Display sets of normal and tree-child networks

Janosch Döcker
Department of Computer Science
University of Tübingen
Tübingen, Germany
janosch.doecker@uni-tuebingen.de

Simone Linz *
School of Computer Science
University of Auckland
Auckland, New Zealand
s.linz@auckland.ac.nz

Charles Semple †
School of Mathematics and Statistics
University of Canterbury
Christchurch, New Zealand
charles.semple@canterbury.ac.nz

Submitted: Nov 9, 2019; Accepted: Dec 8, 2020; Published: TBD

Abstract

Phylogenetic networks are leaf-labelled directed acyclic graphs that are used in computational biology to analyse and represent the evolutionary relationships of a set of species or viruses. In contrast to phylogenetic trees, phylogenetic networks have vertices of in-degree at least two that represent reticulation events such as hybridisation, lateral gene transfer, or reassortment. By systematically deleting various combinations of arcs in a phylogenetic network $\mathcal{N}$, one derives a set of phylogenetic trees that are embedded in $\mathcal{N}$. We recently showed that the problem of deciding if two binary phylogenetic networks embed the same set of phylogenetic trees is computationally hard, in particular, we showed it to be Π_2^P-complete. In this paper, we establish a polynomial-time algorithm for this decision problem if the initial two networks consist of a normal network and a tree-child network; two well-studied topologically restricted subclasses of phylogenetic networks, with normal networks being more structurally constrained than tree-child networks. The running time of the algorithm is quadratic in the size of the leaf sets.

Mathematics Subject Classifications: 05C85, 92D15

*Supported by the New Zealand Marsden Fund.
†Supported by the New Zealand Marsden Fund.

1 Introduction

Phylogenetic (evolutionary) networks rather than phylogenetic trees provide a more faithful representation of the ancestral history of certain collections of extant species. The reason for this is the existence of non-treelike (reticulate) evolutionary processes such as lateral gene transfer and hybridisation. Similar to the study of phylogenetic trees, the development of tools and algorithms to reconstruct phylogenetic networks from biological sequence data is an active area of research [14, 17, 22]. However, in this paper, we focus on the combinatorial properties of phylogenetic networks. A precise understanding of these properties is indispensable for the analysis and comparison of phylogenetic networks as well as for the advancement of network reconstruction algorithms.

At the species-level, evolution is not necessarily treelike. But, at the level of genes, we typically assume treelike evolution. Consequently, as phylogenetic networks are frequently viewed as an amalgamation of the ancestral history of genes, we are interested in the phylogenetic trees embedded (displayed) in a given phylogenetic network. From this viewpoint, there has been a variety of studies including the small maximum parsimony problem for phylogenetic networks [15], deciding if a phylogenetic network is (uniquely) determined by the phylogenetic trees it embeds [6, 20], counting the number of phylogenetic trees displayed by a phylogenetic network [12], and determining if a phylogenetic network embeds a phylogenetic tree more than once [4]. In this context, one of the most well-known studied computational problems is Tree-Containment. Here, the problem is deciding whether or not a given phylogenetic tree is embedded in a given phylogenetic network. In general, the problem is NP-complete [11], but it has been shown to be decidable in polynomial-time for several prominent classes of phylogenetic networks [1, 8, 10].

Recently posed in [8] for reticulation-visible networks, in this paper we study a natural variation of Tree-Containment. In particular, we consider the problem of deciding whether or not two given binary phylogenetic networks embed the same set of phylogenetic trees. Called Display-Set-Equivalence, we recently showed that, in general, this problem is Π_2^P-complete [5], that is, complete for the second level of the polynomial hierarchy. A related problem that is also Π_2^P-complete and that we investigated in the same paper asks whether or not the set of trees embedded in a phylogenetic network is a subset of the set of trees embedded in another network. Problems on the second level of the polynomial hierarchy are computationally more difficult than problems on the first level which include all NP- and co-NP-complete problems. For further details, see [18]. In contrast, the main result of this paper shows that there is a polynomial-time algorithm for Display-Set-Equivalence if one of the two given networks is normal and the other one is tree-child.

Normal [19] and tree-child networks [3] are two structurally constrained subclasses of phylogenetic networks. While formal definitions are given below, we informally mention here that a tree-child network has the property that every non-leaf vertex has a child that does not represent a reticulation event. Moreover, a normal network is tree-child with an additional property concerning the arcs directed into a vertex representing a reticulation event, which we refer to as "no shortcuts". Both subclasses have actively been studied

for the last ten years. Indeed, studying subclasses of phylogenetic networks is particularly appealing from a mathematical perspective because (a) several decision problems that are computationally hard in general can be solved in polynomial time for certain subclasses, and (b) algorithms that reconstruct phylogenetic networks from smaller building blocks, such as networks on three leaves, often only uniquely encode phylogenetic networks of restricted subclasses [4, 7, 9, 10, 20]. The rest of the introduction formally defines Display-Set-Equivalence, states the main result, and provides additional details.

A *binary phylogenetic network* $\mathcal{N}$ *on* X is a rooted acyclic directed graph with no arcs in parallel and satisfying the following properties:

(i) the (unique) root has out-degree two;

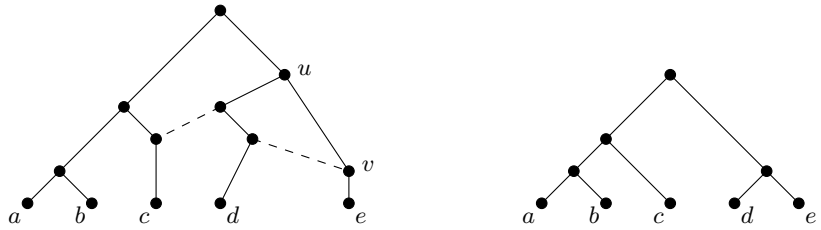
(ii) a vertex with out-degree zero has in-degree one, and the set of vertices with out-degree zero is X; and

(iii) all other vertices have either in-degree one and out-degree two, or in-degree two and out-degree one.

For technical reasons, if $|X| = 1$, we additionally allow a single vertex labelled by the element in X to be a binary phylogenetic network. The vertices in $\mathcal{N}$ of out-degree zero are called *leaves*, and so X is referred to as the *leaf set* of $\mathcal{N}$. Furthermore, vertices of in-degree one and out-degree two are *tree vertices*, while vertices of in-degree two and out-degree one are *reticulations*. The arcs directed into a reticulation are *reticulation arcs*, all other arcs are *tree arcs*. A *binary phylogenetic* X*-tree* is a binary phylogenetic network on X with no reticulations. To ease reading and since all phylogenetic networks considered in this paper are binary, we refer to a binary phylogenetic network (resp. binary phylogenetic tree) as a phylogenetic network (resp. phylogenetic tree).

Let $\mathcal{N}$ be a phylogenetic network. A reticulation arc (u, v) of $\mathcal{N}$ is a *shortcut* if there is a directed path in $\mathcal{N}$ from u to v that does not traverse (u, v). We say that $\mathcal{N}$ is a *tree-child network* if every non-leaf vertex is the parent of a tree vertex or a leaf. If, in addition, $\mathcal{N}$ has no shortcuts, then $\mathcal{N}$ is *normal*. To illustrate, in Fig. 1(i), $\mathcal{N}$ is a tree-child network but it is not normal as the arc (u, v) is a shortcut. As with all other figures in the paper, arcs are directed down the page.

Now let $\mathcal{N}$ be a phylogenetic network on X and let $\mathcal{T}$ be a phylogenetic X-tree. Then $\mathcal{N}$ *displays* $\mathcal{T}$ if $\mathcal{T}$ can be obtained from $\mathcal{N}$ by deleting arcs and vertices, and suppressing the resulting vertices of in-degree one and out-degree one. An equivalent and useful way to view the notion of displaying is as follows. The *root extension* of $\mathcal{T}$ is obtained by adjoining a new vertex, u say, to the root of $\mathcal{T}$ via a new arc directed away from u. It is easily checked that $\mathcal{N}$ displays $\mathcal{T}$ precisely if a subdivision of either $\mathcal{T}$ or the root extension of $\mathcal{T}$ can be obtained from $\mathcal{N}$ by deleting arcs and non-root vertices. We refer to such a subdivision as an *embedding*, $\mathcal{S}$ say, of $\mathcal{T}$ in $\mathcal{N}$. Observe that it follows from the definition of an embedding that the unique vertex of $\mathcal{S}$ with in-degree zero is also the root vertex of $\mathcal{N}$. Having these two vertices coincide is particularly convenient when writing arguments, and so we frequently adopt this viewpoint in the proofs of the paper.



(i) $\mathcal{N}$ (ii) $\mathcal{T}$

Figure 1: (i) A tree-child network $\mathcal{N}$ on $\{a, b, c, d, e\}$ and (ii) a phylogenetic tree $\mathcal{T}$ displayed by $\mathcal{N}$.

To illustrate the notion of display, in Fig. 1, $\mathcal{N}$ displays $\mathcal{T}$, where an embedding of $\mathcal{T}$ in $\mathcal{N}$ is shown as solid arcs. Note that there is one other distinct embedding of $\mathcal{T}$ in $\mathcal{N}$. Furthermore, the root extension $\mathcal{T}'$ of a phylogenetic tree $\mathcal{T}$ is shown in Fig. 2, where $\mathcal{T}$ is displayed by the tree-child network $\mathcal{N}'$ in the same figure since an embedding of $\mathcal{T}$ in $\mathcal{N}'$ can be obtained by deleting the two arcs (u_1, v_1) and (u_2, v_2), the two arcs (u_1, v_1) and (u'_2, v_2), or the two arcs (u'_1, v_1) and (u_2, v_2) in $\mathcal{N}'$. Now, suppose that $\mathcal{S}$ is an embedding of $\mathcal{T}$ in $\mathcal{N}$. If (u, v) is an arc in $\mathcal{N}$, we say $\mathcal{S}$ *uses* (u, v) if (u, v) is an arc in $\mathcal{S}$. The set of phylogenetic X-trees displayed by $\mathcal{N}$, called the *display set* of $\mathcal{N}$, is denoted by $T(\mathcal{N})$.

The problem of interest in this paper is the following decision problem:

Display-Set-Equivalence
Input. Two phylogenetic networks $\mathcal{N}$ and $\mathcal{N}'$ on X.
Output. Is $T(\mathcal{N}) = T(\mathcal{N}')$?
It is shown in [5] that, in general, Display-Set-Equivalence is Π_2^P-complete. In contrast, the main result of this paper shows that this decision problem is solvable in polynomial time if $\mathcal{N}$ is normal and $\mathcal{N}'$ is tree-child. In particular, we have

Theorem 1. *Let $\mathcal{N}$ and $\mathcal{N}'$ be normal and tree-child networks on X, respectively. Then deciding if $T(\mathcal{N}) = T(\mathcal{N}')$ can be done in time quadratic in the size of X.*

Before continuing, we add some remarks. The proof of Theorem 1 turned out to be much longer than we originally anticipated. If $\mathcal{N}'$ has no shortcuts, that is, $\mathcal{N}'$ is normal, then $T(\mathcal{N}) = T(\mathcal{N}')$ if and only if $\mathcal{N}$ is isomorphic to $\mathcal{N}'$ [20]. However, if $\mathcal{N}'$ is allowed to have shortcuts, then it is possible that $T(\mathcal{N}) = T(\mathcal{N}')$, but $\mathcal{N}$ is not isomorphic to $\mathcal{N}'$. For example, consider the normal and tree-child networks $\mathcal{N}$ and $\mathcal{N}'$, respectively, shown in Fig. 2. Clearly, $\mathcal{N}$ is not isomorphic to $\mathcal{N}'$, but it is easily checked that $T(\mathcal{N}) = T(\mathcal{N}')$. While we already knew of instances like that shown in Fig. 2, the allowance of shortcuts raised many more hurdles than we expected. We next explain briefly what causes at least some of these hurdles. Let v be a reticulation vertex of a tree-child network $\mathcal{N}$, and let u and u' be the two parents of v. Since $\mathcal{N}$ is tree-child, it follows from the definition (see Lemma 2) that there is a directed path from u (resp. u') to a leaf ℓ (resp. ℓ') that does not contain a reticulation arc. Importantly, if $\mathcal{N}$ is also normal, then $\ell \neq \ell'$ and the local

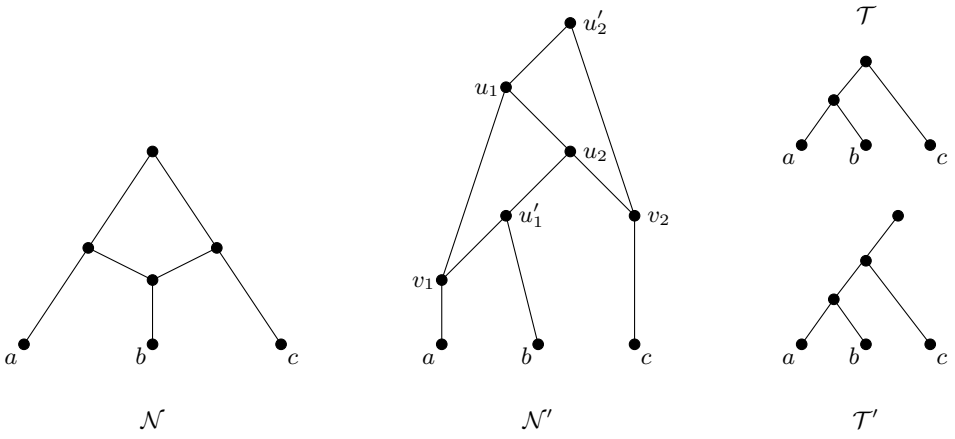


Figure 2: A normal network $\mathcal{N}$ and a tree-child network $\mathcal{N}'$, where $T(\mathcal{N}) = T(\mathcal{N}')$ but $\mathcal{N}$ is not isomorphic to $\mathcal{N}'$. Furthermore, the root extension $\mathcal{T}'$ of a phylogenetic tree $\mathcal{T}$.

structure of $\mathcal{N}$ around v is quite restricted. On the other hand, if $\mathcal{N}$ is not normal and either (u, v) or (u', v) is a shortcut, then it is possible for ℓ and ℓ' to coincide. In turn, this implies that the local structure of $\mathcal{N}$ around v is much less restrictive. To establish that Display-Set-Equivalence is solvable in polynomial time for when the input consists of a normal network $\mathcal{N}$ and a tree-child network $\mathcal{N}'$, we have used a detailed analysis of the local structures of $\mathcal{N}$ and $\mathcal{N}'$ relative to a reticulation in $\mathcal{N}$ under the assumption that $T(\mathcal{N}) = T(\mathcal{N}')$.

Now, let $\mathcal{N}$ be a phylogenetic network, and let u be a vertex of $\mathcal{N}$. We say that u is *visible* if there is a leaf, ℓ say, in $\mathcal{N}$ such that every directed path from the root of $\mathcal{N}$ to ℓ traverses u, in which case, ℓ *verifies the visibility of* u. Furthermore, $\mathcal{N}$ is *reticulation-visible* if every reticulation is visible. To summarise, note that normal networks are a proper subclass of tree-child networks and tree-child networks are a proper subclass of reticulation-visible networks. In particular, tree-child networks are precisely the class of networks in which every vertex is visible [3]. As mentioned in the third paragraph of the introduction, Display-Set-Equivalence was recently posed for when $\mathcal{N}$ and $\mathcal{N}'$ are both reticulation-visible and the computational complexity of this problem remains open. Knowing the hurdles that had to be overcome in the proof of Theorem 1, perhaps an easier problem to consider (depending on its complexity) is Display-Set-Equivalence for when $\mathcal{N}$ and $\mathcal{N}'$ are both tree-child.

The paper is organised as follows. Section 2 contains some additional concepts as well as several lemmas concerning tree-child networks. The proof of Theorem 1 is algorithmic and relies on comparing the structures of $\mathcal{N}$ and $\mathcal{N}'$ local to a common pair of leaves. Section 3 establishes the necessary structural results to make these comparisons. Depending on the outcomes of the comparisons, the algorithm recurses in one of three ways. The lemmas associated with these recursions are given in Section 4. The algorithm, its correctness, and its running time, and thus the proof of Theorem 1, are given in the last section. A more detailed overview of the algorithm underlying the proof of Theorem 1 is given at the end of the next section.

2 Preliminaries

Throughout the paper, X denotes a non-empty finite set and all paths are directed. Furthermore, if D is a set and b is an element, we write $D \cup b$ for $D \cup \{b\}$ and $D - b$ for $D - \{b\}$.

Cluster and visibility sets. Let $\mathcal{N}$ be a phylogenetic network on X with root ρ, and let u be a vertex of $\mathcal{N}$. A vertex v is *reachable* from u if there is a path from u to v. The set of leaves reachable from u, denoted C_u, is the *cluster (set) of* u. Furthermore, the set of leaves verifying the visibility of u, denoted V_u, is the *visibility set of* u. Note that the visibility set of u is a subset of the cluster set of u.

Let $\mathcal{T}$ be a phylogenetic X-tree. A non-empty subset C of X is a *cluster of* $\mathcal{T}$ if there is a vertex u in $\mathcal{T}$ such that $C = C_u$. For non-empty (disjoint) subsets Y and Z of X, we say that $\{Y, Z\}$ is a *generalised cherry* of $\mathcal{T}$ if Y, Z, and $Y \cup Z$ are all clusters of $\mathcal{T}$.

Normal and tree-child networks. Let u be a vertex of a phylogenetic network $\mathcal{N}$ on X. A path P starting at u and ending at a leaf is a *tree-path* if every non-terminal vertex is a tree vertex, in which case, u *has a tree-path* and P is a *tree-path for* u. Observe that u may or may not be a reticulation and that every arc in a tree-path is a tree arc. The next lemma is freely-used throughout the paper. Part (ii) is well-known and follows immediately from the definition of a tree-child network, and (iii) was noted in the introduction.

Lemma 2. *Let $\mathcal{N}$ be a phylogenetic network. Then the following statements are equivalent:*

(i) *$\mathcal{N}$ is tree-child,*

(ii) *every vertex of $\mathcal{N}$ has a tree-path, and*

(iii) *every vertex of $\mathcal{N}$ is visible.*

It follows from Lemma 2 that all visibility sets of a tree-child network are non-empty.

Let a and b be distinct leaves of a phylogenetic network $\mathcal{N}$, and let p_a and p_b denote the parents of a and b, respectively. Then $\{a, b\}$ is a *cherry* if $p_a = p_b$. Furthermore, $\{a, b\}$ is a *reticulated cherry* if the parent of one of the leaves, say b, is a reticulation and (p_a, p_b) is an arc in $\mathcal{N}$. Note that, if this holds, then p_a is a tree vertex. The arc (p_a, p_b) is the *reticulation arc* of the reticulated cherry $\{a, b\}$. As with the previous lemma, the next lemma [2] is freely-used throughout the paper.

Lemma 3. *Let $\mathcal{N}$ be a tree-child network on X, where $|X| \geqslant 2$. Then $\mathcal{N}$ has either a cherry or a reticulated cherry.*

The next lemma is established in [19].

Lemma 4. *Let $\mathcal{N}$ be a normal network on X, and let t and u be vertices in $\mathcal{N}$. Then $C_u \subseteq C_t$ if and only if u is reachable from t.*

Let $\mathcal{N}$ be a phylogenetic network on X. Let $\mathcal{S}$ be an embedding in $\mathcal{N}$ of a phylogenetic X-tree $\mathcal{T}$ and let C be a cluster of $\mathcal{T}$. Analogous to cluster sets of $\mathcal{N}$, each vertex w of $\mathcal{S}$ has a *cluster set* and this set consists of the elements in X at the end of a path in $\mathcal{S}$ starting at w. Of course, the cluster set of w relative to $\mathcal{S}$ is a subset of the cluster set of w relative to $\mathcal{N}$. The vertex in $\mathcal{S}$ *corresponding* to C is the (unique) vertex u whose cluster set relative to $\mathcal{S}$ is C and with the property that every other vertex with cluster set C in $\mathcal{S}$ is on a path from the root of $\mathcal{S}$ to u.

Lemma 5. *Let $\mathcal{N}$ be a normal network on X and let u be a tree vertex of $\mathcal{N}$. Let $\mathcal{T}$ be a phylogenetic X-tree having cluster C_u. If $\mathcal{S}$ is an embedding of $\mathcal{T}$ in $\mathcal{N}$, then the vertex in $\mathcal{S}$ corresponding to C_u is u.*

Proof. Suppose that $\mathcal{S}$ is an embedding of $\mathcal{T}$ in $\mathcal{N}$. Let t be the vertex in $\mathcal{S}$ corresponding to C_u, and observe that t is a tree vertex. Clearly, $C_u \subseteq C_t$ and so, by Lemma 4, u is reachable from t on a path P in $\mathcal{N}$. If $t \neq u$, then, as $\mathcal{N}$ is normal and therefore has no shortcuts, t is the parent of a vertex, v say, that is not on P. Now, there is a tree-path from v to a leaf ℓ. By construction, $\ell \notin C_u$. In turn, regardless of whether or not v is a reticulation, this implies that the cluster in $\mathcal{S}$ corresponding to t contains ℓ, a contradiction. Thus $t = u$, thereby completing the proof of the lemma. □

Deleting arcs and leaves. Let $\mathcal{N}$ be a phylogenetic network on X, and let (u, v) be an arc of $\mathcal{N}$. We denote the directed graph obtained from $\mathcal{N}$ by deleting (u, v) and suppressing any resulting vertices with in-degree one and out-degree one by $\mathcal{N}\backslash(u, v)$. Note that $\mathcal{N}\backslash(u, v)$ may have arcs in parallel. If u is the root of $\mathcal{N}$, we additionally delete u (and its incident arc) after deleting (u, v). Extending this notation in the obvious way, we use $\mathcal{N}\backslash\{(u_1, v_1), (u_2, v_2), \ldots, (u_n, v_n)\}$ to denote the directed graph obtained from $\mathcal{N}$ by deleting the arcs $(u_1, v_1), (u_2, v_2), \ldots, (u_n, v_n)$ and suppressing any resulting vertices with in-degree one and out-degree one. Moreover, if b is a leaf of $\mathcal{N}$, then the directed graph obtained from $\mathcal{N}$ by deleting b (and its incident arc), and suppressing any resulting vertex of in-degree one and out-degree one is denoted by $\mathcal{N}\backslash b$. Again, if the parent of b is the root of $\mathcal{N}$, we additionally delete the root (and its incident arc) after deleting b.

Deleting an arc or a leaf of a phylogenetic network does not necessarily result in another phylogenetic network. The next two lemmas, which are also freely used in the paper, give some sufficient conditions for when these operations result in a phylogenetic network. The proof of the first lemma is straightforward and omitted.

Lemma 6. *Let $\mathcal{N}$ be a tree-child network on X, and suppose that $\{a, b\}$ is a cherry of $\mathcal{N}$. Then $\mathcal{N}\backslash b$ is a tree-child network on $X - b$. Moreover, if $\mathcal{N}$ is normal, then $\mathcal{N}\backslash b$ is normal.*

The next lemma generalises a result in [2]. A shortcut (u, v) in a phylogenetic network $\mathcal{N}$ is *trivial* if the parent of v that is not u is a child of u.

Lemma 7. *Let $\mathcal{N}$ be a tree-child network on X, and suppose that (u, v) is a reticulation arc of $\mathcal{N}$.*

(i) *Then* $\mathcal{N}\backslash(u,v)$ *is a tree-child network on* X. *Moreover, if* $\mathcal{N}$ *is normal, then* $\mathcal{N}\backslash(u,v)$ *is normal.*

(ii) *If* (u,v) *is trivial, then* $T(\mathcal{N}) = T(\mathcal{N}\backslash(u,v))$.

Proof. We first prove (i). Since $\mathcal{N}$ is tree-child, u is a tree vertex, the child of u that is not v is either a tree vertex or a leaf, and the unique child of v is either a tree vertex or a leaf. Therefore, as $\mathcal{N}$ has no parallel arcs, $\mathcal{N}\backslash(u,v)$ has no parallel arcs, and so $\mathcal{N}\backslash(u,v)$ is a phylogenetic network. Moreover, it also follows that no new shortcut is created in deleting (u,v) from $\mathcal{N}$. Furthermore, if w is an arbitrary vertex of $\mathcal{N}$, then no tree-path for w in $\mathcal{N}$ traverses (u,v) and so, every vertex in $\mathcal{N}\backslash(u,v)$ has a tree-path. Part (i) now follows.

For (ii), regardless of whether (u,v) is trivial, $T(\mathcal{N}\backslash(u,v)) \subseteq T(\mathcal{N})$. So assume that (u,v) is trivial, in which case (u,u') is an arc in $\mathcal{N}$, and let $\mathcal{T}$ be a phylogenetic tree displayed by $\mathcal{N}$. Let $\mathcal{S}$ be an embedding of $\mathcal{T}$ in $\mathcal{N}$. If $\mathcal{S}$ does not use (u,v), then it is clear that $\mathcal{N}\backslash(u,v)$ displays $\mathcal{T}$. On the other hand, if $\mathcal{S}$ uses (u,v), then by replacing (u,v) with (u',v) we obtain an embedding of $\mathcal{T}$ in $\mathcal{N}$ that does not use (u,v), and so $\mathcal{N}\backslash(u,v)$ displays $\mathcal{T}$. Note that, as $\mathcal{N}$ is tree-child, $\mathcal{S}$ uses (u,u'). Hence $T(\mathcal{N}) \subseteq T(\mathcal{N}\backslash(u,v))$. This completes the proof of (ii). □

We end this section by briefly outlining the algorithm associated with the proof of Theorem 1. Called SameDisplaySet, the algorithm takes as its input normal and tree-child networks $\mathcal{N}$ and $\mathcal{N}'$, respectively, and proceeds by first finding a cherry or a reticulated cherry, $\{a,b\}$ say, in $\mathcal{N}$. It then considers the structure of $\mathcal{N}'$ (and if necessary $\mathcal{N}$) local to leaves a and b, and decides whether to return $T(\mathcal{N}) \neq T(\mathcal{N}')$ or to continue. This decision is based on three Propositions 8, 10, and 11. These propositions give necessary structural properties if $T(\mathcal{N}) = T(\mathcal{N}')$. If the algorithm continues, it deletes certain arcs and leaves in $\mathcal{N}$ and $\mathcal{N}'$. Lemmas 12–14 show that the resulting normal and tree-child networks after the deletions, $\mathcal{N}_1$ and $\mathcal{N}_1'$ say, display the same set of phylogenetic trees, that is $T(\mathcal{N}_1) = T(\mathcal{N}_1')$, if and only if $T(\mathcal{N}) = T(\mathcal{N}')$. The algorithm now recurses on $\mathcal{N}_1$ and $\mathcal{N}_1'$ by finding a cherry or a reticulated cherry of $\mathcal{N}_1$. Eventually, SameDisplaySet either stops and returns $T(\mathcal{N}) \neq T(\mathcal{N}')$ or it reduces $\mathcal{N}$ and $\mathcal{N}'$ to a phylogenetic network consisting of two leaves, in which case $T(\mathcal{N}) = T(\mathcal{N}')$. The formal description of SameDisplaySet is given at the start of Section 5. The reader may choose to refer to that while reading through Sections 3 and 4.

3 Structural Properties

The purpose of this section is to establish three structural results, namely, Propositions 8, 10, and 11. Let $\mathcal{N}$ and $\mathcal{N}'$ be a normal and a tree-child network on X, respectively. Relative to either a cherry or a reticulated cherry, $\{a,b\}$ say, of $\mathcal{N}$, these results determine the structure of $\mathcal{N}'$ local to a and b if $T(\mathcal{N}) = T(\mathcal{N}')$. The first proposition considers when $\{a,b\}$ is a cherry of $\mathcal{N}$, while the second and third propositions consider when $\{a,b\}$

is a reticulated cherry of $\mathcal{N}$ in which the parent of b is a reticulation and the parent of b in $\mathcal{N}'$ is either a reticulation or a tree vertex, respectively. Each of the proofs first considers the parent vertex of b in $\mathcal{N}'$ and establishes sufficient structure of $\mathcal{N}'$ close to b under the assumption that $T(\mathcal{N}) = T(\mathcal{N}')$, so that the iterative algorithm in Section 5 works correctly.

Throughout the proofs in this section, we repeatedly use the following which immediately follows from results in [16]. If $\mathcal{N}$ is a tree-child network on X, then every embedding of a phylogenetic X-tree displayed by $\mathcal{N}$ uses all of the tree arcs and, for each reticulation v, exactly one of the reticulation arcs directed into v. In particular, this implies that if $\mathcal{S}$ is an embedding of a phylogenetic X-tree in $\mathcal{N}$ and P is a tree-path in $\mathcal{N}$, then $\mathcal{S}$ uses every arc in P. Moreover, if $\mathcal{S}'$ is a subset of arcs of $\mathcal{N}$ consisting of all tree arcs and precisely one reticulation arc directed into each reticulation, then $\mathcal{S}'$ is an embedding of a phylogenetic X-tree that is displayed by $\mathcal{N}$. Hence, in the proofs, when we consider an embedding of a phylogenetic X-tree, the focus is on stating which of the two reticulation arcs directed into a reticulation is used.

Proposition 8. *Let $\mathcal{N}$ and $\mathcal{N}'$ be normal and tree-child networks on X, respectively, and suppose $\mathcal{N}'$ has no trivial shortcuts. Let $\{a, b\}$ be a cherry of $\mathcal{N}$. Then $T(\mathcal{N}) = T(\mathcal{N}')$ only if $\{a, b\}$ is a cherry of $\mathcal{N}'$.*

Proof. Suppose $T(\mathcal{N}) = T(\mathcal{N}')$. Note that $\{a, b\}$ is a cherry of every phylogenetic X-tree displayed by $\mathcal{N}$. Let p'_a and p'_b denote the parents of a and b in $\mathcal{N}'$, respectively. First assume that p'_b is a tree vertex. Then, as $T(\mathcal{N}) = T(\mathcal{N}')$, it follows that $C_{p'_b} = \{a, b\}$; otherwise, there is a phylogenetic X-tree displayed by $\mathcal{N}'$ that is not displayed by $\mathcal{N}$. Thus the child vertex of p'_b in $\mathcal{N}'$ that is not b is either a or p'_a. In particular, $\{a, b\}$ is either a cherry or a reticulated cherry with reticulation leaf a in $\mathcal{N}'$. Consider the latter. If q' denotes the parent of p'_a that is not p'_b in $\mathcal{N}'$, then (q', p'_a) is a shortcut. Otherwise, there is a tree-path from q' to a leaf that is not b, and so, using (q', p'_a) and not (p'_b, p'_a) in an embedding of a phylogenetic X-tree, it follows that $\mathcal{N}'$ displays a phylogenetic X-tree in which $\{a, b\}$ is not a cherry. Now let t' denote the child vertex of q' that is not p'_a. Since $\mathcal{N}'$ is tree-child and (q', p'_a) is a shortcut, t' is a tree vertex and $\{a, b\} \subseteq C_{t'}$. If $C_{t'} - a \neq \{b\}$, then, using (q', p'_a) and not (p'_b, p'_a), it follows that $\mathcal{N}'$ displays a phylogenetic X-tree in which $C_{t'} - a$ is a cluster of size at least two containing b, and thus it is not displayed by $\mathcal{N}$. So $C_{t'} - a = \{b\}$ and, in particular, $t' = p'_b$. Thus (q', p'_a) is a trivial shortcut, a contradiction. Therefore if p'_b is a tree vertex, then $\{a, b\}$ is a cherry of $\mathcal{N}'$. If p'_b is a reticulation in $\mathcal{N}'$, then a similar argument leads to the conclusion that $\mathcal{N}'$ has a trivial shortcut. This completes the proof of the proposition. □

We next consider the relative structure local to leaves a and b in $\mathcal{N}$, where $\{a, b\}$ is a reticulated cherry of $\mathcal{N}$. For the next three results, we suppose that $\{a, b\}$ is a reticulated cherry of $\mathcal{N}$ with reticulation leaf b as shown in Fig. 3. Note that, although not shown, if $C_q - (V_q \cup b)$ is nonempty, then $\mathcal{N}$ contains paths from the root ρ to leaves in $C_q - (V_q \cup b)$ avoiding q. Furthermore, in viewing Fig. 3 as well as the other figures in the remainder of the paper, the structure of the phylogenetic network within a box is unknown. However,

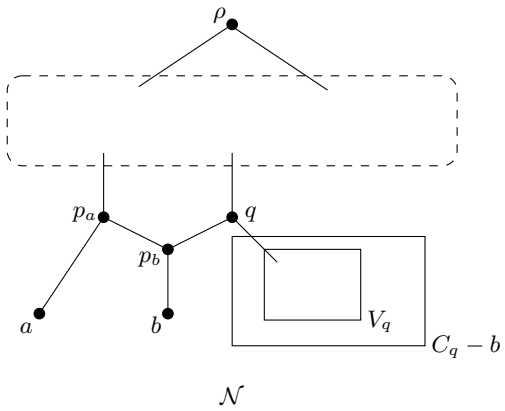


Figure 3: The structure of $\mathcal{N}$ local to the reticulated cherry $\{a, b\}$. Note that, for each leaf $\ell \in C_q - (V_q \cup b)$, there is a path from ρ to ℓ avoiding q.

the label of the box indicates the location of the visibility or cluster set of some particular vertex.

The proof of the next lemma is straightforward and omitted.

Lemma 9. *Let $\mathcal{N}$ be a normal network on X, and suppose that $\{a, b\}$ is a reticulated cherry of $\mathcal{N}$ as shown in Fig. 3. If $\mathcal{T}$ is a phylogenetic X-tree displayed by $\mathcal{N}$, then either*

(i) *$\{a, b\}$ is a cherry of $\mathcal{T}$, or*

(ii) *$\{b, C'_q\}$ is a generalised cherry of $\mathcal{T}$, where $V_q \subseteq C'_q \subseteq C_q - b$ and $a \notin C_q$.*

Moreover, for each $\{A, B\} \in \big\{\{a, b\}, \{b, V_q\}, \{b, C_q - b\}\big\}$, there is a phylogenetic X-tree displayed by $\mathcal{N}$ in which $\{A, B\}$ is a generalised cherry.

Proposition 10. *Let $\mathcal{N}$ and $\mathcal{N}'$ be normal and tree-child networks on X, respectively, and suppose that $\{a, b\}$ is a reticulated cherry of $\mathcal{N}$ as shown in Fig. 3. If the parent of b in $\mathcal{N}'$ is a reticulation, then $T(\mathcal{N}) = T(\mathcal{N}')$ only if, up to isomorphism, $\{a, b\}$ is a reticulated cherry of $\mathcal{N}'$ as shown in Fig. 4, where $V_{q'_2} = V_q$ and $C_{q'_2} = C_q$.*

Proof. Let $\{a, b\}$ be a reticulated cherry of $\mathcal{N}$ as shown in Fig. 3. Thus p_a and p_b denote the parents of a and b in $\mathcal{N}$, respectively, where p_b is a reticulation, and q denotes the parent of p_b in $\mathcal{N}$ that is not p_a. Since $\mathcal{N}$ is normal, (q, p_b) is not a shortcut and $a \notin C_q$. Suppose $T(\mathcal{N}) = T(\mathcal{N}')$, and consider $\mathcal{N}'$. Let p'_a and p'_b denote the parents of a and b in $\mathcal{N}'$, respectively, where p'_b is a reticulation. Let q'_1 and q'_2 denote the parents of p'_b in $\mathcal{N}'$. We will eventually show that one of q'_1 and q'_2, say q'_1, is p'_a.

10.1. *Neither (q'_1, p'_b) nor (q'_2, p'_b) is a shortcut.*

Proof. Assume at least one of (q'_1, p'_b) and (q'_2, p'_b) is a shortcut. Without loss of generality, we may assume (q'_2, p'_b) is a shortcut, and so (q'_1, p'_b) is not a shortcut. Observe that

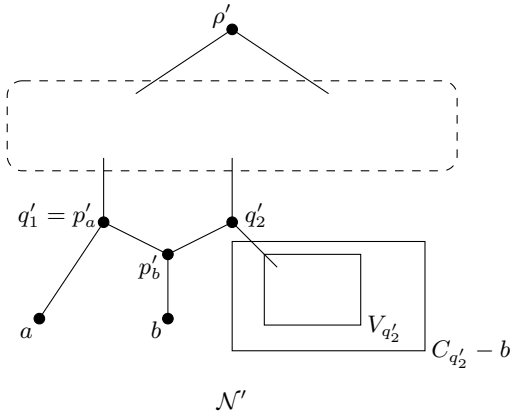


Figure 4: The structure of $\mathcal{N}'$ local to the leaves a and b as established in Proposition 10 when $\mathcal{N}$ is as shown in Fig. 3, the parent of b in $\mathcal{N}'$ is a reticulation, and $T(\mathcal{N}) = T(\mathcal{N}')$. It is also shown that $V_{q'_2} = V_q$ and $C_{q'_2} = C_q$. Note that, for each leaf $\ell \in C_{q'_2} - (V_{q'_2} \cup b)$, there is a path from ρ' to ℓ avoiding q'_2.

$C_{q'_1} \subseteq C_{q'_2}$. Since $T(\mathcal{N}) = T(\mathcal{N}')$, it follows by Lemma 9 that $\mathcal{N}'$ displays a phylogenetic X-tree with $\{a, b\}$ as a cherry and a phylogenetic X-tree with $\{b, V_q\}$ as a generalised cherry. As q'_1 and q'_2 each have a tree-path and every embedding of a phylogenetic X-tree in $\mathcal{N}'$ uses either (q'_1, p'_b) or (q'_2, p'_b), it follows that $V_q \cup a \subseteq C_{q'_2}$. But then, there is an embedding of a phylogenetic X-tree in $\mathcal{N}'$ using (q'_2, p'_b) and not (q'_1, p'_b) which has a generalised cherry $\{b, C_{q'_2} - b\}$. But, $C_{q'_2} - b$ contains $V_q \cup a$ and, by the first part of Lemma 9, $\mathcal{N}$ displays no such tree. Hence neither (q'_1, p'_b) nor (q'_2, p'_b) is a shortcut. □

By (10.1), neither (q'_1, p'_b) nor (q'_2, p'_b) is a shortcut. Therefore, for some $i \in \{1, 2\}$, we have $C_{q'_i} - b = \{a\}$ as $T(\mathcal{N}) = T(\mathcal{N}')$. If not, then one of the following two cases applies.

(i) If $a \notin C_{q'_1} - b$ and $a \notin C_{q'_2} - b$, then, as each of q'_1 and q'_2 has a tree-path, there is no phylogenetic X-tree displayed by $\mathcal{N}'$ with $\{a, b\}$ as a cherry.

(ii) If, for some $i \in \{1, 2\}$, we have $a \in C_{q'_i} - b$ and $|C_{q'_i} - b| \geqslant 2$, then there is a phylogenetic X-tree displayed by $\mathcal{N}'$ in which $\{b, C_{q'_i} - b\}$ is a generalised cherry.

Both cases contradict Lemma 9. Hence, without loss of generality, we may assume that $C_{q'_1} - b = \{a\}$ and so, as $\mathcal{N}'$ is tree-child, $q'_1 = p'_a$. That is, $\{a, b\}$ is a reticulated cherry of $\mathcal{N}'$. Observe that, as (q'_2, p'_b) is not a shortcut by (10.1), we have $a \notin C_{q'_2} - b$.

By Lemma 9, $\mathcal{N}$ displays a phylogenetic X-tree with generalised cherry $\{b, V_q\}$ and so, as $T(\mathcal{N}) = T(\mathcal{N}')$, it follows that $V_q \subseteq C_{q'_2} - b$ and $V_{q'_2} \subseteq V_q$. In turn, as $\mathcal{N}'$ displays a phylogenetic X-tree with generalised cherry $\{b, V_{q'_2}\}$ and $T(\mathcal{N}) = T(\mathcal{N}')$, we have $V_{q'_2} \subseteq C_q - b$ and $V_q \subseteq V_{q'_2}$. Thus $V_q = V_{q'_2}$. Similarly, as $\mathcal{N}$ displays a phylogenetic X-tree with generalised cherry $\{b, C_q - b\}$, and $\mathcal{N}'$ displays a phylogenetic X-tree with generalised cherry $\{b, C_{q'_2} - b\}$, we deduce that $C_q - b \subseteq C_{q'_2} - b$ and $C_{q'_2} - b \subseteq C_q - b$, so $C_q - b = C_{q'_2} - b$. Thus $\{a, b\}$ is a reticulated cherry of $\mathcal{N}'$ as shown in Fig. 4 with $V_{q'_2} = V_q$ and $C_{q'_2} = C_q$, and this completes the proof of the proposition. □

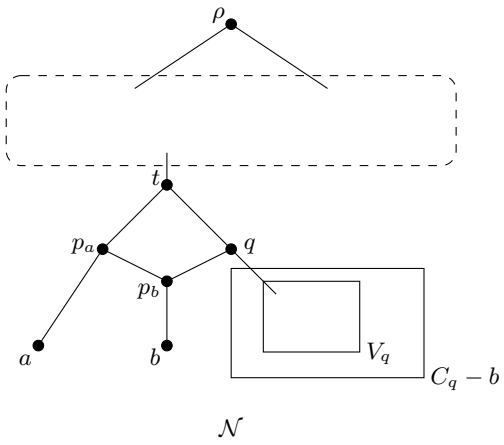


Figure 5: Additional structure of $\mathcal{N}$ local to the leaves a and b as shown in Proposition 11 when $\{a, b\}$ is a reticulated cherry of $\mathcal{N}$ as shown in Fig. 3, the parent of b in $\mathcal{N}'$ is a tree vertex, and $T(\mathcal{N}) = T(\mathcal{N}')$. Note that, for each leaf $\ell \in C_q - (V_q - b)$, there is a path from ρ to ℓ avoiding q.

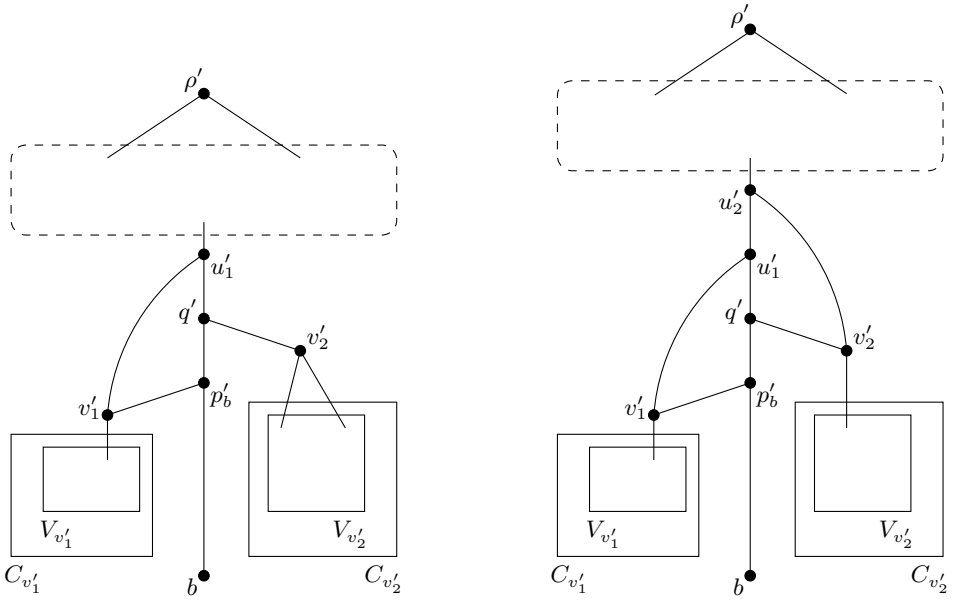


Figure 6: The two possible structures of $\mathcal{N}'$ local to the leaves a and b as shown in Proposition 11 when $\mathcal{N}$ is as shown in Fig. 3, the parent of b in $\mathcal{N}'$ is a tree vertex, and $T(\mathcal{N}) = T(\mathcal{N}')$. It is also shown that $\{V_{v'_1}, V_{v'_2}\} = \{\{a\}, V_q\}$ and $\{C_{v'_1}, C_{v'_2}\} = \{\{a\}, C_q - b\}$. Note that, if $C_{v'_i} \neq \{a\}$ for some $i \in \{1, 2\}$, then, for each leaf $\ell \in C_{v'_i} - V_{v'_i}$, there is a path from ρ' to ℓ avoiding v'_i. Furthermore, in (a), v'_2 could be a leaf.

Proposition 11. *Let $\mathcal{N}$ and $\mathcal{N}'$ be normal and tree-child networks on X, respectively, and suppose that $\mathcal{N}'$ has no trivial shortcuts and $\{a, b\}$ is a reticulated cherry of $\mathcal{N}$ as shown in Fig. 3. If the parent of b in $\mathcal{N}'$ is a tree vertex, then $T(\mathcal{N}) = T(\mathcal{N}')$ only if, up to isomorphism, in $\mathcal{N}$, leaves a and b are as shown in Fig. 5 and, in $\mathcal{N}'$, leaves a and b are as shown in either Fig. 6(a) or Fig. 6(b), where $\{V_{v'_1}, V_{v'_2}\} = \{\{a\}, V_q\}$ and $\{C_{v'_1}, C_{v'_2}\} = \{\{a\}, C_q - b\}$.*

Proof. Let $\{a, b\}$ be a reticulated cherry of $\mathcal{N}$ as shown in Fig. 3, and suppose that $T(\mathcal{N}) = T(\mathcal{N}')$. Let p'_b denote the parent of b in $\mathcal{N}'$, and suppose that p'_b is a tree vertex. Let v'_1 denote the child of p'_b in $\mathcal{N}'$ that is not b. If v'_1 is a tree vertex or a leaf, then either there is no phylogenetic X-tree displayed by $\mathcal{N}'$ in which $\{a, b\}$ is a cherry or there is no phylogenetic X-tree displayed by $\mathcal{N}'$ in which $\{b, V_q\}$ is a generalised cherry. This contradiction to Lemma 9 implies that we may assume v'_1 is a reticulation.

11.1. *Either $C_{v'_1} = \{a\}$ or $V_{v'_1} = V_q$.*

Proof. Using the arc (p'_b, v'_1), there are embeddings of phylogenetic X-trees in $\mathcal{N}'$ in which $\{b, C_{v'_1}\}$ and $\{b, V_{v'_1}\}$ are generalised cherries. Thus, as $T(\mathcal{N}) = T(\mathcal{N}')$, Lemma 9 implies that if $a \in C_{v'_1}$, then $C_{v'_1} = \{a\}$. Furthermore, by the same lemma, if $a \notin C_{v'_1}$, then $V_q \subseteq V_{v'_1}$. But, using (q, p_b) and not (p_a, p_b), there is an embedding of a phylogenetic X-tree in $\mathcal{N}$ in which $\{b, V_q\}$ is a generalised cherry and so, as $T(\mathcal{N}) = T(\mathcal{N}')$, we also have $V_{v'_1} \subseteq V_q$. Hence if $a \notin C_{v'_1}$, then $V_{v'_1} = V_q$. □

Since v'_1 is a reticulation, p'_b is not the root of $\mathcal{N}'$. Let q' denote the parent of p'_b in $\mathcal{N}'$.

11.2. *The vertex q' is either the root of $\mathcal{N}'$ or a tree vertex.*

Proof. Suppose that q' is a reticulation, and let u'_1 and u'_2 denote the parents of q'. First assume that neither (u'_1, q') nor (u'_2, q') is a shortcut. Let ℓ_1 and ℓ_2 be leaves at the end of tree-paths for u'_1 and u'_2, respectively. Note that $\ell_1 \neq \ell_2$ and $\ell_1, \ell_2 \notin C_{v'_1}$.For each $i \in \{1, 2\}$, let $\mathcal{T}_i$ be a phylogenetic X-tree displayed by $\mathcal{N}'$ for which an embedding uses (u'_i, q') and not (p'_b, v'_1). If $V_q = V_{v'_1}$, then by the first part of Lemma 9, either $\mathcal{T}_1$ or $\mathcal{T}_2$ is not displayed by $\mathcal{N}$. Hence, by (11.1), we may assume that $C_{v'_1} = \{a\}$. Since $\mathcal{T}_1$ is displayed by $\mathcal{N}$, we now deduce by the first part of Lemma 9 again that $\ell_1 \in V_q$. But then $\mathcal{T}_2$ has a generalised cherry $\{b, C'_{u'_2}\}$, where $V_q \not\subseteq C'_{u'_2} \subseteq C_{u'_2}$ as $\ell_1 \notin C_{u'_2}$, contradicting Lemma 9. Hence, without loss of generality, we may assume that (u'_2, q') is a shortcut.

Using the arc (u'_1, q') but not (p'_b, v'_1), there are embeddings of phylogenetic X-trees in $\mathcal{N}'$ in which $\{b, C_{u'_1} - b\}$ and $\{b, V_{u'_1}\}$ are generalised cherries. Therefore, as $T(\mathcal{N}) = T(\mathcal{N}')$, it follows by Lemma 9 that if $a \in C_{u'_1}$, then $C_{u'_1} - b = \{a\}$. Moreover, if $a \notin C_{u'_1}$, then, again by Lemma 9, $V_q \subseteq V_{u'_1}$. But $\mathcal{N}$ displays a phylogenetic X-tree in which $\{b, V_q\}$ is a generalised cherry and so, as $T(\mathcal{N}) = T(\mathcal{N}')$, we have $V_{u'_1} \subseteq V_q$. Hence if $a \notin C_{u'_1}$, then $V_{u'_1} = V_q$.

If (u'_2, u'_1) is an arc of $\mathcal{N}'$, then (u'_2, q') is a trivial shortcut. Therefore we may assume that (u'_2, u'_1) is not an arc. Let P' be a path in $\mathcal{N}'$ from u'_2 to u'_1. Since (u'_2, u'_1) is not an arc, P' contains at least one vertex, w' say, in addition to u'_2 and u'_1. Choose w' to be

the first such vertex on P' that has a child that does not lie on P'. As $\mathcal{N}'$ is tree-child, and so each vertex has a tree-path, it is easily checked that w' exists and $w' \neq u'_1$. Let x' denote the child of w' that does not lie on P'. Since x' has a tree-path, there is a leaf in $C_{u'_2}$ that is not in $C_{u'_1}$, that is, $C_{u'_1}$ is a proper subset of $C_{u'_2}$.

Using (u'_2, q') and not (p'_b, v'_1), it is easily seen that there is an embedding of a phylogenetic X-tree in $\mathcal{N}'$ in which $\{b, C_{u'_2} - b\}$ is a generalised cherry. If $C_{u'_1} = \{a\}$, then $a \in C_{u'_2}$ but $|C_{u'_2} - b| \geqslant 2$. Since $T(\mathcal{N}) = T(\mathcal{N}')$, this contradicts the first part of Lemma 9. Thus, $a \notin C_{u'_1}$, and so, by (11.1), $C_{v'_1} = \{a\}$ and $V_{u'_1} = V_q$, in which case, by the first part of Lemma 9, $C_{u'_2} - b \subseteq C_q - b$. On the other hand, $\mathcal{N}$ displays a phylogenetic X-tree $\mathcal{T}$ in which $\{b, C_q - b\}$ is a generalised cherry. Since $V_{u'_1} = V_q$, it follows that, for $\mathcal{N}'$ to display $\mathcal{T}$, we must have $C_q - b \subseteq C_{u'_2} - b$. Thus $C_q - b = C_{u'_2} - b$.

Now using (u'_1, q'), (w', x'), and the arcs on P', but not using (p'_b, v'_1), there is an embedding of a phylogenetic X-tree $\mathcal{T}'$ in $\mathcal{N}'$ that has two distinct clusters $b \cup C_{u'_1}$ and $C_{u'_2}$. But, by Lemma 5, if $\mathcal{S}'$ is an embedding of $\mathcal{T}'$ in $\mathcal{N}$, then the vertex of $\mathcal{S}'$ corresponding to $C_{u'_2}$ is q as $C_{u'_2} = C_q$, but then there is no distinct vertex in $\mathcal{N}$ that corresponds to $b \cup C_{u'_1}$. In particular, $\mathcal{N}$ does not display $\mathcal{T}'$. This completes the proof of (11.2). □

By (11.2), q' is either the root of $\mathcal{N}'$ or a tree vertex. Let v'_2 be the child of q' that is not p'_b. Note that $v'_1 \neq v'_2$; otherwise, (q', v'_2) is a trivial shortcut. Using the arc (q', v'_2) and not (p'_b, v'_1), there are embeddings of phylogenetic X-tree in $\mathcal{N}'$ in which $\{b, C_{v'_2}\}$ and $\{b, V_{v'_2}\}$ are generalised cherries. Therefore, by the first part of Lemma 9, if $a \in C_{v'_2}$, then $C_{v'_2} = \{a\}$. Furthermore, if $a \notin C_{v'_2}$, then, by the same Lemma 9, $V_q \subseteq V_{v'_2}$. But $\mathcal{N}$ displays a phylogenetic X-tree in which $\{b, V_q\}$ is a generalised cherry and so, by Lemma 9 again, we have $V_{v'_2} \subseteq V_q$. Thus if $a \notin C_{v'_2}$, then $V_{v'_2} = V_q$. In combination with (11.1), we now have

11.3. $\{V_{v'_1}, V_{v'_2}\} = \{\{a\}, V_q\}$. *Furthermore, if* $V_{v'_i} = \{a\}$, *then* $C_{v'_i} = \{a\}$ *for each* $i \in \{1, 2\}$.

Using arcs (p'_b, v'_1) and (q', v'_2), there is an embedding of a phylogenetic X-tree $\mathcal{T}'$ in $\mathcal{N}'$ with generalised cherries $\{b, V_{v'_1}\}$ and $\{V_{v'_1} \cup b, V_{v'_2}\}$. Since $T(\mathcal{N}) = T(\mathcal{N}')$, it follows that $\mathcal{N}$ displays $\mathcal{T}'$ as well. But then, by considering an embedding of $\mathcal{T}'$ in $\mathcal{N}$ together with (11.3), it is easily seen that $\mathcal{N}$, and therefore $\mathcal{N}'$, displays a phylogenetic X-tree $\mathcal{T}$ with generalised cherries $\{b, V_{v'_2}\}$ and $\{V_{v'_2} \cup b, V_{v'_1}\}$. To see this, observe that an embedding of $\mathcal{T}$ in $\mathcal{N}$ can be obtained from an embedding of $\mathcal{T}'$ in $\mathcal{N}$ by either deleting (p_a, p_b) and adding (q, p_b), or deleting (q, p_b) and adding (p_a, p_b). It follows that q' is not the root of $\mathcal{N}'$. Let u'_1 be the parent of v'_1 that is not p'_b.

11.4. *The arc* (u'_1, v'_1) *is a shortcut in* $\mathcal{N}'$. *In particular,* (u'_1, q') *is an arc in* $\mathcal{N}'$.

Proof. Consider an embedding $\mathcal{S}'$ of $\mathcal{T}$ in $\mathcal{N}'$, where $\mathcal{T}$ is the phylogenetic X-tree with generalised cherries $\{b, V_{v'_2}\}$ and $\{V_{v'_2} \cup b, V_{v'_1}\}$. Clearly, $\mathcal{S}'$ uses (u'_1, v'_1) and not (p'_b, v'_1). If (u'_1, v'_1) is not a shortcut, then $\mathcal{N}'$ has a tree-path from u'_1 to a leaf that is not in $V_q \cup a$. But then $\mathcal{S}'$ is not an embedding of $\mathcal{T}$ in $\mathcal{N}'$. Thus (u'_1, v'_1) is a shortcut in $\mathcal{N}'$.

Now, in $\mathcal{N}'$, there is a tree-path from u'_1 to a leaf ℓ. Since $\mathcal{S}'$ is an embedding of $\mathcal{T}$ in $\mathcal{N}'$, it is easily checked that either $\ell = b$, or v'_2 is a tree vertex and ℓ is at the end of

a tree-path for v_2'. Both possibilities imply that there is a tree-path P' in $\mathcal{N}'$ from u_1' to b. Let t' denote the parent of q' and observe that t' is on P'. We next show that $t' = u_1'$. Towards a contradiction, assume that $t' \neq u_1'$. Let w' be the child of t' that is not q'. If $w' = v_2'$, then $\mathcal{N}'$ has a trivial shortcut, so $w' \neq v_2'$. It follows by (11.3) that there is a tree-path from w' to a leaf ℓ' such that $\ell' \notin V_q \cup a$. Using (u_1', v_1'), (q', v_2'), (t', w'), and the arcs on P', there is an embedding of a phylogenetic X-tree $\mathcal{T}_1'$ in $\mathcal{N}'$ with generalised cherries $\{b, V_{v_2'}\}$ and $\{V_{v_2'} \cup b, V_{w'}\}$. Note that $\ell' \in V_{w'}$. By considering an embedding of $\mathcal{T}_1'$ in $\mathcal{N}$, it is easily seen that $\mathcal{N}$, and therefore $\mathcal{N}'$ displays a phylogenetic X-tree $\mathcal{T}_1$ with generalised cherries $\{b, V_{v_1'}\}$ and $\{V_{v_2'}, V_{w'}\}$. If v_2' is a tree vertex in $\mathcal{N}'$, then $\mathcal{N}'$ does not display $\mathcal{T}_1$. Therefore we may assume that v_2' is a reticulation in $\mathcal{N}'$.

If w' is not reachable from v_2', then $\ell' \notin C_q \cup a$, in which case, using (u_1', v_1'), (t', w'), the arcs on P', but not (q', v_2'), we deduce that there is an embedding of a phylogenetic X-tree in $\mathcal{N}'$ with a generalised cherry $\{b, C_{w'}\}$, where $\ell' \in C_{w'}$. This contradiction to the first part of Lemma 9 implies w' is reachable from v_2'. But then using (u_1', v_1'), (t', w'), the arcs on P', but not (q', v_2'), it follows that there is an embedding of a phylogenetic X-tree in $\mathcal{N}'$ such that neither $\{a, b\}$ nor $\{b, C_q'\}$, where $V_q \subseteq C_q'$, is a generalised cherry. But then, as $T(\mathcal{N}) = T(\mathcal{N}')$, we again obtain a contradiction to the first part of Lemma 9. Hence $t' = u_1'$, that is (u_1', q') is an arc in $\mathcal{N}'$. □

We next establish the additional structure of $\mathcal{N}$ as shown in Fig. 5. Let $\mathcal{S}$ be an embedding of $\mathcal{T}$ in $\mathcal{N}$, where $\mathcal{T}$ is still the phylogenetic X-tree with generalised cherries $\{b, V_{v_2'}\}$ and $\{V_{v_2'} \cup b, V_{v_1'}\}$. Let t denote the tree vertex in $\mathcal{S}$ corresponding to the cluster $V_q \cup \{a, b\}$, and let P_a and P_q denote the paths in $\mathcal{S}$ from t to p_a and t to q, respectively.

11.5. *In $\mathcal{N}$, the paths P_a and P_q consist of the arcs (t, p_a) and (t, q), respectively.*

Proof. We begin by observing that, apart from p_a and q, there is no vertex on either P_a or P_q which is the start of a tree-path to a leaf avoiding p_a and q. Otherwise, $\mathcal{S}$ is not an embedding of $\mathcal{T}$ in $\mathcal{N}$. First consider P_a, and suppose that (t, u) is an arc on P_a, where $u \neq p_a$. Assume u is a tree vertex. Then u has a child vertex, w say, that is not on either P_a or P_q. To see this, if u has both of its child vertices on P_a, then one of its children is a reticulation, and so there is a tree-path from u to a leaf avoiding p_a and q, a contradiction. Furthermore, if u has a child vertex on P_q, then either $\mathcal{N}$ has a trivial shortcut or there is a tree-path from a vertex on P_q to a leaf avoiding p_a and q, another contradiction. Now, there is a tree-path from w to a leaf ℓ_w such that $\ell_w \notin \{a, b\} \cup V_q$. By (11.3), either $C_{v_1'} = \{a\}$ or $C_{v_2'} = \{a\}$. If $C_{v_1'} = \{a\}$, then, by using (q, p_b), the arcs on P_a and P_q, and (u, w), it is easily checked that there is an embedding of a phylogenetic X-tree in $\mathcal{N}$ that is not displayed by $\mathcal{N}'$. Moreover, if $C_{v_2'} = \{a\}$, then, by using (p_a, p_b), the arcs on P_a and P_q, and (u, w), it is again easily checked that there is an embedding of a phylogenetic X-tree in $\mathcal{N}$ that is not displayed by $\mathcal{N}'$. These contradictions imply that u is not a tree vertex.

Now assume that u is a reticulation. Let s denote the parent of u that is not t. Since $\mathcal{N}$ is acyclic, s is not on P_a. Also, s is not on P_q; otherwise, (t, u) is shortcut, contradicting that $\mathcal{N}$ is normal. As $\mathcal{N}$ is normal, (s, u) is not a shortcut and so there is a tree-path from s to a leaf ℓ_s, where $\ell_s \notin \{a, b\} \cup C_q$. Note that ℓ_s is not reachable from q; otherwise,

s is reachable from q and so (t, u) is a shortcut in the normal network $\mathcal{N}$, contradiction. Applying essentially the same argument to that when u is a tree vertex, we again obtain a contradiction to $T(\mathcal{N}) = T(\mathcal{N}')$ and conclude that P_a consists of the arc (t, p_a).

Now consider P_q and suppose that (t, u) is an arc on P_q. If u is a tree vertex, then there is a child vertex, w say, of u that is not on P_q, and so there is a tree-path from u to a leaf ℓ_w, where $\ell_w \notin V_q \cup \{a, b\}$. If $C_{v'_1} = \{a\}$, then, by using (q, p_b), the arcs on P_q, and (u, w), it is easily seen that there is an embedding of a phylogenetic X-tree in $\mathcal{N}$ that is not displayed by $\mathcal{N}'$. Moreover, if $C_{v'_2} = \{a\}$, then, by using (p_a, p_b), the arcs on P_q, and (u, w), it is again easily seen that there is an embedding of a phylogenetic X-tree in $\mathcal{N}$ that is not displayed by $\mathcal{N}'$. These contradictions imply that u is not a tree vertex, and so we may assume that u is a reticulation. Let s denote the parent of u that is not t. As $\mathcal{N}$ is normal, (s, u) is not a shortcut and there is a tree-path from s to a leaf ℓ_s, where $\ell_s \notin C_q \cup \{a, b\}$. Note that s is not reachable from q; otherwise, $\mathcal{N}$ has a directed cycle. Applying essentially the same argument to that when u is a tree vertex, we conclude that P_q consists of the arc (t, q). This completes the proof of (11.5). □

We complete the proof of Proposition 11 by considering v'_2 in $\mathcal{N}'$. First assume that v'_2 is a tree vertex or a leaf. Then, as $T(\mathcal{N}) = T(\mathcal{N}')$ and $\{V_{v'_1}, V_{v'_2}\} = \{\{a\}, V_q\}$, it follows that $\{C_{v'_1}, C_{v'_2}\} = \{\{a\}, C_q - b\}$. In particular, in combination with (11.3) we have the outcome shown in Fig. 6(a). Now assume that v'_2 is a reticulation. Let u'_2 denote the parent of v'_2 that is not q'. If (u'_2, v'_2) is not a shortcut, then there is a tree-path from u'_2 to a leaf not in $\{a, b\} \cup C_q$, in which case, by using (u'_2, v'_2) and not (q', v'_2), it follows from (11.5) that there is an embedding of a phylogenetic X-tree in $\mathcal{N}'$ not displayed by $\mathcal{N}$, a contradiction. So (u'_2, v'_2) is a shortcut. As $\mathcal{N}'$ is tree-child, u'_2 is a tree vertex and the child vertex of u'_2 that is not v'_2, say w', is also a tree vertex. If $w' \neq u'_1$, then there is a child vertex y' of w' that is the initial vertex of a tree-path to a leaf not in $\{a, b\} \cup C_q$. But then, by using (u'_2, v'_2) and (w', y'), it follows from (11.5) that there is an embedding of a phylogenetic X-tree in $\mathcal{N}'$ that is not displayed by $\mathcal{N}$, a contradiction. Thus $w' = u'_1$, and so (u'_2, u'_1) is an arc in $\mathcal{N}'$. Furthermore, as $T(\mathcal{N}) = T(\mathcal{N}')$, it follows that if $C_{v'_1} = V_{v'_1} = \{a\}$, then $C_{v'_2} = C_q$, while if $C_{v'_2} = V_{v'_2} = \{a\}$, then $C_{v'_1} = C_q$. Thus we have the outcome shown in Fig. 6(b), thereby completing the proof of the proposition. □

4 Recursion Lemmas

With the structural outcomes of Propositions 8, 10, and 11 in hand, we next establish the three lemmas that will allow the algorithm to recurse correctly. The proof of the first lemma is straightforward and omitted.

Lemma 12. *Let $\mathcal{N}$ and $\mathcal{N}'$ be normal and tree-child networks on X, respectively, and suppose that $\{a, b\}$ is a cherry of $\mathcal{N}$ and $\mathcal{N}'$. Then $T(\mathcal{N}) = T(\mathcal{N}')$ if and only if $T(\mathcal{N} \backslash b) = T(\mathcal{N}' \backslash b)$.*

Lemma 13. *Let $\mathcal{N}$ and $\mathcal{N}'$ be normal and tree-child networks on X, and suppose that $\{a, b\}$ is a reticulated cherry of $\mathcal{N}$ and $\mathcal{N}'$ as shown in Figs. 3 and 4, respectively. Then $T(\mathcal{N}) = T(\mathcal{N}')$ if and only if $T(\mathcal{N} \backslash (p_a, p_b)) = T(\mathcal{N}' \backslash (p'_a, p'_b))$.*

Proof. First observe that $T(\mathcal{N}) - T(\mathcal{N}\backslash(p_a, p_b))$ (resp. $T(\mathcal{N}') - T(\mathcal{N}'\backslash(p'_a, p'_b))$) consists of precisely the phylogenetic X-trees displayed by $\mathcal{N}$ (resp. $\mathcal{N}'$) in which $\{a, b\}$ is a cherry. Thus if $T(\mathcal{N}) = T(\mathcal{N}')$, then $T(\mathcal{N}\backslash(p_a, p_b)) = T(\mathcal{N}'\backslash(p'_a, p'_b))$. Suppose $T(\mathcal{N}\backslash(p_a, p_b)) = T(\mathcal{N}'\backslash(p'_a, p'_b))$, and let $\mathcal{T}$ be a phylogenetic X-tree displayed by $\mathcal{N}$. If $\{a, b\}$ is not a cherry in $\mathcal{T}$, then, by the observation, $\mathcal{N}\backslash(p_a, p_b)$, and therefore $\mathcal{N}'\backslash(p'_a, p'_b)$, displays $\mathcal{T}$. This implies that $\mathcal{N}'$ displays $\mathcal{T}$. So assume $\{a, b\}$ is a cherry in $\mathcal{T}$. Let $\mathcal{S}$ be an embedding of $\mathcal{T}$ in $\mathcal{N}$. Note that $\mathcal{S}$ must use the arc (p_a, p_b). Let $\mathcal{S}_1$ be the embedding in $\mathcal{N}$ of a phylogenetic X-tree $\mathcal{T}_1$ obtained from $\mathcal{S}$ by deleting (p_a, p_b) and adding (q, p_b). Since $\{a, b\}$ is not a cherry of $\mathcal{T}_1$, it follows that $\mathcal{N}'$ displays $\mathcal{T}_1$, that is, $\mathcal{N}'$ has an embedding $\mathcal{S}'_1$ of $\mathcal{T}_1$. Now, by replacing (q'_2, p'_b) with (p'_a, p'_b) in $\mathcal{S}'_1$, we have an embedding of $\mathcal{T}$ in $\mathcal{N}'$. Hence $\mathcal{N}'$ displays $\mathcal{T}$, and so $T(\mathcal{N}) \subseteq T(\mathcal{N}')$. Similarly, $T(\mathcal{N}') \subseteq T(\mathcal{N})$. Thus $T(\mathcal{N}) = T(\mathcal{N}')$. □

Lemma 14. *Let $\mathcal{N}$ and $\mathcal{N}'$ be normal and tree-child networks on X, respectively. Suppose that $\{a, b\}$ is a reticulated cherry of $\mathcal{N}$ as shown in Fig. 5, while $\mathcal{N}'$ has the structure local to leaves a and b as shown in either Fig. 6(a) or Fig. 6(b).*

(i) *If $C_{v'_1} = \{a\}$, then $T(\mathcal{N}) = T(\mathcal{N}')$ if and only if*

$$T(\mathcal{N}\backslash(p_a, p_b)) = \begin{cases} T(\mathcal{N}'\backslash(p'_b, v'_1)), & v'_2 \text{ a tree vertex or a leaf;} \\ T(\mathcal{N}'\backslash\{(p'_b, v'_1), (u'_2, v'_2)\}), & \text{otherwise.} \end{cases}$$

(ii) *If $C_{v'_2} = \{a\}$, then $T(\mathcal{N}) = T(\mathcal{N}')$ if and only if*

$$T(\mathcal{N}\backslash(p_a, p_b)) = \begin{cases} T(\mathcal{N}'\backslash(u'_1, v'_1)), & v'_2 \text{ a tree vertex or a leaf;} \\ T(\mathcal{N}'\backslash\{(u'_1, v'_1), (u'_2, v'_2)\}), & \text{otherwise.} \end{cases}$$

Proof. We shall prove (i). The proof of (ii) is similar and omitted. Suppose $C_{v'_1} = \{a\}$. For convenience, let $\mathcal{N}_1$ denote $\mathcal{N}\backslash(p_a, p_b)$. Furthermore, let $\mathcal{N}'_1$ denote $\mathcal{N}'\backslash(p'_b, v'_1)$ if v'_2 is a tree vertex or a leaf; otherwise, let $\mathcal{N}'_1$ denote $\mathcal{N}'\backslash\{(p'_b, v'_1), (u'_2, v'_2)\}$. We begin by observing that $T(\mathcal{N}) - T(\mathcal{N}_1)$ (resp. $T(\mathcal{N}') - T(\mathcal{N}'_1)$) consists of precisely the phylogenetic X-trees displayed by $\mathcal{N}$ (resp. $\mathcal{N}'$) in which $\{a, b\}$ is a cherry. Therefore if $T(\mathcal{N}) = T(\mathcal{N}')$, then $T(\mathcal{N}_1) = T(\mathcal{N}'_1)$.

For the converse, suppose that $T(\mathcal{N}_1) = T(\mathcal{N}'_1)$. Let $\mathcal{T}$ be a phylogenetic X-tree displayed by $\mathcal{N}$. If $\{a, b\}$ is not a cherry in $\mathcal{T}$, then, by the observation, $\mathcal{N}_1$, and therefore $\mathcal{N}'_1$, displays $\mathcal{T}$. It follows that $\mathcal{N}'$ displays $\mathcal{T}$. So assume $\{a, b\}$ is a cherry in $\mathcal{T}$. Let $\mathcal{S}$ be an embedding of $\mathcal{T}$ in $\mathcal{N}$. Since $\{a, b\}$ is a cherry in $\mathcal{T}$, the embedding $\mathcal{S}$ uses (p_a, p_b). Let $\mathcal{S}_1$ denote the embedding in $\mathcal{N}$ of a phylogenetic X-tree $\mathcal{T}_1$ obtained from $\mathcal{S}$ by deleting (p_a, p_b) and adding (q, p_b). Since $\{a, b\}$ is not a cherry in $\mathcal{T}_1$, it follows that $\mathcal{N}'$ has an embedding $\mathcal{S}'_1$ of $\mathcal{T}_1$. This embedding $\mathcal{S}'_1$ must use (u'_1, v'_1). By replacing (u'_1, v'_1) with (p'_b, v'_1) in $\mathcal{S}'_1$, it is easily seen that we have an embedding of $\mathcal{T}$ in $\mathcal{N}'$. Hence $\mathcal{N}'$ displays $\mathcal{T}$ and so $T(\mathcal{N}) \subseteq T(\mathcal{N}')$.

Now let $\mathcal{T}'$ be a phylogenetic X-tree displayed by $\mathcal{N}'$. If $\{a, b\}$ is not a cherry, then, by the observation, $\mathcal{N}_1'$, and therefore $\mathcal{N}_1$, displays $\mathcal{T}'$. So $\mathcal{N}$ displays $\mathcal{T}'$. Assume $\{a, b\}$ is a cherry in $\mathcal{T}'$. Let $\mathcal{S}'$ be an embedding of $\mathcal{T}'$ in $\mathcal{N}'$. As $\{a, b\}$ is a cherry in $\mathcal{T}'$ and as any embedding of $\mathcal{T}'$ in $\mathcal{N}'$ must use (u_1', q'), (q', p_b'), (p_b', b) and, if it exists, (u_2', u_1'), it is easily seen that we may choose $\mathcal{S}'$ so that it uses (p_b', v_1') and (q', v_2'). Let $\mathcal{S}_1'$ be the embedding in $\mathcal{N}'$ of a phylogenetic X-tree $\mathcal{T}_1'$ obtained from $\mathcal{S}'$ by deleting (p_b', v_1') and adding (u_1', v_1'). Since $\{a, b\}$ is not a cherry in $\mathcal{T}_1'$, it follows that $\mathcal{N}$ has an embedding $\mathcal{S}_1$ of $\mathcal{T}_1'$. This embedding $\mathcal{S}_1$ must use (q, p_b). By replacing (q, p_b) with (p_a, p_b) in $\mathcal{S}_1$, it is easily checked that we obtain an embedding of $\mathcal{T}'$ in $\mathcal{N}$. Thus $\mathcal{N}$ displays $\mathcal{T}'$, and so $T(\mathcal{N}') \subseteq T(\mathcal{N})$. We conclude that $T(\mathcal{N}) = T(\mathcal{N}')$. □

5 The Algorithm

We now give a formal description of the algorithm SameDisplaySet for deciding if $T(\mathcal{N}) = T(\mathcal{N}')$, where $\mathcal{N}$ and $\mathcal{N}'$ are normal and tree-child networks on X, respectively. Immediately after the description of the algorithm, we show that SameDisplaySet works correctly and analyse its running time. We end the section by briefly describing how to construct a tree displayed by exactly one of $\mathcal{N}$ and $\mathcal{N}'$ if $T(\mathcal{N}) \neq T(\mathcal{N}')$.

SameDisplaySet
Input: Normal and tree-child networks $\mathcal{N}$ and $\mathcal{N}'$ on X, respectively.
Output: *No* if $T(N) \neq T(\mathcal{N}')$, and *Yes* if $T(\mathcal{N}) = T(\mathcal{N}')$.

1. Delete all trivial shortcuts in $\mathcal{N}'$ and suppress all resulting vertices of in-degree one and out-degree one, and denote the resulting normal and tree-child networks on X as $\mathcal{N}_0$ and $\mathcal{N}_0'$, respectively.

2. Set $i = 0$.

3. If the leaf set of $\mathcal{N}_i$ has size two, return *yes*. Else, find a cherry or a reticulated cherry, say $\{a, b\}$, of $\mathcal{N}_i$.

4. If $\{a, b\}$ is a cherry, then determine if $\{a, b\}$ is a cherry of $\mathcal{N}_i'$.

   (a) If no, then return *No*.

   (b) Else, set $\mathcal{N}_{i+1} = \mathcal{N}_i \backslash b$ and set $\mathcal{N}_{i+1}' = \mathcal{N}_i' \backslash b$. Go to Step 6.

5. Else, $\{a, b\}$ is a reticulated cherry of $\mathcal{N}_i$, where the parent of b is a reticulation.

   (a) If the parent of b in $\mathcal{N}_i'$ is a reticulation, then determine if, up to isomorphism, the structure in $\mathcal{N}_i'$ local to a and b is as shown in Fig. 4.

      (i) If no, then return *No*.

      (ii) Else, set $\mathcal{N}_{i+1} = \mathcal{N}_i \backslash (p_a, p_b)$ and set $\mathcal{N}_{i+1}' = \mathcal{N}_i' \backslash (p_a', p_b')$. Go to Step 6.

   (b) If the parent of b in $\mathcal{N}_i'$ is the root, then return *No*.

(c) Else, the parent of b in $\mathcal{N}_i'$ is a tree vertex. Determine if, up to isomorphism, the structures in $\mathcal{N}_i$ and $\mathcal{N}_i'$ local to a and b are as shown in Fig. 5 and Fig. 6(a) or Fig. 6(b), respectively.

(i) If no, then return *No*.

(ii) Else, set $\mathcal{N}_{i+1}$ to be the normal network $\mathcal{N}_i \backslash (p_a, p_b)$. Further, if $C_{v_1'} = \{a\}$, set $\mathcal{N}_{i+1}'$ to be the tree-child network $\mathcal{N}_i' \backslash \{(p_b', v_1'), (u_2', v_2')\}$. Otherwise, if $C_{v_2'} = \{a\}$, set $\mathcal{N}_{i+1}'$ to be the tree-child network $\mathcal{N}_i' \backslash \{(u_1', v_1'), (u_2', v_2')\}$. Go to Step 6.

6. Increase i by 1 and go back to Step 3.

Theorem 1 immediately follows from the next theorem.

Theorem 15. *Let $\mathcal{N}$ and $\mathcal{N}'$ be normal and tree-child networks on X, respectively. Then* SameDisplaySet *applied to $\mathcal{N}$ and $\mathcal{N}'$ correctly determines if $T(\mathcal{N}) = T(\mathcal{N}')$. Furthermore,* SameDisplaySet *runs in time quadratic in the size of X.*

Proof. Ignoring the running time, by Lemma 7, we may assume that $\mathcal{N}'$ has no trivial shortcuts. Therefore, as there is exactly one phylogenetic tree for when $|X| = 2$, the fact that SameDisplaySet correctly determines whether or not $T(\mathcal{N}) = T(\mathcal{N}')$ follows by combining Propositions 8, 10, and 11 and Lemmas 12, 13, and 14. Thus to complete the proof of the theorem, it suffices to show that the running time of the algorithm is quadratic in the size of X.

Let $n = |X|$ and note that the total number of vertices in a tree-child network is linear in the size of X (see [13]). Thus both $\mathcal{N}$ and $\mathcal{N}'$ have at most $O(n)$ vertices in total. Now consider SameDisplaySet applied to $\mathcal{N}$ and $\mathcal{N}'$. Step 1 is a preprocessing step that considers, for each reticulation v in $\mathcal{N}'$, whether there is an arc joining the parents of v. Since this takes constant time for each reticulation, this step takes $O(n)$ time to complete. For iteration i, Step 3 finds a cherry or a reticulated cherry in $\mathcal{N}_i$. Since $\mathcal{N}_i$ is normal, one way to do this is to construct a maximal path that starts at the root of $\mathcal{N}_i$ and ends at a tree vertex. The two leaves below this tree vertex, say a and b, either form a cherry or a reticulated cherry in $\mathcal{N}_i$. As the total number of vertices in $\mathcal{N}_i$ is $O(n)$, this takes time $O(n)$. If $\{a, b\}$ is a cherry in $\mathcal{N}_i$, then Step 4 determines whether or not $\{a, b\}$ is a cherry in $\mathcal{N}_i'$ and, if so, deletes b in both $\mathcal{N}_i$ and $\mathcal{N}_i'$ and suppresses any resulting vertex of in-degree one and out-degree one. Therefore Step 4 takes constant time. On the other hand, if $\{a, b\}$ is a reticulated cherry in $\mathcal{N}_i$, then Step 5 is called. Similar to Step 4, this step considers the structure in $\mathcal{N}_i$ and $\mathcal{N}_i'$ local to a and b, but is less straightforward. In terms of running time, the longest part of the step to complete is in determining the cluster and visibility sets of certain vertices. A single postorder transversal of each of $\mathcal{N}_i$ and $\mathcal{N}_i'$ can be used to determine all cluster sets of $\mathcal{N}_i$ and $\mathcal{N}_i'$. Since $\mathcal{N}$ and $\mathcal{N}'$ are both binary, the number of arcs in each is $O(n)$, so this takes time $O(n)$. Furthermore, to determine the visibility set of a vertex u of $\mathcal{N}_i$, we delete u and its incident arcs, and check, for each leaf ℓ, whether the resulting rooted acyclic directed graph, D_i say, has a path from the root to ℓ. That is, loosely speaking, we want to find the 'cluster set', X'

say, of the root in D_i. It then follows that the visibility set of u is $X_i - X'$, where X_i is the leaf set of $\mathcal{N}_i$. A single postorder transversal of D_i is sufficient to determine X', so this takes time $O(n)$. Similarly, the visibility set of a vertex in $\mathcal{N}_i'$ can be found in this way. As we only need to find the visibility sets of three vertices in $\mathcal{N}_i$ and $\mathcal{N}_i'$, the total time to determine the necessary visibility sets is $O(n)$. Thus the time to complete Step 5, including the deletion of certain arcs, is $O(n)$. Hence, each iteration of SameDisplaySet takes $O(n)$. Since each iteration deletes at least one vertex or arc in each of $\mathcal{N}$ and $\mathcal{N}'$, it follows that there are $O(n)$ iterations, and so the entire algorithm runs in time $O(n^2)$. □

Algorithm returns *No*

Suppose that SameDisplaySet is applied to normal and tree-child networks $\mathcal{N}$ and $\mathcal{N}'$ on X, respectively, and returns *No*. In this case, it is natural to ask for a phylogenetic X-tree displayed by exactly one of $\mathcal{N}$ and $\mathcal{N}'$. Without going into detail, it is straightforward to amend the algorithm so that such a tree is constructed. To illustrate, assume that SameDisplaySet returns *No* at Step 5(a)(i). Then, at some iteration i, the normal network $\mathcal{N}_i$ has a reticulated cherry $\{a, b\}$, in which the parent of b is a reticulation, and the parent of b in the tree-child network $\mathcal{N}_i'$ is a reticulation, but the structure of $\mathcal{N}_i'$ local to a and b is not as that shown in Fig. 4. Comparing this figure with Fig. 3, this implies that, while the display set of $\mathcal{N}_i$ contains a tree with cherry $\{a, b\}$, a tree with generalised cherry $\{b, V_q\}$, and a tree with generalised cherry $\{b, C_q - b\}$, the display set of $\mathcal{N}_i'$ does not contain a tree of one of these three types. By choosing an embedding in $\mathcal{N}_i$ of such a tree in the display set of $\mathcal{N}_i$ and then reversing the steps in the algorithm that have been performed up to Step 5(a)(i) in iteration i, we can construct a subdivision of a tree displayed by $\mathcal{N}$ but not displayed by $\mathcal{N}'$.

If there exists a phylogenetic tree $\mathcal{T}$ that is displayed by $\mathcal{N}$ and not displayed by $\mathcal{N}'$, then it may be possible for practitioners, who compare $\mathcal{N}$ and $\mathcal{N}'$ with a biological question in mind, to interpret the presence and absence of $\mathcal{T}$ in the display set of $\mathcal{N}$ and $\mathcal{N}'$, respectively, in a biologically meaningful way. For example, if $\mathcal{T}$ is know to be a gene tree that is associated with a DNA segment used to reconstruct $\mathcal{N}$ and $\mathcal{N}'$, then the fact that $\mathcal{T}$ is not displayed by $\mathcal{N}'$ may indicate that $\mathcal{N}$ more faithfully represents the evolutionary history of the species under consideration than $\mathcal{N}'$.

Acknowledgments. We thank the two anonymous referees for their constructive comments.

References

[1] M. Bordewich and C. Semple. Reticulation-visible networks. *Adv. Appl. Math.*, 76:114–141, 2016.

[2] M. Bordewich and C. Semple. Determining phylogenetic networks from inter-taxa distances. *J. Math. Bio.*, 73:283–303, 2016.

[3] G. Cardona, F. Rosselló, and G. Valiente. Comparison of tree-child phylogenetic networks. *IEEE/ACM Trans. Comput. Biol. Bioinf.*, 6:552–569, 2009.

[4] P. Cordue, S. Linz, and C. Semple. Phylogenetic networks that display a tree twice. *B. Math. Biol.*, 76:2664–2679, 2014.

[5] J. Döcker, S. Linz, and C. Semple. Displaying trees across two phylogenetic networks. *Theor. Comput. Sci.*, 796:129–146, 2019.

[6] P. Gambette and K. T. Huber. On encodings of phylogenetic networks of bounded level. *J. Math. Bio.*, 65:157–180, 2012.

[7] A. D. M. Gunawan. Solving the Tree Containment problem for reticulation-visible networks in linear time. In *Algorithms for Computational Biology*, pages 24–36. Springer, 2018.

[8] A. D. M. Gunawan, B. DasGupta, and L. Zhang. A decomposition theorem and two algorithms for reticulation-visible networks. *Inform. Comput.*, 252:161–175, 2017.

[9] L. van Iersel and V. Moulton. Trinets encode tree-child and level-2 phylogenetic networks. *J. Math. Bio.*, 68:1707–1729, 2014.

[10] L. van Iersel, C. Semple, and M. Steel. Locating a tree in a phylogenetic network. *Inform. Process. Lett.*, 110:1037–1043, 2010.

[11] I. A. Kanj, L. Nakhleh, C. Than, and G. Xia. Seeing the trees and their branches in the network is hard. *Theor. Comput. Sci.*, 401:153–164, 2008.

[12] S. Linz, K. St John, and C. Semple. Counting trees in a phylogenetic network is #P-complete. *SIAM J. Comput.*, 42:1768–1776, 2013.

[13] C. McDiarmid, C. Semple, and D. Welsh. Counting phylogenetic networks. *Ann. Comb.*, 19:205–224, 2015.

[14] N. F. Müller, U. Stolz, G. Dudas, T. Stadler, and T. G. Vaughan. Bayesian inference of reassortment networks reveals fitness benefits of reassortment in human influenza viruses. *P. Natl. Acad. Sci. USA*, 117:17104–17111, 2020.

[15] L. Nakhleh, G. Jin, F. Zhao, and J. Mellon-Crummey. Reconstructing phylogenetic networks using maximum parsimony. In *Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conference*, pages 93–102. IEEE, 2005.

[16] C. Semple. Phylogenetic networks with every embedded phylogenetic tree a base tree. *B. Math. Biol.*, 78:32–137, 2016.

[17] C. Solís-Lemus, P. Bastide, and C. Ané. PhyloNetworks: a package for phylogenetic networks. *Mol. Biol. Evol.*, 34:3292–3298, 2017.

[18] L. J. Stockmeyer. The polynomial-time hierarchy. *Theor. Comput. Sci.*, 3:1–22, 1978.

[19] S. J. Willson. Properties of normal phylogenetic networks. *B. Math. Biol.*, 72:340-358, 2010.

[20] S. J. Willson. Regular networks can be uniquely constructed from their trees. *IEEE/ACM Trans. Comput. Biol. Bioinf.*, 8:785–796, 2011.

[21] S. J. Willson. Tree-average distances on certain phylogenetic networks have their weights uniquely determined. *Algorithm. Mol. Biol.*, 7:13, 2012.

[22] J. Zhu, D. Wen, Y. Yu, H. M. Meudt, and L. Nakhleh. Bayesian inference of phylogenetic networks from bi-allelic genetic markers. *PLoS Comput. Biol.*, 14:e1005932, 2018.