Bull Math Biol (2017) 79:1135–1154
DOI 10.1007/s11538-017-0275-4

Society for Mathematical Biology

CrossMark

ORIGINAL ARTICLE

# Binets: Fundamental Building Blocks for Phylogenetic Networks

**Leo van Iersel[1] · Vincent Moulton[2] · Eveline de Swart[1] · Taoyang Wu[2]**

Received: 6 November 2016 / Accepted: 24 March 2017 / Published online: 6 April 2017

**Abstract** Phylogenetic networks are a generalization of evolutionary trees that are used by biologists to represent the evolution of organisms which have undergone reticulate evolution. Essentially, a phylogenetic network is a directed acyclic graph having a unique root in which the leaves are labelled by a given set of species. Recently, some approaches have been developed to construct phylogenetic networks from collections of networks on 2- and 3-leaved networks, which are known as binets and trinets, respectively. Here we study in more depth properties of collections of binets, one of the simplest possible types of networks into which a phylogenetic network can be decomposed. More specifically, we show that if a collection of level-1 binets is compatible with some binary network, then it is also compatible with a binary level-1 network. Our proofs are based on useful structural results concerning lowest stable ancestors in networks. In addition, we show that, although the binets do not determine

Part of this work was conducted while Vincent Moulton was visiting Leo van Iersel on a visitors grant funded by the Netherlands Organization for Scientific Research (NWO). Leo van Iersel was partially supported by NWO, including Vidi grant 639.072.602, and partially by the 4TU Applied Mathematics Institute. We thank the editor and the two anonymous referees for their constructive comments.

✉ Leo van Iersel
l.j.j.v.iersel@gmail.com

Vincent Moulton
v.moulton@uea.ac.uk

Eveline de Swart
Eveline_de_Swart@hotmail.com

Taoyang Wu
Taoyang.Wu@uea.ac.uk

[1] Delft Institute of Applied Mathematics, Delft University of Technology, Delft, The Netherlands

[2] School of Computing Sciences, University of East Anglia, Norwich, UK

the topology of the network, they do determine the number of reticulations in the network, which is one of its most important parameters. We also consider algorithmic questions concerning binets. We show that deciding whether an arbitrary set of binets is compatible with some network is at least as hard as the well-known graph isomorphism problem. However, if we restrict to level-1 binets, it is possible to decide in polynomial time whether there exists a binary network that displays all the binets. We also show that to find a network that displays a maximum number of the binets is NP-hard, but that there exists a simple polynomial-time 1/3-approximation algorithm for this problem. It is hoped that these results will eventually assist in the development of new methods for constructing phylogenetic networks from collections of smaller networks.

**Keywords** Reticulate evolution · Phylogenetic network · Subnetwork · Binet · Algorithm

## 1 Introduction

Phylogenetic networks are a generalization of evolutionary trees which biologists use to represent the evolution of species that have undergone reticulate evolution. Such networks are essentially directed acyclic graphs having a unique root in which the leaves are labelled by a set $X$ of species (Huson et al. 2010). In contrast to evolutionary trees, which can only represent speciation events, phylogenetic networks permit the representation of evolutionary events such as gene transfer and hybridization which are known to occur in organisms such as bacteria and plants, respectively. Although theoretical properties of evolutionary trees have been studied since at least the 1970s, phylogenetic networks have been considered from this perspective only more recently, especially the rooted variants which we will focus on in this paper.

One of the most important open questions concerning phylogenetic networks is how to construct them for biological datasets (Bapteste et al. 2013). It is now common practice for biologists to construct evolutionary trees from molecular data, and several computer programs are available for this purpose (Felsenstein 2004). However, the problem of constructing networks from such data is an active area of research, and there are only a limited number of programs available for biologists to perform this task. A survey of some of these methods and the theory underpinning phylogenetic networks may be found in Gusfield (2014), Huson et al. (2010), Morrison (2011).

One approach that has been recently developed for constructing phylogenetic networks involves building them up from smaller networks, using what can be thought of as a divide-and-conquer approach (Oldman et al. 2016). In particular, for a set $X$ of species, a network is constructed for every subset of $X$ size 3 (called a *trinet*), and then the trinets are puzzled together to build a network (see Fig. 1 for an example of a trinet). This approach constructs and is based on *level-1* networks, networks that are slightly more general than evolutionary trees (see Sect. 2 for the definition of such networks).

At first sight, it might appear that trinets are the simplest possible networks that could be considered for building up networks from smaller ones. However, trinets con-

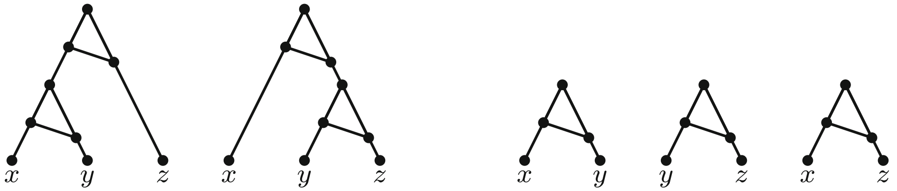


**Fig. 1** An example of two level-1 trinets (*left*) that display the same set of three binets (*right*). All arcs are directed downwards

tain even simpler networks called *binets*, networks with 2 leaves (see e.g. Fig. 1 for a level-1 trinet and the binets that it displays). Note that whereas binets are the smallest informative building blocks for phylogenetic networks, for rooted phylogenetic trees, these are 3-leaf trees (see e.g. Byrka et al. 2010). Interestingly, even though binets are in themselves very simple, the collection of binets displayed by a network can still contain some useful information concerning the network. Indeed, in the aforementioned approach for building level-1 networks from trinets, binets are used in the process of puzzling together the trinets.

In light of these considerations some obvious questions immediately arise concerning binets. For example, when is a collection of binets displayed by some phylogenetic network (the compatibility problem), and how much information might we expect to extract concerning a phylogenetic network by just looking at the collection of binets that it displays? In this paper, we shall address these and related algorithmic questions concerning binets. It is hoped that these results will be useful in future for developing improved methods for constructing phylogenetic networks from smaller networks.

We now present a summary of the rest of the paper. After introducing some preliminaries concerning phylogenetic networks in the next section, we derive a key structural result for networks (Corollary 1) which is useful in identifying which of the two possible types of binet is displayed on two leaves within a binary phylogenetic network (that is a network in which all internal vertices have degree 3). Using this theorem, in Sect. 4 we show that the collection of level-1 binets displayed by *any* binary phylogenetic network can always be displayed by some binary level-1 network (Theorem 3). This reduces the problem of understanding binets displayed by arbitrary binary networks to level-1 networks. To prove this result, we develop a framework which also implies that there is a polynomial-time algorithm in $|X|$ for deciding whether or not a collection of level-1 binets with combined leaf set $X$ can be displayed by some network with leaf set $X$, and, if it is, gives a level-1 network that does this (see Sect. 6). Note that this is related to an algorithm presented in Huber et al. (2015).

In Sect. 5, we turn to the question as to what can be deduced about the features of a phylogenetic network just by considering the collection of binets that it displays. Note that, as might be expected, there are networks—even trinets—that display the same set of binets but that are not equivalent. For example, the two trinets in Fig. 1 both display the same set of binets, but they are not equivalent. Even so, we will show in Theorem 4 that if two level-1 networks both display exactly the same collection of binets, then they must have the same number of reticulation vertices (in-degree-2 vertices). Note that the number of such vertices corresponds to the number of reticulate evolutionary events, such as hybridization, that took place in the evolutionary history of the species labelling the leaves of the network. Consequently, the binets displayed by a network can at least capture a useful course-grained feature of the network in question.

In Sects. 6 and 7, we consider some algorithmic questions concerning binets. As we have mentioned above, it can be decided in polynomial time in $|X|$ as to when a collection of binets with combined leaf set $X$ is displayed by some level-1 network on $X$. However, we show that if we consider arbitrary binets (i.e. not necessarily binary or level-1) then this decision problem becomes at least as hard as the graph isomorphism problem (see Theorem 5), one of the most famous problems whose complexity is still unknown. In addition, in Sect. 7 we consider a related problem which, for a given collection of binary level-1 binets, asks for a network which displays the maximum number of binets in this collection. This is closely related to the maximum rooted triplet consistency problem for evolutionary trees (Byrka et al. 2010). We show that the binet problem is NP-complete (Theorem 6), by giving a reduction from the feedback arc set problem. However, we also show that the problem is 1/3-approximable. In fact, given any collection of binary level-1 binets we can always find some network that displays at least 1/3 of the binets (see Theorem 7). We conclude in Sect. 8 with discussion of some possible future research directions, and a brief discussion of a potential application of our results.

## 2 Preliminaries

Throughout this paper, $X$ is a nonempty finite set (which usually represents a set of species or organisms).

### 2.1 Digraphs

A *directed graph*, or *digraph* for short, $G = (V, E)$ consists of a finite set $V = V(G)$ of *vertices* and a set $E = E(G)$ of *arcs*, where each arc is an ordered pair $(u, v)$ of vertices in $V$ in which $u$ is said to be a *parent* of $v$, denoted by $u = p(v)$, and $v$ a *child* of $u$. All digraphs studied here contain no loops, that is, vertices that are children of themselves. The *in-degree* of vertex $u$ is the number of vertices $v$ in $V$ such that $(v, u)$ is an arc, and the *out-degree* of $u$ is the number of vertices $w$ with $(u, w)$ being an arc. A *root* is a vertex with in-degree 0. A *leaf* is a vertex of out-degree 0, and the set of leaves is denoted by $L(G)$. Any vertex in $G$ that is neither a root nor a leaf is referred to as an *interior vertex*. In addition, an interior vertex is a *tree vertex* if it has in-degree 1, and a *reticulation vertex* if it has in-degree greater than 1.

A *directed path* or *dipath* in a digraph is a sequence $u_0, u_1, \ldots, u_k$ ($k \geq 1$) of vertices such that $(u_{i-1}, u_i)$ is an arc for $1 \leq i \leq k$. An *acyclic digraph* is a digraph that does not contain any directed path starting and ending at the same vertex. If an acyclic digraph $G$ contains a unique root, which is usually designated by $\rho = \rho(G)$, then it will be referred to as a *rooted acyclic digraph*.

An acyclic digraph $G$ induces a canonical partial order $\prec_G$ on its vertex set $V$, that is, $v \prec_G u$ if there exists a directed path from $u$ to $v$. In this case, we shall say that $v$ is *below* $u$. When the digraph $G$ is clear from the context, $\prec_G$ will be written as $\prec$.

In addition, we write $v \preceq u$ if $u = v$ or $u \prec v$. Given a subset $U$ of the vertex set of an acyclic digraph, we say that $u \in U$ is a *lowest* vertex in $U$ if there is no $v \in U$ with $v \prec u$.

Let $\underline{G}$ be the undirected graph obtained from digraph $G$ by ignoring the direction of the arcs in $G$. Then $G$ is *connected* if $\underline{G}$ is connected, that is, there exists an undirected path between every pair of distinct vertices in $\underline{G}$. Note that a rooted acyclic digraph is necessarily connected (since each connected component of an acyclic digraph has at least one root). A *cut vertex* is a vertex of $G$ whose removal disconnects $\underline{G}$. Similarly, a *cut arc* is an arc of $G$ whose removal disconnects $\underline{G}$. A directed graph is *biconnected* if it contains no cut vertex, and a *biconnected component* of $G$ is a maximal biconnected subgraph, which is called *trivial* if it contains precisely one arc (which is necessarily a cut arc), and *non-trivial* otherwise.

### 2.2 Phylogenetic Networks

A *phylogenetic network* $N$ on $X$ is a rooted acyclic digraph whose leaves are bijectively labelled by the elements in $X$ and which does not contain any vertex with in-degree one and out-degree one. For simplicity, we will just write $L(N) = X$ in case there is no confusion about the labelling. To simplify the argument, throughout this paper we will also assume that all leaves in a phylogenetic network have in-degree one. In addition, a phylogenetic network is *binary* if each tree vertex, as well as the root, has out-degree 2, and each reticulation vertex has in-degree 2 and out-degree 1. Finally, we say a binary phylogenetic network is *level-k* ($k \geq 0$) if each of its biconnected components contains at most $k$ reticulation vertices. To some extent, the concept of the level of a phylogenetic network can be regarded as a measure of its 'distance' to being a phylogenetic tree. In particular, a binary phylogenetic network is a phylogenetic tree if and only if it is level-0. A phylogenetic network is called *simple* if it contains precisely one non-trivial biconnected component $H$ and no cut arcs other than the ones leaving $H$.

Two networks $N_1 = (V_1, E_1)$ and $N_2 = (V_2, E_2)$ on $X$ are said to be *isomorphic* if there exists a bijection $f : V_1 \rightarrow V_2$ such that $f(x) = x$ for all $x \in X$, and $(u, v)$ is an arc in $N_1$ if and only if $(f(u), f(v))$ is an arc in $N_2$.

Finally, the *cluster* of a vertex $u$, denoted by $\mathcal{C}_N(u) = \mathcal{C}(u)$, is defined as the subset of $X$ consisting of the leaves below $u$. Here we will use the convention that $\mathcal{C}(u) = \{u\}$ if $u$ is a leaf.

### 2.3 Stable Ancestors and Binets

Given a phylogenetic network $N$ on $X$ and a subset $U \subseteq V(N)$, a *stable ancestor* of $U$ in $N$ is a vertex $v$ in $V(N)\backslash U$ such that every path in $N$ from the root to a vertex in $U$ contains $v$. Note that for two stable ancestors $u$ and $u'$ of $U$, we have either $u \preceq v$ or $v \preceq u$. Therefore, there exists a unique lowest vertex in the set of stable ancestors of $U$, which will be referred to as the *lowest stable ancestor* of $U$ in $N$ and denoted by $\textsc{lsa}_N(U) = \textsc{lsa}(U)$. Note that for a subset $Y$ of $X$ with $|Y| \geq 2$, there exist two elements $x$ and $y$ in $Y$ such that $\mathrm{LSA}(Y) = \mathrm{LSA}(\{x, y\})$. For simplicity, we also write $\mathrm{LSA}(\{x, y\})$ as $\mathrm{LSA}(x, y)$.

The following property of lowest stable ancestors will be useful.

**Lemma 1** *Suppose that $u$ and $v$ are two vertices in a phylogenetic network such that $u \prec v \prec \mathrm{LSA}(u)$, then we have $\mathrm{LSA}(v) \preceq \mathrm{LSA}(u)$.*

*Proof* Since $u \prec v$, we know that there exists a dipath $P$ from $\rho$ to $u$ that contains $v$. By the definition of lowest stable ancestor, we know that $\mathrm{LSA}(u)$ and $\mathrm{LSA}(v)$ are contained in $P$. Hence, either $\mathrm{LSA}(v) \preceq \mathrm{LSA}(u)$ or $\mathrm{LSA}(u) \prec \mathrm{LSA}(v)$. If $\mathrm{LSA}(u) \prec \mathrm{LSA}(v)$, then we have $v \prec \mathrm{LSA}(u) \prec \mathrm{LSA}(v)$. Then there exists a dipath $P'$ from $\rho$ to $v$ that does not contain $\mathrm{LSA}(u)$ (otherwise $\mathrm{LSA}(u)$ would be a stable ancestor of $v$ that is below $\mathrm{LSA}(v)$). Using that $u \prec v \prec \mathrm{LSA}(u)$, it follows that there exists a dipath from $\rho$ to $u$ that does not contain $\mathrm{LSA}(u)$, a contradiction. Therefore, $\mathrm{LSA}(v) \preceq \mathrm{LSA}(u)$. □

For $Y \subseteq X$, the subnet of $N$ on $Y$, denoted by $N|_Y$, is defined as the subgraph obtained from $N$ by deleting all vertices that are not on any path from $\mathrm{LSA}(Y)$ to elements in $Y$ and subsequently suppressing all in-degree 1 and out-degree 1 vertices and parallel arcs until no such vertices or arcs exist. A network $N'$ is said to be *displayed* by network $N$ if $N' = N|_Y$ for some $Y \subseteq X$.

Note that, by definition, $N|_X = N$ if and only if $\mathrm{LSA}(X) = \rho(N)$. In this case, $N$ is referred to as a *recoverable* network. Note that every subnet of $N$ is necessarily recoverable. Moreover, a collection of subnets is displayed by some network if and only if it is displayed by some recoverable network. Therefore, we assume all networks in this paper to be recoverable.

A *binet* is a phylogenetic network with precisely two leaves, while a *trinet* is a phylogenetic network with precisely three leaves. Let

$$\mathcal{B}(N) = \{N|_Y \, : \, Y \subseteq X \text{ and } |Y| = 2\}$$

be the collection of binets displayed by $N$. Note that there are precisely three binary level-1 binets on a set $\{x, y\}$, and they can be grouped into two types: the "tree type", $T(x, y)$, and the "reticulate type" $R(x; y)$ and $R(y; x)$ (see Fig. 2). A collection of binets $\mathcal{B}$ *on* $X$ is a collection of binets such that the union of the leaf sets of the binets is equal to $X$.

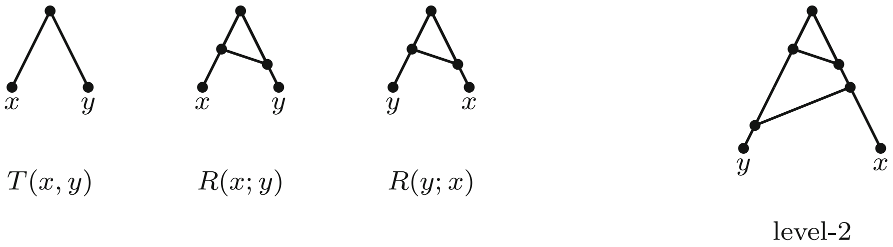


**Fig. 2** The three binary level-1 binets on $\{x, y\}$ and an example of a level-2 binet

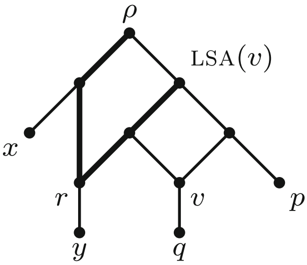


**Fig. 3** Example for the proof of Theorem 1. The lowest stable ancestor of $v$ is not equal to the root $\rho$. An undirected path $P_u$ between $\rho$ and LSA$(v)$ that does not contain the incoming arc of LSA$(v)$ is indicated in bold. Path $P_u$ contains one alternating vertex $r$, for which the lowest stable ancestor is indeed equal to $\rho$

## 3 A Structure Theorem

In this section we present a key result (Corollary 1) concerning the structure of the non-trivial biconnected component of a simple network. Note that a similar result has been obtained for a special collection of (non-binary) phylogenetic networks in Huber et al. (2016).

Let $G$ be a directed acyclic graph and let $P = v_0, v_1, \ldots, v_t$ be an undirected path in the underlying undirected graph $\underline{G}$, then a vertex $v_i$ (with $1 \leq i \leq t-1$) is called *alternating* (with respect to $P$) if we have either $\{(v_{i-1}, v_i), (v_{i+1}, v_i)\} \subseteq E(G)$ or $\{(v_i, v_{i-1}), (v_i, v_{i+1})\} \subseteq E(G)$. The number of alternating vertices contained in $P$ is denoted by ALT$(P)$. Using this concept, we now prove the following theorem. See Fig. 3 for an example.

**Theorem 1** *Let $N$ be a binary phylogenetic network on $X$ whose root $\rho$ is in some non-trivial biconnected component $H$. Then there exists a lowest vertex in $H$ with* LSA$(v) = \rho$.

*Proof* Let $\Gamma_0(H)$ be the set of reticulation vertices $v$ in $H$ for which the distance (length of a shortest directed path) between $\rho$ and LSA$(v)$ is minimum over all reticulation vertices in $H$. Note that $\Gamma_0(H) \neq \emptyset$.

We first show that LSA$(v) = \rho$ for all $v \in \Gamma_0(H)$. Suppose this were not the case. Then there exists a vertex $v \in \Gamma_0(H)$ such that LSA$(v) \prec \rho$. Note that LSA$(v)$ necessarily has out-degree 2 and therefore has in-degree 1 since $N$ is binary and LSA$(v) \neq \rho$. Denote the parent of LSA$(v)$ by $v^*$. Since $H$ is biconnected, there exists some undirected path from $\rho$ to LSA$(v)$ that does not contain the edge $e = \{\text{LSA}(v), v^*\}$. Let $P_u = v_0, \ldots, v_t$, where $v_0 = \rho$ and $v_t =$ LSA$(v)$, be such an undirected path for which ALT$(P_u)$ is minimum.

We claim that ALT$(P_u) = 1$. To see this, note first that since $v_0 = \rho$, $v_t =$ LSA$(v)$ and $v_{t-1} \neq v^*$, we know that $(v_0, v_1)$ and $(v_t, v_{t-1})$ are arcs of $N$. Hence, ALT $(P_u)$ is odd and strictly positive. Assume for the sake of contradiction that ALT $(P_u) \neq 1$, then we have ALT $(P_u) \geq 3$. Let $v_k$ $(1 < k < t)$ be the second alternating vertex contained in $P_u$ (when travelling from $v_0$ to $v_t$).

Now fix a directed path $P_d$ in $N$ from $\rho$ to $v_k$.

If the arc $(v^*, \text{LSA}(v))$ is not contained in $P_d$, then, we can find an undirected path from $\rho$ to LSA$(v)$ that does not contain $e$ and has fewer alternating vertices than $P_u$ by following $P_d$ until we reach a vertex in $\{v_k, \ldots, v_t\}$ and then following $P_u$ to $\text{LSA}(v)$. This gives a contradiction.

Now assume that the arc $\big(v^*, \text{LSA}(v)\big)$ is contained in $P_d$. Then we can find an undirected path from $\rho$ to $\text{LSA}(v)$ that does not contain $e$ and has only one alternating vertex as follows. Follow $P_u$ up to $v_k$ and then follow $P_d$ backward from $v_k$ to $\text{LSA}(v)$. Since this path has fewer alternating vertices than $P_u$, we again obtain a contradiction.

We have thus shown that $\text{ALT}\,(P_u) = 1$. Denoting this alternating vertex in $P_u$ by $r$, then $r$ is necessarily a reticulation by the choice of $P_u$. Hence, $P_u$ consists of two directed paths: a directed path from $\rho$ to $r$ that does not contain $\text{LSA}(v)$ and a directed path from $\text{LSA}(v)$ to $r$. However, this means that $\text{LSA}(v) \prec \text{LSA}(r)$, a contradiction to the assumption that $v \in \Gamma_0(H)$.

Hence, we know that $\Gamma_0(H)$ is the set of reticulation vertices $v$ of $H$ such that $\text{LSA}(v) = \rho$ and that $\Gamma_0(H)$ is not empty.

Now fix a vertex $v$ in $\Gamma_0(H)$ that is lowest over all vertices of $\Gamma_0(H)$, that is, there does not exist a vertex $u$ in $\Gamma_0(H)$ such that $v \prec u$. It remains to show that $v$ is lowest over all vertices of $H$. Assume that this is not the case. Then the child $c$ of $v$ is also in $H$. If $c$ were a reticulation then, by Lemma 1, $\text{LSA}(v) \preceq \text{LSA}(c)$. However, this would imply that $\text{LSA}(c) = \rho$, contradicting the choice of $v$. Hence, $c$ is a tree vertex.

Since $H$ is biconnected, there exists some undirected path from $\rho$ to $c$ that does not contain $v$. Let $P_u = w_0, \ldots, w_t$ be such a path such that $\text{ALT}\,(P_u)$ is minimum. Note that we have $w_0 = \rho$ and $w_t = c$.

Since $c$ is a tree vertex and $P_u$ does not contain its parent $v$, $(w_t, w_{t-1})$ is an arc of $N$. Together with $(w_0, w_1)$ being an arc in $N$, we know that $\text{ALT}\,(P_u)$ is odd and strictly positive. We now show, using a similar proof as above, that $\text{ALT}\,(P_u) = 1$. If this were not the case, then we would have $\text{ALT}\,(P_u) \geq 3$. Let $w_k$ $(1 < k < t)$ be the second alternating vertex contained in $P_u$. We know that $(w_k, w_{k-1})$ and $(w_k, w_{k+1})$ are two arcs contained in $N$. Now fix a directed path $P_d$ in $N$ from $\rho$ to $w_k$.

If the vertex $v$ is not contained in $P_d$, then we can find an undirected path from $\rho$ to $c$ that does not contain $v$ and has fewer alternating vertices than $P_u$ by following $P_d$ from $\rho$ it reaches a vertex from $\{w_k, \ldots, w_t\}$ and then following $P_u$ up to $c$. If $v$ is contained in $P_d$, then we follow $P_u$ from $\rho$ to $w_k$ and then follow $P_d$ from $w_k$ to $c$ and obtain an undirected path from $\rho$ to $c$ that does not contain $v$ and has one alternating vertices, which is less than the number of alternating vertices in $P_u$. In either case, we obtain a contradiction.

We have thus shown that $\text{ALT}\,(P_u) = 1$. Denoting this alternating vertex in $P_u$ by $r$, then $r$ is necessarily a reticulation by the choice of $P_u$. Hence, $P_u$ consists of two directed paths: a directed path from $\rho$ to $r$ that does not contain $v$ and a directed path from $c$ to $r$. However, this means that $v \prec \text{LSA}(r)$, and hence $\text{LSA}(v) \preceq \text{LSA}(r)$ in view of Lemma 1. This implies that $r \in \Gamma_0(H)$, a contradiction to the assumption that $v$ is lowest among $\Gamma_0(H)$. □

The following is a direct consequence of the above theorem.

**Corollary 1** *Suppose that $N$ is a simple binary phylogenetic network. Let $H$ be the unique non-trivial biconnected component of $N$. Then there exists a lowest vertex $v$ of $H$ such that there exist two arc-disjoint directed paths from the root of $N$ to $v$.*

## 4 Displaying Binets by Binary Networks

A collection of binary level-1 binets is *compatible* if there exists some binary network that displays all binets from the collection. In this section, we study the compatibility of binets. Our main result in this section (Theorem 3) shows that when studying the compatibility of binets, we can restrict to binary level-1 networks.

We will restrict ourselves throughout this section to *thin* collections of binets, i.e. collections containing at most one binet on $x$ and $y$ for all distinct $x, y \in X$. Clearly, any collection of binets that is not thin is not compatible.

First, we need some new definitions. Given a digraph $G$, a *sink set* of $G$ is a proper subset $U \subset V(G)$ such that there is no arc leaving $U$, that is, there exists no arc $(x, y)$ with $x \in U$ and $y \in V(G)\backslash U$. A bipartition (or *split*) of $V(G)$ into nonempty sets $A$ and $B$, denoted $A|B$, is called

- *Type I* if both $A$ and $B$ are sink sets (i.e. there is no arc from any element in $A$ to any element in $B$ or vice versa);
- *Type II* if either $A$ or $B$ (but not both) is a sink set; and
- *Type III* if for all $x \in A$, $y \in B$ $(x, y)$ is an arc in $G$ if and only if $(y, x)$ is an arc in $G$.

We say that $A|B$ is a *typed* split of $G$ if it is a split of Type I, II or III.

For a collection $\mathcal{B}$ of binary level-1 binets on $X$, we introduce the digraph $D(\mathcal{B})$ with vertex set $X$ and $(x, y)$ being an arc in $D(\mathcal{B})$ if $T(x, y) \in \mathcal{B}$ or $R(x; y) \in \mathcal{B}$. See Fig. 4 for an example.

The following two lemmas show important properties of typed splits that will be used to establish Theorems 2 and 3.

**Lemma 2** *Suppose that $\mathcal{B}$ and $\mathcal{B}'$ are two thin collections of binary level-1 binets on $X$ with $\mathcal{B} \subseteq \mathcal{B}'$. Then each typed split of $D\left(\mathcal{B}'\right)$ is a typed split of $D(\mathcal{B})$.*

*Proof* Suppose that $A|B$ is a typed split of $D\left(\mathcal{B}'\right)$. If $A|B$ is of Type I in $D\left(\mathcal{B}'\right)$, then it is of Type I in $D(\mathcal{B})$ since $D(\mathcal{B})$ is a subgraph of $D\left(\mathcal{B}'\right)$. Similarly, if $A|B$ is of Type II in $D\left(\mathcal{B}'\right)$, then it is of Type I or II in $D(\mathcal{B})$. If $A|B$ is of Type III in $D\left(\mathcal{B}'\right)$ then (since $\mathcal{B}'$ is thin) any binet on $x$ and $y$ with $x \in A$ and $y \in B$ is $T(x, y)$. Therefore, $A|B$ is of Type I or III in $D(\mathcal{B})$. □

**Lemma 3** *Suppose that $\mathcal{B}$ is a thin collection of binary level-1 binets on $X$. If $\mathcal{B}$ is displayed by a binary network, then $D(\mathcal{B})$ has a typed split.*

*Proof* Suppose that $\mathcal{B}$ is displayed by a binary network. Then $\mathcal{B}$ is displayed by a binary recoverable network $N$. Let $\mathcal{B}'$ be the set of binary level-1 binets contained in $\mathcal{B}(N)$. Then we have $\mathcal{B} \subseteq \mathcal{B}' \subseteq \mathcal{B}(N)$. By Lemma 2, it suffices to show that $D\left(\mathcal{B}'\right)$ has a typed split.

Consider the root $\rho$ of $N$, which is equal to $\text{LSA}(X)$ since $N$ is recoverable. Denote the two children of $\rho$ by $u_1$ and $u_2$. We consider two cases.

The first case is that at least one arc incident with $\rho$ is a cut arc. Then the other arc incident with $\rho$ is also a cut arc. Then let $A = \mathcal{C}(u_1)$ and $B = \mathcal{C}(u_2)$. Note that $A|B$ is a split because neither $A$ nor $B$ is empty. In addition, for all $x \in A$, $y \in B$ we have $N|_{\{x,y\}} = T(x, y)$ and hence $A|B$ is a Type III split with respect to $D\left(\mathcal{B}'\right)$.

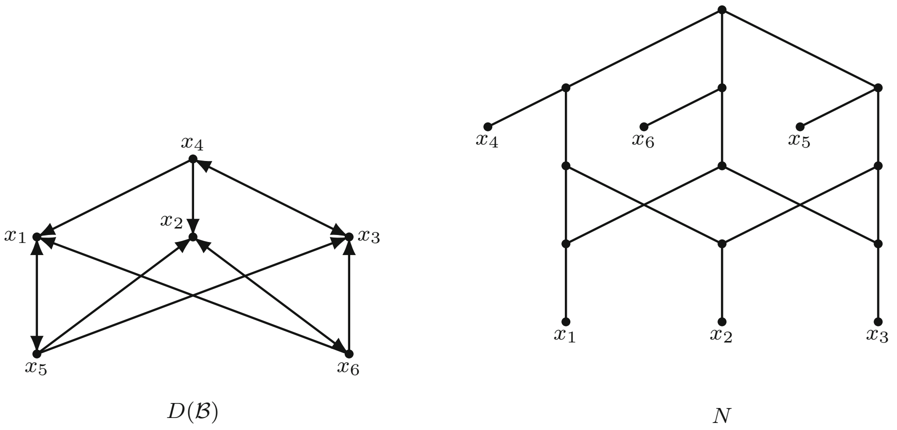


**Fig. 4** A digraph $D(\mathcal{B})$ representing the set of binets $\mathcal{B} = \{T\,(x_4, x_3)\,, R\,(x_4; x_1)\,, R\,(x_4; x_2)\,, T\,(x_5, x_1)\,, R\,(x_5; x_2), R\,(x_5; x_3)\,, T\,(x_6, x_2)\,, R\,(x_6; x_1)\,, R\,(x_6; x_3)\}$ and a network $N$ displaying $\mathcal{B}$. The direction of the arcs in $N$ is downward, and omitted

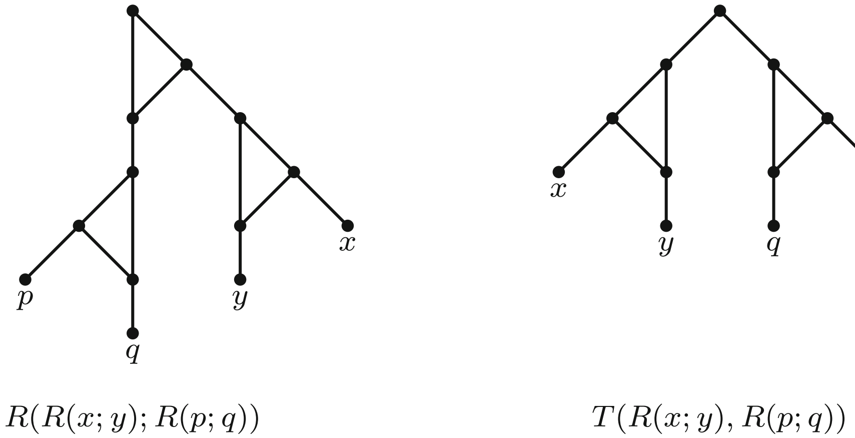
In the second case, both arcs incident with $\rho$ are not cut arcs. Hence, the root $\rho$ is contained in a non-trivial biconnected component $H$ containing $u_1$ and $u_2$. By Corollary 1, there exists a lowest vertex $v$ in $H$ with two arc-disjoint paths $P_1$, $P_2$ from $\rho$ to $v$. Since $v$ is a lowest vertex in $H$, we know that $v$ is a reticulation vertex and the arc leaving $v$ is a cut arc. Let $B = \mathcal{C}(v)$ and $A = X \backslash B$. Then $B$ is clearly nonempty. In addition, $A$ is nonempty, as otherwise $\text{LSA}(X) \preceq v$, a contradiction to the fact that $\text{LSA}(X) = \rho$ (as $N$ is recoverable). Therefore, $A|B$ is a split.

Consider $x \in A$ and $y \in B$ and the subnetwork $N|_{\{x,y\}}$. There is at least one directed path from $\rho$ to $x$, and each such path contains at least one arc of $P_1$ or $P_2$. Hence, in the process of obtaining $N|_{\{x,y\}}$ from $N$, the paths $P_1$, $P_2$ do not become parallel arcs. Therefore, $N|_{\{x,y\}}$ contains two arc-disjoint paths from $\rho$ to $v$ and we can conclude that $N|_{\{x,y\}} \neq T(x, y)$.

Therefore, if $N|_{\{x,y\}} \in \mathcal{B}^*$, that is, $N|_{\{x,y\}}$ is level-1, then $N|_{\{x,y\}} = R(x; y)$. This implies that there is no arc $(y, x)$. Therefore, $A|B$ is a Type I or Type II split of $D\,(\mathcal{B}^*)$. □

Note that the condition that $\mathcal{B}$ is displayed by a binary network in the above lemma can not be weakened to that $\mathcal{B}$ is displayed by a network. For example, consider the binet collection $\mathcal{B}$ and network $N$ in Fig. 4. Although network $N$ displays $\mathcal{B}$, digraph $D\,(\mathcal{B})$ has no typed split (as can be easily checked).

We now introduce two operations, which can be used to combine two phylogenetic networks into a new one. Suppose that $N_1$ and $N_2$ are two phylogenetic networks with disjoint leaf sets. Let $T\,(N_1, N_2)$ be the phylogenetic network obtained from $N_1$ and $N_2$ by adding a new vertex $v$ and two arcs from $v$ to the roots of $N_1$ and $N_2$. In addition, the network $R(N_1; N_2)$ is obtained by taking a binet $R\,(y_1; y_2)$, with $y_1, y_2 \notin L\,(N_1) \cup L\,(N_2)$, and replacing $y_i$ by the root of $N_i$, for $i = 1, 2$. See Fig. 5 for examples.



$R(R(x;y);R(p;q))$ $T(R(x;y),R(p;q))$

**Fig. 5** Examples of two networks built recursively by using the operations introduced in the text

For a binet set $\mathcal{B}$ on $X$ and a subset $A \subseteq X$, we define

$$\mathcal{B}|_A = \{S \in \mathcal{B} : L(S) \subseteq A\}.$$

The next theorem can be used to determine in polynomial time whether a collection of binary level-1 binets is displayed by some binary level-1 network. See Sect. 6 for more details.

**Theorem 2** *Suppose that $\mathcal{B}$ is a thin collection of binary level-1 binets on $X$. If there exists a typed split $A|B$ of $D(\mathcal{B})$ such that $\mathcal{B}|_A$ and $\mathcal{B}|_B$ are both displayed by some binary level-1 network, then $\mathcal{B}$ is displayed by a binary level-1 network. Moreover, if $\mathcal{B}$ is displayed by a binary level-1 network, then there exists at least one typed split of $D(\mathcal{B})$ and, for each typed split $A|B$ of $D(\mathcal{B})$, $\mathcal{B}|_A$ and $\mathcal{B}|_B$ are both displayed by some binary level-1 network.*

*Proof* First suppose that there exists a typed split $A|B$ of $D(\mathcal{B})$ such that $\mathcal{B}|_A$ and $\mathcal{B}|_B$ are displayed by binary level-1 networks $N_A$ and $N_B$, respectively.

If $A|B$ is a Type I or Type III split of $D(\mathcal{B})$, then consider the network $N = T(N_A, N_B)$. Then $N$ is a binary level-1 phylogenetic network on $X$ and

$$\mathcal{B} \subseteq \{T(x, y) : x \in A, y \in B\} \cup \mathcal{B}(N_A) \cup \mathcal{B}(N_B) = \mathcal{B}(N),$$

and so $\mathcal{B}$ is displayed by $N$.

If $A|B$ is a Type II split of $D(\mathcal{B})$, then without loss of generality we may assume that $B$ is a sink set in $D(\mathcal{B})$. Now consider the network $N = R(N_A; N_B)$. Then $N$ is a binary level-1 phylogenetic network on $X$ and

$$\mathcal{B} \subseteq \{R(x; y) : x \in A, y \in B\} \cup \mathcal{B}(N_A) \cup \mathcal{B}(N_B) = \mathcal{B}(N),$$

and so $\mathcal{B}$ is displayed by $N$.

Now suppose that $\mathcal{B}$ is displayed by a binary level-1 network $N$. By Lemma 3, there exists a typed split $A|B$ of $D(\mathcal{B})$. Then $\mathcal{B}|_A \subseteq \mathcal{B}(N|_A)$ and $\mathcal{B}|_B \subseteq \mathcal{B}(N|_B)$. □

We now prove the main result of this section.

**Theorem 3** *Suppose that $\mathcal{B}$ is a thin collection of binary level-1 binets on $X$. Then $\mathcal{B}$ is displayed by a binary level-1 network if and only if it is displayed by a binary network.*

*Proof* Suppose that $\mathcal{B}$ is displayed by a binary network. We claim that $\mathcal{B}$ is also displayed by a binary level-1 network. We shall establish this claim by induction on $|X|$.

If $|X| = 2$, then $\mathcal{B}$ contains at most one binet, which has leaf set $X$. Therefore we know that $\mathcal{B}$ is displayed by a binary level-1 network.

Now assume that $|X| > 2$, and the claim holds for all sets $X'$ with $2 \leq |X'| < |X|$. Let $N$ be a binary network on $X$ with $\mathcal{B} \subseteq \mathcal{B}(N)$. By Lemma 3, there exists a typed split $A|B$ of $D(\mathcal{B})$. Note that $\mathcal{B}|_A \subseteq \mathcal{B}(N|_A)$ and $\mathcal{B}|_B \subseteq \mathcal{B}(N|_B)$. Therefore, by induction, each of $\mathcal{B}|_A$ and $\mathcal{B}|_B$ is displayed by a binary level-1 network. By Theorem 2, it follows that $\mathcal{B}$ is displayed by a binary level-1 network. □

## 5 Binets Determine the Number of Reticulations of a Binary Level-1 Network

In this section we show that, although the collection of binets displayed by a level-1 network does not necessarily determine the network (see Fig. 1), it does in fact determine the number of reticulations in the network. We begin by showing that it suffices to consider level-1 networks in which all cycles (in the underlying undirected graph) have length 3.

First, we introduce some further notation. A *semi-cycle* $C$ of an acyclic-directed graph is the union of two non-identical, internally vertex-disjoint, directed paths from $s$ to $t$, with $s = s(C)$ and $t = t(C)$ two distinct vertices that are referred to as the *source* and *terminal* of $C$, respectively. The *length* of a semi-cycle is the number of distinct vertices that it contains.

We now show that we may restrict to networks in which all semi-cycles have length 3.

**Lemma 4** *If $N$ is a binary level-1 network, then there exists a binary level-1 network $N'$ in which every semi-cycle has length 3, such that $\mathcal{B}\left(N'\right) = \mathcal{B}(N)$ and $N$ and $N'$ have the same number of reticulation vertices.*

*Proof* Consider a semi-cycle of $N$ with source $s$ and terminal $t$ and length at least 4. Let $(u_1, v_1)$, ..., $(u_k, v_k)$, $(t, w)$ be the arcs leaving the semi-cycle. Then $k \geq 2$. Let $N^*$ be a network obtained from a binary tree on $\{v_1, \ldots, v_k\}$ by replacing $v_i$ by the subgraph of $N$ rooted at $v_i$, for $i = 1, \ldots, k$. Let $N_w$ be the subgraph of $N$ rooted at $w$. Then we construct $N'$ from $N$ by replacing the subgraph of $N$ rooted at $s$ by the network $R(N^*; N_w)$. It is straightforward to see that $N'$ is a binary level-1 network with the required properties. □

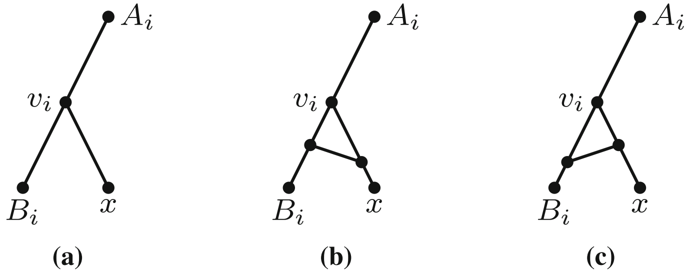


**Fig. 6** The three cases for the location of $x$ in each of the networks $N_1$ and $N_2$ in the proof of Theorem 4. **a** The parent of $x$ is not in a semi-cycle. **b** The parent of $x$ is the terminal of a semi-cycle. **c** The parent of $x$ is a non-terminal non-source vertex of a semi-cycle

We now establish the main result of this section.

**Theorem 4** *If $N_1$ and $N_2$ are rooted binary level-1 phylogenetic networks on $X$ with $\mathcal{B}(N_1) = \mathcal{B}(N_2)$ then $N_1$ and $N_2$ have the same number of reticulation vertices.*

*Proof* The proof is by induction on the number of leaves $|X|$. The induction basis for $|X| = 2$ is clear. Now suppose that $N_1$ and $N_2$ are two non-isomorphic rooted binary level-1 phylogenetic networks on $|X| \geq 3$ with $\mathcal{B}(N_1) = \mathcal{B}(N_2)$ but with different numbers of reticulation vertices. We add an out-degree-1 root to each of $N_1$ and $N_2$ with an arc to the original root. By Lemma 4, we may assume that all semi-cycles in $N_1$ and $N_2$ have length 3.

Choose an arbitrary leaf $x \in X$ and let $X' = X \setminus \{x\}$. Let $N_1'$ and $N_2'$ be the networks obtained from $N_1|_{X'}$ and $N_2|_{X'}$, respectively, by adding an out-degree-1 root with an arc to the original root. Then $N_1'$ and $N_2'$ have the same number of reticulation vertices by induction.

Since all semi-cycles in $N_1$ and $N_2$ are assumed to have length 3, there are three cases for the location of $x$ in each of the networks $N_1, N_2$, illustrated in Fig. 6.

If the parent of $x$ is in a semi-cycle in $N_i$, let $v_i$ be the source of this semi-cycle, and let $v_i$ be the parent of $x$ otherwise. Let $B_i := \mathcal{C}(v_i)\setminus\{x\}$ and $A_i := X'\setminus B_i$ (recall that $\mathcal{C}(v_i)$ denotes the cluster of $v_i$).

We now consider the different ways in which we could add $x$ to both networks. Since $N_1$ and $N_2$ have different numbers of reticulation vertices, there are two cases to consider (after eliminating symmetric cases), as illustrated in Fig. 7.

The first case is that the parent of $x$ is not in a semi-cycle in $N_1$ but is the terminal of a semi-cycle in $N_2$. First suppose that $B_1 \cap B_2 \neq \emptyset$. Then choose an arbitrary vertex $y \in B_1 \cap B_2$. Then $N_1|_{\{x,y\}} = T(x, y)$ while $N_2|_{\{x,y\}} = R(y; x)$, a contradiction. Hence, we may assume that $B_1 \cap B_2 = \emptyset$. Then $B_1 = A_2$ and $B_2 = A_1$. Clearly, $B_1, B_2 \neq \emptyset$. Take $y \in B_1 = A_2$ and $z \in B_2 = A_1$. Then $N_1|_{\{x,y\}} = T(x, y)$ and hence $N_2|_{\{x,y\}} = T(x, y)$, from which we can deduce that $N_2|_{\{z,y\}} = T(z, y)$. In addition, $N_2|_{\{z,x\}} = R(z; x)$ and hence $N_1|_{\{z,x\}} = R(z; x)$, from which we can deduce that $N_1|_{\{z,y\}} = R(z; y)$. This leads to a contradiction since $N_2|_{\{z,y\}} = T(z, y)$.

The second case is that the parent of $x$ is not in a semi-cycle in $N_1$ but is the non-terminal non-source vertex of a semi-cycle in $N_2$. First suppose that $B_1 \cap B_2 \neq \emptyset$. Then choose an arbitrary vertex $y \in B_1 \cap B_2$. Then $N_1|_{\{x,y\}} = T(x, y)$ while $N_2|_{\{x,y\}} =$

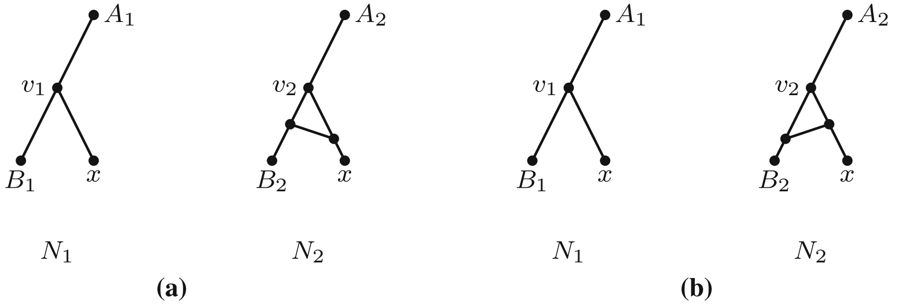


**Fig. 7** The two possible ways to add leaf $x$ to $N_1'$ and $N_2'$ in the proof of Theorem 4 such that the obtained networks $N_1$ and $N_2$ have different numbers of reticulation vertices. **a** Case 1: the parent of $x$ is not in a semi-cycle in $N_1$ but is the terminal of a semi-cycle in $N_2$. **b** Case 2: the parent of $x$ is not in a semi-cycle in $N_1$ but is the non-terminal non-source vertex of a semi-cycle in $N_2$

$R(x; y)$, a contradiction. Hence, we may assume that $B_1 \cap B_2 = \emptyset$. Then, as in the previous case, $B_1 = A_2 \neq \emptyset$ and $A_1 = B_2 \neq \emptyset$. Take $y \in B_1 = A_2$ and $z \in B_2 = A_1$. Then, similar to the previous case, $N_1|_{\{x,y\}} = T(x, y)$ and hence $N_2|_{\{x,y\}} = T(x, y)$, from which we can deduce that $N_2|_{\{z,y\}} = T(z, y)$. In addition, $N_2|_{\{z,x\}} = R(x; z)$ and hence $N_1|_{\{z,x\}} = R(x; z)$, from which we can deduce that $N_1|_{\{z,y\}} = R(y; z)$. This again leads to a contradiction since $N_2|_{\{z,y\}} = T(z, y)$. □

## 6 Complexity of Binet Compatibility

A direct consequence of Theorem 2 is that there exists a simple polynomial-time algorithm to decide whether there exists a binary level-1 network displaying a given collection $\mathcal{B}$ of binary level-1 binets (see Huber et al. 2015 for a related algorithm). In particular, a sink set of $D(\mathcal{B})$ can be found in polynomial time by computing the strongly connected components of $D(\mathcal{B})$ (Tarjan 1972) and checking for each of them whether it is a sink set. This can be used to find a typed split, if it exists. If such a split does not exist, then $\mathcal{B}$ is not compatible. Otherwise, we can try to construct networks for $\mathcal{B}|_A$ and $\mathcal{B}|_B$ recursively and combine them as described in the proof of Theorem 2. This algorithm is similar to the Aho algorithm for deciding whether a set of rooted trees can be displayed by some rooted tree (Aho et al. 1981).

From Theorem 3, it now follows that the following problem can also be solved in polynomial time.

**Binet compatibility** (BC)
**Input:** a set $\mathcal{B}$ of binary level-1 binets.
**Question:** is $\mathcal{B}$ compatible, i.e. does there exist a binary network $N$ with $\mathcal{B} \subseteq \mathcal{B}(N)$?

We show now that the assumption that all binets in $\mathcal{B}$ are binary and level-1 is essential. Indeed, for general binets, the compatibility problem is at least as hard as the well-known graph isomorphism problem (GI) (Goldberg 2003; Zemlyachenko et al. 1985), which is not known to be solvable in polynomial time. This is even true when the given binet set is thin (contains at most one binet for each pair of leaves).

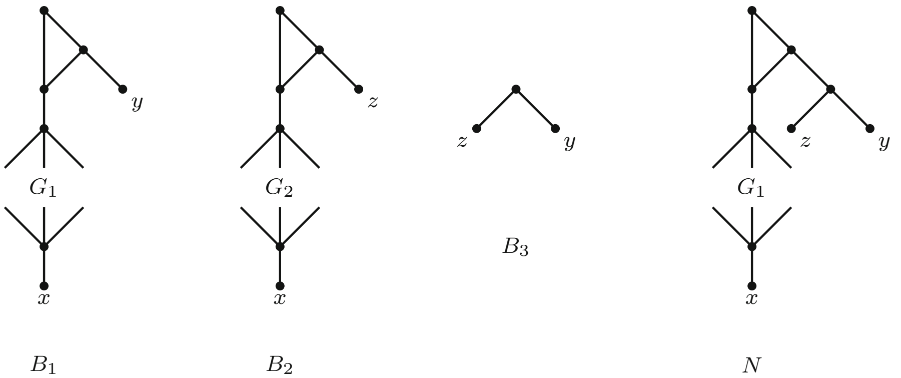


**Fig. 8** The three binets constructed in the proof of Theorem 5, and a network $N$ that displays all three binets if the digraphs $G_1$ and $G_2$ are isomorphic

**Theorem 5** *Deciding whether there exists a phylogenetic network displaying a given thin set $\mathcal{B}$ of binets is GI-hard.*

*Proof* We reduce from DAG-isomorphism, which is known to be GI-complete Zemlyachenko et al. (1985). Let $G_1, G_2$ be two directed acyclic graphs, which form an instance of the DAG-isomorphism problem. For $i = 1, 2$, we add vertices $\rho_i, u_i, v_i, w_i, r_i$, a new leaf labelled $x$, an arc from $w_i$ to each in-degree-0 vertex of $G_i$ and from each out-degree-0 vertex of $G_i$ to $r_i$ and arcs $(\rho_i, u_i)$, $(u_i, v_i)$, $(\rho_i, v_i)$, $(v_i, w_i)$ and $(r_i, x)$. In $G_1$, we add a new leaf labelled $y$ and an arc $(u_1, y)$. In $G_2$, we add a new leaf labelled $z$ and an arc $(u_2, z)$. We have thus transformed $G_1$ into a binet $B_1$ and $G_2$ into a binet $B_2$. The third binet is $B_3 = T(y, z)$. See Fig. 8 for an illustration.

We claim that $G_1$ and $G_2$ are isomorphic if and only if there exists a network displaying $B_1$, $B_2$ and $B_3$.

First assume that $G_1$ and $G_2$ are isomorphic. Then we can construct a network displaying $B_1$, $B_2$ and $B_3$ as follows. Take $B_1$ and subdivide the arc $(u_1, y)$ by a new vertex $u'_1$ and add leaf $z$ with an arc $\left(u'_1, z\right)$. The obtained network clearly displays $B_1$ and $B_3$ and it also displays $B_2$ since $G_1$ and $G_2$ are isomorphic.

Now assume that there exists some network $N$ displaying $B_1$, $B_2$ and $B_3$. Then $N|_{\{x,y\}} = B_1$. Hence, $N$ contains a cycle (in the underlying undirected graph) containing a reticulation $v$, such that $x$ and the image of $G_1$ are below the arc leaving $v$, while $y$ is below some other arc leaving the cycle. Since $N|_{\{y,z\}} = T(y, z)$, leaf $z$ is not below $v$ in $N$. Therefore, deleting $v$, $x$ and the parent of $x$ from the subgraph of $N$ rooted at $v$ gives $G_1$.

Similarly, $N$ contains a cycle containing a reticulation $v'$, such that $x$ and the image of $G_2$ are below the arc leaving $v'$, while $z$ is below some other arc leaving the cycle. Since $N|_{\{y,z\}} = T(y, z)$, leaf $y$ is not below $v'$ in $N$. Therefore, deleting $v'$, $x$ and the parent of $x$ from the subgraph of $N$ rooted at $v'$ gives $G_2$.

Moreover, $v = v'$ since $N|_{\{y,z\}} = T(y, z)$. Hence, $G_1$ and $G_2$ are isomorphic. □

## 7 Maximum Binet Compatibility

If a collection of binets is not compatible, the question arises whether it is possible to find a largest compatible subset of the binets, in polynomial time. Here we show that this is unlikely to be the case. The decision version of this problem is defined as follows.

**Maximum Binet compatibility** (MBC)
**Input:** a set $\mathcal{B}$ of binary level-1 binets and an integer $k$.
**Question:** does there exist a compatible subset $\mathcal{B}'$ of $\mathcal{B}$ with $|\mathcal{B}'| \geq k$?

We now establish the complexity of this problem (see Theorem 6). Recall from Sect. 5 that $s(C)$ and $t(C)$ denote the source and terminal of a semi-cycle $C$, respectively.

**Lemma 5** *If the binet $R(x; y)$ is displayed by a binary level-1 network $N$, then $\textsc{lsa}(x, y)$ is the source of a semi-cycle $C$ in $N$. In addition, $y$ is below $t(C)$ and $x$ is not below $t(C)$.*

*Proof* Let $u = \textsc{lsa}(x, y)$. Note that $u$ is not a reticulation vertex, as otherwise the child of $u$ would be a stable ancestor of $x$ and $y$ that is below $u$. Hence, $u$ has two children, denoted by $u_1$ and $u_2$.

Observe that neither $(u, u_1)$ nor $(u, u_2)$ is a cut arc, since otherwise we would have $N|_{\{x,y\}} = T(x, y)$, while by the assumption of the lemma $N|_{\{x,y\}} = R(x; y)$. Hence, $u$ is the source of a semi-cycle $C$. Let $v := t(C)$ be the terminal of $C$. If neither $x$ nor $y$ is below $v$, then $N|_{\{x,y\}} = T(x, y)$, a contradiction. If both $x$ and $y$ are below $v$, then $v$ is a stable ancestor of $x$ and $y$, a contradiction to $\textsc{lsa}(x, y) = u$. Therefore, precisely one of $x$ and $y$ is below $v$. If $x$ is below $v$ and $y$ is not, then $N|_{\{x,y\}} = R(y; x)$, a contradiction. Therefore, $y$ is below $v$ and $x$ is not. □

In view of the last lemma, for each binet $R(x; y) = N|_{\{x,y\}}$, there exists a unique semi-cycle $C_N(x; y)$ containing $\textsc{lsa}(x, y)$.

**Lemma 6** *If the two binets $R(x; y)$ and $R(y; z)$ are both displayed by a binary level-1 network $N$, then*

$$s\,(C_N(y; z)) \prec t\,(C_N(x; y))\,.$$

*Proof* Let $C_1 = C_N(x; y)$ and $C_2 = C_N(y; z)$. By Lemma 5, $y \prec t\,(C_1)$ but $y$ is not below $t\,(C_2)$, from which we know that $C_1 \neq C_2$. Since $s\,(C_1)$ and $s\,(C_2)$ are stable ancestors of $y$ in view of Lemma 5, we have either $s\,(C_1) \prec s\,(C_2)$ or $s\,(C_2) \prec s\,(C_1)$ but not both.

Note that if $s\,(C_1) \prec s\,(C_2)$, then $s\,(C_1) \prec t\,(C_2)$ and hence $y \prec s\,(C_1) \prec t\,(C_2)$, a contradiction. Thus $s\,(C_2) \prec s\,(C_1)$, from which it follows that $s\,(C_2) \prec t\,(C_1)$. □

Given a digraph $G$, let $\mathcal{R}(G)$ be the collection of binets $\{R(x; y) \mid (x, y) \in E(G)\}$ induced by $G$. Note that $\mathcal{R}(G)$ is a binet set on $V(G)$, i.e. the leaves of the binets in $\mathcal{R}(G)$ correspond to the vertices of $G$.

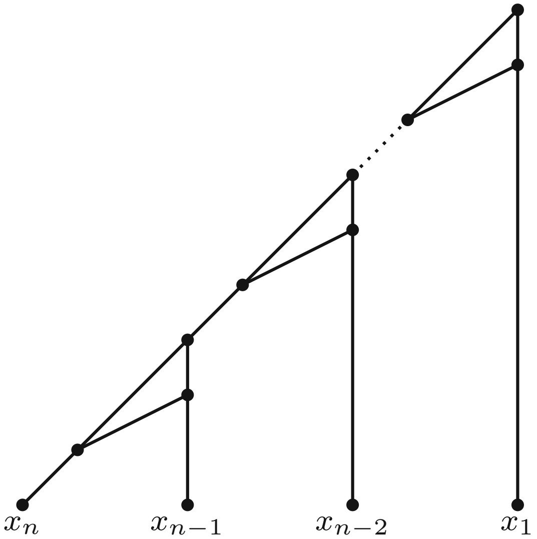


**Fig. 9** A level-1 network $N_*$ on $X = \{x_1, \dots, x_n\}$

**Proposition 1** *Let $G$ be a digraph. Then $G$ is acyclic if and only if $\mathcal{R}(G)$ is compatible.*

*Proof* Let $n = |X|$, with $X$ the vertex set of $G$. Suppose first that $G$ is acyclic, then there exists a topological sorting of $G$, that is, the vertices of $G$ can be ordered as $x_1, \dots, x_n$ so that $\left(x_i, x_j\right) \in E(G)$ implies $i < j$. Hence, the network $N_*$ in Fig. 9 displays $\mathcal{R}(G)$ since $N_*$ displays each binet $R\left(x_i; x_j\right)$ with $i < j$.

Conversely, suppose that $\mathcal{R}(G)$ is compatible. By Theorem 3, there exists a binary level-1 network $N$ with $\mathcal{R}(G) \subseteq \mathcal{B}(N)$. It remains to show that $G$ is acyclic. If not, then there exists a directed cycle $(x_1, x_2, \dots, x_m)$ for some $m \geq 3$. Denote $x_{m+1} = x_1$. In view of Lemma 5, let $C_i = C_N(x_i; x_{i+1})$ be the semi-cycle in $N$ containing LSA $(x_i, x_{i+1})$ for $1 \leq i \leq m$. Then Lemma 5 implies $x_1 \prec s(C_m)$ and that $x_1$ is not below $t(C_1)$. On the other hand, by Lemma 6 we have

$$s(C_m) \prec t(C_{m-1}) \prec s(C_{m-1}) \prec \dots \prec s(C_2) \prec t(C_1).$$

Together with $x_1 \prec s(C_m)$, it follows that $x_1 \prec t(C_1)$, a contradiction. □

A set of binets $\mathcal{B}$ on $X$ is said to be *dense* if for each pair of distinct elements $x$ and $y$ in $X$, there exists precisely one binet on $\{x, y\}$ in $\mathcal{B}$. Hence, a dense set of binets is always thin.

**Theorem 6** *The problem MBC is NP-complete, even if the given set of binets is dense.*

*Proof* We reduce from the NP-hard problem feedback arc set in tournaments (FAST) (Alon 2006; Charbit et al. 2007), which is defined as follows. Given a tournament, i.e. a digraph $G = (V, E)$ with either $(a, b) \in E$ or $(b, a) \in E$ (but not both) for each pair of distinct elements $a$ and $b$ in $V$, and given a positive integer $k'$, does there exist a subset $F \subseteq E$ of at most $k'$ arcs whose removal makes $G$ acyclic. If such an arc set exists, then we call it a *feedback arc set* of $G$.

The reduction is as follows. For each instance $(G, k')$ of FAST, consider the corresponding instance $(\mathcal{R}(G), k)$ of MBC with $k = |\mathcal{R}(G)| - k'$. Since the set $\mathcal{R}(G)$ of binets induced by $G$ can be constructed in polynomial time, it suffices to show that $G$ contains a feedback arc set with size at most $k'$ if and only if there exists a compatible subset of $\mathcal{R}(G)$ of size at least $k$.

First assume that there exists a feedback arc set $E'$ of $G$ with size at most $k'$. That is, $|E'| \leq k'$, and the digraph $G^*$ obtained from $G$ by deleting the arcs in $E'$ is acyclic. Consider the set of binets $\mathcal{B}' = \{R(x; y) : (x, y) \in E \backslash E'\}$. This set contains at least $k$ binets. In addition, since $\mathcal{B}' = \mathcal{R}(G^*)$, it follows by Proposition 1 that $\mathcal{B}'$ is compatible.

Now assume that there exists a compatible binet set $\mathcal{B}' \subseteq \mathcal{R}(G)$ with $|\mathcal{B}'| \geq k$. Consider the set $E' = \{(x, y) : R(x, y) \in \mathcal{R}(G) \backslash \mathcal{B}'\}$ of arcs of $G$. Then by Proposition 1, it follows that $E'$ is a feedback arc set. Moreover, $|E'| \leq k'$, which completes the proof. □

We complete the section by showing that there exists a polynomial-time 1/3-approximation algorithm for the MBC problem, which follows directly from the next theorem and its proof.

**Theorem 7** *Suppose that $\mathcal{B}$ is a set of binary level-1 binets on $X$. Then there exists a binary level-1 network $N$ such that $|\mathcal{B}(N) \cap \mathcal{B}| \geq |\mathcal{B}|/3$.*

*Proof* If at least a third of the binets in $\mathcal{B}$ are tree type, then take $N$ to be any binary tree on $X$ and we are done. Hence we may assume that at least two thirds of the binets are reticulate type.

Impose an arbitrary ordering on the elements in $X$, that is, write $X = \{x_1, \ldots, x_n\}$. Let $\mathcal{B}_1 = \mathcal{B} \cap \{R(x_i; x_j) : 1 \leq i < j \leq n\}$ and $\mathcal{B}_2 = \mathcal{B} \cap \{R(x_j; x_i) : 1 \leq i < j \leq n\}$. Without loss of generality, we may assume that $|\mathcal{B}_1| \geq |\mathcal{B}_2|$ (as the other case can be established in a similar way). Since at least two thirds of the binets are reticulate type, and each of those is contained in either $\mathcal{B}_1$ or $\mathcal{B}_2$ (but not both), we know that $|\mathcal{B}_1| \geq |\mathcal{B}|/3$. Now consider the network $N_*$ in Fig. 9, then clearly we have $\mathcal{B}_1 \subseteq \mathcal{B}(N_*)$. Thus we have $|\mathcal{B}(N_*) \cap \mathcal{B}| \geq |\mathcal{B}_1| \geq |\mathcal{B}|/3$, from which the theorem follows. □

## 8 Discussion

In this paper we have developed some combinatorial results concerning collections of level-1 binets. Several interesting questions arise from these results. For example, we have shown that the collection of level-1 binets displayed by a binary phylogenetic network can be displayed by some level-1 network, but is there some canonical level-1 network that could be used to display such a collection? In addition, can we count the number of binary level-1 networks that display a dense compatible collection of binets? We have also seen that the collection of binets displayed by a binary level-1 network determine its reticulation number. Therefore it is natural to ask which properties of a phylogenetic network in general are determined by its binets?

We have also studied some algorithmic questions concerning binets. Concerning the maximum binet compatibilty problem, note that the constant $1/3$ is sharp in Theorem 7. For example, consider the binet collection $\{R(x; y), T(x, y), R(y; x)\}$. However, can a better bound be achieved by restricting to thin collections of binets, and can improved approximation algorithms also be found?

In another direction, it would be interesting to know whether similar results to those proven in this paper might hold for higher-level networks. For example, what can be said about properties of collections of level-2 binets, and does Theorem 4 hold also for higher-level networks? Also, we could try to generalize some of our results to $k$-nets, i.e. networks on $k$ leaves, $k \geq 2$. For example, does Theorem 3 hold for trinets? In general, it would be interesting to know what additional information the collection of $k$-nets displayed by a network might contain for $k \geq 3$. Note that it has been shown that trinets do not completely determine rooted networks in general Huber et al. (2015). However, do they determine properties of networks such as the number of reticulations?

Similarly, it would be interesting to extend some of our algorithmic results to higher-level networks and $k$-nets. For example, it is known that the compatibility problem is NP-complete for collections of level-1 trinets (Huber et al. 2015). However, to date the maximum trinet compatibility problem has not been studied.

Eventually, it is hoped that new results in these directions could be useful for developing novel methods to construct phylogenetic networks from higher-level networks and $k$-nets. For example, using our results it may be possible to develop approaches to build a consensus network for a collection of phylogenetic trees or networks. Note that consensus networks have already proven themselves useful in the unrooted setting, where they are used to summarize key features displayed by a collection of trees or networks (see e.g. Holland et al. 2004). A consensus method based on binets could work by breaking each of the given networks down into a collection of binets, and then developing methods to pool together the information contained in the resulting binets so as to construct some consensus network, or at least some constraints that any such network should satisfy. Note that similar approaches have been developed to build consensus trees for a collection of phylogenetic trees by breaking each of the trees down into a collection of triplets [see e.g. Bryant (2003, Sect. 2)]. Probably it would be of some interest to first consider how to construct a level-1 consensus network for a collection of level-1 networks by breaking each of them down into level-1 binets. This is already likely to be quite challenging in view of our result concerning NP-completeness of MBC.

## References

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 J Comput 10(3):405–421

Alon N (2006) Ranking tournaments. SIAM J Discret Math 20(1):137–142

Bapteste E, van Iersel LJJ, Janke A, Kelchner S, Kelk S, McInerney JO, Morrison DA, Nakhleh L, Steel M, Stougie L, Whitfield J (2013) Networks: expanding evolutionary thinking. Trends Genet 29(8):439–441

Bryant D (2003) A classification of consensus methods for phylogenetics. DIMACS Ser Discret Math Theor Comput Sci 61:163–184

Byrka J, Guillemot S, Jansson J (2010) New results on optimizing rooted triplets consistency. Discret Appl Math 158(11):1136–1147

Charbit P, Thomassé S, Yeo A (2007) The minimum feedback arc set problem is NP-hard for tournaments. Comb Probab Comput 16(01):1–4

Felsenstein J (2004) Inferring phylogenies. Sinauer Associates, Sunderland

Goldberg M (2003) The graph isomorphism problem. In: Gross JL, Yellen J (eds) Handbook of graph theory. CRC Press, Boca Raton, pp 68–78

Gusfield D (2014) ReCombinatorics—the algorithmics of ancestral recombination graphs and explicit phylogenetic networks. MIT Press, Cambridge

Holland B, Huber K, Moulton V, Lockhart P (2004) Using consensus networks to visualize contradictory evidence for species phylogeny. Mol Biol Evol 21(7):1459–1461

Huber K, van Iersel LJJ, Moulton V, Scornavacca C, Wu T (2015) Reconstructing phylogenetic level-1 networks from nondense binet and trinet sets. Algorithmica. doi:10.1007/s00453-015-0069-8

Huber K, van Iersel LJJ, Moulton V, Wu T (2015) How much information is needed to infer reticulate evolutionary histories? Syst Biol 64:102–111

Huber K, Moulton V, Wu T (2016) Closed sets in phylogenetic networks. Preprint

Huson D, Rupp R, Scornavacca C (2010) Phylogenetic networks: concepts, algorithms and applications. Cambridge University Press, Cambridge

Morrison D (2011) An introduction to phylogenetic networks. RJR Productions, Uppsala, Sweden

Oldman J, Wu T, van Iersel LJJ, Moulton V (2016) Trilonet: piecing together small networks to reconstruct reticulate evolutionary histories. Mol Biol Evol 33(8):2151–2162

Tarjan R (1972) Depth-first search and linear graph algorithms. SIAM J Comput 1(2):146–160

Zemlyachenko VN, Korneenko NM, Tyshkevich RI (1985) Graph isomorphism problem. J Sov Math 29(4):1426–1481