H} be a hybridization network

with vertex set V . Ignoring the direction of the arcs of \mathcal{H} , set

$$[v] = \{v\} \cup \{u \in V : \text{there is a path of hybridization arcs from } u \text{ to } v\},$$

where a *hybridization arc* is an arc that is directed into a hybridization vertex.

Note that we have partitioned V into equivalence classes, where $[v] = \{v\}$ precisely if v is not incident with a hybridization arc. Setting $[V] = \{[v] : v \in V\}$, we define the *temporal digraph* of \mathcal{H} as the digraph whose vertex set is $[V]$ and where $[u]$ and $[v]$ are joined by an arc $([u], [v])$ if there is a vertex a in $[u]$ and a vertex b in $[v]$ such that (a, b) is an arc of \mathcal{H} with $d^-(b) = 1$. For example, the digraph in Fig. 11.16(b) is the temporal digraph of the hybridization network in Fig. 11.16(a).

It turns out that \mathcal{H} has a temporal representation if and only if its temporal digraph is acyclic and this is the basis of the following algorithm whose correctness is shown in [8].

Algorithm: TEMPREP(\mathcal{H})

Input: A hybridization network \mathcal{H} with vertex set V .

Output: A temporal labelling of \mathcal{H} or the statement \mathcal{H} has no temporal labelling.

1. Construct the temporal digraph $D_{\mathcal{H}}$ of \mathcal{H} .
2. Find an acyclic ordering, V_0, V_1, \dots, V_k say, of $D_{\mathcal{H}}$. If there is no such ordering, then return \mathcal{H} has no temporal representation.
3. Define $f : V \rightarrow \mathbb{N}$ by setting $f(v) = i$ for all $v \in V$, where $[v] \in V_i$.
4. Return the map f .

If a map f is returned by the algorithm, then f is a temporal labelling of \mathcal{H} . It is important to note that a temporal labelling of a hybridization network is no more than an ordering of when past or present taxa appeared. Consequently, it is the ordering on the vertices of V that is important and not the actual values.

If one is interested in obtaining, up to isomorphism, all temporal labellings of \mathcal{H} , then the above algorithm can be easily modified to output a list of all such labellings, where a new labelling is outputted in polynomial time and where two labellings are non-isomorphic if the relative orderings of the vertices are not the same. Essentially, one selects non-empty subsets of vertices that have indegree zero instead of a single vertex in the process of finding an acyclic ordering. All such orderings result in a distinct temporal labelling and all such labellings can be obtained this way. For further details, see [8].

We end this subsection with the following remark. If a hybridization network \mathcal{H} does not have a temporal representation, then Moret *et al.* [39] observed that, by allowing for missing taxa, one could resolve this issue without adding to the hybridization number of \mathcal{H} . For example, consider the hybridization network in Fig. 11.16(a). By creating two new vertices that subdivide the arcs (t, b) and (s, c) , and joining pendant arcs to each of these new vertices with new taxa, the resulting hybridization network has a temporal representation. The role of such

taxa is to carry a gene or combination of genes from the past into some time when it can be passed on into the new hybrid species. Of course, whether such taxa exist or existed is a separate question.

11.6.2 Time ordered rooted subtree prune and regraft operations

Realizing the importance that time places on possible scenarios for evolutionary histories, Song and Hein [51, 53] (also see [27]) considered a more restrictive notion of the rooted subtree prune and regraft operation. This restriction allows one to attack the problem of PERFECT PHYLOGENY WITH RECOMBINATION in which the root sequence is not specified in advance using rooted subtree prune and regraft operations.

Let \mathcal{T} be a rooted binary phylogenetic tree and let $\mathring{V} = \{v_1, v_2, \dots, v_{n-2}\}$ be the set of interior vertices of \mathcal{T} . A *total ordering* on \mathring{V} is a binary relation $<_{\mathcal{T}}$ given by $v_i <_{\mathcal{T}} v_j$ if the hypothetical ancestor or speciation event represented by v_i predates the hypothetical ancestor or speciation event represented by v_j . In mathematics, total orderings are also called *linear orderings*. We say that \mathcal{T} is *ordered* if \mathring{V} is totally ordered. By default, such an ordering must preserve the ancestor-descendant relationships given by the topology of \mathcal{T} .

In performing a rooted subtree prune and regraft operation on an ordered tree \mathcal{T} one must preserve the ordering on \mathring{V} . In particular, referring to the notation in the definition of a rSPR operation in Section 11.3, for all $v_i, v_j \in \mathring{V} - \{u\}$, we have that $v_i <_{\mathcal{T}'} v_j$ precisely if $v_i <_{\mathcal{T}} v_j$, where u is the ‘parent’ vertex of the root of the subtree being pruned, \mathcal{T} is the initial tree, and \mathcal{T}' is the tree resulting from the rSPR operation. Given two ordered rooted binary phylogenetic X -trees, there is a sequence of (ordered) rSPR operations that transforms one tree into the other. For further combinatorial results on this operation and the ordinary rSPR operation, see Song [47, 48].

Now let B be a collection of binary sequences of equal length m . For each i , the i -th sites in the sequences induce a character χ_i . Under the infinite-sites model of mutation, let \mathcal{P}_i be the collection of ordered rooted binary phylogenetic X -trees that display χ_i , that is \mathcal{P}_i is the collection of all such trees for which there exists an edge whose deletion induces the bipartition of X induced by the character states in χ_i . Consider the problem of minimizing the following sum:

$$\sum_{i=1}^{m-1} d_{\text{rSPR}}(\mathcal{T}_i, \mathcal{T}_{i+1}), \quad (11.2)$$

where $\mathcal{T}_i \in \mathcal{P}_i$ for all i and $d_{\text{rSPR}}(\mathcal{T}_i, \mathcal{T}_{i+1})$ denotes the minimum number of (ordered) rSPR operations to transform \mathcal{T}_i into \mathcal{T}_{i+1} . It turns out that the minimum value of this sum is equal to $r^*(B)$, the optimal value of PERFECT PHYLOGENY WITH RECOMBINATION in which the root sequence is not specified in advance [50]. Thus $r^*(B)$ can be written in terms of the rSPR distance on ordered rooted binary phylogenetic trees. Moreover, a lower bound for $r^*(B)$ can be obtained by interpreting the terms in the sum in (11.2) as the ordinary rSPR

distance between two rooted binary phylogenetic trees, where the total ordering on the interior vertices is ignored.

The number of ordered rooted binary phylogenetic trees grows significantly faster than the number of (ordinary) rooted binary phylogenetic trees, and so as it currently stands the above approach to computing $r^*(B)$ exactly is limiting in practice. Nevertheless, by studying a particular data set for which previous lower bounds have been calculated, Song and Hein have shown it can work. For further details, see [51, 53] and note that Song and Hein use the terminology “ancestral recombination graph” instead of recombination network.

11.7 Computational complexity

In this section, we discuss some of the computational issues associated with the three main problems that we have discussed in this chapter, namely MINIMUM HYBRIDIZATION, MINIMUM RSPR, and PERFECT PHYLOGENY WITH RECOMBINATION. Throughout this section, the interpretation of the last of these problems will always be the one in which the root sequence is specified in advance.

The following theorem, which we have alluded to several times in this chapter, is due to Bordewich and Semple [12, 13].

Theorem 11.10 *Each of the optimization problems MINIMUM HYBRIDIZATION, MINIMUM RSPR, and PERFECT PHYLOGENY WITH RECOMBINATION is NP-hard.*

The proofs of the NP-hardness of MINIMUM HYBRIDIZATION and MINIMUM RSPR make use of their characterizations in terms of agreement forests and use ideas originating from Hein *et al.* [26]. The NP-hardness of PERFECT PHYLOGENY WITH RECOMBINATION follows from Theorem 11.9 and the polynomial-time constructions mentioned after it. To avoid repetition, these comments are also valid for Theorem 11.11.

Despite the negativity of Theorem 11.10, there are some positive results for these problems. Fixed-parameter algorithms are a practical way to find optimal solutions of NP-hard problems if the parameter measuring the hardness of the problem is small. For MINIMUM RSPR, Bordewich and Semple [12] showed that there is such an algorithm where the rSPR distance itself is the parameter. In particular, instead of computing the rSPR distance between two rooted binary phylogenetic X -trees by an exhaustive search resulting in an algorithm that takes time $O((2n)^{2k})$ where $n = |X|$ and $k = d_{\text{rSPR}}(\mathcal{T}, \mathcal{T}')$, they showed that there is a parameterized algorithm for computing this distance in $O(f(k) \cdot p(n))$ where $f(k)$ is some computable function depending on k and p is a polynomial in n . The important point of this running time is that n and k are now separated which means that, provided k is small, computing $d_{\text{rSPR}}(\mathcal{T}, \mathcal{T}')$ may be efficiently possible even when n is large. The important part of the analysis is Theorem 11.4.

Translating the setting in [23], Hallet and Lagergren give a fixed-parameter algorithm for a problem that is a restriction of MINIMUM HYBRIDIZATION (also see [1] for a description of the algorithm). Parameterized by the hybridization

number of the two trees, Bordewich and Semple [14] recently gave a fixed-parameter algorithm for MINIMUM HYBRIDIZATION in general. For further details of this last algorithm and an analysis of how well it works in practice, we refer the interested reader to [14] and [11], respectively. For those wanting to find out more about fixed-parameter algorithms, we refer the reader to [16] and [30]. The latter is an easy-to-read introduction to fixed-parameter algorithms and describe three techniques for developing such algorithms.

For computationally hard problems, polynomial-time approximation algorithms can efficiently find feasible solutions that are sometimes arbitrarily close to the optimal solution. In particular, for a minimization problem, an r -approximation algorithm means that, for all instances, the ratio of the size of the feasible solution outputted by the algorithm and the size of an optimal solution is at most r . The existence of polynomial-time approximation algorithms varies greatly amongst NP-hard problems. For example, regardless of the choice of r , there is no such algorithm for the general travelling salesman problem unless $P=NP$, while for some problems π , no matter how close r is to 1, there is always such an algorithm. In this latter case, we say that π exhibits a *polynomial-time approximation scheme* (PTAS). Theorem 11.11 is due to Bordewich and Semple [13].

Theorem 11.11 *Each of the optimization problems MINIMUM HYBRIDIZATION, MINIMUM RSPR, and PERFECT PHYLOGENY WITH RECOMBINATION is APX-hard. In particular, for each of these problems there is no polynomial-time approximation scheme unless $P=NP$.*

For each of our optimization problems, the implication of Theorem 11.11 is that, unless $P=NP$, there is some fixed constant r strictly bigger than 1 for which there is no polynomial-time r -approximation algorithm. It is shown in [13] that, for each of these problems, r is at least $\frac{2113}{2112}$.

Two polynomial-time approximation algorithms for MINIMUM RSPR have appeared in the literature [26, 43]. Both are stated as 3-approximation algorithms, however, each of these algorithms have been subsequently shown to be incorrect in some way. Nevertheless, using these approaches, Bonet *et al.* [10] describe a polynomial-time 5-approximation algorithm for MINIMUM RSPR. Intuitively, this algorithm builds an agreement forest locally. Currently, there appears to be no such algorithm for MINIMUM HYBRIDIZATION. One might hope that the algorithm in [10] extends to MINIMUM HYBRIDIZATION, but, due to the additional global condition on an acyclic-agreement forest, it seems unlikely that such an approach will work. For an excellent reference on approximation algorithms, see [4].

11.8 Concluding remarks

The understanding and analysis of reticulation in evolution is playing a prominent role in modern-day phylogenetics. In this chapter, we considered one particular, but central, aspect; namely the problem of finding the smallest number of reticulation events that are required to explain the evolution of a collection

of species under consideration subject to some initial input. For us, the input was a collection of rooted phylogenetic trees. The approach we have taken here is analytical so as to provide a theoretical foundation for algorithmic solutions to the problem. Furthermore, our main interest has been on a general solution rather than one that is restricted in some way. Unfortunately, despite the fixed-parameter algorithms for MINIMUM RSPR and MINIMUM HYBRIDIZATION, and the divide-and-conquer algorithm for MINIMUM HYBRIDIZATION described in this chapter, we are always going to be limited in finding exact solutions because of the NP-hardness of these problems. This turns our attention to future work.

A number of papers have considered efficient algorithms for computing lower bounds for PERFECT PHYLOGENY WITH RECOMBINATION (for example, see [5, 22, 29, 38, 52]). While one could use the constructions outlined after Theorem 11.9 and these results, it appears that little attention has been given to finding such algorithms directly for MINIMUM RSPR and MINIMUM HYBRIDIZATION. Given the incorrectness of previous approximations for MINIMUM RSPR, a mathematically challenging task is to improve the 5-approximation algorithm for this problem. Whether MINIMUM HYBRIDIZATION even has such an algorithm, regardless of the size of the ratio, is an interesting question.

In this chapter, we have only considered combinatorial questions. While a combinatorial understanding of reticulation is far from complete, it is statistical questions that will eventually need to be addressed. For example, how can one use differing bootstrap support values for conflicting phylogenies to quantify and distinguish between genuine reticulation and other biological processes that give rise to conflicts such as lineage sorting? Combinatorial considerations are often the first steps towards statistical-based approaches in phylogenetics and so it is highly likely that combinatorial insights into hybridization networks will aid the development of such approaches to reticulation.

Acknowledgments

Many thanks to Peter Lockhart, Katherine St. John, and Yun Song for a number of helpful discussions during the writing of this chapter, and Simone Linz for providing the figures for the grass data set example in Section 11.4.2. This work was supported by the New Zealand Marsden Fund (UOC310).

REFERENCES

- [1] Addario-Berry, L., Hallett, M., and Lagergren, J. (2003). Towards identifying lateral gene transfer events. In: *Proceedings of the Pacific Symposium on Biocomputing*, Vol. 8, pp. 279-290.
- [2] Allan, H. H. (1961). *Flora of New Zealand, Volume I, Indigenous tracheophyta: Psilopsida, Lycopsida, Filicopsida, Gymnospermae, Dicotyledones*. Government Printer, Wellington, New Zealand.
- [3] Allen, B. L. and Steel, M. (2001). Subtree transfer operations and their induced metrics on evolutionary trees. *Annals of Combinatorics*, **5**, 1-13.
- [4] Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A., and Protasi, M. (1999). *Complexity and Approximation*. Springer, Berlin.
- [5] Bafna, V. and Bansal, V. (2004). The number of recombination events in a sample history: conflict graph and lower bounds. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **1**, 78-90.
- [6] Baroni, M. (2004). Hybrid phylogenies: a graph-based approach to represent reticulate evolution. Unpublished PhD thesis, University of Canterbury.
- [7] Baroni, M., Semple, C., and Steel, M. (2004). A framework for representing reticulate evolution. *Annals of Combinatorics*, **8**, 391-408.
- [8] Baroni, M., Semple, C., and Steel, M. (2006). Hybrids in real time. *Systematic Biology*, **55**, 46-56.
- [9] Baroni, M., Grünewald, S., Moulton, V., and Semple, C. (2005). Bounding the number of hybridization events for a consistent evolutionary history. *Mathematical Biology*, **51**, 171-182.
- [10] Bonet, M. K., St. John, K., Mahindru, R., and Amenta, N. (2006). Approximating subtree distances between phylogenies. Technical Report #669, Centre de Recerca Matemàtica, Barcelona.
- [11] Bordewich, M., Linz, S., St. John, K., and Semple, C. A reduction algorithm for computing the hybridization number of two trees, submitted.
- [12] Bordewich, M. and Semple, C. (2004). On the computational complexity of the rooted subtree prune and regraft distance. *Annals of Combinatorics*, **8**, 409-423.
- [13] Bordewich, M. and Semple, C. Computing the minimum number of hybridization events for a consistent evolutionary history. *Discrete Applied Mathematics*, in press.
- [14] Bordewich, M. and Semple, C. Computing the hybridization number of two phylogenetic trees is fixed-parameter tractable. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, in press.
- [15] Doolittle, W. F. (1999). Phylogenetic classification and the universal tree. *Science*, **284**, 2124-2128.

- [16] Downey, R. and Fellows, M. (1998). *Parameterized Complexity*. Springer, New York.
- [17] Ellstrand, N. C., Whitkus, R., and Rieseberg, L. H. (1996). Distribution of spontaneous plant hybrids. *Proceedings of the National Academy of Sciences*, **93**, 5090-5093.
- [18] Gusfield, D. (2005). Optimal, efficient reconstruction of root-unknown phylogenetic networks with constrained and structured recombination. *Journal of Computer and System Sciences*, **70**, 381-398.
- [19] Gusfield, D. and Bansal, V. (2005). A fundamental decomposition theory for phylogenetic networks and incompatible characters. In: *Proceedings of the Ninth Annual International Conference on Research in Computational Molecular Biology (RECOMB 2005)* (ed. S. Miyano *et al.*), Lecture Notes in Bioinformatics, Vol. 3500, Springer, Berlin, pp. 217-232
- [20] Gusfield, D., Eddhu, S., and Langley, C. (2004). Optimal, efficient reconstruction of phylogenetic networks with constrained recombination. *Journal of Bioinformatics and Computational Biology*, **2**, 173-213.
- [21] Gusfield, D., Eddhu, S., and Langley, C. (2004). The fine structure of galls in phylogenetic networks. *INFORMS Journal on Computing*, **16**, 459-469.
- [22] Gusfield, D., Hickerson, D., and Eddhu, S. An efficiently-computed lower bound on the number of recombinations in phylogenetic networks: theory and empirical study. *Discrete Applied Mathematics*, in press.
- [23] Hallett, M. and Lagergren, J. (2001). Efficient algorithms for lateral gene transfer problems. In: *Proceedings of the Fifth Annual International Conference on Research in Computational Molecular Biology (RECOMB 2001)*, ACM Press, New York, pp. 149-156.
- [24] Hein, J. (1990). Reconstructing evolution of sequences subject to recombination using parsimony. *Mathematical Biosciences*, **98**, 185-200.
- [25] Hein, J. (1993). A heuristic method to reconstruct the history of sequences subject to recombination. *Journal of Molecular Evolution*, **36**, 396-405.
- [26] Hein, J., Jing, T., Wang, L., and Zhang, K. (1996). On the complexity of comparing evolutionary trees. *Discrete Applied Mathematics*, **71**, 153-169.
- [27] Hein, J., Schierup, M., and Wiuf, C. (2005). *Gene Genealogies, Variation and Evolution: A Primer in Coalescent Theory*. Oxford University Press.
- [28] Hinds, D., Stuve, L., Nilsen, G., Halperin, E., Eskin, E., Gallinger, D., Frazer, K., and Cox, D. (2005) Whole-genome patterns of common DNA variation in three human populations. *Science*, **307**, 1072-1079.
- [29] Hudson, R. and Kaplan, N. (1985). Statistical properties of the number of recombination events in the history of a sample of DNA sequences. *Genetics*, **111**, 147-164.
- [30] Hüffner, F., Niedermeier, R., and Wernick, S. Techniques for practical fixed-parameter algorithms, submitted.
- [31] Huson, D. H., Klöpper, T., Lockhart, P. J., and Steel, M. (2005). Reconstruction of reticulate networks from gene trees. In: *Proceedings of the Ninth Annual International Conference on Research in Computational Molecular*

- Biology* (RECOMB 2005) (ed. S. Miyano *et al.*), Lecture Notes in Bioinformatics, Vol. 3500 Springer, Berlin, pp. 233-249.
- [32] Huydn, T. N. D., Jansson, J., Nguyen, N. B., and Sung, W. -K. (2005). Constructing a smallest refining galled phylogenetic network. In: *Proceedings of the Ninth Annual International Conference on Research in Computational Molecular Biology* (RECOMB 2005) (ed. S. Miyano *et al.*), Lecture Notes in Bioinformatics, Vol. 3500 Springer, Berlin, pp. 265-280.
- [33] Jansson, J., Nguyen, N. B., and Sung, W. -K. (2006). Algorithms for combining rooted triples into a galled phylogenetic network. *SIAM Journal on Computing*, **35**, 1098-1121.
- [34] Linz, S. Reticulation in evolution. Unpublished PhD thesis, Heinrich-Heine Universität, in preparation.
- [35] Maddison, W. (1997). Gene trees in species trees. *Systematic Biology*, **46**, 523-536.
- [36] Mallet, J. (2005). Hybridization as an invasion of the genome. *Trends in Ecology and Evolution*, **20**, 229-237.
- [37] McBreen, K. and Lockhart, P. J. (2007). Reconstructing reticulate evolutionary histories of plants. *Trends in Plant Science*, **11**, 398-404.
- [38] Myers, S. and Griffiths, R. (2003). Bounds on the minimum number of recombination events in a sample history. *Genetics*, **163**, 375-394.
- [39] Moret, B. M. E., Nakhleh, L., Warnow, T., Linder, C. R., Tholse, A., Padolina, A., Sun, J., and Timme, R. (2004). Phylogenetic networks: modeling, reconstructibility, and accuracy. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **1**, 1-11.
- [40] Nakhleh, L., Ruths, D., and Wang, L. S. (2005). RIATA-HGT: a fast and accurate heuristic for reconstructing horizontal gene transfer. In: *Proceedings of the Eleventh International Computing and Combinatorics Conference* (COCOON 05) (ed. L. Wang), Lecture Notes in Computer Science, Vol. 3595, Springer, pp. 84-93.
- [41] Nakhleh, L., Warnow, T., Linder, C. R., and St. John, K. (2005). Reconstructing reticulate evolution in species—theory and practice. *Journal of Computational Biology*, **12**, 796-811.
- [42] Olsen, G. J., Matsuda, H., Hagstrom, R., and Overbeek, R. (1994). fastDNAmL: a tool for construction of phylogenetic trees of DNA sequences using maximum likelihood. *Computing Applications in the Biosciences*, **10**, 41-48.
- [43] Rodrigues, E. M., Sagot, M. -F., and Wakabayashi, Y. (2001). Some approximation results for the maximum agreement forest problem. In: *Approximation, Randomization and Combinatorial Optimization: Algorithms and Techniques* (APPROX and RANDOM) (ed. M. Goemans *et al.*), Lecture Notes in Computer Science, Vol. 2129, Springer, Berlin, pp. 159-169.
- [44] Schmidt, H. A. (2003). Phylogenetic trees from large data sets. Unpublished PhD thesis, Heinrich-Heine Universität.
- [45] Semple, C. and Steel, M. (2003). *Phylogenetics*. Oxford University Press.

- [46] Semple, C. and Steel, M. (2006). Unicyclic networks: compatibility and enumeration. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **3**, 84-91.
- [47] Song, Y. S. (2003). On the combinatorics of rooted binary phylogenetic trees. *Annals of Combinatorics*, **7**, 365-379.
- [48] Song, Y. S. (2006). Properties of subtree-prune-and-regraft operations on totally-ordered phylogenetic trees. *Annals of Combinatorics*, **10**, 147-163.
- [49] Song, Y. S. (2006). A concise necessary and sufficient condition for the existence of a galled-tree. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **3**, 186-191.
- [50] Song, Y. S. (2006). Private communication.
- [51] Song, Y. S. and Hein, J. (2003). Parsimonious reconstruction of sequence evolution and haplotype blocks: finding the minimum number of recombination events. In: *Algorithms in Bioinformatics (WABI 2003)* (ed. G. Benson and R. Page), Lecture Notes in Bioinformatics, Vol. 2812, Springer, Berlin, pp. 287-302.
- [52] Song, Y. S. and Hein, J. (2004). On the minimum number of recombination events in the evolutionary history of DNA sequences. *Journal of Mathematical Biology*, **48**, 160-186.
- [53] Song, Y. S. and Hein, J. (2005). Constructing minimal ancestral recombination graphs. *Journal of Computational Biology*, **12**, 147-169.
- [54] Wang, L., Zhang, K., and Zhang, L. (2001). Perfect phylogenetic networks with recombination. *Journal of Computational Biology*, **8**, 69-78.