Is this network proper forest-based?

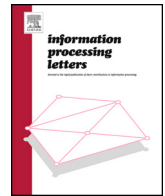Huber, Katharina T; van Iersel, Leo; Moulton, Vincent; Scholz, Guillaume E.

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Is this network proper forest-based?

Katharina T. Huber [b], Leo van Iersel [a],[*],[1], Vincent Moulton [b], Guillaume E. Scholz [c]

[a] *Delft Institute of Applied Mathematics, Delft University of Technology, Mekelweg 4, Delft, 2628 CD, the Netherlands*
[b] *School of Computing Sciences, University of East Anglia, NR4 7TJ, Norwich, United Kingdom*
[c] *Bioinformatics Group, Department of Computer Science and Interdisciplinary Center for Bioinformatics, Universität Leipzig, Leipzig, Germany*

ARTICLE INFO

ABSTRACT

In evolutionary biology, networks are becoming increasingly used to represent evolutionary histories for species that have undergone non-treelike or reticulate evolution. Such networks are essentially directed acyclic graphs with a leaf set that corresponds to a collection of species, and in which non-leaf vertices with indegree 1 correspond to speciation events and vertices with indegree greater than 1 correspond to reticulate events such as gene transfer. Recently forest-based networks have been introduced, which are essentially (multi-rooted) networks that can be formed by adding some arcs to a collection of phylogenetic trees (or phylogenetic forest), where each arc is added in such a way that its ends always lie in two different trees in the forest. In this paper, we consider the complexity of deciding whether a given network is proper forest-based, that is, whether it can be formed by adding arcs to some underlying phylogenetic forest which contains the same number of trees as there are roots in the network. More specifically, we show that it is NP-complete to decide whether a tree-child network with m roots is proper forest-based, for each $m \geq 2$. Moreover, for binary networks the problem remains NP-complete when $m \geq 3$ but becomes polynomial-time solvable for $m = 2$. We also give a fixed parameter tractable (FPT) algorithm, with parameters the maximum outdegree of a vertex, the number of roots, and the number of indegree 2 vertices, for deciding if a semi-binary network is proper forest-based. A key element in proving our results is a new characterization for when a network with m roots is proper forest-based in terms of certain m-colorings.

1. Introduction

Recently, the concept of forest-based networks has been introduced within the area of phylogenetics [8]. Informally, a forest-based network is defined as follows (full definitions of the terms used in the introduction are given in the next section). A *phylogenetic tree* is a rooted tree whose leaf-set is a collection of taxa or species; a *phylogenetic forest* is a collection of leaf-disjoint phylogenetic trees. A *forest-based network* is a directed acyclic graph N that can be formed by adding a set of arcs to a phylogenetic forest F so that for each added arc, the end vertices of that arc lie in two different trees of F; N is *proper* if the number of roots of N is equal to the number of trees in F. For example, the network in Fig. 1 is proper forest-based. Forest-based networks can be regarded as a certain type of *phylogenetic network*, and are related to the intensively studied *tree-based networks* [6] (see e.g. [11,14] for recent reviews of phylogenetic networks, including more details concerning tree-based networks).

Forest-based networks arise in the study of reticulate evolutionary processes in which species exchange genetic information through processes such as introgression [13] and lateral gene transfer [8]. In the case of introgression, the phylogenetic trees underlying a forest-based network correspond to the evolutionary histories of different subgroups or lineages within a certain species. The arcs in between different trees then represent the past interchange of genetic material between these lineages. A well-studied example of this phenomenon is butterfly evolution, where the genetic material that is swapped between lineages influences wing patterns [1,16]. Fig. 1 shows a hypothetical example to illustrate this phenomenom. An application of a special type of forest-based network called an *overlaid species forest* for analyzing introgression in butterflies can be found in [13].
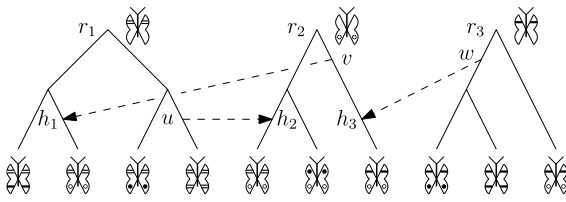
Fig. 1. A proper forest-based network N on ten leaves. Each of the three phylogenetic trees in the underlying forest represents a hypothetical butterfly lineage with main wing pattern indicated next to the root of the tree. The network N is the result of adding dashed arcs in between pairs of trees in the forest. Each added arc corresponds to some genetic material being introduced into a lineage from one of the others, which results in a wing pattern change for the descendants.

In this paper we consider the problem:

(P) Is a given network N proper forest-based?

Our main results are as follows. A network is *binary* if all vertices have indegree and outdegree at most 2 and all non-root vertices have overall degree 1 or 3. A network is *tree-child* if each non-leaf vertex has at least one child with indegree 1. For networks with v vertices, m roots, n leaves, r vertices with indegree at least 2, and maximum outdegree Δ, we shall prove that problem (P):

(R1) can be decided in $O(nr)$ time when N is restricted to be binary and tree-child and $m = 2$ (Theorem 6.2);

(R2) is NP-complete even when N is restricted to be binary and tree-child, for each fixed $m \geq 3$ (Theorem 6.1);

(R3) is NP-complete even when N is restricted to be tree-child, with maximum outdegree 2 and maximum indegree at most 3, for each fixed $m \geq 2$ (Theorems 4.1 and 6.1);

(R4) can be decided using an FPT algorithm with parameters r, m and Δ, which is linear in v, assuming the maximum indegree is 2 (Theorem 7.2).

Before proceeding, it is worth pointing out that recognition problems such as (P) frequently arise in the theory of phylogenetic networks. For example, the recognition of tree-based networks has been intensively studied and there are polynomial-time algorithms for deciding whether a phylogenetic network is tree-based (see e.g. [6,10]). Results are also known for tree-based *unrooted* phylogenetic networks (undirected analogues of phylogenetic networks) where, in contrast to the directed case, it is NP-complete to decide whether such a network is tree-based [4]. More recently, there has also been some interest in the recognition problem for other classes of networks, such as planar [12], orchard [15], and labellable phylogenetic networks [5], as well as the related problem of deciding how far away a given phylogenetic network is from being within a certain class [3]. Generally speaking, developing techniques for recognizing whether a network belongs to a specific class can be useful as it provides insights into the structure of the networks within that class.

We now briefly summarise the rest of the paper. In the next section, we present some formal definitions concerning networks. In Section 3, we derive a key characterization of proper forest-based networks (Lemma 3.1). In Section 4, we establish Statement (R3) by reducing from the SET-SPLITTING problem. Using colorings, in Section 5 we present an alternative proof for Statement (R3) in the special case that $m \geq 3$ by reducing from the GRAPH m-COLORABILTY problem. The construction that we use in this proof is then used again in Section 6 to prove that Statement (R2) holds. Using the concept of so-called omniextensions [8] we also prove Statement (R1). Finally, in Section 7, we prove Statement (R4), before concluding in Section 8 with a brief discussion of potential directions for future research.

2. Definitions

From now on, X is a finite set with $|X| \geq 2$. Suppose that N is a connected, directed acyclic graph (DAG) and $v \in V(N)$. Then v is a *leaf* if it has outdegree 0, a *root* if it has indegree 0, and a *reticulation* if it has indegree at least 2. If v is not a leaf, then v is an *internal vertex* of N. In particular, roots are internal vertices. We call an arc $a = (u, v)$ of N *internal* if u and v are both internal vertices of N. If v is an internal vertex that has indegree at most 1, then v is a *tree-vertex*.

For vertices $u, v \in V(N)$, we say that u is an *ancestor* of v if there exists a directed path from u to v in N. In this case, we say that v is a *descendant* of u. Note that each vertex is an ancestor and a descendant of itself.

We call N a *network (on X)* if it has leaf-set X (which we also denote by $L(N)$), each leaf has indegree 1, every root has outdegree at least 2, every reticulation has outdegree 1, and there is no vertex with indegree and outdegree 1. If N has m roots, we also call it an *m-network*.[2] For example, the network depicted in Fig. 1 is a 3-network. We say that two m-networks N and N' on X are *isomorphic* if there is a DAG isomorphism between N and N' which is the identity on X.

A network is *semi-binary* if all reticulations have indegree 2. It is *quasi-binary* if all tree-vertices have outdegree 2 and all reticulations have indegree 2 or 3. Note that a network is binary if it is quasi-binary and semi-binary. A network is *tree-child* if every internal vertex has a child that is a tree-vertex or a leaf.

A *phylogenetic network (on X)* is a network on X with exactly one root and a *phylogenetic tree (on X)* is a phylogenetic network on X with no reticulations. For technical reasons, we shall also call an isolated vertex v a phylogenetic tree (on $\{v\}$). Two distinct leaves x, y of a phylogenetic tree *form a cherry* if they share a parent. We denote such a cherry by $\{x, y\}$. A phylogenetic tree T on X a *caterpillar tree (on X)* if T has a unique cherry and every internal arc of T lies on the directed path from the root of T to the parent of the cherry.

A *phylogenetic forest (on X)* is a directed graph F whose connected components are phylogenetic trees and such that $X = \bigcup_{T \in F} L(T)$. For convenience, we will sometimes also call a non-empty set of pairwise leaf-set disjoint phylogenetic trees a phylogenetic forest.

Suppose $N = (V, A)$ is an m-network on X, some $m \geq 2$. Then N is *forest-based* if there exists a subset $A' \subseteq A$ such that $F' = (V, A')$ is a forest with leaf set X and each arc in $A - A'$ has end vertices that are in different trees of F'. We call F' a *subdivision forest* of N. The phylogenetic forest F obtained from F' by repeatedly suppressing any vertices of indegree and outdegree 1 and any outdegree 1 roots until this is no longer possible is a *base forest* of N. Note that, in particular, we can think of F as being embedded within N. A forest-based m-network is *proper forest-based* if it has a base forest that contains m trees – see [8,9] for more on this.

To illustrate, consider the network N depicted in Fig. 1. Clearly, N is forest-based since the forest F' obtained from N by removing the dashed arcs is a forest with $L(F') = L(N)$, and each dashed arc has end vertices in distinct trees of F'. Hence, F' is a subdivision forest of N. A base forest F of N is obtained from F' by suppressing the vertices u, v, w, h_1, h_2 and h_3, as all have indegree and outdegree 1 in F'. Since F contains 3 trees and N is a 3-network, N is proper forest-based.

3. Colorings and proper forest-based networks

Suppose G is a (simple) graph and $C \neq \emptyset$ is a finite set of colors. A surjective map $\sigma : V(G) \to C$ is a *$|C|$-coloring of G*. For $v \in V(G)$, the *color of v under σ*, or simply the *color of v* if σ is clear from the context, is $\sigma(v)$. A coloring σ of G is *proper* if $\sigma(x) \neq \sigma(y)$, for any two adjacent vertices $x, y \in V(G)$.

---

[2] Note that this is more general than the definition given in [8].

We now characterize proper forest-based networks in terms of colorings of their vertex sets.

Lemma 3.1. *Let N be an m-network on X, $m \geq 2$. Then N is proper forest-based if and only if there exists an m-coloring of N such that:*

(C1) Each non-root vertex of N has the same color as exactly one of its parents.

(C2) Each internal vertex of N has the same color as at least one of its children.

Proof. Suppose first that N is proper forest-based. Let F' be a corresponding subdivision forest of N. We claim that the map $\sigma : V(N) \to F'$ that assigns to each $v \in V(N)$ the tree in F' that contains v in its vertex set is an m-coloring of N that satisfies (C1) and (C2).

To see (C1), let v be a non-root vertex of N. Since the root of each tree of F' is a root of N, v is not the root of $\sigma(v)$. So at least one of the parents of v has the same color as v. If v has two or more parents satisfying this property, then v has indegree 2 or more in $\sigma(v)$, a contradiction as $\sigma(v)$ is a tree. Hence, (C1) holds.

To see (C2), let v be an internal vertex of N. Since $L(N) = L(F')$, v is not a leaf of $\sigma(v)$. Hence, there exists at least one child u of v in N such that $\sigma(u) = \sigma(v)$.

Conversely, suppose that there exists an m-coloring σ of $N = (V, A)$ satisfying (C1) and (C2). Let F' be the graph with vertex set V and arc set $A' = \{(u, v) \in A : \sigma(u) = \sigma(v)\}$. By (C1), no vertex in F' has indegree greater than 1, and by (C2), $L(N) = L(F')$. So F' is a subdivision forest of N. Since the end vertices of each arc in $A - A'$ have different colors, they appear in different trees of F'. Hence, N is proper forest-based. $\square$

Note that if σ is an m-coloring of an m-network N satisfying (C1) and (C2), then these two properties together with the fact that the image of σ has size m imply that no two roots of N have the same color under σ.

4. Tree-child networks

The main result in this section implies Statement (R3) for the case $m = 2$. To prove it, we shall reduce from the NP-complete SET-SPLITTING decision problem [7, page 221] which is defined as follows.

- Given some finite set X, $|X| \geq 3$, and a set C of size-3 subsets of Y, is there a bipartition $\{A, B\}$ of X such that, for all $S \in C$, $S \cap A \neq \emptyset$ and $S \cap B \neq \emptyset$?

Theorem 4.1. *For $m = 2$, the problem of deciding whether a quasi-binary tree-child m-network is proper forest-based is NP-complete.*

Proof. The problem is in the NP since, for each proper forest-based, quasi-binary, tree-child m-network, an m-coloring satisfying (C1) and (C2) serves as a certificate and (C1) and (C2) can be verified in polynomial time.

We shall prove NP-completeness by giving a reduction from SET-SPLITTING. Suppose that we are given a collection C of size-3 subsets of X. We create a 2-network as follows (see Fig. 2 where we illustrate the various constructions that we perform as part of this proof in terms of an example). Let $X' = X \cup \{\ell\}$ for some element $\ell \notin X$, and let T_1 and T_2 denote two isomorphic caterpillars trees on X' such that ℓ is a leaf in the unique cherry of T_1 (and therefore also of T_2). For all $x \in X$, we identify leaf x of T_1 with leaf x of T_2. The resulting network has $|X|$ reticulations. We call all vertices in the resulting DAG *Gen1 vertices*.

For all $x \in X$, we attach to x a path P_x of length $c(x) + 1$ via an arc (x, a_x), where a_x is the first vertex on P_x and $c(x)$ is the number of sets in C containing x. We then bijectively label for each $x \in X$ the internal vertices of P_x with the elements in C that contain x. We call all vertices

added during this step *Gen2 vertices*. Finally, for all $S \in C$ and all $x \in S$, we create a reticulation h_S with parents the vertices on P_x labeled by S and add a leaf ℓ_S to h_S by adding the arc (h_S, ℓ_S). We call vertices added during this step *Gen3 vertices*.

Let N denote the resulting DAG. By construction, N is an m-network on $L(N)$ that is quasi-binary and tree-child. We now show that there exists a solution to the SET-SPLITTING problem for X and C if and only if N is proper forest-based, which will complete the proof.

Suppose first that $\{A, B\}$ is a solution to the SET-SPLITTING problem for X and C. We define an m-coloring $\sigma : V(N) \to \{1, \ldots, m\}$ for N as follows. Let $v \in V(N)$. If v is a Gen1 tree-vertex, we set $\sigma(v) = 1$ if $v \in V(T_1)$, and $\sigma(v) = 2$ if $v \in V(T_2)$, if $v \in \{r_2, \ell_2, h_2\}$, and if $v \in \{r_m, \ell_m\}$ we put $\sigma(v) = m$. If $v \in V(P_x) \cup \{x\}$ for some $x \in X$, then we put $\sigma(v) = 1$ if $x \in A$ and $\sigma(x) = 2$ if $x \in B$. Finally, if $v \in \{h_S, \ell_S\}$ for some $S \in C$, then we put $\sigma(v) = 1$ if $|S \cap A| = 1$ and $\sigma(v) = 2$ if $|S \cap B| = 1$. Note that since $\{A, B\}$ is a solution to the SET-SPLITTING problem, precisely one of these equalities always holds. We claim that σ satisfies (C1) and (C2), which implies that N is proper forest-based.

To see that (C1) holds, note first that, by construction, all non-root vertices v have at least one parent u satisfying $\sigma(u) = \sigma(v)$. Suppose now that v is a reticulation. If v is a Gen1 vertex, then $\sigma(v) \in \{1, 2\}$ and v has exactly two parents. Calling them v_1 and v_2 we have $\sigma(v_1), \sigma(v_2) \in \{1, 2\}$ and $\sigma(v_1) \neq \sigma(v_2)$. Thus, exactly one of $\sigma(v_1) = \sigma(v)$ or $\sigma(v_2) = \sigma(v)$ holds. If v is a Gen3 vertex, then v has three parents v_1, v_2, v_3 and two of them must have the same color under σ. Without loss of generality assume that $\sigma(v_2) = \sigma(v_3)$. By definition, $\sigma(v) = \sigma(v_1)$ follows. Hence, (C1) holds.

To see that (C2) holds, let $v \in V(N) - L(N)$. If v is a Gen1 vertex that is not a reticulation, then v is an internal vertex of either T_1 or T_2. Since, for all $i \in \{1, 2\}$, T_i is a caterpillar tree and, by definition of σ, all vertices on the directed path from the root of T_i to ℓ have the same color under σ, (C2) follows. If v is a Gen2 vertex or a Gen1 reticulation then $v \in \{x\} \cup V(P_x)$, some $x \in X$. Since P_x is a directed path whose first vertex is adjacent with x, the definition of σ implies (C2) again. Finally, if v is a Gen3 vertex, then $v = h_S$, some $S \in C$. By definition, $\sigma(h_S) = \sigma(\ell_S)$. Hence, (C2) also holds.

Conversely, suppose that N is proper forest-based. By Lemma 3.1, there exists an m-coloring σ of N satisfying (C1) and (C2). Note that, by construction, the set of all Gen1 reticulations of N is X and also that every element in X is a descendant of both r_2 and the root ρ_1 of T_1. By (C1), it follows that either $\sigma(x) = \sigma(r_2)$ or $\sigma(x) = \sigma(\rho_1)$ holds for all $x \in X$. Let $A = \{x \in X : \sigma(x) = \sigma(\rho_1)\}$ and $B = \{x \in X : \sigma(x) = \sigma(r_2)\}$. Clearly, $\{A, B\}$ is a bipartition of X as $\sigma(r_2) \neq \sigma(\rho_1)$.

We claim that $\{A, B\}$ is a solution to the SET-SPLITTING problem. To see this, consider a set $S = \{x, y, z\} \in C$. By construction, h_S has three ancestors, all of which are Gen2 vertices that are a descendant of x, y and z, respectively. Since a Gen2 vertex is a vertex on P_w for some $w \in X$, (C1) implies that $\sigma(w) = \sigma(u)$, for all $u \in V(P_w)$. Moreover, by (C2), exactly one parent u of h_S satisfies $\sigma(u) = \sigma(h_S)$. Without loss of generality, we may assume that u is a descendant of x. Hence, $\sigma(x) \neq \sigma(y) = \sigma(z)$. By definition of A and B, it follows that $S \cap A \neq \emptyset$ and $S \cap B \neq \emptyset$. Since this holds for all $S \in C$, the claim follows. $\square$

5. Tree-child networks revisited

In this section, we prove a result using network colorings from which a weaker form of Statement (R3) also follows. We do this in part because the construction that we shall use in the proof will be used in the next section for establishing our results concerning binary, tree-child m-networks. We shall reduce from the GRAPH m-COLORABILTY decision problem for $m \geq 3$ ([7, page 190]) which is as follows.

- Given a (simple) graph G, does there exist a proper m-coloring of G?
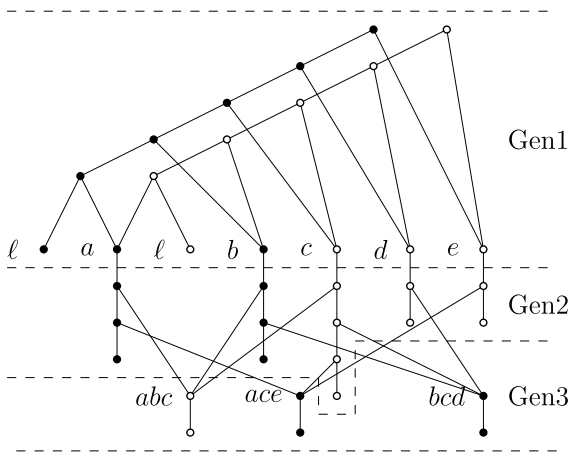
Fig. 2. The 2-network N for $X = \{a, b, c, d, e\}$ and $C = \{\{a, b, c\}, \{a, c, e\}, \{b, c, d\}\}$, as described in the proof of Theorem 4.1. Gen1, Gen2, and Gen3 vertices are indicated as vertices in a band labeled Gen1, Gen2, and Gen3, respectively. For clarity purposes, we have indicated the reticulation with the set in C its three parents correspond to and not the parents themselves. A 2-coloring $\sigma : V(N) \to \{\bullet, \circ\}$ associated to the solution $A = \{a, b\}$, $B = \{c, d, e\}$ of the SET-SPLITTING problem for (X, C).

Note that this problem can be solved in polynomial time for $m = 2$ but is NP-complete for each $m \geq 3$. Hence, the reduction below can only be used for $m \geq 3$.

Proposition 5.1. *For each fixed $m \geq 3$, the problem of determining whether a tree-child m-network is proper forest-based is NP-complete.*

Proof. Membership of NP can be argued in the same way as in the proof of Theorem 4.1. We prove NP-completeness by giving a reduction from GRAPH m-COLORABILTY.

Suppose that we are given a graph G with vertex set X. Then we construct an m-network N as follows (see Fig. 3 where we illustrate the various constructions performed in this proof in terms of an example). Let $T_1, \ldots, T_m$ denote m isomorphic caterpillars trees on a set Y with $|X| + 1$ leaves. Without loss of generality we may assume that $Y = X \cup \{\ell\}$ and that $\ell \notin X$ is a leaf in the unique cherry of T_1 (and therefore also of all T_i, $2 \leq i \leq m$). For all $x \in X$ and all $1 \leq i \leq m$, we identify the leaves x to obtain a vertex x of indegree m. We call all vertices in the resulting graph *Gen1 vertices*.

Denoting for all $x \in X$ the degree of x in G by $\deg_G(x)$, we attach a directed path P_x of length $\deg_G(x) + 1$ via an arc (x, a_x) to the first vertex a_x of P_x. We label each internal vertex of P_x with an edge in G that is incident with x and call all vertices added during this step *Gen2 vertices*. Finally, for all edges e of G, we create a new reticulation h_e with parents the two vertices in the DAG constructed thus far labeled e and add a leaf ℓ_e as a child to h_e. We call vertices added during this step *Gen3 vertices*.

Let N denote the resulting DAG. One can easily verify that N is an m-network that is tree-child. We now show that there exists a solution to GRAPH m-COLORABILTY for G if and only if N is proper forest-based, which will complete the proof.

Suppose first that there exists a proper m-coloring $\kappa : X \to \{1, \ldots, m\}$ of G. From κ, we derive an m-coloring $\sigma : V(N) \to \{1, \ldots, m\}$ of N as follows. Let $v \in V(N)$. If v is a Gen1 tree-vertex then there exists some $1 \leq i \leq m$ such that $v \in V(T_i)$. In this case, we put $\sigma(v) = i$. If v is a Gen2 vertex or a Gen1 reticulation, then $v \in \{x\} \cup V(P_x)$ for some $x \in X$, and we put $\sigma(v) = \kappa(x)$. Finally, if v is a Gen3 vertex, then $v \in \{h_e, \ell_e\}$ for some edge $e = \{x, y\}$ of G. In this case, we choose $\sigma(v) \in \{\kappa(x), \kappa(y)\}$ if $v = h_e$ and we put $\sigma(v) = \sigma(h_e)$ if $v = \ell_e$.

To see that N is proper forest-based it suffices to show by Lemma 3.1 that σ satisfies Properties (C1) and (C2).
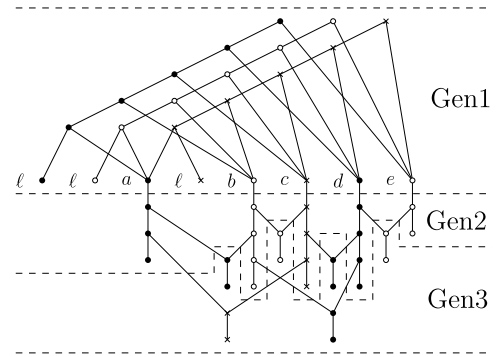
Fig. 3. The 3-network N constructed from the graph G with vertex set $X = \{a, b, c, d, e\}$ and edge set $E(G) = \{\{a, b\}, \{a, c\}, \{b, c\}, \{b, d\}, \{c, d\}, \{d, e\}\}$, as described in the proof of Proposition 5.1. Gen1, Gen2 and Gen3 vertices are indicated as described in Fig. 2. A 3-coloring $\sigma : V(N) \to \{\bullet, \circ, \times\}$ associated to the proper 3-coloring κ of G given by $\kappa(a) = \kappa(d) = \bullet$, $\kappa(b) = \kappa(e) = \circ$ and $\kappa(c) = \times$.

To see that (C1) holds, note first that, by construction, all non-root vertices v of N have at least one parent u satisfying $\sigma(u) = \sigma(v)$. Suppose now that v is a reticulation. Then v is either a Gen1 reticulation or $v = h_e$ for some edge e of G. If v is a Gen1 reticulation then $v \in X$. Hence, v has m parents $v_1, \ldots, v_m$. Since, for each $1 \leq i \leq m$, there exists a unique tree T_i that contains v_i it follows that there exists a unique $1 \leq j \leq m$ such that $\sigma(v) = \sigma(v_j) = j$. If $v = h_e$ then let $x, y \in X$ such that $e = \{x, y\}$. Without loss of generality, assume that v_1 is a vertex on P_x and that v_2 is a vertex on P_y. Assume that we have chosen $\sigma(h_e) = \kappa(x)$ in the definition of σ. Then $\sigma(h_e) = \kappa(x) = \sigma(v_1)$. Since κ is a proper m-coloring of G, we also have $\sigma(v_2) = \kappa(y) \neq \kappa(x)$. Thus, (C1) holds.

To see that (C2) holds, let $v \in V(N) - L(N)$. If v is a Gen1 vertex that is not a reticulation, then v has at least one child u that is a Gen1 vertex that is not a reticulation. In particular, v and u belong to the same tree T_i, $1 \leq i \leq m$. Hence, $\sigma(v) = \sigma(u)$, by the definition of σ. If v is a Gen2 vertex or a Gen1 reticulation, then v has exactly one child u that is a Gen2 vertex. By definition, $\sigma(u) = \sigma(v)$ also holds in this case. Finally, if v is a Gen3 vertex, then $v = h_e$ for some edge e of G. Hence, $\sigma(\ell_e) = \sigma(v)$ holds by definition. Thus, (C2) holds.

Conversely, suppose that N is proper forest-based. By Lemma 3.1, there exists an m-coloring $\sigma : V(N) \to \{1 \ldots, m\}$ of N satisfying (C1) and (C2). Since the set of Gen1 reticulations of N is X, the restriction of σ to X induces an m-coloring $\kappa : X \to \{1, \ldots, m\}$ of G.

We claim that κ is a proper m-coloring of G. To see the claim, consider an edge $e = \{x, y\}$ of G. By construction, there exists a (unique) Gen3 reticulation v such that $v = h_e$. Let v_1 denote the parent of h_e on P_x. Similarly, let v_2 denote the parent of h_e on P_y. By (C1), $\sigma(v_1) = \kappa(x)$ and $\sigma(v_2) = \kappa(y)$ hold. Since, by (C2), $\sigma(h_e) = \sigma(v_i)$ for a unique $i \in \{1, 2\}$, say $i = 1$, it follows that $\kappa(y) = \sigma(v_2) \neq \sigma(h_e) = \sigma(v_1) = \kappa(x)$. Thus, κ is a proper m-coloring of G. $\square$

6. Binary tree-child networks

In this section, we prove Statements (R1) and (R2). We begin with some definitions. Suppose N is an m-network and $v \in V(N)$. We denote by $\mathcal{P}_N(v)$ the set of parents of v and by $C_N(v)$ the set of children of v.

Statement (R2) follows from the following result.

Theorem 6.1. *For each fixed $m \geq 3$, the problem of determining whether a binary tree-child m-network is proper forest-based is NP-complete.*

Proof. Membership of NP follows from Proposition 5.1. We again reduce from the GRAPH m-COLORABILTY problem. Suppose G is a graph with vertex set X. We use the construction described in the proof of Proposition 5.1 to obtain a tree-child m-network N^s. We then create a

binary tree-child network N^b from N^s by applying the following operation to each reticulation x with indegree $d \geq 3$ (by construction, $d = m$).

(i) Introduce vertices $u_2, v_2, w_2, \ldots, u_{d-1}, v_{d-1}, w_{d-1}$.
(ii) Remove arcs $(p_1, x), \ldots, (p_{d-1}, x)$, with $p_1, \ldots, p_d$ the parents of x in N^s (in arbitrary order).
(iii) Add arcs $(p_1, u_2), (p_i, u_i), (u_i, v_i), (v_i, w_i), (v_i, u_{i+1})$ for $i = 2, \ldots, d - 1$ with $u_d = x$.

We illustrate the above construction for a reticulation with indegree $d = 4$ in Fig. 4. We comment that, for a general tree-child network N^s, it does not necessarily hold that N^s is proper forest-based if and only if N^b is proper forest-based. Therefore, we will use the specific structure of the network N^s constructed here.

We next show that N^b is proper forest-based if and only if G admits a proper m-coloring. Suppose first that G admits a proper m-coloring. Then, by the proof of Proposition 5.1, N^s is proper forest-based. Hence, there exists an m-coloring $\sigma : V(N^s) \to C$ of N^s in terms of a set C of colors such that (C1) and (C2) hold.

We define an m-coloring $\sigma' : V(N^b) \to C$ of N^b as follows. For $w \in V(N^s)$, we put $\sigma'(w) = \sigma(w)$. For each vertex introduced in Step (i) for some reticulation x, we do the following. Let p_j be the parent of x satisfying $\sigma(p_j) = \sigma(x)$ (which must exist in view of (C1)). For $1 < i < j$, we put $\sigma'(u_i) = \sigma'(v_i) = \sigma'(w_i) = \sigma(p_1)$. For $j \leq i \leq d - 1$ we put $\sigma'(u_i) = \sigma'(v_i) = \sigma'(w_i) = \sigma(x) = \sigma(p_j)$. To see that N^b is proper forest-based, we show that σ' satisfies (C1) and (C2).

To see that σ' satisfies (C1), let w be a non-root vertex of N^b. If $w \in V(N^s)$ and w has indegree at most 2 in N^s, then $\mathcal{P}_{N^s}(w) = \mathcal{P}_{N^b}(w)$, $\sigma'(w) = \sigma(w)$, and $\sigma'(q) = \sigma(q)$, for all parents $q \in \mathcal{P}_{N^s}(w)$. By (C1), there exists exactly one parent q in $\mathcal{P}_{N^s}(w)$ that satisfies $\sigma(q) = \sigma(w)$. It follows that q is the unique parent in $\mathcal{P}_{N^b}(w)$ satisfying $\sigma'(q) = \sigma'(w)$. So, (C1) holds.

Now assume w is introduced in Step (i) for some reticulation x. Let p_j again be the parent of x satisfying $\sigma(p_j) = \sigma(x)$.

If $w \in \{v_i, w_i\}$, for some $2 \leq i \leq d - 1$, then w has exactly one parent q and $\sigma'(w) = \sigma'(q)$. So (C1) holds.

Before continuing to the next case, observe that, since every root of N has a different color, the parents of x in N^s also have different colors, by (C1).

Now consider the case $w = u_i$ for $i \in \{2, \ldots, j - 1\}$. Then w has two parents v_{i-1}, p_i, with $v_1 = p_1$, and $\sigma'(w) = \sigma'(v_{i-1}) = \sigma'(p_1) \neq \sigma'(p_i)$. So (C1) holds.

If $w = u_j$, then w has two parents v_{j-1}, p_j and $\sigma'(w) = \sigma'(p_j) \neq \sigma'(v_{j-1}) = \sigma'(p_1)$. So (C1) holds.

Finally assume $w = u_i$ for $i \in \{j + 1, \ldots, d\}$ with $u_d = x$. Then w has two parents v_{i-1}, p_i and $\sigma'(w) = \sigma'(v_{i-1}) = \sigma'(p_j) \neq \sigma'(p_i)$. So (C1) holds again.

To show that σ' also satisfies (C2), consider an internal vertex w of N^b. First suppose $w \in V(N^s)$ and w is not a parent of a reticulation with indegree greater than 2 in N^s. Then $\mathcal{C}_{N^s}(w) = \mathcal{C}_{N^b}(w)$. In particular, $\sigma'(w) = \sigma(w)$ and $\sigma'(z) = \sigma(z)$, for all children $z \in \mathcal{C}_{N^s}(w)$. By (C2), there exists at least one child $z \in \mathcal{C}_{N^s}(w)$ with $\sigma(z) = \sigma(w)$. So $\sigma'(z) = \sigma'(w)$ holds.

Now assume w is introduced in Step (i) for some reticulation x. Then, by the construction of σ', w has the same color as at least one of its children.

Now assume w is a parent of a reticulation with indegree greater than 2 in N^s. Then w has a child c with indegree 1 in N^s and with $\sigma(c) = \sigma'(c)$. Hence, $\sigma(c) = \sigma(w)$ and $\sigma'(c) = \sigma'(p)$. So, σ' satisfies (C2).

Conversely suppose N^b is proper forest-based. Then there exists an m-coloring σ of N^b in terms of a set C of colors that satisfies Properties (C1) and (C2). Restricting this coloring to X to obtain a coloring $\kappa : X \to C$ and then applying the same arguments as in the last paragraph of the proof of Proposition 5.1 implies that κ is a proper m-coloring of G. Note that these arguments only use the vertices in X and their descendants and hence they are the same for N^b as for N^s. $\square$
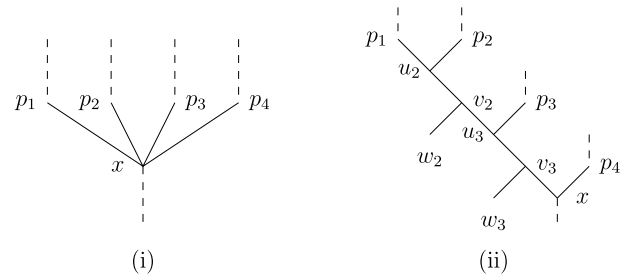

Fig. 4. (i) A reticulation x with 4 parents $p_1, \ldots, p_4$. (ii) The construction described in the proof of Theorem 6.1, which reduces the indegree of x to 2.
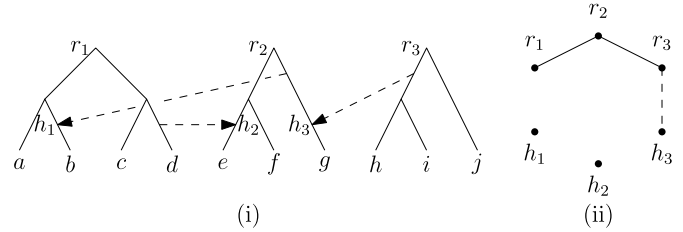

Fig. 5. (i) A forest-based 3-network on $X = \{a, b, c, d, e, f, g, h, i, j\}$. (ii) The graph with vertices r_1, r_2, r_3, h_1, h_2 and h_3 and solid edges $\{r_1, r_2\}$ and $\{r_2, r_3\}$ is the graph $\Gamma(N)$. Adding the dashed edge $\{r_1, h_3\}$ results in a minimal omni-extension of $\Gamma(N)$.

To prove the next result, we require further concepts from [9]. Let N be a semi-binary m-network, $m \geq 2$, and let v be a vertex of N. We denote by $\gamma_N(v)$ the (necessarily unique) lowest ancestor of v in N whose indegree is not 1. Note that $\gamma_N(v)$ is either a root or a reticulation of N. Building on this definition, we define an undirected graph $\Gamma(N)$ as follows. The vertex set of $\Gamma(N)$ is the set of all vertices of N whose indegree is not 1. Two such vertices v_1, v_2 are joined by an edge in $\Gamma(N)$ if there exists two distinct vertices u_1, u_2 in N such that $\gamma_N(u_1) = v_1$, $\gamma_N(u_2) = v_2$, and u_1 and u_2 share a child in N. The intuitive idea behind these edges is that they indicate that the vertices v_1, v_2 need to be contained in different trees in a potential subdivision forest of N and so need to be assigned different colors. Indeed, for an m-coloring σ of N satisfying (C1) and (C2), $\sigma(u_1) = \sigma(v_1)$ and $\sigma(u_2) = \sigma(v_2)$ must hold by (C1) and by definition of $\gamma_N(v)$, $v \in V(N)$. Property (C1) together with the fact that the common child of u_1 and u_2 has indegree 2 implies $\sigma(u_1) \neq \sigma(u_2)$.

As it turns out, the edges of the graph $\Gamma(N)$ are not sufficient to determine whether N is proper forest-based. This is caused by internal vertices of N for which all children have indegree 2 or more. We call such a vertex an *omnian (vertex)* [10]. In order for (C2) to hold, for all omnian vertices v of N, at least one child of v needs to be assigned the same color as v. To ensure that this property is satisfied, we will use certain supergraphs of $\Gamma(N)$ called "omni-extensions". For a semi-binary network N, we define an *omni-extension* of $\Gamma(N)$ as a supergraph $\Gamma'(N)$ of $\Gamma(N)$ such that $V(\Gamma'(N)) = V(\Gamma(N))$ and for all omnians v of N there exists a child h of v such that $\{h, \gamma_N(u)\}$ is an edge of $\Gamma'(N)$, with u being the parent of h in N other than v [8]. Note that $\Gamma(N)$ may be an omni-extension of itself. This is the case, in particular, if N has no omnian vertex. If $\Gamma'(N)$ is an omni-extension of $\Gamma(N)$, and no proper subgraph of $\Gamma'(N)$ is an omni-extension of $\Gamma(N)$, we say that $\Gamma'(N)$ is a *minimal omni-extension* of $\Gamma(N)$.

We illustrate these concepts in Fig. 5. Observe that the common parent v of h_1 and h_3 in Fig. 5(i) is an omnian. At least one of the children of v needs to be assigned the same color as v (in a coloring satisfying (C2) in Lemma 3.1). In this example, we choose child h_3 as h. This means that the other parent u of h_3 needs to be assigned a different color than h_3 (by (C1)) and hence that $\gamma_N(u) = r_3$ needs to be assigned

a different color than h_3 (again using (C1)). This is the intuitive idea behind adding the dashed edge $\{h_3, r_3\}$ in Fig. 5(ii).

Statement (R1) is a consequence of the next result.

Theorem 6.2. *Given a binary tree-child 2-network N, it can be determined in $O(nr)$ time whether N is proper forest-based, where r is the number of reticulations of N and $n = |L(N)|$.*

Proof. [8, Theorem 8] states that a 2-network, in which all reticulations have indegree 2, is proper forest-based if and only if the graph $\Gamma(N)$ has a bipartite omni-extension. This theorem is applicable to N since it is binary. Since N is tree-child, it has no omnians. Hence, $\Gamma(N)$ is an omni-extension of $\Gamma(N)$.

We claim that N is proper forest-based if and only if $\Gamma(N)$ is bipartite. First suppose that N is proper forest-based. Then, by [8, Theorem 8] recalled above, $\Gamma(N)$ has a bipartite omni-extension. Since $\Gamma(N)$ is a subgraph of every omni-extension of $\Gamma(N)$, it follows that $\Gamma(N)$ is bipartite. Conversely, if $\Gamma(N)$ is bipartite, then, since $\Gamma(N)$ is an omni-extension of $\Gamma(N)$, it follows again by [8, Theorem 8] recalled above that N is proper forest-based. The claim therefore holds.

To construct $\Gamma(N)$, we need to find, for each parent v of a reticulation of N, the vertex $\gamma_N(v)$. This takes $O(|V(N)|)$ time per reticulation of N. Hence, the construction of $\Gamma(N)$ takes $O(|V(N)| \cdot r)$ time. Checking whether $\Gamma(N)$ is bipartite takes $O(r)$ time, since $\Gamma(N)$ has at most r edges. Hence, the total running time is $O(|V(N)| \cdot r)$.

Finally, to obtain the running time stated in the theorem, note that by [14, Proposition 10.7] a tree-child 1-network with n leaves has fewer than $4n$ vertices. Adding a vertex ρ and an arc from ρ to each of the two roots in N results in a tree-child 1-network, and increases the number of vertices by only 1. It follows that N has $O(n)$ vertices. This concludes the proof. $\square$

7. An FPT algorithm

In this section, we shall present a fixed-parameter tractable (FPT) algorithm called CHECK for deciding whether a semi-binary m-network N, $m \geq 2$, is forest-based, with respect to the parameter combination of m, the number r of reticulations of N, and the maximum outdegree Δ of N. This will enable us to prove that Statement (R4) holds. Note that we do not require the network to be tree-child.

For N an m-network, $m \geq 2$, we denote the set of roots and reticulations of N by $RH(N)$. Note that $RH(N)$ is precisely the vertex set of $\Gamma(N)$. For G a graph with vertex set $RH(N)$ and σ a coloring of G, we say that a directed path P in N is a *σ-uniform path* if for all vertices $v, v' \in RH(N)$ on P, we have $\sigma(v) = \sigma(v')$. Note that a directed path in N containing at most one vertex of $RH(N)$ is trivially σ-uniform.

Algorithm 1 The algorithm CHECK.

Input: A semi-binary m-network N, $m \geq 2$.
Output: The statement "N is proper forest-based" or the statement "N is not proper forest-based".
1: Construct $\Gamma(N)$ and find all minimal omni-extensions of $\Gamma(N)$.
2: **for all** minimal omni-extension of $\Gamma(N)$ **do**
3:     Add an edge between any two distinct roots of N to obtain a graph $\Gamma^*(N)$.
4:     Find all proper m-colorings of $\Gamma^*(N)$.
5:     **for all** proper m-coloring σ of $\Gamma^*(N)$ **do**
6:         **for all** reticulations h of N and all roots ρ of N with $\sigma(\rho) = \sigma(h)$ **do**
7:             Check that there exists a σ-uniform path from ρ to h in N.
8:         **if** the latter holds for all reticulations h of N and all roots ρ of N with $\sigma(\rho) = \sigma(h)$ **then**
9:             **return** "N is proper forest-based".
10: **return** "N is not proper forest-based".

Correctness of the algorithm CHECK follows from Theorem 7.1 which is a slight strengthening of [8, Theorem 7] to minimal omni-extensions.

Theorem 7.1. *Let N be a semi-binary m-network, $m \geq 2$. Then N is proper forest-based if and only if there exists a minimal omni-extension $\Gamma'(N)$ of $\Gamma(N)$ and a proper m-coloring σ of $\Gamma'(N)$ satisfying:*

(F1) The restriction of σ to the set $R(N)$ of roots of N is a bijection.
(F2) For all $u \in R(N)$ and all reticulations v of N such that $\sigma(u) = \sigma(v)$ there exists a σ-uniform path from u to v in N.

Proof. Suppose first that there exists a minimal omni-extension $\Gamma'(N)$ of $\Gamma(N)$ and a proper m-coloring of $\Gamma'(N)$ satisfying (F1) and (F2). Then by [8, Theorem 7], N is proper forest-based.

Conversely, suppose that N is proper forest-based. Then by [8, Theorem 7], there exists an omni-extension G of $\Gamma(N)$ and a proper m-coloring σ of G satisfying (F1) and (F2). Since (F1) and (F2) are independent of the structure of G, it follows for all subgraphs G^- of G with $V(G) = V(G^-)$ that σ is a proper m-coloring of G^- satisfying (F1) and (F2). In particular, if there exists an omni-extension of $\Gamma(N)$ and a proper m-coloring of G satisfying (F1) and (F2), then there exists a minimal omni-extension of $\Gamma(N)$ and a proper m-coloring of G satisfying (F1) and (F2). $\square$

We now analyze the run time of algorithm CHECK. The graph $\Gamma(N)$ can be constructed in $O(r|V(N)|)$ time and can have at most Δ^ω minimal omni-extensions where ω is the number of omnians of N. Since N is semi-binary, ω is at most $2r$ (each omnian is the parent of at least one reticulation and each reticulation has two parents). All proper m-colorings of $\Gamma^*(N)$ can be found in $O(m^{r+m}(r+m)^2)$ time since $\Gamma^*(N)$ has $r + m$ vertices (Line 4). Since, by construction, the set of roots of N forms a clique in $\Gamma^*(N)$ and we are interested in proper m-colorings of $\Gamma^*(N)$ it follows that for every reticulation h of N there exist a unique root ρ_h of N that has the same color under the m-coloring σ under consideration as h. In the worst case, checking the existence of a σ-uniform path from ρ_h to h in N takes $O(|V(N)|)$ time per pair (ρ_h, h) (by deleting all reticulations that do not have the same color as ρ_h and then doing a depth-first search) (Line 7). The total run time of algorithm CHECK therefore is $O(\Delta^{2r} m^{r+m}(r+m)^2|V(N)|)$.

In conclusion, we have the following result from which Statement (R4) immediately follows.

Theorem 7.2. *There exists an algorithm with running time $O(\Delta^{2r} m^{r+m}(r+m)^2|V(N)|)$ to decide whether a semi-binary m-network with r reticulations and maximum outdegree Δ is proper forest-based, with $m \geq 2$.*

8. Discussion

In this paper, we have shown that it can be decided in polynomial time whether a binary, tree-child, 2-network is proper forest-based. It would be interesting to know if the same problem can be solved in polynomial time for more general binary 2-networks, e.g. for binary tree-sibling 2-networks (i.e. 2-networks in which for every reticulation v, there exists a tree-vertex or leaf u such that u and v share a parent) or even for general binary 2-networks. In addition, although we have shown that there is an FPT algorithm for deciding whether a semi-binary m-network, $m \geq 2$ is proper forest-based, it would be interesting to see if FPT-algorithms with improved run-times can be developed, or if an FTP algorithm can be found for general m-networks.

Also, we have not considered the problem of deciding whether an m-network is forest-based, $m \geq 1$, i.e. the problem where we do not insist that the underlying forest must have m trees. It would be interesting to know whether this is an NP-complete problem. Note that since an m-network N is forest-based if and only if it contains a subdivision forest

in which every tree is an induced, directed path in N (cf. [8, Theorem 1]), this question is closely related to the induced path partition problem which is known to be NP-complete [2].

Finally, as more methods become available for computing networks from biological data (see e.g. [17]), our results should be useful for understanding whether a computed network is proper forest-based. It could also be interesting to develop approaches to decide how close a network is to being forest-based in case it is not forest based (see e.g. [3] for related work on edge-based 1-networks).

CRediT authorship contribution statement

Katharina T. Huber: Writing – review & editing, Writing – original draft, Formal analysis, Conceptualization. **Leo van Iersel:** Writing – review & editing, Writing – original draft, Formal analysis, Conceptualization. **Vincent Moulton:** Writing – review & editing, Writing – original draft, Formal analysis, Conceptualization. **Guillaume E. Scholz:** Writing – review & editing, Writing – original draft, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgement

References

[1] N.B. Edelman, P.B. Frandsen, M. Miyagi, B. Clavijo, J. Davey, R.B. Dikow, G. García-Accinelli, S.M. Van Belleghem, N. Patterson, D.E. Neafsey, et al., Genomic architecture and introgression shape a butterfly radiation, Science 366 (6465) (2019) 594–599.

[2] H. Fernau, F. Foucaud, K. Mann, U. Padariya, K.N.R. Rao, Parameterizing path partitions, in: International Conference on Algorithms and Complexity, Springer, 2023, pp. 187–201.

[3] M. Fischer, T.N. Hamann, K. Wicke, How far is my network from being edge-based? Proximity measures for edge-basedness of unrooted phylogenetic networks, Discrete Appl. Math. 337 (2023) 303–320.

[4] A. Francis, K.T. Huber, V. Moulton, Tree-based unrooted phylogenetic networks, Bull. Math. Biol. 80 (2018) 404–416.

[5] A. Francis, M. Steel, Labellable phylogenetic networks, Bull. Math. Biol. 85 (6) (2023) 46.

[6] A.R. Francis, M. Steel, Which phylogenetic networks are merely trees with additional arcs?, Syst. Biol. 64 (5) (2015) 768–777.

[7] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of on NP-Completeness, Freeman, 1979.

[8] K.T. Huber, V. Moulton, G.E. Scholz, Forest-based networks, Bull. Math. Biol. (2022) 119.

[9] K.T. Huber, V. Moulton, G.E. Scholz, Overlaid species forest, Discrete Appl. Math. 309 (2022).

[10] L. Jetten, L. van Iersel, Nonbinary tree-based phylogenetic networks, IEEE/ACM Trans. Comput. Biol. Bioinform. 15 (1) (2016) 205–217.

[11] S. Kong, J.C. Pons, L. Kubatko, K. Wicke, Classes of explicit phylogenetic networks and their biological and mathematical significance, J. Math. Biol. 84 (6) (2022) 47.

[12] V. Moulton, T. Wu, Planar rooted phylogenetic networks, IEEE/ACM Trans. Comput. Biol. Bioinform. 20 (2) (2022) 1289–1297.

[13] G.E. Scholz, A.A. Popescu, M.I. Taylor, V. Moulton, K.T. Huber, OSF-Builder: a new tool for constructing and representing evolutionary histories involving introgression, Syst. Biol. 68 (2019) 717–729.

[14] M. Steel, Phylogeny: Discrete and Random Processes in Evolution, SIAM, 2016.

[15] L. van Iersel, R. Janssen, M. Jones, Y. Murakami, Orchard networks are trees with additional horizontal arcs, Bull. Math. Biol. 84 (8) (2022) 76.

[16] R.W.R. Wallbank, S.W. Baxter, C. Pardo-Diaz, J.J. Hanly, S.H. Martin, J. Mallet, K.K. Dasmahapatra, C. Salazar, M. Joron, N. Nadeau, et al., Evolutionary novelty in a butterfly wing pattern through enhancer shuffling, PLoS Biol. 14 (1) (2016) e1002353.

[17] L. Zhang, N. Abhari, C. Colijn, Y. Wu, A fast and scalable method for inferring phylogenetic networks from trees by aligning lineage taxon strings, Genome Res. 33 (7) (2023) 1053–1060.