# Heading in the right direction? Using head moves to traverse phylogenetic network space

Janssen, R.

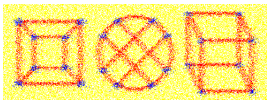
**DOI**
10.7155/jgaa.00559

**Publication date**
2021

**Document Version**
Final published version

**Published in**
Journal of Graph Algorithms and Applications

**Citation (APA)**
Janssen, R. (2021). Heading in the right direction? Using head moves to traverse phylogenetic network space. *Journal of Graph Algorithms and Applications*, *25*(1), 263-310. https://doi.org/10.7155/jgaa.00559


Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 25, no. 1, pp. 263–310 (2021)
DOI: 10.7155/jgaa.00559

# Heading in the right direction? Using head moves to traverse phylogenetic network space

*Remie Janssen*

Delft Institute of Applied Mathematics, Delft University of Technology,
Mekelweg 4, 2628 CD Delft, The Netherlands

| | | |
|---|---|---|
| Submitted: February 2019 | Reviewed: March 2020 | Revised: March 2020 |
| Reviewed: July 2020 | Revised: August 2020 | Accepted: March 2021 |
| Final: March 2021 | Published: April 2021 | |
| Article type: Regular paper | Communicated by: Fabio Vandin | |

**Abstract.** Head moves are a type of rearrangement moves for phylogenetic networks. They have primarily been studied as part of other types of moves, such as rSPR moves. Here, we study head moves as a type of moves on themselves. We show that the tiers ($k > 0$) of phylogenetic network space are connected by *local* head moves. Then, we show tail moves and head moves are closely related: sequences of tail moves can be converted into sequences of head moves and vice versa, changing the length by at most a constant factor. Because the tiers of network space are connected by rSPR moves, this gives a second proof of the connectivity of these tiers. Furthermore, we show that these tiers have small diameter by reproving the connectivity a third time. As the head move neighbourhood is small in general, this makes head moves a good candidate for local search heuristics. Finally, we prove that finding the shortest sequence of head moves between two networks is NP-hard.

## 1 Introduction

For biologists, it is vital to know the evolutionary history of the species they study. Evolutionary histories are, among other things, needed to find the reservoir/initial infection for some disease [e.g., 10, 23], or to learn about the evolution of genes, giving us insight in how they work [e.g., 27, 31, 18, 11, 1].

These histories are traditionally represented as phylogenetic trees. This focus on trees has recently started shifting towards phylogenetic networks, in which more biological processes can

Research funded in part by the Dutch Research Council (NWO), Vidi grant 639.072.602 of dr. Leo van Iersel.

*E-mail address:* R.Janssen-2@tudelft.nl (Remie Janssen)

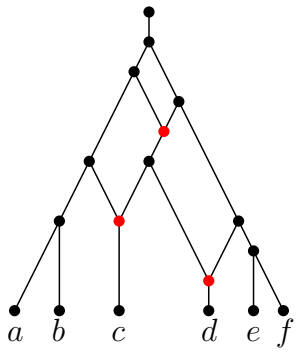



Figure 1: A tier-3 phylogenetic network with six leaves (i.e. taxa) at the bottom, and the root (ancestral taxon) at the top. Edges are directed downwards, showing the passing of time. The red nodes are the three reticulations (i.e. reticulate evolutionary events).

be represented (Figure 1). These biological processes, such as hybridization and horizontal gene transfer, are collectively known as reticulate evolutionary events, because they cause a reticulate structure in the representation of the history.

Phylogenetic networks are generally harder to reconstruct than trees. Reconstruction most often takes the form of an optimization problem. Certain phylogenetic optimization problems can still be solved quickly, even if they involve networks [e.g., 30]. However, most of these problems are already hard when they involve trees, and they do not get easier when networks are introduced as well (e.g., ML based reconstruction [25]). In such cases, some kind of local search is often employed [6, 22, 24, 23]. This is a process where the goal is to find a (close to) optimal tree by exploring the space of trees making only small changes, called rearrangement moves.

Several rearrangement moves have long been studied for phylogenetic trees. The most prominent ones are Nearest Neighbour Interchange (NNI), Subtree Prune and Regraft (SPR), and Tree Bisection and Reconnection (TBR) [6, 26]. The last decade has seen a surge in research on rearrangement moves for phylogenetic networks based on these moves for trees. There are several ways of generalizing the moves to networks, which means there is a relatively large number of moves for networks, including rSPR moves [9], rNNI moves [9], SNPR moves [2], tail moves [17], and head moves (Figure 2).

All these moves are similar in that they only change the location of one edge in the network. Hence, a lot of properties of the search spaces defined by different moves can be related. In this paper we study relations of such properties for the spaces corresponding to tail and head moves. The results we obtain can also be used in the study of rSPR moves, because rSPR moves consist of tail moves and head moves; and it can be used for the study of SNPR moves for the same reason.

We start by proving that each tier of phylogenetic network space is connected by distance-2 head moves, but not by distance-1 head moves (Section 3). Then, in Section 4 we prove that each head move can be replaced by at most 16 tail moves, and each tail move can be replaced by at most 15 head moves. This not only reproves connectivity of tiers of head move space, but also gives relations for distances between two networks measured by different rearrangement moves (rSPR,

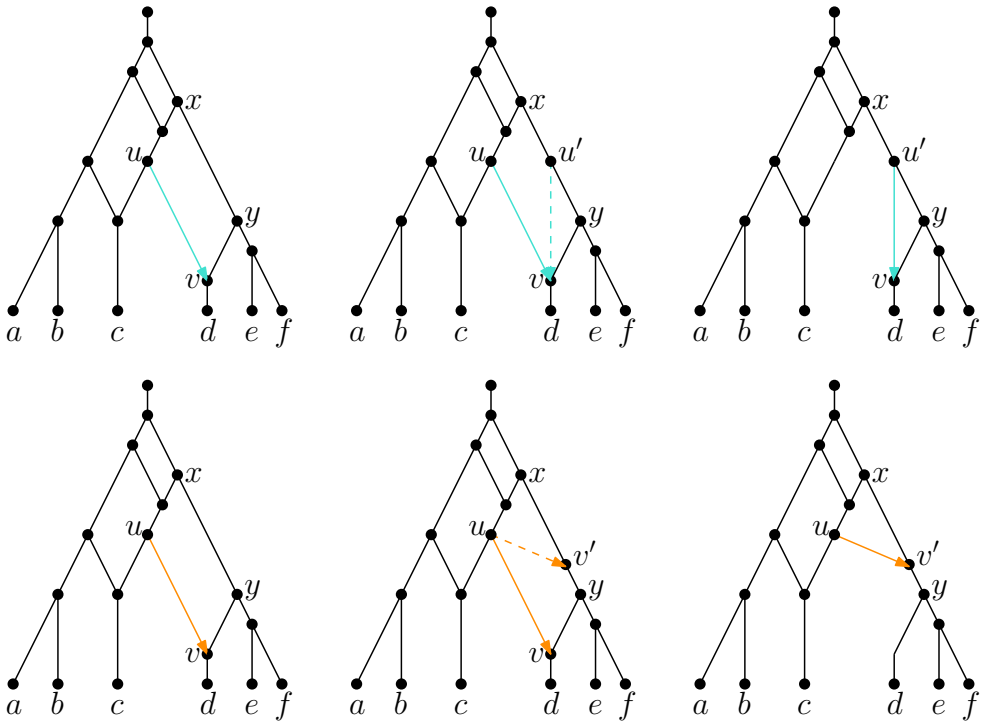



Figure 2: Top: the tail move $(u, v)$ to $(x, y)$; Bottom: the head move $(u, v)$ to $(x, y)$. On the left, the starting networks in which the moving edges are coloured. The right networks are the resulting networks after the moves, with the moved edge coloured differently. The middle graph is a combination of the left and the right network, with the moving edge coloured differently. The solid coloured edge is the moving edge of the network before the move, the dashed coloured edge is the moving edge of the network after the move. We distinguish the moves with edge colours: blue is a tail move, orange is a head move.

head moves, and tail moves). In Section 5, we prove the upper bound $6n + 6k - 1$ for the diameter of tier-$k$ of network space with $n$ taxa. Lastly, in Section 6, we prove that computing the head moves distance between two networks is NP-hard.

# 2 Preliminaries

## 2.1 Phylogenetic networks

**Definition 1** *A* binary phylogenetic network *with leaves $X$ is a directed acyclic graph (DAG) $N$ with:*

- *one* root *(indegree-0, outdegree-1 node)*
- $|X| > 1$ leaves *(indegree-1, outdegree-0 nodes) bijectively labelled with the elements of $X$*

- *all other nodes are either* tree nodes *(indegree-1, outegree-2 nodes), or* reticulations *(indegree-2, outdegree-1 nodes).*

*Incoming edges of reticulation nodes are called* reticulation edges. *The* reticulation number *of $N$ is the number of reticulation nodes. The set of all networks with reticulation number $k$ is called* tier-$k$ *of phylogenetic network space.*

For simplicity, we will often refer to binary phylogenetic networks as phylogenetic networks or as networks. Phylogenetic trees are phylogenetic networks without reticulation nodes. Many phylogenetic problems start with a set of phylogenetic trees and ask for a 'good' network for this set of trees. This often means that the trees must be contained in this network in the following sense.

**Definition 2** Subdividing *an edge $(u, v)$ consists of deleting it and adding a node $x$ and edges $(u, x)$ and $(x, v)$. A* subdivision *of a digraph $G$ is any graph obtained from $G$ by repeatedly subdividing edges.*

*The reverse operation is* suppressing *an indegree-1, outdegree-1 node. Let $x$ be such a node with in-edge $(u, x)$ and out-edge $(x, v)$, then suppressing $x$ consists of removing the edges $(u, x)$ and $(x, v)$ and the node $x$, and then adding an edge $(u, v)$.*

**Definition 3** *A tree $T$ can be* embedded *in a network $N$ if there exists an $X$-labelled subgraph of $N$ that is a subdivision of $T$. We say $T$ is an embedded tree of $N$. The corresponding map, which sends nodes of $T$ to nodes of $N$ and edges of $T$ to directed paths in $N$, is called an* embedding *of $T$ into $N$.*

Because a network is a DAG, there are unambiguous ancestry relations between the nodes. We draw networks with their root at the top, and the leaves at the bottom. This induces the following terminology.

**Definition 4** *Let $u, v$ be nodes in a network $N$. Then we say:*

- *$u$ is* above *$v$, and $v$ is* below *$u$, if there is a directed path from $u$ to $v$ in $N$.*
- *$u$ is* directly above *$v$, and $v$* directly below *$u$, if there is an edge $(u, v)$ in $N$.*

*Similarly, we say an edge $(u, v)$ is above a node $w$ or above an edge $(w, z)$ if $v$ is above $w$. An edge $(u, v)$ is below a node $z$ or an edge $(w, z)$ if $u$ is below $z$.*

*Alternatively, if $u$ is above $v$, we say that $u$ is an* ancestor *of $v$, and if $u$ is directly above $v$, we say that $u$ is a* parent *of $v$ and $v$ a* child *of $u$. A* Lowest Common Ancestor (LCA) *of two nodes $u$ and $v$ is a node $z$ which is above both $u$ and $v$, such that there are no other such nodes below $z$.*

Unlike for trees, there may not be a unique LCA for two nodes in a network. An important substructure in a network is the triangle. This structure has a big impact on the rearrangements we can do in a network.

**Definition 5** *Let $N$ be a network and $t, s, r$ nodes of $N$. If there are edges $(t, s)$, $(t, r)$ and $(s, r)$ in $N$, then we say $t$, $s$ and $r$ form a* triangle *in $N$. We call $t$ the* top *of the triangle, $s$ the* side *of the triangle, and the reticulation $r$ the* bottom *of the triangle.*

The following observation describes one property of triangles that we will use frequently.

**Observation 1** *Let $N$ be a network with a triangle $u, s, v$, where $s$ is a tree node. Then reversing the direction of the edge $(s, v)$ gives a network $N'$. We say the* direction of the triangle is reversed. *This can be achieved by the distance-1 head move which moves $(u, v)$ to the outgoing edge of $s$ that is not part of the triangle (Figure 3).*

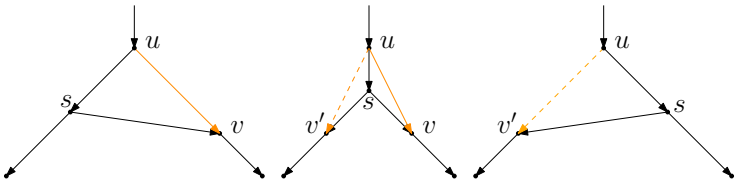



Figure 3: The distance-1 head move used to change the direction of a triangle.

**Definition 6** *Let $X = \{x_1, \ldots, x_n\}$ be an ordered set of labels. The caterpillar $C(X)$ is the tree defined by the Newick string*

$$(\cdots(x_1, x_2), x_3)\cdots, x_n);$$

## 2.2 Rearrangement moves

The main topics of this paper are head and tail moves, which are types of rearrangement moves on phylogenetic networks. Several types of moves have been defined for rooted phylogenetic networks. The most notable ones are tail moves [17], rooted Subtree Prune and Regraft (rSPR) and rooted Nearest Neighbour Interchange (rNNI) moves [9] and SubNet Prune and Regraft (SNPR) moves [2]. These moves typically change the location of one endpoint of an edge, or they remove or add an edge.

We now introduce the basic notions of head and tail moves, following the presentation of [17].

**Definition 7 (Head move)** *Let $e = (u, v)$ and $f$ be edges of a network. A* head move *of $e$ to $f$ consists of the following steps (Figure 4):*

1. *delete $e$;*
2. *subdivide $f$ with a new node $v'$;*
3. *suppress the indegree-1 outdegree-1 node $v$;*
4. *add the edge $(u, v')$.*

*Head moves are only allowed if the resulting digraph is still a network (Definition 1). We say that a head move is a* distance-$d$ head move *if, after step 2, a shortest path from $v$ to $v'$ in the underlying undirected graph has length at most $d + 1$ (counting the edges in the path).*

**Definition 8 (Tail move)** *Let $e = (u, v)$ and $f$ be edges of a network. A* tail move *of $e$ to $f$ consists of the following steps (Figure 5):*

1. *delete $e$;*

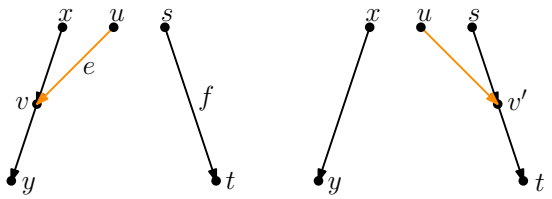



Figure 4: The head move $e = (u, v)$ from $(x, y)$ to $f = (s, t)$ described in Definition 7.

2. *subdivide $f$ with a new node $u'$;*

3. *suppress the indegree-1 outdegree-1 node $u$;*

4. *add the edge $(u', v)$.*

*Tail moves are only allowed if the resulting digraph is still a network (Definition 1). We say that a tail move is a* distance-$d$ tail move *if, after step 2, a shortest path from $u$ to $u'$ in the underlying undirected graph has length at most $d + 1$.*

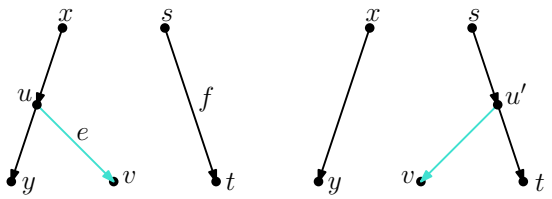



Figure 5: The tail move $e = (u, v)$ from $(x, y)$ to $f = (s, t)$ described in Definition 8.

The networks before and after a tail or head move always lie in the same tier. Hence, we say these moves are *horizontal moves*; this is to contrast them with moves that change the reticulation number, which we call *vertical moves*. Note that rSPR moves are horizontal moves as well. In fact, an rSPR move is either a head move or a tail move. Similarly, rNNI moves are distance-1 rSPR moves (i.e., distance-1 head or tail moves). SNPR moves, however, may be vertical as well: an SNPR move is either a tail move, or a vertical move that simply removes or adds an edge.

### 2.2.1 Validity of moves

As we want to use rearrangement moves to traverse network space, each move must result in a phylogenetic network. The definitions in the previous subsection enforce that this always happens for tail and head moves. In this paper, we often propose a sequence of moves by stating: move the tail of edge $e$ to edge $f$, then move the head of edge $e'$ to $f'$ and so forth. We then check whether these moves are *valid*, or *allowed*; that is, whether applying the steps in the definitions of the previous subsection produces a phylogenetic network. A necessary condition for a rearrangement move to be valid, is that the moving edge is 'movable', which ensures that 'detaching' the edge does not create parallel edges.

**Definition 9** *Let $(u, v)$ be an edge in a network $N$, then $(u, v)$ is* tail-movable *if $u$ is a tree node with parent $p$ and other child $c$, and there is no edge $(p, c)$ in $N$.*

This is equivalent to saying that an edge with tail $u$ is tail-movable if $u$ is a tree node and $u$ is not the side of a triangle. We give a similar definition for head moves.

**Definition 10** *Let $(u, v)$ be an edge in a network $N$, then $(u, v)$ is* head-movable *if $v$ is a reticulation node with other parent $p$ and child $c$, and there is no edge $(p, c)$ in $N$.*

When the type of move is clear from context, we will simply use the term *movable*. Using the concept of movability, we can now succinctly give sufficient conditions for a move to be valid. Besides movability, we need additional conditions to make sure that reattaching the edge does not create parallel edges, and that the resulting network has no cycles. These correspond to the second and third conditions in the following lemma.

**Lemma 1** *A tail move $(u, v)$ to $(s, t)$ is valid if all of the following hold:*

- *$(u, v)$ is tail-movable;*
- *$v \neq t$;*
- *$v$ is not above $s$.*

**Proof:** Because $(u, v)$ is tail-movable, the removal of $(u, v)$ and subsequent suppression of $u$ does not create parallel edges. Because $v \neq t$, subdividing $(s, t)$ with a node $u'$ and adding the edge $(u', v)$ does not create parallel edges either. Hence, the resulting digraph $N'$ of the tail move contains no parallel edges.

Now suppose $N'$ has a cycle. As each path that does not use $(u', v)$ corresponds to a path in $N$, the cycle must use $(u', v)$. This means that there is a path from $v$ to $u'$ in $N'$. Because $u'$ is a tree node with parent $s$, there must also be a path from $v$ to $s$ in $N'$. This implies there was also a path from $v$ to $s$ in $N$, but this contradicts the third condition: $v$ is not above $s$. We conclude that $N'$ is a DAG. As all labelled nodes remain unchanged by the tail move, $N'$ is a phylogenetic network and the tail move is valid. □

The proof of the corresponding lemma for head moves is completely analogous.

**Lemma 2** *A head move $(u, v)$ to $(s, t)$ is valid if all of the following hold:*

- *$(u, v)$ is head-movable;*
- *$u \neq s$;*
- *$t$ is not above $u$.*

We will very frequently use the following corollary of this lemma, which makes it very easy to check whether some moves are valid.

**Corollary 1** *Let $(u, v)$ be a tail-movable edge, then moving the tail of $(u, v)$ to an edge above $u$ is allowed. We also say that* moving the tail of $(u, v)$ up *is allowed. Similarly, moving the head of a head-movable edge down is allowed.*

**Lemma 3** *Let $N$ be a network and $\mathcal{T}$ its set of embedded trees, and let $N'$ with embedded trees $\mathcal{T}'$ be the result of one head move in $N$. Then there is a tree $T \in \mathcal{T}$ which is embedded in $N'$. Furthermore, for each $T' \in \mathcal{T}'$ there is a tree $T \in \mathcal{T}$ at most one tail move away from $T'$.*

**Proof:** Let $(u, v)$ be the edge that is moved in the head move from $N$ to $N'$. Then $v$ is a reticulation and it has another incoming edge $(w, v)$. There is an embedded tree $T$ of $N$ that uses this edge, and therefore does not use $(u, v)$. This means that changing the location of $(u, v)$ does not change the fact that $T$ is embedded in the network.

For the second part: first suppose the embedding of $T'$ in $N'$ does not use the new edge $(u, v')$. Then clearly $T'$ can be embedded in $N$ without the edge $(u, v)$. This means it can also be embedded in $N$.

Now suppose the embedding of the edge $(t, z)$ of $T'$ in $N'$ uses the new edge $(u, v')$. Let $P$ be the path through the embedding of $T'$ in $N'$ starting at the image of $t$, and ending at $v'$. Note that this path passes through $(u, v')$. Now consider the tree obtained by taking the embedding of $T'$, removing $P$ and adding a path of reticulation edges leading from a node $s$ in the embedding of $T'$ to $v'$ via the other incoming edge of $v'$. This tree only uses edges that are also in $N$. Hence, it is embedded in $N$ and it is at most one tail move away from $T'$: the one that moves the subtree below $v'$ to the edge of $T'$ whose embedding is a path containing $s$. □

## 2.3 Phylogenetic network spaces

Using rearrangement moves, we can not only consider phylogenetic networks as sets [e.g. 8], but also as spaces. These spaces can be defined as graphs, where each network is represented by a node, and there is an edge between two networks $N$ and $N'$ if there exists a rearrangement move changing $N$ into $N'$. Several properties of these phylogenetic network spaces have been studied.

The most basic of these properties is connectivity. The introduction of each network rearrangement move was followed by a proof of connectivity of the corresponding spaces [rSPR and rNNI moves: 9] [NNI, SPR and TBR moves: 13] [SNPR moves: 2] [tail moves: 17].

Note that spaces that take the shape of a connected graph come with a metric: the distance between two nodes. Hence, a natural follow up question is to ask about the distances between pairs of networks.

**Definition 11** *Let $N$ and $N'$ be phylogenetic networks with the same leaf set in the same tier. We denote by $d_M(N, N')$ the* distance *between phylogenetic networks $N$ and $N'$ using rearrangement moves of type $M$. That is, $d_M(N, N')$ is the minimum number of $M$-moves needed to change $N$ into $N'$.*

For phylogenetic trees, the distance between two trees is nicely characterized by a concept known as agreement forests. Recently, agreement forest analogues for networks have been introduced [21, 20], which bound distances between networks but do not give the exact distances, except in some special cases. However, not much more is known about such distances for a given pair of networks. The only other known bounds relate to the diameters: the maximal distance between any pair of networks in a phylogenetic network space.

**Definition 12** *Let $k \in \mathbb{Z}_{\geq 0}$ be the number of reticulations, $n \in \mathbb{Z}_{\geq 2}$ be the number of leaves and $M$ a type of rearrangement move. We denote with $\Delta_k^M(n)$ the* diameter *of tier-$k$ of phylogenetic network space with $n$ leaves using moves of type $M$:*

$$\Delta_k^M(n) = \max_{N,N'} d_M(N, N'),$$

*where $N, N'$ are tier-$k$ networks with $n$ leaves.*

For all previously introduced moves, some asymptotic bounds on the diameters are known ([13, 17, 2]). For SNPR moves, the diameter is unbounded as each vertical move can only change the reticulation number by at most one. For all moves, the diameter of each *tier* of phylogenetic networks is finite.

The last property we discuss is the neighbourhood size of a phylogenetic network: the number of networks that can be reached using one rearrangement move. The size of the neighbourhood is important for local search heuristics, as it gives the number of networks that need to be considered at each step. For networks, the only rearrangement move neighbourhood that has been studied is that of the SNPR move [19].

# 3 Connectivity

In this section, we consider the connectivity of tiers of network space under local head moves. One might hope that distance-1 head moves are enough to reach any network from an arbitrary other network in the same tier. For tail moves, such a result has been proved [17], so it seems reasonable to expect a similar result for head moves. We prove that this is, unfortunately, not the case. However, we will show that distance-2 head moves do suffice.

## 3.1 Distance-1 is not enough

We show by example that distance-1 head moves are not enough to connect the tiers of phylogenetic network space (Figure 6). For tier-1 networks, this example can easily be checked, as there are no distance-1 head moves in the left network that result in a different network. For higher tiers, however, there remain many valid distance-1 head moves. Using the following lemma, we will show that the reticulations remain roughly at the same place in all the resulting networks.

**Lemma 4** *Let $N$ be a network, and $N'$ the result of a distance-1 head move in $N$. If, in $N$, all reticulations and their parents are below some tree node $s$, then the same holds for $N'$.*

**Proof:** Suppose the head move between $N$ and $N'$ moves $(u, v)$ from $(x, y)$ to $(z, w)$. Let $q$ be a reticulation or a parent of a reticulation in $N$, with $q$ not equal to $v$, then there is a path from $s$ to $r$ in $N$. Furthermore, we may assume that this path does not pass through $(u, v)$, as it could alternatively use a path using the other in-edge of $v$. Hence, after the head move, $q$ is still below $s$. Now note that the parent of $v$ in $N'$ is below $s$. If $v$ itself were above $s$ in $N'$, there would be a cycle in $N'$. Hence, $v$ is below $s$ in $N'$ as well. We conclude that all reticulations and their parents are below $s$ in $N'$. □

**Proposition 1** *In all tiers of phylogenetic space with $n \geq 3$ leaves, there exist two networks not connected by a sequence of distance-1 head moves.*

**Proof:** For tier-0, no head moves are possible, but there are non-isomorphic networks (trees) with no reticulations and at least 4 leaves. This proves the proposition for tier-$k$ where $k = 0$. We now prove the proposition for tier-$k$ for an arbitrary $k > 0$.

Let $T$ be a caterpillar on $n \geq 3$ leaves, and let $s$ be the common parent of two of the leaves, and let $t$ be the highest tree node in $T$. Now construct the network $N$ by adding $k$ reticulation

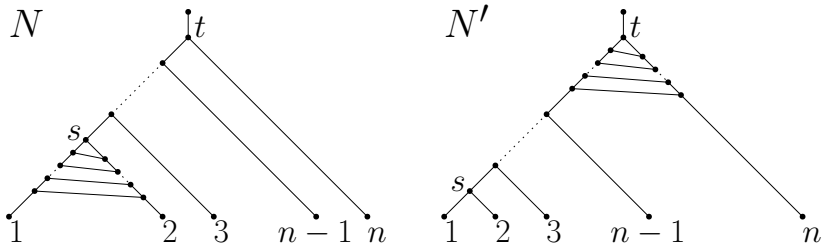



Figure 6: There is no sequence of distance-1 head moves between $N$ and $N'$ (Proposition 1).

edges between the outgoing edges of $s$, and the network $N'$ by adding $k$ reticulation edges between the outgoing edges of $t$ (Figure 6). In $N$, all of the reticulations and their parents are below $s$. Lemma 4 implies that, using distance-1 head moves, only networks with all reticulations below $s$ can be reached. Furthermore, because no part above $s$ can ever be involved, the caterpillar structure above $s$ will remain intact. Hence, any network reachable from $N$ using distance-1 head moves consists of a chain of pendant leaves followed by the node $s$, which must still be above all reticulations and parents of reticulations. Now note that $N'$ is not such a network. We conclude that there is no distance-1 head move sequence between $N$ and $N'$. □

**Proposition 2** *All tiers of phylogenetic network space with one leaf are connected by distance-1 head moves.*

**Proof:** This follows from the fact that all tiers of phylogenetic network space with one leaf are connected by distance-1 tail moves [17]. Indeed, if one reverses the direction of all edges, a network with one leaf becomes another network with one leaf, and each distance-1 tail move in the reversed network is a distance-1 head move in the original network. □

This shows one should not use only distance-1 head moves in local-search heuristics. However, if one wants to use them as part of a search strategy, it would still be interesting to know how disconnected spaces of distance-1 head moves are. For now, we will leave this question open, and pragmatically turn our attention to distance-2 head moves.

## 3.2 Distance-2 suffices

To prove the connectivity of tiers of network space using distance-2 head moves, we present a procedure to generate a sequence between any two networks in the same tier. This sequence first turns both networks into networks that look like a tree, with all reticulations collected at the top. Next, the tree structure of these networks is adjusted, by simulating rSPR moves on the trees using distance-2 head moves.

### 3.2.1 Collecting the reticulations at the top

In this subsection, we show how all reticulations can be collected at the top of the network using distance-2 head moves. This will be achieved by creating triangles, and moving these through the network. First, we define what it means for all reticulations to be at the top.

**Definition 13** *A network has $k$ reticulations at the top if it has the following structure:*

*1) the node $c$: the child of the root;*

*2) nodes $a_i$ and $b_i$ and an edge $(a_i, b_i)$ for each $i \in \{1, \ldots, k\}$;*

*3) the edges $(c, a_1)$ and $(c, b_1)$;*

*4) for each $i \in \{1, \ldots, k-1\}$ there are edges $(a_i, a_{i+1})$ and $(b_i, b_{i+1})$ or edges $(a_i, b_{i+1})$ and $(b_i, a_{i+1})$.*

*We say the there are $k$ reticulations neatly at the top if they are all directed to the same edge, i.e. we replace point 4) with*

*4') for each $i \in \{1, \ldots, k-1\}$ there are edges $(a_i, a_{i+1})$ and $(b_i, b_{i+1})$.*

*Examples are shown in Figure 7.*

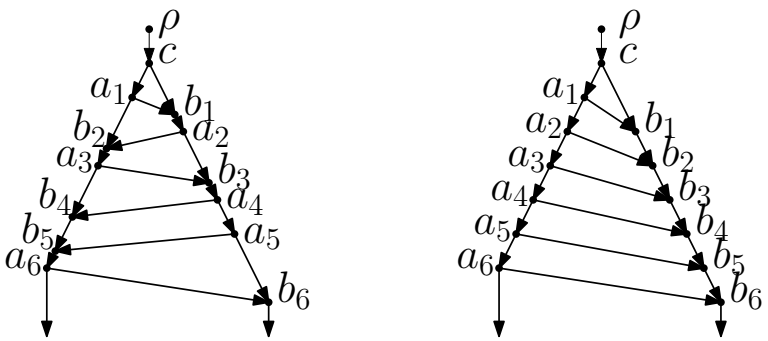



Figure 7: Two networks with 6 reticulations at the top. In the right network, the reticulations are neatly at the top.

The following lemma ensures that the top reticulations can be directed neatly using local head moves. The moves are similar to the one used to change the direction of a triangle (cf. Observation 1).

**Definition 14** *Let $N$ be a network with $k$ reticulations at the top.* Changing the direction *of an edge $(a_i, b_i)$ (as in Definition 13) consists of changing $N$ into a network $N'$ that is isomorphic to $N$ when $(a_i, b_i)$ is replaced by $(b_i, a_i)$. Note that labels $a_j$ and $b_j$ do not coincide between $N$ and $N'$. Changing the direction of a set of such edges at the same time is defined analogously.*

**Lemma 5** *Let $N$ be a network with $k$ reticulations at the top. Then the reticulations can be redirected so that they are neatly on top (directed to either edge) with at most $k$ distance-1 head moves. The network below $a_k$ and $b_k$ (notation as in Definition 13) is not altered in this process.*

**Proof:** We redirect the top reticulations starting with the lowest one. The move $(u_{i-1}, b_i)$ to $(a_i, v_{i+1})$ with $u_{i-1}$ the parent of $b_i$ that is not $a_i$ and $v_{i+1}$ the child of $a_i$ that is not $b_i$ (Figure 8) changes the direction of the chosen edge $(a_i, b_i)$ and all the reticulation edges $(a_j, b_j)$ above; it leaves all other edges fixed as they were. □

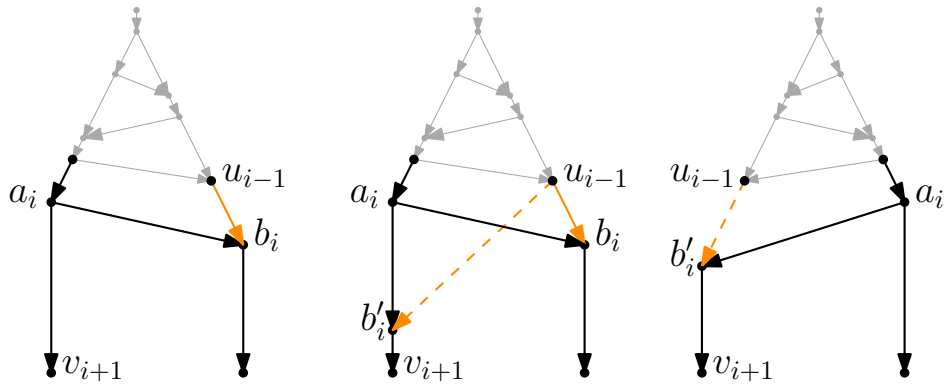



Figure 8: The move used in Lemma 5 to redirect the $i$ highest reticulations at the top: head move $(u_{i-1}, b_i)$ to edge $(a_i, v_{i+1})$. This move changes the direction of all reticulations at the top that are higher than the moved edge. The part of the network below $a_i$ and $b_i$ does not change.

**Definition 15** *Let $N$ be a network with $k$ reticulations at the top (notation as in Definition 13) and a tree node $x$ directly below $a_k$.* Moving a triangle from the top *consists of creating a triangle at $x$ by a head move of $(a_k, b_k)$ to one of the outgoing edges $(x, c(x))$ of $x$.* Moving a triangle to the top *is the reverse of this operation.*

**Lemma 6** *Triangles can be moved along between tree nodes, and to/from the top using distance-2 head moves.*

**Proof:** Suppose the network has a triangle consisting of the edges $(u, v)$, $(u, w)$, and $(v, w)$ with $u$ the child of a tree node $s$. Let the other children of $s$, $v$, and $w$ be $a$, $b$, and $c$ respectively. To move the triangle up to $s$, we use the following sequence of distance-2 head moves. Move $(v, w)$ to $(s, a)$, move $(s, w')$ to $(u, v)$, move $(u, w'')$ to $(v, b)$, and move $(v, w''')$ to $(s, u)$ (Figure 9). None of the intermediate networks in the sequence contain a directed cycle or parallel edges, unless $a = b$. However, in that case, the move $(u, w)$ to $(s, a)$ is a distance-2 head move that moves the triangle up.

Now suppose the network has $k$ reticulations at the top, and there is a triangle $(u, v, w)$ below $s = a_k$. To move the triangle to the top, first move the triangle up using the previous sequence of moves; then reverse the direction of the triangle using the distance-1 head move $(s, w'''')$ to $(v, b_k)$, resulting in the triangle $(s, v, w'''')$; and, lastly, move $(v, w'''')$ to the outgoing edge of $b_k$. □

If the restriction to distance-2 moves is relaxed, the triangle can also be moved using one distance-3 head move: $(u, w)$ to $(s, a)$.

**Lemma 7** *Let $N$ be a network and $v$ a highest reticulation below the top reticulations. Suppose $(u, v)$ to $(x, y)$ is a valid head move resulting in a network $N'$. Then there is a sequence of distance-2 head moves from $N$ to $N'$.*

**Proof:** Pick an up-down path from $v$ to $(x, y)$ not via $(u, v)$. Note that if there is a part of this path above $u$, it is also above $v$ and therefore only contains tree nodes. Sequentially move the head of $(u, v)$ to the pendant branches of this path as in Figure 10. It is clear this works except at the

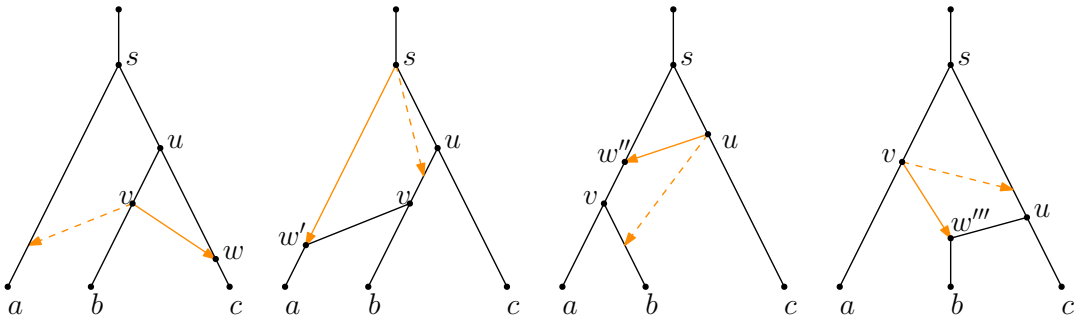



Figure 9: Sequence of moves used to move triangles with distance-2 head moves (Lemma 6).

point where $u$ is on the up-down tree path (the obvious move is a distance-3 move), and at the top.

Note that at the top, we need to move the head to the lowest reticulation edge at the top. This is of course only possible if this reticulation edge is directed away from $u$. If it is not, we redirect it using one distance-1 head move (Lemma 5), and redirect it back after we move the moving head down to the other branch of the up-down tree path.

If $u$ is on the up-down path, we use Lemma 6 to pass this point: Let $c(u)$ be the other child of $u$ (not $v$) and $p(u)$ the parent node of $u$; moving the head from a child edge of $c(u)$ to the other child edge $(p(u), w)$ of $p(u)$ is equivalent to moving the triangle at $c(u)$ to a triangle at $p(u)$.

We have to be careful, because if the child of $u$ is not a tree node, this sequence of moves does not work. However, if $c(u)$ is a reticulation node, there exists a different up-down path from $v$ to $(x, y)$ not through $u$: such a path may use the other incoming edge of $c(u)$.

At all other parts of the up-down path, the head may be simply moved to the edges on the path. Using these steps, we can move the head of $(u, v)$ to $(x, y)$ with only distance-2 head moves. □

Using these lemmas, it is easy to prove we can use distance-2 head moves to move reticulations to the top.

**Lemma 8** *Let $N$ be a tier-$k$ network, then there is a sequence of distance-2 head moves turning $N$ into a network with all reticulations at the top.*

**Proof:** Note that the network induces a partial order on the reticulation nodes. Suppose $N$ has $l < k$ reticulations at the top. Let $r$ be a highest reticulation node that is not yet at the top. One of the two corresponding reticulation edges is head-movable. Let this be the edge $(s, r)$.

If $s$ is a child of $a_l$ or $b_l$ (as in Definition 13; i.e., $s$ is directly below the top reticulations), then one head move suffices to get this reticulation to the top. By Lemma 7, this move can be replaced by a sequence of distance-2 head moves. Otherwise, there is at least one node between $s$ and the top, let $t$ be the lowest such node, that means that $t$ is the parent of $s$. Because $r$ is a highest reticulation that is not at the top, $t$ is a tree node and there are edges $(t, s)$ and $(t, q)$. Moving the head of $(s, r)$ to $(t, q)$ is a valid head move that creates a triangle. By Lemma 7, this head move can be replaced by a sequence of distance-2 head moves.

Now we move this triangle to the top using distance-2 head moves as in Lemma 6. This increases the number of reticulations at the top by one. □

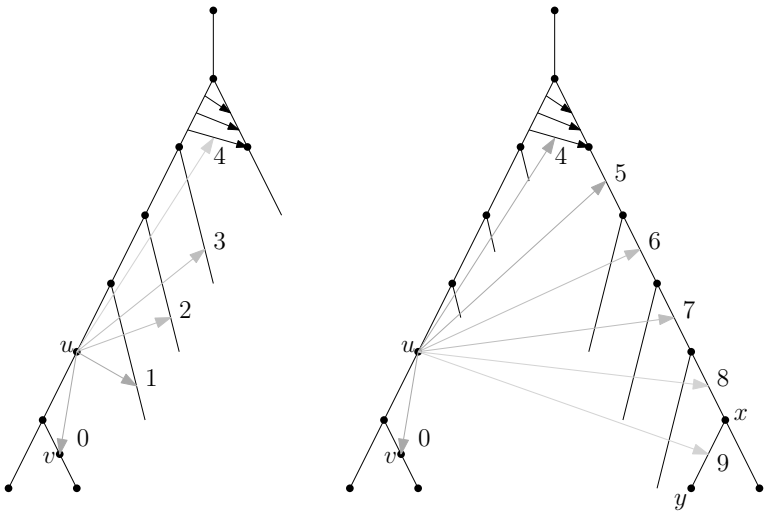



Figure 10: An example of a sequence as used in Lemma 7. Note that on the side of the tree containing the tail of the moving edge, we use the side branches to avoid cycles. The numbers represent the order of the distance-2 head moves. Note that the move from position 0 to position 1 is not a distance-2 head move, in this case we use the sequence of moves described in Lemma 6. Also note that position 4 is only allowed when the lowest reticulation at the top is directed away from the tail of the moving edge.

### 3.2.2 Changing the tree

Networks with all reticulations at the top have exactly one embedded tree. As such networks are essentially determined by this tree, we need to change this embedded tree. To achieve this, we use the lowest reticulation edge $(a_k, b_k)$ to create a triangle that can move around the lower part of the network. Using the reticulation in this triangle, we simulate rSPR moves on the embedded tree.

**Lemma 9** *Let $N$ and $N'$ be tier-$k$ networks on the same leaf set with $k-1$ reticulations at the top and the $k$-th reticulation at the bottom of a triangle. Suppose $N$ and $N'$ have the same embedded trees, then there exists a sequence of distance-2 head moves from $N$ to $N'$.*

**Proof:** Note that the network consists of $k-1$ reticulations at the top, and two pendant subtrees—isomorphic to the two pendant subtrees below the highest tree node of the embedded tree—one of which contains a triangle. The triangle can be moved through one of these subtrees using Lemma 6. To move the triangle anywhere, we need to be able to move it between the pendant subtrees as well. This can be done by moving the triangle to the top, and then moving it down on the other side after redirecting all the top reticulations, using Lemma 6. None of these triangle moves change the embedded tree: each of the intermediate networks has exactly one embedded tree, and doing a head move keeps at least one embedded tree. Hence, moving the triangle to the right place and then redirecting the triangle and the top reticulations as needed gives a sequence from $N$ to $N'$. □

**Lemma 10** *Let $N$ and $N'$ be tier-$k$ networks ($k > 0$) on the same leaf set with all reticulations neatly at the top. Then there exists a sequence of distance-2 head moves turning $N$ into $N'$.*

**Proof:** Note that $N$ and $N'$ both have exactly one embedded tree, $T$ and $T'$ respectively, and we aim to change this embedded tree. It suffices to prove this for any $T'$ that is one rSPR move away from $T$, because the space of phylogenetic trees with the same leaf set is connected by rSPR moves. Hence, let $(u, v)$ to $(x, y)$ be the rSPR move that transforms $T$ into $T'$.

First suppose the rSPR move does not involve the root edge of the embedded trees. We can move triangles anywhere below the $k-1$ reticulations at the top by Lemma 9. Hence, there is a sequence of head moves transforming $N$ into a network $M$ with the following properties: the tree $T$ can be embedded in $M$; $M$ has a reticulation edge $(a, b)$ where $a$ lies on the image of $(x, y)$ in $M$, and the head $b$ lies on the image of the other outgoing edge $(x, z)$ of $x$ if $x$ is not the root and on the image of one of the child edges $(y, z')$ of $y$ otherwise.

This creates a situation where there are edges $(x, a)$, $(a, b)$, $(p, b)$ $(a, y)$ and $(b, \zeta)$ with $p = x$ and $\zeta = z$ or $p = y$ and $\zeta = z'$. The case $p = x$ is depicted in Figure 11.

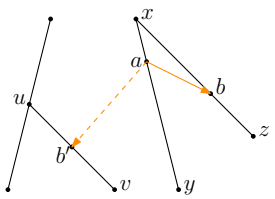



Figure 11: The head move used to simulate the rSPR move $(u, v)$ to $(x, y)$ on the embedded tree $T$ to $T'$. The triangle is already in the position described in the proof of Lemma 10. The starting network consists of the solid arrows, by doing the head move $(a, b)$ to $(u, v)$ (orange) we get the network consisting of the solid black and the dashed orange edges. The embedded tree using the new reticulation edge corresponds to $T'$.

Now do a head move of $(a, b)$ to the image of the $T$-edge $(u, v)$, this is allowed because any reticulation edge in a tier-1 network is movable; $b$ is not equal to the image of $u$ as $b$ is a reticulation node and the image of $u$ a tree node; and the image of $v$ is not above $a$, as otherwise the tail move $(u, v)$ to $(x, y)$ could not be valid. Let $N'$ be the resulting network, and note that the embedded tree using the new reticulation edge is $T'$.

Next, suppose the rSPR move does involve the root edge of the embedded tree. Let $(u, v)$ to $(\rho, c)$ be such an rSPR move to the root edge of the tree, and let $x$ and $y$ be the nodes directly below the top: $x$ on the side of the reticulations $b_i$, and $y$ on the side of the tree nodes $a_i$. Do the rSPR move of $(u, v)$ to $(a_k, x)$, that is, to an edge directly below the top. This produces the network with $k-1$ reticulations at the top, and (in Newick notation) embedded tree $((T \downarrow x, T \downarrow v), T \downarrow y)$, where $T \downarrow z$ denotes the part of tree $T$ below $z$. Then do the rSPR move $(u', x)$ to $(b_k, y)$, the other side of the top, producing a network with $k-1$ reticulations at the top and embedded tree $((T \downarrow x, T \downarrow y), T \downarrow v)$.

This creates the desired network with $v$ below one side of the top, and $x$ and $y$ on the other side. Both these rSPR moves are performed as in the previous case, which did not involve the root edge. After the rSPR moves, we move the triangle back to the top without changing the embedded tree, and redirect the top reticulations as needed to produce $N'$. □

**Theorem 1** *Tier-k of phylogenetic networks is connected by distance-2 head moves, for all $k > 0$.*

**Proof:** Let $N$ and $N'$ be two arbitrary networks in the same tier with the same leaf set. Use Lemma 8 and Lemma 5 to change $N$ and $N'$ into networks $N_n$ and $N'_n$ with all reticulations neatly

at the top using only distance-2 head moves. Now, Lemma 10 tells us that there is a sequence of distance-2 head moves from $N_n$ to $N'_n$. Hence, tier-$k$ of phylogenetic network space is connected by distance-2 head moves. □

**Corollary 2** *Tier-k of phylogenetic networks is connected by head moves, for all $k > 0$.*

# 4 Relation to tail moves

In this section, we show that each tail move can be replaced by a sequence of at most 15 head moves, and each head move can be replaced by a sequence of at most 16 tail moves.

## 4.1 Tail move replaced by head moves

Here, we show how to replace a tail move by a sequence of head moves (Theorem 2). The proof works by case distinction, where the main cases represent different types of tail moves. The first two lemmas prove that we can replace certain types of distance-1 tail moves: in Lemma 11, we replace a distance-1 tail move between the two outgoing arcs of a tree node, and in Lemma 12, we replace a distance-1 tail move between the two incoming arcs of a reticulation. Then, we turn to the remaining cases, where the tail move $(u, v)$ from $(x_L, a_L)$ to $(x_R, a_R)$ is such that $a_L \neq a_R$ and $a_L$ is not above $a_R$ (Lemma 15), and $a_L \neq a_R$ and $a_L$ is not above $a_R$ (Lemma 16). This case is split up into two lemmas, depending on where the head-movable arcs are located in the network in relation to $a_R$ (Lemmas 13 and 14).

In this section, unless stated otherwise, each move is a head move and movable means head-movable.

**Lemma 11** *Let $(u, v)$ from $(x, a_L)$ to $(x, a_R)$ be a valid tail move in a tier $k > 0$ network $N$ resulting in a network $N'$. Then there exists a sequence of head moves from $N$ to $N'$ of length at most 6.*

**Proof:** To prove this, we have to find a reticulation somewhere in the network that we can use, as the described part of the network might not contain any reticulations.

Note that there exists a head-movable reticulation edge $(t, r)$ in $N$ with $t$ not below both $a_L$ and $a_R$: Find a highest reticulation node below $a_L$ and $a_R$; if it exists, one edge is movable, this edge cannot be below both; if there is no such reticulation, then there is a reticulation $r$ that is not below both $a_L$ and $a_R$ and so the same holds for its movable edge.

First assume we find a head-movable edge $(t, r)$ with $t = x$. Note that $r$ cannot be the same node as $u$, as $u$ is a tree node and $r$ is a reticulation. This means that $r = a_R$, and $(x, a_R)$ is movable. Move $(x, a_R)$ to $(u, a_L)$, which is allowed because $x \neq u$, $a_L$ not above $x$ and $(t, r) = (x, a_R)$ is movable. Now moving $(u, r')$ to $(s, z)$, the edge created by suppressing $a_R$ after the previous move, we get network $N'$.

Now assume we find a head-movable edge $(t, r)$ with $x \neq t$. Suppose w.l.o.g. $(t, r)$ is not below $a_L$, then we can use the following sequence of 4 moves except in the cases we mention in bold below the steps. For this lemma, we call this sequence the 'normal' sequence (Figure 12). The validity of each move is checked using Lemma 2.

- Move $(t, r)$ to $(u, a_L)$, keeping $(x, a_R)$ except if $\boldsymbol{x = t}$. This can be done if $(t, r)$ is movable, which it is by choice of $(t, r)$; $t \neq u$, but we note that $\boldsymbol{t = u}$ may occur; and, $a_L$ is not above $t$, which is true by choice of $(t, r)$.

- Move $(u, r')$ to $(x, a_R)$, creating edge $(t, a_L)$ as $(x, a_R) \neq (t, r')$ and $(x, a_R) \neq (r', a_L)$. For this move, note that $(u, r')$ is movable, except when $\boldsymbol{(t, a_L) \in N}$; $u \neq x$ as these nodes are distinct in the original network; and $a_R$ is not above $u$, as it is not above $x$.
- Move $(x, r'')$ to $(t, a_L)$. The edge $(x, r'')$ is movable because $v \neq a_R$ (otherwise the tail move would be not valid); $t \neq x$, as we have assumed so for the first move; and $a_L$ is not above $x$, as it is above $x$.
- Move $(t, r''')$ back to its original position. This move is allowed because this produces the network $N'$.

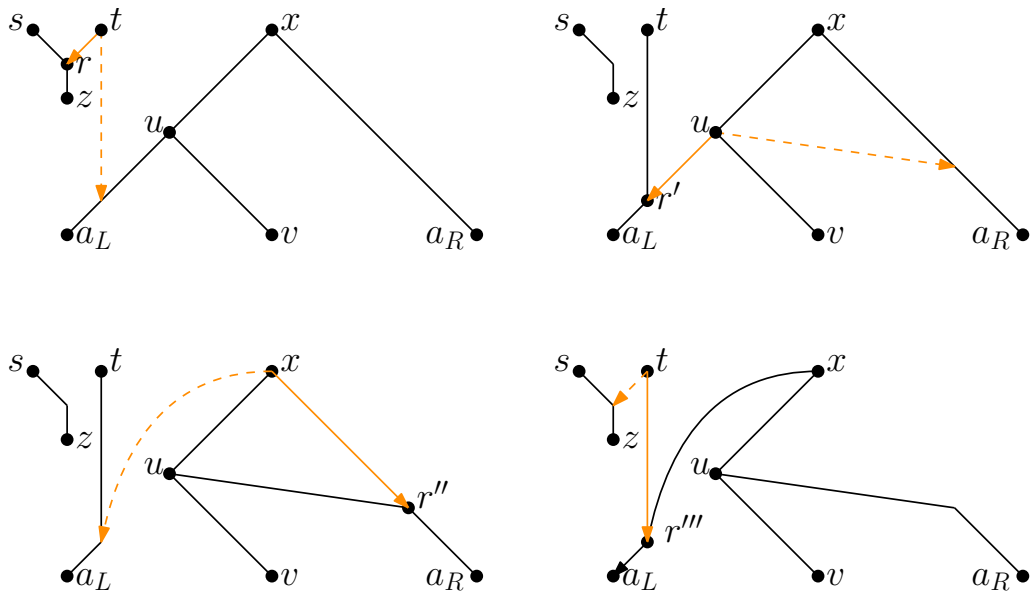



Figure 12: The 'normal' sequence of head moves simulating a tail move in Lemma 11.

We now look at the situations where $t = x$, $t = u$, or $(t, a_L) \in N$ separately. We will split up in cases to keep the proof clear. Recall that $(t, r)$ is a movable edge in $N$.

1. $\boldsymbol{t = u}$.
   (a) $\boldsymbol{r = a_L}$. Move $(t, r)$ to $(x, a_R)$, then move $(x, r')$ back to the original position of $r$, creating $N'$. This is a sequence of 2 head moves.
   (b) $\boldsymbol{r = v}$. Move $(t, r)$ to $(x, a_R)$, creating a triangle at $x$. Now reverse the triangle by moving $(x, r')$ to $(t, a_L)$. Now create $N'$ by moving $(t, r'')$ back to the original position of $r$. This is a sequence of 3 head moves.
2. $\boldsymbol{t \neq u}$. Note that we can assume that $\boldsymbol{(u, a_L)}$ **is not movable**, as otherwise we are in the previous case. Because $t \neq u$, we may also assume $\boldsymbol{(t, a_L) \in N}$, otherwise this is no special case and we can use the sequence of moves from the start of this proof. Hence, $a_L$ is a reticulation node on the side of a triangle formed by $t$, $a_L$, and the child $c(a_L)$ of $a_L$.

(a) **$t$ is below $a_R$.**

i. **$t$ is below $v$.** Since $t$ is below both $a_R$ and $v$, there is a highest reticulation $s$ strictly above $t$ and below both $a_R$ and $v$. Since $s$ is strictly above $t$, it is strictly above $a_L$. Therefore we are either in the 'normal' case, or in Case 1b of this analysis with movable edge $(p(s), s)$. This means this situation can be solved using at most 4 head moves.

ii. **$t$ is not below $v$.** As $(t, a_L)$ is a reticulation edge in the triangle, it is movable, and because $t$ is not below $v$, the head move $(t, a_L = r)$ to $(u, v)$ is allowed. Now the *tail* move $(u, r')$ to $(x, a_R)$ is still allowed, because $v$ is not above $x$. As $(u, c(a_L))$ is movable in this new network, we can simulate this tail move like in Case 1a. Afterwards, we can put the triangle back in its place with one head move, which is allowed because it produces $N'$. All this takes 6 head moves.

(b) **$t$ is not below $a_R$.** Because $t \neq x$, we know that $(t, a_R) \notin N$. Therefore we can do the 'normal' sequence of moves from the start of this proof in reverse order, effectively switching the roles of $a_L$ and $a_R$. Because we use the 'normal' sequence of moves, this case takes at most 4 head moves.

□

To prove the case of a more general tail move, we need to treat another simple case first.

**Lemma 12** *Let $(u, v)$ from $(x_L, r)$ to $(x_R, r)$ be a valid tail move in a network $N$ turning it into $N'$, then there is a sequence of head moves from $N$ to $N'$ of length at most 4.*

**Proof:** Let $z$ be the child of $r$, and note that not all nodes described must necessarily be unique. All possible identifications are $x_L = x_R$ and $v = z$, other identifications create cycles. First note that in the situation $x_L = x_R$, the networks $N$ and $N'$ before and after the tail move are isomorphic. Hence we can restrict our attention to the case that $x_L \neq x_R$. To prove the result, we distinguish two cases.

1. **$z \neq v$.** This case can be solved with two head moves: $(x_R, r)$ to $(x_L, u)$ creating new reticulation node $r'$ above $u$ followed by $(x_L, u)$ to $(u, z)$. The first head move is allowed because $v \neq z$, so $(x_R, r)$ is head-movable; $x_R \neq x_L$; and $u$ is not above $x_R$ because both its children aren't: $z$ is below $a_R$, and if $v$ is above $x_R$, the tail move $N \to N'$ is not allowed. The second head move is allowed because it produces the valid network $N'$. Hence the tail move can be simulated by at most 2 head moves (Figure 13).

2. **$z = v$.** The proposed moves of the previous case are not valid here, because they lead to parallel edges in the intermediate network. To prevent these, we reduce to the previous case by moving $(u, z)$ to any edge $e$ not above $z$ and $e \neq (z, c(z))$ (hence neither above $x_L$ nor above $x_R$) and moving it back afterwards. Note that if there is such an edge $e$, then the head move $(u, z)$ to $e$ is allowed. Let the new node subdividing $e$ be $v'$, then the tail move $(u, v')$ to $(x_R, r)$ is 'still' allowed and can therefore be simulated by 2 head moves as in the previous case, the last head move, moving $(u', v')$ 'back' is allowed because it creates the DAG $N'$ which is a network. Such a sequence of moves uses 4 head moves (Figure 14).

   It remains to prove that there is such a location (not above $z$ and excluding $(z, c(z))$) to move $(u, z)$ to. Recall that we assume any network has at least two leaves. Let $l$ be a leaf not equal to $c(z)$, then its incoming edge $(p(l), l)$ is not above $c(z)$ and not equal to $(z, c(z))$. Hence this edge $e = (p(l), l)$ suffices as a location for the first head move.

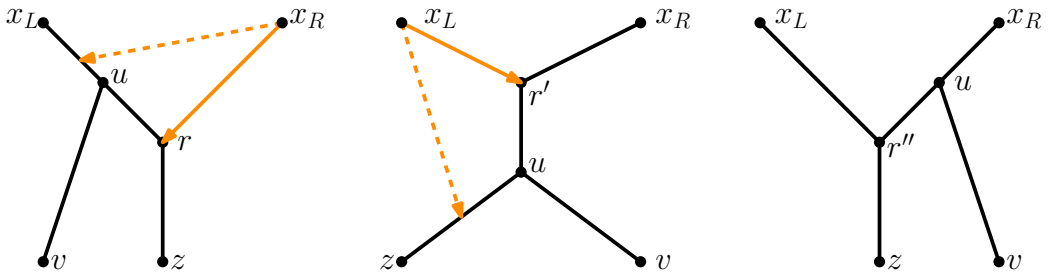



Figure 13: The two moves used to simulate a tail move in Case 1 of Lemma 12.

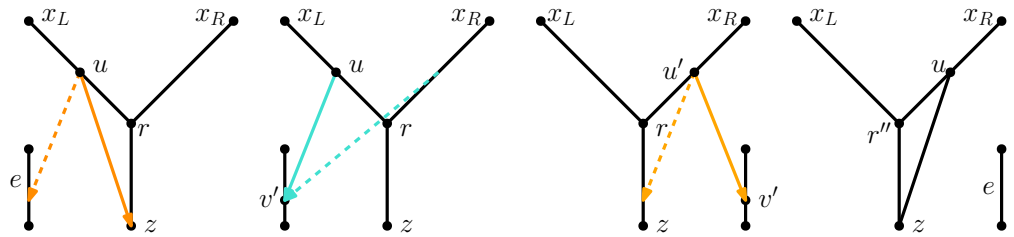



Figure 14: The four moves used in Case 2 of Lemma 12. The middle depicted move is the tail move of Case 1, which can be replaced by two head moves.

We conclude that any tail move of the form $(u, v)$ from $(x_L, r)$ to $(x_R, r)$ can be simulated by 4 head moves. □

**Lemma 13** *Let $(u, v)$ from $(x_L, a_L)$ to $(x_R, a_R)$ be a valid tail move in a network $N$ resulting in a network $N'$. Suppose $a_L \neq a_R$, $a_L$ is not above $a_R$, and there exists a movable reticulation edge $(t, r)$ not below $a_R$. Then there exists a sequence of head moves from $N$ to $N'$ of length at most 7.*

**Proof:** Note that $v$ cannot be above either of $x_L$ and $x_R$. The only possible identifications within the nodes $a_L, a_R, x_L, x_R, u, v$ are $a_L = a_R$, $x_L = x_R$ and $a_R = x_L$ (but not simultaneously), all other identifications lead to parallel edges, cycles in either $N$ or $N'$, a contradiction with the condition "$a_L$ is not above $a_R$", or a trivial situation where the tail move leads to an isomorphic network. The first of these two identifications have been treated in the previous two lemmas, so we may assume $\boldsymbol{a_L \neq a_R}$ and $\boldsymbol{x_L \neq x_R}$. We now distinguish several cases to prove the tail move can be simulated by a constant number of head moves in all cases.

1. $\boldsymbol{(t, a_R) \notin N}$.

   (a) $\boldsymbol{r = x_R}$. As $(t, r)$ is movable and not below $a_L$ or $v$, we can move the head of this edge to $(x_L, u)$. The head move $(x_L, r')$ down to $(u, a_L)$ is then allowed. Let $s$ be the parent of $r$ in $N$ that is not $t$. Since $u \neq s$ (otherwise the original tail move was not allowed), the head move $(u, r'')$ to $(s, a_R)$ is allowed, where $s$ is the other parent of $r$ in $N$ (i.e., not $t$). Lastly $(s, r''')$ to $(t, u)$ gives the desired network $N'$.

   (b) $\boldsymbol{r \neq x_R}$. In this case, we can move $(t, r)$ to $(x_R, a_R)$ in $N$ (if $t = x_R$ then $(t, a_R) \in N$, contradicting the assumptions of this case). Because neither $a_L$ nor $v$ can be above

$x_R$ and $x_L \neq x_R$, we can now move $(x_R, r')$ to $(x_L, u)$. Then we move down the head $(x_L, r'')$ to $(u, a_L)$, followed by $(u, r''')$ to $(t, a_R)$. If $u = t$ and $r = v$, the last move is not allowed, and if $u = t$ and $r = a_L$ these last two moves are not allowed. In these cases, we simply skip these move. Lastly, we move $(t, r'''')$ to $(s, z)$ to arrive at $N'$, where $s$ and $z$ are the other parent and the child of $r$ in $N$. Hence the tail move of this situation can be simulated by 5 head moves (Figure 15).

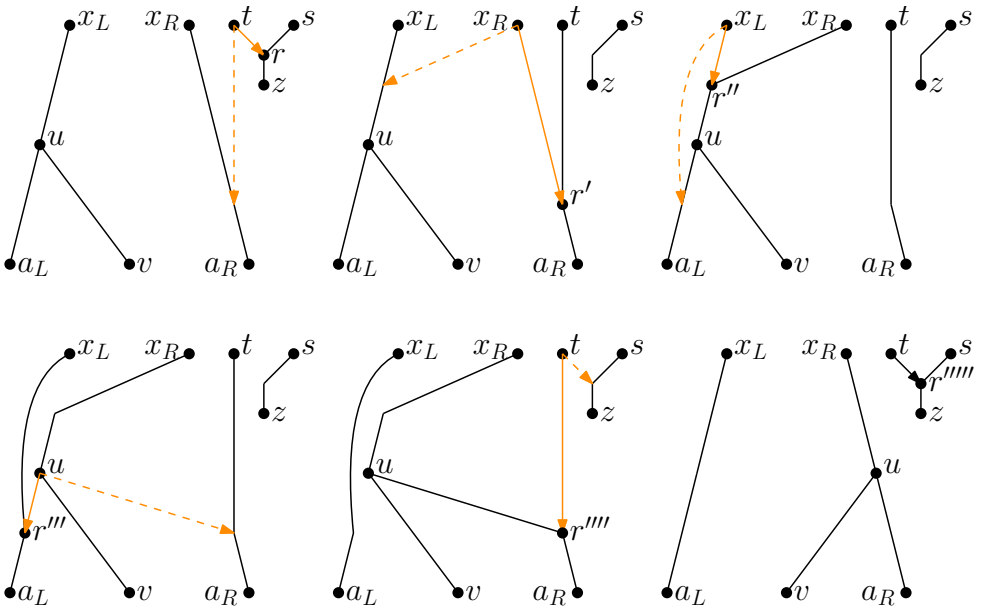



Figure 15: The five moves used to simulate a tail move in Case 1b of Lemma 13.

2. $\boldsymbol{(t, a_R) \in N}$. Again $(t, r)$ is the head-movable edge. Let $z$ be the child of $r$ and $s$ the other parent of $r$.

   (a) $\boldsymbol{z = a_R}$. Note first that in this case, we must have either $x_R = t$ or $x_R = r$, otherwise one of the edges $(t, a_R)$ and $(r, z)$ is not in $N$.

      i. $\boldsymbol{x_R = t}$ This case is quite easy, and can be solved with 3 head moves. Because $r$ and $t = x_R$ are distinct, $a_R = z$ is a reticulation node with movable edge $(t = x_R, z = a_R)$. The sequence of moves is: $(t, z)$ to $(x_L, u)$, then $(x_L, z')$ to $(u, a_L)$, then $(u, z'')$ to $(r, c(z))$.

      ii. $\boldsymbol{x_R = r}$ Note that the *tail* move $(u, v)$ to $(t, a_R)$ is also allowed in this case because $(u, v)$ is tail-movable, $v \neq a_R$ and $v$ not above $t$ (otherwise the tail move to $(r, z)$ is not allowed either). This tail move is of the type of the previous case, and takes at most 3 head moves. Now the move $(u', v)$ to $(r, z)$ is of the type of Lemma 12, which takes at most 4 head moves to simulate. We conclude any tail move of this case can be simulated with 7 head moves.

   (b) $\boldsymbol{z \neq a_R}$.

      i. $\boldsymbol{a_R \neq r}$.

A. $\boldsymbol{x_R = t}$ and $\boldsymbol{v \neq r}$. We can move the *tail* of $(u, v)$ to $(x_R, r)$ with a sequence of four head moves like the sequence in Case 2(a)i. The resulting DAG is a network because $v$ is not above $x_R$ and $v \neq r$. Call the resulting new location of the tail $u'$. We can get to $N'$ with two head moves (Lemma 11 Case 1a): $(u', r)$ to $(t, a_R)$, then $(t, r')$ to $(s, z)$. This case therefore takes at most 6 head moves.

B. $\boldsymbol{x_R = t}$ and $\boldsymbol{v = r}$. The following sequence of four head moves suffices: $(t = x_R, r = v)$ to $(x_L, u)$, then $(x_L, r')$ to $(u, a_L)$, then $(u, r'')$ to $(t, a_R)$ and finally $(t, r''')$ to $(u, z)$. Hence this case takes at most 4 head moves.

C. $\boldsymbol{x_R \neq t}$ and $\boldsymbol{a_R \neq s}$. Because $a_R \neq s$, the edge $(x_R, a_R)$ is movable. Also, because $x_R \neq x_L$ and $u$ not above $x_R$, $(x_R, a_R)$ can be moved to $(x_L, u)$. Now $(x_L, a'_R)$ is movable, and it can be moved down to $(u, a_L)$. Finally, the head move $(u, a''_R)$ to $(t, c(a_R))$ results in $N'$. Hence in this case we need at most 3 head moves.

D. $\boldsymbol{x_R \neq t}$ and $\boldsymbol{a_R = s}$. The following sequence of five head moves suffices: $(t, r)$ to $(u, a_L)$, then $(x_R, s)$ to $(x_L, u)$, then $(x_L, s')$ to $(u, r')$, then $(u, s'')$ to $(t, c(r))$, and finally $(t, r')$ to $(s''', c(r))$. Hence this case takes at most 5 head moves.

ii. $\boldsymbol{a_R = r}$. In this case either $x_R = t$ or $x_R = s$.

A. $\boldsymbol{x_R = t}$. This case is easily solved with 3 head moves: $(x_R, a_R)$ to $(x_L, u)$, then $(x_L, a'_R)$ to $(u, a_L)$, then $(u, a''_R)$ to $(s, z)$.

B. $\boldsymbol{x_R = s}$. If $(s, r)$ is movable (i.e. there is no edge $t, z$), then we can relabel $t \leftrightarrow s$ and treat like the previous case. Otherwise, there is an edge $(t, z)$ and we use the following sequence of moves: $(t, a_R)$ to $(u, a_L)$, then $(x_R = s, z)$ to $(x_L, u)$, then $(x_L, z')$ to $(u, a'_R)$, then $(u, z'')$ to $(t, c(z))$, then $(t, a'_R)$ to $(z''', c(z))$. The tail move of this situation can therefore be replaced by 5 head moves.

□

**Lemma 14** *Let $(u, v)$ from $(x_L, a_L)$ to $(x_R, a_R)$ be a valid tail move in a network $N$ resulting in a network $N'$. Suppose $a_L \neq a_R$, $a_L$ is not above $a_R$, and all movable reticulation edges are below $a_R$. Then there exists a sequence of head moves from $N$ to $N'$ of length at most 15.*

**Proof:** Like in the proof of last lemma, we assume that $a_L$ is not above $a_R$. Because the network has at least one reticulation, we can pick a highest reticulation $r$ in the network, let $(t, r)$ be its movable edge. As each movable reticulation edge is below $a_R$, so is $(t, r)$. Let us denote the root of $N$ with $\rho$, and distinguish two subcases:

1. $\boldsymbol{x_R \neq \rho}$. Because $x_R$ is above $a_R$, it must be a tree node, it has another child edge $(x_R, b)$ with $b \neq a_R$ not above $t$: if $b$ were above $t$, there would have to be a reticulation above $r$, contradicting our choice of $r$.

   (a) $\boldsymbol{r \neq b}$. In this case, we can move $(t, r)$ to $(x_R, b)$ in both $N$ and $N'$, producing networks $M$ and $M'$. Now $(x_R, r')$ is movable in $M$, and by relabelling $t' = x_R$ we can see that there is one tail move between $M$ and $M'$ of the same type as Case 2(b)i of Lemma 13. To see this, take $r'$ as the relevant reticulation with movable edge $(t', r')$ and consider the tail move $(u, v)$ to $(x_R, a_R)$ producing $M'$. This case can therefore be solved with at most $5 + 2 = 7$ head moves.

(b) $\boldsymbol{r = b}$ **and** $\boldsymbol{(t, c(r)) \notin N}$. In this case, $(x_R, r)$ is movable, and not below $a_R$, contradicting our assumptions.

(c) $\boldsymbol{r = b}$ **and** $\boldsymbol{(t, c(r))) \in N}$. Because $N$ has at least two leaves, there must either be at least 2 leaves below $r$, or there is a leaf not below $r$. Let $l$ be an arbitrary leaf below $r$ in the first case, or a leaf not below $r$ in the second case. Note that the head move $(t, c(r))$ to the incoming edge of $l$ is allowed, and makes $(x_R, r)$ movable. Now the tail move $(u, v)$ to $(x_R, a_R)$ is still allowed, because $v \neq a_R$, $v$ is not above $x_R$ and $(u, v)$ is tail-movable. For this tail move we are in a case of Lemma 13 because $(x_R, r)$ is not below $a_R$, hence this tail move takes at most 7 moves. After this move, we can do one head move to put $(t, c(r))$ back. Hence this case takes at most 9 moves.

2. $\boldsymbol{x_R = \rho}$. Let $y, z$ be the children of $a_R$. Now first do the tail move of $(u, v)$ to one of the child edges $(a_R, z)$ of $a_R$. This is allowed because $a_R$ is the top tree node. The sequence of head moves used to do this tail move is as in the previous case. Note that $N'$ is now one tail move away: $(u', z)$ to $(a_R, y)$. This is a horizontal tail move along a tree node as in Lemma 11, which takes at most 6 head moves. As the previous case took at most 9 head moves, this case takes at most 15 head moves in total.

□

**Lemma 15** *Let $(u, v)$ from $(x_L, a_L)$ to $(x_R, a_R)$ be a valid tail move in a network $N$ resulting in a network $N'$. Suppose $a_L \neq a_R$ and $a_L$ is not above $a_R$, then there exists a sequence of head moves from $N$ to $N'$ of length at most 15.*

**Proof:** This is a direct consequence of the previous two lemmas. □

**Lemma 16** *Let $(u, v)$ from $(x_L, a_L)$ to $(x_R, a_R)$ be a valid tail move in a network $N$ resulting in a network $N'$. Suppose $a_L \neq a_R$ and $a_L$ is above $a_R$, then there exists a sequence of head moves from $N$ to $N'$ of length at most 15.*

**Proof:** Note that in this case $a_R$ is not above $a_L$ in $N'$. Reversing the labels $x_L \leftrightarrow x_R$ and $a_L \leftrightarrow a_R$ we are in the situation of Lemma 15 for the reverse tail move $N'$ to $N$. This implies the tail move can be replaced by a sequence of at most 15 head moves. □

**Theorem 2** *Any tail move can be replaced by a sequence of at most* 15 *head moves.*

**Proof:** This follows from the previous lemmas. □

## 4.2 Head move replaced by tail moves

In this section, we show how to replace a head move $(u, v)$ from $(x_1, y_1)$ to $(x_2, y_2)$ by a sequence of at most 16 tail moves (Theorem 3). In the proof, we first show how to efficiently replace downward head moves by tail moves (i.e., when $y_1$ is above $x_2$; Section 4.2.2). This is then used repeatedly to simulate arbitrary head moves in Section 4.2.3.

Unless stated otherwise, each move in this section is a tail move and movable means tail-movable.

### 4.2.1 Distance-1 head moves

We first recall a result from [17]: any distance-1 head move can be replaced by a constant number of tail moves, so the following result holds.

**Lemma 17** *Let $(u, v)$ from $(x_1, y_1)$ to $(x_2, y_2)$ be a valid distance-1 head move in a network $N$ resulting in a network $N'$. Then there is a sequence of at most 4 tail moves between $N$ and $N'$, except if $N$ and $N'$ are different networks with two leaves and one reticulation.*

And there is the following special case, for which we repeat the proof here.

**Lemma 18** *Let $(u, v)$ from $(x_1, y_1)$ to $(x_2, y_2)$ be a valid head move in a network $N$ resulting in a network $N'$. Suppose that $y_1 = x_2$ and $x_2$ is a* tree node*, then there is a sequence of at most 1 tail moves between $N$ and $N'$.*

**Proof:** Let $c(x_2)$ be the other child of $x_2$ (not $y_2$), then the tail move $(x_2, c(x_2))$ to $(x_1, v)$ suffices. □

### 4.2.2 Downward head moves

Now, we prove that the head move can be replaced by a sequence of constant length if $y_1$ is above $x_2$. We start by considering the case that $x_2$ is a tree node. In the proof we use a constant number of moves to create a situation where we simply need to do a distance-1 downward head move.

**Lemma 19** *Let $(u, v)$ from $(x_1, y_1)$ to $(x_2, y_2)$ be a valid head move in a network $N$ resulting in a network $N'$. Suppose that $y_1$ is above $x_2$, $y_1 \neq x_2$, and $x_2$ is a tree node, then there is a sequence of at most 4 tail moves between $N$ and $N'$.*

**Proof:** We split this proof in two cases: $(x_2, y_2)$ is movable, or it is not. We prove in both cases there exists a constant length sequence of tail moves between $N$ and $N'$.

1. **$(x_2, y_2)$ is tail-movable.** Tail move $(x_2, y_2)$ up to $(v, y_1)$, this is allowed because any tail move up is allowed if the moving edge is tail-movable (Corollary 1). Now $(u, v)$ is still head-movable, hence we can move it down to $(x_2', y_2)$. As this is exactly the situation of Lemma 18, we can replace this head move by one tail move. Now tail-moving $(x_2', v')$ back down results in $N'$, so this move is allowed, too. Hence there is a sequence of 3 tail moves between $N$ and $N'$.

2. **$(x_2, y_2)$ is not tail-movable.** Because $x_2$ is a tree node and $(x_2, y_2)$ is not movable, there has to be a triangle with $x_2$ at the side, formed by the parent $p$ of $x_2$ and the other child $c$ of $x_2$. Note that $(p, x_2)$ is tail-movable, and that it can be moved up to $(v, y_1)$. After this move, Lemma 18 tells us we can head-move $(u, v)$ to $(p', x_2)$ using one tail move. The next step is to tail move $(p', v')$ back down to the original position of $p$. The resulting network is allowed because it is one valid distance-1 head move away from $N'$ (as $c$ is not above $u$). Lastly, we do this distance-1 head move, which again can be simulated by one tail move by Lemma 18. Note that this sequence is also valid if $p = y_1$. Hence there is a sequence of at most 4 tail moves between $N$ and $N'$ (Figure 16).

□

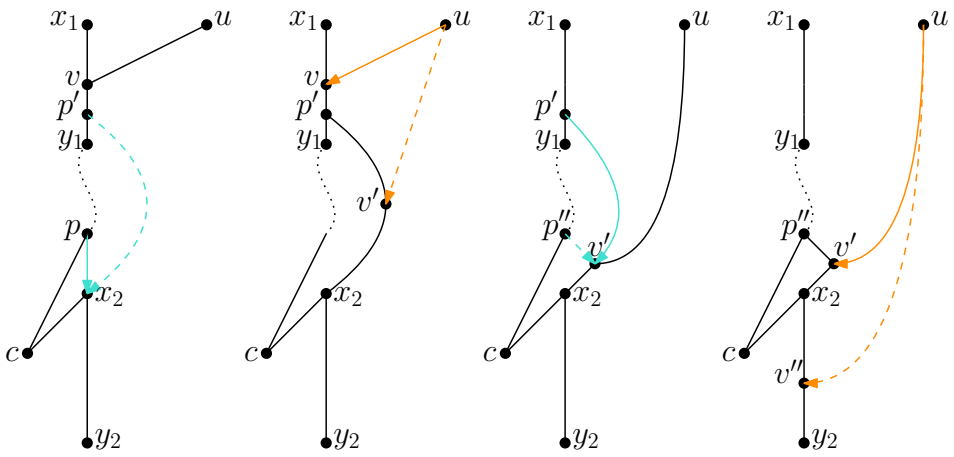



Figure 16: The four moves used in Case 2 of Lemma 19. The two same coloured edges of which one is dashed in each part are the location of an edge before and after a move.

**Lemma 20** *Let $(u, v)$ from $(x_1, y_1)$ to $(x_2, y_2)$ be a valid head move in a network $N$ resulting in a network $N'$. Suppose that $y_1$ is (strictly) above $x_2$ and $x_2$ is a reticulation, then there are networks $M$ and $M'$ such that the following hold:*

1. *turning $N$ into $M$ takes at most one tail move;*
2. *turning $N'$ into $M'$ takes at most one tail move;*
3. *there is a head move between $M$ and $M'$, moving the head* down *to an edge whose top node is a reticulation;*
4. *there is a tail-movable edge $(s, t)$ in $M$ with $t$ not above $x_2$.*

**Proof:** Note that we have to find a sequence consisting of a tail move followed by a head move and finally a tail move again, between $N$ and $N'$ such that the head move is of the desired type and the network after the first tail move has a movable edge not above the top node $x_2$ of the receiving edge of the head move.

Note that if there is a tail-movable edge $(s, t)$ in $N$ with $t$ not above $x_2$, we are done by the previous lemmas: take $M := N$ and $M' := N'$. Hence we may assume that there is no such edge in $N$. Suppose all leaves (of which there are at least 2) are below $y_2$, then there must also be a tree node below $y_2$. And as one of its child edges is movable, there is a tail-movable edge below $y_2$ (and hence not above $x_2$). So if all leaves are below $y_2$, we can again choose $M := N$ and $M' := N$.

Because our networks have at least 2 leaves, the remaining part is to show the lemma assuming that there is a leaf $l_1$ not below $y_2$. Note that there also exists a leaf $l_2$ below $y_2$. Now consider an LCA $j$ of $l_1$ and $l_2$. We note that $j$ is a tree node of which at least one outgoing edge **$(j, m)$ is not above $x_2$**. If $(j, m)$ is tail-movable, then $M := N$ and $M' := N'$ suffices, so assume **$(j, m)$ is not tail-movable**. Let $i$ be the parent of $j$, and $k$ be the other child of $j$; because $j$ is a tree node and $(j, m)$ is not movable, ***i*, *j* and *k* form a triangle** (Figure 17).

The idea is to 'break' this triangle with one tail move in $N$ and $N'$ simultaneously, meaning we either move one of the edges of the triangle, or we move a tail to an edge of the triangle. If we

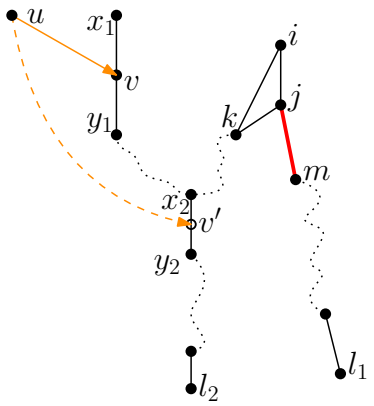



Figure 17: The situation of Lemma 20 in which we want to make the red edge $(j, m)$ movable, in the network before (with orange solid line) and after (with orange dashed line) the head move. Dotted black lines indicate ancestral relations, but are not necessarily edges in the network.

can break the triangle in both networks keeping $(u, v)$ movable, creating new networks $M$ and $M'$, then choosing $(s, t) := (j, m)$ in $M$ will work. The last part of this proof shows how we do this. We have to split in two cases:

- ***i* is the child of the root.** In this case we break the triangle by moving a tail to the triangle. As $v$ is a reticulation and there is no path from any node below $m$ to $v$ (if so, there is a path from $m$ to $x_2$), there must be a tree node $p$ below $k$ and (not necessarily strictly) above both parents of $v$. At least one of the outgoing edges $(p, q)$ is movable in $N$. If $v$ is a child of $p$ and $(p, v)$ is movable, then we choose $q = v$, otherwise any choice of $(p, q)$ will suffice.

  Because $(p, q)$ is movable (by choice of $(p, q)$) and $k$ is above $p$, the tail move $(p, q)$ to $(j, k)$ is valid. Now the head move $(u, v)$ (or $(u', v)$ if $p = u$ in $N$) to $(x_2, y_2)$ is valid, because $x_2$ is below $v$, and $(u, v)$ is movable because $(u, v)$ was movable in $N$, and the only ways to create a triangle with $v$ on the side with one tail move are:

  - suppressing one node of a four-cycle that includes $v$ to create a triangle by moving the outgoing edge of that node that is not included in the four-cycle. As this node is $p$, and $p$ is above both parents of $v$, the suppressed node must be on the incoming edge of $v$ in the four-cycle (Figure 18 top). However, in that case $v$ is a child of $p$ and $(v, p)$ is tail-movable, so we choose to move $(v, p)$ up for the first move, which keeps $(u, v)$ head-movable.
  - moving the other incoming edge of $v$ (not $(u, v)$) to the other incoming edge of the child $c(v)$ of $v$ (so not $(v, c(v))$). But as the tail move moves $(p, q)$ to $(j, k)$, we see that $k = c(v)$ which contradicts the fact that $v$ is strictly below $k$ in $N$. Hence this cannot result in a triangle with $v$ on the side (Figure 18 bottom left).
  - moving the other incoming edge of the child $c(v)$ of $v$ (so not $(v, c(v))$) to the incoming edge of $v$ that is not $(u, v)$. As we move $(p, q)$ to $(j, k)$, we see that $v = k$ and $u = i$. But then $c(v) = q$ must be below the other child $m$ of $j$, and as $x_2$ is below $q$, this

contradicts the fact that $(j, m)$ is not above $x_2$. Hence this cannot result in a triangle with $v$ on the side (Figure 18 bottom right).

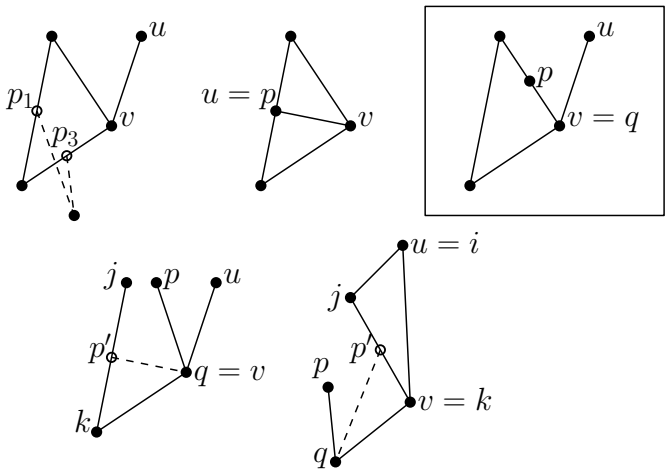



Figure 18: The ways of making $(u, v)$ not head-movable in Lemma 20. Top: creating a triangle by suppressing a node in a four cycle. The first two of these are invalid because $p$ is not above both parents of $v$. The right one does not give any contradictions, but forces us to choose to move $(p, v)$, so that no triangle is produced. Bottom: creating a triangle by moving an edge to become part of the triangle. Both these options contradict our assumptions.

The preceding shows that $(u, v)$ is still head-movable after the first tail move. Because $p$ is above $x_2$ through two paths, $y_1$ is still above $x_2$ after the tail move $(p, q)$ to $(i, j)$. Also we did not change $x_2$, so it still is a reticulation. This means that the head move $(u, v)$ to $(x_2, y_2)$ is still valid and of the right type. Furthermore $(j, m)$ is a tail-movable edge with $m$ not above $x_2$. Now note that after the head head move $(u, v)$ to $(x_2, y_2)$, we can move $(p', q)$ back to its original position to obtain $N'$.

Hence we produce $M$ by tail-moving $(p, q)$ to $(i, j)$ and $M'$ by moving the corresponding edge to $(i, j)$ in $N'$. We can do this because $(i, j)$ is still an edge in $N'$: indeed it is not subdivided by the head move, and $i$ and $j$ are both tree nodes, so they do not disappear either. So this case is proven.

- ***i* is not the child of the root.** In this case we can move the tail of $(i, k)$ (possibly equal to $(u, v)$) up to the root in $N$. Now note that $j$ is a tree node, so the tail move cannot create any triangles with a reticulation on the side. This means that $(u, v)$ is still movable after the tail move. Furthermore, after the tail move $x_2$ is still a reticulation node below $y_1$, and $(j, m)$ is movable and not above $x_2$. Hence the head move $(u, v)$ to $(x_2, y_2)$ is allowed and of the appropriate type. Now moving the tail of $(i', k)$ back to the incoming edge of $j$, we get $N'$.

  Hence this case works with $M$ being the network obtained by moving $(i, k)$ up to the root edge in $N$, and $M'$ the network obtained by moving $(i, j)$ up to the root edge in $N'$.

□

**Lemma 21** *Let $(u, v)$ from $(x_1, y_1)$ to $(x_2, y_2)$ be a valid head move in a network $N$ resulting in a network $N'$. Suppose that $y_1$ is above $x_2$ and $x_2$ is a reticulation, then there is a sequence of at most 8 tail moves between $N$ and $N'$.*

**Proof:** By Lemma 20, with cost of 2 tail moves, we can assume there is a tail-movable edge $(s, t)$ that can be moved to $(x_2, y_2)$. Make this the first move of the sequence. Because the head move $(u, v)$ to $(s', y_2)$ goes down, and $(u, v)$ is head-movable, this head move is allowed. By Lemma 19, there is a sequence of at most 4 tail moves simulating this head move. Now we need one more tail move to arrive at $N'$: the move putting $(s, t)$ back to its original position. This all takes at most 8 moves. □

All previous lemmas together give us the following result.

**Proposition 3** *Let $(u, v)$ from $(x_1, y_1)$ to $(x_2, y_2)$ be a valid head move in a network $N$ resulting in a network $N'$. Suppose that $y_1$ is above $x_2$ or $y_2$ is above $x_1$, then there is a sequence of at most 8 tail moves between $N$ and $N'$.*

### 4.2.3 Non-downward head moves

Finally, we consider head moves where the original position of the head and the location it moves to are incomparable.

**Proposition 4** *Let $(u, v)$ from $(x_1, y_1)$ to $(x_2, y_2)$ be a valid head move in a network $N$ resulting in a network $N'$, where $N$ and $N'$ are not networks with two leaves and one reticulation. Suppose that $y_1$ is not above $x_2$ and $y_2$ is not above $x_1$, then there is a sequence of at most 16 tail moves between $N$ and $N'$.*

**Proof:** Find an LCA $s$ of $x_1$ and $x_2$. We split into different cases for the rest of the proof:

1. $\boldsymbol{s \neq x_1, x_2}$**.** One of the outgoing edges $(s, t)$ of $s$ is tail-movable and it is not above one of $x_1$ and $x_2$. Suppose $t$ is not above $x_1$, then we can do the following (Figure 19):

   - Tail move $(s, t)$ to $(x_1, v)$;
     allowed because $t \neq v$, $(s, t)$ movable, and $t$ not above $x_1$.
   - Tail move $(s', v)$ to $(x_2, y_2)$;
     allowed because $(x_1, t) \notin N$: otherwise $x_1$ was the only LCA of $x_1$ and $x_2$; $y_1$ and hence $v$ is not above $x_2$; $(x_2, y_2) \neq (u, v)$.
   - Distance-1 head move $(u, v)$ to $(s'', y_2)$;
     No parallel edges by removal: if so, they are between $s''$ and $y_1 = y_2$, but then the move actually resolves this; no parallel edges by placing: $u \neq s''$; no cycles: $y_2$ not above $u$, otherwise cycle in $N'$
   - Move $(s'', y_1)$ back up to $(x_1, t)$;
     Moving a tail up is allowed if the tail is movable.
   - Move $(s''', t)$ back up to its original position.
     Moving a tail up is allowed if the tail is movable.

   As the head move used in this sequence is a distance-1 move, it can be simulated with at most 4 tail moves. Hence the sequence for this case takes at most 8 tail moves.

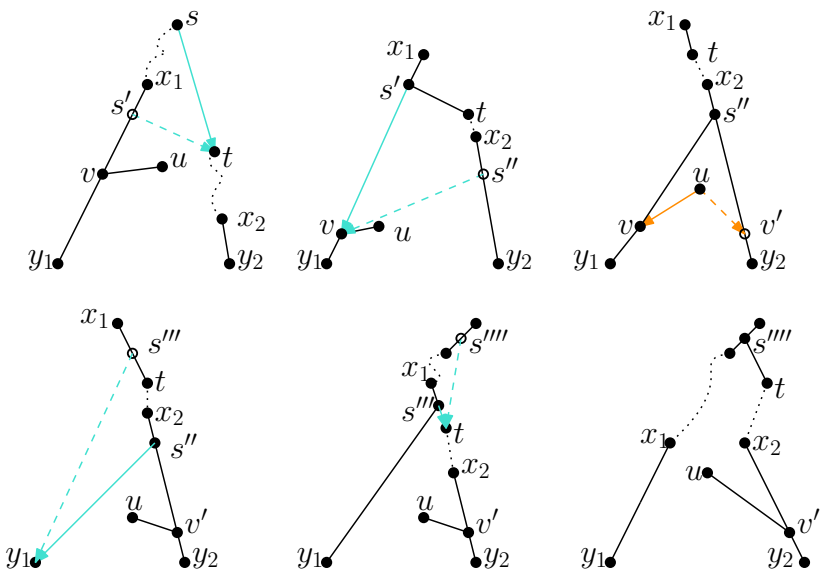



Figure 19: The sequence of moves used in Case 1 of Proposition 4.

2. $s = x_1$.

   (a) **$u$ is not below $x_1$.** Head move $(u, v)$ to the other child edge of $x_1$, this takes at most 4 tail moves by Lemma 17. Now we have to move the head of $(u, v')$ down to create $N'$, this takes at most 8 tail moves by Proposition 3. Hence for this case we need at most 12 tail moves.

   (b) **$u$ is below $x_1$.** In this case the previous approach is not directly applicable, as moving the head of $(u, v)$ to the other child edge of $x_1$ creates a cycle. Hence we need to take a different approach, where we distinguish the following cases:

      i. **$(x_1, v)$ is tail-movable.** Tail move $(x_1, v)$ down to $(x_2, y_2)$, this is allowed because $y_1$ is not above $x_2$. Then do the sideways head move $(u, v)$ to $(x_1', y_2)$, this takes at most 4 tail moves. Then move $(x_1', y_1)$ back up to create $N'$. This takes at most 6 moves

      ii. **$(u, v)$ is tail-movable.** Move $(u, v)$ up to the incoming edge $(t, s)$ of $s$. The head move $(u', v)$ to $(x_2, y_2)$ is still allowed, except if $s, u, v$ form a triangle with the child of $u$ as well as of $v$ being $y_1$ in $N$, but in that case $x_1$ was not the LCA of $x_1$ and $x_2$. Hence we can simulate the head move with at most 12 tail moves by Case 2a of this analysis. As afterwards we can move the tail of $(u', v')$ back to its original position, this case takes at most 16 moves.

      iii. **Neither $(x_1, v)$ nor $(u, v)$ is tail-movable.** We create the situation of Case 1 by reversing the direction of the triangle at $x_1$, this takes at most 4 tail moves because it is a head move. Only if the bottom node of the triangle is $x_2$, we do not get this situation, but then the head move is composed of two head moves, so it can be simulated with 8 tail moves. If we are actually in the situation of Case 1, simulate the head move with at most 8 moves as done in that case. This is allowed because

it produces $N'$ with the direction of a triangle reversed, which is a valid network. Then reverse the direction of the triangle again using at most 4 tail moves. This way we obtain $N'$ with at most 16 tail moves (Figure 20).

3. $s = x_2$. This can be achieved with the reverse sequence for the previous case.

□

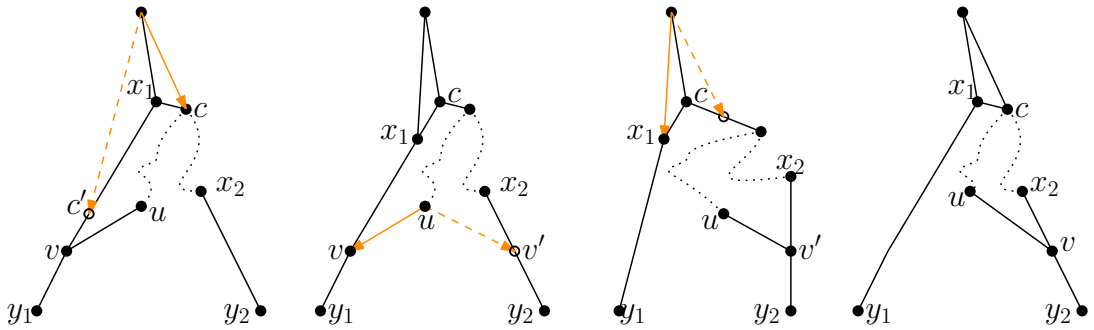



Figure 20: The sequence of moves used in Case 2(b)iii of Proposition 4.

Proposition 4 and Proposition 3 directly imply the following theorem

**Theorem 3** *Suppose there is a head move turning $N$ into $N'$, and $N$ and $N'$ are not non-isomorphic tier-1 networks with at least 2 leaves. Then there is a sequence of at most* 16 *tail moves between $N$ and $N'$.*

# 5 Head move diameter and neighbourhoods

## 5.1 Diameter bounds

There are some obvious results concerning the diameter of head move space found using results from Section 4 and existing bounds on the rSPR diameter. Each rSPR sequence of length $l$ can be replaced by a sequence of head moves of length at most $15l$. Hence, we get upper bounds $\Delta^k_{\text{Head}} \leq 15\Delta^k_{\text{rSPR}}$ on head move diameters. Furthermore, each sequence of head moves is also an rSPR sequence, hence the rSPR diameter gives lower bounds $\Delta^k_{\text{rSPR}} \leq \Delta^k_{\text{Head}}$. Similarly the rSPR bounds give bounds on the tail move diameters. These bounds for tail move diameters are inferior to the bounds in [17]. The tail move diameter bounds from that paper are obtained using a technique where an isomorphism is built incrementally.

In this section we prove a bound for the head move diameter (Theorem 4). The proof employs a technique similar to the one used for tail moves: for any pair of networks, we build an isomorphism between growing subnetworks where in each step we only have to use a small number of moves to grow the isomorphism (Lemma 22). For tail moves and rSPR moves, it is convenient to build this isomorphism bottom-up. Head moves are essentially upside-down tail moves. Hence, for head moves, we build an isomorphism starting at the top. Doing this, we ignore the leaf labels. Consequently, to prove the bound we permute the leaf labels using a small number of head moves (Lemma 23).

Each move in this section is a head move, unless stated otherwise. As we need to explicitly work with the vertices and edges of different networks, we denote a network with nodes $V$ and edges $A$ as $N = (V, A)$. We first define a few structures that we use extensively: upward closed sets, isomorphisms, and induced graphs.

**Definition 16** *Let $N = (V, A)$ be a network with $Y \subseteq V$ a subset of the vertices. We say that $Y$ is* upward closed *if for each $u \in Y$ the parents of $u$ are also in $Y$.*

**Definition 17** *Let $N = (V, A)$ and $N' = (V', A')$ be two directed acyclic graphs, then a map $\phi : V(N) \to V(N')$ is an* (unlabelled) isomorphism *if $\phi$ is bijective and $(u, v) \in A$ if and only if $(\phi(u), \phi(v)) \in A(N')$. If such an isomorphism exists, we say that $N$ and $N'$ are* (unlabelled) isomorphic*. If, additionally, there are labellings $l_N : X \to V(N)$ and $l_{N'} : X \to V(N')$ of the vertices and $\phi(l_N(x)) = l_{N'}(x)$ for all $x \in X$ then $N$ and $N'$ are* labelled isomorphic*.*

**Definition 18** *Let $N = (V, A)$ be a network and $Y \subseteq V$ a subset of the vertices, then $N[Y]$ denotes the directed subgraph of $N$ induced by $Y$:*

$$N[Y] := (Y, A \cap (Y \times Y))$$

**Lemma 22** *Let $N_1$ and $N_2$ be tier $k > 0$ networks with label set $X$ of size $n$, then there exists a pair of head move sequences $S_1$ on $N_1$ and $S_2$ on $N_2$ such that the resulting networks are unlabelled isomorphic and the total length is $|S_1| + |S_2| \leq 4n + 6k - 4$.*

**Proof:** We incrementally build upward closed sets $Y_1 \subseteq V(N_1)$ and $Y_2 \subseteq V(N_2)$ such that $N_1[Y_1]$ and $N_2[Y_2]$ are unlabelled isomorphic with isomorphism $\phi$. Starting with $Y_1 = \{\rho_1\}$ and $Y_2 = \{\rho_2\}$ the roots only, we set the isomorphism $\rho_1 \mapsto \rho_2$. Next we increase the size of $Y_1$ by changing the networks slightly with a constant number of head moves, and then adding a node to $Y_1$ and $Y_2$ and extending the isomorphism. We will add all the leaves to the isomorphism last.

1. **There is a highest node $x_1$ of $N_1$ not in $Y_1$ such that $x_1$ is a tree node.** Because $x_1$ is a highest node not in $Y_1$, the parent $p_1$ of $x_1$ is in $Y_1$ and there is a corresponding node $p_2 := \phi(p_1)$ in $Y_2$. This node must have at least one child $x_2$ that is not in $Y_2$, as otherwise the degrees of $p_1$ and $p_2$ in $N_1[Y_1]$ and $N_2[Y_2]$ do not coincide.

   (a) **The node $x_2$ is a tree node.** In this case we can add $x_1$ and $x_2$ to $Y_1$ and $Y_2$ and set $\phi : x_1 \mapsto x_2$ to get an extended isomorphism. We do not have to use any head moves to do this extension.

   (b) **The node $x_2$ is a reticulation.** We make sure $p_2$ has a tree node $y_2$ as a child not in $Y_2$, using at most 3 head moves. We can then add $x_1$ to $Y_1$ and $y_2$ to $Y_2$ and extend the isomorphism with $x_1 \mapsto y_2$. To create this tree node, we use a tree node $c_2 \in N_2 \setminus Y_2$, which exists because there is a tree node in $N_1 \setminus Y_1$.

      i. **The edge $(p_2, x_2)$ is movable.** Move $(p_2, x_2)$ to the incoming edge $(t_2, c_2)$ of the tree node $c_2$. This move is valid because $c_2$ cannot be above $p_2$ (otherwise $c_2 \in Y_2$, a contradiction), and $t_2 \neq p_2$ as otherwise $p_2$ would have a tree node child not in $Y_2$. Now the edge $(t_2, x_2')$ is movable to any of the outgoing edges of $c_2$. Now $p_2$ has child node $c_2$, which is a tree node, so we can extend the isomorphism with a tree node $\phi : x_1 \mapsto c_2$ using at most 2 head moves (Figure 21).

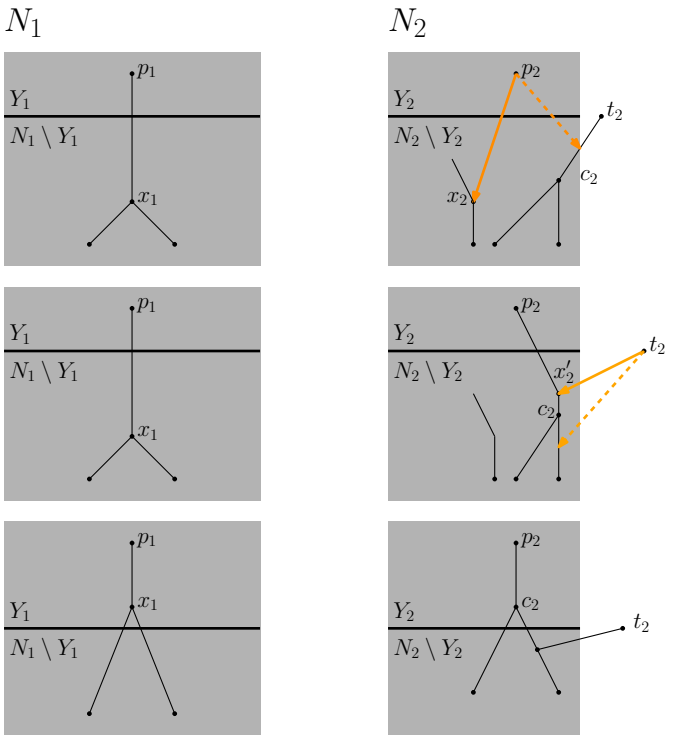



Figure 21: The moves and incremented isomorphism for Lemma 22 Case 1(b)i. For nodes outside of the shaded region, it is not known whether they are in $Y_2$.

ii. **The edge $(p_2, x_2)$ is not movable.** This means that $x_2$ is on the side of a triangle. Denote by $d_2$ the child of $x_2$ and the other parent of $x_2$ with $z_2$. Now note that $(z_2, d_2)$ is movable, and can be moved to an edge $(u_2, v_2)$ with $v_2$ not in $Y_2$ and $(u_2, v_2)$ distinct from both $(x_2, d_2)$ and from the outgoing edge of $d_2$. Such an edge exists: pick a leaf $l$ not equal to the child of $d_2$ (if that node is a leaf); as we add all leaves to the isomorphism last, the leaf is not in $Y_2$, furthermore, $l$ is not above $z_2$, and the incoming edge of $l$ is not equal to $(x_2, d_2)$ nor to the outgoing edge of $d_2$. Doing the head move $(z_2, d_2)$ to the incoming edge of $l$ creates the situation of the previous case (Case 1(b)i), and we can use 2 more head moves to create a network with a tree node $c_2$ below $p_2$ which maintains the isomorphism of the upper part $Y_2$. Hence we can extend the isomorphism with a tree node $\phi : x_1 \mapsto c_2$ using at most 3 head moves.

(c) **The node $x_2$ is a leaf.** Again, note there is a tree node $c_2$ in $N_2 \setminus Y_2$, and let its parent be $t_2$. Note also that $N_2$ has a reticulation node $r_2$ with incoming edge $(s_2, r_2)$ which is movable to $(p_2, x_2)$ (if $p_2 = s_2$, then the other incoming edge $(s_2', r_2)$ is also movable, and can instead be moved to $(p_2, x_2)$).

i. **The nodes $s_2$ and $t_2$ are different nodes.** First move $(s_2, r_2)$ to $(p_2, x_2)$. Now the edge $(p_2, r_2')$ is movable, and can be moved to $(t_2, c_2)$, because $c_2$ is not above $p_2$ and $p_2 \neq t_2$ (otherwise $p_2$ has a tree node as child). This makes $(t_2, r_2'')$ movable,

and we can move it to $(s_2, x_2)$ because $s_2 \neq t_2$ and $x_2$ is a leaf, so it is not above $t_2$. Lastly, we restore the reticulation by moving $(s_2, r_2''')$ back to its original position. Hence, in this situation, 4 head moves suffice to make $p_2$ the parent of a tree node $c_2$, so that we can extend the isomorphism by $\phi : x_1 \mapsto c_2$ with a tree node (Figure 22).

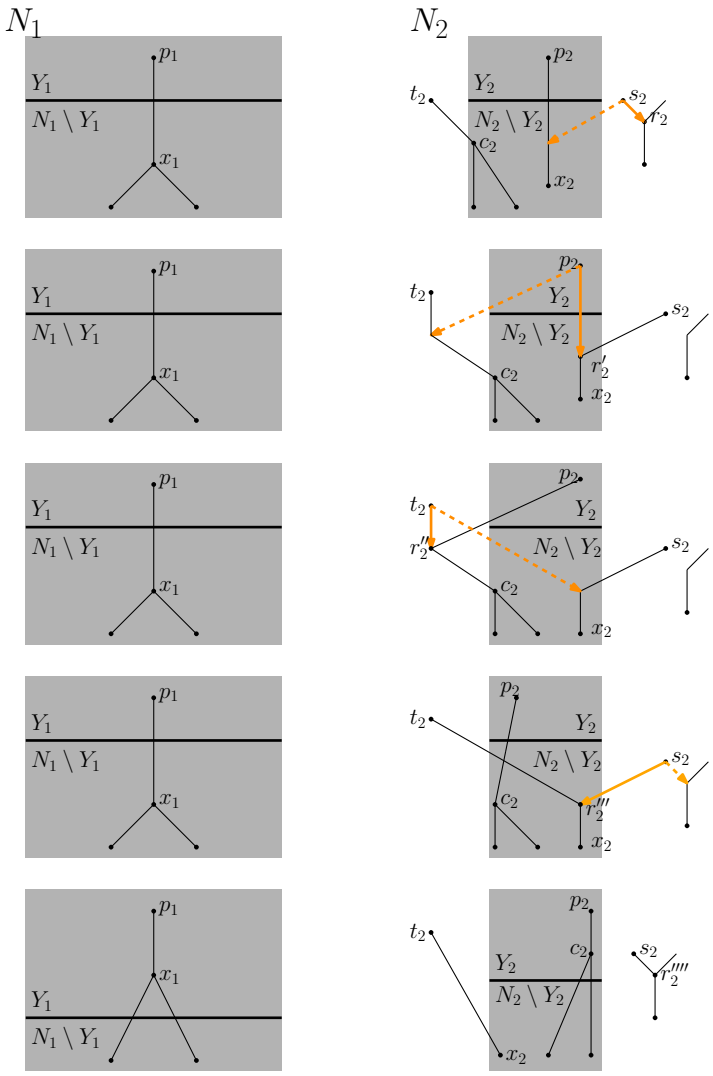



Figure 22: The moves and incremented isomorphism for Lemma 22 Case 1(c)i. For nodes outside of the shaded region, it is not known whether they are in $Y_2$.

ii. **The nodes $s_2$ and $t_2$ are the same.** Note that a child of $t_2$ is a tree node and a child of $s_2$ is a reticulation. This means that $s_2 = t_2$ is a tree node, as it has two distinct children. The edge $(s_2, r_2)$ can be moved to the pendant edge $(p_2, x_2)$. Now the new edge $(p_2, r_2')$ can be moved to $(s_2, c_2)$, because $p_2 \neq s_2$ and $c_2$ is not above $p_2$ (otherwise $c_2$ has to be in $Y_2$, contradicting our assumption). Now we

can move $(s_2, r_2'')$ back to its original position. This all takes three head moves, and makes sure that a child $c_2$ of $p_2$ is a tree node. This means we can extend the isomorphism by setting $\phi : x_1 \mapsto c_2$ (and if $r_2$ was in $Y_2$, changing $\phi : \phi^{-1}(r_2) \mapsto r_2$ to $\phi : \phi^{-1}(r_2) \mapsto r_2'''$) using at most 3 head moves to add a tree node

2. **There is a highest node $x_2$ of $N_2$ not in $Y_2$ such that $x_2$ is a tree node.** Do the same as in the previous case (Case 1) switching the roles of $N_1$ and $N_2$.

3. **Each highest node $x_1$ of $N_1$ not in $Y_1$ and $x_2$ of $N_2$ not in $Y_2$ is a reticulation node or a leaf.**

   (a) **There exists a highest node $x_1$ of $N_1$ not in $Y_1$ which is a reticulation node.** This means the two parents $p_1$ and $q_1$ of $x_1$ are in $Y_1$, and consequently have corresponding nodes $p_2$ and $q_2$ in $Y_2$. Both these nodes also have at least one child not in $Y_2$, say $c_2^p$ and $c_2^q$.

      i. **The children of $p_2$ and $q_2$ are equal (i.e., $c_2^p = c_2^q$).** In this case, we can immediately extend the isomorphism with $\phi : x_1 \mapsto c_2^p$.

      ii. **Both nodes $c_2^p$ and $c_2^q$ are reticulations.** Assume without loss of generality that $c_2^p$ is not below $c_2^q$.

         A. **The edge $(p_2, c_2^p)$ is movable.** Move this edge to $(q_2, c_2^q)$, which is allowed because $c_2^q$ is not above $p_2$, and $p_2 \neq q_2$. Now $p_2$ and $q_2$ have a common child $x_2 := c_2'^p$, so we can add one reticulation to $Y_1$ and $Y_2$ and extend the isomorphism by $\phi : x_1 \mapsto x_2$ using 1 head move.

         B. **The edge $(p_2, c_2^p)$ is not movable.** Because $(p_2, c_2^p)$ is not movable, $c_2^p$ must be the side node of a triangle, and therefore its outgoing edge $(c_2^p, z)$ is movable. By our assumption, $c_2^q$ is not above $c_2^p$, so we can move $(c_2^p, z)$ to $(q_2, c_2^q)$. Now the other incoming edge $(t, c_2^p)$ of $c_2^p$ becomes movable, and we can move it down to $(z', c_2^q)$. Now $p_2$ and $q_2$ have a common child $x_2 := z'$, and the isomorphism can be extended with one reticulation by setting $\phi : x_1 \mapsto x_2$ using at most 2 head moves (Figure 23).

      iii. **The node $c_2^p$ is a reticulation, and $c_2^q$ is a leaf.** The subcases here work exactly like the previous subcases in Case 3(a)ii.

         A. **The edge $(p_2, c_2^p)$ is movable.** Move this edge to $(q_2, c_2^q)$, which is allowed because $c_2^q$ is not above $p_2$, and $p_2 \neq q_2$. Now $p_2$ and $q_2$ have a common child $x_2 := c_2'^p$, so we can add one reticulation to $Y_1$ and $Y_2$ and extend the isomorphism by $\phi : x_1 \mapsto x_2$ using one head move.

         B. **The edge $(p_2, c_2^p)$ is not movable.** Because $(p_2, c_2^p)$ is not movable, $c_2^p$ must be the side node of a triangle, and therefore its outgoing edge $(c_2^p, z)$ is movable. Because $c_2^q$ is a leaf, it is not above $c_2^p$, so we can move $(c_2^p, z)$ to $(q_2, c_2^q)$. Now the other incoming edge $(t, c_2^p)$ of $c_2^p$ becomes movable, and we can move it down to $(z', c_2^q)$. Now $p_2$ and $q_2$ have a common child $x_2 := z'$, and the isomorphism can be extended with one reticulation by setting $\phi : x_1 \mapsto x_2$ using at most 2 head moves.

      iv. **The node $c_2^q$ is a reticulation, and $c_2^p$ is a leaf.** Switch the roles of $p_2$ and $q_2$ and do as in the previous case.

      v. **Both nodes $c_2^p$ and $c_2^q$ are leaves.** Note that because $x_1$ is a reticulation node not in $Y_1$, there must also be a reticulation node $r_2 \in N_2$ not in $Y_2$. Let its movable

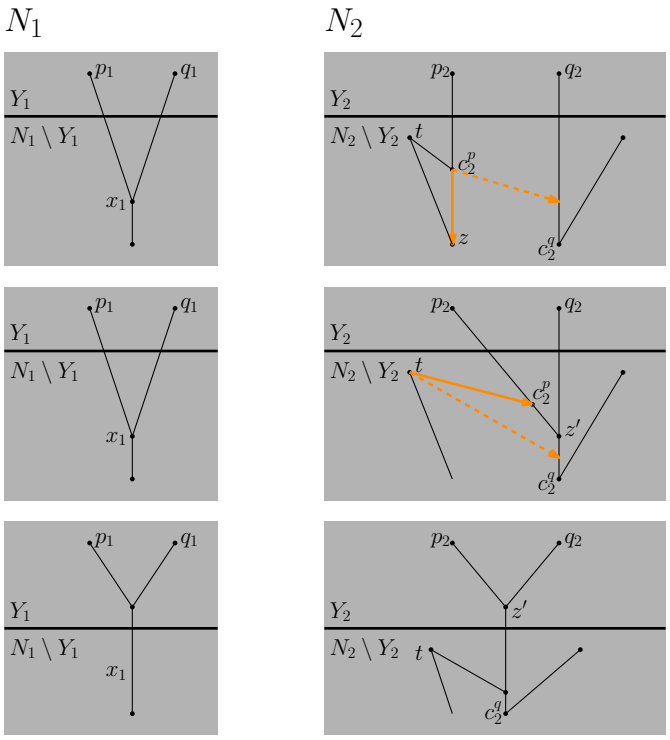



Figure 23: The moves and incremented isomorphism for Lemma 22 Case 3(a)iiB.

incoming edge be $(s_2, r_2)$. As $p_2 \neq q_2$ we know that $s_2$ can be equal to at most one of $p_2$ and $q_2$, hence we can assume without loss of generality that $s_2 \neq p_2$. Then the head move $(s_2, r_2)$ to $(p_2, c_2^p)$ is allowed, because the leaf $c_2^p$ cannot be above $s_2$. Now $(p_2, r_2')$ is movable because the child of $r_2'$ is a leaf, and it can be moved to $(q_2, c_2^q)$ because $p_2 \neq q_2$ and $c_2^q$ is a leaf, and hence not above $p_2$. After this head move, $p_2$ and $q_2$ have a common child $x_2 := r_2''$, and the isomorphism can be extended with one reticulation by setting $\phi : x_1 \mapsto x_2$ using at most 2 head moves.

(b) **There exists a highest node $x_2$ of $N_2$ not in $Y_2$ which is a reticulation node.** Do the same as in the previous case, switching the roles of $N_1$ and $N_2$.

(c) **All highest nodes of $N_1$ not in $Y_1$ and of $N_2$ not in $Y_2$ are leaves.** In this case, the networks are already unlabelled isomorphic: $N_1[Y_1]$ and $N_2[Y_2]$ are isomorphic, and the only nodes not part of the isomorphism are leaves, hence there is only one way (ignoring symmetries of cherries) to complete the isomorphism.

Note that this procedure first adds all tree nodes and reticulations to the isomorphism, using four moves per tree node and two moves per reticulation node at most. Then finally it adds all the leaves, without changing the networks any more. Noting that the number of tree nodes is $n+k-1$, we see that we need to do at most $4(n+k-1)+2k = 4n+6k-4$ moves in $N_1$ and $N_2$ to get $N_1'$ and $N_2'$ which are unlabelled isomorphic. □

**Lemma 23** *Let $N$ and $N'$ be tier $k > 0$ networks with label set $X$ of size $n$, which are unlabelled isomorphic. Then there is a head move sequence from $N$ to $N'$ of length at most $2n$.*

**Proof:** Note that the only difference between $N$ and $N'$ is a permutation of the leaves, say $\pi = (l_1^1, \ldots, l_{\Pi_1}^1)(l_1^2, \ldots, l_{\Pi_2}^2) \cdots (l_1^P, \ldots, l_{\Pi_q})$ to get from $N$ to $N'$ (where all $l_i^j$ are distinct). Note also that there is a reticulation in $N$ with a head-movable edge $(t, r)$, which is movable to the incoming edge of any leaf. A sequence of moves from $N$ to $N'$ consists of the moves

- $(t, r)$ to $(p(l_{\Pi_j}^j), l_{\Pi_j}^j)$;
- $(p(l_{\Pi_j}^j), r^{(1)'})$ to $(p(l_{\Pi_j-1}^j), l_{\Pi_j-1}^j)$;
- $(p(l_{\Pi_j-1}^j), r^{(2)'})$ to $(p(l_{\Pi_j-2}^j), l_{\Pi_j-2}^j)$;
- ...
- $(p(l_2^j), r^{(\Pi_j-1)'})$ to $(p(l_1^j), l_1^j)$;
- $(p(l_1^j), r^{(\Pi_j)'})$ to $(t, l_{\Pi_j}^j)$;
- $(t, r^{(\Pi_j+1)'})$ to $(s, c)$,

for each cycle ($1 \leq j \leq q$) of $\pi$, where $c$ is the child of $r$ in $N$ and $s$ is the other parent of $r$ in $N$. This permutes the leaves in $N$ by $\pi$ so that the resulting network is $N'$. The sequence is allowed provided no two subsequent leaves in a cycle have a common parent (e.g., $p(l_i^j) = p(l_{i-1}^j)$). There is always a permutation in which this does not happen. Indeed, if this were to happen, the two leaves would be in a cherry. The worst case is attained when there are a maximal number of cycles in the permutation, which happens when $\pi$ consists of only 2-cycles. In such a case there will be $n/2$ cycles of length 2. Each such a cycle takes four moves. An upper bound to the length of the sequence is therefore $4(n/2) = 2n$. □

A direct corollary of the previous two lemmas is the following theorem, giving an upper bound on the diameter of head move space. To see this, note that any head move is reversible, and hence we can concatenate sequences in different directions.

**Theorem 4** *Let $N$ and $N'$ be tier $k > 0$ networks with label set $X$ of size $n$, then there is a head move sequence of length at most $6n + 6k - 4$ between $N$ and $N'$.*

## 5.2 Neighbourhood size

In this subsection, we consider a third property of phylogenetic network space: the neighbourhood size. We start by giving simple upper bounds on the head move neighbourhood size, and then compare these to known bounds for other moves. For complete comparisons, we need information about the smallest and the largest neighbourhood size in each tier. However, giving lower and upper bounds for both of these lies beyond the scope of this paper. Hence, we focus on upper bounds for the largest neighbourhood in a tier, as these would, in practice, be limiting.

**Proposition 5** *The size of the head move neighbourhood of a network with $n$ leaves and $k$ reticulations is at most $4kn + 6k^2 - 2k$, the size of the distance-1 head move neighbourhood is at most $8k$, and the size of the distance-2 head move neighbourhood is at most $24k$.*

**Proof:** Head moves can only move reticulation edges, of which there are $2k$ in a tier-$k$ network. Furthermore, there are $2n+3k-1$ edges in a tier-$k$ network with $n$ leaves. Hence, an upper bound on the head move neighbourhood size in a tier-$k$ network with $n$ leaves is $4kn+6k^2-2k$. An upper bound on the size of the distance-1 head move neighbourhood is $8k$: there are $2k$ heads that can be moved, and for each head, there are at most four adjacent edges it can be moved to. Similarly, an upper bound on the size of the distance-2 head move neighbourhood is $24k$, as there are at most twelve edges within distance-2 of a node. □

### 5.2.1 Comparison with tail move neighbourhood

As not much is known about neighbourhood sizes for other rearrangement moves, we will only give a rough comparison. We start with tail moves, as the spaces are the same for head moves as for tail moves. Like for head moves, there are obvious upper bounds on the neighbourhood size: a network with $n$ leaves and $k$ reticulations has at most $4n^2+3k^2+8nk+-6n-7k+2$ tail move neighbours. Indeed, this is the number of edges $(u,v)$ where $u$ is a tree node $(2n+k-2)$ multiplied by the total number of edges $(2n+3k-1)$.

Although the bounds for both the head move and the tail move neighbourhoods are quadratic in some sense, we point out that for tail moves, there is quadratic dependence on the number of leaves (a term $n^2$) which is absent in the head move neighbourhood bound. When the reticulation number is small, this implies the head move neighbourhood will likely be smaller than the tail move neighbourhood. Note that we have not proven this quadratic dependence for tail moves, we only conjecture it to be present on the basis of the following arguments. First, the simple upper bound above contains a quadratic term. Secondly, the tail move neighbourhood for trees also has a quadratic dependence on the number of leaves (Corollary 4.2 [28]). One might try to use this last fact to show that the quadratic term is actually necessary, for example by showing that each neighbour of a tree contained in a network is contained in a neighbour of the network. However, this is not true: the network $N$ in Figure 24 contains only the tree $T$, and one of its neighbours $T'$ is not contained in any of the neighbours of $N$.

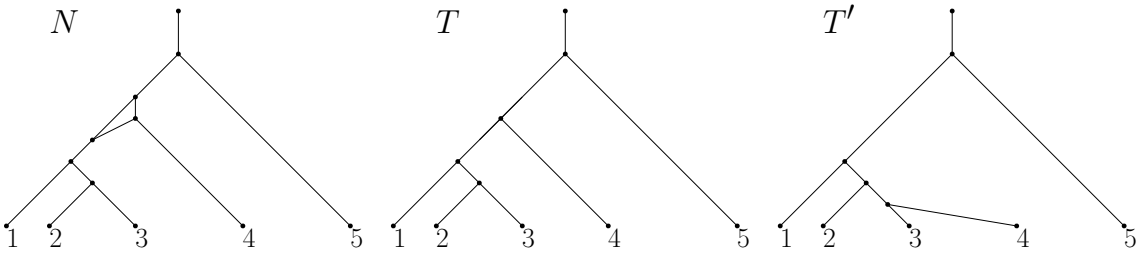



Figure 24: The network $N$ only contains the tree $T$, but none of the tail move neighbours of $N$ contains the tree $T'$, a tail move (rSPR) neighbour of $T$.

A third hint for the quadratic term in the tail move neighbourhood size can be found in the SNPR neighbourhood bounds computed in [19]. The bounds for the largest neighbourhood for a network with $n$ leaves in that paper are quadratic in $n$. However, the comparison is not fair, as this paper considers only tree-child networks—networks with a restricted structure—and it includes vertical moves as well. Lastly, there is a quadratic term in the SPR neighbourhood size for unrooted networks as well ([8] above Proposition 4). Again, a direct comparison is not possible, as not every neighbour of the underlying unrooted network of $N$ may have an orientation which is

a neighbour of $N$.

### 5.2.2 Comparing local neighbourhoods

For local moves, we again start by comparing tail moves and head moves. As before, we can get an upper bound on the distance-1 tail move neighbourhood for a network with $n$ leaves and $k$ reticulations by counting the number of tails at a tree node, and multiplying by the number of edges at distance one. There are $2n+k-2$ edges $(u, v)$ where $u$ is a tree node, and at most 4 edges at distance one from these tails. Hence, an upper bound on the distance-1 tail move neighbourhood is $8n + 4k - 8$. This bound can probably be improved using the approach of Proposition 2 in [9], but it will remain linear in the number of nodes using that technique.

Like for non-local moves, the tail move neighbourhood bound has a strong dependence on the number of leaves, which is absent for head moves. Compared to non-local moves, even less is known about neighbourhoods of local moves. The size of the rNNI neighbourhood is linear in the number of leaves, which indicates that the linear term $n$ for tail moves is necessary, but it does not prove it. Like for tail moves (cf. Figure 24), an rNNI move on an embedded tree of $N$ may not correspond to a distance-1 tail move in $N$ (Figure 25).

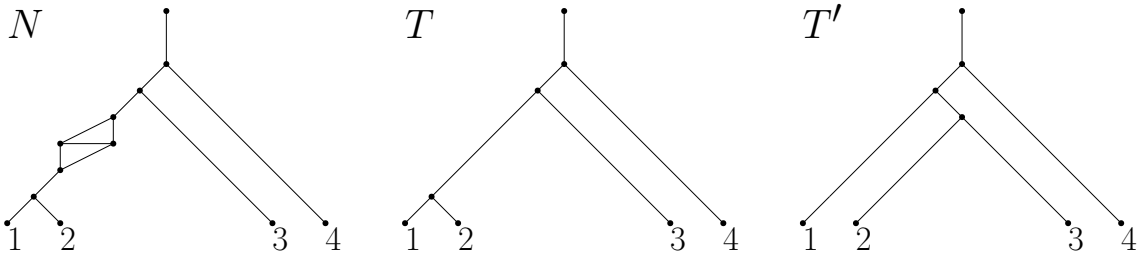



Figure 25: The network $N$ only contains the tree $T$, but none of the rNNI neighbours of $N$ contains the tree $T'$, an rNNI neighbour of $T$.

Although this section contains arguments for bounds on the neighbourhood sizes, the lack of formal proofs is striking. Hence, if anything, this highlights the open questions that remain for all rearrangement moves. We know very little about the neighbourhood sizes of phylogenetic networks. It would be very useful to have bounds or exact sizes to compare different rearrangement moves, and, possibly, to get better bounds on the diameters for spaces defined by these moves.

## 6 Hardness of computing head move distance

In this section, we prove that the problem Head Distance of computing the head move distance between two networks is NP-hard. The proof uses a reduction from Tree rSPR Distance, which is the problem of finding the rSPR distance between two rooted trees. The rough idea is to convert rSPR moves on trees into head moves on specifically constructed networks.

Because rSPR moves change the location of the tail and not the head of an edge, we have to use a trick: we turn the tree upside down, which turns each tail into a head, and hence a tail move into a head move. Just reversing the direction of the edges of the tree is not sufficient, as this gives a graph with multiple roots and one leaf. Hence, we connect all these roots and add a second leaf to create a phylogenetic network. This construction is formalized in the following definitions.

After these definitions, we will show that the minimal number of head moves between two upside down trees is equal to the number of rSPR moves between the two original trees (Lemma 27). For the proof, we show that each sequence of moves between a pair of upside down trees gives an *upside down agreement forest* for these networks (Lemma 26); and each such upside down agreement forest gives a regular agreement forest for the original trees (Lemma 25).

**Definition 19** *Let $T$ be a phylogenetic tree with labels $X = \{x_1, \ldots, x_n\}$, the* upside down version *of $T$ is a network $\mathcal{L}$ with $2n^2 + 2$ leaves ($e_{x,i}$ for $x \in X$ and $i \in [2n]$, $y$, and $\rho$) constructed by:*

1. *Creating the labelled digraph $S$, which is $T$ with all the edges reversed;*
2. *Creating the tree $D$ by taking $C(X \cup \{y\})$ and adding $2n$ pendant edges with leaves labelled $e_{x,1}, \ldots, e_{x,2n}$ to each pendant edge $e = (\cdot, x)$ of $C(X)$;*
3. *Taking the disjoint union of $D$ and $S$;*
4. *identifying the node labelled $x_i$ in $D$ with the node labelled $x_i$ in $S$ and subsequently suppressing this node for all $i$.*

*The* bottom part *of $\mathcal{L}$ is the subgraph of $\mathcal{L}$ below (and including) the parents of the $e_{x,1}$.*

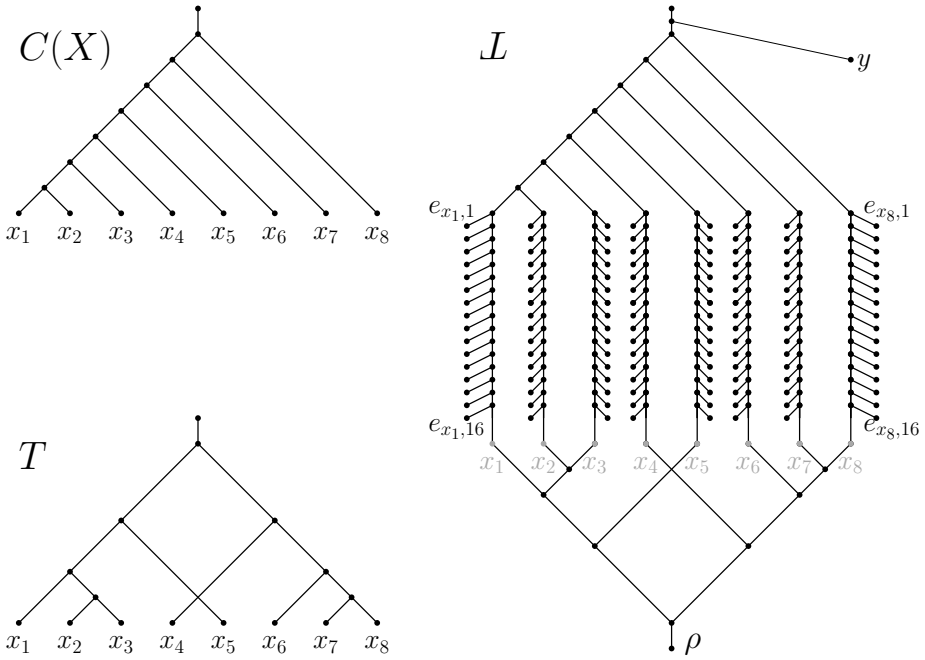



Figure 26: Left the caterpillar $C(X)$ and the tree $T$, right, the upside down version of $T$. In the upside down version of $T$, the original leaves $x_i$ (grey) are suppressed.

The rSPR distance between two trees can be characterized alternatively as the size of an agreement forest [3]. Here, we use this alternative description as part of the reduction. To define agreement forests, we need the following definitions, which we have generalized slightly to work well for networks.

**Definition 20** *Let $T$ be a tree (for digraphs: the underlying undirected graph is a tree) with its degree-1 nodes labelled bijectively with $X$ and let $Y \subseteq X$ be a subset of the labels. Then $T|_Y$ is the* subtree of $T$ induced by $Y$*; that is, it is the union of all shortest (undirected) paths between nodes of $Y$.*

**Definition 21** *Let $G$ and $G'$ be labelled digraphs. Suppose $G$ and $G'$ are labelled isomorphic after suppression of all their* redundant nodes *(indegree-1 outdegree-1 nodes), then we write $G \equiv G'$, or say $G$ is* s-isomorphic *to $G'$ (for suppressed isomorphic).*

*An* embedding *of a graph $H$ in $G$ is an* s-isomorphism *$H \equiv S$ of $H$ with a subgraph $S$ of $G$. We say that $H$* can be embedded *in $G$ if an embedding of $H$ in $G$ exists. Note that any subgraph $H$ of $G$ can be embedded in $G$ as $H \equiv H$.*

Now we look at an important property of embeddings relating to subgraphs, which implies that being embeddable is transitive.

**Lemma 24** *Let $A$, $B$ and $H$ be digraphs with all degree-1 nodes labelled. Suppose $A \equiv B$ and $H$ is a subgraph of $A$, then $H$ can be embedded in $B$.*

**Proof:** The s-isomorphism $A \equiv B$ is an isomorphism of graphs (topological minors) without redundant nodes. This isomorphism is a bijection between the non-redundant nodes of $A$ to the non-redundant nodes of $B$. The map of the edges is a map of paths of $A$ to paths of $B$, where the internal nodes of these paths may only be redundant nodes. Now consider the subgraph $H$ of $A$, and note that the non-redundant nodes of $H$ are non-redundant nodes of $A$ as well. Indeed, the only way to create new non-redundant leaves by taking a subgraph, is to create a leaf from a redundant node, but $L(H) \subseteq L(A) = L(B)$, so each degree-1 node of $H$ corresponds to a degree-1 node of $A$ and of $B$. This means each non-redundant node of $H$ corresponds to a non-redudant node of $B$, and each edge of $H$ to a path between such nodes in $B$, and there is an s-isomorphism of $H$ with the subgraph of $B$ formed by these nodes and edges. □

Now we turn to the definition of an agreement forest, which, as mentioned earlier, characterizes the rSPR distance. Following the definition of the agreement forest, we define a tool similar to an agreement forest tailored to upside down versions of trees. This upside down agreement forest (udAF) can be turned into an agreement forest of the two original trees.

**Definition 22** *Let $T_1$ and $T_2$ be phylogenetic trees with labels $X$ and root $\rho$. Then a partition $\mathcal{P} = \{P_i\}$ of $X \cup \{\rho\}$ is an* agreement forest (AF) *for $T_1$ and $T_2$ if the following hold:*

- *$T_1|_{P_i} \equiv T_2|_{P_i}$ for all $i$;*
- *$T_t|_{P_i}$ and $T_t|_{P_j}$ are node-disjoint for all pairs $i, j$ with $i \neq j$ and fixed $t \in \{1, 2\}$.*

**Definition 23** *Let $\perp\!\!\!L$ be the upside down version of the phylogenetic tree $T$ with label set $X$. Then an* upside down agreement forest (udAF) *for $\perp\!\!\!L$ is a directed graph $F$ such that:*

- *The underlying undirected graph of $F$ is an (undirected) forest;*
- *$F$ is a leaf-labelled graph with label set $\{e_{x,i} : x \in X, i \in [2n]\} \cup \{\rho\}$, where each label appears at most once;*
- *$F \equiv S$ for some subgraph $S$ of the bottom part of $\perp\!\!\!L$.*

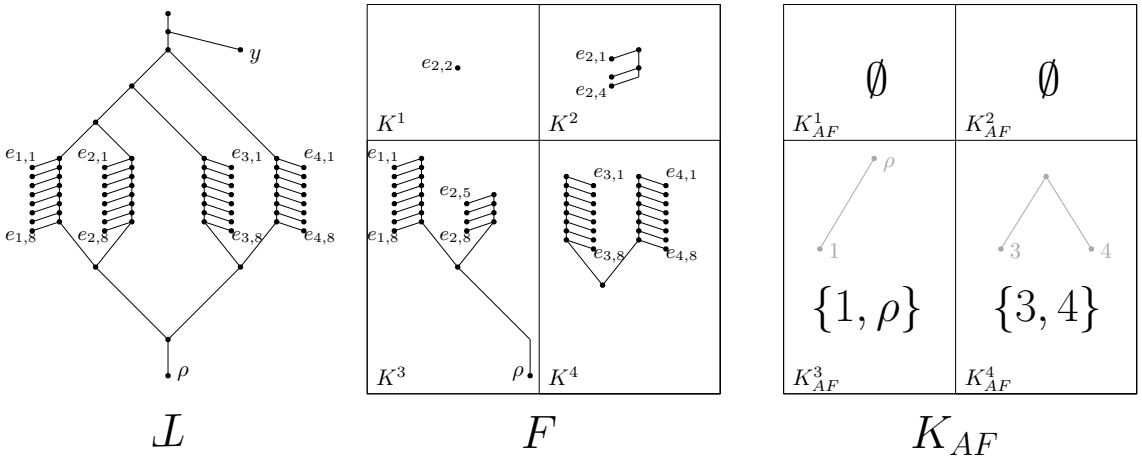



Figure 27: An example of a udAF $F$ of the ud-version of the balanced tree on four leaves $T$. The udAF $F$ consists of four components $K^1, \ldots, K^4$. If $F$ is a udAF for the ud-versions of $T$ and another tree $T'$, then the non-empty $K^i_{AF}$ are parts of a AF for $T$ and $T'$ by Lemma 25.

*Note that the third requirement implies the first (Figure 27).*

**Lemma 25** *Let $T$ and $T'$ be phylogenetic trees with label set $X$. If $F$ is an udAF for $\mathcal{L}$ and for $\mathcal{L}'$, then there exists an AF of $T$ and $T'$ of size at most $|F|$, where $|F|$ denotes the number of components of $F$.*

**Proof:** Let $\mathcal{K}$ be the set of components of $F$. For each $K \in \mathcal{K}$ we define the following part of the agreement forest:

$$K_{AF} := \{x \in X | e_{x,i} \in K \ \forall i \in [1, 2n]\} \cup (K \cap \{\rho\}),$$

where $e_{x,i}$ indicates the $i$-th leaf of $\mathcal{L}$ corresponding to $x$. The agreement forest consists of these parts (ignoring the empty ones, resulting from components that have no complete sets of leaves), together with one part for each leaf that is in none of these parts, i.e.

$$AF := \big\{Y \subseteq X \cup \{\rho\} | \exists K \in \mathcal{K} \text{ s.t. } Y = K_{AF}\big\} \cup \big\{\{x\} \subset X \cup \{\rho\} | \forall K \in \mathcal{K} : x \notin K_{AF}\big\} \setminus \big\{\emptyset\big\}.$$

Note that each component $Y$ of $AF$ corresponds either uniquely to a component $K$ of $F$ which has all $e_{x,i}$ for some leaf $x$, or it corresponds to a leaf $x$ for which not all $e_{x,i}$ are contained in one component of $F$. In the last case, there is a component of $F$ consisting of one leaf $e_{x,i}$ for some $i$. Note that this correspondence $AF \to \mathcal{K}$ must therefore be injective, and $AF$ has size at most $|F|$. What remains to prove is that $AF$ is indeed an agreement forest for $T$ and $T'$.

Let $F'$ be the subgraph of $F$ where each component $K$ is restricted to the subgraph consisting of all paths between the leaves in $K_{AF}$. As (per definition of an udAF) $F$ can be embedded in the bottom part of $\mathcal{L}$, $F'$ can also be embedded in the bottom part of $\mathcal{L}$ as it is a subgraph of $F$ (Lemma 24). This embedding must be unique, because it is of a labelled forest into a labelled tree.

Let $E_x$ be the subgraph of $\mathcal{L}$ induced by the leaves $e_{x,i}$ and their parents for all $i$ and a fixed $x$. Now replace each subgraph $E_x$ with one leaf $x$ in both $F'$ and in the bottom part of $\mathcal{L}$. Let the

resulting graphs be $F^s$ and $B^s$. Subsequently reverse the direction of each edge in both $B^s$ and in $F^s$ with resulting graphs $B^r$ and $F^r$. Note that the resulting graphs are $B^r = T$ and the union $F^r = \cup_{K \in \mathcal{K}} T|_{K_{AF}}$, and all the restricted trees $T|_{K_{AF}}$ are node disjoint.

We repeat this argument for $\lrcorner'$, and note that the modifications from $F$ to $F^r$ are independent of $\lrcorner$, so we have the equality $F^r = \cup_{K \in \mathcal{K}} T'|_{K_{AF}}$, where the parts $T'|_{K_{AF}}$ are again node disjoint. This means $T|_{K_{AF}} \equiv T'|_{K_{AF}}$ for each $K \in AF$ corresponding to a non-trivial component of $F$, and $T|_{P_i}$ and $T|_{P_j}$ are node disjoint for all nontrivial parts $P_i$ and $P_j$ of $AF$ (similarly for $T'$). Hence so far the elements of $AF$ corresponding to non-trivial components of $F$, meet all the requirements of an AF.

The only other elements of $AF$ contain only one label, each of which is not in any of the non-trivial components of $AF$. Hence, for any such label $x$, the restriction $T|_{\{x\}}$ consists of only the node labelled $x$, which is not contained in any other component by definition (and similarly for $T'$). Furthermore, the s-isomorphism $T|_{\{x\}} \equiv T'|_{\{x\}}$ is trivial. Hence, $AF$ is indeed an agreement forest. □

The preceding lemma shows that an udAF for two upside down trees gives an AF for the original trees of the same size. We still lack a connection between the number of head moves and an udAF, however. The following lemma shows that appropriate head move sequences correspond to udAFs of size related to the number of head moves.

**Lemma 26** *Let $T$ and $T'$ be trees with label set $X$, and $|X| = n$. Suppose $S$ is a sequence of head moves $\lrcorner = N_0, \ldots, N_{|S|} = \lrcorner'$ of length $|S| < 2n$. Then there is an udAF $F$ of $\lrcorner$ and $\lrcorner'$ with at most $|S| + 1$ components.*

**Proof:** Let $B$ be the bottom part of $\lrcorner$. We prove this result using induction on the number of moves to prove that there exist subgraphs $F_i$ of $N_i$ which can be embedded in the bottom part of $\lrcorner$ and have $|F_i| \leq i$ components. Finally we prove the subgraph $F_{|S|}$ of $N_{|S|} = \lrcorner'$ must actually be a subgraph of the bottom part of $\lrcorner'$.

As a base of the induction, set $F_0 = B$, which is connected and can clearly be embedded in itself and is a subgraph of $\lrcorner$.

Now suppose we have subgraphs $F_i$ of $N_i$ with embeddings of $F_i$ in $B$ and $|F_i| \leq i$ for all $i < j \leq |S|$. We prove that there also exists a subgraph $F_j$ of $N_j$ with at most $j$ components that can be embedded in $B$.

Note that $F_{j-1}$ is a subgraph of $N_{j-1}$ and therefore the moving edge $e_j = (u, v)$ can be either an edge of $F_{j-1}$, or it is in the complement $N_{j-1} \setminus F_{j-1}$. In the last case $e_j$ can have only its endpoints in $F_{j-1}$. Now construct $F_j$ as follows:

- remove edge $e_j = (u, v)$ from $F_{j-1}$ if it was contained in it;
- clean up the resulting graph by removing all edges not contained in any undirected path between two leaves, and suppressing $v$ if it is a degree 2 vertex after removal of $(u, v)$.
- add the new endpoint if necessary. That is: let the target edge of the move be $t$, if $t$ is contained in the graph after cleaning up, subdivide $t$.

Note that $F_j$ can be embedded in $F_{j-1}$ because the only operations were: restriction to a subset of labels, subdivision, and suppression (Lemma 24).

Because $F_{j-1}$ embeds in $B$, there is also an embedding of $F_j$ into $B$. Furthermore, $F_j$ is a subgraph of $N_j$ by construction: the three steps correspond exactly to the three steps of a head move in $N_{j-1}$. Lastly, $F_j$ has at most one more component than $F_{j-1}$, because the only operation

that can increase the number of components is the removal of the edge in the first step, and because that is an edge removal in a graph, it creates at most one extra component.

We conclude that the desired subgraphs $F_i$ of $N_i$ exist for all $i \in [|S|]$.

Note that we have not yet proven that $F := F_{|S|}$ is an udAF for $\mathcal{L}'$, as $F$ might not embed in the bottom part of $\mathcal{L}'$. We now prove that $F$ is in fact a subgraph of the bottom part of $\mathcal{L}'$.

By construction, $F$ is a directed subgraph of $\mathcal{L}'$. Suppose (for a contradiction) that $F$ is not a subgraph of the bottom part of $\mathcal{L}'$, i.e., some part of $F$ lies in the top part of $\mathcal{L}'$. This means that there is a node $t$ of $F$ that corresponds to a tree node (which we also call $t$) in the upper part of $\mathcal{L}'$. A tree node of $F$ necessarily has two children $c_1$ and $c_2$, as $F$ embeds in the bottom part of $\mathcal{L}$. One of these children (w.l.o.g. $c_1$) must have a unique leaf descendant $e_{x,i}$. The other child ($c_2$) either has a leaf descendant $e_{x,j}$—with the same $x$ as the descendant of $c_1$—or the next non-redundant descendant is a reticulation node.

If $c_2$ has a leaf descendant $e_{x,j}$, we note the following: $t$ is mapped to a tree node in the top part of $\mathcal{L}'$. Hence the leaves below the one child of $t$ and the leaves below the other child of $t$ can never correspond to the same $x \in X$: indeed if $e_{y,i}$ is below $c_1$, then $e_{y,j}$ is also below $c_1$, and similarly for $c_2$; furthermore, as the only reticulations of $\mathcal{L}'$ are in the lower part of the network after the $e_{\cdot,\cdot}$ split off, the leaves below $c_1$ and $c_2$ are disjoint (except for the leaf corresponding to the root of $T'$). Hence, as $c_1$ has a descendant $e_{x,i}$, and $c_2$ has a descendant $e_{x,j}$, we have a contradiction.

Now if $c_2$'s first non-redundant descendant $d$ is a reticulation node, then this node maps to a reticulation in the bottom part of $\mathcal{L}'$. This means the edge $(t, d)$ maps to a path from the top part of $\mathcal{L}'$ to a reticulation in the bottom part of $\mathcal{L}'$. Such a path must necessarily contain all parents of the leaves $e_{x,i}$ for some $x \in X$. As the embeddings of all components in $F$ are node disjoint, and each leaf $e_{\cdot,\cdot}$ is a node of $F$, each leaf $e_{x,i}$ (fixed $x$, for all $i \in [2n]$) is its own component in $F$. Hence $F$ has at least $2n + 1$ components, implying that $|S| \geq 2n$, which gives us a contradiction with the assumptions of the lemma.

Hence there does exist an embedding of $F$ in the bottom part of $\mathcal{L}$, and $F$ is an udAF for $\mathcal{L}'$ with at most $|S| + 1$ components, as $F$ has at most $|S| + 1$ components. □

Finally we put everything together in the following lemma and theorem: each candidate head move sequence defines an udAF, which in turn gives an AF for the original trees, which bounds the rSPR distance between these trees.

**Lemma 27** *Let $T_1$ and $T_2$ be trees with a common label set, then*

$$d_{rSPR}(T_1, T_2) = d_{Head}(\mathcal{L}_1, \mathcal{L}_2).$$

**Proof:** The inequality $d_{\mathrm{rSPR}}(T_1, T_2) \geq d_{\mathrm{Head}}(\mathcal{L}_1, \mathcal{L}_2)$ is obvious, as the rSPR sequence for the trees directly translates into a head move sequence for the upside down trees.

We now prove the other inequality. As $2n > d_{\mathrm{rSPR}}(T_1, T_2) \geq d_{\mathrm{Head}}(\mathcal{L}_1, \mathcal{L}_2)$ we only have to consider head move sequences of length at most $2n$. Suppose we have a sequence of head moves $S$ between $\mathcal{L}_1$ and $\mathcal{L}_2$ of length at most $2n$, then there exists a udAF of size at most $|S| + 1$ for $\mathcal{L}_1$ and $\mathcal{L}_2$ (Lemma 26). Now Lemma 25 tells us that there is an AF for $T$ and $T'$ of size at most $|S| + 1$. Using the fact that the size of the MAF of $T$ and $T'$ minus one is equal to the rSPR distance between $T$ and $T'$ [3], we get the following inequalities:

$$|S| \geq |AF| - 1 \geq d_{\mathrm{rSPR}}(T_1, T_2).$$

We conclude that $d_{\mathrm{rSPR}}(T_1, T_2) = d_{\mathrm{Head}}(\mathrm{L}_1, \mathrm{L}_2)$. □

**Theorem 5** *Computing the head move distance between two networks is NP-hard.*

**Proof:** Direct consequence of the previous lemma, as computing the rSPR distance between two trees is NP-hard [3]. □

Note that the theorem above does not tell us whether it is hard to find the distance between networks of a fixed tier; increasing the size of the input corresponds to increasing the reticulation number in our construction.

# 7 Discussion

When generalizing rSPR moves on rooted trees to rooted networks, it is natural to consider tail moves, because each rSPR move in a tree is a tail move. However, when taking the view that an rSPR move is a move that changes one of the endpoints of an edge, head moves also belong to the generalization of rSPR moves [9]. In this view, it is equally natural to only consider head moves, as to only consider tail moves.

We have showed that head moves are sufficient to connect all tiers of phylogenetic network space except tier-0. This might be surprising because head moves are relatively limited compared to tail moves: the head move neighbourhood is small compared to the tail move neighbourhood. On the other hand, when one reverses all the edges of a network, each tail move becomes a head move. This makes the difference between connectivity results for these types of moves just a mathematical difference in numbers of roots, reticulations, and leaves, instead of a fundamental difference in biological interpretation.

To unify these connectivity results, one could consider head or tail moves in a broader class of networks, which may have multiple roots and at least one leaf (instead of at least two) (Figure 28). For such multi-rooted networks, connectivity results for head moves and for tail moves could easily be related. This reason for studying multi-rooted networks is mathematically inspired.

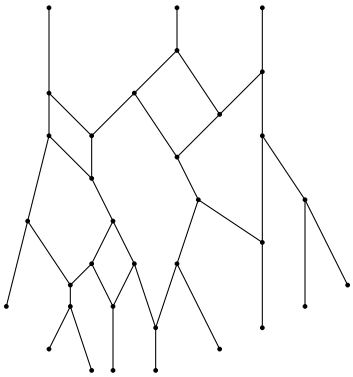


Figure 28: A multi-rooted network, labels of leaves have been omitted.

Another reason to study multi-rooted networks is inspired by biology: these networks could be interesting on their own as subnetworks of ordinary phylogenetic networks. The advantage is that

one does not have to make assumptions about how these roots are connected higher up, that is, about the evolutionary history before the existence of these root genes or species [12]. Additionally, a famous but slightly dated view of the evolutionary history is the net of life by Doolittle, which features multiple roots [5]. A third reason becomes apparent when we take a broader view of phylogenetic networks that includes pedigrees: these often start with multiple individuals that may coalesce in the distant past.

While we focussed mostly on head and tail moves of any distance, we have proven the connectivity of tiers of phylogenetic network space by distance-2 head moves. Distance-1 head moves are not sufficient in general because heads cannot move past their own tails. It would be interesting to see which networks are actually connected by distance-1 head moves.

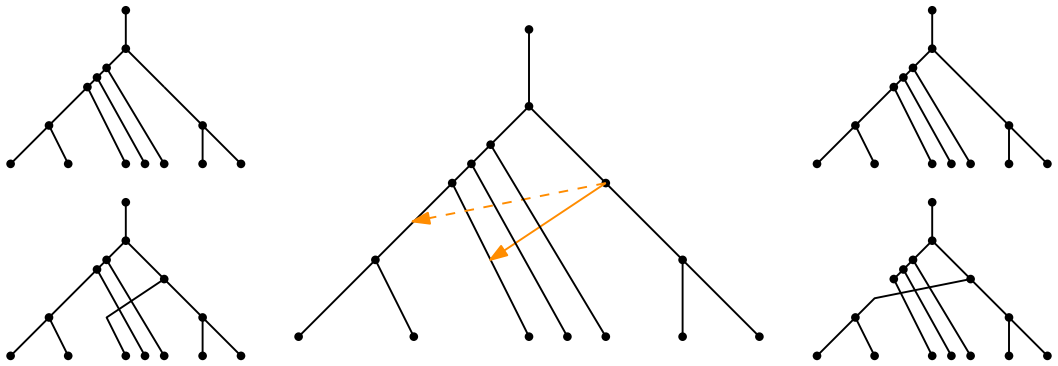


Figure 29: A distance-2 head move in a network and the displayed trees before (left) and after (right) the head move. The top displayed tree is the same before and after the head move. The bottom tree before disappears, and is replaced by the tree bottom-right, which is a distance-3 tail move away from the top tree.

It is unclear if this connectivity result for distance-2 moves is useful, especially in the context of embedded gene trees their relevance may be disputed. After all, local head moves do not generally correspond to local moves in the displayed trees (Figure 29). Problems studying the relation between displayed trees, which are interpreted as gene trees, and phylogenetic networks are often quite hard [4, 16, 15]. Hence, strategies for solving these problems could benefit from local search heuristics.

As mentioned, an important motivation for studying rearrangement moves is their possible use in local search strategies for phylogenetic networks. As such, it is important to understand the topological and geometric properties of the tiers of phylogenetic networks space. In this paper, we started this study by giving bounds on the head move diameters, and finding additional connections between head move distance, tail move distance, and rSPR distance.

Although the bounds for the head move diameter we found are already quite good—both upper bound, and lower bound are linear in the number of leaves and the reticulation number, just like for tail moves and for rSPR moves [17]—they could possibly be improved.

As future research, one could try to discover the exact diameters. Another direction would be to try to apply our techniques for bounding the diameter to other types of moves, such as SNPR

and PR moves [2, 20]. Because these classes of moves also include vertical moves, this might be quite challenging.

The other property of phylogenetic network space defined by head moves we touched on was the neighbourhood size. The head move neighbourhood is relatively small. Nevertheless, it is still possible to reach any network quite quickly, as the diameter still grows linearly. This means head moves might be very well suited for local search heuristics.

Of course, there are other factors to consider. Head moves might not have the proper relation to the studied phylogenetic objectives. For example, they could give irregular optimization landscapes. For example, in phylogenetic tree space, NNI moves give local optima (not globally optimal) for maximum parsimony, whereas SPR moves only give a global optimum for perfect sequence data [29]. It would be interesting to analyse such relations for networks, too. For instance, by studying the occurrence of local optima for different kinds of parsimony [7, 14] using the existing types of rearrangement moves.

Another possible complicating factor in the relation between head moves and the optimization objective could be that head moves might be too restrictive for some types of networks. Indeed, we have not studied head moves for subclasses of networks. It might be useful to see if head moves also connect tiers of tree-child networks for example. Such questions have been answered for other moves [2].

Lastly, in this paper we have studied the problem of computing the head move distance between two networks. For tail moves, rSPR moves [17] and for SNPR moves [21], it was already known that computing the distance between two networks is NP-hard. For the first two of these, we additionally know that computation of distances is hard for each tier. Here, we have shown that computing head move distance is also NP-hard, although we have not shown this for each tier separately. A first step in proving hardness in each tier might be to study head move distance computation in tier-1.

It could also be interesting to find an efficient algorithm for the task of finding a shortest head move sequence, or to characterize the exact distance between two phylogenetic networks in a more abstract way. No efficient (FPT) algorithm for this task is known, nor are there any exact characterizations of distances between networks given by rearrangement moves. A first attempt was recently made using a generalization of agreement forests, this approach currently only yields exact distances between trees and networks, and no exact distances between two networks [20, 21].

## Acknowledgements

The author would like to thank Leo van Iersel, Yukihiro Murakami, and Mark Jones for their valuable input, and especially Leo van Iersel and Yukihiro Murakami for reading and commenting on the manuscript.

## References

[1] H. Atas, N. Tuncbag, and T. Doğan. Phylogenetic and other conservation-based approaches to predict protein functional sites. In *Computational Drug Discovery and Design*, pages 51–69. Springer, 2018. doi:10.1007/978-1-4939-7756-7_4.

[2] M. Bordewich, S. Linz, and C. Semple. Lost in space? Generalising subtree prune and regraft to spaces of phylogenetic networks. *Journal of theoretical biology*, 423:1–12, 2017. doi:10.1016/j.jtbi.2017.03.032.

[3] M. Bordewich and C. Semple. On the computational complexity of the rooted subtree prune and regraft distance. *Annals of combinatorics*, 8(4):409–423, 2005. doi:10.1007/s00026-004-0229-z.

[4] M. Bordewich and C. Semple. Computing the minimum number of hybridization events for a consistent evolutionary history. *Discrete Applied Mathematics*, 155(8):914–928, 2007. doi:10.1016/j.dam.2006.08.008.

[5] W. Doolittle. Phylogenetic classification and the universal tree. *Science*, 284(5423):2124–2128, 1999.

[6] J. Felsenstein. *Inferring Phylogenies*, volume 2. Sinauer associates Sunderland, MA, 2004.

[7] M. Fischer, L. van Iersel, S. Kelk, and C. Scornavacca. On computing the maximum parsimony score of a phylogenetic network. *SIAM Journal on Discrete Mathematics*, 29(1):559–585, 2015. doi:10.1137/140959948.

[8] A. Francis, K. T. Huber, V. Moulton, and T. Wu. Bounds for phylogenetic network space metrics. *Journal of mathematical biology*, 76(5):1229–1248, 2018. doi:10.1007/s00285-017-1171-0.

[9] P. Gambette, L. van Iersel, M. Jones, M. Lafond, F. Pardi, and C. Scornavacca. Rearrangement moves on rooted phylogenetic networks. *PLoS computational biology*, 13(8):e1005611, 2017. doi:10.1371/journal.pcbi.1005611.

[10] F. Gao, E. Bailes, D. L. Robertson, Y. Chen, C. M. Rodenburg, S. F. Michael, L. B. Cummins, L. O. Arthur, M. Peeters, G. M. Shaw, et al. Origin of HIV-1 in the chimpanzee pan troglodytes troglodytes. *Nature*, 397(6718):436, 1999. doi:10.1038/17130.

[11] C. Guyeux, B. Al-Nuaimi, B. AlKindy, J.-F. Couchot, and M. Salomon. On the ability to reconstruct ancestral genomes from mycobacterium genus. In *International Conference on Bioinformatics and Biomedical Engineering*, pages 642–658. Springer, 2017. doi:10.1007/978-3-319-56148-6_57.

[12] L. S. Haggerty, P.-A. Jachiet, W. P. Hanage, D. A. Fitzpatrick, P. Lopez, M. J. O'Connell, D. Pisani, M. Wilkinson, E. Bapteste, and J. O. McInerney. A pluralistic account of homology: adapting the models to the data. *Molecular biology and evolution*, 31(3):501–516, 2013. doi:10.1093/molbev/mst228.

[13] K. T. Huber, V. Moulton, and T. Wu. Transforming phylogenetic networks: Moving beyond tree space. *Journal of theoretical biology*, 404:30–39, 2016. doi:10.1016/j.jtbi.2016.05.030.

[14] L. van Iersel, M. Jones, and C. Scornavacca. Improved maximum parsimony models for phylogenetic networks. *Systematic biology*, 67(3):518–542, 2017. doi:10.1093/sysbio/syx094.

[15] L. van Iersel, S. Kelk, N. Lekic, C. Whidden, and N. Zeh. Hybridization number on three rooted binary trees is EPT. *SIAM Journal on Discrete Mathematics*, 30(3):1607–1631, 2016. doi:10.1137/15M1036579.

[16] L. van Iersel and S. Linz. A quadratic kernel for computing the hybridization number of multiple trees. *Information Processing Letters*, 113(9):318–323, 2013. doi:10.1016/j.ipl.2013.02.010.

[17] R. Janssen, M. Jones, P. L. Erdős, L. van Iersel, and C. Scornavacca. Exploring the tiers of rooted phylogenetic network space using tail moves. *Bulletin of mathematical biology*, 80(8):2177–2208, 2018. doi:10.1007/s11538-018-0452-0.

[18] J. B. Joy, R. H. Liang, R. M. McCloskey, T. Nguyen, and A. F. Poon. Ancestral reconstruction. *PLoS computational biology*, 12(7), 2016. doi:10.1371/journal.pcbi.1004763.

[19] J. Klawitter. The SNPR neighbourhood of tree-child networks. *Journal of Graph Algorithms and Applications*, 22(2):329–355, 2018. doi:10.7155/jgaa.00472.

[20] J. Klawitter. The agreement distance of rooted phylogenetic networks. *Discrete Mathematics and Theoretical Computer Science*, 21(3):10–10, 2019. doi:10.23638/DMTCS-21-3-19.

[21] J. Klawitter and S. Linz. On the subnet prune and regraft distance. *The Electronic Journal of Combinatorics*, 26(2):1–23, 2019. doi:10.37236/7860.

[22] C. Lakner, P. Van Der Mark, J. P. Huelsenbeck, B. Larget, and F. Ronquist. Efficiency of markov chain monte carlo tree proposals in bayesian phylogenetics. *Systematic biology*, 57(1):86–103, 2008. doi:10.1080/10635150801886156.

[23] J. Lessler, L. H. Chaisson, L. M. Kucirka, Q. Bi, K. Grantz, H. Salje, A. C. Carcelen, C. T. Ott, J. S. Sheffield, N. M. Ferguson, et al. Assessing the global threat from zika virus. *Science*, 353(6300), 2016. doi:10.1126/science.aaf8160.

[24] L.-T. Nguyen, H. A. Schmidt, A. von Haeseler, and B. Q. Minh. Iq-tree: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Molecular biology and evolution*, 32(1):268–274, 2014. doi:10.1093/molbev/msu300.

[25] S. Roch. A short proof that phylogenetic tree reconstruction by maximum likelihood is hard. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(1):92–94, 2006. doi:10.1109/TCBB.2006.4.

[26] C. Semple and M. A. Steel. *Phylogenetics*, volume 24. Oxford University Press on Demand, 2003.

[27] I. Shindyalov, N. Kolchanov, and C. Sander. Can three-dimensional contacts in protein structures be predicted by analysis of correlated mutations? *Protein Engineering, Design and Selection*, 7(3):349–358, 1994. doi:10.1093/protein/7.3.349.

[28] Y. S. Song. On the combinatorics of rooted binary phylogenetic trees. *Annals of Combinatorics*, 7(3):365–379, 2003. doi:10.1007/s00026-003-0192-0.

[29] E. Urheim, E. Ford, and K. St. John. Characterizing local optima for maximum parsimony. *Bulletin of mathematical biology*, 78(5):1058–1075, 2016. doi:10.1007/s11538-016-0174-0.

[30] L. Van Iersel, R. Janssen, M. Jones, Y. Murakami, and N. Zeh. Polynomial-time algorithms for phylogenetic inference problems involving duplication and reticulation. *IEEE/ACM transactions on computational biology and bioinformatics*, 17(1):14–26, 2019. doi:10.1109/TCBB.2019.2934957.

[31] Y. Yu, A. J. Harris, C. Blair, and X. He. Rasp (reconstruct ancestral state in phylogenies): a tool for historical biogeography. *Molecular phylogenetics and evolution*, 87:46–49, 2015. doi:10.1016/j.ympev.2015.03.008.