Orchard Networks are Trees with Additional Horizontal Arcs

van Iersel, Leo; Janssen, Remie; Jones, Mark; Murakami, Yukihiro

DOI
10.1007/s11538-022-01037-z

Publication date
2022

Document Version
Final published version

Published in
Bulletin of Mathematical Biology

Citation (APA)
van Iersel, L., Janssen, R., Jones, M., & Murakami, Y. (2022). Orchard Networks are Trees with Additional Horizontal Arcs. *Bulletin of Mathematical Biology*, *84*(8), Article 76. https://doi.org/10.1007/s11538-022-01037-z

Bulletin of Mathematical Biology (2022) 84:76
https://doi.org/10.1007/s11538-022-01037-z

Society for Mathematical Biology

ORIGINAL ARTICLE

Orchard Networks are Trees with Additional Horizontal Arcs

Leo van Iersel[1] · Remie Janssen[1] · Mark Jones[1] · Yukihiro Murakami[1]

Received: 20 October 2021 / Accepted: 30 May 2022

Abstract
Phylogenetic networks are used in biology to represent evolutionary histories. The class of orchard phylogenetic networks was recently introduced for their computational benefits, without any biological justification. Here, we show that orchard networks can be interpreted as trees with additional *horizontal* arcs. Therefore, they are closely related to tree-based networks, where the difference is that in tree-based networks the additional arcs do not need to be horizontal. Then, we use this new characterization to show that the space of orchard networks on n leaves with k reticulations is connected under the rNNI rearrangement move with diameter $O(kn + n\log(n))$.

Keywords Phylogenetic networks · Orchard networks · Rearrangement moves · Connectedness

1 Introduction

Phylogenetic trees and networks are used in biology to represent evolutionary histories of groups of taxa (taxonomic units), which can for example be different species or variants. Phylogenetic trees are used to show how the taxa (hypothetically) evolved from a common ancestor by branching events. Phylogenetic networks are used to

Research funded in part by the Netherlands Organization for Scientific Research (NWO) Vidi Grant 639.072.602 and KLEIN Grant OCENW.KLEIN.125.

✉ Yukihiro Murakami
Y.Murakami@tudelft.nl

Leo van Iersel
L.J.J.vanIersel@tudelft.nl

Remie Janssen
remiejanssen92@gmail.com

Mark Jones
M.E.L.Jones@tudelft.nl

[1] Delft Institute of Applied Mathematics, Delft University of Technology,
Mekelweg 4, 2628 CD Delft, South Holland, The Netherlands

Published online: 21 June 2022

augment this view with the inclusion of reticulate processes such as hybridization, recombination and horizontal gene transfer (HGT), which is also called lateral gene transfer (LGT) (Bapteste et al. 2013; Elworth et al. 2019; Blais and Archibald 2021). In this paper, we consider *rooted* phylogenetic trees and networks, in which the leaves represent the considered extant taxa and other nodes represent hypothetical ancestors. The direction of the arcs represents the direction of evolution. In phylogenetic networks, reticulate processes are represented by *reticulation nodes* with two or more incoming arcs (called *reticulation arcs*). See Fig. 1 for an example.

Sometimes, only a subset of all phylogenetic networks is considered, either to ease the computation of a good representation of the actual evolutionary history (e.g., van Iersel and Moulton 2014; Bordewich et al. 2018; Erdős et al. 2019; van Iersel et al. 2021a; Borst et al. 2020), or because the evolutionary history is expected to have a certain structure (e.g., Markin et al. 2019). Examples of such subsets are the classes of phylogenetic networks called tree-child networks, tree-based networks, and temporal networks. Tree-based networks are mainly used for their biological interpretation as a tree with additional arcs (e.g., Francis and Steel 2015; Cardona et al. 2015; Fischer et al. 2020). Temporal and tree-child networks are especially interesting because of their computation benefits and nice mathematical properties. Tree-child networks can be biologically interpreted as networks in which each reticulate event leaves a trace in the genetic information of the studied species (Cardona et al. 2008a; Markin et al. 2019). Networks with a temporal labelling can be biologically interpreted as networks where each reticulation represents a hybrid speciation event (Moret et al. 2004; Baroni et al. 2006). Networks that are tree-child and have a temporal labelling are sometimes simply called *temporal* (Humphries et al. 2013). It should be noted though that extinctions or undersampling may lead to phylogenetic networks that are not temporal or not tree-based.

Another class of networks is the class of *orchard networks*, which were recently introduced for their computational benefits (Janssen and Murakami 2021; Erdős et al. 2019) while being more general than tree-child networks. They are usually defined as networks that can be reduced to a single leaf by a series of cherry-picking reductions (for this reason, they were called “cherry picking networks” in Janssen and Murakami (2021)). Roughly speaking, a cherry picking reduction consists in (1) deleting one leaf of a pair of leaves with a common parent (reducing a *cherry*) or (2) deleting a reticulation arc of which each endpoint is the parent of a leaf (reducing a *reticulated cherry*; see Sect. 2.1 for formal definitions). Orchard networks come up naturally when one builds networks by repeatedly considering pairs of taxa that seem to be closely related in the data (Bordewich et al. 2018; van Iersel et al. 2021a; Erdős et al. 2019; Bai et al. 2021). Recently, it was shown that they can be characterized more structurally using so-called acyclic cherry covers (van Iersel et al. 2021b).

In this paper, we show that orchard networks can also be characterized in a very natural way. We prove that orchard networks are precisely those phylogenetic networks that can be obtained from a phylogenetic tree by inserting horizontal arcs (when the tree is drawn with the root at the top and all arcs directed downwards), see Fig. 1. Hence, they are closely related to the class of LGT networks (Pons et al. 2019), because the horizontal arcs can be used to model LGT events. However, note that, in contrast to LGT networks, orchard networks do not (necessarily) specify which arcs are

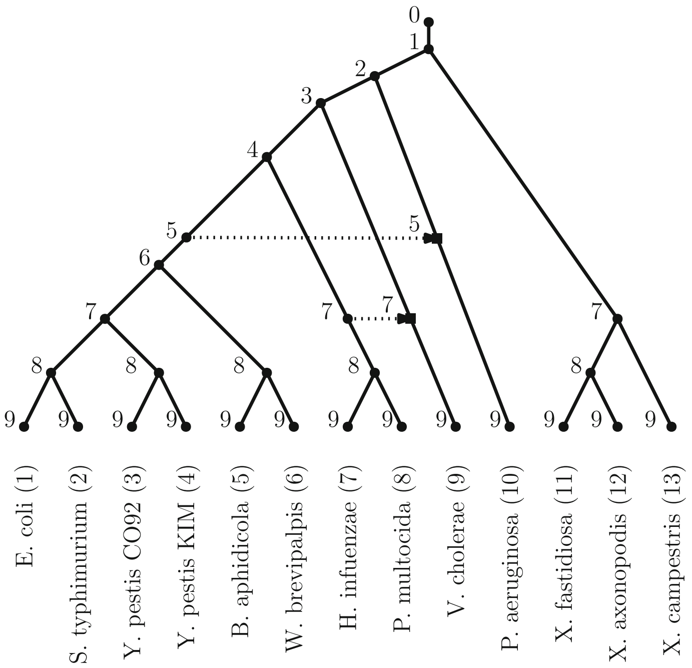

Fig. 1 A phylogenetic network on taxa within the γ-proteobacteria class with two reticulate events (Nakhleh et al. 2005). The dotted arcs with the arrowheads illustrate the passing of genetic material via a horizontal transfer. In this and all subsequent figures, the non-horizontal arcs without arrowheads are directed downwards. The network is a binary orchard network, since it can be reduced by a sequence of cherry reductions (see Sect. 2.1 for formal definitions of such reductions). One possible sequence that would reduce the network is $(1,2)(3,4)(2,4)(5,6)(4,6)(7,8)(9,8)(10,6)(6,8)(8,9)(9,10)(11,12)(12,13)(10,13)$. An HGT-consistent labelling of the network is given, indicated by the number next to each vertex. Note that the endpoints of an arc have the same label precisely if the arc corresponds to a horizontal transfer

horizontal. If a network is orchard, it *can* be drawn as a tree with additional horizontal arcs, but there may exist multiple such drawings as well as other drawings. Our characterization of orchard networks is similar to the previously mentioned concept of *tree-based* networks, which are those networks that can be obtained from a phylogenetic tree, called a *base tree*, by inserting arcs, called *linking arcs*. The difference is that for orchard networks the linking arcs need to be horizontal while for tree-based networks they do not.

Our characterization of orchard networks can be seen as a time-consistency property, which we call an *HGT-consistent labelling* (see Fig. 1). It basically says that orchard networks are consistent with an evolutionary history in time in which reticulate events represent instantaneous (horizontal) transfers such as HGT events. This is similar, but not the same, as the notion of time-consistency, or temporal labelling, that is commonly found in the phylogenetic networks literature (Moret et al. 2004; Baroni et al. 2006). The difference is that in a temporal labelling *both* arcs entering a reticulation need to be horizontal, which is more natural when reticulations represent hybridization events. This notion has been widely popular in defining network classes, which have been explored in relation to metrics (Cardona et al. 2008b), encodings (Cardona et al. 2008b), and reconstruction methods (Borst et al. 2020). Another related notion of time-consistency was introduced in Erdem et al. (2006), which allows bidirectional horizontal arcs as well.

Unlike the other definitions of orchard networks, the HGT-consistent labelling that we introduce here can easily be interpreted biologically. This makes the class of orchard networks even more relevant, as it has a natural biological interpretation as a tree with horizontal arcs. Nevertheless, as with the classes of temporal and tree-based networks, extinctions or undersampling may lead to networks that are not orchard, because one or more arcs corresponding to HGT events may need to be drawn forward in time.

To showcase the mathematical utility of the new characterization, we use it to prove that the space of orchard networks is connected under *rooted Nearest Neighbor Interchange* (rNNI) moves. This confirms the speculation that was mentioned in the concluding remarks of Erdős et al. (2021). Connectedness of search spaces under rearrangement moves, such as the rNNI move, is important because of their use in local search heuristics and Bayesian methods in phylogenetics (e.g., Markin et al. 2019; Bouckaert et al. 2019; Wen et al. 2018). Although it has already been proven that rearrangement moves are sufficient to connect several classes of networks (Bordewich et al. 2017; Klawitter 2020; Erdős et al. 2021), the connectedness of the space of orchard networks has not been investigated yet.

In this paper, we fill this gap in the literature. Because orchard networks naturally occur as results of statistical models for network generation, such as the level-k LGT networks from Pons et al. (2019), this result may prove to be especially important. Indeed, our results then show that it is possible to use such a model as a prior in a Bayesian method in conjunction with rNNI or *rooted Subtree Prune and Regraft* (rSPR) moves, while keeping the search space connected.

2 Preliminaries

In this section, we define phylogenetic networks and the original definition of orchard networks. Then we recall the definition of an rNNI move on a phylogenetic network, and some results related to these moves.

Definition 1 A *directed phylogenetic network* on a set of *taxa* X is a directed acyclic graph whose nodes are of the following types:

Root:	a node with indegree-0 and outdegree-1.
Tree node:	a node with indegree-1 and outdegree at least 2.
Reticulation:	a node with indegree at least 2 and outdegree-1.
Leaf:	a node with indegree-1 and outdegree-0.

The leaves are bijectively labelled by the taxa in X. Because of this, we refer to a leaf l by its label in X and vice versa. Non-leaf nodes will also be called *internal* nodes. Furthermore, we refer to directed phylogenetic networks as *networks* for short.

A network where each tree node and each reticulation has a total degree (sum of indegree and outdegree) of exactly 3 is called a *binary* network. We use the term *non-binary* to mean *not necessarily binary*; in particular, binary networks are also non-binary. Most networks we consider in this paper are binary networks. We prove results on non-binary networks only in Sect. 3.2.

As a network N is a directed acyclic graph, we have a natural ordering on the nodes of N. If N contains an arc (u, v), then we say that u is a *parent* of v, that v is a *child* of u, that u is the *tail* of (u, v) and that v is the *head* of (u, v). If there is a directed path from u to v, then we say that u is *above* v and v is *below* u. An arc (u, v) is a *reticulation arc* if v is a reticulation; it is a *tree arc* otherwise.

The *reticulation number* $r(N)$ of a network N is the total number of reticulation arcs minus the total number of reticulation nodes in N. In a binary network, the reticulation number is simply the number of reticulation nodes.

We say that two networks N and N' are *isomorphic* if there exists a bijection f between the nodes of N and the nodes of N' such that (u, v) is an arc of N if and only if $(f(u), f(v))$ is an arc of N' and each labelled node of N is mapped to a node in N' with the same label.

Let N be a network and let v be a node in N. For an arc (u, v) in N, *contracting the arc* (u, v) is the action of deleting the arc (u, v) and identifying u and v. We say that a network N' is a *binary resolution* of N if N' is binary and a network isomorphic to N can be obtained from N' by contracting arcs.

2.1 Orchard Networks

Orchard networks were first introduced as networks that can be reduced by picking *cherries* and *reticulated cherries* (Janssen and Murakami 2021; Erdős et al. 2019). In this section, we recall this definition of orchard networks.

Definition 2 An ordered pair of leaves (x, y) in a non-binary network N is a *cherry* if x and y share a common parent. The pair (x, y) is a *reticulated cherry* if the parent p_x of x is a reticulation, and p_x and y share a common parent. If (x, y) is a cherry or a reticulated cherry, we call it a *reducible pair*.

Observe that (x, y) is a cherry in N if and only if (y, x) is a cherry in N. Such an equivalence does not hold for reticulated cherries. *Suppressing* a node v with exactly one parent u and exactly one child w refers to deleting the node v and adding an arc (u, w).

Definition 3 Let N be a non-binary network and let (x, y) be a reducible pair in N. Let p_x, p_y denote the parents of x, y, respectively. *Reducing* (or *picking*) the pair (x, y) in N consists of the following operations (see Fig. 2).

- If (x, y) is a cherry, then delete x and suppress p_x if p_x is consequently a node of indegree-1 and outdegree-1.
- If (x, y) is a reticulated cherry, then delete the arc (p_y, p_x) and suppress indegree-1 and outdegree-1 nodes (p_x and p_y are the only candidates for this suppression).

It is easy to see that the graph obtained by reducing a pair from a network is still a network, precisely because we suppress indegree-1 and outdegree-1 nodes. We

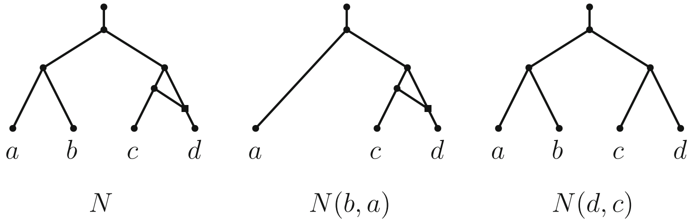

Fig. 2 The network N on the set of taxa $\{a, b, c, d\}$ has a cherry (a, b) and a reticulated cherry (d, c). The networks $N(b, a)$ and $N(d, c)$ are the networks obtained by reducing the respective pair from N

let $N(x, y)$ denote the network obtained by reducing a pair (x, y) from a network N. Let S be a sequence of ordered pairs on distinct elements. The network obtained by repeatedly reducing the pairs of S in order from N is denoted NS.

Definition 4 A non-binary network is *orchard* if there exists a sequence of ordered pairs S such that NS is a tree with one leaf. In such a case, we say that S reduces N.

We call a sequence of ordered pairs a *cherry-picking sequence* if it reduces some orchard network, such that the length of the sequence is minimal. It was shown independently in Janssen and Murakami (2021) and Erdős et al. (2019) that picking any reducible pair in an orchard network results in another orchard network (also for non-binary networks Janssen and Murakami 2021). This means that in general, orchard networks may have multiple cherry-picking sequences that reduce them. See Fig. 1 for an example of an orchard network.

2.2 Tree-Based Networks

Tree-based networks were introduced as those that can be obtained by adding arcs between arcs of a given *base tree* (Francis and Steel 2015). We recall the following definition.

Definition 5 A binary network N is *tree-based* with *base tree* T if N can be obtained from T in the following steps.

1. Replace some arcs of T by paths, whose internal nodes are called *attachment points*; each attachment point is of indegree-1 and outdegree-1.
2. Place arcs between attachment points, called *linking arcs*, so that the graph contains no nodes of total degree greater than 3, and so that it remains acyclic.
3. Suppress all attachment points not incident to any linking arcs.

Definitions regarding rearrangement moves are given in Sect. 4.1.

3 Characterization of Orchard Networks

In this section, we prove that orchard networks can be characterized as phylogenetic networks that admit a certain type of time-consistent labelling. We show this first for binary orchard networks, and later extend the characterization to non-binary orchard networks.

3.1 Binary Orchard Networks

Definition 6 Let N be a binary phylogenetic network with node set V. An *HGT-consistent labelling* of N is a labelling $t : V \to \mathbb{R}$ such that

1. For all arcs (u, v), $t(u) \leq t(v)$ and equality is only allowed if v is a reticulation.
2. For each internal node u, there is a child v of u such that $t(u) < t(v)$.
3. For each reticulation r with parents u and v, exactly one of $t(u) = t(r)$ and $t(v) = t(r)$ holds.

The properties listed in Definition 6 will be referred to as Properties 1,2, and 3, respectively. The biological interpretation of a network with an HGT-consistent labelling is that reticulations are caused within the network by horizontal transfers. Given a network with an HGT-consistent labelling, we call the arcs whose endpoints have the same labels *horizontal* arcs. Property 3 ensures that reticulation nodes are contemporaneous with one of their parents—in particular, with the parent from which genetic material is passed via the horizontal arc. Intuitively, one can view a network with an HGT-consistent labelling as a tree with horizontal arcs added to it. This means that a base tree of a given network with HGT-consistent labelling can be obtained by deleting all horizontal arcs. The following observation follows immediately.

Observation 1 *Each binary network that admits an HGT-consistent labelling is tree-based. In particular, a base tree can be obtained by deleting all reticulation arcs where the tail and head have the same label.*

The converse of Observation 1 does not hold. This can be seen as follows. We say that a network *contains a crown* if there exists a set of nodes $\{u_1, \ldots, u_k, v_1, \ldots, v_k\}$ with arcs $\{(u_i, v_i), (u_i, v_{i+1}) : i \in [k]\}$ where $[k] = \{1, \ldots, k\}$, and where the indices are taken modulo k. Consider a tree-based network that contains a crown, such as the network in Fig. 3. Such a network does not admit an HGT-consistent labelling. To see this, first note that in such a labelling t, without loss of generality, $t(u_1) = t(v_1)$ (by Property 3). Moreover, if $t(u_i) = t(v_i)$ then it follows that $t(u_i) < t(v_{i+1})$ (by

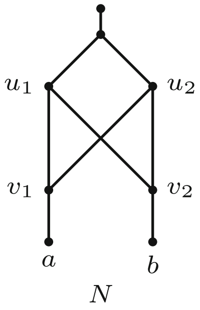

Fig. 3 A binary tree-based network N on the taxa set $\{a, b\}$ that does not admit an HGT-consistent labelling since it contains a crown on the nodes u_1, u_2, v_1, v_2

Property 2) and hence that $t(u_{i+1}) = t(v_{i+1})$ (by Property 3). Then, by induction, it follows that $t(u_1) < t(u_2) < \cdots < t(u_k) < t(u_1)$, a contradiction.

We now show that every binary orchard network admits an HGT-consistent labelling of a certain form.

Lemma 1 *Each binary orchard network admits an HGT-consistent labelling t where two nodes have the same label only if they are a parent–child pair where the parent is a tree node and the child a reticulation.*

Proof Let N be a binary orchard network with leaves $l_1, \ldots, l_n$, and let $S = (x_1, y_1), \ldots, (x_m, y_m)$ be a cherry-picking sequence for N. Because N is binary, it can be reconstructed from S by starting with the one-leaf tree with leaf y_m and reattaching the pairs from S in reverse order[1]. Now label the root node ρ with $t(\rho) = 0$, each leaf l_j with $t(l) = m + j$, and each internal node v added when reattaching the pair (x_i, y_i) with $t(v) = m + 1 - i$.

We show that t is indeed an HGT-consistent labelling of N. When adding a pair to a network, the two newly introduced nodes are not above any existing internal nodes, and have a greater labelling than any other existing internal nodes. Thus any internal node has a label greater than or equal to that of any of its parents. Adding to the fact that we label the leaves so that they have labels of at least $m + 1$ and internal nodes have labels of at most m, we have that for all arcs (u, v), $t(u) \leq t(v)$. The labelling of the leaves also means that two nodes in the network have the same label under t only if they are a parent–child pair where the parent is a tree node and the child is a reticulation. Therefore, Property 1 of the HGT-consistent labelling is satisfied. To see that t satisfies Property 2, we look at tree nodes and reticulations separately. A tree node u has two children, one of which is possibly added to the network at the same time as u. The other child v of u is either a leaf or an internal node that is added to the network after u has been added. But this would mean that $t(u) < t(v)$. On the other hand, a reticulation r has one child c. Every reticulation node is added to the network with one of its (non-reticulation) parents; this means that c is either a leaf or an internal node that is added to the network after r has been added. This implies $t(r) < t(c)$. So Property 2 is satisfied. Finally, to see that Property 3 is also satisfied, consider a reticulation r with parents u and v. The reticulation r must have been added to the network with one of its parents, say u, so that $t(u) = t(r)$. This means that the node v was already in the network when r was added; due to how we have defined t, we must have that $t(v) < t(r)$. Thus t satisfies Property 3, and therefore it is an HGT-consistent labelling.

In addition, this gives a labelling t in which each label is used at most twice. Indeed for each $i \in \{1, \ldots, m\}$, the nodes with label i are added to the network when (x_i, y_i) is reattached, and for each such reattachment, at most two nodes are added to the network. Observe that under this construction, if two nodes have the same label, then they must be a parent-child pair where the parent is a tree node and the child a reticulation. □

A proof of the following lemma was sketched in Janssen and Liu (2021). We give a more rigorous proof here.

[1] We refer the interested reader to Janssen and Murakami (2021) for more information on this construction.

Lemma 2 *Each binary network that admits an HGT-consistent labelling is orchard.*

Proof Suppose a binary network N admits an HGT-consistent labelling t. We prove that N is orchard by proving that each network that admits an HGT-consistent labelling which has at least one internal node must contain a cherry or reticulated cherry. Moreover, after reducing such a pair, the resulting network still admits an HGT-consistent labelling. Therefore, any HGT-consistent network can be reduced to a tree with one leaf and is thus orchard.

Let x be an internal node with $t(x)$ maximal. If x is a reticulation, then its child l must be a leaf, and one of its parents p has label $t(p) = t(x)$, by Property 3. The node p cannot be a reticulation as this would contradict Property 2 of HGT-consistent labellings. Hence, the other child l' of p must be a leaf as well, and the reducible pair (l, l') is a reticulated cherry in N. Reducing this reticulated cherry, we obtain a new network N', which still admits an HGT-consistent labelling $t' = t|_{V(N')}$. If x is a tree node, then either both of its children are reticulations, one of its children v is a reticulation node, or both its children are leaves. In the first case, the reticulation children v_1 and v_2 must have labels $t(v_1) = t(v_2) = t(x)$, since x has the greatest label out of internal nodes and internal nodes have label greater than or equal to that of their parents, by Property 1. But this contradicts Property 2, so this case is not possible. In the second case, the reticulation child has label $t(v) = t(x)$, and we can reduce the reticulated cherry involving x and v as in the previous case. In the third case, x has two leaf children, which must thus form a cherry. After reducing this cherry, the network still admits an HGT-consistent labelling, which can be obtained by restricting t to the remaining nodes. □

A direct consequence of these lemmas is the following new characterization of orchard networks as networks that admit an HGT-consistent labelling.

Theorem 1 *A binary network N is orchard if and only if it admits an HGT-consistent labelling.*

The next corollary, which was also shown in Huber et al. (2019) and van Iersel et al. (2021b), follows from Observation 1 and Theorem 1.

Corollary 1 *The class of binary orchard networks is contained in the class of binary tree-based networks.*

This means in particular that orchard networks have a base tree.

3.2 Non-binary Orchard Networks

By recalling a key lemma from van Iersel et al. (2021b) regarding non-binary orchard networks and their binary resolutions, we extend the HGT-consistent labelling characteristics to non-binary orchard networks.

Lemma 3 (Lemma 11 of van Iersel et al. (2021b)) *A non-binary network N is orchard if and only if some binary resolution of N is orchard.*

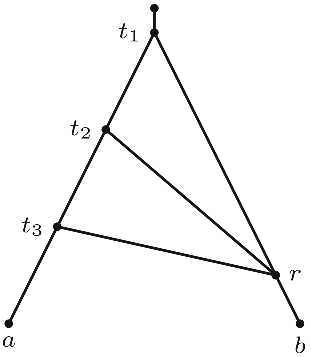

Fig. 4 A non-binary orchard network on $\{a, b\}$ that does not admit a labelling on the nodes that adheres to the 'reticulations have the same labels as all but one of its parents' rule. Indeed, under this rule, exactly two of t_1, t_2, t_3 must have the same label, but no labelling can satisfy this rule. Therefore, this labelling rule does not fully characterize the class of non-binary orchard networks

Theorem 2 *A non-binary network N is orchard if and only if some binary resolution of N admits an HGT-consistent labelling.*

Proof The combination of Theorem 1 and Lemma 3 immediately gives the claim. □

Ideally, we would like to extend this characterization by finding a direct labelling of non-binary networks that captures orchard networks, with a meaningful biological interpretation. A natural generalization would be to consider a labelling where every reticulation is contemporaneous with all but one parent in the network. This implies the existence of a base tree, for which all arcs that are not in the base tree are horizontal arcs. Unfortunately, this characterization does not fully capture the class of non-binary orchard networks. Figure 4 gives an example of a non-binary orchard network that does not admit a labelling under this property.

4 Orchard Network Space

We prove that the space of binary orchard networks is connected using a strategy that is reminiscent of the proofs of connectedness in Janssen (2021) for local head moves and in Erdős et al. (2021) for tree-based networks. We first move all reticulations to the top of the network and then change the pendant trees below these reticulations. Before we prove connectedness, we first give the formal definitions and introduce some structures and subgraphs we use in the proofs.

4.1 Rearrangement Moves

We start by defining rSPR and rNNI moves on phylogenetic networks. Intuitively, rSPR moves can be seen as moving either the head or the tail of an arc and rNNI moves are local rSPR moves.

Definition 7 (See Fig. 5) Let N be a binary network with an arc (z, w) and an arc e with endpoints x and y (either $e = (x, y)$ or $e = (y, x)$), and let p and c be, respectively, the parent and child of x that are not y. The *rSPR move* $(p, x, c) \xrightarrow{e} (z, w)$ consists of the following: replace the arcs (p, x), (x, c), and (z, w) with the arcs (p, c), (z, x), and (x, w). If $\{p, c\} \cap \{z, w\} \neq \emptyset$, then the move is an *rNNI move*.

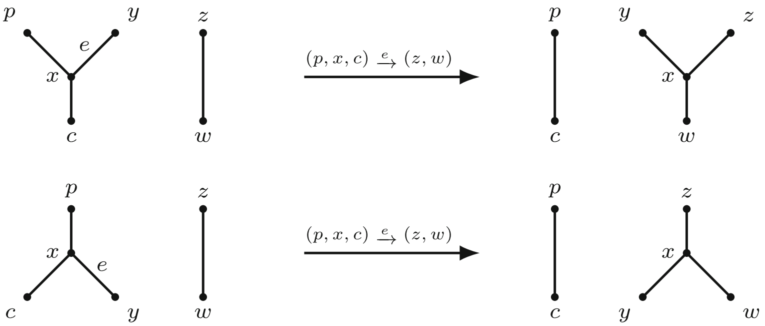
p
y
e
x
c
z
w
$(p,x,c) \xrightarrow{e} (z,w)$
p
c
y
z
x
w
p
x
e
c
y
z
w
$(p,x,c) \xrightarrow{e} (z,w)$
p
c
z
x
y
w

Fig. 5 Application of an rSPR move. The top figure shows the case that the arc *e* being moved is a reticulation arc, the bottom figure the case that *e* is a tree arc

An rSPR (or rNNI) move is *valid* if the resulting graph is a network. We first show that this holds automatically for orchard networks if the resulting graph has an HGT-consistent labelling. For that, we need to define HGT-consistent labellings on more general directed graphs. Note that the following definition is equivalent to Definition 6 if D is a network.

Definition 8 Let $D = (V, A)$ be a directed graph that may contain parallel arcs, with indegree and outdegree at most 2 and total degree at most 3. An *HGT-consistent labelling* of D is a labelling $t : V \to \mathbb{R}$ such that:

1. For all arcs (u, v), $t(u) \leq t(v)$ and equality is only allowed if v has indegree 2.
2. For each node u with at least one child, there is a child v of u such that $t(u) < t(v)$.
3. For each node r with two parents u and v, exactly one of $t(u) = t(r)$ and $t(v) = t(r)$ holds.

Lemma 4 *Let N be a binary orchard network and N' the result of an rSPR move on N. If N' admits an HGT-consistent labelling, then N' is a network and hence the rSPR move is valid.*

Proof Suppose the rSPR move is $(p, x, c) \xrightarrow{e} (z, w)$, where e is the arc incident on x that is not incident on p nor c. Let t be an HGT-consistent labelling of N'.

First note that N' cannot contain parallel arcs say from a to b. Indeed, this would make b into a reticulation node, and one of its parents must have the same label, so $t(a) = t(b)$. However, at least one child of a must have a larger label than a by Property 2 of Definition 8, so $t(a) < t(b)$, a contradiction.

In addition, N' cannot contain a directed cycle. Suppose for a contradiction that N' does contain a directed cycle. Then, by Property 1 of Definition 8, all nodes in this cycle must have the same label, and they must be reticulation nodes. However, reticulation nodes only have one outgoing arc, and the head of this arc must have a strictly larger label, a contradiction. Hence, N' cannot contain a directed cycle.

Since N' does not contain parallel arcs or directed cycles, and rSPR moves do not change the degrees, if follows that N' is a network and hence the rSPR move is valid. □

If the result of a valid rNNI move on N is a network that is isomorphic to network N' (respecting leaf labels), then we say that N *can be transformed into* N' using one rNNI move. It is not too difficult to observe from the definition that rNNI moves are symmetric, i.e., N can be transformed into N' using one rNNI move if and only if N' can be transformed into N using one rNNI move.

Definition 9 The *rNNI space of orchard networks* with n leaves and k reticulations is the graph $\mathrm{Orch}(n, k)$, whose nodes are binary orchard networks, and there is an arc between two networks if one can be transformed into the other in one rNNI move.

4.2 Connectedness of the rNNI Space of Orchard Networks

In this section, we prove that $\mathrm{Orch}(n, k)$ is connected for all n and k. The main idea of the proof is to show that we can transform any orchard network into some canonical network in which all reticulations are stacked just below the root of the network. This is formalized as follows. See also Fig. 6.

Definition 10 Let N be a binary orchard network where v_ρ is the child of the root. Then we say that N has *k reticulations at the top* if it contains two directed paths $v_\rho, a_1, \ldots, a_k$ and $v_\rho, b_1, \ldots, b_k$ where all a_i and b_j are distinct, and a set of k reticulation arcs $\{(x_i, y_i)\}_{i=1}^{k}$ where $\{x_i, y_i\} = \{a_i, b_i\}$, which are called the *horizontal arcs at the top*. In addition, there is no arc between the child of y_k and the child of x_k that is not y_k (because otherwise N would have $k + 1$ reticulations at the top). The reticulations $y_1, \ldots, y_k$ are *reticulations at the top*. If $(x_i, y_i) = (a_i, b_i)$ for all i, then we say that N has k reticulations *neatly* at the top.

We now show that the name "horizontal arcs at the top" is well chosen, i.e., that they are horizontal with respect to any HGT-consistent labelling.

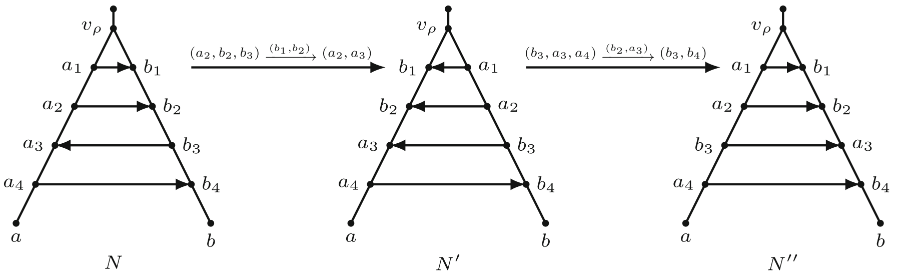

Fig. 6 The application of Lemma 6 twice in order to get all reticulations neatly at the top. Left: A network N on a set of taxa $\{a, b\}$ with four reticulations at the top. The network N contains a triangle at v_ρ with arcs (v_ρ, a_1), (v_ρ, b_1), and (a_1, b_1), where v_ρ is the child of the root. Middle: A binary network N' obtained by applying the valid rNNI move $(a_2, b_2, b_3) \xrightarrow{(b_1,b_2)} (a_2, a_3)$ on N. This illustrates Lemma 6, where the highest 2 horizontal arcs at the top are reoriented. Right: A network N'' by applying the valid rNNI move $(b_3, a_3, a_4) \xrightarrow{(b_2,a_3)} (b_3, b_4)$ to N'. The highest 3 horizontal arcs at the top are reoriented. The network N'' has four reticulations neatly at the top

Lemma 5 *If N is a binary orchard network with k reticulations at the top, and t an HGT-consistent labelling, then for each horizontal arc at the top (x_i, y_i), holds that $t(x_i) = t(y_i)$.*

Proof The proof is by induction on i. First consider $i = 1$. Since y_1 is a reticulation, we have, by the definition of HGT-consistent labelling, that either $t(y_1) = t(x_1)$ or $t(y_1) = t(v_\rho)$. We have $t(y_1) \geq t(x_1)$ because (x_1, y_1) is an arc. Furthermore, $t(x_1) > t(v_\rho)$ because x_1 is not a reticulation. Hence, we have $t(y_1) \geq t(x_1) > t(v_\rho)$. So $t(y_1) = t(v_\rho)$ is not possible and we must have $t(y_1) = t(x_1)$.

Now assume that $t(x_i) = t(y_i)$. We show that $t(x_{i+1}) = t(y_{i+1})$. First note that we do not know whether x_i is a parent of x_{i+1} and y_i of y_{i+1} or if x_i is a parent of y_{i+1} and y_i of x_{i+1}, but this does not matter for the proof since $t(x_i) = t(y_i)$. Since y_{i+1} is a reticulation, we have that either $t(y_{i+1}) = t(x_{i+1})$ or $t(y_{i+1}) = t(x_i) = t(y_i)$. We have $t(y_{i+1}) \geq t(x_{i+1})$ because (x_{i+1}, y_{i+1}) is an arc. Furthermore, $t(x_{i+1}) > t(x_i) = t(y_i)$ because x_{i+1} is not a reticulation. Hence, we have $t(y_{i+1}) \geq t(x_{i+1}) > t(x_i) = t(y_i)$. So $t(y_{i+1}) = t(x_i) = t(y_i)$ is not possible and we must have $t(y_{i+1}) = t(x_{i+1})$. □

Reorienting an arc (u, v) of a network N refers to modifying N into a network N' that is isomorphic to N with (u, v) replaced by (v, u). Note that reorienting any subset of the horizontal arcs at the top of a binary orchard network results in a binary orchard network, as the node labelling remains HGT-consistent. The following lemma shows that, if a network has k reticulations at the top and $k' \leq k$, then the highest k' horizontal arcs at the top (i.e., the arcs (x_i, y_i) for $i = 1, \ldots, k'$) can be reoriented using one rNNI move in total.

Lemma 6 (Lemma 5 of Janssen (2021)) *Let N be a binary orchard network with k reticulations at the top and $k' \leq k$. Then, using one rNNI move, the highest k' horizontal arcs at the top can be reoriented.*

We apply Lemma 6 twice in Fig. 6, where we use one rNNI move to reorient the highest two horizontal arcs at the top of a network, and another rNNI move to reorient the highest three horizontal arcs at the top of the resulting network. In this way, we use in total two rNNI moves to get all reticulations neatly at the top.

We prove that we can transform any orchard network into a network in which all reticulations are at the top. We first consider the case of moving a reticulation r that is part of a *triangle*, i.e., when there are arcs (v, p), (v, r), and (p, r). In this case, we also say that the triangle is *at* v. Observe that, by the same argument as in the first part of the proof of Lemma 5, $t(p) = t(r)$ for any HGT-consistent labelling t. The following lemma shows that such a reticulation r can either be moved up or directly to the top.

Lemma 7 *Let N be a binary orchard network with $k < r(N)$ reticulations at the top. Let t be an HGT-consistent labelling and r a reticulation not at the top minimizing $t(r)$. Suppose that r is part of a triangle (w, p), (w, r), (p, r). Then we can either reduce the number of nodes above r by* 1*, in* 2 *rNNI moves, or transform N into an orchard network with $k + 1$ reticulations at the top, in at most* 4 *rNNI moves.*

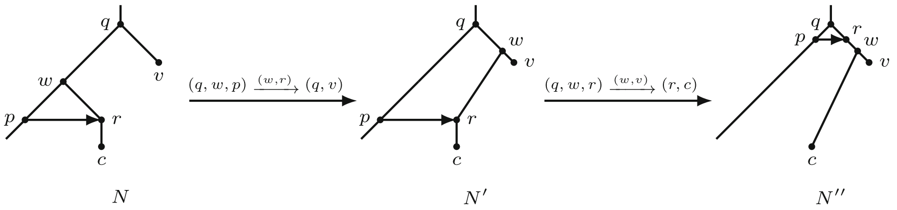

Fig. 7 The first case in the proof of Lemma 7, where q is not an endpoint of a horizontal arc at the top. Two rNNI moves are applied to the orchard network, which results in an orchard network where the triangle is at q. The heights of the nodes represent an HGT-consistent labelling

Proof Let t be an HGT-consistent labelling for N. Let q be the parent of w. First suppose that q is not an endpoint of a horizontal arc at the top. Then q is a tree node by our choice of r. Let v be its child other than w. In that case, the graph N' obtained by rNNI move $(q, w, p) \xrightarrow{(w,r)} (q, v)$ admits an HGT-consistent labelling (see Fig. 7)

$$t'(x) = \begin{cases} \min\{t(w), t(v)\} - \epsilon & \text{if } x = w; \\ t(x) & \text{otherwise,} \end{cases}$$

where $\epsilon > 0$ is small enough. In particular, choose ϵ so that $t'(q) < t'(w)$. Then, N' is an orchard network by Lemma 4 and Theorem 1. Let c be the child of r in N', and we apply to N' the move $(q, w, r) \xrightarrow{(w,v)} (r, c)$ to obtain the graph N''. The graph N'' is an orchard network, since it admits the HGT-consistent labelling

$$t''(x) = \begin{cases} t'(w) - \epsilon & \text{if } x \in \{p, r\}; \\ t'(x) & \text{otherwise,} \end{cases}$$

where again, $\epsilon > 0$ is very small. Observe that N'' contains a triangle at q consisting of the arcs (q, p), (q, r), and (p, r). Hence, we have reduced the number of nodes above r by 1, using 2 rNNI moves, and r is still a reticulation not at the top minimizing $t''(r)$.

Now suppose that q is an endpoint of a horizontal arc at the top. Then we can use the following rNNI moves to move (p, r) to the top. First, if q is not a reticulation, reorient the reticulation arc leaving q so that q becomes a reticulation, using one rNNI move (Lemma 6). Let s be the parent of q with $t(q) = t(s)$, let v be the child of s that is not q and let c be the child of r (see Fig. 8). Apply the following three moves: $(q, w, p) \xrightarrow{(w,r)} (s, q)$, $(s, w, q) \xrightarrow{(w,r)} (s, v)$, and $(s, w, r) \xrightarrow{(w,v)} (r, c)$. To see that the graphs after the three moves are orchard networks, note that the following are HGT-consistent labellings of the respective resulting graphs:

$$t'(x) = \begin{cases} \min\{t(w), t(v)\} - \epsilon & \text{if } x \in \{q, w\}; \\ t(x) & \text{otherwise,} \end{cases}$$

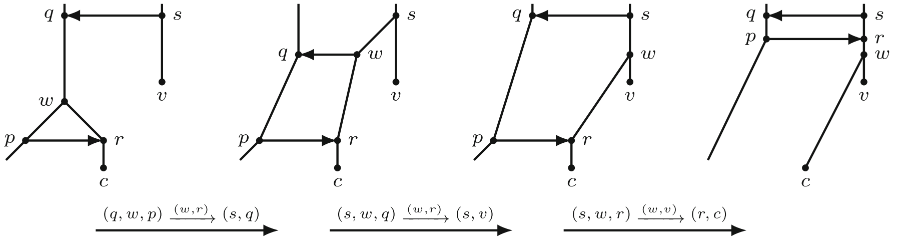
q
s
w
v
p
r
c
q
w
s
v
p
r
c
q
s
w
v
p
r
c
q
s
p
r
w
v
c
$(q, w, p) \xrightarrow{(w,r)} (s, q)$
$(s, w, q) \xrightarrow{(w,r)} (s, v)$
$(s, w, r) \xrightarrow{(w,v)} (r, c)$

Fig. 8 The second case in the proof of Lemma 7, where q is a reticulation at the top. Three rNNI moves are applied to the orchard network, which results in an orchard network where reticulation r is also at the top. If q is not a reticulation in the original network, one extra rNNI move is applied first to make q a reticulation

$$t''(x) = \begin{cases} t'(s) & \text{if } x = q; \\ t'(s) + \delta & \text{if } x = w; \\ t'(x) & \text{otherwise,} \end{cases}$$

and

$$t'''(x) = \begin{cases} t''(w) - \gamma & \text{if } x \in \{p, r\}; \\ t''(x) & \text{otherwise,} \end{cases}$$

where $0 < \gamma < \delta < \epsilon$ are very small. Observe that reticulation r is now at the top and (p, r) a horizontal arc at the top (see Fig. 8). Hence, the resulting graph obtained using at most 4 rNNI moves is an orchard network with $k + 1$ reticulations at the top. □

We are now ready to prove that we can move all reticulations to the top.

Lemma 8 *Let N be a binary orchard network with $k < r(N)$ reticulations at the top and n leaves. Then, using at most $2n$ rNNI moves, N can be transformed into a binary orchard network with $k + 1$ reticulations at the top.*

Proof Let N be an orchard network with HGT-consistent labelling t such that two nodes have the same label only if they are a parent–child pair for which the parent is a tree node and the child a reticulation (which exists by Lemma 1). Let r be a reticulation not at the top minimizing $t(r)$. Let p be the parent of r with $t(r) = t(p)$ and u the other parent of r. Observe that, since $t(r) = t(p)$, p must have a second child with larger label. Hence, p cannot be a reticulation. Let q be the parent of p. In addition, let y be a lowest common ancestor (LCA) of u and q in N.

First suppose $t(u) > t(q)$ (and hence $u \neq q$). Note that u is either a tree node or a reticulation at the top, by the choice of r. If u is a tree node, we can apply the move $(u, r, c) \xrightarrow{(p,r)} (s, u)$ where s is the parent of u, v is the child of u other than r, and c is the child of r (Fig. 9). Then

$$t'(x) = \begin{cases} t(u) & \text{if } x \in \{p, r\}, \\ t(u) + \epsilon & \text{if } x = u, \\ t(x) & \text{otherwise,} \end{cases}$$

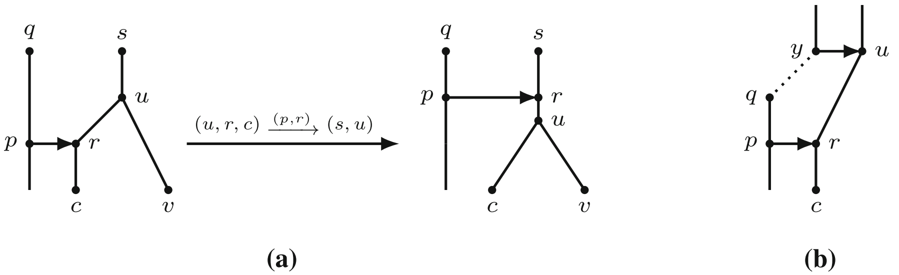

Fig. 9 Illustration of the proof of Lemma 8 for the case $t(u) > t(q)$. **a** When u is a tree node, the head of the reticulation arc (p, r) is moved up. **b** When u is a reticulation at the top, an LCA y of u and q is a parent of u with $t(y) = t(u)$, leading to a contradiction to the assumption $t(u) > t(q)$

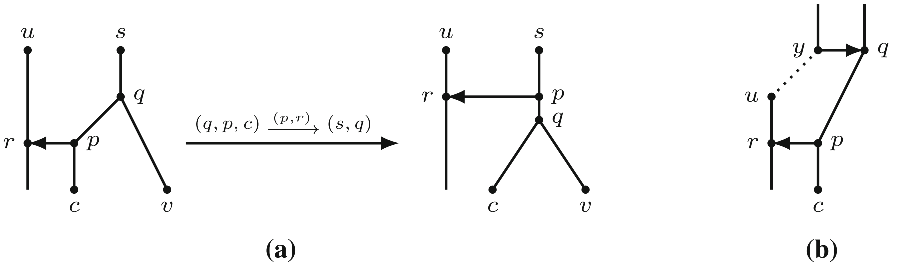

Fig. 10 Illustration of the proof of Lemma 8 for the case $t(q) > t(u)$. **a** When q is a tree node, the tail of the reticulation arc (p, r) is moved up. **b** When q is a reticulation at the top, an LCA y of u and q is a parent of q with $t(y) = t(q)$, leading to a contradiction to the assumption $t(q) > t(u)$

with $\epsilon > 0$ small enough, is an HGT labelling of the resulting network, which is therefore an orchard network by Lemma 4. In the resulting network, the number of nodes above r is reduced by 1.

If u is a reticulation at the top, then y is the parent of u with $t(y) = t(u)$. However, because y is above q, we have $t(y) \leq t(q)$ and so $t(u) \leq t(q)$, contradicting our assumption $t(u) > t(q)$.

The case that $t(u) < t(q)$ is symmetric. We can argue as in the previous paragraph but replacing u, r by q, p respectively, see Fig. 10.

The last case is that $t(u) = t(q)$. We claim that $u = q = y$. Otherwise, by our choice of the labelling t, we either have an arc (q, u) and u is a reticulation, or we have an arc (u, q) and q is a reticulation. In the first scenario, u must be a reticulation at the top by our choice of r. However, since r is a child of u and p a child of q, it follows that r is also a reticulation at the top, a contradiction. In the second scenario, q must be a reticulation at the top and we again obtain a contradiction, by the same reasoning. Therefore, we conclude that $u = q = y$. This means that there is a triangle (y, r), (y, p), (p, r) and we can apply Lemma 7 to either increase the number of reticulations at the top by 1 in at most 4 rNNI moves, or reduce the number of nodes above r by 1 in 2 rNNI moves.

It remains to bound the number of moves. We say that a node is a *base node* if it is an internal node of the base tree obtained by deleting all arcs (u, v) with $t(u) = t(v)$

and suppressing indegree-1 outdegree-1 nodes. Hence, an internal node is a base node precisely if it has no parent or child with the same label in the HGT-consistent labelling t. Define $a_b(v)$ as the number of base nodes above v. Hence, a reticulation r is at the top if and only if $a_b(r) = 2$. Since a rooted phylogenetic tree with n leaves has n internal nodes, the network N has n base nodes.

In the proof above, we consider a reticulation r not at the top minimizing $t(r)$. Consequently, all nodes above r are base nodes or endpoints of horizontal arcs at the top. If r is not part of a triangle, we reduce the number of nodes above r, and hence $a_b(r)$, by 1 in one rNNI move. If r is part of a triangle, we achieve the same in 2 rNNI moves, or move r to the top in at most 4 rNNI moves. We end with $a_b(r) = 2$. Hence, we need at most 2 rNNI moves per base node, except the root and its child, plus at most 4 additional moves. The total number of rNNI moves required to move r to the top is at most $2(n-2)+4 = 2n$. □

The next step is to move all but one of the leaves to one side of the network, see Fig. 11.

Lemma 9 *Let N be a binary network on n leaves with $r(N)$ reticulations at the top. Let l be a leaf below the head of the lowest horizontal arc at the top. Then, using at most $2n-4$ rNNI moves, N can be transformed into a network with $r(N)$ reticulations at the top, where l is the only leaf below the head of the lowest horizontal arc at the top.*

Proof Let (x, y) be the lowest horizontal arc at the top. Let u_0 be the child of x other than y. Let $y, u_1, \ldots, u_m = l$ be the unique directed path from y to l. Let v_i be the child of u_i other than u_{i+1} or l, for $i = 1, \ldots, m-1$. We apply the following sequence of rNNI moves: $(y, u_i, u_{i+1}) \xrightarrow{(u_i, v_i)} (x, y)$ and $(x, u_i, y) \xrightarrow{(u_i, v_i)} (x, u_{i-1})$ for all $i = 1, \ldots, m-1$. See Fig. 12. It can easily be checked that the following maps $t_1^{(i)}$ and $t_2^{(i)}$ are HGT-consistent labellings of the graphs obtained after the first and, respectively, second rNNI move, for $i = 1, \ldots, m-1$. Letting $t = t_2^{(0)}$ denote an HGT-labelling for N, we have

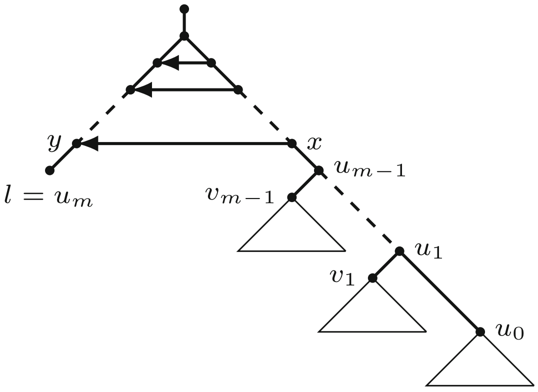

Fig. 11 The network used in the proof of Lemma 9 and Theorem 3, in which the triangles below $u_0, v_1, \ldots, v_{m-1}$ indicate trees rooted at those nodes

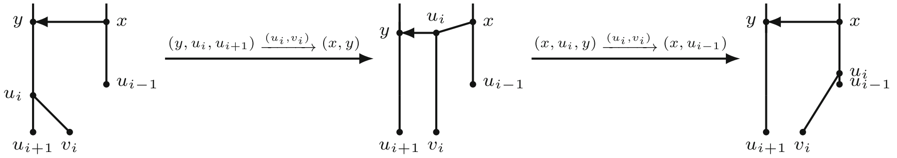

Fig. 12 Illustration of the rNNI moves in the proof of Lemma 9

$$t_1^{(i)}(z) = \begin{cases} t(x) + \epsilon & \text{if } z \in \{u_i, y\}; \\ t_2^{(i-1)}(z) & \text{otherwise,} \end{cases}$$

and

$$t_2^{(i)}(z) = \begin{cases} t(x) & \text{if } z = y; \\ \min\{t(u_0), t(u_1)\} - i\epsilon & \text{if } z = u_i; \\ t_1^{(i)}(z) & \text{otherwise,} \end{cases}$$

where $\epsilon > 0$ is very small. Hence, all intermediate graphs, as well as the final one, are orchard networks. The final network (see Fig. 11) still has $r(N)$ reticulations at the top, and l is the child of y. Moreover, as $m \leq n - 2$, the sequence of moves contains at most $2(n-2) = 2n - 4$ rNNI moves. □

We are now ready to prove that the space of orchard networks is connected by rNNI moves, for a fixed number of reticulations and set of taxa. We basically prove this theorem by showing that each orchard network can be transformed into one particular network (see Fig. 11). This shows that, in the space of orchard networks with the same number of leaves and reticulations, there is a path from each node to one given node, and, thus, the space is connected.

Theorem 3 *The space of binary orchard networks with n leaves and k reticulations is connected under rNNI moves with diameter at most* $4kn + n\lceil\log_2(n)\rceil + 2k + 6n - 8$ *which is* $O(kn + n\log(n))$.

Proof Consider any two binary orchard networks N_1, N_2 with k reticulations on the same set of n taxa. Fix an arbitrary leaf l. In each of the two networks, we do the following. First move all reticulations to the top using the moves in Lemma 8. For each reticulation, this takes at most $2n$ moves, so at most $2kn$ rNNI moves in total. Using at most k rNNI moves, the horizontal arcs at the top can be reoriented such that the network has k reticulations neatly at the top, and such that l is below the head of the lowest horizontal arc at the top, by Lemma 6. Using at most $2n - 4$ moves, we can move all leaves except l below the tail x of the lowest horizontal arc at the top, by Lemma 9. This is done on both N_1 and N_2, so the contribution toward the number of moves up until this point is $2(2kn + k + 2n - 4)$.

Let $\text{diam}_{\text{rNNI}}(n, k)$ denote the diameter of $\text{Orch}(n, k)$. Let T_i denote the subtree rooted at the child of x other than y in N_i (i.e., the subtree rooted at u_{m-1} in Fig. 11). Note that T_i contains all leaves except l. We can change T_1 into T_2 using at most $\text{diam}_{\text{rNNI}}(n-1,0) \leq 2n + n\lceil\log_2(n)\rceil$ moves (Li et al. 1996; Erdős et al. 2021). It follows that $\text{diam}_{\text{rNNI}}(n,k) \leq 2(2kn+k+2n-4) + \text{diam}_{\text{rNNI}}(n-1,0) \leq 4kn + n\lceil\log_2(n)\rceil + 2k + 6n - 8 = O(kn + n\log(n))$ moves. □

Since rSPR moves are generalizations of rNNI moves, the following result follows immediately from Theorem 3.

Corollary 2 *The space of binary orchard networks with n leaves and k reticulations is connected under rSPR moves with diameter at most* $4kn + n\lceil\log_2(n)\rceil + 2k + 6n - 8$ *which is* $O(kn + n\log(n))$

5 Discussion

In this paper, we have shown that binary orchard networks can be characterized as networks with an HGT-consistent labelling, meaning that they can be obtained from a tree by inserting horizontal arcs. Hence, this class of networks, which was introduced for its computational benefits, also has a biological interpretation.

This does not mean that orchard networks can only be applied in situations where reticulations represent HGT events. Although orchard networks can be drawn as HGT networks, they can also be drawn differently. Hence, orchard networks may still be useful in applications where reticulations represent hybridizations or other reticulate events. Restricting to orchard networks may exclude some scenarios in that case, but the nice mathematical properties may outweigh that, especially when they can be exploited to develop efficient algorithms.

We have also shown that non-binary orchard networks can be characterized using this labelling on a binary refinement. However, this characterization is less satisfying as it does not specify which binary refinement to use. We leave it as an open question to find a characterization that uses a labelling directly on the nonbinary network.

To show the mathematical utility of the new characterization, we have used this new characterization to prove that the space of orchard networks is connected under rNNI moves. As mentioned, this may prove important, because some statistical network generators introduce reticulations as HGT events (Pons et al. 2019), which naturally leads to orchard networks. Hence, if such generators are used as a prior in a Bayesian method for network inference, the prior probability of all non-orchard networks will be zero, so it is important to know that the space of orchard networks is connected.

To see whether it makes sense computationally to restrict to orchard networks, it would also be interesting to know whether our upper bound on the diameter of the space of orchard networks of order $O(kn + n\log n)$ is asymptotically tight. Indeed, smaller diameters can be favorable when deciding on a network space to search through, as it could mean a shorter mixing time for Markov Chain Monte Carlo methods (Klawitter 2020). The bound is, in any case, close to the bound for tree-based networks ($O(kn + k^2 + n\log n)$ Erdős et al. 2021). It is clear the asymptotic bound cannot get smaller than $O(n\log n)$, as this is tight for trees, but it may be possible to remove or reduce the kn part.

Note that our results are only given for rNNI moves, and not for local tail or local head moves separately, as is done for tree-based networks in Erdős et al. (2021). It might be true that the space of orchard networks is also connected under distance-1 tail moves, or distance-2 head moves, but our proof does not imply this, as we have used a mix of distance-1 tail moves and distance-1 head moves. It would be of interest to investigate this further.

References

Bai A, Erdős PL, Semple C, Steel M (2021) Defining phylogenetic networks using ancestral profiles. Math Biosci 332:108537

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

Baroni M, Semple C, Steel M (2006) Hybrids in real time. Syst Biol 55(1):46–56

Blais C, Archibald JM (2021) The past, present and future of the tree of life. Curr Biol 31(7):314–321

Bordewich M, Linz S, Semple C (2017) Lost in space? Generalising subtree prune and regraft to spaces of phylogenetic networks. J Theor Biol 423:1–12

Bordewich M, Semple C, Tokac N (2018) Constructing tree-child networks from distance matrices. Algorithmica 80(8):2240–2259

Borst S, van Iersel L, Jones M, Kelk S (2020) New FPT algorithms for finding the temporal hybridization number for sets of phylogenetic trees. arXiv preprint arXiv:2007.13615

Bouckaert R, Vaughan TG, Barido-Sottani J, Duchêne S, Fourment M, Gavryushkina A, Heled J, Jones G, Kühnert D, De Maio N et al (2019) BEAST 2.5: an advanced software platform for Bayesian evolutionary analysis. PLoS Comput Biol 15(4):1006650

Cardona G, Rosselló F, Valiente G (2008) Comparison of tree-child phylogenetic networks. IEEE/ACM Trans Comput Biol Bioinform 6(4):552–569

Cardona G, Llabrés M, Rosselló F, Valiente G (2008) A distance metric for a class of tree-sibling phylogenetic networks. Bioinformatics 24(13):1481–1488

Cardona G, Pons JC, Rosselló F (2015) A reconstruction problem for a class of phylogenetic networks with lateral gene transfers. Algorithms Mol Biol 10(1):1–15

Elworth RL, Ogilvie HA, Zhu J, Nakhleh L (2019) Advances in computational methods for phylogenetic networks in the presence of hybridization. In: Warnow T (ed) Bioinformatics and phylogenetics. Springer, Cham, pp 317–360

Erdem E, Lifschitz V, Ringe D (2006) Temporal phylogenetic networks and logic programming. Theory Pract Logic Program 6(5):539–558

Erdős PL, Semple C, Steel M (2019) A class of phylogenetic networks reconstructable from ancestral profiles. Math Biosci 313:33–40

Erdős PL, Francis A, Mezei TR (2021) Rooted NNI moves and distance-1 tail moves on tree-based phylogenetic networks. Discrete Appl Math 294:205–213

Fischer M, Galla M, Herbst L, Long Y, Wicke K (2020) Classes of tree-based networks. Vis Comput Ind Biomed Art 3(1):1–26

Francis AR, Steel M (2015) Which phylogenetic networks are merely trees with additional arcs? Syst Biol 64(5):768–777

Huber KT, van Iersel L, Janssen R, Jones M, Moulton V, Murakami Y, Semple C (2019) Rooting for phylogenetic networks. arXiv preprint arXiv:1906.07430

Humphries PJ, Linz S, Semple C (2013) On the complexity of computing the temporal hybridization number for two phylogenies. Discrete Appl Math 161(7–8):871–880

Janssen R (2021) Heading in the right direction? Using head moves to traverse phylogenetic network space. J Graph Algorithms Appl 25:263–310

Janssen R, Liu P (2021) Comparing the topology of phylogenetic network generators. J Bioinform Comput Biol 19(06):2140012

Janssen R, Murakami Y (2021) On cherry-picking and network containment. Theor Comput Sci 856:121–150

Klawitter J (2020) Spaces of phylogenetic networks. PhD thesis, University of Auckland

Li M, Tromp J, Zhang L (1996) On the nearest neighbour interchange distance between evolutionary trees. J Theor Biol 182(4):463–467

Markin A, Anderson TK, Vadali VSKT, Eulenstein O (2019) Robinson–Foulds reticulation networks. In: Proceedings of the 10th ACM international conference on bioinformatics, computational biology and health informatics, pp 77–86

Moret BM, Nakhleh L, Warnow T, Linder CR, Tholse A, Padolina A, Sun J, Timme R (2004) Phylogenetic networks: modeling, reconstructibility, and accuracy. IEEE/ACM Trans Comput Biol Bioinform 1(1):13–23

Nakhleh L, Ruths D, Wang L-S (2005) RIATA-HGT: a fast and accurate heuristic for reconstructing horizontal gene transfer. In: International computing and combinatorics conference, pp 84–93. Springer

Pons JC, Scornavacca C, Cardona G (2019) Generation of level-*k* LGT networks. IEEE/ACM Trans Comput Biol Bioinform 17(1):158–164

van Iersel L, Moulton V (2014) Trinets encode tree-child and level-2 phylogenetic networks. J Math Biol 68(7):1707–1729

van Iersel L, Janssen R, Jones M, Murakami Y, Zeh N (2021) A unifying characterization of tree-based networks and orchard networks using cherry covers. Adv Appl Math 129:102222

van Iersel L, Janssen R, Jones M, Murakami Y, Zeh N (2021) A practical fixed-parameter algorithm for constructing tree-child networks from multiple binary trees. Algorithmica. To appear. arXiv preprint arXiv:1907.08474 (2019)

Wen D, Yu Y, Zhu J, Nakhleh L (2018) Inferring phylogenetic networks using PhyloNet. Syst Biol 67(4):735–740