Eindhoven University of Technology

BACHELOR

A New Method to Measure Tree-basedness of Rooted Binary Phylogenetic Networks

Bongaerts, Niek M.

Award date:
2023

UNIVERSITY OF TECHNOLOGY EINDHOVEN

BACHELOR FINAL PROJECT

COMBINATORIAL OPTIMIZATION

A New Method to Measure Tree-basedness of Rooted Binary Phylogenetic Networks

Name
Niek BONGAERTS

Student number
1564854

Supervisors
dr. Martin FROHN
dr. Judith KEIJSPER

2023

Abstract

This bachelor thesis explores the possibility of a new proximity measure for tree-basedness of rooted binary phylogenetic networks. The main research question addressed in this study is how to make such proximity measure using rooted Nearest Neighbour Interchange (rNNI) moves. Moreover, we show interesting upper bounds for such proximity measure, we construct a method that transforms an arbitrary rooted binary phylogenetic network into a tree-based network using rNNI moves and we discuss the applicability of this field's research in biology.

To this end, a comprehensive literature review was conducted to gather existing knowledge on the topic of rooted binary phylogenetic networks, their characterisations for tree-basedness, already existing proximity measures and rNNI moves.

The methodology employed in this study involved analysing the rNNI induced metric on the set of rooted binary phylogenetic networks, how to formally define a proximity measure, proving that an rNNI proximity measure is well-defined, providing an upper bound for this proximity measure using the existing knowledge on characterisations for tree-basedness and proximity measures, and giving a short review on the applicability in biology.

The results of this thesis revealed that an rNNI-based proximity measure is indeed well-defined. Furthermore, we show insights on connections to other proximity measures; an upper bound is proved related to the proximity measure p, as introduced in A. Francis et al. (2018). Furthermore, as the scope of the literature review in this thesis is broad, the thesis can be used as a summary for the existing knowledge on rooted binary phylogenetic networks. More so with a focus on rNNI, properties for tree-basedness, proximity measures for tree-basedness, and biological applicabilities. The results of the thesis also allow for many possibilities in future research, such as extending the results to the non-binary case, extending the results to the unrooted case, or researching whether the upper bound related to p is tighter than proved in this thesis.

Contents

1 Introduction **3**
1.1 Mathematical preliminaries . . . 3
1.1.1 Phylogenetic networks . . . 4
1.1.2 Subtree Transfer Operations and their induced metric . . . 5

2 On the combinatorial properties of tree-based phylogenetic networks **11**
2.1 Matching characterisation . . . 11
2.2 Antichain characterisation . . . 12
2.3 Other characterisations . . . 17

3 Proximity measures for tree-basedness **21**
3.1 Definition of a proximity measure . . . 21
3.2 Review of known proximity measures . . . 21
3.2.1 Definitions of the known proximity measures . . . 21
3.2.2 Complexity of the known proximity measures . . . 24

4 A new proximity measure for tree-basedness **27**
4.1 The rNNI proximity measure . . . 27
4.2 An upper bound related to proximity measure p . . . 31
4.2.1 Edge flips . . . 32
4.2.2 A conjecture of a strong upper bound . . . 33
4.2.3 The proof of a weaker upper bound . . . 34
4.2.4 A first illustration of the upper bound construction . . . 36
4.2.5 Further illustrations of the upper bound construction . . . 38
4.3 Complexity of the rNNI proximity measure . . . 42

5 Significance **43**
5.1 Origins from biology . . . 43
5.2 Classes of phylogenetic networks . . . 43
5.3 Exploring the network space . . . 44
5.4 Connection to this thesis . . . 45

6 Conclusion **46**
6.1 Findings in this thesis . . . 46
6.2 Topics of future research . . . 46

1 Introduction

A rooted phylogenetic network is a graphical representation of evolutionary relationships among species, also called *taxa*. It is an extension of the traditional phylogenetic tree, which assumes that every evolutionary event is a speciation event. In contrast, a rooted phylogenetic network assumes another type of evolutionary event, namely reticulation events. Hybridization and lateral gene transfers are such reticulation events and are common in the evolutionary histories of a number of species. Moreover, reticulation events may express uncertainty in a species' evolutionary history. This way, rooted phylogenetic networks provide a more flexible and realistic representation of evolutionary history. These graphs are commonly used in evolutionary biology and bio-informatics. Hence studying the mathematical properties of such graphs might aid in these fields' further research.

Networks with only speciation events are trees. In contrast, trees containing reticulation events lead to the formation of taxa with multiple ancestrial lineages. This makes a representation of such evolutionary processes less clear. We call a rooted phylogenetic network tree-based if we can remove edges from the network such that every leaf has one single path from root to leaf. This means, in a tree-based network the underlying evolutionary process is inherently tree-like with reticulation events seen as uncertainty about the true process. The class of rooted tree-based phylogenetic networks has first been fully characterised for binary networks by Zhang (2016). Subsequently, Pons et al. (2019) gave a full characterisation of general rooted tree-based phylogenetic networks. As it turns out, some evolutionary processes are not fundamentally tree-like in nature, like lateral gene transfer among prokaryotes Dagan & Martin (2006). For biologists it may be significant to know whether a phylogenetic network is tree-based or not; tree-basedness of rooted phylogenetic processes characterise the importance of reticulation events or speciation events in an evolutionary process. Since importance of reticulation events or speciation events in a phylogenetic network is not dichotomous, it is also interesting to mathematically describe how close a given rooted phylogenetic network is to being tree-based. To quantify how close a phylogenetic network is to being tree-based, several authors introduced proximity measures. A. Francis et al. (2018) described five different proximity measures for rooted binary phylogenetic networks, of which they proved the first three were equivalent and the last two were left for further research. Fischer & Francis (2020) introduced several proximity measures for unrooted phylogenetic networks. In particular, the proximity measure based on the Nearest Neighbour Interchanges (NNI) moves in this paper was deemed interesting for further research. Fischer & Francis (2020) also mentioned some issues in lifting proximity measures for unrooted phylogenetic networks to the rooted case. Gambette et al. (2017) generalized NNI moves to rooted binary phylogenetic networks, these are rNNI moves. In this thesis we explore a proximity measure for rooted binary phylogenetic networks based on the rNNI operator, in the same fashion as Fischer & Francis (2020) did with the NNI moves for unrooted phylogenetic networks.

In the first part of the thesis we will introduce some preliminaries on rooted phylogenetic networks and their mathematical properties. Here we also explain the both the NNI and rNNI operator and relevant properties that have been proven for them. After which we will review the different ways in which tree-basedness has been characterised. This gives us the tools to introduce the proximity measures defined in A. Francis et al. (2018) and show how their equivalence was proven. Thereafter we will consider the proximity measure for rooted binary phylogenetic networks based on rNNI moves and show that it is well-defined. For this new proximity measure we prove an upper bound with respect to another proximity measure. Using the content of this proof, we also obtain a method to transform any non-tree-based rooted binary phylogenetic network into a tree-based one, while using only rNNI-moves in the process. Lastly we will reflect and give a perspective on the utility of tree-based phylogenetic networks in biology.

1.1 Mathematical preliminaries

Here we discuss some mathematical preliminaries for the thesis. We first formally define phylogenetic networks and the property of tree-basedness. After that we will discuss the moves

Nearest Neighbour Interchange and rooted Nearest Neighbour Interchange, and show how the latter induces a metric for the set of rooted binary phylogenetic networks.

1.1.1 Phylogenetic networks

We begin with the definition of a *rooted phylogenetic network*:

Definition 1. *A* rooted phylogenetic network $\mathcal{N} = (V, E)$ *on a non-empty finite set* $X \subseteq V$ *is an acyclic weakly connected simple directed graph which contains the following types of vertices:*

- *a single vertex with indegree 0 and outdegree 2, called the root;*
- *vertices with indegree 1 and outdegree 0, called leaves. Every leaf is labelled with an element of X;*

and may contain the following types of vertices:

- *vertices with indegree greater than or equal to 2 and outdegree 1, called reticulation vertices;*
- *vertices with indegree 1 and outdegree greater than or equal to 2, called tree vertices.*

If the vertices of $\mathcal{N}$ *contain only vertices of summed in- and outdegree at most three, then we call* $\mathcal{N}$ binary. *If* $\mathcal{N}$ *contains no reticulation vertices, we call* $\mathcal{N}$ *a* phylogenetic tree.

We call X the *leaf set* or the set of *taxa*, edges directed into reticulation vertices are called *reticulation edges* and all other edges are called *tree edges*.
Rooted binary phylogenetic networks are generally easier to characterise and were first characterised by Zhang (2016). In this thesis, rooted binary phylogenetic networks will be our primary subject of study.

For the following definition we need to refresh our knowledge on some notions from graph theory. Recall that a *tree* is a connected acyclic directed graph. A *spanning tree* of a directed graph $G = (V, E)$ is a tree $T = (V, E')$, where $E' \subseteq E$.

Definition 2. *A phylogenetic network* $\mathcal{N} = (V, E)$ *on leaf set* $X \subseteq V$ *is called* tree-based *if and only if there exists a spanning tree* $T = (V, E')$ *with* $E' \subseteq E$ *on the same leaf set* X*. If* $\mathcal{N}$ *is tree-based, we call such* T *a* base tree *of* $\mathcal{N}$*.*

To illustrate Definition 2, consider Figure 1.

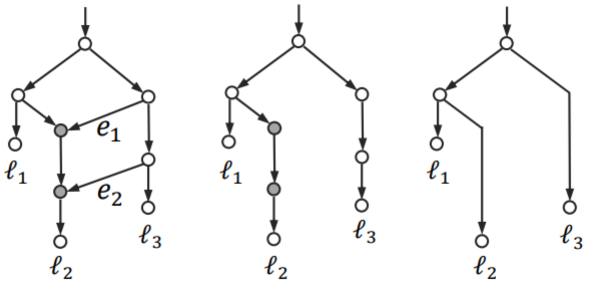

Figure 1: (Zhang, 2016) (left) A tree-based phylogenetic network. (middle) A base tree of the left phylogenetic network. The base tree is a subtree of the network that can be obtained by the removal of the edges e_1 and e_2. (right) A phylogenetic tree can be obtained by removing the reticulation vertices from the base tree and connecting vertices that were initially connected via a path consisting of only reticulation vertices (except for the endpoints of the path). Reticulation nodes in the network are represented by shaded circles.

Furthermore, to discuss Nearest Neighbour Interchanges in the next subsection we need the following definition on *unrooted phylogenetic networks*:

Definition 3. *An* unrooted phylogenetic network $\mathcal{N} = (V, E)$ *on a non-empty finite set* $X \subseteq V$ *is a connected simple undirected graph where vertices in* X *have degree 1. If the vertices of* $\mathcal{N}$ *contains only vertices of degree 1 or 3, then we call* $\mathcal{N}$ binary*.*

An example of an unrooted phylogenetic network can be seen in Figure 2.

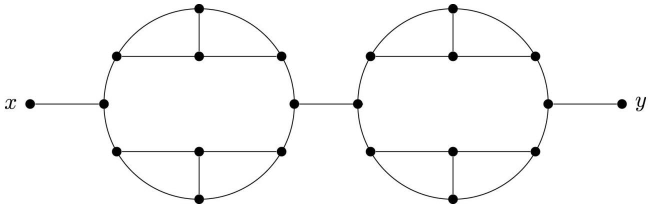

Figure 2: (Fischer & Francis, 2020) An unrooted phylogenetic network on leaf set $X = \{x, y\}$.

An edge of a phylogenetic tree incident with a leaf is a *pendant edge*, any other edge is called an *internal edge*.

1.1.2 Subtree Transfer Operations and their induced metric

Both rooted and unrooted binary phylogenetic networks can have edges to which we can apply Subtree Transfer Operations. Such operations take a sub-tree of a network and swaps it with another sub-tree. We discuss the Subtree Transfer Operations Nearest Neighbour Interchange and rooted Nearest Neighbour Interchange.

Definition 4. *Any internal edge of a unrooted binary phylogenetic tree has four subtrees attached to it. A* nearest neighbour interchange (NNI) *is an operation that takes one subtree on one side of an internal edge and swaps it with a subtree on the other side of the edge, as illustrated in Figure 3.*

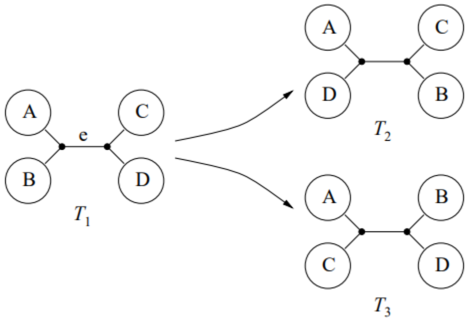

Figure 3: an NNI move on an edge of T_1, producing either T_2 or T_3

We can generalize NNI to rooted binary phylogenetic networks, which yields the rooted Nearest Neighbour Interchange (rNNI) operation:

Definition 5. *Any internal edge of a rooted binary phylogenetic network has four subtrees attached to it. A* rooted Nearest Neighbour Interchange (rNNI) *is an operation that takes one subtree on one side of an internal edge and swaps it with a subtree on the other side of the edge and may or may not change the direction of the internal edge, as illustrated in Figure 4.*

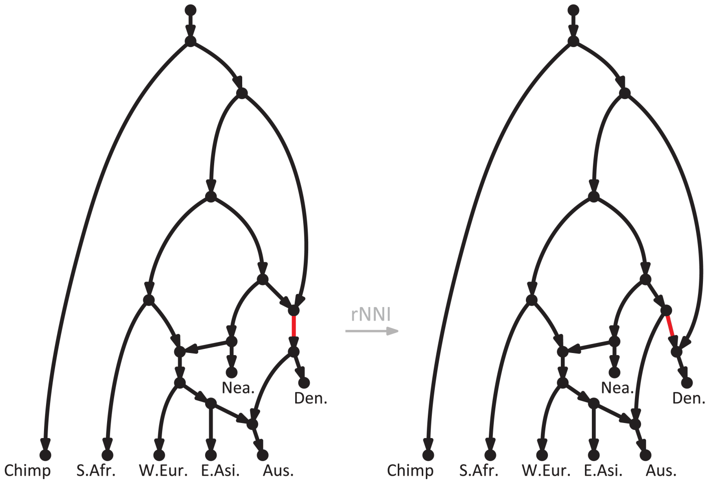

Figure 4: (Gambette et al., 2017) Phylogenetic network showing hypothetical evolutionary scenarios relating modern human populations and their closest relatives. The edge to which rNNI is applied is marked red. The edge is not flipped in this case.

Gambette et al. (2017) generalized NNI for rooted binary phylogenetic networks as follows:

Definition 6. *Let $\mathcal{N}$ be a rooted binary phylogenetic network on X with edges on vertex pairs $\{s,u\}$, $\{u,v\}$ and $\{v,t\}$ but not on $\{u,t\}$ and $\{s,v\}$. The* rNNI *operation on these edges respectively replaces them by edges on $\{u,t\}$, $\{u,v\}$ and $\{s,v\}$ with the following rules:*

1. *the in- and outdegrees of s and t are not affected by the move;*
2. *$\mathcal{N}$ remains a rooted binary phylogenetic network. That is, the in- and outdegrees of u and v remain at most 2 and the obtained network is acyclic.*

An rNNI move replacing edges a_1, a_2, a_3 by edges a_4, a_5, a_6 is denoted by $(a_1, a_2, a_3 \to a_4, a_5\,, a_6\,)$.

Observe that rNNI moves are reversible: if $(a_1, a_2, a_3 \to a_4, a_5, a_6)$ is an rNNI move turning $\mathcal{N}_1$ into $\mathcal{N}_2$, then $(a_4, a_5, a_6 \to a_1, a_2, a_3)$ is an rNNI move turning $\mathcal{N}_2$ into $\mathcal{N}_1$. We say the latter's move is the first move's inverse. Gambette et al. (2017) derived the following conditions for each possible rNNI move:

Lemma 1. *Each rNNI move on rooted binary phylogenetic network N is of one of the following types.*

(1) $(us, uv, vt \to ut, uv, vs)$ *and there is no s-v path in $\mathcal{N}$;*

(1)* $(us, uv, vt \to ut, vu, vs)$*, there is no s-v path and v is a reticulation in $\mathcal{N}$;*

(2) $(su, uv, tv \to sv, uv, tu)$ *and there is no u-t path in $\mathcal{N}$;*

(2)* $(su, uv, tv \to sv, vu, tu)$*, there is no u-t path and u is a tree vertex in $\mathcal{N}$;*

(3) $(su, uv, vt \to sv, uv, ut)$*, u is a reticulation and v a tree vertex in $\mathcal{N}$;*

(3)* $(su, uv, vt \to sv, vu, ut)$ *and there is no u-v path besides uv in $\mathcal{N}$;*

(4) $(us, uv, tv \to vs, uv, tu)$ *and there is no s-t path in $\mathcal{N}$.*

These moves are illustrated in Figure 5.

Proof. (Gambette et al., 2017)
An rNNI move assumes the existence of three edges: one on $\{s, u\}$, one on $\{u, v\}$ and one on $\{v, t\}$. We consider the four possible edge directions for $\{s, u\}$ and $\{v, t\}$, while without loss of generality the third edge is fixed as uv. For each of these combinations, we consider two possible moves: one leaving the direction of uv unchanged, which gives cases (1)-(4), and one reversing its direction, which gives cases $(1^*) - (3^*)$. Note that a (4^*) move $(us, uv, tv \to vs, vu, tu\ )$ is not an rNNI move, as v would have indegree 0. For each of the seven resulting cases, restrictions are also provided that ensure that the conditions given in Definition 6 are satisfied (e.g., u has to be a reticulation in (3)). It can be easily verified that the associated restrictions on every move are correct. See Figure 5. ■

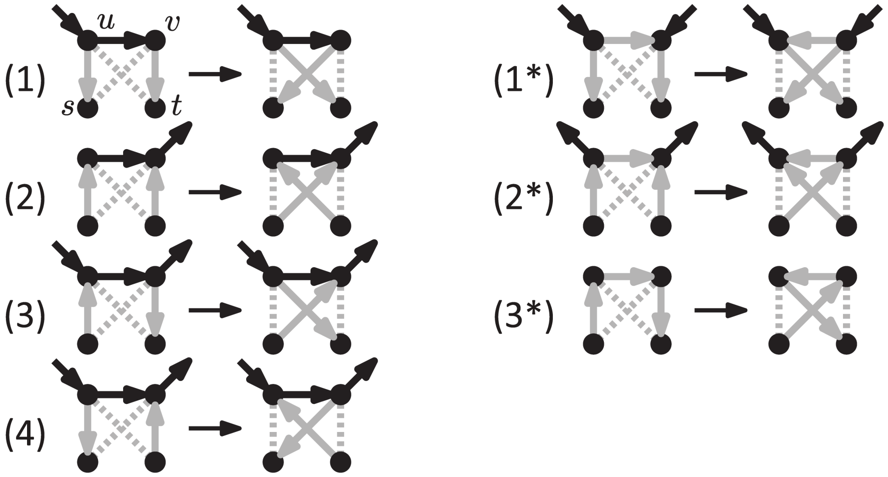

Figure 5: All possible rNNI moves on a rooted binary phylogenetic network. Dashed edges indicate that there is no edge between those vertices. Gray edges are those that change with the move. If vertices have additional incident edges that are not drawn, then these may be oriented either way.

For ease of notation, we will say 'rNNI applied to $\mathcal{N}$' to refer to applying an rNNI move to any of the edges of rooted binary phylogenetic network $\mathcal{N}$. rNNI is a generalization of the NNI move. NNI could be restricted to rooted phylogenetic networks by taking operations (1) to (4). Instead, with rNNI we have the possibility to flip the direction of the internal edge on which rNNI is applied, visualised by the additional operations (1*) to (3*).

Apart from defining the rNNI move, Gambette et al. (2017) also explored the space of rooted binary phylogenetic networks that are mutually *reachable* by rNNI moves.

Definition 7. *Let $\mathcal{N}_1$ and $\mathcal{N}_2$ be rooted binary phylogenetic networks on X.*
We say $\mathcal{N}_2$ is (rNNI-)reachable *from $\mathcal{N}_1$ if there exists a finite sequence of rNNI moves that transforms $\mathcal{N}_1$ to $\mathcal{N}_2$.*

Note that if $\mathcal{N}_2$ is reachable from $\mathcal{N}_1$, then $\mathcal{N}_1$ is reachable from $\mathcal{N}_2$ by reversibility of rNNI moves. To explore the space of reachable rooted binary phylogenetic networks further, let us introduce the concept of the *reticulation number* of a network.

Definition 8. *Let $\mathcal{N} = (V, E)$ be a phylogenetic network on X. The* reticulation number *of $\mathcal{N}$ is defined as $\mathcal{R}(\mathcal{N}) = |E| - |V| + 1$.*

Recall that a tree-based network is merely a tree with additional horizontal edges. The intuition behind the reticulation number is that it equals the number of horizontal edges that are missing from a base tree. However, we know that not every phylogenetic network has a base tree. Hence such definition would not be sufficient in the extension to non-tree-based phylogenetic networks.

Proposition 1. *Let $\mathcal{N}_1$ and $\mathcal{N}_2$ be two rooted binary phylogenetic networks on X. Then the following are equivalent:*

1. *$\mathcal{N}_1$ and $\mathcal{N}_2$ have the same number of reticulation vertices r;*
2. *$\mathcal{N}_1$ and $\mathcal{N}_2$ have the same number of vertices n;*
3. *$\mathcal{N}_1$ and $\mathcal{N}_2$ have the same number of edges m;*

Proof. Let $\mathcal{N}$ be a rooted binary phylogenetic network on X. Let t be the number of tree vertices of $\mathcal{N}$. Let r be the number of reticulation vertices of $\mathcal{N}$. Let n be the total number of vertices of $\mathcal{N}$. Let m be the total number of edges of $\mathcal{N}$.
We first show equivalence between 1 and 2:
Let I be the sum of indegrees of the vertices of $\mathcal{N}$ and let O be the sum of the outdegrees of the vertices of $\mathcal{N}$. We know that the sum of the outdegrees of vertices of a directed graph equals the sum of indegrees of the vertices of a directed graph. Hence, writing $I = t + 2r + |X|$ and $O = 2 + 2t + r$, we see that $t + 2r + |X| = 2 + 2t + r$, where rewriting yields $t = r + |X| - 2$. Furthermore, notice we can write n as $n = 1 + t + r + |X|$. Substituting t into the formula for n we get

$$n = 2(r + |X|) - 1.$$

This shows that two networks on the same leaf set X have the same number of reticulation vertices if and only if they have the same number of vertices.
We now show equivalence between 1 and 3:
Note that the number of edges in a directed graph is equal to the sum of the indegrees of its vertices, which we have already derived above. By substituting the expression above for t in that for I, we see

$$m = 3r + 2|X| - 2.$$

This shows that two networks on X have the same number of reticulations if and only if they have the same number of edges.

By transitivity, these two cases prove equivalence of the three statements. ■

Corollary 1. *Let $\mathcal{N} = (V, E)$ be a rooted binary phylogenetic network on X. Then $\mathcal{R}(\mathcal{N})$ equals the number of reticulation vertices r.*

Proof. Let $\mathcal{N} = (V, E)$ be a rooted binary phylogenetic network on X. Using the results in the proof of Proposition 1 we see that $|V| = 2(r + |X|) - 1$ and that $|E| = 3r + 2|X| - 2$. Now, using $\mathcal{R}(\mathcal{N}) = |E| - |V| + 1$ from Definition 8 and substituting for $|E|$ and $|V|$, we see

$$\mathcal{R}(\mathcal{N}) = |E| - |V| + 1 = 3r + 2|X| - 2 - (2(r + |X|) - 1) + 1 = r.$$

So $\mathcal{R}(\mathcal{N}) = r$. ■

Proposition 2. *Let $\mathcal{N}$ be a rooted binary phylogenetic network on X. Applying an rNNI move to $\mathcal{N}$ preserves the network's reticulation number.*

Proof. Let $\mathcal{N}_1$ be a rooted binary phylogenetic network on X with r reticulation vertices. Let $\mathcal{N}_2$ be the rooted binary phylogenetic network on X after applying an rNNI move to $\mathcal{N}_1$. Since an rNNI move does not add or remove any vertices, it follows from Proposition 1 that an rNNI does not change the number of reticulation vertices. In other words, $\mathcal{N}_2$ contains r reticulation vertices. ■

These results imply that for some arbitrary rooted binary phylogenetic network $\mathcal{N}$ on X the set of rNNI-reachable networks is a subset of the set of networks with the same reticulation number. In fact, these sets are equal. The proof that the set of networks with the same reticulation number is a subset of rNNI-reachable networks relies on results on the topic of unrooted phylogenetic networks, hence we will omit the proof in this thesis and refer to (Gambette et al., 2017). This result allows us to derive the following theorem:

Theorem 1. *Let $\mathcal{N}_1$ be a rooted binary phylogenetic network on X with reticulation number $\mathcal{R}(\mathcal{N}_1)$. Let $\mathcal{N}_2$ be a rooted binary phylogenetic network on X with reticulation number $\mathcal{R}(\mathcal{N}_2) = \mathcal{R}(\mathcal{N}_1)$. Then, $\mathcal{N}_2$ is rNNI-reachable from $\mathcal{N}_1$.*

Theorem 1 implies that rNNI moves induce a natural metric over the sets of the rooted binary networks of fixed reticulation number. We will show how rNNI moves induce a natural metric by constructing a *metric space*.

Definition 9. *Kreyszig (1989) A* metric space *is a pair (X, d), where X is a set and d is a metric on X (or distance function on X), that is, a function defined on $X \times X$ such that for all $x, y, x \in X$ we have:*

(M1) d is real-valued, finite and non-negative;

(M2) $d(x,y) = 0 \Leftrightarrow x = y$;

(M3) $d(x,y) = d(y,x)$;

(M4) $d(x,z) \leq d(x,y) + d(y,z)$.

For ease of notation, we formally define the following sets:

Definition 10. *We define $\Omega(X)$ as the set of rooted binary phylogenetic networks on leaf set X. Furthermore we define $\Omega(X;r)$ as the set of rooted binary phylogenetic networks on leaf set X with fixed reticulation number r. We define $TBN(X)$ as the set of rooted binary phylogenetic networks on X that are tree-based. Finally, we define $TBN(X;r)$ as the set of rooted binary phylogenetic networks on X with fixed reticulation number r that are tree-based.*

And we introduce the following function based on rNNI-moves:

Definition 11. *We define $d_{rNNI} : \Omega(X;r) \times \Omega(X;r) \to \mathbb{N} \cup \{0\}$ as a function describing the minimum number of rNNI moves required to transform $\mathcal{N}_1 \in \Omega(X;r)$ into $\mathcal{N}_2 \in \Omega(X;r)$.*

We prove that, in fact, d_{rNNI} is a metric on $\Omega(X;r)$. To prove the points of Definition 9, we first need to define a notion of equality on elements of the sets introduced in Definition 10.

Definition 12. *Let $G = (V, E)$, $G' = (V', E')$ be two graphs. We say G' and G are* isomorphic *if there exists a bijection $f : V \to V'$ such that $(u, v) \in E$ if and only if $(f(u), f(v)) \in E'$. Then f is called a* graph isomorphism*.*

Definition 13. *Let $\mathcal{N}_1 = (V_1, E_1) \in \Omega(X)$. Let $\mathcal{N}_2 = (V_2, E_2) \in \Omega(X)$. We say $\mathcal{N}_1 = \mathcal{N}_2$ if there exists a graph isomorphism f, where additionally $f(x) = x$, for all $x \in X$.*

We define equality like this, since we only consider the leaves as labelled vertices, as per Definition 1. So all internal vertices may be relabeled in any way for two phylogenetic networks to be considered equal.

Theorem 2. *$(\Omega(X;r), d_{rNNI})$ is a metric space.*

Proof.

(M1) Let $\mathcal{N}_1, \mathcal{N}_2 \in \Omega(X;r)$. Since d_{rNNI} counts the number of operations applied to a network in $\Omega(r;X)$ to reach another network in $\Omega(r;X)$ it follows that d_{rNNI} is integral and non-negative. Moreover, Theorem 1 implies that $d_{rNNI}(\mathcal{N}_1, \mathcal{N}_2) < \infty$.

(M2) Let $\mathcal{N}_1, \mathcal{N}_2 \in \Omega(X; r)$. Suppose $d_{rNNI}(\mathcal{N}_1, \mathcal{N}_2) = 0$. Then there exists a sequence of zero rNNI moves to transform $\mathcal{N}_1$ into $\mathcal{N}_2$, but then $\mathcal{N}_1 = \mathcal{N}_2$.
Now suppose $\mathcal{N}_1 = \mathcal{N}_2$. To transform $\mathcal{N}_1$ into $\mathcal{N}_2$, we can use zero rNNI-moves to turn $\mathcal{N}_1$ into $\mathcal{N}_2$. Moreover since d_{rNNI} is non-negative, we get that $d_{rNNI}(\mathcal{N}_1, \mathcal{N}_2) = 0$.

(M3) Let $\mathcal{N}_1, \mathcal{N}_2 \in \Omega(X; r)$. Suppose $d_{rNNI}(\mathcal{N}_1, \mathcal{N}_2) < d_{rNNI}(\mathcal{N}_2, \mathcal{N}_1)$. Then $d_{rNNI}(\mathcal{N}_2, \mathcal{N}_1)$ does not describe the minimum number of rNNI moves required to transform $\mathcal{N}_2$ into $\mathcal{N}_1$. We can construct a shorter sequence of rNNI moves transforming $\mathcal{N}_2$ to $\mathcal{N}_1$ as follows. Take a sequence of rNNI moves transforming $\mathcal{N}_1$ to $\mathcal{N}_2$ of length $d_{rNNI}(\mathcal{N}_1, \mathcal{N}_2)$. Of this sequence, take each individual move's inverse and reverse the order of the moves in the sequence to obtain a sequence of rNNI moves transforming $\mathcal{N}_2$ to $\mathcal{N}_1$. This sequence has length $d_{rNNI}(\mathcal{N}_1, \mathcal{N}_2)$, which we assumed to be smaller than $d_{rNNI}(\mathcal{N}_2, \mathcal{N}_1)$. We reach a contradiction. It follows that $d_{rNNI}(\mathcal{N}_1, \mathcal{N}_2) \not< d_{rNNI}(\mathcal{N}_2, \mathcal{N}_1)$ (In a similar manner, one can prove that $d_{rNNI}(\mathcal{N}_1, \mathcal{N}_2) \not> d_{rNNI}(\mathcal{N}_2, \mathcal{N}_1)$). It follows that $d_{rNNI}(\mathcal{N}_1, \mathcal{N}_2) = d_{rNNI}(\mathcal{N}_2, \mathcal{N}_1)$.

(M4) Let $\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3 \in \Omega(X; r)$. Take a sequence of rNNI moves transforming $\mathcal{N}_1$ to $\mathcal{N}_2$ of length $d_{rNNI}(\mathcal{N}_1, \mathcal{N}_2)$. Append to this sequence another sequence of rNNI moves transforming $\mathcal{N}_2$ to $\mathcal{N}_3$ of length $d_{rNNI}(\mathcal{N}_2, \mathcal{N}_3)$ to obtain a sequence of rNNI moves transforming $\mathcal{N}_1$ to $\mathcal{N}_3$. This sequence has length $d_{rNNI}(\mathcal{N}_1, \mathcal{N}_2) + d_{rNNI}(\mathcal{N}_2, \mathcal{N}_3)$. It follows that $d_{rNNI}(\mathcal{N}_1, \mathcal{N}_3) \leq d_{rNNI}(\mathcal{N}_1, \mathcal{N}_2) + d_{rNNI}(\mathcal{N}_2, \mathcal{N}_3)$.

■

2 On the combinatorial properties of tree-based phylogenetic networks

In this section we discuss several characterisations of the class of tree-basedness. We discuss the characterisations introduced by Zhang (2016) and A. Francis et al. (2018) in particular. Zhang (2016) was the first to fully characterise tree-based binary networks, A. Francis et al. (2018) introduced several characterisations. Mapping out these alternative characterisations help in understanding tree-based rooted binary phylogenetic networks better. Moreover, it also helps in understanding and developing proximity measures, making this chapter a toolbox for the next chapter about proximity measures. Many characterisations rely on a *matching* in a *bipartite graph*.

Definition 14. *Let $G = (V, E)$ be a simple graph. A* matching *is a subset M of pairwise non-adjacent edges.*

Definition 15. *A* bipartite graph *is a graph $G = (V, E)$ whose vertices can be partitioned into two disjoint sets V_1 and V_2 such that for all $e \in E$: $|e \cup V_1| = |e \cup V_2| = 1$.*

In this thesis, we will write $G = (U \cup V, E)$ to denote a bipartite graph with edge set E and bipartition of the vertex set $U \cup V$.

2.1 Matching characterisation

Zhang (2016) characterised rooted binary phylogenetic networks using bipartite graphs.

Definition 16. *Let $\mathcal{N} \in \Omega(X)$. Let T be the set of tree vertices in $\mathcal{N}$ that are parents of a reticulation. Here we consider the root as a tree vertex. Let R be the set of reticulations in $\mathcal{N}$. Let $E' = \{\{t, r\} : t \in T, r \in R, (t, r) \in E\}$. We then define $\mathcal{Z}_{\mathcal{N}} = (T \cup R, E')$ as a bipartite graph on T and R.*

Furthermore we note that a path is *maximal* if one cannot append any vertices to the path to make it longer. Zhang (2016) established the following characterisations.

Theorem 3. *Let $\mathcal{N}$ be a rooted binary phylogenetic network. Then the following are equivalent:*

1. *$\mathcal{N}$ is tree-based;*
2. *$\mathcal{Z}_{\mathcal{N}}$ has a matching such that each reticulation vertex is matched;*
3. *$\mathcal{Z}_{\mathcal{N}}$ has no maximal path that starts and ends with reticulations.*

Before we prove this Theorem, we need to introduce a classic result in combinatorics: *Hall's marriage Theorem* (Hall, 1935) in its graph theory context:

Theorem 4. *Let $G = (V_1 \cup V_2, E)$ be a bipartite graph. Then, there exists a matching M such that each vertex in V_1 is matched if and only if for all $W \subseteq V_1$ the number of vertices in G adjacent to vertices in W is at least $|W|$.*

We omit the proof as we would digress too much from the topic of the thesis. We can now prove Theorem 3.

Proof. Let $\mathcal{N} \in \Omega(X)$.
First, we show that 1 and 2 are equivalent. Assume $\mathcal{N}$ is tree-based. Then, we can remove a set of reticulation edges $\mathcal{E}$ from $\mathcal{N}$ to obtain a base tree S. Observe that no pair of edges in $\mathcal{E}$ is adjacent because $\mathcal{N}$ is binary and S is a tree. Hence, $\mathcal{E}$ is a matching in $\mathcal{Z}_{\mathcal{N}}$. Conversely, assume $\mathcal{Z}_{\mathcal{N}}$ has a matching M such that each reticulation vertex is matched. Since $\mathcal{N} = (V, E)$ is binary, exactly one reticulation edge per reticulation vertex is not in M. Thus, $(V, E \backslash M)$ is a base tree of $\mathcal{N}$, i.e., $\mathcal{N}$ is tree-based.

Next, we show that 2 and 3 are equivalent. Without loss of generality, $\mathcal{Z}_{\mathcal{N}} = (T \cup R, E')$ is connected. Otherwise, we consider the connected components of $\mathcal{Z}_{\mathcal{N}}$.

Assume $\mathcal{Z}_\mathcal{N}$ has a matching such that each reticulation vertex is matched and suppose by contradiction that $v_1, v_2 \in R$ define a maximal path P in $\mathcal{Z}_\mathcal{N}$ which starts in v_1 and ends in v_2. Then, Hall's theorem tells us that the number of vertices in $\mathcal{Z}_\mathcal{N}$ adjacent to vertices in $U \subseteq R$ is at least $|U|$. In particular, $|T| \geq |R|$. However, reticulation vertices and tree vertices alternate in P. This means, if $P = \mathcal{Z}_\mathcal{N}$, then $|\mathrm{T}| = |\mathrm{R}| - 1$ because $v_1, v_2 \in R$. Moreover, if $P \neq \mathcal{Z}_\mathcal{N}$, i.e., $\mathcal{Z}_\mathcal{N}$ is a cycle, then we have to remove more tree vertices than reticulation vertices from $\mathcal{Z}_\mathcal{N}$ to obtain P. Thus, we arrive at a contradiction.

Conversely, assume $\mathcal{Z}_\mathcal{N}$ has no matching such that each reticulation vertex is matched. Then, by Hall's theorem, there exists a subset of reticulation vertices $U \subseteq R$ such that the number of vertices in $\mathcal{Z}_\mathcal{N}$ adjacent to vertices in U is less than $|U|$. Since $\mathcal{Z}_\mathcal{N}$ is connected and $\mathcal{N}$ is binary, each vertex adjacent to vertices in U has exactly two adjacent vertices in U. This means, U induces a path $P = (W, E'')$ in $\mathcal{Z}_\mathcal{N}$ starting and ending in a reticulation vertex such that $|W \backslash U| = |U| - 1$. Thus, P is a maximal path containing U. ■

To illustrate the matching characterisation, consider Figure 6. Notice here that $\mathcal{Z}_\mathcal{N}$ does not have a matching such that each reticulation is matched and hence $\mathcal{N}$ is not tree-based.

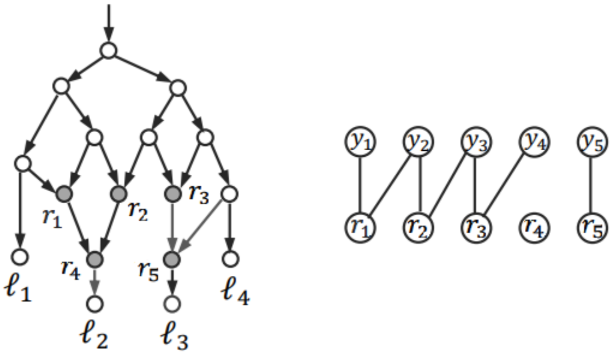

Figure 6: (Zhang, 2016) (left) A rooted binary phylogenetic network $\mathcal{N}$ on taxa $\{\ell_1, \ell_2, \ell_3, \ell_4\}$ in which reticulation vertices are indicated by shaded circles and labeled by r_n, $n \in \{1, 2, 3, 4, 5\}$. (right) The bipartite graph $\mathcal{Z}_\mathcal{N}$ where $R = r_n$ and $T \supset y_n$. tree vertices that have no edge to a reticulation vertex are left out of this figure.

2.2 Antichain characterisation

We will now discuss the antichain characterisation, as first introduced in A. Francis et al. (2018). Let us introduce definitions on *posets*, *chains* and *antichains* before showing the characterisation itself.

Definition 17. *A* partially ordered set (poset) *is an ordered pair $P := (A, \preceq)$, where A is a set and $\preceq$ is a partial order on A. I.e. a binary relation that is reflexive, anti-symmetric and transitive.*

Definition 18. *A* strict poset *is an ordered pair $P := (A, \prec)$, where A is a set and $\prec$ is a strict partial order on A, i.e. a binary relation that is irreflexive, asymmetric and transitive.*

Definition 19. *Let $P = (A, \prec)$ be a strict poset. A* chain *of length n is a sequence $a_1 \prec a_2 \prec ... \prec a_n$ for $a_i \in A$, $i \in \{1, ..., n\}$.*

Definition 20. *Let $P = (A, \preceq)$ be a poset. An* antichain *is a set $\mathcal{A} \subseteq A$ with the property that for all distinct $u, v \in \mathcal{A}$: $u \not\preceq v$ and $v \not\preceq u$.*

We can view a rooted phylogenetic network as a poset, as shown in the next Lemma.

Lemma 2. *Let $\mathcal{N} = (V, E) \in \Omega(X)$. We define a partial order $(\preceq)$ on V as follows:*
For any two vertices $u, v \in V$, $u \preceq v$ if and only if there exists a directed path from v to u.

Proof. We prove reflexivity, anti-symmetry and transitivity of $\preceq$ on an arbitrary rooted phylogenetic network. Let $\mathcal{N} = (V, E) \in \Omega(X)$. Let $u, v \in V$.

- Reflexivity: By taking the trivial path $\{v\}$, it follows that $v \preceq v$.
- Anti-symmetry: Assume $u \preceq v$, then $v \preceq u$ would be cycle-creating, which is not allowed in a rooted binary phylogenetic network. Hence $v \npreceq u$.
- Transitivity: Concatenating the directed paths from v to u and w to v yields a directed path from w to u, hence $u \preceq w$.

■

We also define a strict partial order $(\prec)$ on V if for $u, v \in V$ we have $u \preceq v$, but $u \neq v$.

Any tree-based network $\mathcal{N}$ satisfies the antichain-to-leaf property (A. Francis et al., 2018).

Definition 21. *Let $\mathcal{N} = (V, E) \in \Omega(X)$.*
We say $\mathcal{N}$ satisfies the antichain-to-leaf property *if for every antichain $\mathcal{A} \subseteq V$ of size k, there exist k vertex disjoint paths from the elements of the antichain to the leaves of $\mathcal{N}$.*

Lemma 3. *Let $\mathcal{N} \in \Omega(X)$. If, in fact, $\mathcal{N} \in TBN(X)$, then $\mathcal{N}$ satisfies the antichain-to-leaf property.*

Proof. Let $\mathcal{N} \in \Omega(X)$. Assume that, in fact, $\mathcal{N} \in TBN(X)$. Since $\mathcal{N}$ has a base tree, the size of any antichain is clearly bounded above by the number of leaves by the definition of a tree. But then also for any antichain of $\mathcal{N}$ there exists a vertex disjoint path from every element of the antichain to the leaves of $\mathcal{N}$. These paths can be formed by following for each element in the antichain the subsequent outgoing edge in the base tree until a leaf is encountered. ■

The antichain-to-leaf property alone is not sufficient to show that an arbitrary rooted binary phylogenetic network is tree-based. See Figure 7.

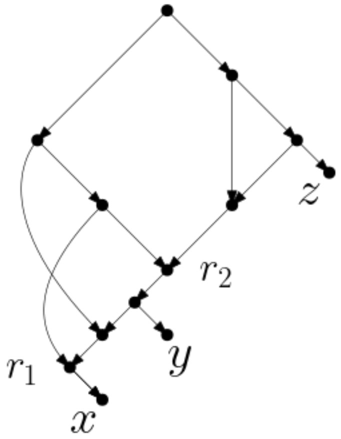

Figure 7: This rooted binary phylogenetic network on the leaf set $\{x, y, z\}$ is not tree-based: r_1 and r_2 are two reticulation vertices that both cannot be matched to a tree vertex parent. However, this network satisfies the antichain-to-leaf property.

A. Francis et al. (2018) characterized tree-based rooted binary phylogenetic networks by strengthening the antichain-to-leaf property and by introducing another bipartite graph.

Definition 22. *Let $\mathcal{N} = (V, E) \in \Omega(X)$. Let V_1, V_2 be copies of V. Let $E' = \{\{u, v\} : u \in V_1, v \in V_2, (u, v) \in E\}$. We then define $\mathcal{G}_{\mathcal{N}} = (V_1 \cup V_2, E')$ as a bipartite graph on V_1 and V_2.*

Theorem 5. *Let $\mathcal{N} \in \Omega(X)$. Then the following are equivalent:*

1. *$\mathcal{N}$ is tree-based;*
2. *$\mathcal{N}$ has an antichain $\mathcal{A} \subseteq V$, and a partition Π of V into $|\mathcal{A}|$ chains each of which forms a path in $\mathcal{N}$ ending at a leaf in X;*
3. *For all $U \subseteq V$, there exists a set of vertex disjoint paths in $\mathcal{N}$ each ending at a leaf in X such that each element of U is on exactly one path;*
4. *There is no pair of subsets $U_1, U_2 \subseteq V$ such that $|U_1| > |U_2|$ and (i) every path from a vertex in U_1 to a vertex in X traverses a vertex in U_2, and (ii) for $\{i, j\} = \{1, 2\}$, if there is a path from a vertex in U_i to a vertex in U_i, then this path traverses a vertex in U_j;*
5. *The vertex set of $\mathcal{N}$ can be partitioned into a set of vertex disjoint paths, each of which ends at a leaf in X;*
6. *$\mathcal{G}_{\mathcal{N}}$ has a matching of size $|V| - |X|$.*

Here, 2 to 5 strengthen the antichain-to-leaf property and 6 characterises tree-based rooted binary phylogenetic networks in terms of bipartite graphs. A visualization of this characterisation for properties 5 and 6 can be seen in Figure 8.

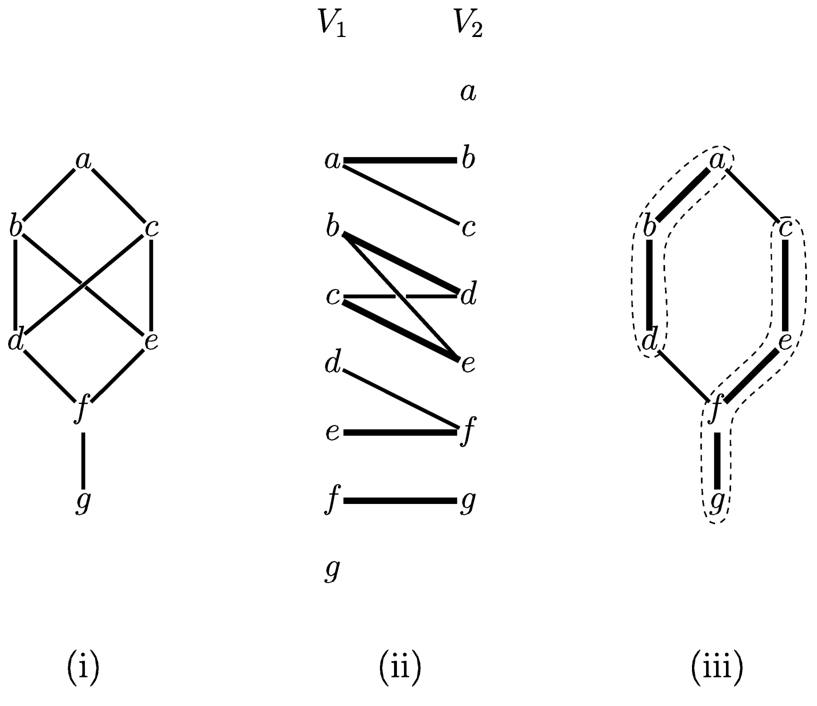

Figure 8: (A. Francis et al., 2018) (i) A rooted binary phylogenetic network $\mathcal{N}$ on taxon $\{g\}$ that is not tree-based. The author of this figure left out edge directions as they are implied by the positioning of the vertices. (ii) The bipartite graph $\mathcal{G}_{\mathcal{N}}$ and a maximum-sized matching of $\mathcal{G}_{\mathcal{N}}$ indicated by the bold edges. (iii) Two corresponding vertex disjoint paths for $\mathcal{N}$, partitioning the vertex set of $\mathcal{N}$.

Before we prove Theorem 5, we need to introduce some operations.

Definition 23. *Let G be a digraph. Let $e \in E(G)$. To* subdivide *e is the operation of deleting e, adding a new vertex v, and joining v to both endpoints of e, where the new edges are directed in the original direction of e. Any directed graph obtained from G by a sequence of edge subdivisions is called a* subdivision *of G.*

Definition 24. *Let $\mathcal{N} \in \Omega(X)$. Let $e \in E(\mathcal{N})$. Consider subdividing e, which creates a new vertex v_1. Create another vertex v_2 and add the edge (v_1, v_2). Notice that v_1 is a tree vertex and that v_2 is a leaf. This operation is referred to as* attaching a leaf *to e (or to $\mathcal{N}$, when speaking more generally).*

We also introduce a Lemma on subdivisions in rooted binary phylogenetic networks.

Lemma 4. *Let $\mathcal{N} \in \Omega(X)$. Let T be a subdivision of $\mathcal{N}$. Then, for any non-empty subset U of $V(T)$ there exists a set of vertex disjoint directed paths in T, each of which ends at a leaf of T and each vertex in U lies on exactly one path.*

Proof. (A. Francis et al., 2018)
We apply strong induction on the number of vertices n of T. Base case $n = 1$ trivially holds. Now assume $n \geq 2$ and assume Lemma 4 holds for all subdivisions of a rooted binary tree with at most $n - 1$ vertices. Let $U \subseteq V(T)$, with U non-empty. Since $n \geq 2$, T has exactly one of the following vertices:

(1) A leaf $x \in X$, whose parent u has degree 2; or

(2) A vertex v that is a parent of two leaves, say x and y.

In each case, we establish the induction hypothesis, starting with (1).

For (1), let T' be the subdivision of a rooted binary tree obtained from T by deleting x and its incident edge, so that u is now a leaf of T'. Define

$$U' := \begin{cases} U, & \text{if } U \text{ does not contain } x; \\ U - \{x\}, & \text{if } U \text{ contains } x \text{ and also contains } u; \\ (U - \{x\}) \cup \{u\}, & \text{if } U \text{ contains } x \text{ but not } u. \end{cases}$$

Observe that $U' \subseteq V(T')$. Therefore, since T' contain $n - 1$ vertices, it follows by induction that Lemma 4 holds and so there is a set of disjoint paths in T', each of which ends at a leaf of T' and each vertex in U lies on exactly one path. Now one of these paths ends at u. Replacing this path with the one that extends it to end at x gives a set of vertex disjoint paths in T, each of which ends at a leaf of T and each vertex in U lies on exactly one path. Thus Lemma 4 holds for (1).

Now consider (2). Let T' be the subdivision of a rooted binary phylogenetic tree obtained from T by deleting y and its incident edge. Note that T' has $n - 1$ vertices. If U does not contain y, then let $U' := U$. By induction, there is a set of vertex disjoint paths in T', each of which ends at a leaf of T' and each vertex in U' lies on exactly one path. This set of paths also works for U in T. On the other hand, if U does contain y, then let $U' := U \setminus \{y\}$. By induction, there is a set of at most $|U'| = |U| - 1$ vertex disjoint paths in T', each of which ends at a leaf of T' and each vertex in U' lies on exactly one path. Adding the trivial path consisting of y to this set of paths, we obtain a set of vertex disjoint paths in T, each ending at a leaf of T and each vertex in U lying on exactly one path. This completes the proof for (2). ■

We are now ready to prove the majority of Theorem 5. We prove equivalence of points 1 to 5. Equivalence between 1 to 5 and 6 is proven independently as Corollary 2 in Chapter 3.

Proof. Let $\mathcal{N} = (V, E) \in \Omega(X)$.
We first show that 1 implies 2.
Assume that $\mathcal{N}$ is tree-based. Let T be a base tree for $\mathcal{N}$. Let $U = V - X$. Then by Lemma 4 it follows that there is a collection of vertex disjoint paths in T each ending at a vertex in X and each vertex in $\mathcal{N}$ lying on exactly one path. Choose $\mathcal{A} = X$; the vertex sets of these paths form the blocks of the required partition Π of V.
We now show that 2 implies 3.

Assume Π is a partition of V with the property that each block in Π is the vertex set of a path in $\mathcal{N}$ ending at a leaf in X. Let $U \subseteq V$. Then Π provides a set of vertex disjoint paths each ending at a leaf in X and with each vertex in U on exactly one path.
We now show that 3 implies 4. We prove the contrapositive.
Suppose 4 is false. Then there exist subsets U_1 and U_2 of V with $|U_1| > |U_2|$ that satisfy the two traversal conditions (i) and (ii). We show $U = U_1$ fails to satisfy property 3. First observe that if P is a path in $\mathcal{N}$ ending at X, then P contains at least as many vertices of U_2 as U_1. To see this, observe that because of traversal conditions (i) and (ii), moving along P, we alternate between vertices in U_1 and vertices in U_2. That is, for $\{i, j\} = \{1, 2\}$, if we traverse a vertex in U_i, then the next vertex we traverse in $U_i \cup U_j$ is a vertex in U_j. Moreover, for each vertex in U_1 on P, there is a subsequent vertex in U_2 on P. Hence there are at least as many vertices of U_2 as U_1 in P. Thus any set of vertex disjoint paths in $\mathcal{N}$ each ending at a leaf in X collectively contains at least as many vertices in U_2 as U_1. But then it is not possible for such a set of paths to collectively collect all vertices in U_1 since $|U_2| < |U_1|$. So $U = U_1$ violates property 3.
We now show that 4 implies 1. We prove the contrapositive.
Suppose $\mathcal{N}$ is not tree-based. Then by Theorem 3 there is a maximal path in $\mathcal{Z}_{\mathcal{N}}$ that starts and ends in R. Denote such path as $r_1 t_1 r_2, ..., t_{k-1}, r_k$ for some $k \in \mathbb{N}$. Let y be the parent of r_1, that is not t_1, and let y' be the parent of r_k that is not t_{k-1}. Since the path is maximal, both y and y' must be reticulations of $\mathcal{N}$. Let $U_1 = \{y, t_1, t_2, ..., t_{k-1}, y'\}$ and let $U_2 = \{r_1, r_2, ..., r_k\}$. These sets have the following properties:

(1) $|U_1| > |U_2|$

(2) U_1 is the set of all parents of all vertices in U_2, and

(3) U_2 is the set of all children of all vertices in U_1.

Now (2) implies that every path from a vertex in U_1 to a vertex in X traverses a vertex in U_2. Furthermore, (2) and (3) imply that, if there is a path from a vertex in U_1 to another vertex in U_1, then this path traverse a vertex in U_2. Similarly, if there is a path from a vertex in U_2 to another vertex in U_2, then this path traverses a vertex in U_1. It follows that U_1 and U_2 provide an instance for which property 4 fails.
We now show that 3 implies 5.
Taking $U = V$ in property 3 immediately gives a path system satisfying property 5.
Lastly, we show that 5 implies 2. Assume P is a set of paths satisfying property 5. Taking $\mathcal{A} = X$ and $\Pi = P$ gives an antichain and partition of V into $|\mathcal{A}|$ chains that satisfies property 2.

These implications are sufficient for the proof of equivalence between 1 to 5. ∎

Theorem 5.2 is closely related to a classical result in combinatorics, namely Dilworth's Theorem (Dilworth, 1950). We show how Dilworth's Theorem relates to this property. Recall that we can view a rooted binary phylogenetic $\mathcal{N}$ as a poset. Let us first begin with a definition.

Definition 25. *Let $G = (V, E)$ be a graph. A stable set is a set $S \subseteq V$ for which no vertices in S are adjacent to other vertices in S.*

We can now state Dilworth's Theorem.

Theorem 6. *Let P be a finite poset. The minimum number of chains into which the elements of P can be partitioned is equal to the maximum number of elements in an antichain of P.*

Part of the proof of Dilworth's Theorem relies on a result in Gallai & Milgram (1960).The proof of this result will be omitted as we would digress too much from the topic of the thesis.

Proof. (Bondy & Murty, 2008) Let $P := (V, \prec)$, and denote by $D := D(P)$ the digraph whose vertex set is V and whose edges are the ordered pairs (u, v) such that $u \prec v$ in P. Let α denote the number of vertices in a largest stable set. Let π be the minimum number of disjoint directed paths that is needed to cover the vertex set of D. Chains and antichains in P correspond in D to directed paths and stable sets, respectively. Because no two elements in an antichain of P can belong to a common chain, the minimum number of chains in a chain partition is at least as large as the maximum number of elements in an antichain; that is, $\pi \geq \alpha$. Gallai & Milgram (1960) showed $\pi \leq \alpha$. Therefore $\pi = \alpha$. ■

Hence, if we regard a rooted binary phylogenetic network as a poset as in Lemma 2, we see that Theorem 5.2 is exactly Dilworth's Theorem with the additional requirement that chains must form directed paths, i.e. we cannot use the transitivity of a poset to 'jump' from one vertex from another. Figure 9 visualizes how Dilworth's Theorem relates to tree-basedness in rooted binary phylogenetic networks.

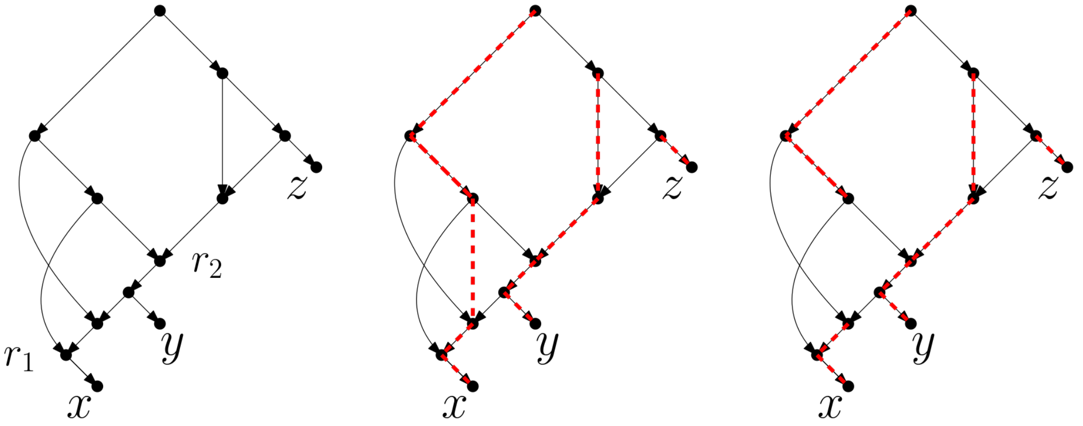

Figure 9: (left) A phylogenetic network $\mathcal{N}$ on taxa $\{x, y, z\}$ that is not tree-based: reticulation vertices r_1 and r_2 cannot both be matched with a tree vertex parent. (middle) $\mathcal{N}$ has an antichain of size 3 ($\{x, y, z\}$). The vertices of $\mathcal{N}$ are partitioned into three chains denoted by red dashed lines. (right) $\mathcal{N}$ can be divided into 4 vertex disjoint paths, denoted by the red dashed lines, but not 3.

2.3 Other characterisations

Next to the matching characterisation and the antichain characterisation, there exist other characterisations for tree-basedness of rooted phylogenetic networks. These other characterisations will not be of focus in the remainder of the thesis. Hence, these characterisations will be given less attention compared to the matching and antichain characterisation. Still, it is good practice to list them here as they help us in getting a stronger understanding of the combinatorial properties of tree-based rooted binary phylogenetic networks.

The anti-chain-to-leaf property property is sufficient to show that an arbitrary rooted binary phylogenetic network is tree-based in the class of *temporal networks*, introduced by Baroni et al. (2006).

Definition 26. *Let $\mathcal{N} = (V, E)$ be a rooted binary phylogenetic network. Let $u, v \in V$. $\mathcal{N}$ is* temporal *if there exists a map $\lambda : V \to \mathbb{R}$ such that $\lambda(u) < \lambda(v)$ for all tree edges (u, v), and $\lambda(u) = \lambda(v)$ for all reticulation edges (u, v). Then λ is said to be a* temporal map *for $\mathcal{N}$.*

Theorem 7. *Let $\mathcal{N}$ be a temporal network. Then the following are equivalent:*

1. *$\mathcal{N}$ is tree-based;*
2. *$\mathcal{N}$ satisfies the antichain-to-leaf property.*

Proof. (A. Francis et al., 2018) Let $\mathcal{N} \in \Omega(X)$. Furthermore, let $\mathcal{N}$ be temporal.
Assume $\mathcal{N}$ is tree-based. Then $\mathcal{N}$ satisfies the antichain-to-leaf property as proven in Lemma 3.

Now, assume $\mathcal{N}$ is not tree-based. We will prove that it then cannot satisfy the antichain-to-leaf property. Since $\mathcal{N}$ is not tree-based, it follows by Theorem 3.3 that $\mathcal{Z}_{\mathcal{N}} = (T \cup R, E)$ contains a maximal path

$$r_1 t_1 r_2 \cdots t_{k-1} r_k$$

that starts and ends in R. If $k = 1$, then both parents, q and q' say, of r_1 are reticulations, in which case $U = \{q, q'\}$ is an antichain in $\mathcal{N}$ that violates the antichain-to-leaf property. Thus we may assume that $k \geq 2$. Let q be the parent of r_1 that is not t_1 and let q' be the parent of r_k that is not t_{k-1}. Since the path is maximal, both q and q' are reticulations in $\mathcal{N}$. Let $U = \{q, t_1, t_2, \ldots, t_{k-1}, q'\}$. Since $\mathcal{N}$ is temporal, there is a temporal map λ for $\mathcal{N}$ which necessarily gives

$$\lambda(q) = \lambda(r_1) = \lambda(t_1) = \lambda(r_2) = \cdots = \lambda(t_{k-1}) = \lambda(r_k) = \lambda(q'),$$

and so λ is constant on U. If U is an antichain, then U violates the antichain-to-leaf property, since any set of paths that connects the $k+1$ vertices in U to the leafs in X need to pass through the k vertices in $\{r_1, r_2 \ldots, r_k\}$ and so these paths cannot be disjoint. Therefore, suppose that U is not an antichain. Then there is a directed path P in $\mathcal{N}$ from a vertex $u \in U$ to another vertex $u' \in U$. Moreover, every edge in P must be a reticulation edge of $\mathcal{N}$. Otherwise, if P contains a tree edge, then $\lambda(u') > \lambda(u)$, contradicting the constancy of λ on U. In particular, the only possible tree vertex of $\mathcal{N}$ in P is the first vertex. Thus P must include q or q'. If q (respectively q') can be reached by a directed path from a vertex in $\{r_1, r_2, \ldots, r_k\}$, denote this vertex by r_q (respectively $r_{q'}$). It is easily checked that $r_q \neq r_{q'}$. Now let

$$U' = \begin{cases} U - \{q\}, & \text{if } r_q \text{ exists;} \\ U - \{q'\}, & \text{if } r_{q'} \text{ exists;} \\ U - \{q, q'\}, & \text{if } r_q \text{ and } r_{q'} \text{ exist.} \end{cases}$$

The set U' is an antichain of size k or $k-1$. Now any path in $\mathcal{N}$ that connects a vertex in U' with a vertex in X must traverse a vertex in $\{r_1, r_2, \ldots, r_k\}$. But if r_q exists, then any path traversing r_q must also traverse r_1. Similarly, if $r_{q'}$ exists, then any path traversing $r_{q'}$ must also traverse r_{k-1}. In all possibilities for U', it follows that U' does not satisfy the antichain-to-leaf property.

■

Jetten & Van Iersel (2018) introduced another characterisation for tree-based rooted binary phylogenetic networks based on a matching in some bipartite graph.

Definition 27. *Let $\mathcal{N} \in \Omega(X)$. Let O denote the set of vertices whose only children are reticulations. Let R denote the set of reticulations in $\mathcal{N}$. Let $E' = \{\{o, r\}, o \in O, r \in R, (o, r) \in E\}$. We then define $\mathcal{B}_{\mathcal{N}} = (O \cup R, E')$ as a bipartite graph on O and R.*

Vertices whose only children are reticulations are referred to as *omnians*.

Theorem 8. *Let $\mathcal{N}$ be a rooted binary phylogenetic network. Then the following are equivalent:*

1. *$\mathcal{N}$ is tree-based;*
2. *There exists a matching M in $\mathcal{B}_{\mathcal{N}}$ with $|M| = |O|$.*

Proof. (Jetten & Van Iersel, 2018)
Let $\mathcal{N} \in \Omega(X)$.
Assume that $\mathcal{N}$ is tree-based with base tree T. Colour every edge of $\mathcal{B}_{\mathcal{N}}$ that corresponds to an edge in T. When an omnian has outdegree 2 and both outgoing edges are contained in T, decolourize one of the two corresponding edges of $\mathcal{B}_{\mathcal{N}}$, arbitrarily. Hence, each vertex of O is incident to at most one coloured edge. Since T is a rooted tree, it contains at most one incoming edge of each reticulation vertex. Hence, also each vertex of R is incident to at most one coloured edge. So the coloured edges of $\mathcal{B}_{\mathcal{N}}$ form a matching M. Because T is a base tree, we are on the same leaf set as $\mathcal{N}$, and so all omnians are covered by M.

Now, assume there exists a matching M in $\mathcal{B}_\mathcal{N}$ with $|M| = |O|$, i.e., all omnians are covered by M. Construct a set E' of edges as follows: add the outgoing edge of every reticulation vertex and the incoming edge of all tree vertices to E'. Additionally, for each edge of M, add the corresponding edge of $\mathcal{N}$ to E', if it has not yet been added. For every reticulation vertex that has not yet been covered, add one of its incoming edges to E', arbitrarily. The tree T, consisting of all vertices of $\mathcal{N}$ and the set of edges E', is a rooted spanning tree, because there is precisely one incoming edge of every vertex contained in T. Moreover, T has the same leaf set as $\mathcal{N}$, because O is covered by M. Hence, it follows that $\mathcal{N}$ is tree-based. ■

A. R. Francis & Steel (2015) also introduced characterisations for tree-based rooted binary phylogenetic networks based on *admissible sets*.

Definition 28. *Let $\mathcal{N} = (V, E) \in \Omega(X)$. Let S_1 denote the subset of edges whose tail has out-degree 1 or whose head has in-degree 1. Let $v \in V$. A subset $S \subseteq E$ is* admissible *if*

- *$S1 \subseteq S$; and*

(C1) if v has in-degree 2, then exactly one of its incoming edges is in S; and

(C2) if v has out-degree 2, then exactly one of its outgoing edges is in S.

Theorem 9. *Let $\mathcal{N} = (V, E) \in \Omega(X)$. Then the following are equivalent:*

1. *$\mathcal{N}$ is tree-based;*
2. *There exists an admissible subset of the edges of $\mathcal{N}$;*
3. *there exists an independent subset of edges E' of E for which $T' = (V, E - E')$ is a base tree of $\mathcal{N}$.*

A. R. Francis & Steel (2015) used a different, and frankly cumbersome, definition for tree-based phylogenetic networks. We introduce them for the sake of the proof of Theorem 9, but it will play no further role in this thesis.

Definition 29. *Let $\mathcal{N} \in \Omega(X)$. We say that $\mathcal{N}$ is* tree-based *with* sparse base tree *T if $\mathcal{N}$ can be obtained from the following procedure. First, subdivide each edge of T some number $n_a \in \mathbb{N}$ of times and call the resulting degree-2 vertices* attachment points*. The resulting tree is a base tree T' for $\mathcal{N}$ derived from T. Next, sequentially place additional edges between any two attachment points, provided that the network remains binary and acyclic, until $\mathcal{N}$ is recovered. We call these additional edges* linking edges*.*

The procedure described in Definition 29 can be seen in Figure 1 when reading the Figure from right to left. Here, the graph on the right is the so-called sparse base tree.

Proof. (A. R. Francis & Steel, 2015)
We first show equivalence between 1 and 2. Assume that $\mathcal{N}$ is tree-based, and let T' be a base tree for $\mathcal{N}$. Then the set S of edges of T' contains S_1, and S also satisfies conditions (C_1) and (C_2) for every vertex $v \in V$ of in-degree or out-degree 2 , respectively. Thus S is admissible.
Now, assume that S is an admissible subset of the edges of $\mathcal{N}$. Consider the network $\mathcal{N}' = (V, S)$ consisting of all the vertices in $\mathcal{N}$ and just the edges in S. We claim that this is a base tree, with root ρ (the root of $\mathcal{N}$) and leaf set X (the leaf set of $\mathcal{N}$). First, notice that T' has no vertex of in-degree 2, by condition (C_1). Second, every edge e that is incoming to a leaf $x \in X$ of $\mathcal{N}$ is present in S', and so e is also an edge of T' and so the leaf set of T' contains X. It remains to check that (1) T' contains no other leaves, and (2) the only vertex of in-degree 0 in T' is ρ. For (1), suppose v is vertex of T' that is not in X. Then in $\mathcal{N}$, v has strictly positive out-degree. If v has out-degree 1 in $\mathcal{N}$, then the outgoing edge from v is present in S_1 and thereby in S, while if v has out-degree 2, at least one of the two outgoing edges is present in S by condition (C_2). Thus, v cannot be a leaf of T'.
We now show equivalence between 1 and 3. Let $\mathcal{N} \in \Omega(X)$.
Assume that $\mathcal{N}$ is tree-based. Then no two linking edges can share the same vertex. Moreover, a

linking edge is never incoming to an in-degree 1 vertex, or outgoing from an out-degree 1 vertex, and so deleting linking edges will not disconnect the network. Thus if we take E' to be the linking edges for $\mathcal{N}$ then $T' = (V, E - E')$ is an associated base tree for $\mathcal{N}$. So T' is a base tree for $\mathcal{N}$
Now assume 2 holds for some set E'. Since T' is connected it has leaf set X, and so T' is a subdivision of some sparse base tree T. Now if we regard each edge in E' as a linking edge then we recover $\mathcal{N}$ (since E' is independent, and T' is connected, these arcs are all placed validly). ■

3 Proximity measures for tree-basedness

Proximity measures were introduced by A. Francis et al. (2018) to determine how close rooted binary phylogenetic networks were to being tree-based. Later, several proximity measures were extended to non-binary rooted phylogenetic networks and unrooted phylogenetic networks, though a proximity measure for a specific type of phylogenetic network might not translate directly to a phylogenetic network of another type. This can be seen in the open problems from Fischer & Francis (2020). For a proximity measure we require some sort of measurement on a topological property of a phylogenetic network, that relates it to being tree-based. Because there are different ways to characterise tree-based networks as we have seen in previous chapters, there exist many different ways to measure proximity to tree-basedness. Practically, the topological property that defines the proximity measure should be some property that is useful to measuring closeness to tree-based in biological applications. Moreover, to prove that a measure is well-defined usually requires knowledge on different characterisations of tree-based networks. It is important that a measure is well-defined, else it may not always return a value for some networks, or, it may not return a consistent value for tree-basedness.

3.1 Definition of a proximity measure

Informally put, we would like for a proximity measure for tree-basedness of rooted binary phylogenetic networks to be a function that maps a rooted binary phylogenetic network on taxa X to a natural number n, where n indicates proximity to tree-basedness of $\mathcal{N}$. From this follows that closeness to being tree-based cannot be determined for some $\mathcal{N} \in \Omega(X)$ by the proximity measure μ if $\mu(\mathcal{N})$ does not return a value, i.e. $\forall \mathcal{N} \in \Omega(X) : \mu(\mathcal{N}) \notin \mathbb{N} \cup \{0\}$. We would also like for $\mu(\mathcal{N})$ to attain its minimum value if and only if $\mathcal{N}$ is tree-based for the sake of consistency of the proximity measure. These two requirements for proximity measures were implied (informally) for all proximity measures in A. Francis et al. (2018). We form the following formal requirements for a proximity measure:

Definition 30. *A* proximity measure *is a function μ on the metric space $\Omega(X)$ such that*

1. $\forall \mathcal{N} \in \Omega(X) : \mu(\mathcal{N}) \in \mathbb{N} \cup \{0\}$;
2. $\forall \mathcal{N} \in \Omega(X) : \mu(\mathcal{N})$ *is minimal if and only if* $\mathcal{N} \in TBN(X)$.

In this thesis, as well as in other research papers, the convention is used that a function will be introduced as a proximity measure, after which it is claimed that the proximity measure is *well-defined* if it meets the criteria of Definition 30.

3.2 Review of known proximity measures

In this subsection we will review several proximity measures that have been introduced in A. Francis et al. (2018).

3.2.1 Definitions of the known proximity measures

A. Francis et al. (2018) defined the following proximity measures:

Definition 31. *Let $\mathcal{N} \in \Omega(X)$*

1. *Let $l(\mathcal{N})$ be the minimum number of leaves in $V(\mathcal{N}) \setminus X$ that must be present as leaves in a rooted spanning tree of $\mathcal{N}$.*
2. *Let $p(\mathcal{N}) = d(\mathcal{N}) - |X|$, where $d(\mathcal{N})$ is the smallest number of vertex disjoint paths that partition the vertices of $\mathcal{N}$.*

3. *Let $t(\mathcal{N})$ be the minimum number of leaves that need to be attached to $\mathcal{N}$ so the resulting network is tree-based on a different set of taxa.*

Notice that in Figure 8.(iii) we have $d(\mathcal{N}) = 2$ and $|X| = 1$, implying $p(\mathcal{N}) = 2 - 1 = 1$.

Proposition 3. *The proximity measures l, p, t are well-defined.*

Proof. Let $\mathcal{N} \in \Omega(X)$.
We show for each measure that Definition 30.1 and Definition 30.2 hold
First consider measure l.

1. $V(\mathcal{N}) \setminus X$ is a finite set. Moreover, $l(\mathcal{N})$ is clearly non-negative and integral. Hence $l(\mathcal{N}) \in \mathbb{N} \cup \{0\}$.

2. Clearly 0 is the minimal value for $l(\mathcal{N})$. We claim $l(\mathcal{N}) = 0$ if and only if $\mathcal{N} \in TBN(X)$.
Assume $l(\mathcal{N}) = 0$, i.e. a rooted spanning tree of $\mathcal{N}$ requires zero leaves in $V(\mathcal{N}) \setminus X$, then this rooted spanning tree is a base tree of $\mathcal{N}$, so $\mathcal{N} \in \text{TBN}(X)$.
Now, assume $\mathcal{N} \in \text{TBN}(X)$, then $\mathcal{N}$ has a base tree, which is a rooted spanning tree with leaves that of leaf set X. So, there exists a rooted spanning tree of $\mathcal{N}$ that requires zero leaves in $V(\mathcal{N}) \setminus X$.

So l is well-defined.
Now consider measure p.

1. The number of vertex disjoint paths that partition $V(\mathcal{N})$is bounded from above by $|V(\mathcal{N})|$, which is a finite set. Moreover, p($\mathcal{N}$) is non-negative since we need $d(\mathcal{N}) = |X|$ vertex disjoint paths to cover every leaf $x \in X$ alone. Lastly, $p(\mathcal{N})$ is clearly integral. Hence $p(\mathcal{N}) \in \mathbb{N} \cup \{0\}$.

2. The result follows from equivalence between Theorem 5.1 and Theorem 5.2.

So p is well defined.
Now consider measure t.

1. We propose a procedure which proves that $t(\mathcal{N})$ attains a value. Attach a new leaf to every reticulation edge of $\mathcal{N}$. The operation of attaching a new leaf to every reticulation edge includes subdividing every reticulation edge by Definition 24. The vertex that is created in the process of subdivision is a tree vertex, since one of its children is the reticulation vertex on which the reticulation edge is directed to, and the other child is the newly created leaf by Definition 24. After having done this, we can match every reticulation vertex to each of their newly created tree vertex parents. It follows from Theorem 3 that the resulting network is tree-based on taxa X unionized with all newly created leaves. From this procedure follows that $t(\mathcal{N}) \leq E(\mathcal{N})$. Moreover, $t(\mathcal{N})$ is clearly non-negative and integral. Hence $t(\mathcal{N}) \in \mathbb{N} \cup \{0\}$.

2. Clearly 0 is the minimal value for $t(\mathcal{N})$. We claim $t(\mathcal{N}) = 0$ if and only if $\mathcal{N} \in TBN(X)$.
Assume $t(\mathcal{N}) = 0$, i.e. zero leaves need to be attached to $\mathcal{N}$ so that the resulting network is tree-based. Then by definition, $\mathcal{N} \in \text{TBN}(X)$.
Now, assume $\mathcal{N} \in \text{TBN}(X)$. Then zero leaves need to be attached so that the resulting network is tree-based.

■

In fact, A. Francis et al. (2018) proved the following Theorem.

Theorem 10. *Let $\mathcal{N} \in \Omega(X)$. Then*

$$l(\mathcal{N}) = p(\mathcal{N}) = t(\mathcal{N}).$$

Proof. (A. Francis et al., 2018)
Let $\mathcal{N} \in \Omega(X)$.

We first show $l(\mathcal{N}) \leq p(\mathcal{N})$. Assume that Π is the partition of the vertex set V of $\mathcal{N}$ induced by a set of $d(\mathcal{N}) = p(\mathcal{N}) + |X|$ vertex disjoint paths of $\mathcal{N}$. Due to the minimality of $d(\mathcal{N})$, there are $|X|$ paths ending at a leaf; $p(\mathcal{N})$ of these are not. Let $p = p(\mathcal{N})$. Consider the paths $\pi_1, \pi_2, \ldots, \pi_p$ not ending at an element in X. Since the paths in Π are vertex disjoint and partition V, the paths Π forms a spanning sub-forest of $\mathcal{N}$. So, by adding, for each path in Π, one edge of $\mathcal{N}$ directed into the starting vertex, except for the root, we construct a rooted spanning tree T of $\mathcal{N}$. Note that such edge is guaranteed to exist since every vertex that is not the root has a positive indegree. Also, the ingoing edge of the starting vertex must originate from a vertex in different path of Π. The leaves of T not in X are precisely the last vertices of the paths $\pi_1, \pi_2, \ldots, \pi_p$. Since there are $p(\mathcal{N})$ of these paths, it follows that $l(\mathcal{N}) \leq p(\mathcal{N})$.

We next show that $p(\mathcal{N}) \leq t(\mathcal{N})$. Let $\mathcal{N}'$ be a tree-based network that is obtained from $\mathcal{N}$ by attaching $t(\mathcal{N})$ leaves. Let T be a base tree for $\mathcal{N}'$, and let U denote the leaf set of T. If we now apply Lemma 4 with the same choice of U and T, then T can be partitioned into at most $|U| = t(N) + |X|$ paths each of which ends at an element in U, of which $d(\mathcal{N})$ paths partition $V(\mathcal{N})$. Thus $d(\mathcal{N}) \leq t(\mathcal{N}) + |X|$, implying $p(\mathcal{N}) \leq t(\mathcal{N})$

Lastly, we show that $t(\mathcal{N}) \leq l(\mathcal{N})$. Let T be a rooted spanning tree of $\mathcal{N}$ that realises $l(\mathcal{N})$. For each leaf ℓ of T that is not in X, attach a new leaf to an edge directed out of ℓ. If ℓ is a tree vertex of $\mathcal{N}$, then choose arbitrarily one of the outgoing edges to attach the new leaf. Let $\mathcal{N}'$ denote the resulting phylogenetic network. Since T is a rooted spanning tree of $\mathcal{N}$, it is easily seen that we can extend T to give a rooted spanning tree of $\mathcal{N}'$ whose leaf set coincides with the leaf set of $\mathcal{N}'$. Hence $\mathcal{N}'$ is tree-based, and it follows that $t(\mathcal{N}) \leq l(\mathcal{N})$. This completes the proof of the theorem. ■

A. Francis et al. (2018) introduced two proximity measures at the end of their paper, for which they did not prove any bounds like they did for their first three measures.

Definition 32. *Let $\mathcal{N} = (V, E) \in \Omega(X)$. We say that $\mathcal{N}' = (V', E')$ is* embedded *in $\mathcal{N}$ if $\mathcal{N}'$ is a subgraph of $\mathcal{N}$ up to edge subdivisions.*

Definition 33. *Let $\mathcal{N} \in \Omega(X)$, A. Francis et al. (2018) defined the following proximity measures:*

1. *Let $a(\mathcal{N})$ be the minimum value of $|V(\mathcal{N}) \setminus V(T)|$, where T is a rooted tree embedded in $\mathcal{N}$ on the same leafset as $\mathcal{N}$.*

2. *Let $b(\mathcal{N})$ be the minimum number n of rooted trees T_i, $i \in \{1, ..., n\}$, embedded in $\mathcal{N}$ on the same leaf set as $\mathcal{N}$ such that $\bigcup_{i=1}^{n} V(T_i) = V(\mathcal{N})$.*

Proposition 4. *The proximity measures a, b are well-defined.*

Proof. Let $\mathcal{N} \in \Omega(X)$.
We show for each measure that Definition 30.1 and Definition 30.2 hold
First consider measure a.

1. Clearly, $a(\mathcal{N}) \leq |V(\mathcal{N})|$. Moreover, $a(\mathcal{N})$ is integral and non-negative. Hence $a(\mathcal{N}) \in \mathbb{N} \cup \{0\}$.

2. Clearly 0 is the minimal value for $a(\mathcal{N})$. We claim $a(\mathcal{N}) = 0$ if and only if $\mathcal{N} \in TBN(X)$.
   Assume $a(\mathcal{N}) = 0$, i.e. $|V(\mathcal{N}) \setminus V(T)|$ for some rooted tree T embedded in $\mathcal{N}$ on the same leaf set as $\mathcal{N}$. Then the rooted tree that attains this is a rooted spanning tree; a base tree for $\mathcal{N}$. So $\mathcal{N} \in \text{TBN}(X)$.
   Now, assume $\mathcal{N} \in \text{TBN}(X)$, then $\mathcal{N}$ has a base tree, which by definition, is a rooted spanning tree with vertex set $V(\mathcal{N})$. We get $a(\mathcal{N}) = |V(\mathcal{N}) \setminus V(\mathcal{N})| = 0$.

Now consider measure b.

1. One can see that $b(\mathcal{N}) \leq |V(\mathcal{N})|$ since we can take $|V(\mathcal{N})|$ rooted trees, each of which aimed to cover a certain vertex in $\mathcal{N}$. Moreover, $b(\mathcal{N})$ is non-negative and integral, hence $b(\mathcal{N}) \in \mathbb{N} \cup \{0\}$.

2. Clearly 1 is the minimal value for $b(\mathcal{N})$. We claim $b(\mathcal{N}) = 1$ if and only if $\mathcal{N} \in TBN(X)$. Assume $b(\mathcal{N}) = 1$, i.e. there exists a rooted tree T embedded in $\mathcal{N}$ with the same leaf set as $\mathcal{N}$ such that $V(T) = V(\mathcal{N})$. Such tree is a rooted spanning tree; a base tree for $\mathcal{N}$. So $\mathcal{N} \in \mathrm{TBN}(X)$
   Now, assume $\mathcal{N} \in \mathrm{TBN}(X)$. Then $\mathcal{N}$ has a base tree T. This is a rooted tree with the same leaf set as $\mathcal{N}$ where $V(T) = V(\mathcal{N})$. So $b(\mathcal{N}) \leq 1$. Moreover, since clearly $b(\mathcal{N}) \geq 1$, we get $b(\mathcal{N}) = 1$.

■

3.2.2 Complexity of the known proximity measures

A. Francis et al. (2018) proposed a polynomial time algorithm for measures l, p and t. Davidov et al. (2020) gave a polynomial time algorithm for a. By Theorem 10, it suffices to construct a polynomial time algorithm for one of the measures of l, p or t to get a polynomial time algorithm for every other measure l, p or t. Let us first relate p to the size of a maximum-sized matching in bipartite graphs.

Corollary 2. *Let $\mathcal{N} \in \Omega(X)$. Let $u(\mathcal{G}_\mathcal{N})$ denote the number of unmatched vertices of V_1 in a maximum-sized matching of $\mathcal{G}_\mathcal{N}$. Then*

$$p(\mathcal{N}) = u(\mathcal{G}_\mathcal{N}) - |X|.$$

Proof. (A. Francis et al., 2018)
Let $\mathcal{N} \in \Omega(X)$. We first show that $p(\mathcal{N}) \leq u(\mathcal{G}_\mathcal{N}) - |X|$. Let M be a maximum-sized matching of $\mathcal{G}_\mathcal{N}$. Let U_2 denote the set of unmatched vertices in V_2. For each vertex $u \in U_2$, we recursively construct a directed path P_u in $\mathcal{N}$ as follows. Set $u = u_0$ and initially set $P_u = u_0$. If u_0 is unmatched in V_1, then terminate the process and set $P_u = u_0$; otherwise, u_0 is matched in V_1, in which case set $P_u = u_0u_1$, where $(u_0, u_1) \in M$. If u_1 is unmatched in V_1, then terminate the process and set $P_u = u_0u_1$. Otherwise, u_1 is matched in V_1, in which case set $P_u = u_0u_1u_2$, where $(u_1, u_2) \in M$. Since $\mathcal{N}$ is acyclic, this process eventually terminates with the last vertex, u_k say, added to P_u being unmatched in V_1. Repeating this construction for each vertex in U_2, we eventually obtain a collection $\mathcal{P} = \{P_u : u \in U_2\}$ of directed paths in $\mathcal{N}$. Since every edge in a path of $\mathcal{P}$ corresponds to an edge of the matching M in $\mathcal{G}_\mathcal{N}$, the paths in $\mathcal{P}$ are vertex disjoint. Furthermore, every vertex in $\mathcal{N}$ is on some path in $\mathcal{P}$. To see this, suppose there is a vertex $v \in V$ not on a path in $\mathcal{P}$. Clearly, v is matched in V_2. But then, by reversing the above construction starting at v in V_2, it is easily seen that v is on such a path. Since each vertex in X is unmatched in V_1, and noting that the number of paths in $\mathcal{P}$ equals the number of unmatched vertices in V_2, and therefore the number of unmatched vertices in V_1, it follows from the fact that M is of maximum size that

$$p(\mathcal{N}) \leq |\mathcal{P}| - |X| = u\left(\mathcal{G}_\mathcal{N}\right) - |X|$$

We next show that $p(\mathcal{N}) \geq u\left(\mathcal{G}_\mathcal{N}\right) - |X|$. Now let $\mathcal{P}$ be a minimum sized collection of vertex disjoint paths that partitions the vertices of $\mathcal{N}$. Let M be the matching of $\mathcal{G}_\mathcal{N}$ obtained from $\mathcal{P}$ as follows. The edge $(u, v) \in M$ precisely if u and v are consecutive vertices on some path in $\mathcal{P}$. Since the paths in $\mathcal{P}$ are vertex disjoint, M is certainly a matching. As every vertex in $\mathcal{N}$ is on some path in $\mathcal{P}$, the number u_1 of unmatched vertices in V_1 is at least the number of paths in $\mathcal{P}$, as the last vertex of each path in $\mathcal{P}$ is unmatched in V_1. Thus, it follows from the fact that $\mathcal{P}$ is of minimum size that

$$p(\mathcal{N}) = |\mathcal{P}| - |X| = u_1 - |X| \geq u\left(\mathcal{G}_\mathcal{N}\right) - |X|$$

This completes the proof of the Corollary. ■

Notice in Figure 8.(ii) that $u(\mathcal{G}_{\mathcal{N}}) = 2$ and $|X| = 1$, implying $p(\mathcal{N}) = 2 - 1 = 1$.
We can find a maximum-sized matching in a bipartite graph in $\mathcal{O}(n^{5/2})$ (Hopcroft & Karp, 2006), where n is the size of the vertex set of said graph. Furthermore we can construct $\mathcal{G}_{\mathcal{N}}$ in polynomial time.

Algorithm 1 constructG
Input: $\mathcal{N} \in \Omega(X)$
Output: $\mathcal{G}_{\mathcal{N}}$

```
$V_1 \leftarrow V(\mathcal{N})$
$V_2 \leftarrow V(\mathcal{N})$
$E \leftarrow \{\}$
for $v_1 \in V_1$ do
    for $v_2 \in V_2$ do
        if $(v_1, v_2) \in E(\mathcal{N})$ then
            $E \cup \{(v_1, v_2)\}$
return $(V_1 \cup V_2, E)$
```

Proposition 5. *Let $\mathcal{N} \in \Omega(X)$, where $n = |V(\mathcal{N})|$. Then* constructG($\mathcal{N}$) *runs in $\mathcal{O}(n^2)$.*

Proof. Let $\mathcal{N} \in \Omega(X)$ with $n = |V(\mathcal{N})|$. Iteration over two copies of V runs in $\mathcal{O}(n^2)$. What is left is copying, unionizing and every other operation, which each do not exceed $O(n^2)$. ■

If we now find a maximum-sized matching M in $\mathcal{G}_{\mathcal{N}}$ and return the number of unmatched vertices of V_1, we can compute $p(\mathcal{N})$, for all $\mathcal{N} \in \Omega(X)$:

Algorithm 2 getP
Input: $\mathcal{N} \in \Omega(X)$, X
Output: $p(\mathcal{N})$

```
$\mathcal{G} \leftarrow$ constructG($\mathcal{N}$)
$M \leftarrow$ match($\mathcal{G}$)
for $(u, v) \in M$ do
    $V_1 \setminus \{u\}$
return $|V_1| - |X|$
```

Here we assume *match* as a black box $\mathcal{O}(n^{5/2})$ algorithm with parameters a tuple of a vertex set and an edge set, returning a maximum sized matching as an edge set.

Proposition 6. *Let $\mathcal{N} \in \Omega(X)$. running getP($\mathcal{N}$) returns a value equal to $p(\mathcal{N})$.*

Proof. First we construct $\mathcal{G}_{\mathcal{N}}$ using constructG($\mathcal{N}$). This algorithm is correct since we define V_1, V_2 to be equal to a copy of the original vertex set, furthermore we iterate over every vertex pair $v_1 \in V_1$, $v_2 \in V_2$ to check if there is an edge between them and append such edges to E.
Now, we assumed that match($\mathcal{G}$) returns a maximum sized matching in $\mathcal{G}$. By taking u, for all $(u, v) \in M$ and removing them from V_1, we remove every vertex in V_1 that is matched in M. This leaves V_1 as the number of unmatched vertices in V_1. We return the size of this final set V_1 and subtract the size of the leaf set. But then we return $u(\mathcal{G}_{\mathcal{N}}) - |X| = p(\mathcal{N})$. ■

Proposition 7. *Let $\mathcal{N} \in \Omega(X)$, where $n = |V(\mathcal{N})|$. Then* getP($\mathcal{N}$) *runs in $\mathcal{O}(n^{5/2})$.*

Proof. As shown by Proposition 5, constructG($\mathcal{N}$) runs in $\mathcal{O}(n^2)$. Furthermore match($\mathcal{G}$) can run in $\mathcal{O}(n^{5/2})$ (Hopcroft & Karp, 2006). Iterating over every edge in a matching between V_1 and V_2 runs in $\mathcal{O}(n)$. Taking the setminus runs in $\mathcal{O})(1)$. Hence getP($\mathcal{N}$) runs in $\mathcal{O}(n^{5/2})$. ■

In conclusion,

Proposition 8. *Let $\mathcal{N} \in \Omega(X)$, where $n = |V(\mathcal{N})|$. Then $l(\mathcal{N})$, $p(\mathcal{N})$ and $t(\mathcal{N})$ can be computed in $O(n^{5/2})$.*

Proof. Let $\mathcal{N} \in \Omega(X)$. Run getP($\mathcal{N}$). This can be done in $\mathcal{O}(n^{5/2})$ by Proposition 7. The return value of $getP(\mathcal{N})$ equals $p(\mathcal{N})$ by Proposition 6. It follows from Theorem 10 that $l(\mathcal{N}) = p(\mathcal{N}) = t(\mathcal{N})$. ■

Proposition 9. *Let $\mathcal{N} \in \Omega(X)$. Then $a(\mathcal{N})$ can be computed in polynomial time.*

The proof is the content of Davidov et al. (2020); it is too lengthy to cover here. Instead we show a sketch of the proof.
Let $\mathcal{N} \in \Omega(X)$. Notice that $\mathcal{N}$ without a minimal set of $a(\mathcal{N})$ vertices forms a maximal covering subtree: a phylogenetic tree covering the maximum number of vertices in a phylogenetic network. Davidov et al. (2020) provided a polynomial time algorithm to find the maximal covering subtree. First they showed a transformation of $\mathcal{N} \in \Omega(X)$ into a flow network F such that the minimum-cost flow in F induces a maximal-covering subtree of $\mathcal{N}$. This transformation and finding a minimum-cost flow can be done in polynomial time. From the resulting maximal-covering subtree $a(\mathcal{N})$ can be found in polynomial time by counting the vertices of $\mathcal{N}$ that are not present in F.

4 A new proximity measure for tree-basedness

Fischer & Francis (2020) defined several proximity measures for unrooted phylogenetic networks. One open question from that paper regarded lifting the proximity measures from the unrooted case to the rooted case. One proximity measure in particular concerned NNI-moves. Note that we did not formally define sets and proximity measures for unrooted networks; we can omit those since the ideas are very similar to the rooted case and we will not use unrooted networks in any proofs.

Definition 34. *Let $\mathcal{N}$ be an unrooted phylogenetic network on X, then an unrooted proximity measure based on a minimum number of NNI moves is formally defined as*

$$\delta_{NNI}(\mathcal{N}) = \min\left\{d_{NNI}\left(\mathcal{N},\mathcal{N}'\right) \mid \mathcal{N}' \text{ is tree-based}\right\}$$

Where d_{NNI} is the NNI equivalent of d_{rNNI} as in Definition 11. It would be interesting if we were able to lift this proximity measure to the rooted case and if we could generalize to rNNI moves. Moreover, one might ask an inverse question. How tree-based is a given tree-based network - What is the number of moves required to make a tree-based network not tree-based (Fischer & Francis, 2020)? This question may allow us to define a "reverse" proximity measure of sorts. We define it for rooted binary phylogenetic networks.

Definition 35. *A* reverse proximity measure *is a function μ on the metric space $\Omega(X)$ such that*

1. $\forall \mathcal{N} \in \Omega(X) : \mu(\mathcal{N}) \in \mathbb{N} \cup \{0\}$;
2. $\forall \mathcal{N} \in \Omega(X) : \mu(\mathcal{N})$ *is minimal if and only if* $\mathcal{N} \notin TBN(X)$.

Definition 36. *Let $\mathcal{N}$ be an unrooted phylogenetic network on X, then we can define a reverse unrooted proximity measure based on a minimum number of NNI moves as*

$$\delta^{-1}_{NNI}(\mathcal{N}) = \min\left\{d_{NNI}\left(\mathcal{N},\mathcal{N}'\right) \mid \mathcal{N}' \text{ is not tree-based}\right\}.$$

If we now combine Definition 34 and Definition 36, we can define a novel measure that for any phylogenetic network describes the shortest NNI distance to the boundary of the class of tree-based networks, i.e. a tree-based network that can become non-tree-based with one single NNI move. Fischer & Francis (2020) described this for NNI moves in unrooted networks:

Definition 37. *Let $\mathcal{N}$ be an unrooted phylogenetic network on X. A novel measure for unrooted networks associated to tree-basedness with NNI moves is formally defined as*

$$||\mathcal{N}||_{TB} = \begin{cases} \delta_{rNNI}(\mathcal{N}) & \text{if } \mathcal{N} \text{ is not tree-based,} \\ \delta^{-1}_{rNNI}(\mathcal{N}) - 1 & \text{if } \mathcal{N} \text{ is tree-based.} \end{cases}$$

4.1 The rNNI proximity measure

We define our own proximity measure for rooted binary phylogenetic networks based on rNNI moves.

Definition 38. *Let $\mathcal{N} \in \Omega(X)$, then a proximity measure based on a minimum number of rNNI moves is formally defined as*

$$\delta_{rNNI}(\mathcal{N}) = \min\left\{d_{rNNI}\left(\mathcal{N},\mathcal{N}'\right) \mid \mathcal{N}' \in TBN(X)\right\}$$

Proposition 10. *The proximity measure δ_{rNNI} is well-defined.*

Proof. Let $\mathcal{N} \in \Omega(X)$. We show that Definition 30.1 and Definition 30.2 hold.

1. We show how to construct $\mathcal{N}' \in TBN(X;r)$ for every value of $|X|$ and r. We then claim $\delta_{rNNI}(\mathcal{N}) \in \mathbb{N} \cup \{0\}$ by Theorem 1.

   First consider the case where $|X| = 1$.
   Note that $r \geq 2$: if $r = 0$, then $n = 1$ by the formula for n in Proposition 1, but a single vertex is not considered a rooted binary phylogenetic network by Definition 1. If $r = 1$, then $n = 3$: one reticulation vertex, one root and one leaf. These three vertices cannot form a rooted binary phylogenetic network together.
   We construct $\mathcal{N}_{basis1} \in TBN(X;2)$ with $|X| = 1$, which serves as a basis for constructing $\mathcal{N}' \in TBN(X;r)$ with $|X| = 1$ for every fixed value $r \geq 2$.
   Define $\mathcal{N}_{basis1} = (V,E) \in TBN(X;2)$ by

   $$\mathcal{N}_{basis1} = (\{root, t_1, r_1, r_2, x\}, \{(root, t_1), (root, r_1), (t_1, r_1), (t_1, r_2), (r_1, r_2), (r_2, x)\}),$$

   where $root$ is the root of $\mathcal{N}_{basis1}$, t_i are tree vertices, r_i are reticulation vertices and $x \in X$. Note that $|X| = 1$. $(root, t_1, r_1, r_2, x)$ is a path that traverses all vertices of $\mathcal{N}_{basis1}$ and ends in x. It follows from Theorem 5 that $\mathcal{N}_{basis1}$ is tree-based.
   We now show a procedure on how to construct $\mathcal{N}' \in TBN(X;r)$, with $|X| = 1$ for every fixed value $r \geq 2$ from $\mathcal{N}_{basis1}$. Let $r \geq 0$. Take $\mathcal{N}_{basis1}$ and subdivide both outgoing edges of the root. This creates two new vertices v_1, v_2, where initially v_1 is the parent of t_1 and v_2 is the parent of r_1. Add the edge (v_2, v_1). Then v_1 is a reticulation vertex and v_2 is a tree vertex. Execute the process of subdividing both outgoing edges of the root a total number of $r - 2$ times, where each execution we alternate between adding the edge (v_2, v_1) or the edge (v_1, v_2), initially (v_2, v_1). Define $\mathcal{N}'$ to be the resulting network of this procedure. Notice that $\mathcal{N}'$ has r reticulation vertices. The vertices that are created by subdivisions form a path by our construction of alternating edges. Furthermore we can append the path (t_1, r_1, r_2, x) since the initial vertex v_1 after subdivision is the parent of t_1. Lastly we can prepend the root since we always subdivide on edges connected to the root. The resulting path traverses all vertices of $\mathcal{N}'$ and ends in x. It follows that $\mathcal{N}'$ is tree-based by Theorem 5, i.e. $\mathcal{N}' \in TBN(X,r)$ with $|X| = 1$. A visualization of this process on $\mathcal{N}_{basis1}$ can be seen in Figure 10.

   Now consider the case where $|X| \geq 2$.
   We construct $\mathcal{N}_{basis2} \in TBN(X;0)$ with $|X| = 2$, which serves as a basis for constructing $\mathcal{N}' \in TBN(X;r)$ for every fixed value $|X| \geq 2$, $r \geq 0$.
   Define $\mathcal{N}_{basis2} = (V,E) \in TBN(X;0)$ by

   $$\mathcal{N}_{basis2} = (\{root, x_1, x_2\}, \{(root, x_1), (root, x_2)\}),$$

   where $root$ is the root of $\mathcal{N}_{basis2}$ and $x_1, x_2 \in X$. Note that $|X| = 2$. $\mathcal{N}_{basis2}$ is a tree, hence it is tree-based.
   We now show a procedure on how to construct $\mathcal{N}' \in TBN(X;r)$ with $|X| \geq 2$, $r \geq 0$ from $\mathcal{N}_{basis2}$. Let $r \geq 0$. Let $|X| \geq 2$. Take $\mathcal{N}_{basis2}$ and subdivide both outgoing edges of the root. This creates two new vertices v_1, v_2. Add the edge (v_1, v_2). v_1 is then a tree vertex, while v_2 is a reticulation vertex. Execute this process of subdivision and adding edges a total number of r times. Now arbitrary select one outgoing edge of the root. Repeatedly attach a new leaf to this edge $|X| - 2$ times. Define $\mathcal{N}'$ to be the resulting network of this procedure. Notice that $\mathcal{N}'$ has r reticulation vertices and $|X|$ leaves. For every reticulation vertex v_1 that we add to the network, we add and connect a new tree vertex parent v_2 to it. So every reticulation vertex can be matched to a tree vertex parent. It follows from Theorem 3 that $\mathcal{N}'$ is tree-based, i.e. $\mathcal{N}' \in TBN(X;r)$ with $|X| \geq 2$. A visualization of this process on $\mathcal{N}_{basis2}$ can be seen in Figure 11.

   Since there exists $\mathcal{N}' \in TBN(X;r)$ for every fixed value of $|X|$ and r, it follows from Theorem 1 that such $\mathcal{N}'$ is reachable in a finite number of rNNI moves starting from $\mathcal{N} \in \Omega(X;r)$. Furthermore, since d_{rNNI} is a metric on $\Omega(X;r)$, it follows that $\delta_{rNNI}(\mathcal{N}) \in \mathbb{N} \cup \{0\}$.

2. Because d_{rNNI} is a metric, $\delta_{rNNI}(\mathcal{N}) \geq 0$. We claim that $\delta_{rNNI}(\mathcal{N}) = 0$ if and only if $\mathcal{N} \in TBN(X)$.
   Assume $\delta_{rNNI}(\mathcal{N}) = 0$. Then by Definition 38, min $\{d_{rNNI}(\mathcal{N}, \mathcal{N}') \mid \mathcal{N}' \in \text{TBN}(X)\} = 0$, implying $d_{rNNI}(\mathcal{N}, \mathcal{N}') = 0$ for some $\mathcal{N}' \in TBN(X)$. But then by Definition 9.(M2), $\mathcal{N} = \mathcal{N}'$, implying $\mathcal{N} \in TBN(X)$.
   Now, assume $\mathcal{N} \in TBN(X)$. Then by Definition 9.(M2), $d_{rNNI}(\mathcal{N}, \mathcal{N})$=0, implying min $\{d_{rNNI}(\mathcal{N}, \mathcal{N}') \mid \mathcal{N}' \in \text{TBN}(X)\} = \delta_{rNNI}(\mathcal{N}) \leq 0$. But since also $\delta_{rNNI}(\mathcal{N}) \geq 0$, we conclude that $\delta_{rNNI}(\mathcal{N}) = 0$.

■

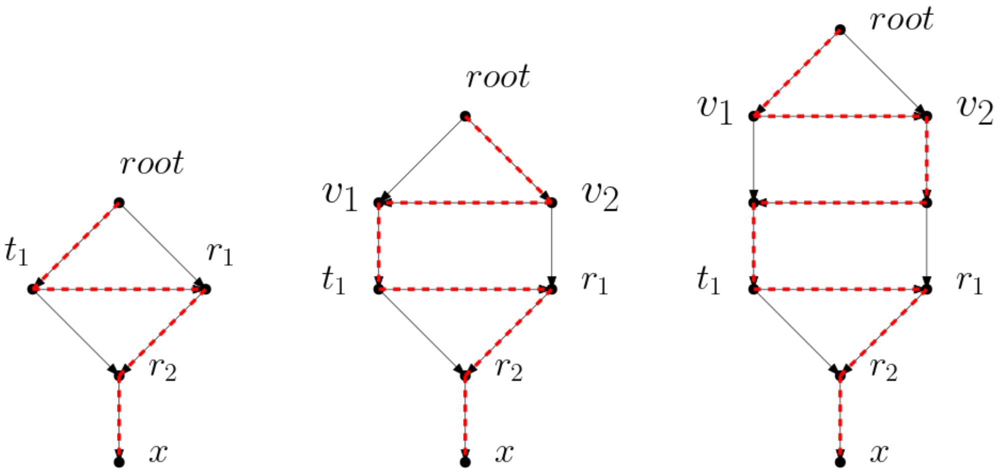

Figure 10: (left) The network $\mathcal{N}_{basis1}$. The vertex disjoint path partitioning the vertices of $\mathcal{N}_{basis1}$ is colored red. (middle) The network after one execution step of increasing the reticulation number r. (right) The network after two execution steps of increasing r. Notice that we alternate between adding edge (v_2, v_1) and (v_1, v_2), else we cannot form a vertex disjoint path partitioning the vertices of the resulting network. We could endlessly increase r by doing more execution steps in this manner.

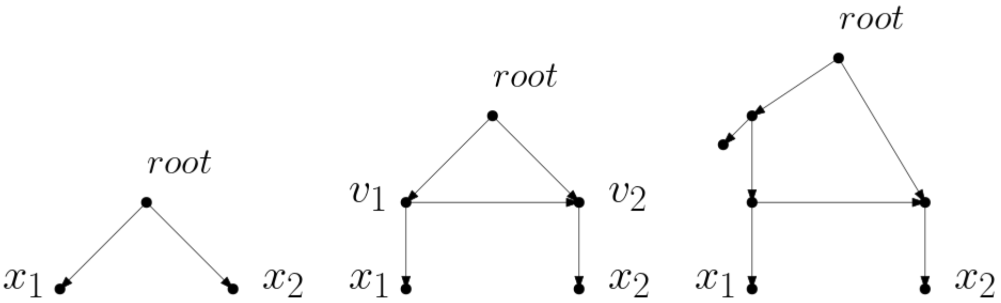

Figure 11: (left) The network $\mathcal{N}_{basis2}$ which is a phylogenetic tree. (middle) The network after one execution step to increase the reticulation number r. This process can clearly be repeated endlessly, where the resulting network is tree-based due to reticulation vertex v_2 being able to be matched to v_1. (right) The network after one execution step to increase r and one execution step to increase $|X|$. We could endlessly attach more leaves to increase $|X|$, with the resulting network being tree-based after every execution step.

Similarly, we define a reverse proximity measure for rooted binary phylogenetic networks based on rNNI moves.

Definition 39. *Let $\mathcal{N} \in \Omega(X)$, then we can define a reverse proximity measure based on a minimum number of rNNI moves as*

$$\delta^{-1}_{rNNI}(\mathcal{N}) = \min\left\{d_{rNNI}\left(\mathcal{N},\mathcal{N}'\right) \mid \mathcal{N}' \notin TBN(X)\right\}.$$

Proposition 11. *The reverse proximity measure δ^{-1}_{rNNI} is well-defined for $\mathcal{N} \in \Omega(X;r)$ with $r \geq 3$*

Proof. We show that Definition 35.1 and Definition 35.2 hold for all $\mathcal{N} \in \Omega(X;r)$ with $r \geq 3$. We also show why Definition 35.2 does not hold for $\mathcal{N} \in \Omega(X;r)$ with $r < 3$.

1. Let $\mathcal{N}_0 \in \Omega(X;0)$, then trivially $\mathcal{Z}_{\mathcal{N}_0}$ always has a matching such that each reticulation vertex is matched. So the space $\Omega(X;0)$ contains no networks that are not tree-based, making $\delta^{-1}_{rNNI}(\mathcal{N}_0)$ ill-defined.

   Now let $\mathcal{N}_1 \in \Omega(X;1)$. $\mathcal{N}_1$ has one reticulation vertex. This reticulation vertex must have at least one tree vertex as parent, hence $\mathcal{Z}_{\mathcal{N}_1}$ has a matching such that each reticulation vertex is matched. So the space $\Omega(X;1)$ contains no networks that are not tree-based, making $\delta^{-1}_{rNNI}(\mathcal{N}_1)$ ill-defined.

   Now let $\mathcal{N}_2 \in \Omega(X;2)$. Suppose for sake of contradiction that $\mathcal{N}_2$ is not tree-based. $\mathcal{N}_2$ by definition has two reticulation vertices. Since $\mathcal{N}_2$ is not tree-based, $\mathcal{Z}_{\mathcal{N}_2}$ does not have a matching such that every reticulation vertex is matched. Assume w.l.o.g reticulation vertex r_1 cannot be matched. r_1 must then have reticulation vertex r_2 as parent. The second parent of r_1 is a tree-vertex or the root, say t_1. Since r_1 cannot be matched, t_1 must also be a parent of r_2. r_2 has another parent, say t_2, which must either be a tree-vertex or the root. The matching $M = \{(t_2, r_2), (t_1, r_1)\}$ is a matching in $\mathcal{Z}_{\mathcal{N}_2}$ such that every reticulation vertex is matched, contradicting our assumption that $\mathcal{N}_2$ is not tree-based. Now, assume both r_1 and r_2 cannot be matched in $\mathcal{Z}_{\mathcal{N}_2}$, then r_1, r_2 both have only reticulation vertices as parents. This is clearly not possible, contradicting our assumption that $\mathcal{N}_2$ is not tree-based. So the space $\Omega(X;2)$ contains no networks that are not tree-based, making $\delta^{-1}_{rNNI}(\mathcal{N}_2)$ ill-defined.

   Now, similarly to the proof of Proposition 10.2, we show how to construct a non-tree-based network $\mathcal{N}'$ for every fixed value of $|X|$ and $r \geq 3$. We first construct $\mathcal{N}_{basis} \in \Omega(X;3)$, with $|X| = 1$, which serves as a basis for constructing non-tree-based $\mathcal{N} \in \Omega(X;r)$ for every fixed value $|X| \geq 1, r \geq 3$.
   Define $\mathcal{N}_{basis} = (V, E) \in \Omega(X;3)$ by

   $$\begin{aligned}\mathcal{N}_{basis} = (&\{root, t_1, t_2, r_1, r_2, r_3, x\},\\ &\{(root, t_1), (root, t_2), (t_1, r_1), (t_1, r_2), (t_2, r_1), (t_2, r_2), (r_1, r_3), (r_2, r_3), (r_3, x))\}),\end{aligned}$$

   where x is the single leaf, *root* is the root, t_i are tree vertices and r_i are reticulation vertices. Note that $|X| = 1$. Since r_3 has two reticulation vertices as parents, it follows from Theorem 3 that $\mathcal{N}_{basis}$ is not tree-based. We now show a procedure on how to construct non-tree-based $\mathcal{N}' \in \Omega(X;r)$ with $|X| \geq 1$, $r \geq 3$ from $\mathcal{N}_{basis2}$. Let $r \geq 3$. Let $|X| \geq 1$. Consider the following procedure on $\mathcal{N}_{basis}$: Subdivide both outgoing edges of the root, resulting in two new vertices v_1, v_2. Add the edge (v_1, v_2). Execute this process a total number of $r - 3$ times. Now, attach a leaf to the ingoing edge of x a total number of $|X| - 1$ times. Define $\mathcal{N}'$ to be the resulting network of this process. Notice that $\mathcal{N}'$ has r reticulation vertices and leaf set size $|X|$, but is non-tree-based because r_3 still has two reticulation vertex parents. It follows from Theorem 1 that $\mathcal{N}'$ is reachable in a finite number of rNNI moves from $\mathcal{N} \in \Omega(X;r)$. Furthermore, since d_{rNNI} is a metric on $\Omega(X;r)$, it follows that $\delta^{-1}_{rNNI}(\mathcal{N}) \in \mathbb{N} \cup \{0\}$. A visualization of the process on $\mathcal{N}_{basis1}$ can be seen in Figure 12.

2. Let $\mathcal{N} \in \Omega(X;r)$ with $r \geq 3$. Because d_{rNNI} is a metric, $\delta^{-1}_{rNNI}(\mathcal{N}) \geq 0$. We claim that $\delta^{-1}_{rNNI}(\mathcal{N}) = 0$ if and only if $\mathcal{N} \notin TBN(X)$.
   Assume $\delta^{-1}_{rNNI}(\mathcal{N}) = 0$. Then by Definition 39, $\min\left\{d_{rNNI}\left(\mathcal{N},\mathcal{N}'\right) \mid \mathcal{N}' \notin \mathrm{TBN}(X)\right\} = 0$,

implying $d_{rNNI}(\mathcal{N},\mathcal{N}') = 0$ for some $\mathcal{N}' \notin TBN(X)$. But then by Definition 9.(M2), $\mathcal{N} = \mathcal{N}'$, implying $\mathcal{N} \notin TBN(X)$.
Now, assume $\mathcal{N} \notin TBN(X)$. Then by Definition 9.(M2), $d_{rNNI}(\mathcal{N},\mathcal{N})$=0, implying $\min\{d_{rNNI}(\mathcal{N},\mathcal{N}') \mid \mathcal{N}' \notin \text{TBN}(X)\} = \delta^{-1}_{rNNI}(\mathcal{N}) \leq 0$. But since also $\delta^{-1}_{rNNI}(\mathcal{N}) \geq 0$, we conclude that $\delta^{-1}_{rNNI}(\mathcal{N}) = 0$.

■

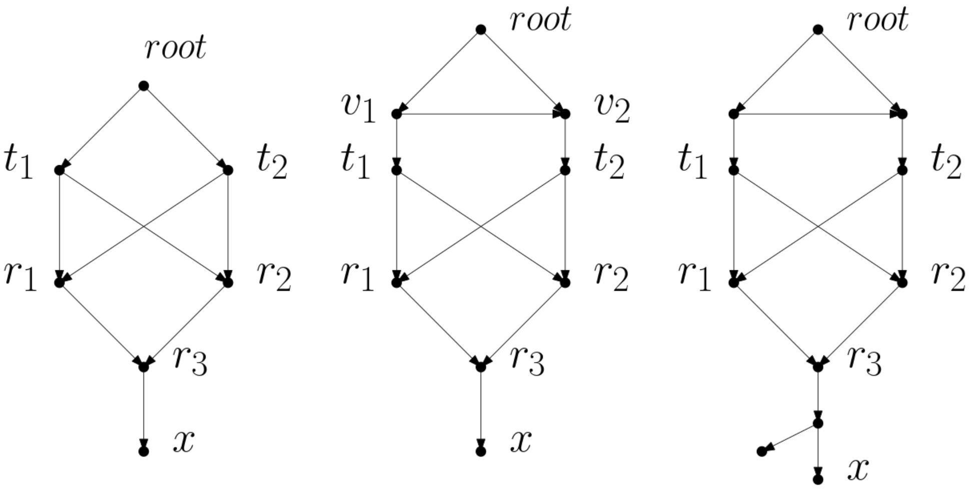

Figure 12: (left) The network $\mathcal{N}_{basis}$. The network is not tree-based due to r_3 having two reticulation vertices as parents. (middle) The network after one execution step of increasing the reticulation number r. This step can clearly be executed endlessly to increase r. (right) The network after one execution step to increase r and one execution step to increase $|X|$. We could endlessly attach more leaves to increase $|X|$. Notice that in the resulting network, r_3 still has two reticulation vertices as parents, making the resulting network non-tree-based.

Lastly, combining Definition 38 and Definition 39 We can also define the novel measure associated to the boundary of tree-basedness with rNNI moves, similar to Definition 37.

Definition 40. *Let $\mathcal{N} \in \Omega(X)$. A novel measure associated to tree-basedness with rNNI moves is formally defined as*

$$||\mathcal{N}||_{rTB} = \begin{cases} \delta_{rNNI}(\mathcal{N}) & \text{if } \mathcal{N} \notin TBN(X), \\ \delta^{-1}_{rNNI}(\mathcal{N}) - 1 & \text{if } \mathcal{N} \in TBN(X). \end{cases}$$

Proposition 12. *Let $\mathcal{N} \in \Omega(X;r)$ with $r \geq 3$. Then $||\mathcal{N}||_{rTB} \in \mathbb{N} \cup \{0\}$.*

Proof. Assume $\mathcal{N} \in \Omega(X;r)$ with $r < 3$. Then by the result in Proposition 11 we see that we must have that $\mathcal{N} \in \text{TBN}(X)$. So we are in the second case of Definition 40, i.e. $\delta_{rNNI}(\mathcal{N}) - 1$. However, it also follows from Proposition 11 that $\delta^{-1}_{rNNI}(\mathcal{N})$ is then ill-defined.
Now assume $\mathcal{N} \in \Omega(X;r)$ with $r \geq 3$. Then we can either be in the first case of Definition 40 or the second case. In any case, $\delta_{rNNI}(\mathcal{N})$ and $\delta^{-1}_{rNNI}(\mathcal{N})$ are both well-defined by Proposition 10 and Proposition 11 respectively.
All in all, $||\mathcal{N}||_{rTB} \in \mathbb{N} \cup \{0\}$ for $\mathcal{N} \in \Omega(X;r)$ with $r \geq 3$. ■

4.2 An upper bound related to proximity measure p

In this subsection we will provide an upper bound for δ_{rNNI}. We do this by relating δ_{rNNI} to the proximity measure p.

4.2.1 Edge flips

To connect the measures p and δ_{rNNI}, we first introduce a useful result from Gambette et al. (2017), which states that we can *flip* the direction of certain edges in a rooted binary phylogenetic network in exactly two rNNI-moves.

Definition 41. *Let $\mathcal{N} \in \Omega(X)$. Let $u \in V(\mathcal{N})$ be a tree vertex and let $v \in V(\mathcal{N})$ be a reticulation vertex. Assume $(u, v) \in E(\mathcal{N})$ and there exists no other directed path from u to v. We say that $\mathcal{N}$ allows an* edge flip *on (u, v), which replaces the edge (u, v) by (v, u).*

We call a path $u - v$ that does not consist of only the edge (u, v) a *non-elementary path*. Note that the conditions imposed on the edge flip guarantee that the resulting network is in $\Omega(X)$.

Lemma 5. *Let $\mathcal{N} = (V, E) \in \Omega(X)$ such that it allows an edge flip on $(u, v) \in E$, let $\mathcal{N}' \in \Omega(X)$ be the network after applying an edge flip to (u, v). Then $\mathcal{N}$ can be transformed into $\mathcal{N}'$ in exactly two rNNI moves, except if $\mathcal{N} = \mathcal{N}'$.*

Proof. (Gambette et al., 2017) Let (u, v) be the edge being flipped in $\mathcal{N}$. First assume that the parent s of u and the parent $t \neq u$ of v are distinct vertices. Then we apply a type-(2) rNNI move $(su, uv, tv \to sv, uv, tu)$. This is an allowed move because if there were a $u - t$ path, there would be a non-elementary $u - v$ path in $\mathcal{N}$, which is not the case by the assumption that edge (u, v) can be flipped. Now we can apply a type-(2*) move $(sv, uv, tu \to su, vu, tv)$. This is allowed because no $u - s$ path can exist in $\mathcal{N}$, else (u, v) would have been an edge before the initial type-(2) rNNI move, which is not possible as it would have been a parallel edge. Also, u is a tree-vertex since we assume (u, v) can be flipped. The net effect of these two moves is that edge (u, v) is reversed to (v, u), see Figure 13.

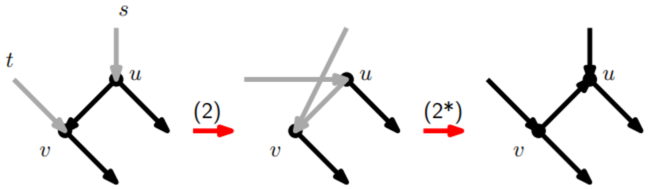

Figure 13: Reversing an edge (u, v) when u and v have different parents.

Now assume that u and v have a common parent p but the child $\hat{s} \neq v$ of u and the child $\hat{t}$ of v are distinct vertices. Then we apply a type-(1) rNNI move $(u\hat{s}, uv, v\hat{t} \to u\hat{t}, uv, v\hat{s})$. This is allowed because if there were an $\hat{s} - v$ path in $\mathcal{N}$, this path would need to pass through p, and hence imply the existence of a directed cycle in $\mathcal{N}$. Now we can apply a type-(1*) move $(u\hat{t}, uv, v\hat{s} \to u\hat{s}, vu, v\hat{s})$. This is allowed because no $\hat{t} - v$ path can exist in $\mathcal{N}$, else it would have been a parallel edge before the first type-(1) rNNI move. Also, v is a reticulation vertex, as it has indegree 2. The net effect of these two moves is that edge (u, v) is reversed to (v, u), see Figure 14.

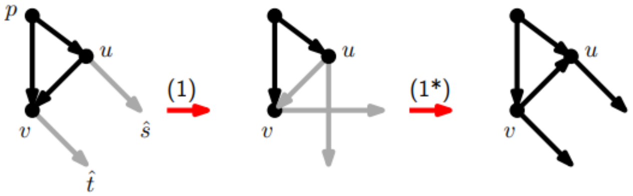

Figure 14: Reversing an edge (u, v) when u and v have a common parent but different children.

If we are in neither of the previous cases, u and v have a common parent p and a common child c. But then it is easy to see that that $\mathcal{N} = \mathcal{N}'$. ■

4.2.2 A conjecture of a strong upper bound

Using the result of Lemma 5, we believe that we can flip edges in a non-tree-based rooted binary phylogenetic network $\mathcal{N} \in \Omega(X)$ until we have $d(\mathcal{N})$ vertex disjoint paths partitioning the vertices of $\mathcal{N}$.

Conjecture 1. *Let $\mathcal{N} \in \Omega(X)$, then*

$$\delta_{rNNI}(\mathcal{N}) \leq 2p(\mathcal{N}).$$

We show using several illustrations why we believe this result to be true. First consider Figure 15.

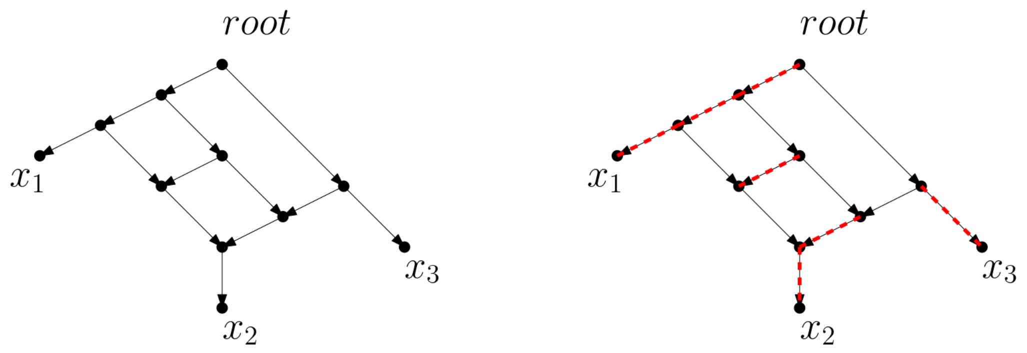

Figure 15: (left) A non-tree-based phylogenetic network $\mathcal{N}$ on taxa $\{x_1, x_2, x_3\}$. (right) A partitioning of the vertex set of $\mathcal{N}$ into four vertex disjoint paths.

Notice that $\mathcal{N}$ is non-tree-based since the parent of x_2 has two reticulation vertex parents. Since we can partition the vertices of $\mathcal{N}$ into four vertex disjoint paths ending in leaves, we see that $p(\mathcal{N}) = 1$. Now, consider the network $\mathcal{N}$ without directed edges in Figure 16. We try to construct $d(\mathcal{N})$ vertex disjoint paths ending in leaves.

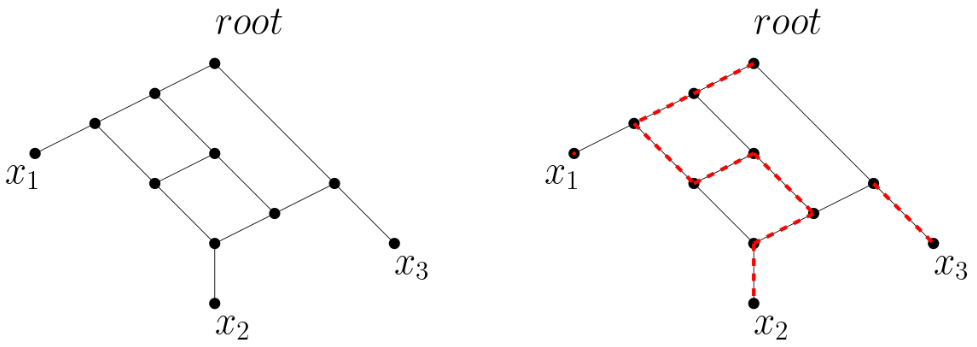

Figure 16: (left) The network $\mathcal{N}$ when replacing all directed edges with undirected edges. (right) A partitioning of the vertex set into three vertex disjoint paths.

Now, we take the paths constructed in the right graph of Figure 16 and use that path in the network of Figure 15 and we look for conflicts, i.e. edges that would not allow the path to form. Such conflicting edges are marked in red in Figure 17.

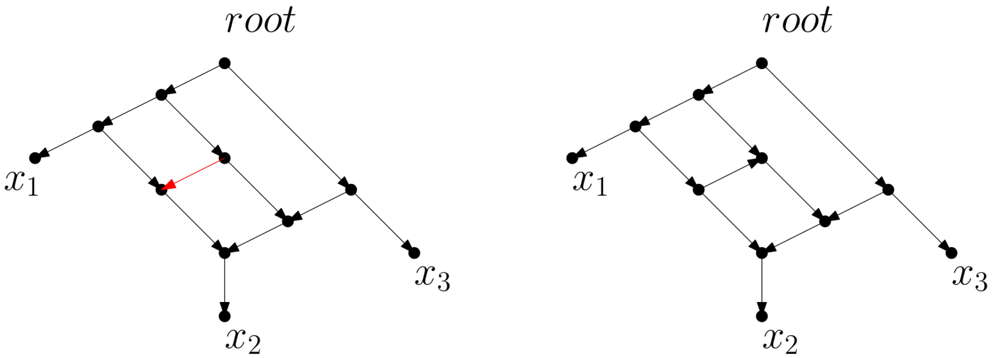

Figure 17: (left) The network $\mathcal{N}$, with conflicting edges marked in red. (right) The network after applying an edge flip to all edges marked in red. The resulting network $\mathcal{N}'$ is tree-based.

Using the path constructed in Figure 16, we see that the final network of Figure 17 is tree-based, since $p(\mathcal{N}') = 0$. Since we used one edge flip in the process to reduce $p(\mathcal{N})$ and one edge flip consists of two rNNI moves, we argue that $\delta_{rNNI}(\mathcal{N}) \leq 2p(\mathcal{N})$.

The problem in proving this upper bound for all $\mathcal{N} \in \Omega(X)$, is that the partitioning of the vertices of $\mathcal{N}$ into $d(\mathcal{N})$ vertex disjoint paths of Figure 16 cannot be arbitrary. We would need a set of paths that limit the number of conflicting edges and the conflicting edges should be allowed to sequentially get flipped. Moreover, we have to argue that the resulting network is then always tree-based. Our knowledge on the combinatorial properties of phylogenetic networks is too limited to prove Conjecture 1.

4.2.3 The proof of a weaker upper bound

In this section we prove a weaker upper bound for δ_{rNNI} than that of Conjecture 1.

Proposition 13. *Let $\mathcal{N} \in \Omega(X)$ with $n = V(\mathcal{N})$, then*

$$\delta_{rNNI}(\mathcal{N}) \leq \left\lfloor \frac{n}{2} \right\rfloor p(\mathcal{N}).$$

Proof. Assume $\mathcal{N} \in TBN(X)$. Then clearly $\delta_{\text{rNNI}}(\mathcal{N}) \leq \lfloor \frac{n}{2} \rfloor p(\mathcal{N})$.
Assume $\mathcal{N} \notin TBN(X)$. For all $x \in X$, define a connected subgraph H_x of $\mathcal{N}$ having a maximum number of vertices such that there exists a directed path from v to x for all $v \in V(H_x)$. Let G_x be defined as $H_x \setminus \{H_y : y \in X \setminus \{x\}\}$. Then, for $x, y \in X$, $G_x \cap G_y = \emptyset$. Let $G = \bigcup_{x \in X} G_x$ and let $Y \subset V$ such that $y \in Y$ if either $(y, x) \in E(G)$ for $x \in X$ and G_x is not a directed path or $y \in X$ and G_y is a directed path. Then, $\mathcal{Y}$ is an antichain in G. Suppose $\mathcal{Y}$ contains a tree vertex u. This means, for $(u, v) \in E(G)$ with $u \in Y$, $v \in X$, there exists a vertex w with $(u, w) \in E(G)$. Then, by definition of G_v, there exists a directed path from w to v which leads to a contradiction because v is a leaf and $(u, v) \in E(G)$. Hence, all vertices in $\mathcal{Y}$ are either reticulation vertices or leaves. Observe, if no vertex in $\mathcal{Y}$ is a reticulation vertex, then G_x is a directed path for all $x \in X$. This means, by definition of G, that $\mathcal{N}$ is a tree with leaf set X and therefore tree-based, contradicting our assumption that $\mathcal{N} \notin TBN(X)$. Thus, there exists a reticulation vertex $v \in Y$ with parents u_1 and u_2. Replace $\mathcal{Y}$ by $\mathcal{Y} \setminus \{v\} \cup \{u_1, u_2\}$ until there exists no reticulation vertex $v \in Y$ such that $(v, w) \in E(G)$ for some $w \in X$. Observe that after all replacements $\mathcal{Y}$ is still an antichain in G.

Next, assume there still exists a reticulation vertex $v \in \mathcal{Y}$ with parents u_1 and u_2. For $i = 1, 2$, if there exists no path from u_i to a vertex in $\mathcal{Y} \setminus \{v\}$, then add u_i to $\mathcal{Y}$ and remove v from $\mathcal{Y}$. Repeat this argument until we are unable to remove any more elements from $\mathcal{Y}$. Clearly, $\mathcal{Y}$ is a maximal antichain in the subnetwork of G induced by all vertices processed in the construction of $\mathcal{Y}$. Now, suppose by contradiction there exists a parent u of a vertex $v \in \mathcal{Y}$ which is a reticulation vertex or has indegree and outdegree 1. Then, there exists no path from u to a vertex in $Y \setminus \{v\}$, contradicting the termination of our construction process for $\mathcal{Y}$. Therefore, the parents of vertices in $\mathcal{Y}$ are tree vertices. First, assume G_x has a root. Then, for $x \in X$, let $G_{x,\mathcal{Y}}$ be the subnetwork of G_x with the same root as G_x and $\mathcal{Y}$ as leaf set. If the directed path from the root of $G_{x,\mathcal{Y}}$ to $\mathcal{Y}$ is unique for all $y \in Y$, then $G_{x,\mathcal{Y}}$ is a binary tree. Hence, $\mathcal{Y}$ constrained to G_x is a maximal antichain in G_x. Otherwise there exists $y \in Y$ such that the directed path from the root of $G_{x,\mathcal{Y}}$ to y is not unique. Recursively reapply our construction of the graph G and vertexset $\mathcal{Y}$ for $G_{x,\mathcal{Y}}$ and $\mathcal{Y}$ constrained to G_x instead of $\mathcal{N}$ and X. Thus, we conclude that the final transformation of $\mathcal{Y}$ yields a maximal antichain of G. Now, if G_x has no root, consider instead $H_{x,\mathcal{Y}}$ as the subnetwork of H_x with Y as leaf set. If $H_{x,\mathcal{Y}}$ contains a rooted binary tree T up to edge subdivisons such that $V(G_x) \subseteq V(T)$, then $\mathcal{Y}$ is a maximal antichain in G_x. Otherwise follow the same recursion for $G_{x,\mathcal{Y}}$ as in the case where G_x has a root. Again, we can conclude that the final transformation of $\mathcal{Y}$ yields a maximal antichain of G.

Now, for all $x \in X$, consider G_x. Let $Y(G_x)$ denote the vertex set $\mathcal{Y}$ constrained to G_x. Since $\mathcal{N}$ is not tree-based there exists at least one $x \in X$ such that $Y(G_x)$ contains at least two elements. Choose $v_1 \in Y(G_x)$ at maximum distance from x among all elements in $Y(G_x)$. By construction, v_1 has at least one parent vertex u and all parents of v_1 are tree vertices. Let $(u, v_2) \in E(G_x)$, $v_2 \neq v_1$. Suppose by contradiction that for every choice for v_1 we have $v_2 \notin Y(G_x)$. Then, there exists a path from v_2 to vertices $w_1, \ldots, w_p \in Y(G_x)$, $p \geq 2$. Otherwise we can remove w_1 from $Y(G_x)$ and add v_2 to $Y(G_x)$ without violating the antichain property of $Y(G_x)$, contradicting our choice of v_1. Recursively, consider $w_1, \ldots, w_p$ instead of v_1 until reaching a contradiction. This recursion terminates because $Y(G_x)$ is a maximal antichain. Thus, we can always find two vertices $v_1, v_2 \in Y(G_x)$ with a shared parent u. Let D_1 and D_2 be the shortest directed paths starting in v_1 and v_2, respectively, and ending in x such that $|D_1 \cap D_2|$ is minimum. Assume $D_2 \setminus D_1$ is a shorter path than $D_1 \setminus D_2$ and replace D_2 by $D_2 \setminus D_1$. Append vertex u to path D_1 to obtain path D_1'. We show that G_x can be transformed into a directed graph G_x' (which is a rooted binary phylogenetic network up to edge subdivisions) in $\kappa = |V(D_2 \setminus D_1)|$ many rNNI moves such that a maximum antichain of G_x' has strictly smaller cardinality than a maximum antichain in G_x.

First, consider the edge (u, v_2) and apply rNNI move (3*) by attaching the parent of u (u is a tree vertex) to v_2, the child of v_2 in D_2 to u and replacing (u, v_2) by (v_2, u). Then, replace D_2 by $D_2 \setminus \{v_2\}$, replace D_1' by $D_1 \cup \{v_2\}$ and observe that u is the parent of the first vertex of paths D_1 and D_2. Next, process the first vertex b_1 in D_2.

Case 1: b_1 has at least one parent and no children in D_1'. Among them, choose the parent a_1 which appears second in D_1' and apply rNNI move (3*) to edge (a_1, b_1) by reversing its direction and reattaching the child s of b_1 and the parent t of a_1.

Case 2: b_1 is a tree vertex with one parent a_1 and one child c_1 in D_1' such that a_1 appears before c_1 in D_1'.

Case 2.1: $(a_1, c_1) \notin E(D_1')$. Then, apply rNNI move (3*) to edge (a_1, b_1) by reversing its direction and reattaching c_1 and the parent of t of a_1.

Case 2.2: $(a_1, c_1) \in E(D_1')$. Then, apply the same rNNI move as in Case 1 by choose the child of b_1 different from c_1.

Case 3: b_1 has at least one child and no parents in D_1'. By construction, this case can not occur, even after applying Cases 1 or 2 any number of times.

Accordingly, we replace D_1' by concatenating the subpath of D_1' from u to t, (t, b_1), (b_1, a_1) and the subpath of D_1' from a_1 to x. Moreover, we replace D_2 by $D_2 \setminus \{b_1\}$. Clearly, $V(D_1') \cup V(D_2)$ partition the same vertexset as in the original graph G_x and both D_1' and D_2 are directed paths while D_1' ends in x. Repeatedly process the first vertex b_1 of D_2 until $D_2 = \emptyset$. This is possible because the size of D_2 strictly decreases. Thus, we arrive at a graph G_x' which only has x as a leaf and the vertexset $V(D_1') \cup V(D_2) = V(D_1')$ induces a directed path from u (the parent of two elements of a maximum antichain in G_x to x. This means, a maximum antichain of G_x' has strictly smaller cardinality than a maximum antichain in G_x and we constructed G_x' from G_x in κ rNNI moves.

Now, we can replace G_x in G by G'_x to obtain a directed graph G' in which every connected component is a rooted binary phylogenetic network up to edge subdivisions. G' differs from G by κ rNNI moves and G' has a strictly smaller maximum antichain than G. In the construction of G' the order of vertices in the intermediate directed paths D'_1 and D_2 who are adjacent to vertices outside of $V(D'_1) \cup V(D_2)$ never changed. Hence, adding all edges that were removed from $\mathcal{N}$ to construct G back to G' we obtain a rooted binary phylogenetic network $\mathcal{N}'$ on taxa X.

Finally, we check if $\mathcal{N}'$ is tree-based. If not, then we restart our whole procedure for $\mathcal{N}'$ instead of $\mathcal{N}$. For each new rooted binary phylogenetic network we construct on taxa X the cardinality of the maximum antichain decreases and we require κ rNNI moves. Clearly $\kappa \leq \lfloor n/2 \rfloor$. Moreover, the cardinality of the maximum antichain in a rooted binary tree-based phylogeneitc network is $|X|$. Thus, our claim follows. ■

4.2.4 A first illustration of the upper bound construction

We dedicate this section to illustrate how a non-tree-based rooted binary phylogenetic network can be transformed into a tree-based one using rNNI moves, as the proof of Proposition 13 describes. First we define a rooted binary phylogenetic network $\mathcal{N}$ on taxa X. We then create the subgraphs H_x. This is illustrated in Figure 18.

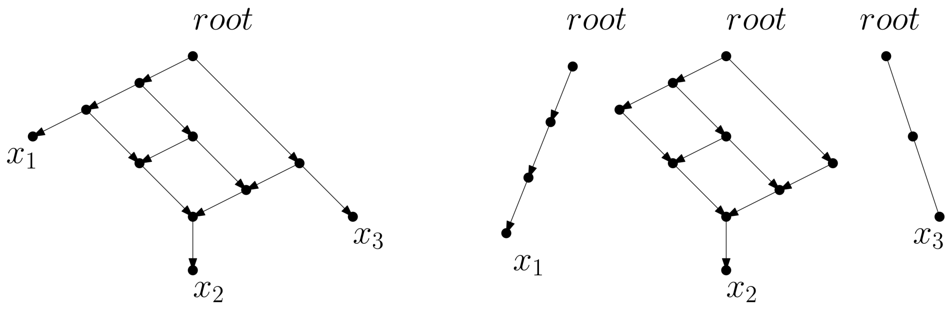

Figure 18: (left) A non-tree-based rooted binary phylogenetic network $\mathcal{N}$ on taxa $X = \{x_1, x_2, x_3\}$. (right) Subgraphs H_x, $x \in X$.

Notice that $\mathcal{N}$ is non-tree-based because the parent vertex of x_2 is a reticulation vertex with two reticulation vertex parents. We now turn H_x into G_x, for all $x \in X$, resulting in the graph in Figure 19. We do this by removing overlapping vertices and removing the edges of connected to removed vertices. We also define the set Y for G_x.

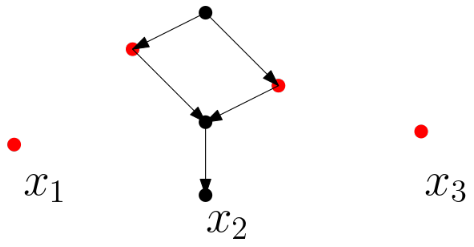

Figure 19: Subgraphs G_x, $x \in X$. Simultaneously, the entire graph is the graph G by definition. The elements of Y are denoted by the red-colored vertices. Notice that the red-colored vertices in G_{x_2} are reticulation vertices in $\mathcal{N}$. Also notice that G_{x_1} and G_{x_2} form directed paths.

After this, we take any reticulation vertex $v \in \mathcal{Y}$ and we replace $\mathcal{Y}$ by $\mathcal{Y} \setminus \{v\} \cup \{u_1, u_2\}$ until there exists no $v \in \mathcal{Y}$ such that $(v, w) \in E(G)$ for some $w \in X$. In this example, we do not need to replace $\mathcal{Y}$. For all $x \in X$ we construct $G_{x,\mathcal{Y}}$. These graphs have $\mathcal{Y}$ constrained to G_x as leaf set and prove a maximum antichain when forming binary trees. It can be seen from Figure 20 that this is indeed the case for all $x \in X$

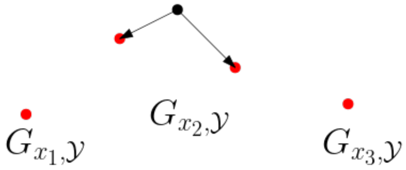

Figure 20: (left) G with recursively defined $\mathcal{Y}$ marked in red. (right) The graphs $G_{x,\mathcal{Y}}$.

We then consider G_x again. $\mathcal{Y}(G_{x_2})$ contains two elements, so we choose $v_1 \in \mathcal{Y}(G_{x_2})$ at maximum distance from x, which has a shared parent u with the other vertex in $\mathcal{Y}(G_{x_2})$. We then construct paths D_1' and D_2, as illustrated in Figure 21.

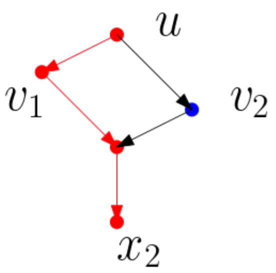

Figure 21: The graph G_{x_2} with paths D_1' marked in red, D_2 marked in blue.

We can now start applying rNNI-moves. We first apply rNNI move (3*) to (u, v_2), as illustrated in Figure 22. We also replace D_2 by $D_2 \setminus \{v_2\}$ and D_1' by $D_1' \cup \{v_2\}$.

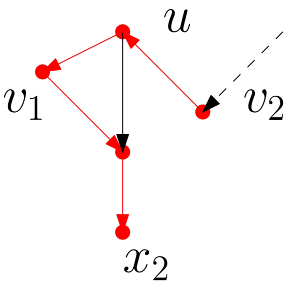

Figure 22: The graph G_{x_2} after applying rNNI move (3*) to (u, v_2), resulting in G_{x_2}'. The dotted line represents an outgoing edge from a vertex not in G_{x_2}.

Now, D_2 is empty and D_1' forms a directed path from u to x. Replacing G_{x_2} in G by G_{x_2}' yields a directed graph G' in which every connected component is a rooted binary phylogenetic network up to edge subdivisions. We define $\mathcal{N}'$ by reconstructing $\mathcal{N}$ using G', resulting in the network of Figure 23.

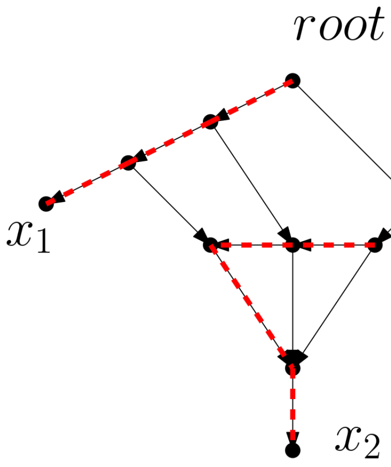

Figure 23: The resulting network $\mathcal{N}'$ after merging back G'. It is tree-based, since we can form three vertex disjoint paths (marked in red), partitioning the vertices of $\mathcal{N}'$.

4.2.5 Further illustrations of the upper bound construction

In the previous example, some steps included in the proof of Proposition 13 were able to be skipped. We dedicate this section to showing some edge cases of the method such that these steps are not skipped.

Let us first consider the case where we need to reapply the construction of G_x. In Figure 20, the directed paths from the root of $G_{x_2,\mathcal{Y}}$ to $\mathcal{Y}(G_{x_2})$ are unique. In Figure 24, one can see an example of a similar network, where the directed paths from the root of $G_{x_2,\mathcal{Y}}$ to $\mathcal{Y}(G_{x_2})$ are not unique.

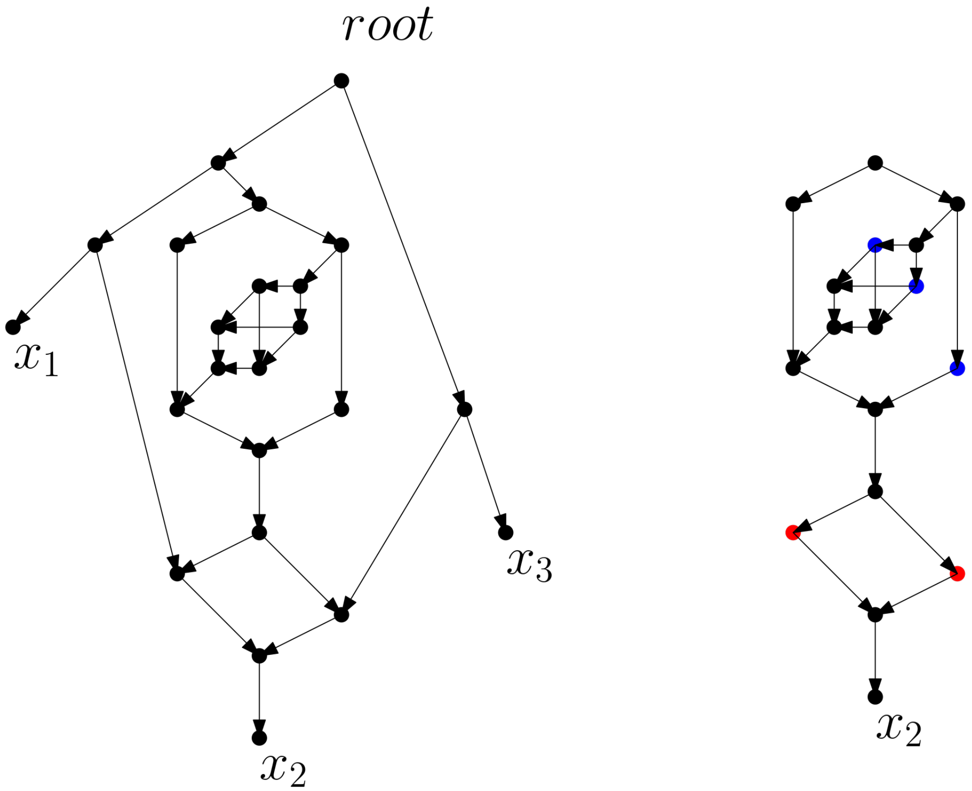

Figure 24: (left) A non-tree-based phylogenetic network. Similar to that of Figure 18. (right) The graph G_{x_2}. Notice that all directed paths from the root of G_{x_2} to $y \in \mathcal{Y}(G_{x_2})$, marked red, are not unique. Moreover, the vertices marked red are not a maximal antichain in G_{x_2}. The vertices marked in blue are.

Hence, it is required to recursively reapply the construction of G_x and $\mathcal{Y}$ for $G_{x,\mathcal{Y}}$ and $Y(G_x)$ instead of $\mathcal{N}$ and X. This way, the blue vertices in Figure 24 are fetched. These cases specifically occur whenever G_x is ’bottle-shaped’, like in Figure 24.

Now, we consider an example where we have to recursively replace $\mathcal{Y}$ by $\mathcal{Y} \setminus \{v\} \cup \{u_1, u_2\}$ and have to consider $H_{x,\mathcal{Y}}$. Consider Figure 25 for a phylogenetic network where this is the case.

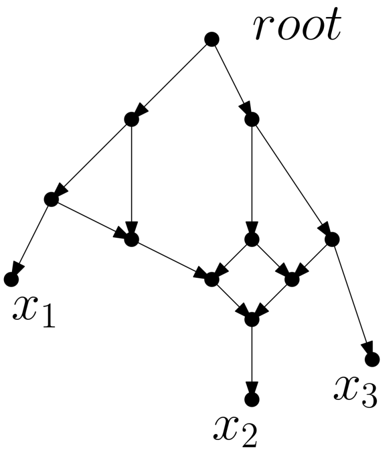

Figure 25: Non-tree-based rooted binary phylogenetic network $\mathcal{N}$ on taxa $\{x_1, x_2, x_2\}$.

Consider Figure 26 for construction of G_{x_2} and $\mathcal{Y}(G_{x_2})$.

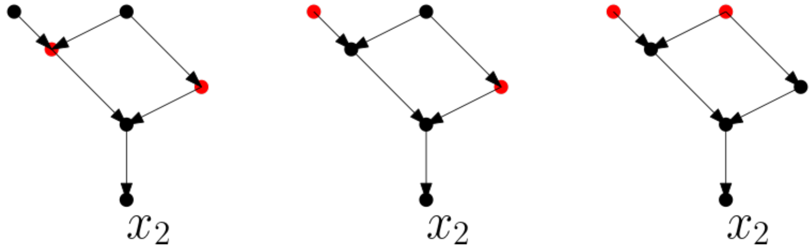

Figure 26: (left) Initial configuration of $\mathcal{Y}$ (constrained to G_{x_2}) marked in red. The parent of x_2 is a tree vertex, $\mathcal{Y}$ is set to include the parents of this tree vertex. There exists a vertex $v \in \mathcal{Y}$ with parent u_1 such that there exists no path from u_1 to a vertex in $\mathcal{Y} \setminus \{v\}$. (middle) We replace $\mathcal{Y}$ by $\mathcal{Y} \cup \{u_1\} \setminus \{v\}$. Again, there exists $v \in \mathcal{Y}$ with parent u_1 such that there exists no path from u_1 to a vertex in $\mathcal{Y} \setminus \{v\}$. (right) We replace $\mathcal{Y}$ by $\mathcal{Y} \cup \{u_1\} \setminus \{v\}$.

Since G_x has two vertices that can be considered as root, we take $H_{x,\mathcal{Y}}$ as subnetwork of H_x with $\mathcal{Y}$ as leaf set. This can be seen in Figure 27.

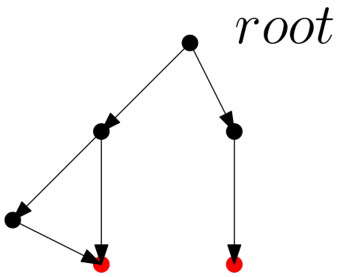

Figure 27: The graph $H_{x_2,\mathcal{Y}}$.

$H_{x_2,\mathcal{Y}}$ is a binary tree up to one edge, which is parallel up to one edge subdivision, hence we could continue with the method by considering G_x again.

Lastly, we consider an example where paths D_1 and D_2 are longer than the example in Figure 21, subsequently requiring more rNNI-moves until D_2 is empty. Consider Figure 28, which is a subgraph of some phylogenetic network where we have already taken the courtesy to form paths D_1', D_2.

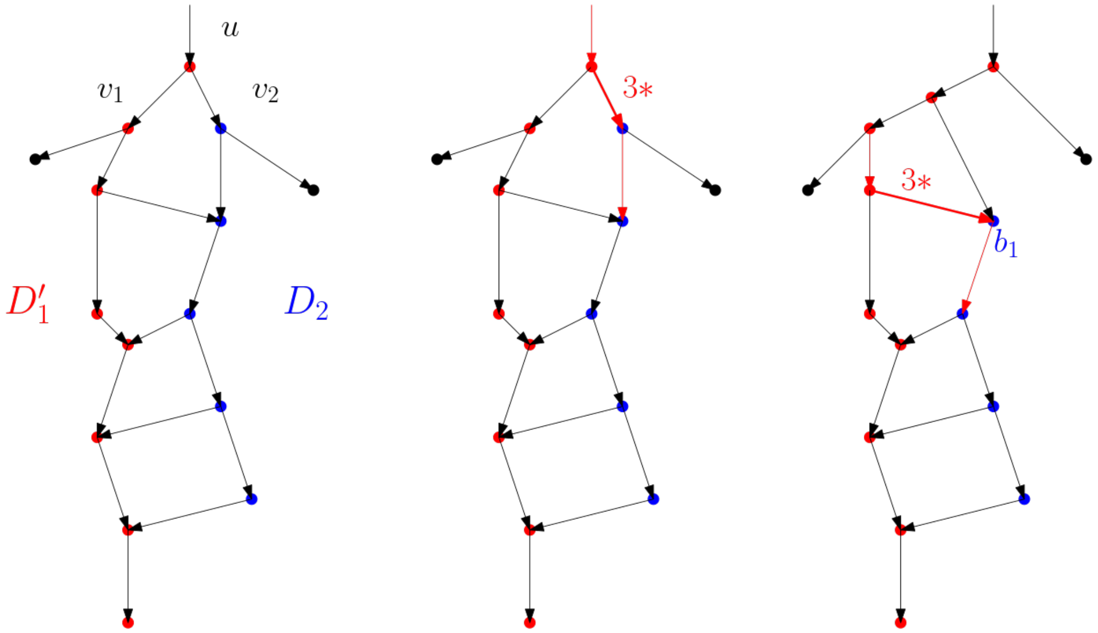

Figure 28: (left) Some graph G_x with paths D_1', D_2 colored red and blue respectively. (middle) We apply rNNI move (3*) on the edge marked red and bold, resulting in the graph on the right. (right) Vertex b_1 has one parent in D_1' and no children in D_1'. Hence we are in case 1 and apply rNNI move (3*) on the edge marked red and bold. We replace D_2, D_1' accordingly.

We continue in Figure 29.

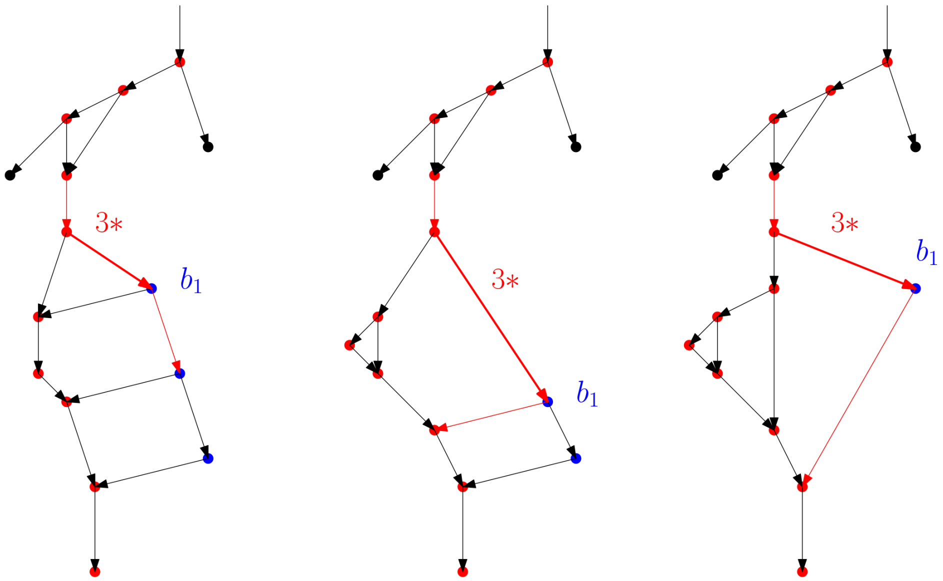

Figure 29: (left) Vertex b_1 is a tree vertex, hence we are in case 2. The parent of b_1 has an edge connected to the child of b_1 in D_1', hence we are in case 2.2 and apply rNNI move (3*) to the edge marked in red and bold. (middle) Similarly, we are in case 2.2 here and apply rNNI move (3*) to the edge marked in red and bold. (right) Looking at b_1, we are in case 1 and apply rNNI move (3*) marked in red and bold.

All in all, the sequence of rNNI moves results in the graph of Figure 30.

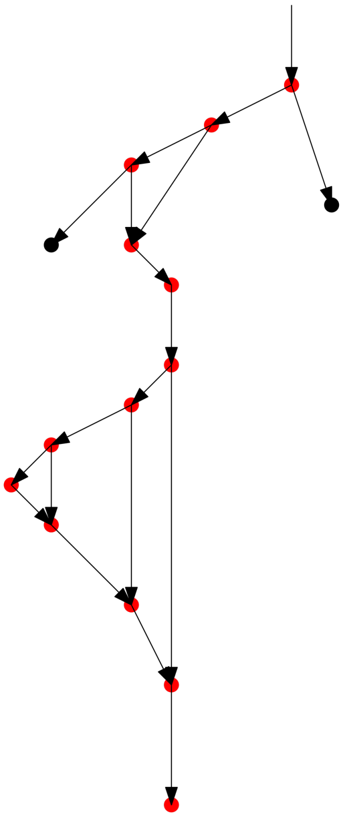

Figure 30: Graph G_x after applying rNNI moves until D_2 is empty. The order of vertices in the intermediate directed paths D_1' and D_2 who are adjacent to vertices outside of $V(D_1') \cup V(D_2)$ never changed. Hence this subgraph can be merged back into its original network.

4.3 Complexity of the rNNI proximity measure

Since we have not provided an explicit calculation for δ_{rNNI}, we cannot argue for its complexity. However, to get a lead on the complexity, we can consider results from Janssen (2021). These results state that finding the minimum number of rNNI moves to transform one arbitrary rooted binary phylogenetic network into another is conjectured to be NP-hard. Since we are not interested in finding this minimum number of moves between two networks, but between two classes of networks, we cannot use this result to conclude that an explicit calculation of δ_{rNNI} is NP-hard. However, the result may be useful in further research.

What can definitely be argued for, is that the transformation of a non-tree-based rooted binary phylogenetic network into a tree-based one using only rNNI moves in Proposition 13 can be done in polynomial time. Clearly, no steps in this transformation are of non-polynomial time complexity.

5 Significance

In this section we will reflect on the mathematical and biological significance of what we have discussed thus far using the content of Kong et al. (2022). In the next subsections, we begin with a refresher on phylogenetic networks and its origins from a more biologic perspective. Next, we will discuss the significance of different classes of phylogenetic networks, e.g. tree-based networks, as well as how to explore a network space of phylogenetic networks and its significance, e.g. with rNNI moves. We finish with a reflection of these applications in combination with the results from this thesis.

5.1 Origins from biology

Reconstructing and analyzing the evolutionary relationships among organisms is a central goal in evolutionary biology. Traditionally, rooted phylogenetic trees have been used to represent the evolutionary history for a set of species. Here, the leaves, or X in this thesis, represent the sampled extant taxa. The root represents the most recent common ancestor of all taxa X. All other vertices (tree vertices) represent speciation events. One can easily verify that in the case of a rooted phylogenetic tree, so a rooted phylogenetic network with just speciation events, every pair of leaves in X has a unique most recent common ancestor. In particular, rooted phylogenetic trees assume vertical inheritance, where genomic material is transmitted from an ancestral species to a descendant species. However, nowadays it is widely accepted that organisms do not always evolve by just vertical inheritance; many organisms experience horizontal inheritance as well. In biology, such events include hybridization: The process in which two complementary single-stranded DNA and/or RNA molecules bond together to form a double-stranded molecule (*Hybridization*, 2023).
Such events also include lateral gene transfer, or LGT: An all-encompassing term for the movement of DNA between diverse organisms (Sieber et al., 2017).
And finally such events also include recombination: A process by which pieces of DNA are broken and recombined to produce new combinations of alleles. This recombination process creates genetic diversity at the level of genes that reflects differences in the DNA sequences of different organisms (*recombination*, 2014). Because phylogenetic trees are not adequate to represent nontreelike evolutionary histories such as those described above, rooted phylogenetic networks have been proposed as a generalization of rooted phylogenetic trees in the literature (Kong et al., 2022).

5.2 Classes of phylogenetic networks

Certain various classes of rooted binary phylogenetic networks have been linked to evolutionary processes in literature and are thought to be biologically significant. Though only mentioned once in this thesis, the class of temporal or time-consistent networks are such example of a biologically significant class. Temporal networks provide a framework to explore evolutionary processes and phylogenetic relationships. By integrating temporal information into phylogenetic analyses, researchers can track evolutionary changes, infer ancestral states and reconstruct the evolutionary history of species or genes. More specifically, for a hybdridization event to have occured the two species involved (along with the hybrid they formed) must have been extant at the same time, which is denoted by an equivalent time-labeling.

Another biologically significant class of phylogenetic networks is the one that has been the main topic of this thesis: tree-based networks. Tree-based networks were introduced by A. R. Francis & Steel (2015) as a way to approach the question of whether a phylogenetic network is merely a phylogenetic tree with some additional horizontal edges, or whether a phylogenetic network has little resemblance to a tree and the concept of an underlying tree should be discarded (Kong et al., 2022). In the field of phylogenetics, there is an ongoing debate on whether evolutionary processes are inherently tree-based, or network-based. This ongoing debate has implications for our understanding of biodiversity, species relationships, and the mechanisms driving evolutionary change. It highlights the need for more sophisticated models and analytical tools that can accommodate reticulation and network-like patterns. Classifying evolutionary processes as ’tree-based’ or ’not tree-based’ is a rather unfair binary classification of complex evolutionary networks. Here the concept of proximity measures is one way to quantify the notion of tree-basedness of a given rooted phylogenetic network. For biologists, having access to such quantitative methods, could be a useful tool in their analysis of evolutionary processes regarding tree-basedness. Especially when such results can be computed in polynomial time. Also, having access to a diverse number of proximity measures allow biologists to test their evolutionary processes on different metrics. In the process of developing these proximity measures, it may occur that we get a better understanding of evolutionary processes in general, like the result of Theorem 10 from A. Francis et al. (2018). We get that older results from mathematics, like Dilworth’s Theorem, can be linked to ongoing debates in biology, which is a very intriguing idea.

Note that with temporal networks and tree-based networks we are just scraping the surface of network classes. Kong et al. (2022) also discuss classes such as tree-child networks, normal and regular networks, tree-sibling networks, stack-free networks, LGT networks and species graphs, orchard or cherry-picking networks, galled trees, galled networks, and level-k networks. Each with their own biological and mathematical implications. Also note that this only just covers classes of phylogenetic networks in the rooted (!) case.

5.3 Exploring the network space

Another topic of this thesis has been the subtree transfer move rNNI. Generalised subtree transfer moves are called rearrangement moves. The rearrangement move rNNI is just one of an arsenal of moves for phylogenetic networks that allow us to explore the space of phylogenetic networks. To sketch a significant application of these moves we introduce a challenge in phylogenetics that has not been the topic of discussion in this thesis so far: reconstruction. we consider the task of estimating a phylogenetic network given data for a collection of taxa. This leads to two distinct challenges related to scalability. First, we must evaluate the fit of a specified network to a given data set under a chosen model or optimality criterion. Second, we must search the space of possible networks for those that are optimal under the selected model or criterion (Kong et al., 2022). Omitting the details of this first challenge, we get that for a given rooted phylogenetic network, we have some objective function that we need to optimize. Optimizing such network can be done using heuristics that make use of re-arrangement moves to traverse the space of phylogenetic networks. Roughly speaking, if we consider proximity to tree-basedness as an optimality criterion, then the process of turning an arbitrary phylogenetic network into a tree-based phylogenetic network using rNNI moves is a challenge in this thesis that is similar to the challenge posed here.

Extending on the topic of rearrangement moves, rearrangement moves are defined as moves that take one edge of a network and move one or both endpoints to other locations in the network. Several of such moves in the rooted cases include tail moves, head moves, rNNI moves and rSPR moves. Some of these moves can be seen as ’localized’ version of the others. This is visualized in Figure 31.

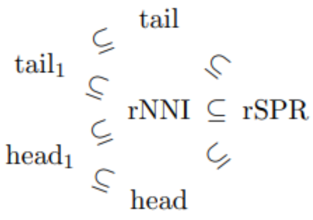

Figure 31: Diagram that shows some inclusions of move types. Figure from Janssen (2021)

As it turns out, finding the minimum number of rearrangement moves to move from one arbitrary rooted phylogenetic network to another is conjectured to be NP-hard (Janssen, 2021). Janssen (2021) has proven that this is NP-hard for some, but not all of the rearrangement moves. Such topics also give some mathematical significance to the topic of phylogenetic networks.

5.4 Connection to this thesis

In this thesis we have seen the likes of both classes of phylogenetic networks and rearrangement moves. While searching numerous amounts of papers, there were little instances where the intersection between rearrangement moves and network classes were explored. The only instance found was in Fischer & Francis (2020), which defined a proximity measure based on NNI-moves for unrooted phylogenetic networks. Since finding the minimum number of rearrangement moves between two arbitrary phylogenetic networks is conjectured to be NP-hard, finding the minimum number of moves between several network classes and researching whether it can be found in polynomial time would be an interesting topic for future research work. This thesis explored the number of rNNI moves required to go from a non-tree-based rooted phylogenetic network to a tree-based phylogenetic network. This work could be extended to different rearrangement moves and network classes.

6 Conclusion

In this section we will reflect on our findings and discuss the feasibility of some leads to future research.

6.1 Findings in this thesis

In this thesis we have done a thorough review of the current literature on rooted binary phylogenetic networks, characterisations for tree-basedness, proximity measures and rNNI-moves. In particular, we thoroughly discussed the matching characterisation, first introduced in Zhang (2016), and the the antichain characterisation, first introduced in A. Francis et al. (2018). We were able to distill the definition of a proximity measure as first introduced in A. Francis et al. (2018) and give it a more rigorous definition. With this knowledge at hand we were then able to define our own proximity measure δ_{rNNI}. We proved that δ_{rNNI} is a well-defined proximity measure. We also gave an upper bound for δ_{rNNI}, as shown in Proposition 13. Simultaneously, the proof of Proposition 13 provides a method to transform an arbitrary rooted binary phylogenetic network into a tree-based network with a quality guarantee. At the end of the thesis, we researched the applicability of the results in this thesis and the more general subject of phylogenetic networks. We found that this research is interesting for mathematical purposes, while it is unclear whether this also the case for biological purposes.

6.2 Topics of future research

In future research, one could look at the extension of this thesis' work to non-binary rooted phylogenetic networks, i.e. rooted phylogenetic networks where tree vertices and reticulation vertices are allowed to have summed in- and outdegree greater than 3. There appear some challenges when considering such extension. For one, the matching characterisation does not apply for non-binary rooted phylogenetic networks (Janisse, 2018). Furthermore it is unclear whether the antichain characterisation holds up in the non-binary generalization. A lot of proofs in this thesis depend on these characterisations. Another challenge in extending this thesis' work to the non-binary case is that rNNI is a defined for operations on edges in rooted binary phylogenetic networks. In the non-binary case, one would need to research how to extend the rNNI operation.

One could also look at the extension of this thesis' research to unrooted phylogenetic networks. In particular, one could research the δ_{NNI} proximity measure for unrooted phylogenetic networks, which appeared as an open question in Fischer & Francis (2020). The challenge with the extension to the unrooted case, is that unrooted phylogenetic networks are characterised in different ways than rooted phylogenetic networks; this is partially due to edges being undirected in unrooted phylogenetic networks.

Additionally, one could look at the rNNI reverse proximity measures and the rNNI novel measure in future research, respectively defined in Definition 39 and Definition 40. One could analyse the combinatorial properties of tree-based phylogenetic networks to make these networks non-tree-based using only rNNI moves. Analysing such reverse and novel measures would introduce quantitative tools to argue how 'strongly' tree-based an arbitrary rooted binary phylogenetic network is.

Lastly, one could look at an explicit calculation for δ_{rNNI}. One could start by researching the tightness of the upper bound in Proposition 13, by for example proving or disproving Conjecture 1. Explictly calculating δ_{rNNI} or researching Conjecture 1 requires more advanced knowledge on the combinatorial properties of phylogenetic networks. Additionally, when researching the complexity of a calculation for δ_{rNNI}, one could consider the results on the complexity of rearrangement moves in Janssen (2021).

References

Baroni, M., Semple, C., & Steel, M. (2006, 2). Hybrids in Real Time. *Systematic Biology*, *55*(1), 46–56.

Bondy, J., & Murty, U. (2008). *Graph theory* (Vol. 6; S. Axler & K. Ribet, Eds.) (No. 1). New York: Springer.

Dagan, T., & Martin, W. (2006). The tree of one percent. *Genome Biology*, *7*(118), 118.

Davidov, N., Hernandez, A., Jian, J., McKenna, P., Medlin, K. A., Mojumder, R., … Uraga, M. (2020, 9). Maximum Covering Subtrees for Phylogenetic Networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *18*(6), 2823–2827.

Dilworth, R. P. (1950, 1). A Decomposition Theorem for Partially Ordered Sets. *The Annals of Mathematics*, *51*(1), 161–166.

Fischer, M., & Francis, A. (2020, 9). How tree-based is my network? Proximity measures for unrooted phylogenetic networks. *Discrete Applied Mathematics*, *283*, 98–114.

Francis, A., Semple, C., & Steel, M. (2018). New characterisations of tree-based networks and proximity measures. *Advances in Applied Mathematics*, *93*.

Francis, A. R., & Steel, M. (2015, 9). Which Phylogenetic Networks are Merely Trees with Additional Arcs? *Systematic Biology*, *64*(5), 768–777.

Gallai, T., & Milgram, A. (1960). Verallgemeinerung eines graphentheoretischen satzes. *r ́edei. acta sci. math. (szeged)*, *21*, 181–186.

Gambette, P., van Iersel, L., Jones, M., Lafond, M., Pardi, F., & Scornavacca, C. (2017, 8). Rearrangement moves on rooted phylogenetic networks. *PLOS Computational Biology*, *13*(8), e1005611.

Hall, P. (1935, 1). On Representatives of Subsets. *Journal of the London Mathematical Society*, *s1-10*(1), 26–30.

Hopcroft, J. E., & Karp, R. M. (2006, 7). An $n^{5/2} $ Algorithm for Maximum Matchings in Bipartite Graphs. *SIAM Journal on Computing*, *2*(4), 225–231.

Hybridization. (2023, 6). Retrieved from `https://www.genome.gov/genetics-glossary/hybridization`

Janisse, F. (2018, 1). *Measuring how far a nonbinary phylogenetic network is from being tree-based* (Tech. Rep.). Delft: Delft University of Technology.

Janssen, R. (2021). *Rearranging Phylogenetic Networks* (Doctoral dissertation, Delft University of Technology, Delft). doi: https://doi.org/10.4233/uuid:1b713961-4e6d-4bb5-a7d0-37279084ee57

Jetten, L., & Van Iersel, L. (2018). Nonbinary Tree-Based Phylogenetic Networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *15*(1).

Kong, S., Kubatko, L., Wicke, K., & Pons, J. C. (2022). Mathematical Biology Classes of explicit phylogenetic networks and their biological and mathematical significance. *Journal of Mathematical Biology*, *84*, 47.

Kreyszig, E. (1989). *Introductory Functional Analysis with Applications* (Revised Edition ed.). WILA.

Pons, J. C., Semple, C., & Steel, M. (2019). Tree-based networks: characterisations, metrics, and support trees. *Journal of Mathematical Biology*, *78*(4).

recombination. (2014). Retrieved from `https://www.nature.com/scitable/definition/recombination-226/`

Sieber, K. B., Bromley, R. E., & Dunning Hotopp, J. C. (2017, 9). Lateral gene transfer between prokaryotes and eukaryotes. *Experimental cell research*, *358*(2), 421.

Zhang, L. (2016, 7). On Tree-Based Phylogenetic Networks. *Journal of Computational Biology*, *23*(7), 553–565.