Optimizing tree and character compatibility across several phylogenetic trees

Simone Linz[a,b], Katherine St. John[c], and Charles Semple[a]

[a] *Biomathematics Research Centre, Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand.*
[b] *Center for Bioinformatics, University of Tübingen, Germany.*
[c] *Department of Mathematics and Computer Science, Lehman College, City University of New York, Bronx, NY, United States.*

Abstract

Given a set $\mathcal{R}$ of rooted phylogenetic trees on overlapping taxa, it takes polynomial time to decide whether or not there exists a rooted phylogenetic tree that is compatible with $\mathcal{R}$. Since not all evolutionary histories for a set of species can be explained by a single tree, it is natural to ask for the minimum number of rooted phylogenetic trees needed such that each tree in $\mathcal{R}$ is compatible with at least one tree. This paper shows that it is computationally hard to compute this minimum number. In particular, if $\mathcal{R}$ contains rooted triples (rooted binary phylogenetic trees on three leaves), it is NP-complete to decide whether there exist two rooted phylogenetic trees such that each rooted triple in $\mathcal{R}$ is compatible with at least one of the two trees. Furthermore, for a set Σ of binary characters and a positive integer k, we show that to decide if there exists a set $\mathcal{P}$ of k rooted phylogenetic trees such that each character in Σ is compatible with at least one tree in $\mathcal{P}$ is NP-complete for all $k \geq 3$, but solvable in polynomial time for $k = 2$. This generalizes the result for $k = 1$, where it is well-known to be polynomial time.

Keywords: character, compatibility, phylogenetic tree, rooted triple

1. Introduction to tree and character compatibility

Ever since Charles Darwin laid the foundations for our current understanding of evolution, biologists have been interested in the reconstruction of phylogenetic (evolutionary) trees that correctly represent the ancestral history of a set of species. While Darwin was mainly aiming at illuminating the diversity he observed in nature, nowadays the study of phylogenetic trees also influences new research fields such as metagenomics [10], as well as medical related questions since, for example, a profound knowledge of the evolution of a certain pathogen (e.g. the influenza virus [11]) is often an essential first step in the development of any medication against it.

With the growing amount of available molecular sequence data, researchers now reconstruct phylogenetic trees with several thousand leaves, and in doing so frequently make use of *supertree algorithms* [3] as they allow one to easily amalgamate the analyses of different studies. Supertree methods combine a set of source trees on overlapping taxa sets into a single parent tree. If this tree captures all the ancestral information of the source trees, then the source trees are compatible and do not contain any contradictory information. In the context of rooted phylogenetic trees, it is computationally easy (i.e. polynomial-time solvable) to decide if a set of rooted phylogenetic trees is compatible and, if so, to construct a parent tree [1, 12]. In contrast, the problem is NP-complete for source trees that are unrooted [13]. A similar dichotomy also exists for phylogenetic analyses whose initial input is a set of characters rather than a set of trees. Characters describe attributes of the species under consideration and can be morphological (e.g. wings versus no-wings) or molecular (e.g. the nucleotide at a certain position on a DNA sequence). Informally, a set of characters is compatible if there is a phylogenetic tree $\mathcal{T}$ that realizes each character c in the set without any reverse or convergent character state transitions, in which case c is said to be *convex* on $\mathcal{T}$ (for a formal definition, see Section 2). For a motivating example of a character compatibility study, we refer the interested reader to Holland et al. [8], who have

Email addresses: `linz@informatik.uni-tuebingen.de` (Simone Linz), `stjohn@lehman.cuny.edu` (Katherine St. John), `charles.semple@canterbury.ac.nz` (and Charles Semple) recently investigated the concerted evolution of cormorants and shags by analyzing groups of mutually compatible characters. While a polynomial-time algorithm exists to determine if a set of 2-state characters is compatible and, if so, to reconstruct a tree on which each character in the input is convex [7], this problem becomes NP-complete for r-state characters if r is unbounded [4, 13].

Historically, given an initial input, one constructs a single phylogenetic tree to represent the ancestral history of the present-day species under consideration. Ideally, the resulting tree is consistent with the input, that is, the input is compatible. However, in practice, sets of rooted phylogenetic trees and 2-state characters are rarely compatible. For many sets of species, this incompatibility is not due to some type of error, but reflects the fact that the species' ancestral past cannot be explained by a single tree due to processes such as horizontal gene transfer or hybridization. To this end, we investigate the problem of calculating the minimum size of a set $\mathcal{P}$ of phylogenetic trees such that each tree (resp. character) in a set of rooted phylogenetic trees (resp. 2-state characters) is compatible with at least one tree in $\mathcal{P}$ and settle several questions related to the complexity of this problem. Details of this problem and the questions addressed are formally described in the next section. Lastly, we remark that Baroni et al. [2] have investigated a similar problem that asks for when a set of so-called (incompatible) clusters can be explained by two rooted phylogenetic trees.

2. Preliminaries

This section provides preliminary definitions and formally states several problems whose complexity is analyzed in the subsequent two sections. Unless otherwise stated, notation and terminology follow [12].

Phylogenetic trees. A phylogenetic X-tree $\mathcal{T}$ is a tree whose internal vertices have degree at least three and whose leaf set is X. In the rooted setting, a *rooted phylogenetic X-tree* $\mathcal{T}$ is a rooted tree in which the root has degree at least two, all other interior vertices have degree at least three, and whose leaf set is X. Furthermore, a rooted phylogenetic X-tree is *binary* if, apart from the root which has degree two, all other interior vertices have degree three. The leaf set X of a (rooted) phylogenetic tree $\mathcal{T}$ is often referred to as the *label set* of $\mathcal{T}$ and is denoted by $\mathcal{L}(\mathcal{T})$. For a collection $\mathcal{R}$ of (rooted) phylogenetic trees, we use $\mathcal{L}(\mathcal{R})$ to denote the union $\bigcup_{\mathcal{T}\in\mathcal{R}} \mathcal{L}(\mathcal{T})$. In Section 3, we will exclusively deal with rooted phylogenetic trees, while the material presented in Section 4 is concerned with (unrooted) phylogenetic trees.

Let $\mathcal{T}$ be a rooted phylogenetic X-tree, and let $\{u, v\}$ be an edge in $\mathcal{T}$ such that u lies on the path from the root of $\mathcal{T}$ to v. We say that u is the *parent* of v and v is a *child* of u. Now, let X' be a subset of X. The *restriction of $\mathcal{T}$ to X'*, denoted by $\mathcal{T}|X'$, is the rooted phylogenetic tree that is obtained from the smallest subtree of $\mathcal{T}$ that connects all leaves labeled with elements in X' by contracting non-root degree-2 vertices. Further, the *most recent common ancestor* of X' in $\mathcal{T}$, denoted by $\mathrm{mrca}_{\mathcal{T}}(X')$, is the vertex v in $\mathcal{T}$ such that

(i) the set of leaf descendants of v is a superset of X', and
(ii) there exists no vertex v' in $\mathcal{T}$ that is a descendant of v and whose set of leaf descendants is a superset of X'.

Lastly, let x_1 and x_2 be two elements, and let $O = (x_3, x_4, \ldots, x_n)$ be a tuple. Furthermore, let $\mathcal{T}$ be a rooted phylogenetic tree with label set $\{x_1, x_2, \ldots, x_n\}$. If x_1 and x_2 have the same parent in $\mathcal{T}$ and if, for each $i \in \{2, 3, \ldots, n-1\}$, the parent of x_i is a child of the parent of x_{i+1}, then $\mathcal{T}$ is said to be the *caterpillar on* $(x_1, x_2, \ldots, x_n)$ or, for short, (x_1, x_2, O).

Rooted triples. A rooted binary phylogenetic tree on three leaves is called a *rooted triple*. A rooted triple with leaves labeled a, b, and c is denoted $ab|c$ (or, equivalently, $ba|c$), if the path from a to b does not intersect the path from c to the root.

Characters. An *r-state full character* on X (or *r-state character* for short) is a function $c : X \to C$, where $C = \{s_1, s_2, \ldots, s_r\}$ is a collection of character states. If $r = 2$, then c is called a *binary* or *2-state character*. We say that c is *convex* on a phylogenetic X-tree $\mathcal{T}$ with vertex set V if there exists a function $\bar{c} : V \to \{s_1, s_2, \ldots, s_r\}$ which extends c such that, for each $i \in \{1, 2, \ldots, r\}$, the subgraph of $\mathcal{T}$ induced by $\{v \in V : \bar{c}(v) = s_i\}$ is connected.

Biologically speaking, if a character c is convex on a phylogenetic tree $\mathcal{T}$, then c can be explained without any reverse or convergent character state transitions.

Compatibility. Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted phylogenetic trees on X and X', respectively, where $X' \subseteq X$. Then $\mathcal{T}$ *displays* $\mathcal{T}'$ if $\mathcal{T}'$ is a restriction of $\mathcal{T}$. Intuitively, if $\mathcal{T}$ displays $\mathcal{T}'$, then all of the ancestral relationships represented by $\mathcal{T}'$ are represented by $\mathcal{T}$. Furthermore, $\mathcal{T}$ *displays* a set $\mathcal{R}$ of rooted phylogenetic trees if each element in $\mathcal{R}$ is displayed by $\mathcal{T}$, in which case $\mathcal{T}$ is said to be *compatible* with $\mathcal{R}$. Moreover, a set $\mathcal{R}$ of rooted phylogenetic trees is said to be *k-compatible* if there exists a collection $\mathcal{P}$ of at most k rooted phylogenetic trees such that each element in $\mathcal{R}$ is displayed by at least one tree in $\mathcal{P}$. If $\mathcal{R}$ is not 1-compatible, we will sometimes write that $\mathcal{R}$ is *incompatible* for short.

Now, let Σ be a collection of binary characters on X. Then Σ is *compatible* if there exists a phylogenetic X-tree on which each character in Σ is convex. A consequence of this definition and the Splits-Equivalence Theorem [5] is the following *four-gamete condition* that characterizes when a pair of binary characters is incompatible. Let $c_1 : X \rightarrow \{0,1\}$ and $c_2 : X \rightarrow \{0,1\}$ be two binary characters on X. Then, c_1 and c_2 are incompatible if and only if there exists a subset $\{x_1, x_2, x_3, x_4\} \subseteq X$ such that the following properties are satisfied:

(i) $c_1(x_1) = 1$ and $c_2(x_1) = 1$,
(ii) $c_1(x_2) = 1$ and $c_2(x_2) = 0$,
(iii) $c_1(x_3) = 0$ and $c_2(x_3) = 1$, and
(iv) $c_1(x_4) = 0$ and $c_2(x_4) = 0$.

Furthermore, Buneman proved that a collection Σ of binary characters is compatible if no pair of characters in Σ is incompatible [5]. Similar to the concept of k-compatibility for a set of rooted triples, we say that a set Σ of binary characters is *k-compatible* if there exists a collection $\mathcal{P}$ of at most k phylogenetic trees such that each element in Σ is convex on at least one tree in $\mathcal{P}$. Equivalently, Σ is *k-compatible* if Σ can be partitioned into at most k blocks such that each block consists of a set of compatible characters. If Σ is not 1-compatible, we will sometimes write that Σ is *incompatible* for short.

The concept of k-compatibility for trees and characters leads to the following two decision problems:

k-Tree-Compatibility
Instance. A set $\mathcal{R}$ of rooted phylogenetic trees and a positive integer k.
Question. Is $\mathcal{R}$ k-compatible?

k-Character-Compatibility
Instance. A set Σ of binary characters and a positive integer k.
Question. Is Σ k-compatible?

While 1-Tree-Compatibility is solvable in polynomial time [1, 12], we show in Section 3 that 2-Tree-Compatibility is NP-complete by establishing the result for the special case when $\mathcal{R}$ is a set of rooted triples. In turn this implies that the following optimization problem is NP-hard.

Min-k-Tree-Compatibility
Instance. A set $\mathcal{R}$ of rooted phylogenetic trees.
Goal. Find a collection $\mathcal{P} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k\}$ of rooted phylogenetic trees such that each tree in $\mathcal{R}$ is displayed by an element in $\mathcal{P}$ and k is minimized.
Measure. The value of k.

The problem 1-Character-Compatibility is equivalent to the well-known perfect phylogeny problem for binary characters, for which a polynomial-time algorithm exists [7]. In Section 4, we show that k-Character-Compatibility is NP-complete for any $k > 2$ but polynomial-time solvable for $k = 2$. Section 5 finishes the paper with some concluding remarks.

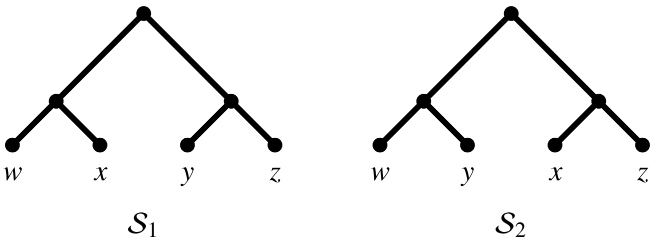

Figure 1: The set $\{wx|y, wx|z, yz|w, yz|x, wy|x, wy|z, xz|w, xz|y\}$ of rooted triples is 2-definitive for the rooted phylogenetic trees $\mathcal{S}_1$ and $\mathcal{S}_2$.

3. Tree compatibility across several trees

In this section, we establish the following theorem.

Theorem 3.1. *The optimization problem* Min-k-Tree-Compatibility *is NP-hard.*

To show that the theorem holds, we use a polynomial-time reduction from the well-known NP-complete problem Set Splitting [6] to 2-Tree-Compatibility, when an instance of the latter problem consists of a set of rooted triples. The NP-completeness of 2-Tree-Compatibility implies NP-hardness of Min-k-Tree-Compatibility. The decision problem Set Splitting is as follows.

Set Splitting
Instance. A set $S = \{s_1, s_2, \ldots, s_n\}$ and a collection $C = \{C^1, C^2, \ldots, C^m\}$ of subsets of S, where $|C^j| = 3$ for each $j \in \{1, 2, \ldots, m\}$.
Question. Does there exist a bipartition of S into S_1 and S_2 such that $C^j \cap S_1 \neq \emptyset$ and $C^j \cap S_2 \neq \emptyset$ for each $C^j \in C$?

If the answer to an instance (S, C) of Set Splitting is 'yes', then we say that (S, C) has a *set splitting* (S_1, S_2).

We begin the reduction with a lemma. Let $\mathcal{R}$ be a set of rooted triples that is 2-compatible with two rooted phylogenetic trees $\mathcal{T}_1$ and $\mathcal{T}_2$ with $\mathcal{L}(\mathcal{T}_1) = \mathcal{L}(\mathcal{T}_2) = \mathcal{L}(\mathcal{R})$. If $\mathcal{T}_1$ and $\mathcal{T}_2$ are the unique such trees, then $\mathcal{R}$ is said to be 2-*definitive*. Note that, if $\mathcal{R}$ is 2-definitive, then $\mathcal{T}_1$ and $\mathcal{T}_2$ are necessarily binary.

Lemma 3.2. *The set* $\mathcal{R} = \{wx|y, wx|z, yz|w, yz|x, wy|x, wy|z, xz|w, xz|y\}$ *of rooted triples is* 2*-definitive.*

Proof. It is easily checked that each of the rooted triples in $\mathcal{R}$ is displayed by either $\mathcal{S}_1$ or $\mathcal{S}_2$ as shown in Figure 1. Thus $\mathcal{R}$ is 2-compatible. To see that $\mathcal{R}$ is 2-definitive, let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two rooted phylogenetic trees both with label set $\{w, x, y, z\}$ such that each rooted triple in $\mathcal{R}$ is displayed by either $\mathcal{T}_1$ or $\mathcal{T}_2$. Without loss of generality, we may assume that $\mathcal{T}_1$ displays $wx|y$ and that $\mathcal{T}_2$ displays $wy|x$. Since $yz|x$ and $xz|y$ are incompatible, one of them is displayed by $\mathcal{T}_1$ and one of them is displayed by $\mathcal{T}_2$. Assume that $xz|y$ is displayed by $\mathcal{T}_1$ and that $yz|x$ is displayed by $\mathcal{T}_2$. Then, $\mathcal{T}_1$ does not display $yz|w$ or $wy|z$, and it follows that these two incompatible rooted triples are both displayed by $\mathcal{T}_2$; a contradiction. Thus $\mathcal{T}_1$ displays $wx|y$ and $yz|x$, and $\mathcal{T}_2$ displays $wy|x$ and $xz|y$; thereby implying that $\mathcal{T}_1$ is isomorphic to $\mathcal{S}_1$ and $\mathcal{T}_2$ is isomorphic to $\mathcal{S}_2$. This completes the proof of the lemma. □

The next theorem establishes the NP-completeness of 2-Tree-Compatibility. The reduction that we use for the proof has a flavor that is similar to that in [9, Theorem 7], which shows that it is computationally hard to determine if a set of rooted triples is compatible with a so-called 'simple phylogenetic network'.

Theorem 3.3. *The decision problem* 2-Tree-Compatibility *is NP-complete.*

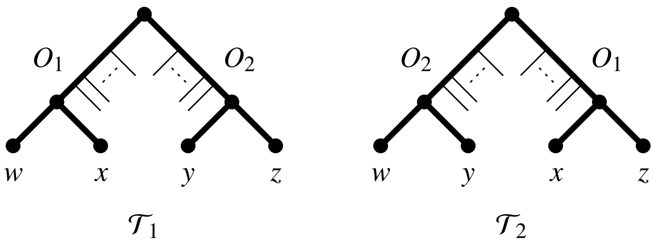

Figure 2: The rooted phylogenetic trees $\mathcal{T}_1$ and $\mathcal{T}_2$ that are reconstructed from a set splitting in the proof of Theorem 3.3, where the unlabeled leaves are labeled with the elements in O_1 and O_2 (for details, see text).

Proof. We establish the theorem by showing that 2-Tree-Compatibility is NP-complete for an instance that consists of a set of rooted triples. Since a set of rooted triples is a special case of a set of rooted phylogenetic trees, this immediately implies NP-completeness of 2-Tree-Compatibility in its general form. Let $\mathcal{R}$ be a set of rooted triples. The problem 2-Tree-Compatibility is clearly in NP because, given two rooted phylogenetic trees $\mathcal{T}_1$ and $\mathcal{T}_2$, it can be verified in polynomial time whether or not each rooted triple in $\mathcal{R}$ is displayed by $\mathcal{T}_1$ or $\mathcal{T}_2$.

We now describe a polynomial-time reduction from Set Splitting to 2-Tree-Compatibility. Let (S, C) be an instance of Set Splitting, where $S = \{s_1, s_2, \ldots, s_n\}$ and $C = \{C^1, C^2, \ldots, C^m\}$ with $|C^j| = 3$. Without loss of generality, we may assume that, for each $s_i \in S$, there exists a $C^j \in C$ such that $s_i \in C^j$. Throughout the proof, we write $\{s_a^j, s_b^j, s_c^j\}$ to denote a set $C^j \in C$. We next define four sets of rooted triples:

(1) Let $\mathcal{R}_1$ be the set

$$\mathcal{R}_1 = \{wx|y, wx|z, yz|w, yz|x, wy|x, wy|z, xz|w, xz|y\}.$$

(2) Each set $C^j = \{s_a^j, s_b^j, s_c^j\}$ is represented by six rooted triples in

$$\mathcal{R}_2 = \bigcup_{C^j \in C} \{ws_a^j|s_b^j, ws_b^j|s_c^j, ws_c^j|s_a^j, zs_a^j|s_b^j, zs_b^j|s_c^j, zs_c^j|s_a^j\}.$$

(3) Each $s_i \in S$ is represented by $2m$ rooted triples in

$$\mathcal{R}_3 = \bigcup_{s_i \in S} \{s_i^1 w|z, s_i^1 z|w, s_i^2 w|z, s_i^2 z|w, \ldots, s_i^m w|z, s_i^m z|w\}.$$

(4) Each $s_i \in S$ is represented by $2\binom{m}{2}$ rooted triples in

$$\mathcal{R}_4 = \bigcup_{s_i \in S} \{s_i^1 s_i^2|w, s_i^1 s_i^2|z, s_i^1 s_i^3|w, s_i^1 s_i^3|z, \ldots, s_i^1, s_i^m|w, s_i^1 s_i^m|z, s_i^2 s_i^3|w, s_i^2 s_i^3|z, \ldots, s_i^{m-1} s_i^m|w, s_i^{m-1} s_i^m|z\}.$$

Now, let

$$\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3 \cup \mathcal{R}_4$$

be an instance of 2-Tree-Compatibility. Noting that $|\mathcal{R}|$ is in the order of $O(m^2 n)$, the reduction can be carried out in polynomial time.

The remainder of the proof essentially consists of proving the following claim.

Claim. (S, C) has a set splitting if and only if $\mathcal{R}$ is 2-compatible.

First, suppose that (S, C) has a set splitting (S_1, S_2). Let $S'_1 = \{s_i^j : s_i \in S_1 \text{ and } j \in \{1, \ldots, m\}\}$ and, similarly, let $S'_2 = \{s_i^j : s_i \in S_2 \text{ and } j \in \{1, \ldots, m\}\}$. Furthermore, for each $\ell \in \{1, 2\}$, let O_ℓ be an ordering on the elements in S'_ℓ such that, for each C^j, precisely one of the following holds:

(i) If $s_a^j, s_b^j \in S'_\ell$, then s_a^j precedes s_b^j in O_ℓ.

(ii) If $s_b^j, s_c^j \in S'_\ell$, then s_b^j precedes s_c^j in O_ℓ.

(iii) If $s_a^j, s_c^j \in S'_\ell$, then s_c^j precedes s_a^j in O_ℓ.

Note that not all three of s_a^j, s_b^j, and s_c^j are elements of S'_ℓ since (S_1, S_2) is a set splitting of (S, C). Now, let $\mathcal{T}_1$ be the rooted phylogenetic tree obtained from the two caterpillars C_1 on (w, x, O_1) and C'_1 on (y, z, O_2) by creating a new vertex ρ_1 and adjoining the root vertices of C_1 and C'_1, respectively, to ρ_1 via two new edges. Similarly, let $\mathcal{T}_2$ be the rooted phylogenetic tree obtained from the two caterpillars C_2 on (w, y, O_2) and C'_2 on (x, z, O_1) by creating a new vertex ρ_2 and adjoining the root vertices of C_2 and C'_2, respectively, to ρ_2 via two new edges. This construction is illustrated in Figure 2.

We next show that each rooted triple in $\mathcal{R}$ is displayed by either $\mathcal{T}_1$ or $\mathcal{T}_2$. Clearly, by construction, either $\mathcal{T}_1$ or $\mathcal{T}_2$ displays each rooted triple in $\mathcal{R}_1$. Furthermore, since S_1 and S_2 is a set splitting of (S, C), two elements of each $C^j = \{s_a^j, s_b^j, s_c^j\}$ are contained in S_ℓ while the remaining element is contained in $S_{\ell'}$, where $\{\ell, \ell'\} = \{1, 2\}$. Now, because (i), (ii), or (iii) holds, a routine check shows that, for each C^j, each of the corresponding six rooted triples in $\mathcal{R}_2$ is displayed by either $\mathcal{T}_1$ or $\mathcal{T}_2$. In particular, for any pair $(i, i') \in \{(a, b), (b, c), (c, a)\}$, the rooted triple $ws_i^j|s_{i'}^j$ is displayed by $\mathcal{T}_1$ (resp. $\mathcal{T}_2$) if and only if the rooted triple $zs_i^j|s_{i'}^j$ is displayed by $\mathcal{T}_2$ (resp. $\mathcal{T}_1$). Turning to the rooted triples that are contained in $\mathcal{R}_3$, we observe that, for each element s_i^j in S'_1, the rooted triple $s_i^jw|z$ is displayed by $\mathcal{T}_1$ while the rooted triple $s_i^jz|w$ is displayed by $\mathcal{T}_2$ and, similarly, for each element s_i^j in S'_2, the rooted triple $s_i^jw|z$ is displayed by $\mathcal{T}_2$ while the rooted triple $s_i^jz|w$ is displayed by $\mathcal{T}_1$. Hence, all rooted triples in $\mathcal{R}_3$ are displayed by either $\mathcal{T}_1$ or $\mathcal{T}_2$. Lastly, noting that $\{s_i^1, s_i^2, \ldots, s_i^m\} \subseteq S'_\ell$ for some $\ell \in \{1, 2\}$, it is easily checked that, for each $i \in \{1, 2, \ldots, n\}$, the rooted triples in

$$\left\{s_i^j s_i^{j'}|z : j, j' \in \{1, 2, \ldots, m\} \text{ and } j < j'\right\}$$

are displayed by $\mathcal{T}_\ell$ while the rooted triples in

$$\left\{s_i^j s_i^{j'}|w : j, j' \in \{1, 2, \ldots, m\} \text{ and } j < j'\right\}$$

are displayed by $\mathcal{T}_{\ell'}$, where $\{\ell, \ell'\} = \{1, 2\}$. Thus, each rooted triple in $\mathcal{R}_4$ is displayed by either $\mathcal{T}_1$ or $\mathcal{T}_2$. Since each rooted triple in $\mathcal{R}$ is displayed by either $\mathcal{T}_1$ or $\mathcal{T}_2$, it follows that $\mathcal{R}$ is 2-compatible.

Second, suppose that $\mathcal{R}$ is 2-compatible. Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two rooted phylogenetic trees such that each rooted triple in $\mathcal{R}$ is displayed by either $\mathcal{T}_1$ or $\mathcal{T}_2$. Without loss of generality, we may assume that $\mathcal{T}_1$ and $\mathcal{T}_2$ are binary. To ease reading throughout this part of the proof, let $i \in \{1, 2, \ldots, n\}$, and let $j \in \{1, 2, \ldots, m\}$. By Lemma 3.2, the set $\mathcal{R}_1$ is 2-definitive and it immediately follows that $\mathcal{T}_1|\{w, x, y, z\}$ and $\mathcal{T}_2|\{w, x, y, z\}$ are the two trees that are shown in Figure 1. Furthermore, since, for each fixed i and j, the two rooted triples $s_i^jw|z$ and $s_i^jz|w$ are incompatible, we have $\mathcal{L}(\mathcal{T}_1) = \mathcal{L}(\mathcal{T}_2)$.

Now, for $\ell \in \{1, 2\}$, assume that the path from the root of $\mathcal{T}_\ell$ to a leaf s_i^j does not contain $\text{mrca}_{\mathcal{T}_\ell}(\{w, z\})$. Then, the two incompatible rooted triples $s_i^jw|z$ and $s_i^jz|w$ are both displayed by $\mathcal{T}_{\ell'}$, where $\{\ell, \ell'\} = \{1, 2\}$; a contradiction. Therefore, we may assume for the remainder of the proof that the roots of $\mathcal{T}_1$ and $\mathcal{T}_2$ coincide with $\text{mrca}_{\mathcal{T}_1}(\{w, z\})$ and $\text{mrca}_{\mathcal{T}_2}(\{w, z\})$, respectively. In the following, we say that s_i^j is on the *w-side* of $\mathcal{T}_1$ (resp. $\mathcal{T}_2$) if the path in $\mathcal{T}_1$ (resp. $\mathcal{T}_2$) from the root to w intersects with the path in $\mathcal{T}_1$ (resp. $\mathcal{T}_2$) from the root to s_i^j at a vertex other than the root. Similarly, we say that s_i^j is on the *z-side* of $\mathcal{T}_1$ (resp. $\mathcal{T}_2$) if the path in $\mathcal{T}_1$ (resp. $\mathcal{T}_2$) from the root to z intersects with the path in $\mathcal{T}_1$ (resp. $\mathcal{T}_2$) from the root to s_i^j at a vertex other than the root.

We next show that, for any fixed i, each element in $S_i = \{s_i^j : j \in \{1, 2, \ldots, m\}\}$ is on the w-side of $\mathcal{T}_1$ or each such element is on the z-side of $\mathcal{T}_1$. Assume the contrary, i.e. for some fixed i, there exists a leaf $s_i^j \in S_i$ that is on the w-side of $\mathcal{T}_1$ and there exists a leaf $s_i^{j'} \in S_i$ that is on the z-side of $\mathcal{T}_1$ with $j \neq j'$. Hence, the four rooted triples $s_i^js_i^{j'}|w$, $s_i^js_i^{j'}|z$, $s_i^jz|w$, and $s_i^{j'}w|z$ are all displayed by $\mathcal{T}_2$. Since the root of $\mathcal{T}_2$ coincides with $\text{mrca}_{\mathcal{T}_2}(\{w, z\})$, we may assume that $\text{mrca}_{\mathcal{T}_2}(\{s_i^j, s_i^{j'}\})$ does not lie on the path from the root of $\mathcal{T}_2$ to w or z. It now follows that $s_i^jz|w$ or $s_i^{j'}w|z$

is not displayed by $\mathcal{T}_2$ depending on whether the path from the root of $\mathcal{T}_2$ to w or the path from the root of $\mathcal{T}_2$ to z intersects with the path from the root of $\mathcal{T}_2$ to $\mathrm{mrca}_{\mathcal{T}_2}(\{s_i^j, s_i^{j'}\})$ at a vertex other than the root; again, a contradiction. Thus, the elements in S_i are either all on the w-side of $\mathcal{T}_1$ or all on the z-side of $\mathcal{T}_1$. Suppose that each element in S_i is on the w-side of $\mathcal{T}_1$, then, for each $s_i^j \in S_i$, the rooted triple $s_i^j w|z$ is displayed by $\mathcal{T}_1$; thereby implying that the rooted triple $s_i^j z|w$ is displayed by $\mathcal{T}_2$. It is now easily checked that, each element in S_i is on the z-side of $\mathcal{T}_2$. Applying a similar argument for when each element in S_i is on the z-side of $\mathcal{T}_1$, we derive the following fact.

(F) Each element in S_i is on the w-side (resp. z-side) of $\mathcal{T}_1$ if and only if each element in S_i is on the z-side (resp. w-side) of $\mathcal{T}_2$.

Now, let $\mathcal{W}_1$ and $\mathcal{Z}_1$ be the two rooted phylogenetic trees that are obtained from $\mathcal{T}_1$ by deleting the two edges that are incident with its root such that $w \in \mathcal{L}(\mathcal{W}_1)$ and $z \in \mathcal{L}(\mathcal{Z}_1)$. Analogously, let $\mathcal{W}_2$ and $\mathcal{Z}_2$ be the two rooted phylogenetic trees that are obtained from $\mathcal{T}_2$ by deleting the two edges that are incident with its root such that $w \in \mathcal{L}(\mathcal{W}_2)$ and $z \in \mathcal{L}(\mathcal{Z}_2)$. By (F), we have

$$\mathcal{L}(\mathcal{W}_1) - \{w, x\} = \mathcal{L}(\mathcal{Z}_2) - \{x, z\} \text{ and } \mathcal{L}(\mathcal{Z}_1) - \{y, z\} = \mathcal{L}(\mathcal{W}_2) - \{w, y\}.$$

We complete the proof by showing that (W, Z) is a set splitting for (S, C), where $W = \{s_i \in S : s_i^1 \in \mathcal{L}(\mathcal{W}_1)\}$ and $Z = \{s_i \in S : s_i^1 \in \mathcal{L}(\mathcal{Z}_1)\}$. Clearly, $W \cup Z = S$ and $W \cap Z = \emptyset$. Now, assume that there exists a $C^j \in C$ for which $C^j \cap W = \emptyset$ or $C^j \cap Z = \emptyset$. If $C^j \cap W = \emptyset$, then s_a^j, s_b^j, and s_c^j are all on the z-side of $\mathcal{T}_1$ and, by (F), are all on the w-side of $\mathcal{T}_2$. Since $\mathcal{T}_1$ does not display any of the rooted triples $ws_a^j|s_b^j$, $ws_b^j|s_c^j$, and $ws_c^j|s_a^j$, and a straightforward check shows that $\mathcal{T}_2$ displays at most two such rooted triples, this contradicts that each rooted triple in $\mathcal{R}$ is displayed by either $\mathcal{T}_1$ or $\mathcal{T}_2$. Similarly, if $C^j \cap Z = \emptyset$, then s_a^j, s_b^j, and s_c^j are all on the w-side of $\mathcal{T}_1$ and, by (F), are all on the z-side of $\mathcal{T}_2$. Since $\mathcal{T}_2$ does not display any of the rooted triples $ws_a^j|s_b^j$, $ws_b^j|s_c^j$, and $ws_c^j|s_a^j$, and $\mathcal{T}_1$ displays at most two such rooted triples, this again contradicts that each rooted triple in $\mathcal{R}$ is displayed by $\mathcal{T}_1$ or $\mathcal{T}_2$. This establishes that (W, Z) is a set splitting for (S, C) and completes the proof of the theorem □

We are now in a position to establish Theorem 3.1 which is an almost immediate consequence of Theorem 3.3.

Proof of Theorem 3.1. Assume that Min-k-Tree-Compatibility is not NP-hard. Then, an instance I of 2-Tree-Compatibility can be solved by using a polynomial-time algorithm for Min-k-Tree-Compatibility and returning 'yes' if and only if the answer to Min-k-Tree-Compatibility for I is at most 2; a contradiction. □

4. Character compatibility across several trees

In this section, we show that k-Character-Compatibility is NP-complete for any positive integer $k > 2$ while, perhaps surprisingly, given the result of the last section, the problem is solvable in polynomial time for $k = 2$.

To establish the main result of this section,Theorem 4.3, we make use of the following decision problem.

Graph-k-Colorability
Instance. A graph $G = (V, E)$ and a positive integer $k \leq |V|$.
Question. Is G k-colorable (i.e. does there exist a function $f : V \to \{1, 2, \ldots, k'\}$ with $k' \leq k$ such that $f(u) \neq f(v)$ for each edge $\{u, v\} \in E$)?

This classical decision problem can be solved in polynomial time for $k = 2$ but is NP-complete for $k > 2$ [6]. Without loss of generality, we may assume that an instance of Graph-k-Colorability always consists of a simple graph G.

For a collection Σ of binary characters, the *incompatibility graph* of Σ is the graph whose vertex set is Σ and where an edge joins two characters precisely if they are incompatible. We denote this graph by G_Σ. We next establish two lemmas.

Lemma 4.1. *Let Σ be a collection of binary characters. Then Σ is k-compatible if and only if the incompatibility graph G_Σ is k-colorable.*

Proof. Suppose Σ is k-compatible. Then there exists a partition of the vertex set of G_Σ into at most k blocks $B_1, B_2, \ldots, B_{k'}$ with $k' \leq k$ such that no edge in G_Σ joins two vertices of the same block. Clearly, by assigning all vertices of B_i to color C_i for each $i \in \{1, 2, \ldots, k'\}$, we obtain a k'-coloring of G_Σ and thus G_Σ is k-colorable. Now, suppose that G_Σ is k-colorable. For each color C_i with $i \in \{1, 2, \ldots, k'\}$ and $k' \leq k$, let B_i be the set of vertices that have color C_i. Since no edge in G_Σ joins two vertices in B_i, it follows that the characters in Σ corresponding to vertices in B_i are 1-compatible. Hence, Σ is k'-compatible and thus also k-compatible. □

Lemma 4.2. *Let $G = (V, E)$ be a simple graph. There exists a collection of binary characters Σ whose incompatibility graph G_Σ is isomorphic to G.*

Proof. Let $O = (v_1, v_2, \ldots, v_k)$ be an ordering on the vertices of G with $|V| = k$. We construct a set S of $4 \cdot |E|$ sequences, each having length k and where position $\ell \in \{1, 2, \ldots, k\}$ corresponds to the ℓ^{th} vertex in O. More precisely, for each edge $\{v_i, v_j\}$ in G, we assume without loss of generality that $i < j$ and represent it by the following four sequences in S:

$$\begin{array}{ll}
S^{1,1}_{i,j} & s_1, s_2, \ldots, s_{i-1}, 1, s_{i+1}, \ldots, s_{j-1}, 1, s_{j+1}, \ldots, s_k \\
S^{1,0}_{i,j} & s_1, s_2, \ldots, s_{i-1}, 1, s_{i+1}, \ldots, s_{j-1}, 0, s_{j+1}, \ldots, s_k \\
S^{0,1}_{i,j} & s_1, s_2, \ldots, s_{i-1}, 0, s_{i+1}, \ldots, s_{j-1}, 1, s_{j+1}, \ldots, s_k \\
S^{0,0}_{i,j} & s_1, s_2, \ldots, s_{i-1}, 0, s_{i+1}, \ldots, s_{j-1}, 0, s_{j+1}, \ldots, s_k,
\end{array}$$

where each s_ℓ with $\ell \notin \{i, j\}$ has character state 0 in each of the four sequences. Let Σ be the set of characters induced by S, i.e. we have a character c_ℓ for each column $\ell \in \{1, 2, \ldots, k\}$ in S. Note that, for each sequence in S, the character state 1 occurs at most twice.

We proceed by induction on the number $|E|$ of edges in G. If G does not contain any edge, then S is the empty set (i.e. it does not contain any character) and so the result is vacuously true for the base case. Now, let G' be the simple graph obtained from G by deleting an edge $\{v_i, v_j\}$, and let Σ' be the set of characters induced by $S - \{S^{1,1}_{i,j}, S^{1,0}_{i,j}, S^{0,1}_{i,j}, S^{0,0}_{i,j}\}$. For the purpose of induction, assume that the incompatibility graph $G_{\Sigma'}$ is isomorphic to G'. As the characters c_i and c_j are incompatible in Σ by the four-gamete condition, it follows that G is a subgraph of the incompatibility graph G_Σ. To show that G and G_Σ are indeed isomorphic, assume that G_Σ has an edge joining two characters $c_{i'}$ and $c_{j'}$ while there is no edge $\{v_{i'}, v_{j'}\}$ in G. Since the two characters $c_{i'}$ and $c_{j'}$ are incompatible, there exists a sequence in S that has character state 1 for both $c_{i'}$ and $c_{j'}$, a sequence that has character state 0 for both $c_{i'}$ and $c_{j'}$, a sequence that has character state 1 for $c_{i'}$ and character state 0 for $c_{j'}$, and a sequence that has character state 0 for $c_{i'}$ and character state 1 for $c_{j'}$; thereby contradicting that $\{v_{i'}, v_{j'}\}$ is not an edge in G. It now follows that G and G_Σ are isomorphic. □

The next theorem is an immediate consequence of Lemmas 4.1 and 4.2, and the fact that the decision problem Graph-k-Colorability is NP-complete for any positive integer $k > 2$ and polynomial-time solvable for $k \leq 2$ [6].

Theorem 4.3. *The decision problem k-Character-Compatibility is NP-complete for any positive integer $k > 2$ and polynomial-time solvable for $k \leq 2$.*

5. Concluding remarks

In this paper, we investigated the problem of computing the minimum size of a set $\mathcal{P}$ of rooted phylogenetic trees such that each tree (resp. binary character) in a given set of rooted phylogenetic trees (resp. binary characters) is compatible with at least one tree in $\mathcal{P}$. We established NP-hardness of solving an instance of the optimization problem Min-k-Tree-Compatibility. However, while we have explicitly shown that it is computationally hard to decide whether or not a set of rooted phylogenetic trees is 2-compatible, there appears to be no straightforward polynomial-time reduction to extend our result to 3-Tree-Compatibility and, ultimately, to k-Tree-Compatibility for $k \geq 3$. This is, for example, in stark contrast to the problem GRAPH-k-COLORABILITY for which a reasonably easy polynomial-time reduction from GRAPH-$(k-1)$-COLORABILITY to GRAPH-k-COLORABILITY exists that gives the desired result for each $k \geq 4$. A complexity result for k-TREE-COMPATIBILITY with $k \geq 3$ remains therefore open. Furthermore, for an input that consists of a set of binary characters and a positive integer $k > 2$, we have shown that it is NP-complete to decide if there exists a set $\mathcal{P}$ of at most k phylogenetic trees such that each character in the input is convex on some tree in $\mathcal{P}$. However, for $k = 2$, we have shown that this problem is polynomial-time solvable.

Acknowledgements. The first author was supported by a Marie Curie International Outgoing Fellowship within the 7^{th} European Community Framework Programme. The second author was supported by the US National Science Foundation Grant #0920920. The third author was supported by the Allan Wilson Centre for Molecular Ecology and Evolution and the New Zealand Marsden Fund.

References

[1] A.V. Aho, Y. Sagiv, T.G. Syzmanski, J.D. Ullman, Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions, *SIAM J. Comput.* 10 (1981) 405–421.

[2] M. Baroni, C. Semple, and M. Steel, A framework for representing reticulate evolution, *Ann. Comb.* 8 (2004) 391–408.

[3] O.R.P. Bininda-Emonds (Ed.), Phylogenetic supertrees: combining information to reveal the tree of life, Springer, 2004.

[4] H.L. Bodlaender, M.R. Fellows, T.J. Warnow, Two strikes against perfect phylogeny, in: Proceedings of the International Colloquium on Automata, Languages and Programming (LNCS 623) (1992) 273–283.

[5] P. Buneman, The recovery of trees from measures of dissimilarity, in: Mathematics in the Archaeological and Historical Sciences, Edinburgh University Press, (1971) 387–395.

[6] M.R. Garey, D.S. Johnson, Computers and intractability: a guide to the theory of NP-completeness, W. H. Freeman and Company, New York, 1979.

[7] D. Gusfield, Efficient algorithms for inferring evolutionary trees, *Networks* 21 (1991) 19–28.

[8] B.R. Holland, H.G. Spencer, T.H. Worth, M. Kennedy, Identifying cliques of convergent characters: concerted evolution in the cormorants and shags, *Syst. Biol.* 59 (2010) 433–445.

[9] J. Jansson, N.B. Nguyen, W.-K. Sung, Algorithms for combining rooted triplets into a galled phylogenetic network, *SIAM J. Comput.* 35 (2006) 1098–1121.

[10] A.C. McHardy, I. Rigoutsos, What's in the mix: phylogenetic classification of metagenome sequence samples, *Curr. Opin. Microbiol.* 10 (2007) 499–503.

[11] M.I. Nelson, E.C. Holmes, The evolution of epidemic influenza, *Nat. Rev. Genet.* 8 (2007) 196–205.

[12] C. Semple, M. Steel, Phylogenetics, Oxford University Press, 2003.

[13] M. Steel, The complexity of reconstructing trees from qualitative characters and subtrees, *J. Classif.* 9 (1992) 91–116.