Journal of Mathematical Biology manuscript No.
(will be inserted by the editor)

Caterpillars on three and four leaves are sufficient to reconstruct binary normal networks

Simone Linz · Charles Semple

July 15, 2020

Abstract While every rooted binary phylogenetic tree is determined by its set of displayed rooted triples, such a result does not hold for an arbitrary rooted binary phylogenetic network. In particular, there exist two non-isomorphic rooted binary temporal normal networks that display the same set of rooted triples. Moreover, without any structural constraint on the rooted phylogenetic networks under consideration, similarly negative results have also been established for binets and trinets which are rooted subnetworks on two and three leaves, respectively. Hence, in general, piecing together a rooted phylogenetic network from such a set of small building blocks appears insurmountable. In contrast to these results, in this paper, we show that a rooted binary normal network is determined by its sets of displayed caterpillars (particular type of subtrees) on three and four leaves. The proof is constructive and realises a polynomial-time algorithm that takes the sets of caterpillars on three and four leaves displayed by a rooted binary normal network and, up to isomorphism, reconstructs this network.

Keywords Normal networks · rooted triples · quads

Mathematics Subject Classification (2010) 05C85 · 68R10

1 Introduction

Rooted phylogenetic networks are a generalisation of rooted phylogenetic trees that allow for the representation of non-treelike processes such as hybridisa-

The authors were supported by the New Zealand Marsden Fund.

Simone Linz
School of Computer Science, University of Auckland, Auckland, New Zealand
E-mail: s.linz@auckland.ac.nz

Charles Semple (Corresponding Author)
School of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand
E-mail: charles.semple@canterbury.ac.nz

tion and lateral gene transfer. Recently many structural properties of rooted phylogenetic networks have been described that have led to a classification of such networks into several well-studied network classes (e.g. normal [25], tree-child [7], and tree-based [9]). Furthermore, the development of algorithms to reconstruct rooted phylogenetic networks from smaller building blocks remains an active area of research for the last fifteen years. While historically rooted triples were most prominent as building blocks (see, for example, [11, 15,18,19]), the focus has become broader and now also includes rooted phylogenetic trees and clusters. In turn, this has led to many algorithmic advances (for a recent overview, see [6, Section 5.2.3] and references therein). However, when reconstructing a rooted phylogenetic network from a given set of building blocks, most approaches infer new building blocks, i.e. the resulting network contains rooted triples, trees, or clusters that are not part of the input. Indeed, very little is known about when a set of building blocks determines a rooted phylogenetic network. This question motivates the work presented in this paper.

In the context of reconstructing rooted phylogenetic trees, supertree methods collectively provide fundamental tools for reconstructing and analysing rooted phylogenetic trees. In general, these classical and commonly-used methods take as input a collection of smaller rooted phylogenetic trees on overlapping leaf sets and output a parent tree (supertree) that 'best' represents the entire input collection. For practical reasons, most of these methods do not require the input collection to be consistent. Nevertheless, the property typically underlying any supertree method for reconstructing rooted phylogenetic trees is the following well-known theorem (see, for example, [1,23]).

Theorem 1 *Let $\mathcal{R}$ be the set of rooted triples displayed by a rooted binary phylogenetic X-tree $\mathcal{T}$. Then, up to isomorphism,*

(i) *$\mathcal{T}$ is the unique rooted binary phylogenetic X-tree whose set of displayed rooted triples is $\mathcal{R}$, and*
(ii) *$\mathcal{T}$ can be reconstructed from $\mathcal{R}$ in polynomial time.*

For an excellent review of supertree methodology, see [3]. As an initial step towards developing supertree-type methods for reconstructing and analysing rooted phylogenetic networks, we would like analogues of Theorem 1 for rooted phylogenetic networks.

Gambette and Huber [10] established that rooted binary level-one networks, that is, rooted binary phylogenetic networks whose underlying cycles are vertex disjoint, are determined by their sets of displayed rooted triples provided each underlying cycle has length at least five. However, there exist two non-isomorphic rooted binary level-two networks that have the same set of displayed rooted triples [10, Fig. 11]. This begs the question whether or not displayed subtrees on more than three leaves are sufficient to determine rooted phylogenetic networks in general. While Willson [24] has shown that rooted binary regular networks, which include the class of rooted binary normal networks, on n leaves can be determined and reconstructed (in polynomial time)

from their sets of displayed rooted phylogenetic trees on n leaves, arbitrary rooted binary phylogenetic networks cannot be determined in this way, even if branch lengths are considered [22, Fig. 3]. As an interesting aside, Francis and Moulton [8, Theorem 3.5] have shown that rooted binary tree-child networks are determined by their sets of embedded spanning trees which, importantly, are not necessarily rooted phylogenetic trees.

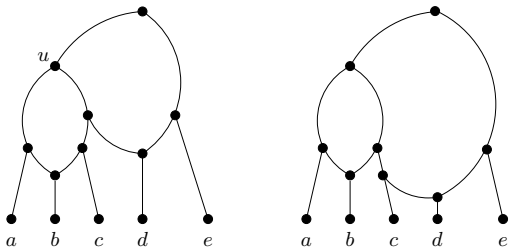
Partly due to the aforementioned negative deterministic results, recent studies have investigated whether or not rooted phylogenetic networks are determined by their embedded subnetworks like binets [12,17] and trinets [12,14, 16], that is, rooted phylogenetic networks on two and three leaves, respectively. It has been shown that trinets determine rooted binary level-two and rooted binary tree-child networks [16]. Furthermore, binets determine the number of vertices in a rooted phylogenetic network whose in-degree is at least two but do not contain enough information to determine the structural properties of a rooted phylogenetic network even for restricted network classes [17]. Lastly, for an arbitrary rooted binary phylogenetic network $\mathcal{N}$ on n leaves, Huber et al. [13] have considered larger subnetworks and shown that, even if for each $n' \in \{1, 2, \ldots, n-1\}$ all embedded subnetworks of $\mathcal{N}$ on n' leaves are given, $\mathcal{N}$ cannot necessarily be determined by the resulting set of subnetworks. On a positive note, a binary level-k tree-child network $\mathcal{N}$, where $k \geq 2$, can be determined and reconstructed in polynomial time from a collection of subnetworks that are obtained from $\mathcal{N}$ by deleting one arc that is directed into a reticulation (formally defined below) from each biconnected component with at most k reticulations [21].

In this paper, we return to the simpler tree-like building blocks of small size for reconstructing rooted phylogenetic networks. In particular, the main result of the paper establishes that rooted binary normal networks are determined by their sets of caterpillars (particular type of subtrees) on three and four leaves and that they can be reconstructed from these sets in polynomial time.

To formally state the main result, we need some notation and terminology. Throughout the paper, X will always denote a non-empty finite set. A *rooted binary phylogenetic network $\mathcal{N}$ on X* is a rooted acyclic directed graph with no parallel arcs satisfying the following three properties:

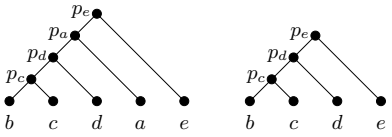
(i) the (unique) root has in-degree zero and out-degree two;
(ii) a vertex of out-degree zero has in-degree one, and the set of vertices with out-degree zero is X; and
(iii) all other vertices either have in-degree one and out-degree two, or in-degree two and out-degree one.

For technical reasons, if $|X| = 1$, then we additionally allow $\mathcal{N}$ to consist of the single vertex in X. The vertices of $\mathcal{N}$ 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*. Arcs directed into a reticulation are called *reticulation arcs*, all other arcs are *tree arcs*. A *rooted binary phylogenetic X-tree* is a rooted binary phylogenetic network on X with no reticulations. To ease reading, for



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

Fig. 1 Two normal networks on $\{a, b, c, d, e\}$ with the same set $\mathcal{R}$ of displayed triples, where $\mathcal{R} = \{ab|c, ab|d, ab|e, ac|d, ac|e, ad|e, bc|a, bc|d, bc|e, bd|a, bd|e, cd|a, cd|b, cd|e, de|a, de|b, de|c\}$. Observe that the quad (b, c, d, e) is displayed by $\mathcal{N}$, but it is not displayed by $\mathcal{N}'$.



(i) $\mathcal{T}$ (ii) $\mathcal{T}'$

Fig. 2 Two caterpillars $\mathcal{T} = (b, c, d, a, e)$ and $\mathcal{T}' = (b, c, d, e)$, where p_a, p_c, p_d, and p_e are the parents of a, c, d, and e, respectively.

the rest of the paper, we will refer to rooted binary phylogenetic networks and rooted binary phylogenetic trees as phylogenetic networks and phylogenetic trees, respectively, as all such networks and trees are rooted and binary.

Let $\mathcal{N}_1$ and $\mathcal{N}_2$ be two phylogenetic networks on X with vertex and arc sets V_1 and E_1, and V_2 and E_2, respectively. We say $\mathcal{N}_1$ is *isomorphic* to $\mathcal{N}_2$ if there is a bijection $\varphi : V_1 \to V_2$ such that $\varphi(x) = x$ for all $x \in X$, and $(u, v) \in E_1$ if and only if $(\varphi(u), \varphi(v)) \in E_2$ for all $u, v \in V_1$.

Let $\mathcal{N}$ be a phylogenetic network on X. A reticulation arc (u, v) of $\mathcal{N}$ is a *shortcut* if $\mathcal{N}$ has a directed path from u to v avoiding (u, v). We say $\mathcal{N}$ is *tree-child* if every non-leaf vertex is the parent of a tree vertex or a leaf. Moreover, $\mathcal{N}$ is *normal* if it is tree-child and has no shortcuts. An example of two normal networks is shown in Fig. 1, where, as with all figures in this paper, arcs are directed down the page. Additionally, the two phylogenetic networks that are obtained from the middle and the right-hand side networks of Fig. 3 by ignoring the dashed arcs are tree child, but not normal.

Let $\mathcal{T}$ be a phylogenetic X-tree with X. For each $x \in X$, let p_x be the parent of x in $\mathcal{T}$. We call $\mathcal{T}$ a *caterpillar* if we can order its leaf set X, say $x_1, x_2, \ldots, x_n$, such that, $p_1 = p_2$ and, for all $i \in \{2, 3, \ldots, n-1\}$ we have that (p_{i+1}, p_i) is an arc in $\mathcal{T}$. Furthermore, we denote such a caterpillar $\mathcal{T}$ by $(x_1, x_2, x_3, \ldots, x_n)$ or, equivalently, $(x_2, x_1, x_3, \ldots, x_n)$ where x_1 and x_2 have been interchanged. As an example, the two phylogenetic trees $\mathcal{T}$ and $\mathcal{T}'$ in Fig. 2 are caterpillars on five and four leaves, respectively. Here, $\mathcal{T}$ is denoted by (b, c, d, a, e) and $\mathcal{T}'$ is denoted by (b, c, d, e). For a caterpillar $\mathcal{T}$ on X, we say

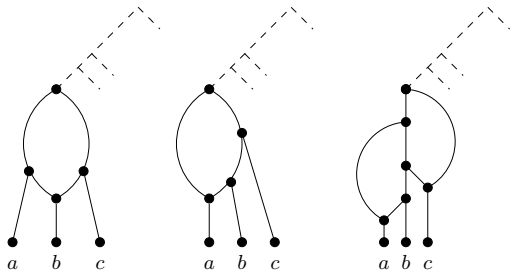


Fig. 3 Three tree-child networks on $\{a, b, c\}$ (solid arcs) that display the same set of triples. By adding additional leaves, say $x_1, x_2, \ldots, x_n$, as indicated by the dashed arcs, the three networks can be extended to tree-child networks of arbitrary size that display the same set of triples and phylogenetic trees on $\{a, b, c\} \cup \{x_1, x_2, \ldots, x_n\}$.

that $\mathcal{T}$ is a *triple* if $|X| = 3$ and we say that $\mathcal{T}$ is a *quad* if $|X| = 4$. While we will denote quads as 4-tuples, we will denote the triple (x_1, x_2, x_3) by $x_1x_2|x_3$ in keeping with standard notation (e.g. see [23]). Note that triples are also referred to as triplets, rooted triples, and rooted triplets in the literature.

Now let $\mathcal{N}$ be a phylogenetic network on X, and let $\mathcal{T}$ be a phylogenetic X'-tree, where X' is a non-empty subset of X. Then $\mathcal{N}$ *displays* $\mathcal{T}$ if $\mathcal{T}$ can be obtained from $\mathcal{N}$ by deleting arcs and vertices, and suppressing any resulting vertices of in-degree one and out-degree one. To illustrate, consider Fig. 1. The caterpillar (b, c, d, a, e) is displayed by $\mathcal{N}$, but the caterpillar (a, c, d, e) is not displayed by $\mathcal{N}$. Hence, (a, c, d, e) is not an element of the set of quads that are displayed by $\mathcal{N}$. The main result of this paper is the following theorem.

Theorem 2 *Let $\mathcal{R}$ and $\mathcal{Q}$ be the sets of triples and quads displayed by a normal network $\mathcal{N}$ on X, respectively. Then, up to isomorphism,*

(i) *$\mathcal{N}$ is the unique normal network on X whose sets of displayed triples and quads are $\mathcal{R}$ and $\mathcal{Q}$, and*
(ii) *$\mathcal{N}$ can be reconstructed from $\mathcal{R}$ and $\mathcal{Q}$ in $O(|X|^6|\mathcal{R}||\mathcal{Q}|)$ time, that is, in $O(|X|^{13})$ time.*

It is natural to ask whether Theorem 2(i) can be strengthened. In particular, are (a) normal networks determined by their displayed triples and are (b) tree-child networks determined by their displayed triples and quads? For both (a) and (b), the answer, in general, is no. To see this, first consider the two normal networks $\mathcal{N}$ and $\mathcal{N}'$ on $\{a, b, c, d, e\}$ shown in Fig. 1. Here, both networks display the same set of triples but are not isomorphic. Hence, by Theorem 2(i), $\mathcal{N}$ and $\mathcal{N}'$ do not display the same set of quads. Indeed, (b, c, d, e) is only contained in the set of quads of $\mathcal{N}$. For (b), consider the three tree-child networks shown in Fig 3. All three networks display the same sets of triples and quads, but no two networks are isomorphic. In fact, all three networks display the same set of phylogenetic X-trees, where X is the leaf set of each of the three tree-child networks.

The paper is organised as follows. The next section contains some preliminaries that are used throughout the paper. The proof of Theorem 2 relies on being able to recognise so-called cherries and reticulated cherries, certain structures involving two leaves, using only triples and quads. This recognition is established in Sections 3 and 4. The proof of Theorem 2 as well as the associated algorithm for reconstructing a normal network from its triples and quads are given in Section 5. In the last section, we consider the class of temporal normal networks, and highlight how the approach taken in Theorem 2 can be simplified for reconstructing such networks from their sets of triples and quads.

Lastly, normal networks are a rich class of phylogenetic networks. Thus, given the negative results mentioned earlier in the introduction, it is a little surprising that they are determined by their sets of triples and quads as Theorem 2 establishes. Nevertheless, knowing that they are determined by these two sets provides impetus for developing a supertree-type method for constructing normal networks. To this end, an intermediate step is to develop an algorithm for deciding if, given an arbitrary set of triples and quads on overlapping leaf sets, there is a normal network that displays each caterpillar in the given set.

2 Preliminaries

In this section, we state some further notation and terminology used in the paper. We begin by noting that an immediate consequence of the definition of a tree-child network is that a phylogenetic network $\mathcal{N}$ is tree-child if and only if, for every vertex u of $\mathcal{N}$, there is a directed path from u to a leaf, ℓ say, in which each vertex, except ℓ and possibly u, is a tree vertex. In particular, every vertex in a normal network has this property. We refer to such a path as a *tree path (for u)*. Note that if u is a leaf, then the path consisting of just u is a tree path for u.

2.1 Embeddings

Let $\mathcal{N}$ be a phylogenetic network on X, and let $\mathcal{T}$ be a phylogenetic X'-tree, where $X' \subseteq X$. An equivalent and convenient way to view the notion of display 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 seen that $\mathcal{N}$ displays $\mathcal{T}$ if and only if a subdivision, $\mathcal{S}$ say, of either $\mathcal{T}$ or the root extension of $\mathcal{T}$ can be obtained from $\mathcal{N}$ by deleting arcs and non-root vertices, in which case, the roots of $\mathcal{S}$ and $\mathcal{N}$ coincide. This equivalence is freely used throughout the paper. We refer to $\mathcal{S}$ as an *embedding* of $\mathcal{T}$ in $\mathcal{N}$ and, for convenience, sometimes view $\mathcal{S}$ as the arc set of $\mathcal{S}$. To illustrate, observe that the triple $bc|a$ is displayed by the phylogenetic network $\mathcal{N}$ shown in Fig. 1(i) because a subdivision of the root extension of this triple can be obtained from $\mathcal{N}$ by deleting arcs and non-root vertices of $\mathcal{N}$.

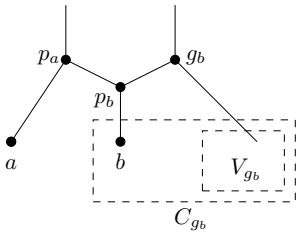


Fig. 4 A reticulated cherry $\{a, b\}$ in which b is the reticulation leaf. The dashed boxes labelled V_{g_b} and C_{g_b} indicate the visibility and cluster set of g_b, respectively. Note that $b \notin V_{g_b}$ and that $V_{g_b} \subset C_{g_b}$.

Let $\mathcal{S}$ be an embedding of a phylogenetic tree $\mathcal{T}$ in $\mathcal{N}$, and let (u, v) be an arc in $\mathcal{N}$. If (u, v) is an arc in $\mathcal{S}$, then $\mathcal{S}$ *uses* (u, v); otherwise, it *avoids* (u, v). Analogous terminology holds for the vertices in $\mathcal{N}$.

2.2 Cherries and reticulated cherries

For each leaf x of a phylogenetic network, we denote the parent of x by p_x. Now let a, b, and c be distinct leaves of a phylogenetic network $\mathcal{N}$. If $p_a = p_b$, then $\{a, b\}$ is a *cherry* of $\mathcal{N}$. Furthermore, if p_b is a reticulation and (p_a, p_b) is an arc, then $\{a, b\}$ is a *reticulated cherry* of $\mathcal{N}$ in which b is the *reticulation leaf*. Moreover, in this case, we denote the parent of p_b that is not p_a by g_b. As an example, in Fig. 1(i), $\{b, c\}$ is a reticulated cherry in which b is the reticulation leaf in $\mathcal{N}$. However, in Fig. 1(ii), $\{b, c\}$ is not a reticulated cherry in $\mathcal{N}'$. A generic reticulated cherry $\{a, b\}$ in which b is the reticulation leaf and with vertices p_a, p_b, and g_b is shown in Fig. 4.

The next lemma is well known for tree-child networks (for example, see [5]). The restriction to normal networks is immediate. We will use it freely throughout the paper.

Lemma 1 *Let $\mathcal{N}$ be a normal network on X, where $|X| \geq 2$. Then $\mathcal{N}$ has either a cherry or a reticulated cherry.*

2.3 Cluster and visibility sets

Let $\mathcal{N}$ be a phylogenetic network on X, and let u be a vertex of $\mathcal{N}$. The *cluster set* of u, denoted C_u, is the subset of X consisting exactly of each leaf ℓ in X for which there is a directed path from u to ℓ. Furthermore, the *visibility set* of u, denoted V_u, is the subset of X consisting exactly of each leaf ℓ in X for which every directed path from the root of $\mathcal{N}$ to ℓ traverses u. Observe that $V_u \subseteq C_u$ and that, if there is a tree path from u to a leaf ℓ, then $\ell \in V_u$. However, the converse is not necessarily true, i.e. if $\ell' \in V_u$, then there may or may not be a tree path from u to ℓ'. Furthermore, if $\mathcal{N}$ is normal, then

V_u, and thus C_u, is non-empty. To illustrate, consider the vertex u in Fig 1(i). The cluster and visibility sets of u are $C_u = \{a, b, c, d\}$ and $V_u = \{a, b, c\}$, respectively. Note that $d \notin V_u$ as there is a directed path from the root of $\mathcal{N}$ to d avoiding u. More generically, the cluster and visibility set of a particular vertex in a reticulated cherry is shown in Fig. 4.

3 Recognising cherries

The key idea in the proof of Theorem 2 is recognising cherries and reticulated cherries in a normal network $\mathcal{N}$ using only the triples and quads displayed by $\mathcal{N}$. In this section, we establish the lemmas for doing this. We begin by recognising cherries.

Lemma 2 *Let $\mathcal{N}$ be a normal network on X, where $|X| \geq 3$, and let $\mathcal{R}$ be the set of triples displayed by $\mathcal{N}$. Let $\{a, b\} \subseteq X$. Then $\{a, b\}$ is a cherry of $\mathcal{N}$ if and only if $\{a, b\}$ satisfies the following property: if $xy|z \in \mathcal{R}$ and $\{a, b\} \subseteq \{x, y, z\}$, then $\{a, b\} = \{x, y\}$.*

Proof If $\{a, b\}$ is a cherry of $\mathcal{N}$, then it is easily checked that $\{a, b\}$ satisfies the property in the statement of the lemma. Now suppose that $\{a, b\}$ satisfies this property. First assume p_a is a reticulation. Let u and u' be the two parents of p_a. Since $\mathcal{N}$ is normal, there are two distinct elements, say ℓ and ℓ', in $X - \{a\}$ such that ℓ is at the end of a tree path for u and ℓ' is at the end of a tree path for u'. If $\ell \neq b$, then either $a\ell|b \in \mathcal{R}$ or $b\ell|a \in \mathcal{R}$, a contradiction. So $\ell = b$, but then $a\ell'|b \in \mathcal{R}$, another contradiction. Thus p_a is a tree vertex or the root of $\mathcal{N}$. If p_a is the root of $\mathcal{N}$, then, as $|X| \geq 3$, there exists a triple $b\ell|a \in \mathcal{R}$, where $\ell \in X - \{a, b\}$, a contradiction. So p_a is a tree vertex.

Let v be the child of p_a that is not a. If $b \notin C_v$, then $a\ell|b \in \mathcal{R}$, where $\ell \in C_v$, a contradiction. Therefore $b \in C_v$. If $|C_v| > 1$, then there is an element $\ell \in C_v - \{b\}$ such that $b\ell|a \in \mathcal{R}$. This last contradiction implies that $C_v = \{b\}$, and so $\{a, b\}$ is either a cherry or a reticulated cherry of $\mathcal{N}$ with reticulation leaf b. If the latter, then $p_b = v$ is a reticulation. In this case, let g_b be the parent of p_b that is not p_a, and let ℓ' be a leaf at the end of a tree path for g_b. If $\ell' = a$, then there is a directed path from g_b to v that traverses p_a and, so, (g_b, v) is a shortcut; a contradiction since $\mathcal{N}$ is normal. Hence, we have $\ell' \neq a$ and, therefore, $b\ell'|a \in \mathcal{R}$. This last contradiction implies that $\{a, b\}$ is a cherry, thereby completing the proof of the lemma. □

We next consider the recognition of reticulated cherries. For the purposes of establishing Theorem 2, in addition to recognising a reticulated cherry, say $\{a, b\}$ in which b is the reticulation leaf, we also want to determine the visibility set of g_b. To this end, we next introduce the notion of a candidate set.

Let $\mathcal{N}$ be a phylogenetic network on X, and let $\mathcal{R}$ and $\mathcal{Q}$ be the sets of triples and quads displayed by $\mathcal{N}$, respectively. Let $\{a, b\} \subseteq X$. A *candidate set for b* is a non-empty subset, W_b say, of $X - \{a, b\}$ satisfying the following properties:

(I) For all $c \in W_b$ and all $x \in X - (W_b \cup \{b\})$, the triple $bc|x \in \mathcal{R}$, but the triple $ac|b \notin \mathcal{R}$.
(II) For all distinct $c, c' \in W_b$, the triple $bc|c' \notin \mathcal{R}$.
(III) For all $c \in W_b$, there is no $x \in X-(W_b \cup \{a,b\})$ such that $(x,b,c,a) \in \mathcal{Q}$.

If $\{a,b\}$ is indeed a reticulated cherry with reticulation leaf b of a normal network, then W_b is intended as a candidate for the visibility set of g_b. To illustrate the definition of a candidate set, consider the network $\mathcal{N}'$ that is shown in Fig. 1(ii), where $X = \{a,b,c,d,e\}$. Let $\mathcal{R}$ and $\mathcal{Q}$ be the sets of triples and quads displayed by $\mathcal{N}'$, and let $\{a,b\}$ be a 2-element subset of X. Then a candidate set for b is $W_b = \{c\}$. To verify that W_b is indeed a candidate set for b, observe that $bc|a$, $bc|d$, and $bc|e$ are elements in $\mathcal{R}$, and that $ac|b \notin \mathcal{R}$. Hence, W_b satisfies (I). Moreover, since $|W_b| = 1$, (II) vacuously holds. Lastly, (III) is satisfied because the quads (d,b,c,a) and (e,b,c,a) are not elements in $\mathcal{Q}$. Note that $\{c,d\}$ is not a candidate set for b as it does not satisfy (II) since $bc|d \in \mathcal{R}$.

Lemma 3 *Let $\mathcal{N}$ be a normal network on X, where $|X| \geq 3$, and let $\mathcal{R}$ and $\mathcal{Q}$ be the sets of triples and quads displayed by $\mathcal{N}$, respectively. Let $\{a,b\} \subseteq X$, and let W_b be a candidate set for b. Then $\{a,b\}$ is a reticulated cherry of $\mathcal{N}$ in which b is the reticulation leaf and W_b is the visibility set of g_b if and only if a, b, and W_b satisfy the following properties:*

(i) *For all $x \in X - \{a,b\}$, the triple $ab|x \in \mathcal{R}$.*
(ii) *For all $c \in W_b$, there is no $x \in X - (\{a,b\} \cup W_b)$ such that (x,b,a,c) or (x,a,b,c) is in $\mathcal{Q}$.*
(iii) *If there exists an $x \in X-(\{a,b\}\cup W_b)$ such that $ac|x \in \mathcal{R}$, where $c \in W_b$, then (a,b,c,x) and (c,b,a,x) are in $\mathcal{Q}$.*

Proof Suppose that $\{a,b\}$ is a reticulated cherry of $\mathcal{N}$ in which b is the reticulation leaf and W_b is the visibility set of g_b. It is easily seen that a, b, and $W_b = V_{g_b}$ satisfy (i) and (ii). To see that a, b, and V_{g_b} satisfy (iii), assume that there is an $x \in X - (\{a,b\} \cup V_{g_b})$ such that $ac|x \in \mathcal{R}$. Let $\mathcal{S}$ be an embedding of $ac|x$ in $\mathcal{N}$. Then $\mathcal{S}$ uses (p_a, a) as well as the arc directed into g_b and the arc directed out of g_b that is not (g_b, p_b). Hence, by adjoining the arcs (p_a, p_b) and (p_b, b) to $\mathcal{S}$, we construct an embedding of (a,b,c,x) in $\mathcal{N}$ and, by adjoining the arcs (g_b, p_b) and (p_b, b) to $\mathcal{S}$, we construct an embedding of (c,b,a,x) in $\mathcal{N}$. Thus $(a,b,c,x), (c,b,a,x) \in \mathcal{Q}$.

For the converse, suppose that a, b, and W_b satisfy (i), (ii), and (iii). We first establish that p_b is a reticulation. Assume that p_b is either a tree vertex or the root of $\mathcal{N}$. Let v denote the child of p_b that is not b, and let ℓ be the leaf at the end of a tree path for v.

3.1 *The vertex v is a reticulation.*

Proof If $v = \ell$, then, as $|X| \geq 3$, it follows by (i) that $\ell = a$; otherwise, $ab|\ell \notin \mathcal{R}$. But then $bc|a \notin \mathcal{R}$ for any $c \in W_b$, contradicting (I). Thus $v \neq \ell$. Moreover, if v is a tree vertex, then, as $ac|b \notin \mathcal{R}$ for each $c \in W_b$ by (I), at

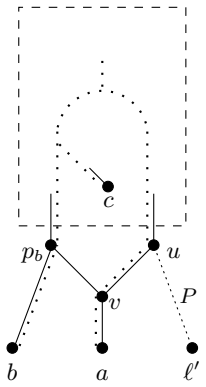


Fig. 5 Setting as described in the proof of (3.3) under the assumptions that p_b is a tree vertex and $C_v = \{a\}$, where the dotted arc joining u and ℓ' represents a tree path from u to ℓ', and bold dotted lines represent an embedding $\mathcal{S}$ of $bc|a$ in $\mathcal{N}$.

most one of a and c is an element of C_v. If $a \notin C_v$, then, it is easily checked that $ab|\ell \notin \mathcal{R}$, contradicting (i). Thus $a \in C_v$ and so $c \notin C_v$ for each $c \in W_b$. But then $bc|\ell \notin \mathcal{R}$ for any c, contradicting (I) in the choice of W_b. Therefore v is a reticulation. □

Now consider the cluster set C_v of v in which v is a reticulation.

3.2 *Exactly one of $a \in C_v$ and $C_v \cap W_b \neq \emptyset$ holds.*

Proof If $a \in C_v$ and $C_v \cap W_b \neq \emptyset$, then $ac|b \in \mathcal{R}$, where $c \in C_v \cap W_b$, contradicting (I) in the choice of W_b. Furthermore, if $a \notin C_v$ and $C_v \cap W_b = \emptyset$, then we can extend an embedding of the triple $ab|c$ in $\mathcal{N}$, where $c \in W_b$, to an embedding of the caterpillar (ℓ, b, a, c), and so $(\ell, b, a, c) \in \mathcal{Q}$, contradicting (ii). Thus exactly one of $a \in C_v$ and $C_v \cap W_b \neq \emptyset$ holds. □

Let u denote the parent of v that is not p_b. Since $\mathcal{N}$ is normal, (u, v) is not a shortcut and there is a tree path from u to a leaf, ℓ' say. Let P denote the arc set of this tree path. We next show that

3.3 *$a \notin C_v$.*

Proof Suppose that $a \in C_v$. If $|C_v| \geq 2$, then, as $C_v \cap W_b$ is empty, $\mathcal{Q}$ contains a caterpillar of the form (x, a, b, c), where $x \in C_v - \{a\}$ and $c \in W_b$. This contradiction to (ii) implies that $|C_v| = 1$, and so $C_v = \{a\}$. If $W_b \cap C_u \neq \emptyset$, then there is a triple $ac|b \in \mathcal{R}$, where $c \in W_b$, contradicting (I) in the choice of W_b. Therefore $W_b \cap C_u$ is empty. Let $c \in W_b$. Since W_b satisfies (I), $bc|a \in \mathcal{R}$. If $\mathcal{S}$ is an embedding of $bc|a$ in $\mathcal{N}$, then $\mathcal{S}$ uses (u, v). For an illustration of this set-up, see Fig. 5. It is now easily checked that the set of arcs

$$(\mathcal{S} - \{(p_b, b), (u, v)\}) \cup (P \cup \{(p_b, v)\})$$

are the arcs of an embedding of $ac|\ell'$ in $\mathcal{N}$. Since $\ell' \in X - (\{a,b\} \cup W_b)$, it follows from (iii) that $(c,b,a,\ell') \in \mathcal{Q}$. This is not possible as any phylogenetic tree displayed by $\mathcal{N}$ with leaf set $\{a,b,c,\ell'\}$ in which $\{b,c\}$ is a cherry, also has $\{a,\ell'\}$ as a cherry. This contradiction implies that $a \notin C_v$. □

It now follows by (3.2) and (3.3) that $C_v \cap W_b \neq \emptyset$. As W_b satisfies (II), W_b is a subset of the visibility set V_v of v. If $V_v - W_b$ is non-empty, then $bc|x \notin \mathcal{R}$, where $c \in W_b$ and $x \in V_v - W_b$, contradicting (I) in the choice of W_b. Therefore $V_v = W_b$. If $a \in C_u$, then there is a triple $ac|b \in \mathcal{R}$, where $c \in W_b$, contradicting (I) in the choice of W_b. Thus $a \notin C_u$. Let $c \in W_b$. By (i), we have $ab|c \in \mathcal{R}$. If $\mathcal{S}$ is an embedding of $ab|c$ in $\mathcal{N}$, then $\mathcal{S}$ uses (u,v), and so

$$(\mathcal{S} - \{(p_b,b),(u,v)\}) \cup (P \cup \{(p_b,v)\})$$

are the arcs of an embedding of $ac|\ell'$ in $\mathcal{N}$. By (iii), $(a,b,c,\ell') \in \mathcal{Q}$. But this is not possible as any phylogenetic tree in $\mathcal{Q}$ with leaf set $\{a,b,c,\ell'\}$ in which $\{a,b\}$ is a cherry, also has $\{c,\ell'\}$ as a cherry. This last contradiction establishes that

3.4 *p_b is a reticulation.*

Let u_1 and u_2 denote the parents of p_b, and let ℓ_1 and ℓ_2 denote the leaves at the end of tree paths for u_1 and u_2, respectively. Note that, since W_b satisfies (II), if there exists an element c in W_b such that $c \in C_{u_i}$ for some $i \in \{1,2\}$, then $W_b \subseteq V_{u_i}$. In turn, this implies that $W_b = V_{u_i}$; otherwise, $bc|x \notin \mathcal{R}$, where $c \in W_b$ and $x \in V_{u_i} - W_b$, contradicting (I). If $W_b \cap (C_{u_1} \cup C_{u_2})$ is empty, then, by considering an embedding of $bc|a$ in $\mathcal{N}$, where $c \in W_b$, it is easily seen that either $(\ell_1,b,c,a) \in \mathcal{Q}$ or $(\ell_2,b,c,a) \in \mathcal{Q}$, contradicting (III) in the choice of W_b. Without loss of generality we may therefore assume that the visibility set of u_2 is W_b. If $a \notin C_{u_1}$, then, by considering an embedding of $ab|c$ in $\mathcal{N}$ for some $c \in W_b$, we deduce that $(\ell_1,b,a,c) \in \mathcal{Q}$, contradicting (ii). Thus $a \in C_{u_1}$. If $|C_{u_1}| \geq 2$, then $(x,a,b,c) \in \mathcal{Q}$, where $x \in C_{u_1} - \{a\}$ and $c \in W_b$, contradicting (ii). Hence $C_{u_1} = \{a\}$. We conclude that $\{a,b\}$ is a reticulated cherry of $\mathcal{N}$ in which b is the reticulation leaf and W_b is the visibility set of g_b. □

4 Finding a candidate set

The algorithm associated with Theorem 2 involves finding the candidate sets for one of two leaves of a potential reticulated cherry. In this section, we consider how this can be done in polynomial time.

Let $\mathcal{N}$ be a phylogenetic network on X, where $|X| \geq 3$, and let $\mathcal{R}$ and $\mathcal{Q}$ be the sets of triples and quads displayed by $\mathcal{N}$, respectively. Let $\{a,b\} \subseteq X$, and suppose we want to find all candidate sets for b if such a set exists or determine that there are no such sets. Potentially, we may have to consider all subsets of $X - \{a,b\}$. However, if $c \in X - \{a,b\}$, the next lemma shows that a candidate set for b containing c, if it exists, is unique. Thus the task

reduces to finding, for each $c \in X - \{a, b\}$, the candidate set for b containing c or determining that no such set exists.

Lemma 4 *Let $\mathcal{R}$ and $\mathcal{Q}$ be the sets of triples and quads, respectively, displayed by a phylogenetic network $\mathcal{N}$ on X, where $|X| \geq 3$. Let $\{a, b\} \subseteq X$ and let $c \in X - \{a, b\}$. If W_b is a candidate set for b containing c, then it is the unique candidate set for b containing c.*

Proof Let W_b' be a candidate set for b containing c. If $|X|$=3, then $W_b = W_b' = \{c\}$ is clearly the unique candidate set for b. Hence, we may assume that $|X| > 3$. Let $x \in X - \{a, b, c\}$. Then, as W_b' satisfies (I) and (II) in the definition of a candidate set, $x \in W_b'$ if and only if $bc|x \notin \mathcal{R}$. It follows that $W_b = W_b'$, that is, W_b is the unique candidate set for b containing c. □

Called Candidate Set, the following algorithm takes as its input X, $\mathcal{R}$, $\mathcal{Q}$, $\{a, b\}$, and c and either finds a candidate set for b containing c or determines that there is no such set.

1. Set $U = \{x \in X - \{b, c\} : bc|x \in \mathcal{R}\}$.
2. Set $W_b = (X - (U \cup \{b\})) \cup \{c\}$.
3. If a, b, and W_b satisfy their namesakes in (I), (II), and (III), then return W_b.
4. Else, return *no candidate set for b containing c.*

The next lemma establishes the correctness and running time of Candidate Set.

Lemma 5 *Let $\mathcal{N}$ be a phylogenetic network on X, where $|X| \geq 3$, and let $\mathcal{R}$ and $\mathcal{Q}$ be the sets of triples and quads displayed by $\mathcal{N}$, respectively. Let $\{a, b\} \subseteq X$ and $c \in X - \{a, b\}$. Then* Candidate Set *applied to X, $\mathcal{R}$, $\mathcal{Q}$, $\{a, b\}$, and c correctly returns a candidate set for b containing c if it exists or the statement* no candidate set for b containing c *if none exists. Furthermore, this application runs in time $O(|X|^2|\mathcal{R}| + |X|^2|\mathcal{Q}|)$, that is, $O(|X|^6)$.*

Proof If Candidate Set returns a set W_b, then, by Step 3, it is a candidate set for b containing c. Conversely, suppose that W_b' is a candidate set for b containing c. Then, by Lemma 4, W_b' is the unique such set and so, by construction, at the end of Step 2, Candidate Set constructs W_b'. It follows that Candidate Set correctly returns W_b'.

For the running time, Steps 1 and 2 take $O(|X||\mathcal{R}|)$ and $O(1)$ time, respectively, while Step 3 takes $O(|X|^2|\mathcal{R}| + |X|^2|\mathcal{R}| + |X|^2|\mathcal{Q}|)$ time. Thus Candidate Set completes in $O(|X|^2|\mathcal{R}| + |X|^2|\mathcal{Q}|)$ time, that is, in $O(|X|^6)$ time as $|\mathcal{R}| \leq |X|^3$ and $|\mathcal{Q}| \leq |X|^4$. □

5 Proof of Theorem 2

In this section, we prove Theorem 2. The first subsection establishes the uniqueness part of this theorem while the second subsection establishes an algorithm that, given the sets of displayed triples and quads of a normal network $\mathcal{N}$, reconstructs $\mathcal{N}$ in time polynomial in the size of the leaf set of $\mathcal{N}$.

5.1 Proof of Theorem 2(i)

We start with two lemmas and the description of an operation on networks that underlies the induction in the proof of this theorem. Let $\mathcal{N}$ be a phylogenetic network on X, and let $\{a, b\}$ be a subset of X. Suppose that $\{a, b\}$ is either a cherry or a reticulated cherry in which b is the reticulation leaf. If $\{a, b\}$ is a cherry, then *deleting* b is the operation of deleting b and its incident arc, and suppressing p_a while, if $\{a, b\}$ is a reticulated cherry, then *deleting* b is the operation of deleting b, p_b, and their incident arcs, and suppressing p_a and g_b. Note that the operation of deleting b in a network with reticulated cherry $\{a, b\}$ in which b is the reticulation leaf can also be viewed as deleting (g_b, p_b) and suppressing the resulting two degree-two vertices followed by the deletion of b in the resulting network that has cherry $\{a, b\}$. The next lemma, which we will freely use throughout the rest of the paper, is now an immediate consequence of [4, Lemma 3.2].

Lemma 6 *Let $\mathcal{N}$ be a normal network on X, and let $\{a, b\} \subseteq X$, where $\{a, b\}$ is either a cherry or a reticulated cherry in which b is the reticulation leaf. If $\mathcal{N}'$ is obtained from $\mathcal{N}$ by deleting b, then $\mathcal{N}'$ is a normal network on $X - \{b\}$.*

Let $\mathcal{R}$ and $\mathcal{Q}$ be the sets of triples and quads displayed by a phylogenetic network $\mathcal{N}$ on X, respectively. Let $b \in X$, and let

$$\mathcal{R}' = \{xy|z \in \mathcal{R} : b \notin \{x, y, z\}\}$$

and

$$\mathcal{Q}' = \{(w, x, y, z) \in \mathcal{Q} : b \notin \{w, x, y, z\}\}.$$

We say that $\mathcal{R}'$ and $\mathcal{Q}'$ have been obtained from $\mathcal{R}$ and $\mathcal{Q}$, respectively, by *deleting* b. The proof of the next lemma is elementary and omitted.

Lemma 7 *Let $\mathcal{N}$ be a normal network on X, and let $\{a, b\} \subseteq X$, where $\{a, b\}$ is either a cherry or a reticulated cherry in which b is the reticulation leaf. Furthermore, let $\mathcal{N}'$ be the normal network obtained from $\mathcal{N}$ by deleting b. If $\mathcal{R}$ and $\mathcal{Q}$ are the sets of triples and quads displayed by $\mathcal{N}$, respectively, then the sets of triples and quads displayed by $\mathcal{N}'$ are obtained from $\mathcal{R}$ and $\mathcal{Q}$ by deleting b.*

We now prove the uniqueness part of Theorem 2.

Proof of Theorem 2(i) The proof is by induction on the size of X. Since $\mathcal{N}$ is normal, if $|X| = 1$, then $\mathcal{N}$ consists of an isolated vertex and, if $|X| = 2$, then $\mathcal{N}$ consist of two leaves adjoined to the root. In both cases, the theorem holds. Now suppose that $|X| \geq 3$, and that the theorem holds for all normal networks with at most $|X| - 1$ leaves. Let $\mathcal{R}$ and $\mathcal{Q}$ be the sets of triples and quads, respectively, displayed by $\mathcal{N}$. Let $\mathcal{N}_1$ be a normal network on X such that the sets of triples and quads displayed $\mathcal{N}_1$ are $\mathcal{R}$ and $\mathcal{Q}$, respectively. By

Lemma 1, $\mathcal{N}$ has either a cherry, $\{a, b\}$ say, or a reticulated cherry, $\{a, b\}$ with reticulation leaf b say.

First suppose that $\{a, b\}$ is a cherry of $\mathcal{N}$. Then, by Lemma 2, $\{a, b\}$ is a cherry of $\mathcal{N}_1$. Let $\mathcal{N}'$ and $\mathcal{N}_1'$ denote the normal networks obtained from $\mathcal{N}$ and $\mathcal{N}_1$, respectively, by deleting b. By Lemma 7, the sets of triples and quads of $\mathcal{N}'$ and $\mathcal{N}_1'$ are the same and so, by the induction assumption, up to isomorphism, $\mathcal{N}' = \mathcal{N}_1'$. Since $\{a, b\}$ is a cherry of both $\mathcal{N}$ and $\mathcal{N}_1$, it follows that, up to isomorphism, $\mathcal{N} = \mathcal{N}_1$. Thus, if $\{a, b\}$ is a cherry of $\mathcal{N}$, part (i) of the theorem holds.

Now suppose that $\{a, b\}$ is a reticulated cherry of $\mathcal{N}$ in which b is the reticulation leaf and V_{g_b} is the visibility set of g_b. Then, by Lemma 3, $\{a, b\}$ is a reticulated cherry of $\mathcal{N}_1$ in which b is the reticulation leaf. Furthermore, if p_b' denotes the parent of b in $\mathcal{N}_1$, and g_b' denotes the parent of p_b' that is not the parent of a in $\mathcal{N}_1$, then, by the same lemma, V_{g_b} is the visibility set of g_b'. Let $\mathcal{N}'$ and $\mathcal{N}_1'$ denote the normal networks obtained from $\mathcal{N}$ and $\mathcal{N}_1$, respectively, by deleting b. By Lemma 7, the sets of triples and quads of $\mathcal{N}'$ and $\mathcal{N}_1'$ coincide. Therefore, by the induction assumption, up to isomorphism, $\mathcal{N}' = \mathcal{N}_1'$. We complete the proof by showing that, by subdividing, in $\mathcal{N}'$ (equivalently, $\mathcal{N}_1'$) there is exactly one arc to insert p_a to adjoin (p_a, p_b) and exactly one arc to insert g_b to adjoin (g_b, p_b) so that, together with the arc (p_b, b), the resulting network is normal, and displays $\mathcal{R}$ and $\mathcal{Q}$. We do this using only $\mathcal{R}$, $\mathcal{Q}$, and V_{g_b}.

Evidently, there is exactly one arc to adjoin (p_a, p_b) in $\mathcal{N}'$, namely the arc incident to a. Now consider the placement of g_b in $\mathcal{N}'$. Let U be the subset of vertices of $\mathcal{N}'$ consisting of those vertices u having the property that $V_u = V_{g_b}$. Observe that in $\mathcal{N}$, the child of g_b that is not p_b has this property, and so U is non-empty. We analyse the vertices in U to eventually show that there is exactly one arc in $\mathcal{N}'$ to adjoin (g_b, p_b).

2.1 *If $|U| \geq 2$, and u and u' are distinct vertices in U, then there is a directed path in $\mathcal{N}'$ from either u to u' or u' to u, but not both.*

Proof Assume that there is no directed path from u to u' or from u' to u. Then, if ℓ' is the leaf at the end of a tree path for u', there is a directed path from the root of $\mathcal{N}'$ to ℓ' avoiding u, and so $\ell' \notin V_u$. But $\ell' \in V_{u'}$, a contradiction. Since $\mathcal{N}'$ is acyclic, (2.1) now follows. □

2.2 *There is a directed path P in $\mathcal{N}'$ from its root to a leaf such that the vertex set of P contains U.*

Proof Let P be a directed path from the root of $\mathcal{N}'$ to a leaf ℓ such that no directed path from the root of $\mathcal{N}$ to a leaf has more vertices in U than P. Order the vertices in U on P, say $u_1, u_2, \ldots, u_k$, so that it is consistent with P. That is, u_i is before u_j on P precisely if $i < j$. If there is a vertex v in U that is not on P, then, by (2.1) and the maximality of P, for some $i \in \{1, 2, \ldots, k-1\}$, there is a directed path from each of $u_1, u_2, \ldots, u_i$ to v, and no such path from u_{i+1} to v, but there is a directed path from v to u_{i+1}.

Since $\mathcal{N}'$ is acyclic, it follows that we can construct a path in $\mathcal{N}'$ by taking the subpath of P from the root of $\mathcal{N}'$ to u_i, and then adjoining a directed path from u_i to v, a directed path from v to u_{i+1}, and the subpath of P from u_{i+1} to ℓ. This constructed path contradicts the maximality of P. Thus we may assume that the vertex set of P contains U. Note that an analogous argument applies if there is no path from u_1 to v. □

Let $u_1, u_2, \ldots, u_k$ denote the order in which the vertices of U appear in P. As $\mathcal{N}'$ is normal, we may assume that the subpath of P from u_k to ℓ is a tree path. Let P' denote the subpath of P from u_1 to u_k.

2.3 *Every vertex in P' is in U and, except possibly u_1, every vertex in P' is a tree vertex or a leaf.*

Proof First assume that $w \neq u_1$ is a reticulation in P'. Without loss of generality, we may choose w to be the reticulation in P' closest to u_1. Let v_1 and v_2 denote the parents of w in $\mathcal{N}'$, and let ℓ_1 and ℓ_2 be leaves at the end of tree paths for v_1 and v_2, respectively. We may assume v_1 is in P'. Then $\ell_1 \in V_{u_1}$ as either $u_1 = v_1$ or there is no reticulation between u_1 and v_1 in P' except possibly u_1. But $\ell_1 \notin V_{u_k}$, a contradiction. Thus every vertex in P', except possibly u_1, is a tree vertex or a leaf. Now assume that there is a vertex w on P' that is not in U. Then, as the subpath of P' from w to u_k consists of tree vertices, it follows that $V_{u_k} \subseteq V_w$. If $V_{u_k} = V_w$, then $V_w \in U$, a contradiction. Moreover, if $V_w - V_{u_k}$ is non-empty, then, as the subpath of P' between u_1 and w consists of tree vertices, $V_{u_k} \subsetneqq V_{u_1}$, again a contradiction. Thus every vertex on P' is in U, thereby completing the proof of (2.3). □

Now, let $\mathcal{I} = \{1, 2, \ldots, k-1\}$ if u_1 is a tree vertex, and let $\mathcal{I} = \{2, 3, \ldots, k-1\}$ otherwise. Furthermore, for all $i \in \mathcal{I}$, let v_i denote the child of u_i that is not on P, and let m_i denote the leaf at the end of a tree path for v_i. An illustration of the set-up is shown in Fig. 6, where u_1 is a reticulation so $\mathcal{I} = \{2, 3, \ldots, k-1\}$, and dotted arcs represent a tree path to a leaf. If, for some $i \in \mathcal{I}$, the vertex v_i is a tree vertex or a leaf, then it is easily checked that $m_i \in V_{u_i}$, but $m_i \notin V_{u_k}$. This contradiction implies that v_i is a reticulation for each $i \in \mathcal{I}$. Clearly, the elements in $\{m_i : i \in \mathcal{I}\}$ are pairwise distinct. Also, if $u_k \neq \ell$, let u'_k be the child of u_k that is not on P. If u'_k is a reticulation, then the visibility set of the child of u_k that is on P is V_{u_k}, and therefore an element in U, a contradiction. Hence, if $u_k \neq \ell$, then u'_k is a tree vertex or a leaf.

At last, we consider the placement of g_b in $\mathcal{N}'$. By the construction of P, the vertex g_b corresponds to a subdivision of an arc directed into a vertex in $\{u_i : i \in \mathcal{I} \cup \{k\}\}$. If $k = 1$, then u_1 is a tree vertex or a leaf, and the unique placement of g_b is a subdivision of the arc directed into u_1. If $k = 2$ and u_1 is a reticulation, then u_2 is a tree vertex or a leaf, and the unique placement of g_b is a subdivision of the arc directed into u_2. So assume that either $k = 2$ and u_1 is a tree vertex, or $k \geq 3$. Let

$$\mathcal{Q}_P = \{(m_i, \ell, b, a) \in \mathcal{Q} : i \in \mathcal{I}\}.$$

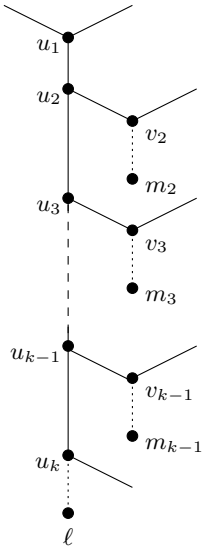


Fig. 6 The vertex set $U = \{u_1, u_2, \ldots, u_k\}$ in the proof of Theorem 2(i), where dotted arcs represent a tree path to a leaf.

Furthermore, let i' be the minimum element in $\mathcal{I}$ for which $(m_{i'}, \ell, b, a) \in \mathcal{Q}_P$. Then it is easily seen that each of

$$(m_{i'}, \ell, b, a), (m_{i'+1}, \ell, b, a), \ldots, (m_{k-1}, \ell, b, a)$$

is in $\mathcal{Q}_P$. In particular, the unique placement of g_b is a subdivision of the arc directed into u_k if $\mathcal{Q}_P$ is empty and $u_{i'}$ otherwise. This completes the proof of part (i) of the theorem. □

5.2 The algorithm

Let $\mathcal{R}$ and $\mathcal{Q}$ be the sets of triples and quads, respectively, of a normal network $\mathcal{N}$ on X. Called Construct Normal, we now present a recursive algorithm whose input is X, $\mathcal{R}$, and $\mathcal{Q}$ and that returns a normal network $\mathcal{N}_0$ that is isomorphic to $\mathcal{N}$. The correctness of the algorithm is essentially established in the constructive proof of Theorem 2(i), and so it is omitted. The running time of Construct Normal is given immediately after its description.

1. If $|X| = 1$, then return the phylogenetic network consisting of the single vertex in X.
2. If $|X| = 2$, then return the phylogenetic network consisting of the two leaves in X adjoined to the root.
3. Else $|X| \geq 3$. Find either $\{a, b\} \subseteq X$ satisfying the sufficiency condition of its namesake in the statement of Lemma 2, in which case $\{a, b\}$ is a cherry, or $\{a, b\} \subseteq X$ and $W_b \subseteq X - \{a, b\}$, where W_b is a candidate set for

b, satisfying the sufficiency conditions of their namesakes in the statement of Lemma 3, in which case $\{a, b\}$ is a reticulated cherry where b is the reticulation leaf and W_b is the visibility set of g_b.

4. Delete b in $\mathcal{R}$ and $\mathcal{Q}$ to give the sets $\mathcal{R}'$ and $\mathcal{Q}'$ of triples and quads, respectively, on $X' = X - \{b\}$.
   - 4.1. If $\{a, b\} \subseteq X$ satisfies the sufficiency condition in Lemma 2, then apply Construct Normal to input X', $\mathcal{R}'$, and $\mathcal{Q}'$, construct $\mathcal{N}_0$ from the returned normal network $\mathcal{N}_0'$ on X' by subdividing the arc directed into a with a new vertex p_a, adjoin a new leaf b to p_a via the new arc (p_a, b), and return $\mathcal{N}_0$.
   - 4.2. Else, $\{a, b\} \subseteq X$ and $W_b \subseteq X - \{a, b\}$ satisfy the sufficiency conditions of Lemma 3. Apply Construct Normal to input X', $\mathcal{R}'$, and $\mathcal{Q}'$, and construct $\mathcal{N}_0$ from the returned normal network $\mathcal{N}_0'$ on X' as follows.
     - 4.2.1. Find the vertex u whose visibility set is W_b and whose cluster set is minimal with respect to containing W_b.
     - 4.2.2. Find the path $u_1, u_2, \ldots, u_k$ of vertices, where $u_k = u$, consisting of precisely those vertices in $\mathcal{N}_0'$ whose visibility set is W_b. Let ℓ denote the leaf at the end of a tree path for u_k.
     - 4.2.3. Let $\mathcal{I} = \{1, 2, \ldots, k-1\}$ if u_1 is a tree vertex; otherwise, let $\mathcal{I} = \{2, 3, \ldots, k-1\}$. For each $i \in \mathcal{I}$, let v_i denote the reticulation child of u_i, and let m_i denote the leaf at the end of a tree path for v_i.
       - 4.2.3.1. If $k = 1$, subdivide the arc directed into u_1 with a new vertex g_b.
       - 4.2.3.2. If $k = 2$ and u_1 is a reticulation, subdivide the arc directed into u_2 with a new vertex g_b.
       - 4.2.3.3. Else, we have either $k = 2$ and u_1 is a tree vertex, or $k \geq 3$. If there is no quad (m_i, ℓ, b, a) in $\mathcal{Q}$, where $i \in \mathcal{I}$, subdivide the arc directed into u_k with a new vertex g_b. Otherwise, subdivide the arc directed into $u_{i'}$, where i' is the smallest i such that $(m_{i'}, \ell, b, a) \in \mathcal{Q}$, with a new vertex g_b.
     - 4.2.4. Subdivide the arc directed into a with a new vertex p_a, adjoin a new vertex p_b via new arcs (p_a, p_b) and (g_b, p_b), adjoin a new leaf b via a new arc (p_b, b), and set $\mathcal{N}_0$ to be the resulting network.
     - 4.2.5. Return $\mathcal{N}_0$.

We now consider the running time of Construct Normal.

Proof of Theorem 2(ii) The algorithm takes as input a set X, and sets $\mathcal{R}$ and $\mathcal{Q}$ of triples and quads, respectively, of a normal network $\mathcal{N}$ on X. If $|X| \in \{1, 2\}$, then the algorithm runs in constant time. If $|X| \geq 3$, then the algorithm begins by either finding $\{a, b\} \subseteq X$ satisfying the sufficiency conditions of its namesake in the statement of Lemma 2, in which case $\{a, b\}$ is a cherry, or finding $\{a, b\} \subseteq X$ and $W_b \subseteq X - \{a, b\}$, where W_b is a candidate set for b, satisfying the sufficiency conditions of their namesakes in Lemmas 3, in which

case $\{a, b\}$ is a reticulated cherry where b is the reticulation leaf and W_b is the visibility set of g_b. In the worst possible instance, the longest running part of this process involves finding a 2-element subset $\{a, b\}$ of X and a candidate set W_b satisfying Lemma 3. Clearly, there are at most $|X|^2$ choices for a 2-element subset $\{a, b\}$ of X. Furthermore, by Lemma 4, for each 2-element subset $\{a, b\}$ of X, there are at most $|X|$ candidate sets for b. Now, by Lemma 5, it takes

$$O(|X|^2|\mathcal{R}| + |X|^2|\mathcal{Q}|)$$

time to find each of the possible $|X|$ candidate sets for b. Therefore, as it takes

$$O(|X||\mathcal{R}| + |X|^2|\mathcal{Q}| + |X|^2|\mathcal{R}||\mathcal{Q}|) = O(|X|^2|\mathcal{R}||\mathcal{Q}|)$$

time to check whether a given 2-element subset and candidate set satisfy the sufficiency conditions in Lemma 3, it follows that the running time to complete Step 3 is

$$O(|X|^3(|X|^2|\mathcal{R}| + |X|^2|\mathcal{Q}| + |X|^2|\mathcal{R}||\mathcal{Q}|)) = O(|X|^5|\mathcal{R}||\mathcal{Q}|).$$

We next delete b in $\mathcal{R}$ and $\mathcal{Q}$, and this takes at most $O(|\mathcal{R}| + |\mathcal{Q}|)$ time. Clearly, Step 4.1 takes less time to complete than Step 4.2, so we may assume that the latter is reached. To determine the cluster set of a vertex u of $\mathcal{N}_0'$, a single postorder transversal of $\mathcal{N}_0'$ is sufficient. Furthermore, to determine the visibility set of a vertex u of $\mathcal{N}_0'$, we delete u and its incident arcs and check, for each leaf ℓ in X' whether the resulting rooted acyclic digraph, D' say, has a directed path from its root ρ' to ℓ. Effectively, we are finding the 'cluster set' $C_{\rho'}$ of ρ' in D'. The visibility set of u in $\mathcal{N}_0'$ consists precisely of those leaves in X' not in $C_{\rho'}$. A single postorder transversal of D' is sufficient to determine $C_{\rho'}$. Since $\mathcal{N}$ has at most $O(|X|)$ vertices and, therefore, at most $O(|X|)$ arcs in total [2] (also see [20]), it takes $O(|X|)$ time to find the visibility set of u, and so it takes $O(|X|^2)$ time to complete Steps 4.2.1 and 4.2.2. Once $u_1, u_2, \ldots, u_k$ are determined, finding the leaves $m_2, m_3, \ldots, m_{k-1}$, and possibly m_1 if u_1 is a tree vertex and $k \geq 2$, takes $O(|X|^2)$ time as $k \leq |X|$. If performed, each of Steps 4.2.3.1 and 4.2.3.2 takes constant time, while Step 4.2.3.3 takes $O(|X||\mathcal{Q}|)$ time. Thus the location of g_b, ignoring the running time of Step 3, can be found in time $O(|X|^2 + |X||\mathcal{Q}|)$. Since Step 4.2.4 takes constant time, it follows that Step 4 takes $O(|X|^2 + |X||\mathcal{Q}|)$ time to complete. Hence $\mathcal{N}_0$ can be returned in $O(|X|^5|\mathcal{R}||\mathcal{Q}|)$ time which is also the total time of each iteration.

When recursing, the input to the recursive call is a set X', and sets $\mathcal{R}'$ and $\mathcal{Q}'$ of triples and quads of a normal network on $|X| - 1$ leaves. Therefore the total number of iterations is $O(|X|)$. Hence Construct Normal completes in $O(|X|^6|\mathcal{R}||\mathcal{Q}|)$ time, that is, in $O(|X|^{13})$ time as $|\mathcal{R}| \leq |X|^3$ and $|\mathcal{Q}| \leq |X|^4$. This completes the proof of Theorem 2(ii). □

6 Temporal normal networks

In this section, we consider a certain subclass of normal networks and briefly outline how they can be reconstructed from their sets of displayed triples and quads. Let $\mathcal{N}$ be a phylogenetic network on X, and let V be the vertex set of $\mathcal{N}$. We say that $\mathcal{N}$ is *temporal* if there exists a map $t : V \to \mathbb{R}^+$ such that for all $u, v \in V$, we have $t(u) = t(v)$ if (u, v) is a reticulation arc and $t(u) < t(v)$ if (u, v) is a tree arc. Note that the two networks shown in Fig. 1 are temporal and, so, temporal normal networks cannot be determined by their set of displayed triples. Biologically, if a phylogenetic network is temporal, then it satisfies two natural timing constraints. Firstly, speciation events occur successively and, secondly, reticulation events occur contemporaneously and so such events are realised by coexisting ancestral species.

Let $\mathcal{N}$ be a phylogenetic network on X, and let $\{a, b, c\}$ be a three-element subset of X. If p_b is a reticulation, and both $\{a, b\}$ and $\{b, c\}$ are reticulated cherries, then $\{a, b, c\}$ is referred to as a *double-reticulated cherry* of $\mathcal{N}$ in which b is the *reticulation leaf*.

Now, for a temporal normal network $\mathcal{N}$, let u be a tree vertex such that, for all other tree vertices u', we have $t(u) \geq t(u')$. It is then straightforward to show that $\mathcal{N}$ has either a cherry or a double-reticulated cherry (see, for example, [5]). For a double-reticulated cherry $\{a, b, c\}$ in $\mathcal{N}$ in which b is the reticulation leaf, consider the operation of deleting b (as defined for a more general reticulated cherry in Section 5). Recall that this operation corresponds to deleting b, p_b, and their incident arcs, and suppressing the resulting degree-two vertices. Noting that a temporal tree-child network is normal, it follows from [5, Lemma 5.1] that the phylogenetic network obtained from $\mathcal{N}$ by deleting b is temporal and normal. Analogous to Lemma 3, the next lemma establishes necessary and sufficient conditions to recognise a double-reticulated cherry in a normal network.

Lemma 8 *Let $\mathcal{N}$ be a normal network on X, where $|X| \geq 3$, and let $\mathcal{R}$ and $\mathcal{Q}$ be the sets of triples and quads displayed by $\mathcal{N}$, respectively. Let $\{a, b, c\} \subseteq X$. Then $\{a, b, c\}$ is a double-reticulated cherry of $\mathcal{N}$ in which b is the reticulation leaf if and only if $\{a, b, c\}$ satisfies the following three properties:*

(i) *For all $x \in X - \{a, b\}$, the triple $ab|x \in \mathcal{R}$ and, for all $x \in X - \{b, c\}$, the triple $bc|x \in \mathcal{R}$, but the triple $ac|b \notin \mathcal{R}$.*
(ii) *There is no $x \in X - \{a, b, c\}$ such that (x, b, a, c), (x, b, c, a), (x, a, b, c), or (x, c, b, a) is in $\mathcal{Q}$.*
(iii) *If there exists an $x \in X - \{a, b, c\}$ such that $ac|x \in \mathcal{R}$, then (a, b, c, x) and (c, b, a, x) are in $\mathcal{Q}$.*

We omit the proof as it is a consequence of Lemma 3. In particular, in viewing a double-reticulated cherry $\{a, b, c\}$ with reticulation leaf b as a reticulated cherry $\{a, b\}$ with reticulation leaf b, observe that the visibility set of g_b is $\{c\}$. Of course, the same observation applies to $\{b, c\}$ but with the roles of a and c interchanged. We now turn back to temporal normal networks since

they are phylogenetic networks for which we can repeatedly delete the reticulation leaf of a double-reticulated cherry or a leaf of a cherry until we are left with a single vertex. Hence, with Lemma 8 in hand, we can reconstruct a temporal normal network from its sets of displayed triples and quads using an algorithm that is a simplification of Construct Normal. Without going into details, Steps 3 and 4.2 of Construct Normal can be simplified in the following way, while the other steps remain unchanged. If the input is a temporal normal network $\mathcal{N}$ on X as well as its sets of displayed triples and quads, Step 3 finds $\{a, b\} \subseteq X$ or $\{a, b, c\} \subseteq X$ that satisfies the conditions of their namesakes in the statements of Lemmas 2 or 8, respectively. Moreover, Step 4.2 reconstructs a temporal normal network on X from a temporal normal network on $X - \{b\}$ by subdividing the arc directed into a (resp. c) with a new vertex p_a (resp. p_c), adjoining a new vertex p_b via new arcs (p_a, p_b) and (p_c, p_b), and adjoining a new leaf b via a new arc (p_b, b).

Similar to establishing the running time of Construct Normal for the proof of Theorem 2(ii), the running time of the above simplification of Construct Normal is as follows. The total number of iterations is $O(|X|)$. Furthermore, the running time of each iteration is determined by the time it takes to find a 3-element subset $\{a, b, c\}$ of X satisfying the sufficiency conditions of its namesake in the statement of Lemma 8. It is easily checked that this time is $O(|X||\mathcal{R}| + |X||\mathcal{Q}| + |X||\mathcal{R}||\mathcal{Q}|)$, and so the total running time is

$$O(|X| \cdot |X|^3(|X||\mathcal{R}| + |X||\mathcal{Q}| + |X||\mathcal{R}||\mathcal{Q}|)) = O(|X|^{12}).$$

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

References

1. Aho AV, Sagiv Y, Szymanski TG, Ullman, JD (1981) Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. SIAM Journal on Computing 10:405–421
2. Bickner DR, (2012) On normal networks. PhD thesis, Iowa State University, Ames, Iowa
3. Bininda-Emonds ORP (2004) The evolution of supertrees. Trends in Ecology and Evolution 19:315–322
4. Bordewich M, Huber KT, Moulton V, Semple C (2018) Recovering normal networks from shortest inter-taxa distance information. Journal of Mathematical Biology 77:571–594
5. Bordewich M, Semple C (2016) Determining phylogenetic networks from inter-taxa distances. Journal of Mathematical Biology 73:283–303
6. Bulteau L, Weller M (2019) Parameterized algorithms in bioinformatics: an overview. Algorithms 12:256
7. Cardona G, Rosselló F, Valiente G (2009) Comparison of tree-child phylogenetic networks. IEEE/ACM Transactions on Computational Biology and Bioinformatics 6:552–569
8. Francis A, Moulton V (2018) Identifiability of tree-child phylogenetic networks under a probabilistic recombination-mutation model of evolution. Journal of Theoretical Biology 446:160–167

9. Francis AR, Steel M (2015) Which phylogenetic networks are merely trees with additional arcs? Systematic Biology 64:768–777
10. Gambette P, Huber KT (2012) On encodings of phylogenetic networks of bounded level. Journal of Mathematical Biology 65:157–180
11. Habib M, To TH (2012) Constructing a minimum phylogenetic network from a dense triplet set. Journal of Bioinformatics and Computational Biology 10:1250013
12. Huber KT, van Iersel L, Moulton V, Scornavacca C, Wu T (2017) Reconstructing phylogenetic level-1 networks from nondense binet and trinet sets. Algorithmica 77:173–200
13. Huber KT, van Iersel L, Moulton V, Wu T (2015) How much information is needed to infer reticulate evolutionary histories? Systematic Biology 64:102–111.
14. Huber KT, Moulton V (2012) Encoding and constructing 1-nested phylogenetic networks with trinets. Algorithmica 66: 714–738
15. van Iersel L., Kelk S (2011) Constructing the simplest possible phylogenetic network from triplets. Algorithmica 60:207–235
16. van Iersel L, Moulton V (2014) Trinets encode tree-child and level-2 phylogenetic networks. Journal of Mathematical Biology 68:1707–1729
17. van Iersel L, Moulton V, de Swart E, Wu T (2017) Binets: Fundamental building blocks for phylogenetic networks. Bulletin of Mathematical Biology 79:1135–1154
18. Jansson J, Nguyen NB, Sung WK (2006) Algorithms for combining rooted triplets into a galled phylogenetic network. SIAM Journal on Computing 35:1098–1121
19. Jansson J, Sung WK (2006) Inferring a level-1 phylogenetic network from a dense set of rooted triplets. Theoretical Computer Science 363:60–68
20. McDiarmid C, Semple C, Welsh D (2015) Counting phylogenetic networks. Annals of Combinatorics 19:205–224
21. Murakami Y, van Iersel L, Janssen R, Jones M, Moulton V (2019) Reconstructing tree-child networks from reticulate-edge-deleted subnetworks. Bulletin of Mathematical Biology 81:3823–3863
22. Pardi F, Scornavacca C (2015) Reconstructible phylogenetic networks: Do not distinguish the indistinguishable. PLoS Computational Biology 11:e1004135
23. Semple C, Steel M (2003) Phylogenetics, Oxford University Press, New York
24. Willson SJ (2011) Regular networks can be uniquely constructed from their trees. IEEE/ACM Transactions on Computational Biology and Bioinformatics 8:785–796
25. Willson SJ (2010) Properties of normal phylogenetic networks. Bulletin of Mathematical Biology 72:340–358