The University of Auckland

Doctoral Thesis

# Spaces of phylogenetic networks

*Author:*
Jonathan Klawitter

*A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy*

*in the*

School of Computer Science

March, 2020

# Abstract

Rooted phylogenetic trees and networks are rooted, acyclic, leaf-labelled graphs that are used to model the inferred evolutionary history of taxa; for example species. While a phylogenetic tree models only bifurcating events, a phylogenetic network can also model reticulation events like hybridisation, recombination, and horizontal gene transfer.

A rearrangement operation transforms one phylogenetic tree into another via a local graph-based change. For example, the subtree prune and regraft (SPR) operation prunes (cuts) a subtree of a phylogenetic tree and then regrafts (attaches) it to an edge of the remaining tree, resulting in another phylogenetic tree. Another operation is nearest neighbour interchange (NNI), which is a special case of SPR where the pruned edge has to be regrafted closely to where it was pruned. The set of all phylogenetic trees for a fixed set of taxa together with a rearrangement operations forms a graph where the vertices are the trees and two trees are adjacent when one can be transformed into the other by applying the rearrangement operation exactly once. In such a space, the distance of two trees is given by the minimum number of operations needed to transform one into the other. The SPR-distance of two trees can be characterised with a maximum agreement forest; a forest with a minimum number of components that covers both trees.

In this thesis we study spaces of phylogenetic networks under generalisations of NNI and SPR, in particular, the subnet prune and regraft (SNPR) operation and the here introduced prune and regraft (PR) operation. First, we consider connectedness and diameters of spaces of different classes of phylogenetic networks. We then look at the size of the neighbourhood of a phylogenetic network. Furthermore, we investigate properties of shortest paths under SNPR and PR. This includes several bounds on the distances of two networks. Finally, we introduce maximum agreement graphs as a generalisation of maximum agreement forests for phylogenetic networks. We show that maximum agreement graphs induce a metric – the agreement distance – and study its relation to the SNPR- and PR-distance.

# Acknowledgements

I would like to thank Simone Linz, my primary supervisor, for her guidance, support, and advice; Mark C. Wilson, my co-supervisor, for his support and collaboration on OA; Charles Semple for acting as second co-supervisor; the anonymous referees of included papers for their detailed and helpful feedback; Jordan Douglas, Wai Loong Tham, and Monika Karmin for proofreading parts of the thesis; Tuan Chien and Monika Karmin for their emotional support through my ups and downs; and the administrative staff for all their help.

I am grateful for the financial support received from the New Zealand Marsden Grant.

# Contents

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND

**School of Graduate Studies**
AskAuckland Central
Alfred Nathan House
The University of Auckland
Tel: +64 9 373 7599 ext 81321
Email: postgradinfo@auckland.ac.nz

# Co-Authorship Form

This form is to accompany the submission of any PhD that contains published or unpublished co-authored work. **Please include one copy of this form for each co-authored work**. Completed forms should be included in all copies of your thesis submitted for examination and library deposit (including digital deposit), following your thesis Acknowledgements. Co-authored works may be included in a thesis if the candidate has written all or the majority of the text and had their contribution confirmed by all co-authors as not less than 65%.

| Please indicate the chapter/section/pages of this thesis that are extracted from a co-authored work and give the title and publication details or details of submission of the co-authored work.<br>Chapter 5 and Section 6.3.2 based on<br>"On the Subnet Prune and Regraft Distance", Electronic Journal of Combinatorics, vol. 22, no. 2, pp. 329-355, 2019. | |
|---|---|
| Nature of contribution by PhD candidate | conceptualisation, writing, visualisation |
| Extent of contribution by PhD candidate (%) | 70 |

## CO-AUTHORS

| Name | Nature of Contribution |
|---|---|
| Simone Linz | writing, supervision |
| | |
| | |
| | |
| | |
| | |

## Certification by Co-Authors

The undersigned hereby certify that:

- the above statement correctly reflects the nature and extent of the PhD candidate's contribution to this work, and the nature of the contribution of each of the co-authors; and
- that the candidate wrote all or the majority of the text.

| Name | Signature | Date |
|---|---|---|
| Simone Linz | S. Linz | 22/08/19 |
| | | |
| | | |
| | | |
| | | |
| | | |

# 1. Introduction

In mathematics, a *graph* is a structure that consists of a set of *vertices* and a set of *edges* that connect pairs of vertices. The edges of a graph may be directed and may have a weight or length. Graphs are a versatile tool to describe structures in our world. Loosely speaking, the vertices of a graph represent some entities and the edges their relationships.

In phylogenetics, a *phylogeny* describes the evolutionary histories and relationships of a set of *taxa*. In general, each *taxon* of such a set represents some species, population, or individual organism whose evolutionary history is of interest to us. For example, the phylogeny of present-day species may be their evolutionary species tree, and the phylogeny of individuals within a species, like humans, may represent their genetic lineages. Depending on the context, the set of taxa may be single genes, nucleotide sequences, chromosomes, or also words and languages [SS03, Dun14].

A *phylogenetic tree* is a graph used to model and visualise a phylogeny. In graph theory, it is a tree where the leaves are labelled with the taxa and that can be either rooted or unrooted. A *rooted phylogenetic tree* has a designated root vertex and the edges are directed from the root towards the leaves. Two rooted phylogenetic trees are shown in Figure 1.1 (a) and (b). Depending on the context, inner vertices of a rooted phylogenetic tree may be interpreted differently, for example, as bifurcation (or multifurcation) events, such as speciation or lineage splits, or as most recent common ancestors [SS03]. The edges of a phylogenetic tree may be equipped with lengths and the vertices thus with a height, like in Figure 1.1 (a), in order to represent the passing of time, distances between leaves, or the number of mutations along an edge. An *unrooted phylogenetic tree* has neither a root nor are its edges directed. An example is shown in Figure 1.1 (c). Compared to the rooted case, an unrooted phylogenetic tree illustrates the evolutionary relatedness instead of the history of the taxa.

A *phylogenetic network* is a generalisation of a phylogenetic tree in the sense that as a graph it is not necessarily a tree. More precisely, a *rooted phylogenetic network* may contain two types of inner vertices. On the one hand, it contains inner tree vertices that have one incoming edge but two or more outgoing edges. On the other hand, it may contain *reticulations* with two or more incoming edges. With this, a phylogenetic network can model the phylogeny of taxa whose past includes reticulation events such as hybridisation, horizontal gene transfer, recombination, or reassortment [HRS10]. Such reticulation events arise in all domains of life [TN05, RW07, TKR09, ALC+14, MMM+17, WWK+17]. A phylogenetic network may also combine a set of conflicting phylogenetic trees in a single graph [HRS10]. See Figure 1.2 (a) for an example of a rooted phylogenetic network. As

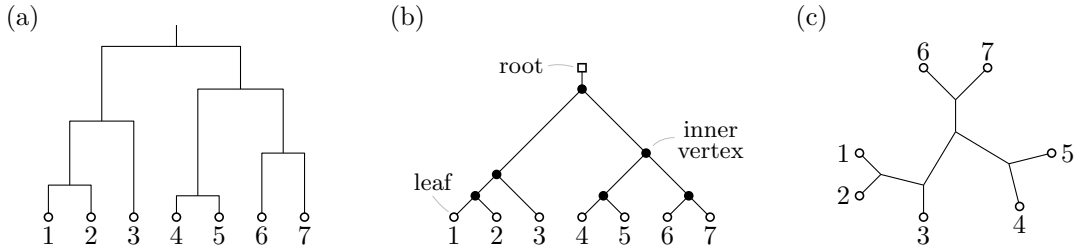


Figure 1.1: (a) A rooted phylogenetic tree with edge lengths and vertex heights (scale not shown). (b) A rooted phylogenetic tree drawn with a style that highlights the underlying graph, but where edge lengths bear no further meaning. (c) An unrooted phylogenetic tree with edge lengths (scale not shown).

with trees, an *unrooted phylogenetic network* is the counterpart of a rooted phylogenetic network without a determined root. An example is shown in Figure 1.2 (b). For completeness, we want to mention that the term "unrooted phylogenetic network" sometimes also encompass so-called split networks, median networks, and haplotype networks [HRS10]. Unless mentioned otherwise, the phylogenetic networks under consideration are assumed to have no edge lengths.

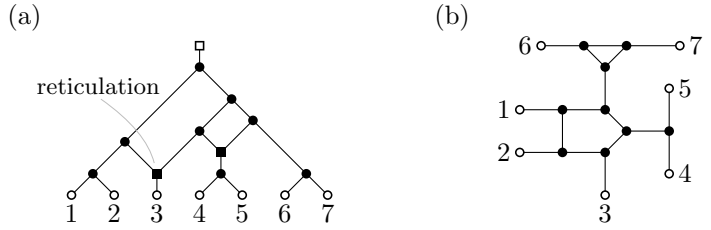


Figure 1.2: (a) A rooted phylogenetic network with two reticulation vertices. (b) An unrooted phylogenetic network.

A phylogeny is usually constructed with a phylogenetic inference method and a model of evolution based on available data of the taxa. The data could for example be DNA sequences or morphological characters of the considered species. Inference methods can be based on a variety of concepts such as distance-matrices, maximum parsimony, maximum likelihood, Bayesian inference like Markov chain Monte Carlo (MCMC), and others [Fel04, Gus14]. Here we are interested in the solution and search spaces of such inference methods.

A *space of phylogenetic networks* is a set of phylogenetic networks on the same set of taxa that share a certain property. The size of such a space grows super-exponentially in the number of taxa [MSW15]. For example, the space of rooted phylogenetic trees on fifteen taxa contains over 200 trillion different trees. Most inference methods operate by searching for networks that are "close" to the current network [Pag93, BHK$^+$14, RH03, GDL$^+$10, YBN13, YDLN14, WMI15]. However, this requires a metric on the space of phylogenetic networks that measures this notion of closeness or (dis)similarity of networks numerically and thus also imposes a structure on the space. Furthermore, by using a metric results obtained for different data or by different inference methods can be compared, for instance, to evaluate their robustness or to find outliers and clusters.

*Tree rearrangement operations* can be used to obtain metrics and structures for a space of phylogenetic trees. Such operations make small, graph-theoretical changes to a phylogenetic tree to obtain another phylogenetic tree. An example on rooted phylogenetic trees is *subtree prune and regraft (SPR)*, which, as the name suggests, prunes (cuts) a subtree

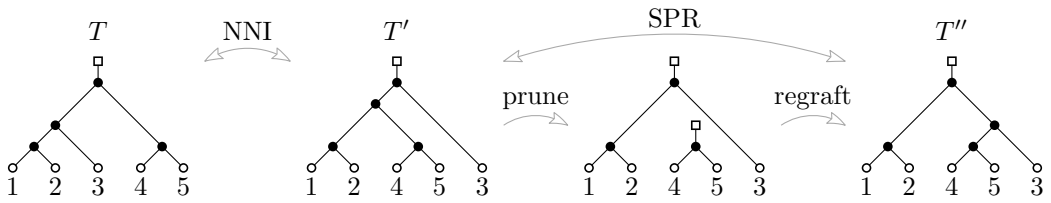


Figure 1.3: The tree rearrangement nearest neighbour interchange (NNI) swaps two branches incident to the same edge. The tree rearrangement operation subtree prune and regraft (SPR) first prunes (cuts) an edge and then regrafts (attaches) it to obtain a new tree.

and then regrafts (attaches) it again. This operation and *nearest neighbour interchange (NNI)*, which works more locally, are illustrated in Figure 1.3. For unrooted phylogenetic trees, there are the analogues of NNI and SPR as well as *tree bisection and reconnection (TBR)*, which deletes an edge and then reconnects the resulting two smaller trees with a new edge. A rearrangement operation turns a space of phylogenetic trees into a graph, where the phylogenetic trees are the vertices and where two vertices are adjacent if one can be transformed into the other by applying the rearrangement operation exactly once. For example, the space on rooted phylogenetic trees with four taxa under NNI is shown in Figure 1.4. The distance of two phylogenetic trees in this space is given by the minimum number of steps required to go from one to the other.

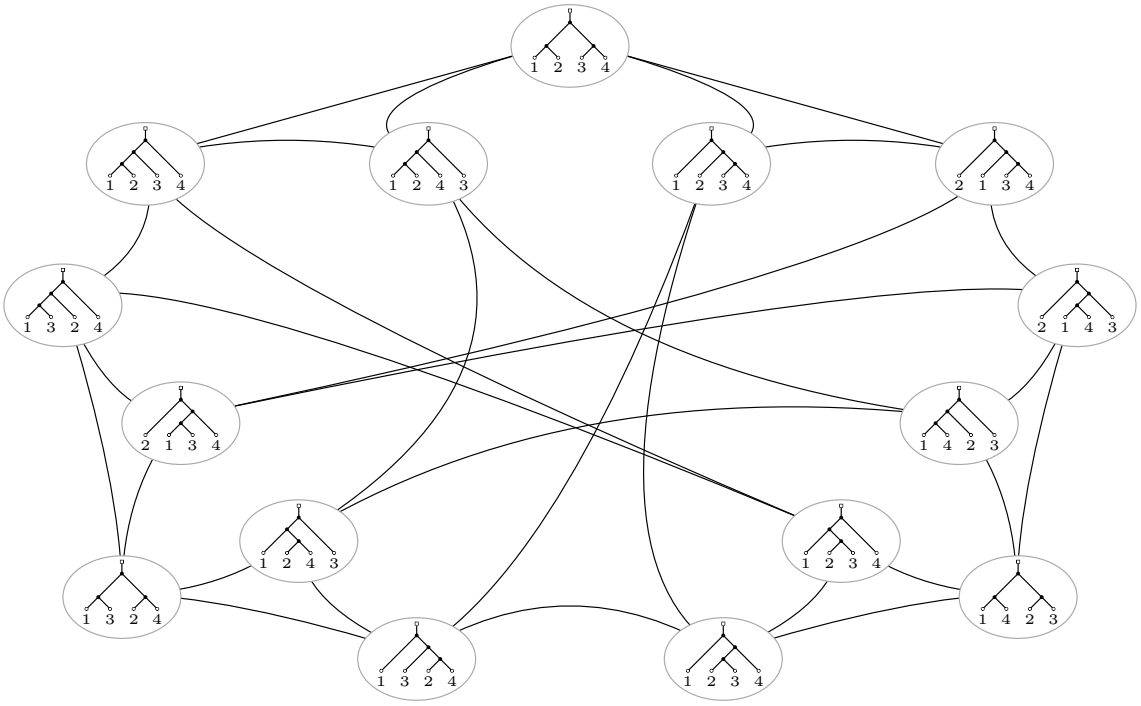

Figure 1.4: The space of rooted phylogenetic trees on four taxa under the NNI operation.

The spaces of phylogenetic trees under NNI, SPR, and TBR have been well studied (see for example St. Johns review [SJ17]). Most importantly, it has been established that the operations induce metrics [Rob71, SOW96]. Properties of these spaces that have been studied range from local properties such as the size and structure of neighbourhoods of a tree [AS01, Son03, HW13, dJMS16, CCLSJ13] to global properties such as the diameter (the maximum distance between two trees in a space) [LTZ96, DGH11] and curvature [WMI17].

The main objective regarding these spaces is the computation of the distance between two trees. However, it has been shown that it is NP-hard to compute the NNI-, SPR-, and TBR-distance [LTZ96,BS05,HDRCB08,AS01]. Nevertheless, there exists a variety of exact and fixed-parameter tractable algorithms [AS01, BS05, Wu09, vIKLS14, CFS15] as well as approximation algorithms [LTZ96, BSJMA06, BMS08, BSJ09, WBZ13, vIKLS14, CFS15].

*Maximum agreement forests* are a major tool to deal with the SPR- and TBR-distance. Roughly speaking, a maximum agreement forest of two rooted or unrooted phylogenetic trees consists of the subtrees that stay unchanged by a shortest sequence of SPR or TBR operations, respectively, from one tree to the other. In other words, an agreement forest is a set of trees on which the two trees "agree" upon and that, if put together, cover each tree. See Figure 1.5 for an example. The usefulness of maximum agreement forests stems from the fact that they characterise the SPR-distance of two rooted phylogenetic trees and the TBR-distance of two unrooted phylogenetic trees; that is, their size of a maximum agreement forests determines the distance [AS01, BS05]. All algorithms referenced above on the SPR- and TBR-distance are either based on agreement forests or agreement forests are used in the analysis of their correctness or approximation ratios.

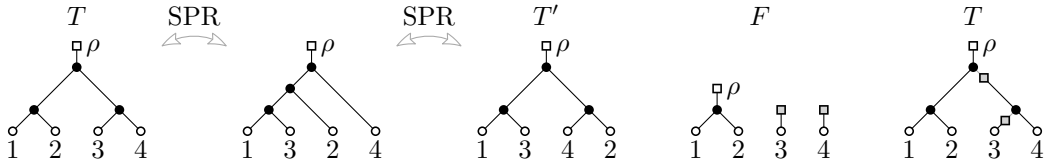


Figure 1.5: An SPR-sequence of length two that transforms $T$ into $T'$. This transformation yields the agreement forest $F$, which in turn can be put together to obtain $T$ again (right side).

In recent years, more and more focus has been cast on phylogenetic networks [HRS10]. Since the definition of a phylogenetic network is quite broad, a panoply of classes of phylogenetic networks with certain structural properties have been defined [GEL03,Bar04, BSS06, CLRV08, CRV09, Wil10, vIK11, CPR15, FS15, ESS19]. Some of these properties are based on biological considerations, while others serve the purpose of limiting the complexity of networks. An example for the former kind are *tree-child networks* where there is path from every inner vertex to a leaf that does not pass through a reticulation [CRV09]. The study of phylogenetic networks includes their relation to phylogenetic trees [GBP09, LSJS13, GGL$^+$15, HMSW16, HL18], how to reconstruct them [HS10, EOZN19], distance functions on them [HRS10, CLRV08, CLRV09a, CLRV09b, CLRV09c], and others.

It is of interest to us that the three tree rearrangement operations NNI, SPR, and TBR have been generalised to network rearrangement operations [HLMW16, BLS17, GvIJ$^+$17, FHMW18]. Consequently, the study of spaces of phylogenetic networks under rearrangement operations has begun and this is where this thesis starts.

## 1.1 Thesis outline

The goal of this thesis is to look at previously done work on spaces of rooted phylogenetic trees under rearrangement operations and see how it can be lifted to spaces of phylogenetic networks. We consider the similarities and differences of these spaces and how they relate to each other. For example, knowing that agreement forests characterise the SPR-distance of rooted phylogenetic trees, we want to find out whether they can be generalised to networks and how they relate to network rearrangement operations. We limit ourselves to rooted, binary phylogenetic networks without edge lengths, where binary means that

inner vertices have degree three. The following is a brief summary of the results obtained in the individual chapters.

### Chapter 2 – Preliminaries

This chapter introduces the necessary background from graph theory and combinatorial phylogenetics. We define phylogenetic trees and phylogenetic networks. Moreover, we look at several classes of phylogenetic networks and their relations. We show how the classic tree rearrangement operations are generalised to network rearrangement operations. In particular, we define nearest neighbour interchange (NNI), subnet prune and regraft (SNPR), and introduce the new prune and regraft (PR) operation on networks. Furthermore, we make it precise how a rearrangement operation induces a distance on a space of phylogenetic networks.

### Chapter 3 – Connectedness

We investigate the connectedness of different classes of phylogenetic networks under rearrangement operations. More precisely, we ask whether each network of a space can be reached from any other network under the considered operation. As a result, we get that SNPR and PR form metric spaces with most, but not all, classes of phylogenetic networks. For SNPR, we find that spaces of networks with the maximum number of reticulations in a class are often not connected. For each space, we either show that the diameter of a space is unbounded or give an asymptotic bound.

### Chapter 4 – Neighbourhoods

Two phylogenetic networks are neighbours with respect to a rearrangement operation if they can be transformed into each other by applying this operation exactly once. The neighbourhood of a network is the set of all its neighbours. For example, in Figure 1.4 every tree has exactly four neighbours under NNI. In this chapter we look at the size of the neighbourhood of a network. We give exact expressions and bounds for the neighbourhood size for networks of several classes under NNI and SNPR. Furthermore, we give bounds on minimum and maximum neighbourhood sizes of networks.

### Chapter 5 – Shortest paths

Next, we look at what general statements we can make about the SNPR- and PR-distance of two networks in a spaces of phylogenetic network. For this we consider shortest paths between these networks. We start with the distance of a tree $T$ and a network $N$. Finding that shortest paths from $T$ to $N$ can behave nicely, we show that this distance can be characterised by the set of trees that are embedded in $N$. We use this result to obtain a fixed-parameter tractable algorithm to compute the distance of $T$ and $N$. Furthermore, we consider the properties of shortest paths between two networks $N$ and $N'$. We find that these paths can behave, in some sense, unfortunately for search algorithms. Furthermore, we show that most classes of networks are not isometric subgraphs of the class of all phylogenetic networks under SNPR and PR; i.e. the distance of $N$ and $N'$ in that particular class and in the general class differ.

### Chapter 6 – Agreement graph and distance

A maximum agreement forests (MAF) of two phylogenetic trees $T$ and $T'$ is a forest with the minimum number of components that covers both $T$ and $T'$. Since MAFs characterise

the SPR-distance, they have been a major tool in the development of approximation and fixed-parameter tractable algorithms to compute this distance. Here we generalise this notion to maximum agreement graphs for two phylogenetic networks $N$ and $N'$. With this, we introduce a new metric on the class of all phylogenetic networks – the agreement distance. We prove that the agreement distance equals the PR- and SNPR-distance of a tree and a network. However, in general it does not characterise the PR- or SNPR-distance. Nevertheless, we show that the agreement distance still bounds these two distances with constant factors.

# 2. Preliminaries

In this chapter we give a brief overview of the main concepts in graph theory and mathematical phylogenetics that are important in the context of this thesis. We assume, however, a basic knowledge of algorithmic and complexity-theoretic concepts (for example, Big O notation, NP-hardness, and fixed-parameter tractability). Introductions to these concepts are found in standard textbooks on algorithms and complexity theory, see, for example, Cormen et al. [CLRS09] and Garey and Johnson [GJ79]. For an introduction into fixed-parameter tractable algorithms, see Niedermeier [Nie06]. Diestel [Die17] provides an excellent introduction to graph theory.

## 2.1 Graphs

As described in Chapter 1, phylogenetic trees and networks are special types of graphs. In this section we define some basic concepts from graph theory.

### 2.1.1 Undirected graph

An *undirected graph* is an ordered pair $G = (V, E)$ comprised of a set of *vertices* $V$ together with a (multi)set of *edges* $E$, which are two-element subsets of $V$. Two vertices $u, v \in V$ are called *adjacent* if there exists an edge $e = \{u, v\} \in E$. In this case, $u$ and $v$ are *incident* to $e$ and vice versa. The *degree* of a vertex is the number of edges it is incident to. A *leaf* of an undirected graph is a vertex with degree one. An *isolated vertex* or *singleton* is a vertex with degree zero.

A *subgraph* $H = (V', E')$ of $G$ is an undirected graph such that $V' \subseteq V$ and $E' \subseteq E$. A *path* $P = (v_0, v_1, \ldots, v_k)$ in $G$ is a subgraph of $G$ such that the vertices $v_i$ are distinct and where $\{v_i, v_{i+1}\} \in E$, for $i \in \{0, \ldots, k-1\}$. The *length* of a path $P$ is $k$, which we require to be at least one. Note that the length of a path is also the number of edges in this path. A *cycle* $C = (v_0, v_1, \ldots, v_k)$ in $G$ is a subgraph such that the $v_i$'s are pairwise distinct and where $\{v_i, v_{i+1}\} \in E$, for $i \in \{0, \ldots, k-1\}$, and $\{v_k, v_0\} \in E$. A cycle on three edges is called a *triangle*.

An undirected graph is *connected* if there exists a path between any two of its vertices. Otherwise the graph is *disconnected*. The *diameter* $\mathrm{diam}(G)$ of a connected graph $G$ is the maximum length of a shortest path between any two vertices in $G$. A maximal connected subgraph of $G$ is a *component* of $G$. A connected graph is a *tree* if it does not contain a cycle. A graph consisting of a set of trees as components is a *forest*.

A *cut-vertex* of a connected graph is a vertex whose removal disconnects the graph. Analogously, a *cut-edge* of a connected graph is an edge whose removal disconnects the graph. A graph is *biconnected* if two vertices have to be removed to create a disconnected graph. A *biconnected component* is a maximal biconnected subgraph. Such a biconnected component is *nontrivial* if it contains at least two edges.

### 2.1.2 Directed graph

A *directed graph* is an ordered pair $G = (V, E)$ comprised of a set of *vertices* $V$ together with a (multi)set of *edges* $E$, which are ordered pairs of vertices of $V$. Two vertices $u, v \in V$ are *adjacent* if there exists an edge, say, $e = (u, v)$ in $E$. In this case, $e$ is an *outgoing* edge of $u$ and an *incoming* edge of $v$. Furthermore, $u$ is the *tail* of $e$ and $v$ is the *head* of $e$. Like for an undirected graph, $u$ and $v$ are considered *incident* to $e$ and vice versa. The *indegree* and *outdegree* of a vertex $v$ are the number of incoming and outgoing edges of $v$, respectively. The *degree* of $v$ is the sum of its indegree and outdegree. The graph $G$ is *rooted* if it contains exactly one vertex with indegree zero. A *leaf* of $G$ is a vertex with indegree one and outdegree zero.

Let $G = (V, E)$ be a directed graph. A *subgraph* $H = (V', E')$ of $G$ is a directed graph such that $V' \subseteq V$ and $E' \subseteq E$. A *path* $P = (v_0, v_1, \ldots, v_k)$ in $G$ is a subgraph of $G$ such that the $v_i$'s are distinct and where $(v_i, v_{i+1}) \in E$, $i \in \{0, \ldots, k-1\}$. We require again that $k$ is at least one. An edge $e = (u, v)$ of $G$ is a *transitive edge* if $G$ contains a path from $u$ to $v$ that does not contain $e$. A *cycle* $C = (v_0, v_1, \ldots, v_k)$ in $G$ is a subgraph such that the $v_i$'s are distinct and where $(v_i, v_{i+1}) \in E$, $i \in \{0, \ldots, k-1\}$, as well as $(v_k, v_0) \in E$. An *acyclic directed graph* is a directed graph that does not contain a cycle. The *underlying graph* of $G$ is the undirected graph $G' = (V, E')$ derived from $G$ by dropping the directions of the edges in $E$. An *underlying path* or *cycle* of $G$ is a path or cycle in its underlying graph $G'$. Unless mentioned otherwise, it is assumed that a path or a cycle in a directed graph is a (directed) path or cycle and not an underlying path or cycle. A *triangle* of $G$ is a triangle of its underlying graph $G'$. Connectedness, components, cut-vertices, cut-edges, trees, and forests of directed graphs are defined on their underlying graphs. A *pendant* subgraph of $G$ is a subgraph that can be separated from the rest of $G$ by removing a cut-edge.

Let $G$ be a directed acyclic graph. A degree-two vertex $v$ of $G$ with incoming edge $(u, v)$ and outgoing edge $(v, w)$ gets *suppressed* by deleting $v$ and the edges $(u, v)$ and $(v, w)$ and adding the edge $(u, w)$. A new vertex $v$ *subdivides* an edge $(u, w)$ of $G$ by deleting the edge $(u, w)$ and adding the edges $(u, v)$ and $(v, w)$. In general, *subdividing* an edge $(u, v)$ means removing $(u, v)$ and adding a path that starts at $u$ and ends at $v$. A *subdivision* $G^*$ of $G$ is a graph that can be obtained from $G$ by subdividing edges of $G$. Note that if $G$ does not contain indegree one, outdegree one vertices, then there exists a canonical mapping of vertices of $G$ to vertices of $G^*$ and of edges of $G$ to paths of $G^*$.

Let $G$ be a connected, directed graph and let $H$ be a directed graph. Then we say $G$ has an *embedding* into $H$ if there exists a subdivision $G^*$ of $G$ that is a subgraph of $H$. Now assume that $G$ has components $C_1, \ldots, C_k$. Then we say $G$ has an *embedding* into $H$ if the components $C_i$ of $G$, for $i \in \{1, \ldots, k\}$, have embeddings into $H$ to pairwise edge-disjoint subgraphs of $H$. If $G$ contains a vertex $v$ with a label, then we require that such an embedding maps $v$ to a vertex of $H$ with the same label.

## 2.2 Phylogenetic networks

Recall that with phylogenetic trees and networks we want to model phylogenies of a set of taxa. For this, let $\mathcal{X} = \{1, 2, \ldots, n\}$ be a finite, nonempty set, which represents our taxa.

Like with graphs, we distinguish between the directed and undirected case.

### 2.2.1 Rooted phylogenetic networks

A *rooted phylogenetic network* $N = (V, E)$ on a set of *taxa* $\mathcal{X}$ is a rooted directed acyclic graph with the following properties.

- The unique root is labelled $\rho$ and has indegree zero and outdegree one.
- The leaves are bijectively labelled with $\mathcal{X}$.
- All other vertices are either *inner tree vertices* with indegree one and outdegree at least two or *reticulations* with indegree at least two and outdegree one.

A rooted phylogenetic network $N$ is called *binary* if every inner tree vertex and reticulation of $N$ has degree three, otherwise it is called *non-binary* or *multifurcating*. More precisely, in a binary phylogenetic network inner tree vertices have indegree one and outdegree two, and reticulations have indegree two and outdegree one. The *tree vertices* of $N$ are the union of the inner tree vertices, the leaves, and the root. The *topology* of $N$ is the graph obtained from $N$ by removing the leaf labels.

The unique edge incident to the root is called the *root edge* $e_\rho$. An edge $e = (u, v)$ is called a *reticulation edge* if $v$ is a reticulation, and is called a *tree edge* if $v$ is a tree vertex. Furthermore, $e = (u, v)$ is *pure* if $u$ and $v$ are both either tree vertices or both reticulations, and *impure* otherwise. An edge is an *external edge* if it is incident to the root or a leaf, and an *internal edge* otherwise. Following Bordewich et al. [BLS17], edges in $N$ can be in *parallel*; that is, two distinct edges join the same pair of vertices.

The focus of this thesis is on rooted binary phylogenetic networks on $\mathcal{X}$. Therefore, to ease reading, we refer to a rooted binary phylogenetic network on $\mathcal{X}$ simply as a phylogenetic network or network. Let $\mathcal{N}_n$ denote the set of all phylogenetic networks on $\mathcal{X} = \{1, 2, \ldots, n\}$.

**Relationships.** Let $N \in \mathcal{N}_n$ and let $u$ and $v$ be two distinct vertices of $N$. If there is an edge $(u, v)$ in $N$, then $u$ is a *parent* of $v$ and $v$ is a *child* of $u$. Vertices that have a common parent are *siblings*. A pair of leaves $\{u, v\}$ is called a *cherry* if $u$ and $v$ are siblings. The vertex $u$ is an *ancestor* of $v$ and $v$ is a *descendant* of $u$ if there is a path from $u$ to $v$ in $N$. Note that with this definition a vertex is neither its own ancestor nor its own descendant. The vertex $u$ is an *uncle* of $v$ if $u$ is sibling of a parent of $v$. In reverse, $v$ is then the *nephew* of $u$.

Let $(u, v), (x, y)$ be two distinct edges of $N$. The edge $(u, v)$ is a *parent edge* of $(x, y)$ if $v = x$. In this case, $(x, y)$ is a *child edge* of $(u, v)$. The two edges are *siblings edges* if $u = x$, and *partner edges* if $v = y$. Note that two parallel edges are both sibling and partner edges. The edge $(u, v)$ is an *ancestor* of the edge $(x, y)$ and a vertex $x$ if $v = x$ or if $v$ is an ancestor of $x$. In this case, $(x, y)$ is a *descendant* of $(u, v)$ and $v$.

### 2.2.2 Unrooted phylogenetic networks

An *unrooted phylogenetic network* $N = (V, E)$ on a set of *taxa* $\mathcal{X}$ is an undirected graph such that the leaves are bijectively labelled with $\mathcal{X}$. An unrooted phylogenetic network $N$ is called *binary* if every non-leaf vertex of $N$ has degree three. An *unrooted phylogenetic tree* is an unrooted phylogenetic network that is a tree.

While we do not work directly with unrooted phylogenetic networks in this thesis, they are still important for us to understand related work. In fact, most of the concepts and

problems we look at were first considered for the unrooted case. If not mentioned otherwise, we assume that unrooted phylogenetic trees and networks are binary.

### 2.2.3 The landscape of network classes

A *class of phylogenetic networks* is a subset of networks in $\mathcal{N}_n$ that share a certain structural property. Such a property can be motivated by phylogenetic concepts or simply to limit the complexity of a network. We now define several important classes of phylogenetic networks and at the end of this section look at their relationships.

#### Phylogenetic trees

A *rooted phylogenetic tree* is a phylogenetic network that is a tree and, hence, has no reticulation. Again for simplicity, we refer to a rooted binary phylogenetic tree on $\mathcal{X}$ simply as phylogenetic tree or tree. See the tree $T$ in Figure 2.1 for an example.

Let $\mathcal{T}_n$ denote the class of all phylogenetic trees with $n$ leaves. The size of $\mathcal{T}_n$ is well known as it is the solution to Schröder's third problem [Sch70].

**Theorem 2.1.**
*For $n \geq 2$, the size of $\mathcal{T}_n$ is $(2n-3)!! = 1 \cdot 3 \cdot 5 \cdot \ldots \cdot (2n-5) \cdot (2n-3)$.*

For example, for $n = 10$ there are 34 459 425 different trees and for $n = 15$ over $213 \cdot 10^{12}$ different trees in $\mathcal{T}_n$. This illustrates how vast $\mathcal{T}_n$ is and why searching it can be computationally difficult and a traversal outright unfeasible. Moreover, all other classes we define have $\mathcal{T}_n$ as a subset and are thus even larger.

We now define a special type of phylogenetic trees. A phylogenetic tree $T$ is a *caterpillar* if it contains a path $P$ from the root to a leaf such that the parent of every leaf of $T$ is on $P$. Note that a caterpillar contains exactly one cherry.

#### Tree-child and normal networks

A *tree-child* network is a phylogenetic network where each non-leaf vertex has at least one tree child; that is, it has a child that is a tree vertex. A *normal* network is a tree-child network that does not contain a transitive edge [Wil10]. Note that, while every normal network is a tree-child network, the reverse does not hold. This is also illustrated by the normal and the tree-child networks shown in Figure 2.1. Let $\mathcal{TC}_n$ and $\mathcal{NN}_n$ denote the class of tree-child and normal networks with $n$ leaves, respectively.

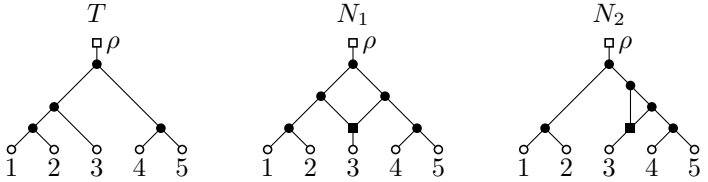


Figure 2.1: A phylogenetic tree $T \in \mathcal{T}_5$, a normal network $N_1 \in \mathcal{NN}_5$, and a tree-child network $N_2 \in \mathcal{TC}_5$. Note that $N_1 \notin \mathcal{T}_5$ and $N_2 \notin \mathcal{NN}_5$. As in this figure and throughout the whole thesis, a vertex depicted with □ is a root, with ● is an inner tree vertex, with ■ is a reticulation, and with ○ is a leaf.

A well known property of a tree-child network $N$ is that each vertex $v$ of $N$ contains a path to a leaf consisting only of tree edges. Such a path is called a *tree path* of $v$. As a consequence and as stated in Table 2.2, a tree-child network can have at most $n-1$

reticulations [CRV09, Proposition 1]. This can be seen as the root and each reticulation have a tree path to a different leaf. A normal network can have at most $n-2$ reticulations, since the root of a normal network has tree paths to two distinct leaves [Bic12, Proposition 2]. Furthermore, note that a tree-child network cannot contain any parallel edges.

### Temporal normal networks

A *temporal* network is a network $N = (V, E)$ for which a *time function* $f\colon V \to \mathbb{N}$ exists such that $f(u) < f(v)$ for each tree edge $(u, v)$ and $f(u) = f(v)$ for each reticulation edge $(u, v)$ [Bar04]. A *temporal normal* network is a network that is both temporal and normal. Let $\mathcal{TP}_n$ denote the class of temporal normal networks with $n$ leaves. Note that by definition $\mathcal{TP}_n \subseteq \mathcal{NN}_n$ and, in fact, it holds that $\mathcal{TP}_n \subsetneq \mathcal{NN}_n$. The tight upper bound for reticulations in a temporal network is $n-2$, since temporal networks are also normal networks. See Figure 3.6 for an example of a temporal network with $n-2$ reticulations.

### Tree-sibling networks

A *tree-sibling* network is a phylogenetic network where each reticulation has at least one tree vertex as sibling. An example is shown in Figure 2.2. Let $v$ be a reticulation of a tree-sibling network. A tree vertex $w$ is called a *tree-sibling witness* (or simply witness) of $v$ if $w$ is a sibling of $v$. Note that in a tree-sibling network a reticulation can have two witnesses, but a tree vertex $w$ can only be witness for one reticulation since $w$ has only one sibling. Let $\mathcal{TS}_n$ denote the class of tree-sibling networks with $n$ leaves.

### Reticulation-visible networks

A vertex $v$ of a phylogenetic network is called *visible* if there exists a leaf $l$ such that every path from the root to $l$ contains $v$. A *reticulation-visible* network is a phylogenetic network without parallel edges where each reticulation is visible. See for example $N_4$ in Figure 2.2. Let $\mathcal{RV}_n$ denote the class of reticulation-visible networks with $n$ leaves.

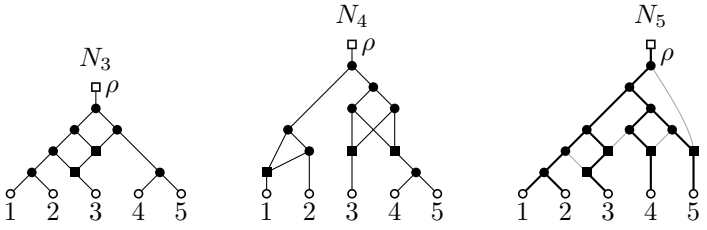


Figure 2.2: A tree-sibling network $N_3 \in \mathcal{TS}_5$, a reticulation-visible network $N_4 \in \mathcal{RV}_5$, and a tree-based network $N_5 \in \mathcal{TB}_5$. Note that $N_3, N_5 \notin \mathcal{RV}_5$ and $N_4, N_5 \notin \mathcal{TS}_5$. Note that $N_5$ has $T$ from Figure 2.1 as base tree (as indicated).

### Tree-based networks

A *tree-based* network $N \in \mathcal{N}_n$ is a phylogenetic network for which an embedding of a phylogenetic tree $T \in \mathcal{N}_n$ into $N$ exists that covers all vertices of $N$. In this case, $T$ is called a *base tree* for $N$. For example, the network $N_5$ shown in Figure 2.2 is a tree-based network. Francis and Steel [FS15] introduced tree-based networks as networks that can be obtained by adding edges between the edges of a base tree. Further characterisations of tree-based networks are known [Zha16, PSS19, FSS18].

Let $\mathcal{TB}_n$ denote the class of tree-based networks with $n$ leaves. Let $T \in \mathcal{T}_n$. Then let $\mathcal{TB}_n(T)$ denote the class of tree-based networks that have $T$ as base tree. There exist so-called universal tree-based networks, which are tree-based networks that have every tree $T \in \mathcal{T}_n$ as a base tree [Hay16, Zha16, BS18]. The class of tree-based networks has also been generalised to rooted non-binary networks [JvI18], unrooted networks [FHM18], and unrooted, non-binary networks [Hen18, FGH$^+$18].

#### Tree-set displaying networks

Let $T \in \mathcal{T}_n$ and $N \in \mathcal{N}_n$. Then $N$ *displays* $T$ if $T$ has an embedding into $N$. Displaying $T$ is a weaker condition than having $T$ as a base tree, since the embedding does not have to cover all vertices. For example, in Figure 2.1 the network $N_1$ displays $T$, but $N_2$ does not. A network $N$ *displays* a set of trees $P \subseteq \mathcal{T}_n$ if $N$ displays every tree $T \in P$. Let $\mathcal{N}_n(T)$ and $\mathcal{N}_n(P)$ denote the class of phylogenetic networks in $\mathcal{N}_n$ that display $T$ and $P$, respectively. In reverse, let $D(N) \subseteq \mathcal{T}_n$ denote the set of trees displayed by $N$.

Now let $N, N' \in \mathcal{N}_n$. Then $N$ *displays* $N'$ if $N'$ has an embedding into $N$. Note that this implies that $N'$ has at most as many reticulations as $N$.

#### Level-$k$ networks

A *blob* $B$ of a network $N$ is a subgraph corresponding to a nontrivial biconnected component of the underlying graph of $N$. The *level* of a blob $B$ of $N$ is the number of reticulations in $B$. The *level* of $N$ is the maximum level over all blobs of $N$. A network $N$ is a *level-$k$* network if its level is at most $k$. A network $N$ is a *strict level-$k$* network if its level is exactly $k$. Let $\mathcal{LV}_{k,n}$ and $s\mathcal{LV}_{k,n}$ denote the classes of level-$k$ and strict level-$k$ networks.

For example, in Figure 2.2 the networks $N_3$ and $N_4$ are level-2 networks with one and two blobs, respectively, and the network $N_5$ is a level-4 network. Note that the level-0 networks $\mathcal{LV}_{0,n}$ are phylogenetic trees $\mathcal{T}_n$. The level of a network can be regarded, to some extent, as a measure of its distance of being a phylogenetic tree.

Level-$k$ networks have also been defined in the unrooted case. There the level of a blob is the minimum number of edges that have to be removed from the blob to make it acyclic. The level of an unrooted phylogenetic network is again the maximum level over all its blobs [HLMW16].

#### Tiers

Let $\mathcal{C}_n$ be a class of phylogenetic networks. The *tier* $r$ of $\mathcal{C}_n$ is the subset of networks of $\mathcal{C}_n$ that have exactly $r$ reticulations. Let $\mathcal{C}_{n,r}$ denote tier $r$ of $\mathcal{C}_n$. For example, note that tier zero of $\mathcal{N}_n$ is precisely $\mathcal{T}_n$.

#### Relationships and properties

We now look at the relations of different classes of networks in terms of inclusions and some of their properties. On the one hand, note that $\mathcal{T}_n$ is a subclass of all other classes, except for strict level-$k$ networks with $k > 0$. On the other hand, note that $\mathcal{N}_n$ is a superclass of all classes. Further inclusions are given in Table 2.1 and in Figure 2.3.

In addition to Table 2.1 and as illustrated in Figure 2.2, note that not every tree-sibling network is a reticulation-visible network nor vice versa. Furthermore, Semple [Sem16] showed that the class of tree-child networks is precisely the class of networks with the property that every embedded phylogenetic tree of a network is also a base tree of that network.

Table 2.1: Subclass relations between different classes of phylogenetic networks.

| # | inclusion | reference |
|---|---|---|
| 1 | $\mathcal{TP}_n \subsetneq \mathcal{NN}_n$ | by definitions |
| 2 | $\mathcal{NN}_n \subsetneq \mathcal{TC}_n$ | by definition |
| 3 | $\mathcal{TC}_n \subsetneq \mathcal{TS}_n$ | [CLRV08] |
| 4 | $\mathcal{TC}_n \subsetneq \mathcal{RV}_n$ | [HRS10] |
| 5 | $\mathcal{TS}_n \subsetneq \mathcal{TB}_n$ | [FS15, Corollary 3.3] |
| 6 | $\mathcal{RV}_n \subsetneq \mathcal{TB}_n$ | [GGL$^+$15, Lemma 1] |
| 7 | $\mathcal{LV}_{i,n} \subsetneq \mathcal{LV}_{i+1,n}$ | by definition |

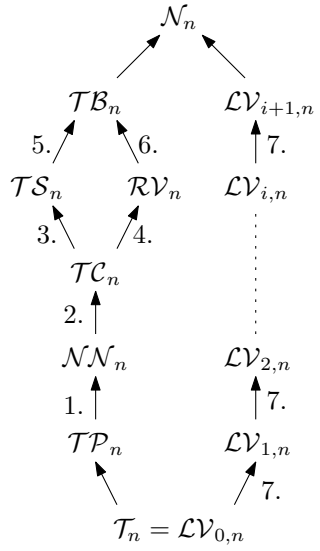


Figure 2.3: Diagram illustrating the subclass relations of classes of phylogenetic networks with respect to Table 2.1.

Next, we look at the maximum number of reticulations a phylogenetic network of a certain class $\mathcal{C}_n$ may have. We already pointed out the results for temporal, normal, and tree-child networks. Further bounds are shown in Table 2.2. Note that an unbounded number of reticulations implies that there are infinitely many networks in $\mathcal{C}_n$.

Theorem 2.1 gives a precise formula for the number of trees in $\mathcal{T}_n$. In addition, the size of $\mathcal{T}_n$ can also be expressed asymptotically by $|\mathcal{T}_n| \in 2^{n \log n + \mathcal{O}(n)}$. Similarly, the size of $|\mathcal{TC}_n|$ is bounded by $|\mathcal{TC}_n| \in 2^{2n \log n + \mathcal{O}(n)}$ [MSW15]. Cardona et al. [CPS19] gave a recursive algorithm to uniquely generate each network in $\mathcal{TC}_n$ . They used this to show that for $n = 6$ there are 101 833 875 networks in $\mathcal{TC}_6$, compared to 945 trees in $\mathcal{T}_6$. Further asymptotic results for tiers of $\mathcal{NN}_n$ and $\mathcal{TC}_n$ were given by Fuchs et al. [FGM19].

## 2.3 Rearrangement operations

A *rearrangement operation* is the process of making graph-theoretic changes to a phylogenetic network such that the resulting graph is again a phylogenetic network. A rearrangement operation is a *horizontal move* if it does not change the number of reticulations; i.e. both the starting and the resulting network are in the same tier. On the other hand, a rearrangement operation that changes the number of reticulations is a *vertical move*.

Table 2.2: The maximum number of reticulations a phylogenetic network of a certain class may have.

| class | max number of reticulations | reference |
|---|---|---|
| $\mathcal{T}_n$ | 0 | |
| $\mathcal{NN}_n$ | $n-2$ | [Bic12, Proposition 2] |
| $\mathcal{TC}_n$ | $n-1$ | [CRV09, Proposition 1] |
| $\mathcal{TP}_n$ | $n-2$ | Table 2.1 and Figure 3.6 |
| $\mathcal{TS}_n$ | unbounded | |
| $\mathcal{RV}_n$ | $3(n-1)$ | [GZ15, Theorem 3.3] |
| $\mathcal{TB}_n$ | unbounded | |
| $\mathcal{LV}_{i,n}, i>0$ | unbounded | |
| $\mathcal{N}_n$ | unbounded | |

### 2.3.1 Prune and regraft

Let $G$ be a directed graph. Let $(u,v)$ be an edge of $G$ where $u$ is either labelled (like the root of a network) or has degree three. Then *pruning* $(u,v)$ at $u$ is the process of deleting $(u,v)$ and adding a new edge $(u',v)$, where $u'$ is a new vertex. If $u$ is now an indegree one outdegree one vertex, then $u$ gets suppressed. Now let $(u',v)$ be an edge where $u'$ is an unlabelled, outdegree one vertex. Then *regrafting* $(u',v)$ to an edge $(x,y)$ is the process of subdividing $(x,y)$ with a new vertex $u$ and identifying $u'$ with $u$. Also, *regrafting* $(u',v)$ to a vertex $u$ is the process of identifying $u'$ with $u$. Pruning and regrafting an edge $(u,v)$ at $v$ is analogously defined.

Let $N \in \mathcal{N}_n$ and let $(u,v)$ be an edge of $N$. Then the *prune and regraft* (PR) operation is the rearrangement operation that transforms $N$ into a phylogenetic network $N' \in \mathcal{N}_n$ in one of the following four ways:

($\mathrm{PR}^0$) If $u$ is an inner tree vertex, then prune $(u,v)$ at $u$ and regraft it to an edge that is not a descendant of $v$; or
if $v$ is a reticulation, then prune $(u,v)$ at $v$ and regraft it to an edge that is not an ancestor of $u$.

($\mathrm{PR}^+$) Subdivide $(u,v)$ with a new vertex $v'$, subdivide an edge in the resulting graph that is not a descendant of $v'$ with a new vertex $u'$, and add the edge $(u',v')$.

($\mathrm{PR}^-$) If $(u,v)$ is an impure reticulation edge, then delete $(u,v)$ and suppress both $u$ and $v$.

A $\mathrm{PR}^0$ operation that prunes an edge $(u,v)$ at its head vertex $v$ (resp. tail vertex $u$) is called a *head (tail)* $\mathrm{PR}^0$. Note that a $\mathrm{PR}^0$ does not change the number of reticulations, while a $\mathrm{PR}^-$ decrease it by one and a $\mathrm{PR}^+$ increase it by one. In other words, $\mathrm{PR}^0$ operations are horizontal moves and $\mathrm{PR}^-$ and $\mathrm{PR}^+$ operations are vertical moves. The operations are illustrated in Figure 2.4.

The *subtree prune and regraft* (SPR) operation equals the $\mathrm{PR}^0$ operation restricted to $\mathcal{T}_n$. Note that some authors write rSPR (rooted SPR) for SPR. Here, however, we simply use SPR as the rootedness is implicitly given by the tree under consideration.

The *subnet prune and regraft* (SNPR) operation equals the PR operation except that it excludes head $\mathrm{PR}^0$. SNPR was first defined by Bordewich et al. [BLS17]. Gambette

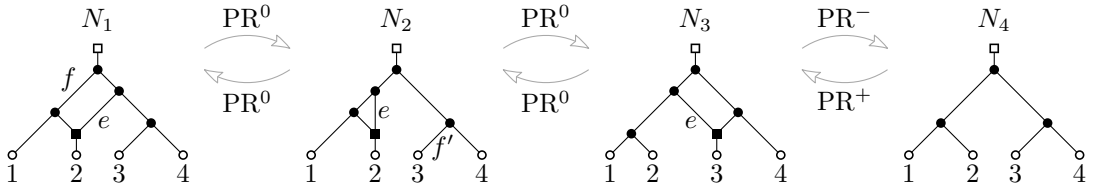


Figure 2.4: The phylogenetic network $N_2$ (resp. $N_3$) can be obtained from $N_1$ (resp. $N_2$) by the tail $\mathrm{PR}^0$ (resp. head $\mathrm{PR}^0$) that prunes $e$ and regrafts it to $f$ (resp. $f'$). The phylogenetic network $N_4$ can be obtained from $N_3$ with the $\mathrm{PR}^-$ that removes $e$. Each operation has its corresponding $\mathrm{PR}^0$ or $\mathrm{PR}^+$ operation that reverses the transformation.

et al. [GvIJ+17] defined *head* and *tail moves* that conceptually equal head and tail $\mathrm{PR}^0$, but restricted this generalisation of SPR to networks without parallel edges.

Bordewich et al. [BLS17] and Gambette et al. [GvIJ+17] have shown that the different types of PR operations are all reversible. This means that for every $\mathrm{PR}^0$ (or $\mathrm{SNPR}^0$) that transforms $N$ into $N'$ there exists a $\mathrm{PR}^0$ (resp. $\mathrm{SNPR}^0$) that transforms $N'$ into $N$, and that for every $\mathrm{PR}^+$ (resp. $\mathrm{SNPR}^+$) there exists an inverse $\mathrm{PR}^-$ (resp. $\mathrm{SNPR}^-$).

### 2.3.2 Nearest neighbour interchange

Let $N \in \mathcal{N}_n$ and let $(u, v)$ be an inner edge of $N$. If $u$ is tree vertex, let $w$ be the second child of $u$. Then the *nearest neighbour interchange* (NNI) operation is the rearrangement operation that transforms $N$ into a phylogenetic network $N' \in \mathcal{N}_n$ in one of the following ways:

($\mathrm{NNI}^0$) If $(u, v)$ is a pure tree edge, then prune an outgoing edge of $v$ at $v$ and regraft it to the edge $(u, w)$; or
if $(u, v)$ is an impure tree edge, then prune an outgoing edge of $v$ at $v$ and regraft it to an incoming edge of $u$; or
if $(u, v)$ is an impure reticulation edge, then prune $(u, w)$ at $u$ and regraft it to an edge incident to $v$ that is not the edge derived from $(u, v)$; or
if $(u, v)$ is a pure reticulation edge, then prune the incoming edge of $v$ that is not $(u, v)$ at $v$ and regraft it to an incoming edge of $u$.

($\mathrm{NNI}^+$) Subdivide $(u, v)$ with a new vertex $v'$, subdivide an edge incident to $u$ that is not $(u, v')$ with a new vertex $u'$, and add the edge $(u', v')$.

($\mathrm{NNI}^-$) If $u$ is a tree vertex and $v$ is a reticulation and both are adjacent to a third vertex $w$, then delete $(u, v)$ and suppress $u$ and $v$.

The edge $(u, v)$ of an $\mathrm{NNI}^0$ is called the *axis* of the operation. Note that an $\mathrm{NNI}^0$ can be seen as contracting the edge $(u, v)$ and reversing the contraction such that the resulting graph is again a phylogenetic network. Similarly, an $\mathrm{NNI}^-$ can be seen as contracting a triangle of $N$ into a vertex and an $\mathrm{NNI}^+$, as the reverse of an $\mathrm{NNI}^-$, can be seen as adding a triangle. These operations are illustrated in Figure 2.5. Consider an $\mathrm{NNI}^0$ with the axis $e$. Note that an $\mathrm{NNI}^0$ is a tail $\mathrm{PR}^0$ if $e$ is a tree edge, a head $\mathrm{PR}^0$ if $e$ is a pure reticulation edge, and either a head or a tail $\mathrm{PR}^0$ if $e$ is an impure reticulation edge. Furthermore, $\mathrm{NNI}^+$ and $\mathrm{NNI}^-$ operations are special cases of $\mathrm{PR}^+$ and $\mathrm{PR}^-$ operations.

NNI on networks was first defined by Huber et al. [HLMW16] on the restricted case of rooted and unrooted level-1 networks. Later, Gambette et al. [GvIJ+17] defined the $\mathrm{NNI}^0$

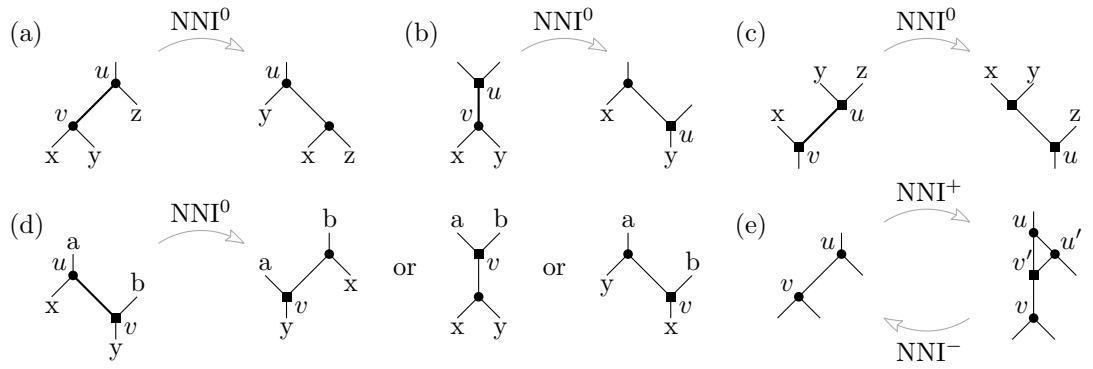


Figure 2.5: Illustration of an $\mathrm{NNI}^0$ when the axis $(u, v)$ is a (a) pure tree edge, (b) impure tree edge, (c) pure reticulation edge, or (d) impure reticulation edge, and of an $\mathrm{NNI}^+$ and $\mathrm{NNI}^-$ (e). Note that in (e) the vertices $u$ and $v$ could be reticulations.

operation on networks without parallel edges. In general, for either of these generalisations, the $\mathrm{NNI}^0$ operations on $\mathcal{T}_n$ equals the classical NNI operation.

### 2.3.3 Unrooted versions

Rearrangement operations have also been defined on unrooted phylogenetic networks. A major difference between rooted and unrooted networks is of course that in unrooted networks there are no "reticulations" and "tree vertices", only inner vertices with degree three.

An NNI on an unrooted (binary) phylogenetic tree contracts an inner edge and then splits the resulting degree four vertex such that it results in a binary phylogenetic tree again [Rob71]. The generalisation of NNI to unrooted phylogenetic networks works the same for horizontal moves, but adds vertical moves that add or remove a triangle (like $\mathrm{NNI}^+$ and $\mathrm{NNI}^-$) [HLMW16, HMW16]. Again, some authors do not allow parallel edges in their unrooted phylogenetic networks [FHMW18].

The SPR operation on unrooted phylogenetic trees prunes an edge from a degree three vertex and regrafts it such that the resulting tree is again connected. Generalisations of SPR to unrooted phylogenetic networks work the same [FHMW18, JK19]. Vertical moves for SPR on unrooted phylogenetic networks work like $\mathrm{PR}^+$ and $\mathrm{PR}^-$ as they just add or remove an edge [JK19].

A third rearrangement operation on unrooted phylogenetic trees is *tree bisection and reconnection* (TBR), which first removes an edge and then adds a new edge to reconnect the tree. Francis et al. [FHMW18] considered the straightforward generalisation of this horizontal move to unrooted phylogenetic networks. Janssen and Klawitter [JK19] further investigated a version of TBR that includes vertical moves.

## 2.4 Metric spaces

Let $S$ be a set. A *metric* (or *distance function*) on $S$ is a function $d\colon S \times S \to [0, \infty)$ where for all $x, y, z \in S$ the following conditions hold:

1. $d(x, y) = 0 \Leftrightarrow x = y$ (non-negativity)

2. $d(x, y) \geq 0$ (identity of indiscernibles)

3. $d(x, y) = d(y, x)$ (symmetry)

4. $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality)

A *metric space* $(S, d)$ is a set $S$ together with a metric $d$ on $S$.

Let $G = (V, E)$ be a graph. The distance of two vertices in $G$ is the length of a shortest path between those vertices. If no such path exists, then their distance is considered to be infinite. Hence, this distance defines a metric on $G$ if $G$ is undirected and connected.

**Phylogenetic network space.** Let $\mathcal{C}_n$ be a class of phylogenetic networks and consider a type of operation $op \in \{\text{NNI}, \text{SNPR}, \text{PR}\}$ on $\mathcal{C}_n$. Then $op$ together with $\mathcal{C}_n$ define the *rearrangement graph* $\mathcal{C}_n^{op} = (\mathcal{C}_n, E)$ where two distinct networks $N, N' \in \mathcal{C}_n$ are adjacent if $N$ can be transformed into $N'$ by a single operation $op$. This graph is undirected since NNI, SNPR, and PR are reversible.

Let $N, N' \in \mathcal{C}_n$. An *$op$-sequence* from $N$ to $N'$ is a sequence

$$\sigma = (N = N_0, N_1, N_2, \ldots, N_k = N')$$

of phylogenetic networks such that, for each $i \in \{1, 2, \ldots, k\}$, $N_i \in \mathcal{C}_n$ and $N_i$ can be obtained from $N_{i-1}$ by a single operation $op$. The *length* of $\sigma$ is $k$. We define the *$op$-distance* $\mathrm{d}_{op}(N, N')$ of $N, N'$ as the length of a shortest $op$-sequence from $N$ to $N'$ in $\mathcal{C}_n^{op}$ or infinite if no such path exists.

A *phylogenetic network space* is a metric space over a set of phylogenetic networks together with a metric $d$. Note that a rearrangement operation $op$ and a class $\mathcal{C}_n$ form a space if the graph $\mathcal{C}_n^{op}$ is connected. Let $\mathcal{C}_n^{op}$ and $\mathcal{C}_n'^{op}$ be two metric spaces of phylogenetic networks such that $\mathcal{C}_n \subseteq \mathcal{C}_n'$. Then we say that $\mathcal{C}_n^{op}$ is an *isometric subgraph* of $\mathcal{C}_n'^{op}$ if the $op$-distance of two networks $N$ and $N'$ in $\mathcal{C}_n$ equals their $op$-distance in $\mathcal{C}_n'$.

# 3. Connectedness and diameter

In this chapter we investigate when a class of phylogenetic networks and a rearrangement operation form a metric space. Consider the rearrangement graph $\mathcal{C}_n^{op}$ for a class of phylogenetic networks $\mathcal{C}_n$ and a rearrangement operation *op*. We know that such a graph is undirected for the NNI, the SNPR, and the PR operation. Hence, the usual distance of vertices in a graph yields a metric on $\mathcal{C}_n^{op}$ precisely when $\mathcal{C}_n^{op}$ is connected. Recall that a graph is connected if there exists a path between any two of its vertices. If on the other hand there exists no path between two vertices, their distance is undefined and while it may be set to infinity, this would not comply with the definition of a metric. In other words, if $\mathcal{C}_n^{op}$ is connected, then the distance between any two networks in $\mathcal{C}_n$ is well defined. Here we consider connectedness of different classes of phylogenetic networks under NNI, SNPR, and PR to establish a basis for the subsequent chapters. Our focus is on the latter two rearrangement operations. We also look at the tiers of these classes. In the case where a class and an operation form a metric space, we give a bound on the diameter. In particular, we look at tree-child networks (Section 3.3), normal networks (Section 3.4), temporal normal networks (Section 3.5), tree-sibling networks (Section 3.6), reticulation-visible networks (Section 3.7), tree-based networks (Section 3.8), and level-$k$ networks (Section 3.9). A summary of the results of this chapter is given at the end in Section 3.10.

The question of whether a class is connected under an operation is not only important to obtain a metric space, but also from a practical point of view. Consider a local search on $\mathcal{C}_n^{op}$. Connectedness of the graph then means that the search can theoretically reach every network. However, if the graph is unconnected, then such a search stays in the component where it started, while the optimal solution may be found in a different component.

Given a finite graph $G$ with $n$ vertices and $m$ edges it can be tested in $\mathcal{O}(n+m)$ time whether $G$ is connected [CLRS09]. However, this requires a representation of $G$ that allows an efficient traversal. In the case of $\mathcal{C}_n^{op}$, we only know that the vertices are networks with certain properties and under which conditions two vertices are adjacent. Furthermore, from a computational point of view such a graph is usually too vast for a full traversal, since by Theorem 2.1 the number of rooted phylogenetic trees in $\mathcal{T}_n$ is $(2n-3)!!$. If the graph is not finite, then connectedness cannot be tested with a simple traversal at all. Hence, we need to prove or disprove the connectedness of a space. Here we do this in one of two ways. First, if we already know that a subgraph $H$ of $\mathcal{C}_n^{op}$ is connected, then it is sufficient to show that every vertex of $\mathcal{C}_n^{op}$ is connected to a vertex in $H$. Second, we pick a *target network* $M$ and show that all vertices of $\mathcal{C}_n^{op}$ are connected to $M$. In some cases, we

prove disconnectedness by showing that no path between two networks exists. Note that, since PR generalises SNPR, the graph $\mathcal{C}_n^{\mathrm{SNPR}}$ is a subgraph of the graph $\mathcal{C}_n^{\mathrm{PR}}$. Since both these graphs have the same vertex set $\mathcal{C}_n$, it follows that connectedness of $\mathcal{C}_n^{\mathrm{SNPR}}$ implies connectedness of $\mathcal{C}_n^{\mathrm{PR}}$.

Recall that the diameter $\mathrm{diam}(G)$ of a connected graph $G$ is the maximum distance between any two of its vertices. If the graph is not finite, then the diameter can be unbounded. Note that an unbounded diameter does not mean that two networks (i.e. two vertices of the graph) have an infinite distance, but that, for any integer $c$, we can find a pair of networks that has a distance of at least $c$. Knowing bounds for the diameter is of interest for sampling algorithms like MCMC. For example, a small diameter may affect the mixing time of MCMC positively, whereas a large diameter means that any MCMC needs a long time to potentially reach every region of the graph. If the diameter is unbounded, then it is not even possible for a walk to reach every element of the graph. For a class $\mathcal{C}_n$ with bounded size, we derive results concerning the diameter of $\mathcal{C}_n^{op}$ by bounding the maximum length of paths between networks that we use to show the connectedness of $\mathcal{C}_n^{op}$.

Concerning phylogenetic trees, it is well known that $\mathcal{T}_n^{\mathrm{NNI}}$ and $\mathcal{T}_n^{\mathrm{SPR}}$ are connected. Furthermore, Li et al. [LTZ96] gave asymptotic bounds of $\Theta(n \log n)$ on the maximum distance between two phylogenetic trees under NNI. Song [Son03, Proposition 5.1] gave the upper bound $\max\{n-2, 0\}$ for the diameter under SPR. Rephrasing these results we get the following theorems.

**Theorem 3.1** (Li et al. [LTZ96])**.**
*The graph* $\mathcal{T}_n^{\mathrm{NNI}}$ *is connected with* $\mathrm{diam}(\mathcal{T}_n^{\mathrm{NNI}}) \in \Theta(n \log n)$.

**Theorem 3.2** (Song [Son03])**.**
*The graph* $\mathcal{T}_n^{\mathrm{SPR}}$ *is connected with* $\mathrm{diam}(\mathcal{T}_n^{\mathrm{SPR}}) \in \Theta(n)$.

Recall that SPR, SNPR, and PR act equivalently on $\mathcal{T}_n$. Hence Theorem 3.2 also holds for SNPR and PR.

The first result showing the connectedness of a class of phylogenetic networks was given in the pioneering work by Huber et al. [HLMW16] who showed that the spaces of unrooted and rooted level-1 networks are connected under NNI. This was extended to general unrooted phylogenetic networks and their tiers by Huber et al. [HMW16]. Based on this, Francis et al. [FHMW18] obtained that these spaces are also connected under generalisations of SPR and TBR on unrooted networks. Concerning rooted phylogenetic networks, Bordewich et al. [BLS17] proved connectedness and bounds on diameters of spaces of tree-child, reticulation-visible, tree-based, and general phylogenetic networks under SNPR. Gambette et al. [GvIJ+17], Janssen et al. [JJE+18], and Janssen [Jan18] established that $\mathcal{N}_{n,r}$ is connected and gave bounds on the diameter for NNI, SNPR, and head PR. We look at these results more closely below, after the following preliminary section.

## 3.1 Preliminaries

In this section we devise several lemmata that hold for more than one class of phylogenetic networks and define specific phylogenetic networks that will serve as target networks. Recall that we consider only rooted, binary phylogenetic trees and networks.

**Lemma 3.3.**
*Let* $\mathcal{C}_n$ *be a class of phylogenetic networks with* $\mathcal{T}_n \subset \mathcal{C}_n$*. Let* $op \in \{\mathrm{NNI}, \mathrm{SNPR}, \mathrm{PR}\}$*.*
*If for every* $N \in \mathcal{C}_n$ *with* $r > 0$ *reticulations there exists a network* $N' \in \mathcal{C}_n$ *that has* $r-1$ *reticulations and is adjacent to* $N$ *in* $\mathcal{C}_n^{op}$*, then* $\mathcal{C}_n^{op}$ *is connected.*

*Proof.* Let $N_r$ be any network in $\mathcal{C}_n$ and suppose it has $r$ reticulations. By the requested property of $\mathcal{C}_n^{op}$, we can construct a sequence $\sigma = (N_r, \ldots, N_0)$ where $N_i$ has $i$ reticulations for $i \in \{0, \ldots, r\}$. In other words, there is a path from $N_r$ to a tree $N_0$ in $\mathcal{C}_n^{op}$. Since $\mathcal{T}_n^{op}$ is connected by Theorems 3.1 and 3.2, it follows that $\mathcal{C}_n^{op}$ is connected. □

**Lemma 3.4.**
*Let $\mathcal{C}_n$ be a class of phylogenetic networks where the maximum number of reticulations of a network in $\mathcal{C}_n$ is unbounded. Let $op \in \{\mathrm{NNI}, \mathrm{SNPR}, \mathrm{PR}\}$.*
*If $\mathcal{C}_n^{op}$ is connected, then $\mathrm{diam}(\mathcal{C}_n^{op})$ is unbounded.*

*Proof.* Let $N$ be a network in $\mathcal{C}_n$ with the minimum number of reticulations of a network in $\mathcal{C}_n$, say, $k$ reticulations. For any $i > k$, let $N_i$ be a network in $\mathcal{C}_n$ with $i$ reticulations. Since $\mathcal{C}_n^{op}$ is connected and since a vertical move of $op$ can increase the number of reticulations by at most one, the distance $d$ of $N$ and $N_i$ is at least $i - k$. Then, since $i$ is unbounded, it follows that $d$ and hence also $\mathrm{diam}(\mathcal{C}_n^{op})$ are unbounded. □

Let $N \in \mathcal{N}_n$. Fix an order $\tau = (l_1, \ldots, l_n)$ of the leafs on the topology $G$ of $N$. We define the *leaf order* $\sigma(N)$ of $N$ as the permutation from the leaf order $\tau$ to the leaf labelling of $N$. For example, if $l_i$ of $N$ has label $j$ we write $\sigma_i(N) = j$. Note that it only makes sense to compare leaf orders of two networks with the same topology.

Let $n \geq 2$ and let $N \in \mathcal{N}_{n,r}$. We define three special types of networks and illustrate them in Figure 3.1. Following Bordewich et al. [BLS17], for $r \leq n - 1$, we call $N$ a *strict caterpillar network* if

- each reticulation $v_i$ of $N$ has a leaf $l_i$ as child, and
- there exists an ordering $(v_1, \ldots, v_r)$ of the reticulations such that there is a tree path

  $$\rho, p_1, q_1, p_2, q_2, \ldots, p_r, q_r, t$$

  where for each $i \in \{1, \ldots, r\}$ the parents of $v_i$ are $p_i$ and $q_i$, and $t$ is a tree vertex that has only tree vertices as descendants.

Note that $t$ is a leaf, say $l_n$, for $r = n - 1$. In this case, we describe the leaf order of $N$ with respect to the ordering $(l_1, l_2, \ldots, l_r, l_n)$. Next, for $r \leq n - 2$, we call $N$ a *ladder* if

- each reticulation $v_i$ of $N$ has a leaf $l_i$ as child, and
- there exists an ordering $(v_1, \ldots, v_r)$ of the reticulations such that there exist two tree paths

  $$\rho, w, p_1, p_2, \ldots, p_r, t' \text{ and } \rho, w, q_1, q_2, \ldots, q_r, t$$

  where $w$ is the child of $\rho$, for each $i \in \{1, \ldots, r\}$ the parents of $v_i$ are $p_i$ and $q_i$, $t'$ is a leaf, and $t'$ is a tree vertex that has only tree vertices as descendants.

Similar to strict caterpillar networks, if $r = n - 2$, we describe the leaf order of $N$ with respect to the ordering $(l_1, l_2, \ldots, l_r, t = l_{n-1}, t' = l_n)$. Lastly, we call $N$ a *stack network* if there exists an ordering $(v_1, \ldots, v_r)$ of the reticulations such that

- there exist a path $v_1, v_2, \ldots, v_r$,
- there exists a tree path $\rho, p_1, q_1, q_2, \ldots, q_r$ where $p_1$ is a parent of $v_1$ and $q_i$ is a parent of $v_i$ for $i \in \{1, \ldots, r\}$, and
- the child of $v_r$ is a leaf.

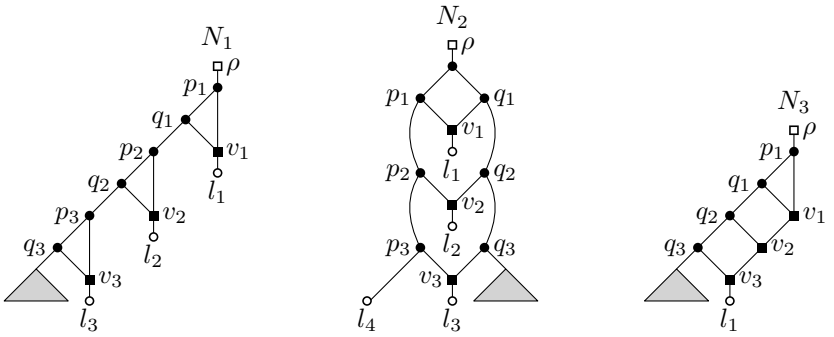


Figure 3.1: A strict caterpillar network $N_1$, a ladder $N_2$, and a stack network $N_3$ on three reticulations each. The triangle indicates a pendant subtree, called the tail.

For a strict caterpillar network, a ladder, or a stack network we call the pendant tree below $q_r$ the *tail* (see again Figure 3.1).

A reticulation $v$ is *in parallel* if its two incoming edges form a pair of parallel edges. We say $N$ has its reticulation *in series below the root* if all reticulations are in parallel and there exists an ordering $(v_1, \ldots, v_r)$ of the reticulations such that there exist a path $\rho, p_1, v_1, \ldots, p_k, v_k$ where $p_i$ is the parent of $v_i$ for $i \in \{1, \ldots, r\}$.

A *free leaf* of a network $N \in \mathcal{C}_n$ with respect to $\mathcal{C}_n$ is a leaf of $N$ that if removed from $N$ yields a network $N'$ that is in $\mathcal{C}_{n-1}$ after a potential relabelling of the leaves of $N'$. In the following lemma let $\mathcal{C}_n$ be one of the classes defined in Section 2.2.3.

**Lemma 3.5.**
*Let $N$ be a network in $\mathcal{C}_n$ with a free leaf $l$.*
*Then $N$ can be transformed into a network $N^*$ with the same topology but different leaf order with at most $2n$ SNPR$^0$.*

*Proof.* Let $e = (u, l)$ be the edge incident to $l$. Since $l$ is a free leaf, we can prune $e$ at $u$ and regraft it to the incident edge of another leaf $l'$ such that the resulting network $N'$ is in $\mathcal{C}_n$. Note that $l$ and $l'$ are free leaves in $N'$. Using this, it is straightforward to permute the leafs of $N$ to obtain the leaf order $\pi(N^*)$ with at most $2n$ SNPR, for example, by resolving one permutation cycle of $\sigma(N)$ after the other. □

## 3.2 General networks

We start with $\mathcal{N}_n$, the class of all phylogenetic networks on $n$ leaves. For NNI, Gambette et al. [GvIJ$^+$17, Theorem 3, Proposition 3] established that $\mathcal{N}_n$ is connected. Bordewich et al. [BLS17, Proposition 3.2] showed that SNPR induces a metric on $\mathcal{N}_n$. The same is thus true for PR, since PR generalises SNPR. Note that $\mathcal{N}_n$ satisfies the conditions of Lemma 3.4 and that thus the diameter of $\mathcal{N}_n$ under NNI or SNPR or PR is unbounded. Together, we get the following results.

**Theorem 3.6** (Gambette et al. [GvIJ$^+$17],Bordewich et al. [BLS17])**.**
*Let $op \in \{\text{NNI}, \text{SNPR}\}$. The graph $\mathcal{N}_n^{op}$ is connected with unbounded diameter.*

**Corollary 3.7.**
*The graph $\mathcal{N}_n^{\text{PR}}$ is connected with unbounded diameter.*

Recall that $\mathcal{T}_n$ equals $\mathcal{N}_{n,0}$; that is, the space of phylogenetic trees is tier 0 of phylogenetic networks. Janssen et al. [JJE$^+$18] proved that not only $\mathcal{T}_n$, but also the other tiers of $\mathcal{N}_n$

are connected under NNI, SNPR, and PR. Furthermore, they showed that the maximum distance for two networks in $\mathcal{N}_{n,r}^{\mathrm{SNPR}}$ is bounded linearly by $n$ and $r$. While Janssen et al. do not allow parallel edges, one can easily check that their results still hold for our definitions. In fact, allowing parallel edges we can strengthen Theorem 3.8 by including the cases $n = 2$ and $r = 1$ as well as $n = 1$.

**Theorem 3.8** (Janssen et al. [JJE+18])**.**
*Let $n \geq 3, r \geq 1$. Let $op \in \{\mathrm{SNPR}, \mathrm{PR}\}$.*
*The graph $\mathcal{N}_{n,r}^{op}$ is connected with $\mathrm{diam}(\mathcal{N}_{n,r}^{op}) \in \Theta(n+r)$.*

**Lemma 3.9.**
*Let $n \geq 1, r \geq 0$. Let $op \in \{\mathrm{SNPR}, \mathrm{PR}\}$.*
*The graph $\mathcal{N}_{n,r}^{op}$ is connected with $\mathrm{diam}(\mathcal{N}_{n,r}^{op}) \in \Theta(n+r)$.*

*Proof.* Extending Theorem 3.8, we only have to prove the cases $n = 2$ and $r = 1$ as well as $n = 1$. We start with the former. For this, Figure 3.2 shows all five networks in $\mathcal{N}_{2,1}$ and that they are connected under SNPR. This also implies the statement for $\mathcal{N}_{2,1}^{\mathrm{PR}}$.

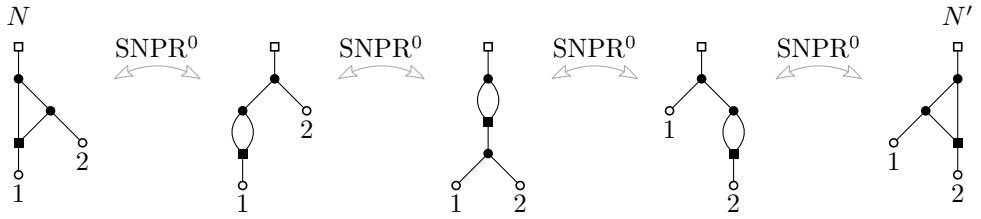


Figure 3.2: The five networks in $\mathcal{N}_{2,1}$ and how they are connected under SNPR.

Next, let $N \in \mathcal{N}_{1,r}$ for $r > 0$. We show how to transform $N$ into a target network $M \in \mathcal{N}_{1,r}$ that has its $r$ reticulations in series below the root. First, suppose that there is a reticulation $v$ with parents $u$ and $w$ where $u$ is a tree vertex and $u \neq w$. Then apply the $\mathrm{SNPR}^0$ that prunes $(u, v)$ at $u$ and regrafts it to $(w, v)$. Reapply this case as often as possible. Since this creates a pair of parallel edges but does not break one, this case applies at most $r$ times. Second, suppose that both parents $u$ and $w$ of $v$ are reticulations. Let $P$ be a path from $\rho$ to $u$. Let $u'$ be the tree vertex whose two children are tree vertices and that is closest to $u$ in $P$. Note that such $u'$ must exist since the first case does not apply and since there has to be a path from $\rho$ to $u$ and $w$ each. Then apply the $\mathrm{SNPR}^0$ that prunes the edge of $P$ at $u'$ and regrafts it to $(w, v)$. Next, the first case applies again to $v$. Again, since this creates a pair of parallel edges without breaking one, this happens at most $r$ times. Note that if neither the first nor the second case applies anymore, then each reticulation is in parallel. Since $n = 1$, it follows that they are also in series below the root. Hence, $\mathcal{N}_{1,r}$ is connected and, since this required at most $2r$ $\mathrm{SNPR}^0$, the statement on the diameter follows. □

Janssen et al. [JJE+18, Theorem 4.12] also gave bounds on the diameter of $\mathcal{N}_{n,r}^{\mathrm{NNI}}$.

**Theorem 3.10** (Janssen et al. [JJE+18])**.**
*The graph $\mathcal{N}_{n,r}^{\mathrm{NNI}}$ is connected with $\mathrm{diam}(\mathcal{N}_{n,r}^{\mathrm{NNI}})$ in $\Omega((n+r)\log(n+r))$ and $\mathcal{O}((n+r)^2)$.*

The next two results concern subspaces of $\mathcal{N}_n$ and $\mathcal{T}_n$ that display a set of trees. Bordewich et al. [BLS17, Theorem 6.2] showed that the phylogenetic networks $\mathcal{N}_n^{op}(P)$ that display a set of phylogenetic trees $P \subseteq \mathcal{T}_n$ are connected under SNPR. They also gave the bound $2(t+1)n + r + r'$ on the distance of two networks $N, N' \in \mathcal{N}_n^{op}(P)$ with $r$ and $r'$ reticulations, respectively, and where $t = |P|$. Note that, however, $r$ and $r'$ are unbounded and thus likewise is the diameter.

**Theorem 3.11** (Bordewich et al. [BLS17])**.**
*Let $P \subseteq \mathcal{T}_n$. The graph $\mathcal{N}_n^{\mathrm{SNPR}}(P)$ is connected with unbounded diameter.*

Let $T \in \mathcal{T}_n$, $n > 2$, be a phylogenetic tree on taxa $\mathcal{X}$. A *triplet* $\tau$ is a binary phylogenetic tree on three leaves $\{a, b, c\} \subseteq \mathcal{X}$. We say $T$ *displays* $\tau$ if there exists a subdivision of $\tau$ that is a subgraph of $T$. Let $\mathcal{T}_n(P)$ denote the set of trees in $\mathcal{T}_n$ that display a set $P$ of triplets. Bordewich [Bor03, Proposition 2.9] showed that $\mathcal{T}_n(P)$ is connected under NNI. Mark et al. [MMS16, Theorem 2] pointed out that this implies that also the space $\mathcal{T}_n(P)$ with $P$ being the intersection of triplets displayed by two trees $T, T' \in \mathcal{T}_n$ is connected.

**Theorem 3.12** (Bordewich [Bor03], Mark et al. [MMS16])**.**
*Let $n \geq 3$. Let $P$ be a set of triplets with labels in $\mathcal{X}$. Let $op \in \{\mathrm{NNI}, \mathrm{SPR}\}$.*
*Then the graph $\mathcal{T}_n^{op}(P)$ is connected.*

## 3.3 Tree-child networks

Recall that a network $N$ is tree child if every non-leaf vertex of $N$ has a tree vertex as child. We first look the class of tree-child networks $\mathcal{TC}_n$ under SNPR and then under NNI. Bordewich et al. [BLS17, Proposition 3.2] proved the connectedness of $\mathcal{TC}_n^{\mathrm{SNPR}}$ and bounds on the diameter. We give a sketch of the proof.

**Theorem 3.13** (Bordewich et al. [BLS17])**.**
*The graph $\mathcal{TC}_n^{\mathrm{SNPR}}$ is connected with* $\mathrm{diam}(\mathcal{TC}_n^{\mathrm{SNPR}}) \in \Theta(n)$.

*Proof sketch.* Observe that applying an $\mathrm{SNPR}^-$ operation to a tree-child network preserves the tree-child property. Since further $\mathcal{T}_n \subset \mathcal{TC}_n$, the connectedness thus follows from Lemma 3.3. For $N, N' \in \mathcal{TC}_n$ with $r$ and $r'$ reticulations, respectively, the SNPR-distance is thus at most $r + r' + \mathcal{O}(n)$ by Theorem 3.2. Since $r, r' \in \mathcal{O}(n)$ (see Table 2.2) it follows that $\mathrm{diam}(\mathcal{TC}_n^{\mathrm{SNPR}}) \in \mathcal{O}(n)$. □

The same arguments apply for PR.

**Corollary 3.14.**
*The graph $\mathcal{TC}_n^{\mathrm{PR}}$ is connected with* $\mathrm{diam}(\mathcal{TC}_n^{\mathrm{PR}}) \in \Theta(n)$.

Bordewich et al. [BLS17, Theorem 4.1] further gave the following connectedness result for the tiers of tree-child networks $\mathcal{TC}_{n,r}^{\mathrm{SNPR}}$. Recall that a tree-child network can have at most $n - 1$ reticulations.

**Theorem 3.15** (Bordewich et al. [BLS17])**.**
*Let $n \geq 1$ and $r < n-1$. Then the graph $\mathcal{TC}_{n,r}^{\mathrm{SNPR}}$ is connected with* $\mathrm{diam}(\mathcal{TC}_{n,r}^{\mathrm{SNPR}}) \in \Theta(n)$.

They also gave the more precise upper bound $4n + 12r - 2$ for $\mathrm{diam}(\mathcal{TC}_{n,r}^{\mathrm{SNPR}})$, $r < n - 1$. The situation looks different for the extremal case $r = n - 1$. See again Figure 3.2, where the two tree-child networks $N, N' \in \mathcal{TC}_{2,1}^{\mathrm{SNPR}}$ (the only two networks in $\mathcal{TC}_{2,1}$) cannot be transformed into each other by a single $\mathrm{SNPR}^0$, but only via networks with parallel edges. In fact, we can show that tier $n - 1$ is not connected for any $n > 1$ under SNPR.

**Theorem 3.16.**
*Let $n \geq 2$ and $r = n - 1$. Then the graph $\mathcal{TC}_{n,r}^{\mathrm{SNPR}}$ is not connected.*

*Proof.* Let $N \in \mathcal{TC}_{n,n-1}$ for $n \geq 2$. We know that there always exists a leaf in $N$ that is not a descendant of any reticulation and $n - 1$ leaves that are descendants of reticulations. This is because there exist distinct tree paths from the root and from each of the $n - 1$

reticulations to a leaf. Let $l_1, l_2$ be leaves of $N$ such that $l_1$ is at the end of a tree path from the root and $l_2$ is the descendant of a reticulation $v$. To change with an SNPR$^0$ on $N$ that $l_1$ is not on a tree path from the root or that $l_2$ is not descendant of a reticulation requires that an edge of the tree path from the root to $l_1$ or from $v$ to $l_2$ has to be pruned, respectively. However, both such prunings are not possible with a single SNPR$^0$. Consequently, there is no SNPR$^0$ on $N$ that simultaneously makes $l_1$ a descendant of a reticulation and creates a tree path from the root to $l_2$. Hence, networks in $\mathcal{TC}_{n,n-1}$ with different leaves on the tree path of the root are not connected in $\mathcal{TC}_{n,r}^{\mathrm{SNPR}}$. □

However, Bordewich et al. [BLS17] showed that there exist SNPR-sequences connecting any two $N, N' \in \mathcal{TC}_{n,r}$ of length in $\mathcal{O}(n)$ such that each intermediate network is either tree child or almost tree-child; that is, an intermediate phylogenetic network may have at most one pair of parallel edges such that if one of these edges gets removed, then the resulting phylogenetic network is tree child.

We now show that $\mathcal{TC}_{n,r}^{\mathrm{PR}}$ is also connected in the extremal case $r = n - 1$.

**Theorem 3.17.**
*Let $n \geq 1$ and $r \leq n - 1$. Then the graph $\mathcal{TC}_{n,r}^{\mathrm{PR}}$ is connected with* $\mathrm{diam}(\mathcal{TC}_{n,r}^{\mathrm{PR}}) \in \Theta(n)$.

*Proof.* By Theorem 3.15 and PR generalising SNPR, we only have to prove the case $r = n - 1$. Let $N \in \mathcal{TC}_{n,r}$. By Lemma 4.3 of Bordewich et al. [BLS17], we know that we can transform $N$ into a strict caterpillar network $M \in \mathcal{TC}_n$ with a PR$^0$-sequence of length $\mathcal{O}(n)$ such that each intermediate phylogenetic network is tree child. Our target network is a strict caterpillar network $M^*$. We now show how to find a PR$^0$-sequence that transforms $M$ into $M^*$ by permuting the leaf order $\pi(M)$.

Suppose that $\pi_n(M) \neq \pi_n(M^*)$. For $M$, let $l_i$, $v_i$, $p_i$, and $q_i$ be defined as in the definition of a strict caterpillar network in Section 3.1. Let $v_i$ be the reticulation in $M$ that has leaf $l_i$ with label $\pi_n(M^*)$ as child. The following process is illustrated in Figure 3.3. With a tail PR$^0$ first prune the edge $(p_i, v_i)$ and regraft it to $(q_r, l_n)$. With another tail PR$^0$ prune $(q_i, v_i)$ and regraft it to $(p_i, l_n)$. Let $M_2$ be the resulting network. With a head PR$^0$ prune $(p_i, v_i)$ and regraft it to $(q_i, l_n)$. In the resulting strict caterpillar network $M_3$, the leaf $l_i = \pi^*(n)$ is now the end vertex of the tree path starting at the root of the caterpillar network. Hence, for the leaf order of $M_3$ we have that $\pi_n(M_3) = \pi_n(M^*)$. It is easy to see that each intermediate network in this process is tree-child and has $r$ reticulations.

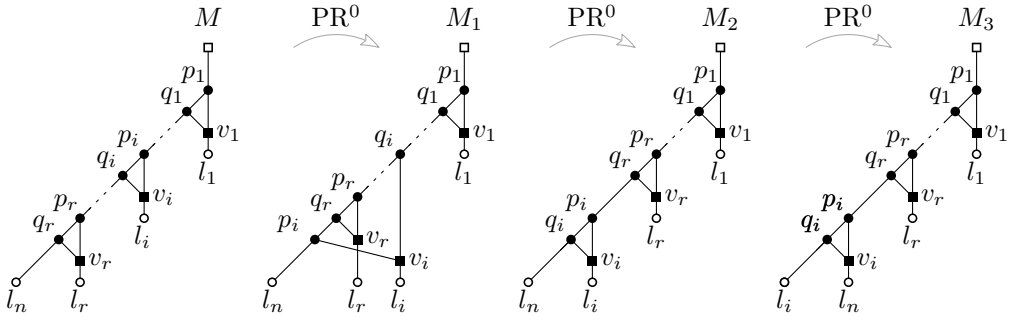


Figure 3.3: Following the proof of Theorem 3.17, illustration of how the leaf $l_n$ at the end of the tree path of a strict caterpillar network $M'$ can be exchanged with another leaf $l_i$.

Next, to transform $M_3$ into $M^*$, a sequence of tail PR$^0$ of length at most $2r$ prunes the two incoming edges of reticulations and regrafts them along the tree path of the root $\rho$ to

achieve the desired leaf order $\pi(M^*)$. Since this transformation works for any such $N$, the connectedness statement follows.

Since we needed at most $\mathcal{O}(n)$ $\mathrm{PR}^0$ from $N$ to $M$ and at most $3+2r$ $\mathrm{PR}^0$ from $M$ to $M^*$ and since $r \in \mathcal{O}(n)$, the diameter of $\mathcal{TC}^{\mathrm{PR}}_{n,r}$ is in $\mathcal{O}(n)$. An example for the lower bound on the diameter can easily be found. For example, two strict caterpillar networks with reversed leaf orders suffice. Hence, the statement on the diameter follows. □

Next, we look at tree-child networks under NNI. Recall that every NNI is a PR, but that there are $\mathrm{NNI}^0$ that are not $\mathrm{SNPR}^0$.

**Theorem 3.18.**
*Let $n \geq 1$ and $r \leq n-1$. Then the graph $\mathcal{TC}^{\mathrm{NNI}}_{n,r}$ is connected with* $\mathrm{diam}(\mathcal{TC}^{\mathrm{NNI}}_{n,r}) \in \mathcal{O}(n^2)$.

*Proof.* Let $N \in \mathcal{TC}_{n,r}$. First, we show how to transform $N$ into a strict caterpillar network $M$ with an $\mathrm{NNI}^0$-sequence. Let $v$ be a reticulation of $N$ that has no reticulation as ancestor. Let $p$ and $q$ be the parents of $v$ such that $p$ is not a descendant of $q$. Note that $p$ and $q$ are tree vertices, since $N$ is a tree-child network. Let $u$ be the lowest common ancestor of $p$ and $q$, or set $u = p$ if $p$ is ancestor of $q$. Let $l$ be a leaf on the end of a tree path from $v$.

We now move $p$ and $q$ upwards such that $p$, $q$ and $v$ form a triangle below the root $\rho$, and then make $l$ the child of $v$. Let $e$ be the incoming edge of $p$. Then as long as $e$ is not incident to $\rho$, apply an $\mathrm{NNI}^0$ with $e$ as axis to move $p$ (and $(p, v)$) closer to $\rho$. Consider one of these $\mathrm{NNI}^0$, as also depicted in Figure 3.4 (a). Let $w \neq v$ be the child of $p$, let $x$ be the parent of $p$, and let $z$ be the parent of $x$. Note that these three vertices are tree vertices. Then each of these $\mathrm{NNI}^0$ gives $x$ the child $w$, $p$ the child $x$, and $z$ the child $p$, while all other vertices keep their children. Therefore each intermediate network is a tree-child network. Note that at the end of this step, $p$ is ancestor of $q$. Next, let $e$ be the incoming edge of $q$. Then as long as $e$ is not incident to $p$, apply an $\mathrm{NNI}^0$ with $e$ as axis to move $q$ (and $(q, v)$) closer to $p$. With the same arguments as for the first step, it follows that every intermediate network is tree child. Note that at the end of this step, $p$, $q$, and $v$ form a triangle. Assume that $l$ is not the child of $v$. Then let $x$ be the child of $v$ and let $y$ be the child of $x$ that is not on the tree path from $v$ to $l$. Then move $(x, y)$ to the outgoing tree edge of $q$ with the $\mathrm{NNI}^0$-sequence illustrated in Figure 3.4 (b) or (c), depending on whether $(x, y)$ is a tree edge or a reticulation edge. We observe that every intermediate network is again tree child. Repeat this step until $l$ is the child of $v$. At the end of this process, we have built the first triangle below the root with a reticulation that has a leaf as child. Since $l$ has at most $\mathcal{O}(n)$ ancestor vertices, this takes at most $\mathcal{O}(n)$ $\mathrm{NNI}^0$. Repeat this process on the pendant network below $q$ to obtain a strict caterpillar network $M$. Overall, this needs at most $\mathcal{O}(n^2)$ $\mathrm{NNI}^0$.

Assume for now that $r = n-1$. Next, we transform $M$ into a strict caterpillar network $M^*$ with a specific leaf order. This can be achieved with the procedure used in the proof of Theorem 3.17 for the same task and as illustrated in Figure 3.3 by replacing the $\mathrm{PR}^0$ that move reticulations edges with $\mathrm{NNI}^0$-sequences. Note that one such $\mathrm{PR}^0$ can be replaced with a sequence of at most $\mathcal{O}(n)$ $\mathrm{NNI}^0$. Since the process in Theorem 3.17 uses at most $\mathcal{O}(n)$ $\mathrm{PR}^0$, $M$ can be transformed into $M^*$ with at most $\mathcal{O}(n^2)$ $\mathrm{NNI}^0$. Note that this process can easily be extended to the case $r < n-1$. Therefore, we have shown that every network in $\mathcal{TC}_{n,r}$ can be transformed into $M^*$ using at most $\mathcal{O}(n^2)$ $\mathrm{NNI}^0$ operations. Hence, $\mathcal{TC}^{\mathrm{NNI}}_{n,r}$ is connected with the proclaimed upper bound on the diameter. □

We now use Theorem 3.18 to prove the connectedness of $\mathcal{TC}_n$ under NNI.

**Theorem 3.19.**
*The graph $\mathcal{TC}^{\mathrm{NNI}}_{n}$ is connected with* $\mathrm{diam}(\mathcal{TC}^{\mathrm{NNI}}_{n}) \in \mathcal{O}(n^2)$.

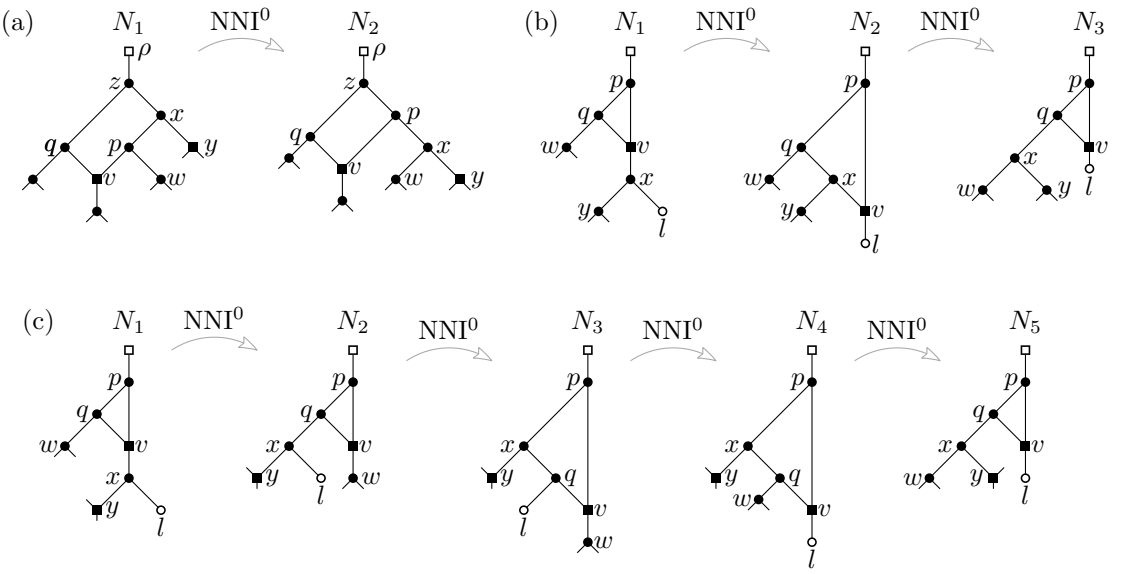


Figure 3.4: Illustration of steps used in the proof of Theorem 3.18. (a) An NNI$^0$ that moves $p$ closer to the root. (b) An NNI$^0$ to move a tree edge $(x, y)$ to $(q, w)$. (c) An NNI$^0$-sequence to move a reticulation edge $(x, y)$ to $(q, w)$.

*Proof.* Let $N$ and $N'$ be in $\mathcal{TC}_n$ with $r$ and $r'$ reticulations, respectively. By Theorem 3.18, we know that $N$ (resp. $N'$) is connected to a strict caterpillar network $M$ (resp. $M'$) with a specific leaf order and $r$ (resp. $r'$) reticulations. By Theorem 3.13, we know that removing a reticulation edge from $M$ yields again a tree-child network. Moreover, there is an NNI$^-$ on $M$ that yields again a strict caterpillar network. Assuming that $r > r'$, we may therefore choose $M$ and $M'$ such that $M'$ can be obtained from $M$ by $r - r'$ NNI$^-$. There is thus a path from $N$ to $N'$ via $M$ and $M'$, which proves the connectedness of $\mathcal{TC}_n^{\text{NNI}}$. To see that the diameter of $\mathcal{TC}_n^{\text{NNI}}$ is in $\mathcal{O}(n^2)$, note that the diameter of the tiers is in $\mathcal{O}(n^2)$ and that the NNI-distance of $M$ and $M'$ is at most $n - 1$ since $r \leq n - 1$. □

## 3.4 Normal networks

Recall that a normal network $N$ is a tree-child network without transitive edges. The results for normal networks $\mathcal{NN}_n$ under SNPR and PR are similar to those for tree-child networks.

**Theorem 3.20.**
*Let $n > 0$. Let $op \in \{\text{SNPR}, \text{PR}\}$.*
*Then the graph $\mathcal{NN}_n^{op}$ is connected with* $\text{diam}(\mathcal{NN}_n^{op}) \in \Theta(n)$.

*Proof.* To apply Lemma 3.3, we only have to show that applying an SNPR$^-$ (or equivalently a PR$^-$) to a normal network yields again a normal network. This is straightforward, as removing a reticulation edge in a normal network cannot create a transitive edge and, by Theorem 3.13, preserves the tree-child property.

The upper bound on the diameter follows from $\text{diam}(\mathcal{T}_n^{\text{PR}}) \in \mathcal{O}(n)$ and the fact that a normal network can have at most $n-2$ reticulations. An example of two normal networks proving the lower bound on the diameter is a tree and a normal network with $n - 2$ reticulations. □

**Theorem 3.21.**
*Let $n > 0$ and $r < n - 2$.*
*Then the graph $\mathcal{NN}_{n,r}^{\mathrm{SNPR}}$ is connected with* $\mathrm{diam}(\mathcal{NN}_{n,r}^{\mathrm{SNPR}}) \in \Theta(n)$.

*Proof.* We prove this by showing that we can transform any $N \in \mathcal{NN}_{n,r}$ into a ladder $M \in \mathcal{NN}_{n,r}$ with an $\mathrm{SNPR}^0$-sequence such that each intermediate network is also normal. This process is illustrated in Figure 3.5.

Let $w$ be the child of the root and $t$ and $t'$ its children. Let $v$ be a reticulation that has no reticulation as ancestor. Let $p$ and $q$ be the parents of $v$. We now "move $v$ up" below $w$ such that $p$ and $q$ become the two children of $w$. Assume therefore for now that $p$, $q$, $t$, and $t'$ are distinct. Further assume, without loss of generality, that $p$ is a descendant of $t$. Note that by the choice of $v$ and since $N$ is normal, neither $p$ nor $q$ is a descendant of both $t$ and $t'$. This implies that $p$ is not a descendant of $t'$. Therefore, pruning $(q, v)$ at $q$ and regrafting it to $(w, t')$ with an $\mathrm{SNPR}^0$ does not create a transitive edge. Note that $q$ is not a descendant of $t$ in the resulting network $N_1$. We can thus prune $(p, v)$ at $p$ and regraft it to $(w, t)$ to obtain a normal network $N_2$. Note that we need less $\mathrm{SNPR}^0$ if $p = t$ or $q = t'$ in $N$. In $N'$ let $t$ and $t'$ be the child of $p$ and $q$ that is not $v$, respectively (see again Figure 3.5). Also let $v_1$ denote $v$.

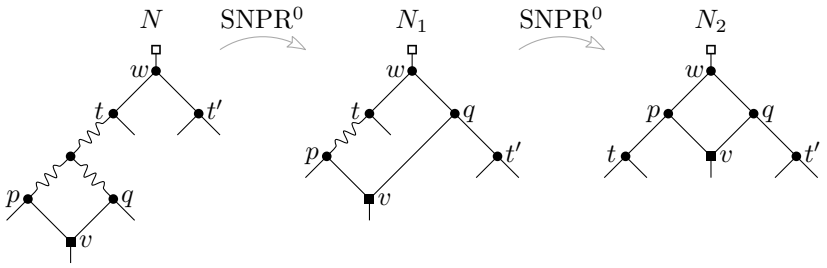


Figure 3.5: Illustration of a step in the proof of Theorem 3.21, showing how a reticulation $v_1$ can be moved directly below $w$, the child of $\rho$. Wobbly lines represent paths of tree edges.

With $v_1$ we have the first "rung" of our ladder. For the next one, let $v_2$ be a reticulation that has either no reticulation as ancestor or at most $v_1$. Note that again neither of the parents of $v_2$ is descendant of both $t$ and $t'$. We can thus move $v_2$ up with an analogous process to $v_1$ with at most two $\mathrm{SNPR}^0$. In general, after moving $v_{i-1}$ up, we use this inductive argument to pick $v_i$ as a reticulation that has most reticulations $v_j$ with $j \in \{1, \ldots, i-1\}$ as ancestors and then apply the same procedure. Let $N'$ be the resulting network. Note that in $N'$ each reticulation has only a pendant tree as descendants. If the pendant tree below any reticulations contains more than one leaf, move all but one leaf of such a pendant tree to the pendant tree below $q_r$. Finally, if the pendant tree $p_r$ contains more than one leaf, we move them again to the pendant tree below $q_r$. Building the pendant tree below $q_r$ takes at most $n$ $\mathrm{SNPR}^0$. Let $M$ be the resulting network, which is a ladder as desired.

Since $r < n - 2$, the pendant tree below $q_r$ in $M$ contains a free leaf. Hence, we can arrange the pendant tree below $q_r$ into a caterpillar and obtain any leaf order in $\mathcal{O}(n)$ $\mathrm{SNPR}^0$ by Lemma 3.5. This proves the connectedness of $\mathcal{NN}_{n,r}^{\mathrm{SNPR}}$. Since this process uses at most $\mathcal{O}(n)$ $\mathrm{SNPR}^0$, the upper bound on the diameter follows. It is easy to see that the diameter is also in $\Omega(n)$ and thus in $\Theta(n)$. □

With the same arguments as in the proof of Theorem 3.16, we get the following theorem.

**Theorem 3.22.**
*Let $n > 2$ and $r = n - 2$. Then the graph $\mathcal{NN}_{n,r}^{\mathrm{SNPR}}$ is not connected.*

Again contrary to SNPR, if we use PR operations, the tier with the maximum number of reticulations is connected.

**Theorem 3.23.**
*Let $n > 0$ and $r \leq n - 2$. Then the graph $\mathcal{NN}_{n,r}^{\mathrm{PR}}$ is connected with* $\mathrm{diam}(\mathcal{NN}_{n,r}^{\mathrm{PR}}) \in \Theta(n)$.

*Proof.* By Theorem 3.21 and PR generalising SNPR, we only have to prove the case $r = n - 2$. Let $N \in \mathcal{NN}_{n,n-2}$ . By the proof of Theorem 3.21, we know that we can transform $N$ into a ladder $M \in \mathcal{NN}_{n,r}$ with a $\mathrm{PR}^0$-sequence of length $\mathcal{O}(n)$ such that each intermediate network is normal. We show how to find a $\mathrm{PR}^0$-sequence that transforms $M$ into a ladder $M^*$ with a fixed leaf order $\pi(M^*)$. For $M$, let $l_i$, $v_i$, $p_i$, and $q_i$ be defined as in the definition of a ladder in Section 3.1.

Assume that neither $\pi_n(M)$ nor $\pi_{n-1}(M)$ equals $\pi_n^*(M^*)$. This implies that the leaf with taxa $\pi_n^*(M^*)$ is the child of a reticulation in $M$. Let $l_i$ be this leaf. The following process is illustrated in Figure 3.6. With a tail $\mathrm{PR}^0$ prune the edge $(p_i, v_i)$ and regraft it to $(p_r, l_{n-1})$. With another tail $\mathrm{PR}^0$ prune $(q_i, v_i)$ and regraft it to $(q_r, l_n)$. Let $M_2$ be the resulting network. With a head $\mathrm{PR}^0$ prune $(p_i, v_i)$ and regraft it to $(q_i, l_n)$. In the resulting ladder $M_3$, we have that $\pi_n(M_3) = \pi_n(M^*)$. Analogously, we can move the leaf that has taxa $\pi_{n-1}(M^*)$, say $l_j$, to position $n - 1$ as illustrated again in Figure 3.6. Each of these two steps takes at most three $\mathrm{PR}^0$ and each intermediate network is normal. Let $M_4$ be the resulting network.

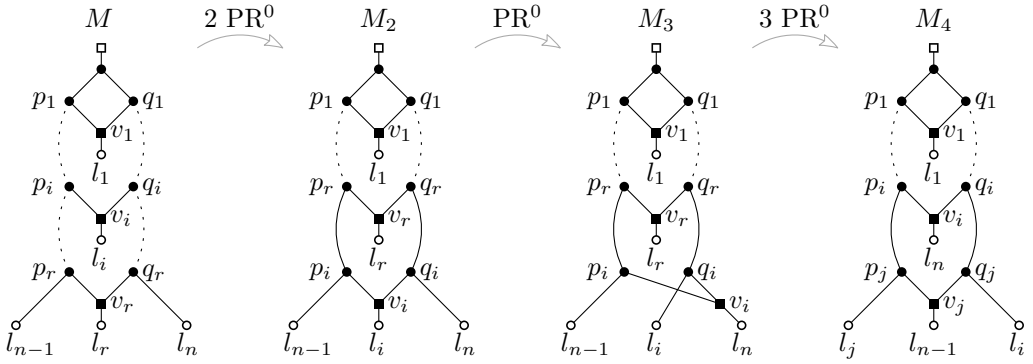


Figure 3.6: Illustration of a step in the proof of Theorem 3.23 showing how the leaves $l_{n-1}, l_n$ at the end of two tree paths of a ladder $M$ can be exchanged with other leaves $l_i, l_j$.

Similar to strict caterpillar networks, we can sort the leaves below reticulations in $M_4$ to match the desired order of $M^*$ with at most $2n - 6$ $\mathrm{PR}^0$. Hence, we need in total at most $2n$ $\mathrm{PR}^0$ and since this transformation works for any such $N$, the connectedness statement and the upper bound for the diameter follow. For the lower bound, note that two ladders with different leaf orders can have a $\mathrm{PR}^0$-distance in $\Omega(n)$. □

Since a normal network $N$ cannot contain a transitive edge, it cannot contain a triangle. Therefore, neither $\mathrm{NNI}^+$ nor $\mathrm{NNI}^-$ can be applied to $N$ and there are no edges between the tiers of $\mathcal{NN}_n^{\mathrm{NNI}}$.

**Theorem 3.24.**
*Let $n > 2$. Then the graph $\mathcal{NN}_n^{\mathrm{NNI}}$ is disconnected.*

## 3.5 Temporal normal networks

Next, we look at temporal normal networks $\mathcal{TP}_n$. Recall that a network is temporal if there is a time function $f\colon V(N) \to \mathbb{N}$ that increases along tree edges and stays the same along reticulation edges. Since a temporal normal network is be definition normal, it has at most $n-2$ reticulations. We start with the proof that $\mathcal{TP}_n$ is connected under SNPR and PR.

**Theorem 3.25.**
*Let $n > 0$. Let $op \in \{\mathrm{SNPR}, \mathrm{PR}\}$.*
*Then the graph $\mathcal{TP}_n^{op}$ is connected with $\mathrm{diam}(\mathcal{TP}_n^{op}) \in \Theta(n)$.*

*Proof.* Let $N \in \mathcal{TP}_n$ with time function $f$. Let $(u, v)$ be a reticulation edge of $N$. We show that the network $N'$ obtained from $N$ by removing $(u, v)$ with an $\mathrm{SNPR}^-$ (or equivalently a $\mathrm{PR}^-$) is a temporal network. Since $N$ is normal, we know by Theorem 3.20 that $N'$ is normal. Furthermore, the two edges incident to $u$ that are not $(u, v)$ are both tree edges, as is the outgoing edge of $v$. Thus $f$ restricted to $V(N') = V(N) \setminus \{u, v\}$ is a time function for $N'$. The connectedness of $\mathcal{TP}_n^{op}$ thus follows from Lemma 3.3. For the diameter the same arguments as in Theorem 3.20 apply. □

The results concerning the connectedness of a tier of $\mathcal{TP}_n$ are similar to the results of normal networks. In the extremal case $r = n-2$, the graph induced by SNPR is not connected. This can be proven with analogous arguments to those used for Theorems 3.16 and 3.22.

**Theorem 3.26.**
*Let $n > 2$ and $r = n-2$. Then the graph $\mathcal{TP}_n^{\mathrm{SNPR}}$ is not connected.*

**Theorem 3.27.**
*Let $n > 2$ and $r < n-2$. Then the graph $\mathcal{TP}_{n,r}^{\mathrm{SNPR}}$ is connected with $\mathrm{diam}(\mathcal{TP}_{n,r}^{\mathrm{SNPR}}) \in \Theta(n)$.*

*Proof.* Let $N \in \mathcal{TP}_{n,r}$. With the same process to the one for normal networks (Theorem 3.21), we transform $N$ into a ladder. However, we pick the reticulations we "move up" more carefully. We choose $v_1$ as a reticulation such that neither parent of $v_1$ is a descendant of a parent of another reticulation. Such $v_1$ exists as $N$ is temporal. Move $v_1$ up and let $N'$ be the resulting network. The intermediate network and $N'$ are temporal normal networks by the choice of $v_1$ (and because we know they are normal). Next, note that in $N'$ there exists a reticulation $v_2$ whose parents are only descendants of the parents of $v_1$ but not of a parent of another reticulation. Again, such $v_2$ exists as $N$ is temporal. Move $v_2$ up and pick the subsequent reticulations $v_3, \ldots, v_r$ with analogous conditions. Sorting the leaves of the resulting ladder works like for normal networks. This proves the connectedness. The arguments regarding the diameter are as for normal networks. □

**Theorem 3.28.**
*Let $n > 2$ and $r \leq n-2$. Then the graph $\mathcal{TP}_{n,r}^{\mathrm{PR}}$ is connected with $\mathrm{diam}(\mathcal{TP}_{n,r}^{\mathrm{PR}}) \in \Theta(n)$.*

*Proof.* For $r < n-2$, this follows from Theorem 3.27 and only the case $r = n-2$ remains. Let $N \in \mathcal{TP}_{n,n-2}$. By the proof of Theorem 3.27, we know that we can transform $N$ into a ladder $M \in \mathcal{TP}_{n,n-2}$ with a $\mathrm{PR}^0$-sequence of length $\mathcal{O}(n)$ such that each intermediate phylogenetic network is in $\mathcal{TP}_{n,n-2}$. We now show how to find a $\mathrm{PR}^0$-sequence that transforms $M$ into a ladder $M^*$ with a fixed leaf order $\pi(M^*)$. For $M$, let $l_i$, $v_i$, $p_i$, and $q_i$ be defined as in the definition of a ladder.

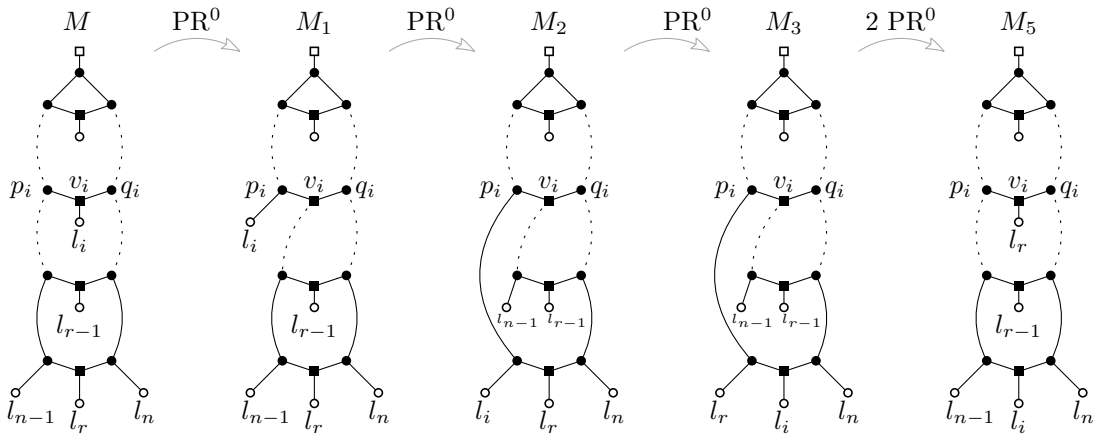


Figure 3.7: Illustration of a step in the proof of Theorem 3.28 showing how the leaves $l_i$ and $l_r$ can be swapped with five $\mathrm{PR}^0$. The last two $\mathrm{PR}^0$ reverse the topological changes made by the first two.

Suppose we want to move leaf $l_i$ of $M$ to where $l_r$ (or $l_n$ or $l_{n-1}$) is. Figure 3.7 illustrates how to exchange the leaves $l_i$ and $l_r$ with five $\mathrm{PR}^0$. Observe that each network is in $\mathcal{TP}_{n,r}$. With an additional $\mathrm{PR}^0$ we can swap $l_i$ with $l_n$ or $l_{n-1}$. Furthermore, the same process can be used to swap $l_i$ with $l_j$ for $j < r$. Therefore, we can transform $M$ int $M^*$ with at most $5n + 2$ $\mathrm{PR}^0$. This proves the connectedness and an upper bound on the diameter of $\mathcal{TP}_{n,r}^{\mathrm{PR}}$. For a lower bound, note again that two ladders with different leaf orders can have a $\mathrm{PR}^0$-distance in $\Omega(n)$. □

## 3.6 Tree-sibling networks

Recall that a network $N$ is a tree-sibling network if each reticulation $v$ of $N$ has a tree vertex $w$ as sibling. In this case, $w$ is the (tree-sibling) witness of $v$. Note that $w$ can be witness of at most one reticulation. A vertex $u$ is called a *witness parent* of $v$ if it is a parent of both $v$ and a witness $w$ of $v$. Note that a tree-sibling network contains no parallel edges and that none of its reticulations can be the child of two other reticulations.

**Theorem 3.29.**
*Let $op \in \{\mathrm{SNPR}, \mathrm{PR}\}$. The graph $\mathcal{TS}_n^{op}$ is connected with unbounded diameter.*

*Proof.* Let $N \in \mathcal{TS}_n$. Let $v$ be a reticulation of $N$ with no reticulation as a descendant. Let $w$ be a tree-sibling witness for $v$ and let $u$ be their witness parent. Let $N'$ be the network obtained from $N$ by the $\mathrm{SNPR}^-$ that removes $(u, v)$. Note that if $u$ was a witness of a reticulation $v'$, then $w$ is a witness of $v'$ in $N'$. Since all other reticulations keep their witnesses if $N'$, it follows that $N'$ is a tree-sibling network. Therefore, the connectedness of $\mathcal{TS}_n^{op}$ follows from Lemma 3.3. The diameter is unbounded by Lemma 3.4. □

The following lemma will help us prove the connectedness of the tiers of $\mathcal{TS}_n$.

**Lemma 3.30.**
*Let $N \in \mathcal{TS}_n$ with $r \geq 1$.*
*An $\mathrm{SNPR}^0$ that prunes a reticulation edge $(u, v)$ at a witness parent $u$ of $v$ and regrafts it to a tree edge yields a tree-sibling network $N'$.*

*Proof.* Let $N'$ be the resulting network of such an $\mathrm{SNPR}^0$. Suppose the $\mathrm{SNPR}^0$ regrafted to an edge $(x, y)$. Then $y$ is a witness of $v$ in $N'$. Let $w$ be the witness of $v$ via witness

parent $u$ in $N$. If $u$ is a witness of a reticulation $v'$ in $N$, then $w$ is a witness of $v'$ in $N'$. Since all other reticulations keep their witnesses, it follows that $N' \in \mathcal{TS}_n$. □

**Theorem 3.31.**
*Let $n \geq 3$ and $r \geq 0$. Let $op \in \{\text{SNPR}, \text{PR}\}$.*
*Then the graph $\mathcal{TS}^{op}_{n,r}$ is connected with $\operatorname{diam}(\mathcal{TS}^{op}_{n,r}) \in \Theta(n+r)$.*

*Proof.* Let $N \in \mathcal{TS}_{n,r}$. We prove this for SNPR by showing how to transform $N$ into a stack network $M$ within $\mathcal{TS}_{n,r}$. The process for this consists of five steps, which are illustrated in Figure 3.8. In the first two steps, we create a path $P_w$ containing at least one witness per reticulation and a path $P_r$ containing all reticulations, respectively. In the third step, the witnesses on $P_w$ are sorted according to the order of their reticulations on $P_r$. Next, we move any pendant trees attached to $P_r$ to the tail to obtain a stack network. In the fifth and last step, we transform the tail of the stack network into a caterpillar and sort the leaves.

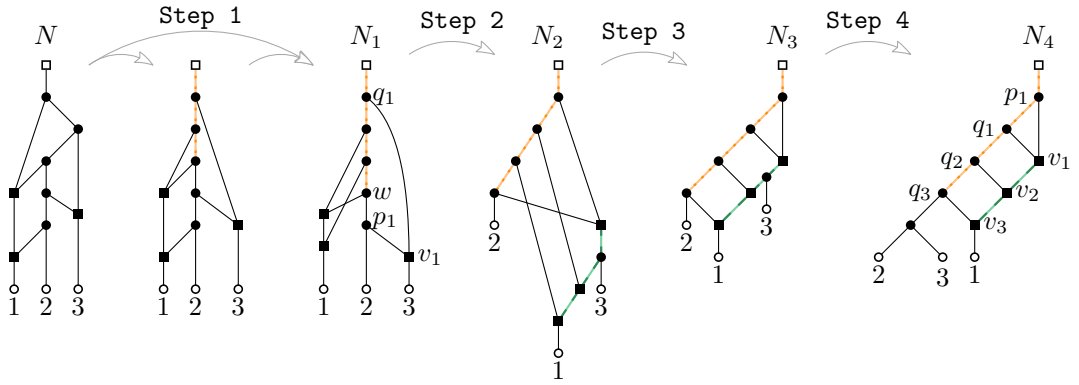


Figure 3.8: Following the proof of Theorem 3.31, illustration of how a tree-sibling network $N$ can be transformed into a stack network $N_4$. In $N_1$ all witnesses are on a path, in $N_2$ all reticulations are on a path, and in $N_3$ the order of the reticulations and their witnesses is the same.

`Step 1:` We apply SNPR$^0$ to create a path containing a witness of each reticulation of $N$ as follows. Let $v$ be a reticulation of $N$ with no ancestral reticulation and let $q$ be a witness parent of $v$. Prune $(q, v)$ at $q$ and regraft it to the root edge. The resulting network $N'$ is a tree-sibling network by Lemma 3.30. Let $P_w$ be the path consisting of $\rho$, $q$, and $w \neq v$ (the witness of $v$). For a reticulation $v'$ that has no witness on $P_w$, prune the reticulation edge $(q', v')$ at a witness parent $q'$ of $v'$, and regraft it to the last edge on $P_w$, which then contains one more witness. Repeat this until $P_w$ contains a witness of each reticulation. By Lemma 3.30 each such SNPR$^0$ results in a tree-sibling network. Let $N_1$ be the resulting network. Overall, since we only need one witness per reticulation on $P_w$, Step 1 needs at most $r$ SNPR$^0$.

`Step 2:` We apply SNPR$^0$ to $N_1$ to create a path containing all reticulations while maintaining the tree-sibling property and $P_w$ as follows. For each reticulation $v_i$ let $q_i$ be the (witness) parent of $v_i$ that lies on $P_w$ and let $p_i$ be the second parent. If a reticulation $v_i$ has both parents on $P_w$, let $q_i$ be the parent that is an ancestor of $p_i$.

Consider when we can prune an edge $(p_i, v_i)$ at $p_i$, for example, to make $v_i$ a descendant of another reticulation. This is not possible, if $p_i$ is a reticulation, say $v_j$. However, then $v_i$ and $v_j$ already lie on a path. If $p_i$ is the parent of two reticulations, say $v_i$ and $v_j$, and part of a triangle that does not contain $v_i$, then pruning $(p_i, v_i)$ would result in a pair of parallel edges. However, note that this may occur only once in $N_1$, namely, only if the

last two vertices on $P_w$ are the two parents of $v_j$ (see for example the second network in Figure 3.10). Furthermore, if $p_i$ is the last vertex of $P_w$ and parent of two reticulations, then pruning $(p_i, v_i)$ results in the loss of the last witness of $P_w$. We call this the special case. In all other cases, we may prune $(p_i, v_i)$ knowing that that $P_w$ and the tree-sibling property is maintained.

Let $H$ be the underlying graph of the Hasse diagram of the ancestor relation graph of the reticulations of $N_1$; that is, the vertices of $H$ represent the reticulations of $N_1$ and there is an edge $(u, v)$ if $u$ is an ancestor of $v$ and there is no reticulation $w$ that is a descendant of $u$ and an ancestor of $v$. Note that the connected components of $H$ are rooted trees, since in $N_1$ each reticulation $v_i$ can only have a reticulation as ancestor via the edge $(p_i, v_i)$ but not via the edge $(q_i, v_i)$. See Figures 3.9 and 3.10 for examples.

Knowing that we can prune any reticulation that is not the child of another reticulation (except for the special case), we transform each component of $H$ into a path as follows and as illustrated in Figure 3.9. Suppose two reticulations $v_i$ and $v_j$ are siblings in $H$. This implies that $p_i$ is a tree vertex. We can thus prune $(p_i, v_i)$ at $p_i$ and regraft it to the outgoing edge of $v_j$. In the resulting network, $v_i$ is a child of $v_j$ in $H$.

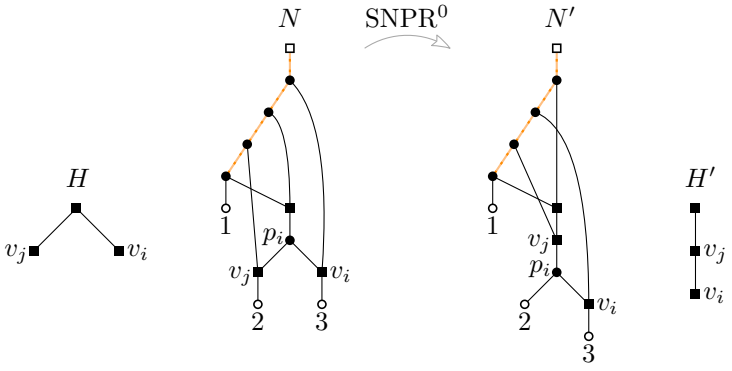


Figure 3.9: Illustration of how a component of the graph $H$ of a tree-sibling network $N$ can be made a path, for the proof of Theorem 3.31.

Next, we want to join these paths of $H$ of the resulting network into one long path $P_r$. First, suppose the special case occurs; that is, the last vertex $w$ of $P_w$ is also $p_i$ and $p_j$ of two reticulations $v_i$ and $v_j$. Note that in this case $v_i$ and $v_j$ are the topmost reticulations of two paths $Q$ and $Q'$ of $H$ (as indicated in Figure 3.10). We then follow the process illustrated with a minimal example in Figure 3.10 to concatenate $Q$ and $Q'$. Note that this situation can occur at most once, since there is now a pendant tree below $w$ (which is leaf 2 in the figure). After we have handled the special case, we merge the remaining paths of $H$ (of the resulting network) as follows. Suppose we want to merge path $Q'$ to $Q$. Let $v_i$ be the top reticulation of $Q'$ and $v_j$ the lowest reticulation of $Q$. Then prune $(p_i, v_i)$ at $p_i$ and regrafting it to the outgoing edge of $v_j$. Since the special case occurs only once and at most $r - 1$ reticulations have to be merged to another path, this steps needs at most $r + 1$ SNPR$^0$. Let $N_2$ be the resulting network.

`Step 3:` The order of the reticulations $v_i$ on $P_r$ might differ from the order of their witnesses $q_i$ on $P_w$. Let $(v_1, \ldots, v_r)$ be the order of the reticulations in $P_r$ in $N_2$. Prune $(p_1, v_1)$ at $p_1$ and regraft it to the root edge. Note that this maintains $P_w$, since either $p_1$ is not on $P_w$ in the first place or $p_1$ has a tree vertex as child that becomes the new last vertex of $P_W$ (see again Figure 3.8). Next, prune $(q_1, v_1)$ at $q_1$ and regraft it at the outgoing edge of $p_1$ that is in $P_w$. Then, for $i \in \{2, \ldots, r\}$, prune $(q_i, v_i)$ at $q_i$ and regraft it at the outgoing edge of $q_{i-1}$ that is in $P_w$. Lemma 3.30 ensures that each intermediate network is a tree-sibling network. This step takes at most $r + 1$ SNPR$^0$.

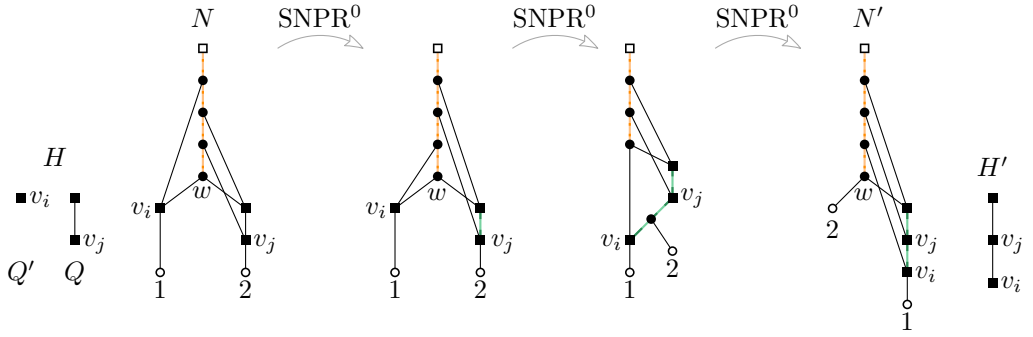


Figure 3.10: Illustration of how two reticulation paths $Q$ and $Q'$ of $H$ can be joined where their topmost reticulations have the same parent $w$ that is also the last vertex of the witness path, for the proof of Theorem 3.31.

`Step 4:` The path $P_r$ might contain tree vertices between the reticulations. It is easy to see that each such tree vertex $u$ is the root of a pendant subtree. We can prune each such subtree and regraft it to the outgoing tree edge of the last vertex of $P_w$. Moreover, with further SNPR$^0$ we can reduce the pendant subtree below the last reticulation $v_r$ of $P_r$ to a single leaf. In total this needs at most $n-1$ SNPR$^0$. The resulting network is a stack and $P_w$ ensures that each intermediate network is a tree-sibling network.

`Step 5:` Since $n \geq 3$, we can now straightforwardly transform the tail of the stack into a caterpillar. Furthermore, Lemma 3.5 applies and we can sort the leaves to a leaf order of our choice. The resulting network is our target network $M$. This step can be done in $n$ SNPR$^0$.

In total, $2n+2r$ SNPR$^0$ suffice to transform each tree-sibling network $N \in \mathcal{TS}_{n,r}$ into $M$, a stack with a caterpillar as tail and a particular leaf order. Thus the connectedness and the upper bound on the diameter follow. For the lower bound, note that for example the two networks $N_1$ and $N_2$ with $r = 4$ reticulations in Figure 3.11 have an SNPR-distance in $\Omega(n + r)$, since from $N_1$ to $N_2$ the stack has to be formed with $\Omega(r)$ SNPR$^0$ and the leaves have to be sorted with $\Omega(n)$ SNPR$^0$. This also holds for PR. □

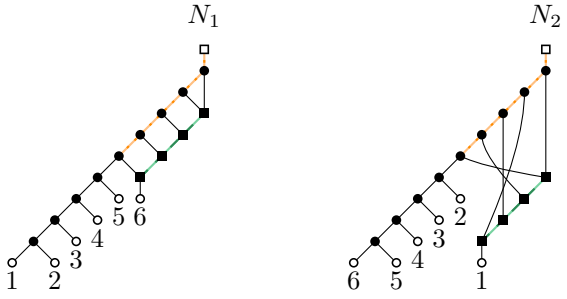


Figure 3.11: Two tree-sibling networks of $\mathcal{TS}_{6,4}$ that can be generalised to two tree-sibling networks in $\mathcal{TS}_{n,r}$ that have SNPR- and PR-distance in $\Omega(n + r)$.

## 3.7 Reticulation-visible networks

A phylogenetic network without parallel edges is a reticulation-visible network if for every reticulation $v$ there is a leaf $l$ such that every path from the root to $l$ goes through $v$. In this case we say that $l$ is a witness for $v$. Note that in a reticulation-visible network no

reticulation $u$ has another reticulation $v$ as child, since then $u$ would not be visible. We observe that removing a reticulation edge from a reticulation-visible network maintains the reticulation-visible property, which gives us the following results.

**Theorem 3.32** (Bordewich et al. [BLS17])**.**
*The graph* $\mathcal{RV}_n^{\mathrm{SNPR}}$ *is connected with* $\mathrm{diam}(\mathcal{RV}_n^{\mathrm{SNPR}}) \in \Theta(n)$.

**Corollary 3.33.**
*The graph* $\mathcal{RV}_n^{\mathrm{PR}}$ *is connected with* $\mathrm{diam}(\mathcal{RV}_n^{\mathrm{PR}}) \in \Theta(n)$.

For the tiers of $\mathcal{RV}_n$, Bordewich et al. [BLS17] included reticulation-visible networks that allow parallel edges. Let ${}^*\mathcal{RV}_{n,r}$ denote these supersets of $\mathcal{RV}_{n,r}$.

**Theorem 3.34** (Bordewich et al. [BLS17])**.**
*The graph* ${}^*\mathcal{RV}_{n,r}^{\mathrm{SNPR}}$ *is connected with* $\mathrm{diam}({}^*\mathcal{RV}_{n,r}^{\mathrm{SNPR}}) \in \mathcal{O}(nr)$.

The situation looks different for $\mathcal{RV}_{n,r}$ when $r$ is $3n-3$ and thus maximal. In this case, $\mathcal{RV}_{n,r}^{\mathrm{PR}}$ is not only disconnected but also contains networks that are not adjacent to any other network.

**Lemma 3.35.**
*Let* $n \geq 2$ *and* $r = 3n-3$*. Let* $op \in \{\mathrm{NNI}, \mathrm{SNPR}, \mathrm{PR}\}$.
*Then the graph* $\mathcal{RV}_{n,r}^{op}$ *is disconnected.*

*Proof.* Let $N_n$ be the network obtained as follows. For $n = 2$ and 3, let $N_n$ be as in Figure 3.12. For $n \geq 4$, obtain $N_n$ from $N_{n-1}$ in the same way as $N_3$ extends $N_2$. We now show that no edge can be pruned and regrafted in $N_n$ with a $\mathrm{PR}^0$ to obtain a reticulation-visible network different from $N_n$. We start with tail $\mathrm{PR}^0$. Here we can rule out all edges that cannot be pruned because their tail is the root or a reticulation or because pruning them creates a pair of parallel edges or a reticulation with a reticulation as child. This leaves only edges of triangles. Let $u$, $v$, $w$ be vertices that form a triangle with the edges $(u, v)$, $(u, w)$, and $(v, w)$. Let $p$ be the parent of $u$, which is either a reticulation or $\rho$. We observe that a tail $\mathrm{PR}^0$ on $(u, v)$, $(u, w)$, or $(v, w)$ either yields again $N_n$ or $p$ becomes non-visible if $p \neq \rho$. This is the case since an edge can only be regrafted to an edge that is not a descendant of it. Next, consider a head $\mathrm{PR}^0$ and note that in order to not create a reticulation with a reticulation as child, a head $\mathrm{PR}^0$ can regraft only to a pure tree edge. The only pure tree edges in $N_n$ are tree edges of triangles, like $(u, v)$. Observe that regrafting a reticulation edge to $(u, v)$ with a head $\mathrm{PR}^0$, either yields a reticulation that is non-visible or makes $p$ non-visible. Hence, the network $N_n$ is isolated in $\mathcal{RV}_{n,n-3}^{\mathrm{PR}}$ and the space disconnected.

Since every $\mathrm{NNI}^0$ and $\mathrm{SNPR}^0$ is a $\mathrm{PR}^0$, the statement also holds for NNI and SNPR. □

Note that the arguments of the proof of Lemma 3.35 also apply for SNPR and the networks $M_n \in \mathcal{RV}_{n,3n-5}$ as shown in Figure 3.12.

**Theorem 3.36.**
*Let* $n \geq 2$ *and* $r \leq n-2$*. Let* $op \in \{\mathrm{SNPR}, \mathrm{PR}\}$.
*Then the graph* $\mathcal{RV}_{n,r}^{op}$ *is connected with* $\mathrm{diam}(\mathcal{RV}_{n,r}^{op}) \in \mathcal{O}(n^2)$.

*Proof.* We prove this by showing that we can transform any $N \in \mathcal{RV}_{n,r}$ into a caterpillar network $M \in \mathcal{RV}_{n,r}$ with an $\mathrm{SNPR}^0$-sequence such that each intermediate network is also a reticulation-visible network. This also implies the result for PR. Note that in $M$ every reticulation has exactly one witness, namely its child, and that no two reticulations share

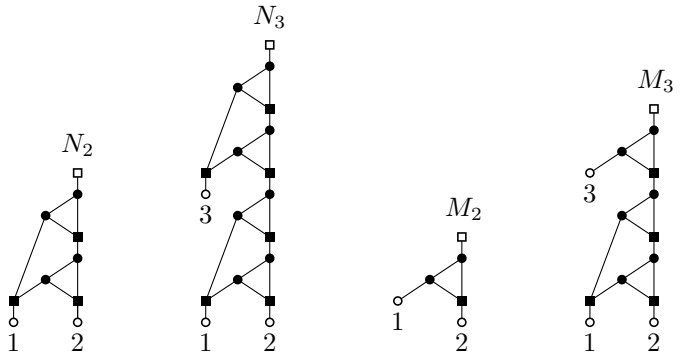


Figure 3.12: The two networks $N_2$ and $N_3$ are isolated vertices in $\mathcal{RV}^{\mathrm{PR}}_{n,3n-3}$ for $n = 2$ and $n = 3$. Similarly, $M_2$ and $M_3$ are isolated vertices in $\mathcal{RV}^{\mathrm{SNPR}}_{n,3n-5}$.

a witness. Our strategy is to repeatedly obtain a free leaf that we can then use to create a triangle.

For $N$, fix an assignment of leaves to reticulations for which they are witnesses such that each reticulation is assigned only one leaf. In particular, if a reticulation $v$ has witnesses $l$ and $l'$ where $l'$ lies on a path from $v$ through another reticulation but where $l$ does not, then pick $l$ as witness for $v$. Let $W$ be the set of leaves of $N$ that are witnesses under this assignment. Note that $|W| \leq r \leq n-2$ and hence there are at least two leaves $l$ and $l'$ that are not assigned as witnesses. Note that $l$ (and $l'$) is a free leaf unless pruning its incident edge would would create a pair of parallel edges. Suppose neither $l$ nor $l'$ is a free leaf. We now show how to change this. For this, let $v$ be a reticulation that is part of a triangle and has leaf $l$ as sibling. Let $p$ and $q$ be the parents of $v$. Note that since $l$ is not an assigned witness and by the choice of our witness assignment, the witness of $v$ is not the witness of an ancestral reticulation of $v$. Therefore, if we prune $(p, v)$ at $p$ with an $\mathrm{SNPR}^0$ and regraft it at the outgoing edge of the root, every reticulation keeps its assigned witness. Furthermore, it ensures that $l$ does not become a witness. We can do the same for $(q, v)$ and thus form form a triangle below the root with $v$, $p$ and $q$. Apply this procedure alternatingly to $l$ and $l'$ until either $l$ or $l'$ can be pruned without creating parallel edges. Let $N'$ be the resulting network. Note that this procedure terminates at least when all reticulations form triangles chained up below the root, since then only of $l$ and $l'$ can be sibling of a reticulation in a triangle. This takes at most $\mathcal{O}(r) = \mathcal{O}(n)$ $\mathrm{SNPR}^0$. Without loss of generality, assume that $l$ is now a free leaf.

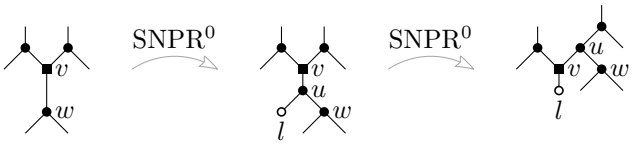


Figure 3.13: $\mathrm{SNPR}^0$-sequence to give a reticulation $v$ a leaf $l$ as witness and child.

Next, continuing from $N'$, we use $\mathrm{SNPR}^0$ to make the witness of each reticulation also its child. Let $v$ be a reticulation whose child is not a leaf. Prune the incident edge of the free leaf $l$ and regrafted it below $v$ with an $\mathrm{SNPR}^0$. Update $W$ such that $l$ is now the witness of $v$ and of any ancestral reticulation of $v$ that had the same witness as $v$. In the resulting network, let $u$ be the child of $v$. Apply another $\mathrm{SNPR}^0$ to the outgoing edge of $u$ that is not incident to $l$ and regraft it above $v$. This is also shown in Figure 3.13. Repeat the procedure from above to obtain another free leaf $l''$ that is not in $W$. We can then use $l''$ for the next reticulation. Repeat this procedure for each reticulation whose child is not

a leaf. This can be done with $\mathcal{O}(n^2)$ SNPR$^0$. Let $N''$ be the resulting network.

In the last step, we arrange the reticulations of $N''$ as triangles chained up below the root to form the caterpillar network. More precisely, we want to create a path $\rho$, $p_1$, $q_1$, $\ldots$, $p_r$, $q_r$ where $p_i$ and $q_i$ are the parents of $v_i$, for $i \in \{1, \ldots, r\}$, for some ordering of the reticulations. Suppose we already have the path $P$ consisting $\rho$, $p_1$, $q_1$, $\ldots$, $p_k$, $q_k$ and $k < r$. Let $e$ be the outgoing tree edge of $p_k$. When extending the path for the remaining reticulations we only have to make sure to not create parallel edges, since every reticulation has its witness as child. Let $l$ be a leaf not in $W$. If $l$ is a not a free leaf, then we use the procedure above to make $l$ a free leaf, which also extends $P$. Therefore, suppose that $l$ is a free leaf and $P$ does not contain the parents of a reticulation $v_i$. If pruning $(p_i, v_i)$ and $(q_i, v_i)$ does not create a pair of parallel edges, then we can extend $P$ with $v_i$. So assume otherwise. Then prune the outgoing edge of $l$ and regraft it to the $(p_i, w)$ where $w \neq v_i$. Then we can prune $(p_i, v_i)$ and regraft it to $e$. We do the same for $q_i$. This adds $p_i$ and $q_i$ (and the parents of other reticulations if we have to free $l$) to $P$. Repeat this until we reach a caterpillar network $M$. This step takes at most $\mathcal{O}(r) = \mathcal{O}(n)$ SNPR$^0$.

Since $r \leq n - 2$, the pendant tree below $q_r$ in $M$ contains a free leaf. Hence, we can arrange the pendant tree below $q_r$ into a caterpillar and obtain any leaf order in $\mathcal{O}(n)$ SNPR$^0$ by Lemma 3.5. This proves the connectedness of $\mathcal{RV}_{n,r}^{\mathrm{SNPR}}$. The diameter is in $\mathcal{O}(n^2)$ since each of the steps above needs at most $\mathcal{O}(n^2)$ SNPR$^0$. □

## 3.8 Tree-based networks

A tree-based network $N$ has a base tree $T$ such that $T$ has an embedding into $N$ that covers all vertices of $N$. Assuming that $N$ has $r$ reticulations, this implies that $N$ can be partitioned into the edges covered by an embedding of $T$ and $r$ edges $e_1, \ldots, e_r$ that are vertex-disjoint. Furthermore, note that no such $e_i$ is incident to a leaf or the root.

Bordewich et al. [BLS17] considered the connectedness of tree-based networks under SNPR.

**Theorem 3.37** (Bordewich et al. [BLS17])**.**
*The graph $\mathcal{TB}_n^{\mathrm{SNPR}}$ is connected with unbounded diameter.*

**Theorem 3.38** (Bordewich et al. [BLS17])**.**
*Let $T \in \mathcal{T}_n$.*
*The graphs $\mathcal{TB}_{n,r}^{\mathrm{SNPR}}$ and $\mathcal{TB}_{n,r}^{\mathrm{SNPR}}(T)$ are connected with diameters* $\mathrm{diam}(\mathcal{TB}_{n,r}^{\mathrm{SNPR}}) \in \mathcal{O}(nr)$ *and* $\mathrm{diam}(\mathcal{TB}_{n,r}^{\mathrm{SNPR}}(T)) \in \mathcal{O}(nr)$.

The connectedness results of Theorems 3.37 and 3.38 also hold for PR. However, we can improve the bounds on the diameter for the tiers of $\mathcal{TB}_n$ under PR, with and without a fixed base tree.

**Corollary 3.39.**
*The graph $\mathcal{TB}_n^{\mathrm{PR}}$ is connected with unbounded diameter.*

**Lemma 3.40.**
*Let $T \in \mathcal{T}_n$. The graph $\mathcal{TB}_{n,r}^{\mathrm{PR}}(T)$ is connected with diameter* $\mathrm{diam}(\mathcal{TB}_{n,r}^{\mathrm{PR}}(T)) \leq 2r$.

*Proof.* Let $N, N' \in \mathcal{TB}_{n,r}(T)$. We show how to transform $N$ into $N'$. Consider an embedding of $T$ into $N$ and an embedding of $T$ into $N'$. Let $S$ (resp. $S'$) be the set of all edges not covered by this embedding in $N$ (resp. $N'$). Since $N$ is tree-based, note that $S = \{e_1, \ldots, e_r\}$ consists of vertex-disjoint impure reticulation edges $e_i = (u_i, v_i)$. Therefore, each $e_i$ can be pruned at both end vertices. For $i \in \{1, \ldots, r\}$, we move $e_i$ with a

tail $\mathrm{PR}^0$ and a head $\mathrm{PR}^0$ from $N$ to where $e'_i$ is in $N'$ with respect to the embeddings of $T$. This requires at most $2r$ $\mathrm{PR}^0$. Since $N$ and $N'$ were picked arbitrary, this implies the connectedness of $\mathcal{TB}^{\mathrm{PR}}_{n,r}(T)$ and concludes the proof. □

**Lemma 3.41.**
*The graph $\mathcal{TB}^{\mathrm{PR}}_{n,r}$ is connected with diameter* $\mathrm{diam}(\mathcal{TB}^{\mathrm{PR}}_{n,r}) \in \Theta(n+r)$.

*Proof.* Let $N \in \mathcal{TB}_{n,r}$ with base tree $T$. We show how to transform $N$ into a target network $M$ that has a caterpillar $T'$ as base tree and all reticulations in series below the root. Like in the proof of Lemma 3.40, obtain a set $S = \{e_1, \dots, e_r\}$ via an embedding of $T$ into $N$. For $i \in \{1, \dots, r\}$, first prune $e_i$ with a tail $\mathrm{PR}^0$ and regraft it to the outgoing edge of the root and then prune $e_i$ with a head $\mathrm{PR}^0$ and regraft it such that it forms a pair of parallel edges. This requires at most $2r$ $\mathrm{PR}^0$. The resulting network $N'$ is now not only tree-based on $T$ but contains $T$ as pendant subtree. The same holds for $M$ and $T'$ and therefore, by Theorem 3.2, $N'$ can be transformed into $M$ in $\mathcal{O}(n)$. Hence, the connectedness and that $\mathrm{diam}(\mathcal{TB}^{\mathrm{PR}}_{n,r}) \in \mathcal{O}(n+r)$ follows.

For the lower bound on the diameter, take two trees $T, T' \in \mathcal{T}_n$ such that their PR-distance is in $\Omega(n)$. This is possible by Theorem 3.2. Obtain a network $N$ from $T$ by adding $r$ reticulations in series below the root. Obtain a network $N'$ from $T'$ by subdividing the root edge $2r$ times with vertices $u_1, \dots, u_r, v_1, \dots, v_r$ and adding the edges $(u_i, v_i)$ for $i \in \{1, \dots, r\}$. Note that $N$ and $N'$ are in $\mathcal{TB}_{n,r}$ and have PR-distance in $\Omega(n+r)$. □

Next, we consider spaces of tree-based networks under NNI.

**Theorem 3.42.**
*Let $T \in \mathcal{T}_n$.*
*The graph $\mathcal{TB}^{\mathrm{NNI}}_{n,r}(T)$ is connected with diameter* $\mathrm{diam}(\mathcal{TB}^{\mathrm{NNI}}_{n,r}(T)) \in \mathcal{O}(nr + r^2)$.

*Proof.* Let $N \in \mathcal{TB}_{n,r}(T)$. Fix an embedding of $T$ into $N$ and let $S = \{e_1, \dots, e_r\}$ be the set of edges not covered by this embedding. Let $e_i = (u_i, v_i)$. We show how to transform $N$ into the network $M \in \mathcal{TB}_{n,r}(T)$ that has its reticulations in series below the root. For this, note that $u_i$ and $v_i$ can be moved upwards along the edges of the embedding of $T$ into $N$ with $\mathrm{NNI}^0$ (as long as $v_i$ does not become an ancestor of $u_i$). Also note that each network obtained from such an $\mathrm{NNI}^0$ is again in $\mathcal{TB}_{n,r}(T)$ since this embedding of $T$ is maintained. Now, towards $M$, first move $u_1$ upwards with $\mathrm{NNI}^0$ below the root and then move $v_1$ upwards with $\mathrm{NNI}^0$ below $u_1$. The reticulation $v_1$ is then in parallel below the root. This takes at most $\mathcal{O}(n+r)$ $\mathrm{NNI}^0$, since $N$ contains $\mathcal{O}(n+r)$ vertices. For $i \in \{2, \dots, r\}$, move $u_i$ upwards below $v_{i-1}$ and then $v_i$ below $u_i$. Repeat this process for $i = \{2, \dots, r\}$ to obtain $M$. Overall, this takes at most $\mathcal{O}(nr + r^2)$ $\mathrm{NNI}^0$. This proves the connectedness of $\mathcal{TB}^{\mathrm{NNI}}_{n,r}(T)$ and the claim on the diameter. □

**Corollary 3.43.**
*The graph $\mathcal{TB}^{\mathrm{NNI}}_{n,r}$ is connected with diameter* $\mathrm{diam}(\mathcal{TB}^{\mathrm{NNI}}_{n,r}) \in \mathcal{O}(nr + r^2 + n \log n)$.

*Proof.* Consider two networks $N, N' \in \mathcal{TB}_{n,r}$ with base tree $T$ and $T'$, respectively. By Theorem 3.42 we can transform $N$ and $N'$ into networks $M$ and $M'$ that have base tree $T$ and $T'$, respectively, and $r$ pairs of parallel edges chained up below the root. This takes at most $\mathcal{O}(nr + r^2)$ $\mathrm{NNI}^0$. Then by Theorem 3.1 $M$ can be transformed into $M'$ in $\mathcal{O}(n \log n)$. Therefore, $N$ and $N'$ are connected with $\mathrm{NNI}^0$-distance in $\mathcal{O}(nr + r^2 + n \log n)$ in $\mathcal{TB}_{n,r}$. □

Note that for $n \geq 2$, $\mathcal{TB}_{n,r}(T)^{\mathrm{NNI}}$ also contains networks with triangles where two edges belong to the embedding of $T$. Hence, the tiers of $\mathcal{TB}_n(T)$ are connected by vertical NNI. This gives us the following corollary.

**Corollary 3.44.**
*Let $n \geq 2$. The graphs $\mathcal{TB}_n^{\mathrm{NNI}}$ and $\mathcal{TB}_n^{\mathrm{NNI}}(T)$ are connected with unbounded diameter.*

## 3.9 Level-$k$ networks

The last classes we consider are level-$k$ networks $\mathcal{LV}_{k,n}$ and strict level-$k$ networks $s\mathcal{LV}_{k,n}$. Recall that a blob $B$ of a network is a nontrivial biconnected component and that the level of $B$ is the number of reticulations of $B$. In a (strict) level-$k$ network every blob is at most (resp. exactly) level $k$. Before we start with strict level-$k$ networks, we define a special type of blob. A *$k$-burl* is recursively defined: a 1-burl is the blob consisting of a pair of parallel edges; a $k$-burl is the blob obtained by placing a pair of parallel edges on one of the parallel edges of a $k-1$-burl for all $k > 1$ [JK19].

**Theorem 3.45.**
*Let $k \geq 1$. Let $op \in \{\mathrm{SNPR}, \mathrm{PR}\}$.*
*The graph $s\mathcal{LV}_{k,n}^{op}$ is connected with unbounded diameter.*

*Proof.* Note that $s\mathcal{LV}_{k,n}$ does not contains $\mathcal{T}_n$ since every network in $s\mathcal{LV}_{k,n}$ must contain at least one blob with level $k$. Let $N \in s\mathcal{LV}_{k,n}$. Let $T$ be a fixed tree in $\mathcal{T}_n$. To show connectedness, our target network $M \in s\mathcal{LV}_{k,n}$ is obtained from $T$ by adding a $k$-burl to the root edge of $T$. We construct an SNPR-sequence (or PR-sequence) from $N$ to $M$ as follows. First, with $k$ $\mathrm{SNPR}^+$ we add add a $k$-burl to the root edge of $N$. Let $N'$ be the resulting network. Second, using only $\mathrm{SNPR}^-$ we remove every other blob of $N'$, which is possible by Lemma 7.4 by Bordewich et al. [BLS17] (see also Corollary 5.2). Let $M'$ be the resulting network. Third and last, we transform $M'$ into $M$ with $\mathrm{SNPR}^0$ by transforming the pendant tree below the $k$-burl into $T$, which is possible by Theorem 3.2. Clearly, every intermediate network from $N$ to $M$ is a level-$k$ network. This proves the connectedness of $s\mathcal{LV}_{k,n}^{op}$. That the diameter is unbounded follows from Lemma 3.4. □

For the tiers of strict level-$k$ networks, we restrict our attention to strict level-1 networks.

**Theorem 3.46.**
*Let $op \in \{\mathrm{SNPR}, \mathrm{PR}\}$. The graph $\mathcal{SLV}_{1,n}^{op}$ is connected with $\mathrm{diam}(\mathcal{SLV}_{1,n}^{op}) \in \mathcal{O}(n+r)$.*

*Proof.* Let $N \in \mathcal{SLV}_{1,n}$. Let our target network $M$ be obtained from a caterpillar with a fixed leaf order by adding $r$ reticulations in series below the root. Let $v$ be a reticulation in $N$. Note that since $N$ is a level-1 network, $v$ has no reticulation as parent. Therefore, we can bring $v$ in parallel with a $\mathrm{SNPR}^0$. Since, this does not affect any blob except for the one containing $v$, this clearly results in a strict level-1 network. To bring all reticulations of $N$ in parallel requires at most $r$ $\mathrm{SNPR}^0$. Let $N'$ be the resulting network. Next, suppose there are $r' < r$ reticulations in series below the root in $N'$. Of those reticulations, let $v'$ be the lowest of them or the root if $r' = 0$. Let $u$ be the parent of a reticulation $v'$ of $N'$ that is neither the child of another reticulation nor of $\rho$. Thus $v$ is not one of the $r'$ reticulations in series below the root. Then prune the incoming edge of $u$ with an $\mathrm{SNPR}^0$ and regraft it to the outgoing edge of $v'$. Let $x$ be the new parent of $u$ and let $y$ be the sibling of $u$. Prune $(x, y)$ at $x$ and regraft it to the outgoing edge of $v$. This increases $r'$ by one. Repeat this process until all reticulations are in series below the root. This takes at most $2r$ $\mathrm{SNPR}^0$. Lastly, to obtain $M$ we transforms the pendant tree below the

reticulations into a caterpillar with the specific leaf order in $\mathcal{O}(n)$ SNPR$^0$. This proves the connectedness of $\mathcal{SLV}^{op}_{1,n}$ and the upper bound on the diameter. □

It remains open whether the tiers of strict level-$k$ networks for $k > 1$ are connected. However, for (non-strict) level-$k$ we can prove the following.

**Theorem 3.47.**
*Let op* $\in \{\text{SNPR}, \text{PR}\}$. *The graph* $\mathcal{LV}^{op}_{k,n,r}$ *is connected with* $\text{diam}(\mathcal{LV}^{op}_{k,n,r}) \in \mathcal{O}(n+r)$.

*Proof.* Let $N \in \mathcal{LV}_{k,n,r}$ and let $M$ be the same target network as in Theorem 3.47, namely, a caterpillar with additionally $r$ reticulations in series below the root. First, to make all reticulations of $N$ in parallel, apply the two steps from the proof of Lemma 3.9. Recall that this works the same for reticulations with a tree vertex as parent as in Theorem 3.47, but also describes how to handle reticulations with two reticulations as parent. Note that none of the $\mathcal{O}(r)$ SNPR$^0$ used for this can increase the level. For the resulting network $N'$, which contains $r$ reticulations in parallel, we do the same as in Theorem 3.47 to transform it into $M$. This proves the connectedness and the bound on the diameter of $\mathcal{LV}^{op}_{k,n,r}$. □

Lastly, note that $\mathcal{LV}_{k,n}$ contains $\mathcal{T}_n$ and that applying an SNPR$^-$ does not increase the level. Hence, from Lemma 3.3 and Lemma 3.4 we get the following corollary.

**Corollary 3.48.**
*Let op* $\in \{\text{SNPR}, \text{PR}\}$. *The graph* $\mathcal{LV}^{op}_{k,n}$ *is connected with unbounded diameter.*

## 3.10 Concluding remarks

In this chapter we have studied the connectedness of classes of phylogenetic networks under rearrangement operations. This is the defining property of a metric space of phylogenetic networks and thus builds the foundation for the problems we consider in the next two chapters. Alongside connectedness, we also looked at asymptotic bounds for the diameters of these spaces.

Building on the work of Bordewich et al. [BLS17], who showed that the classes of all networks, tree-child networks, reticulation-visible networks, and tree-based networks are connected under SNPR, we established connectedness for the classes of normal networks, temporal normal networks, tree-sibling networks, and level-$k$ networks. These results also imply that PR induces metrics on these spaces. For the classes with bounded maximum number of reticulations, we found that the diameters behave asymptotically the same under SNPR and PR. Gambette et al. [GvIJ$^+$17] showed that $\mathcal{N}_n$ is connected under NNI. We established connectedness under NNI for $\mathcal{TC}_n$ and $\mathcal{TB}_n$. Furthermore, we observed that $\mathcal{NN}_n$ is not connected under NNI, since there are no vertical NNI possible on a normal network.

We also considered the spaces of tiers of networks of certain classes and rearrangement operations. For tree-child, normal, and temporal normal networks we saw that the tiers are connected under SNPR except for the tier with the maximum number of reticulations. In contrast to that, PR also induces metric spaces with these extremal tiers. However, for reticulation-visible networks the highest tier is not connected under PR and thus also not under NNI and SNPR. In addition, we showed that tiers of tree-sibling and tree-based networks are connected. We found lower asymptotic upper bounds of the diameter of tree-based networks for PR than for SNPR. Lastly, we have proven that $\mathcal{TC}_{n,r}$ and $\mathcal{TB}_{n,r}$ are connected under NNI. The precise results concerning the tiers are summarised in Table 3.1.

Open problems remain for connectedness of classes under NNI. Furthermore, some of the asymptotic bounds on diameters are not tight or only upper bounds are known. Besides that, there are of course also more classes of phylogenetic networks, like orchard networks [JM18, ESS19], that could be considered.

Table 3.1: Overview of results on connectedness and diameter of tier $r$ of classes of phylogenetic networks under NNI, SNPR, and PR. The symbol $\times$ means that the respective space is disconnected. Here $n \geq 3$ and $m = n + r$.

| class | NNI | SNPR | PR |
|---|---|---|---|
| $\mathcal{T}_n$ | $\Theta(n \log n)$ [LTZ96] | $\Theta(n)$ [Son03] (as SPR) | |
| $\mathcal{N}_{n,r}$ | $\Omega(m \log m), \mathcal{O}(m^2)$ [JJE+18] | $\Theta(n + r)$ [JJE+18], L. 3.9 | |
| $\mathcal{TC}_{n,r}$ | $\mathcal{O}(n^2)$ T. 3.18 | $r < n - 1 : \Theta(n)$ [BLS17] | $\Theta(n)$ T. 3.17 |
| | | $r = n - 1 : \times$ T. 3.16 | |
| $\mathcal{NN}_{n,r}$ | | $r < n - 2 : \Theta(n)$ T. 3.21 | $\Theta(n)$ T. 3.23 |
| | | $r = n - 2 : \times$ T. 3.22 | |
| $\mathcal{TP}_{n,r}$ | | $r < n - 2 : \Theta(n)$ T. 3.27 | $\Theta(n)$ T. 3.28 |
| | | $r = n - 2 : \times$ T. 3.26 | |
| $\mathcal{TS}_{n,r}$ | | $\Theta(n + r)$ T. 3.31 | |
| ${}^*\mathcal{RV}_{n,r}$ | | $\mathcal{O}(nr)$ [BLS17] | |
| $\mathcal{RV}_{n,r}$ | | $r \leq n - 2$: $\mathcal{O}(n^2)$ T. 3.36 | |
| | $r = 3n - 3$: $\times$ L. 3.35 | | |
| $\mathcal{TB}_{n,r}(T)$ | $\mathcal{O}(nr + r^2)$ T. 3.42 | $\mathcal{O}(nr)$ [BLS17] | $\leq 2r$ L. 3.40 |
| $\mathcal{TB}_{n,r}$ | $\mathcal{O}(nr + r^2 + n \log n)$ C. 3.43 | $\mathcal{O}(nr)$ [BLS17] | $\Theta(n + r)$ L. 3.41 |
| $\mathcal{LV}_{k,n,r}$ | | $\mathcal{O}(n + r)$ T. 3.47 | |

To close this section, let us consider again the proof that tier $r = 3n - 3$ of $\mathcal{RV}_{n,r}$ is not connected under PR. There we looked at what operations are possible on a particular network. In particular, we examined which edges can be pruned and where they may be regrafted to obtain a reticulation-visible network. In other words, we looked at what networks are adjacent to the network at hand. This is the subject of the next chapter.

# 4. Neighbourhood size

A central problem of phylogenetics is finding an optimal phylogenetic tree or network to fit a given data set. Since the space of possible solutions is huge (recall Theorem 2.1), most algorithms use a local search strategy where the next network in the search is chosen from the neighbours of the current network (as noted by St. John [SJ17]). In view of these algorithmic applications it is of interest to study the neighbourhood of phylogenetic networks. In particular, the *neighbourhood problem* is the problem of determining the neighbourhood size of a network within a particular space. For example, the tree-child network in Figure 4.1 has a neighbourhood of size fifteen in the space of tree-child networks under SNPR. In general, we want a solution to the neighbourhood problem given in the form of a closed formula that is based on the network and the space considered. Such a formula often depends not only on the size of the network (given by the number of leaves $n$ and the number of reticulations $r$), but also on the topology of the network. In such a case, we also care about extreme values.

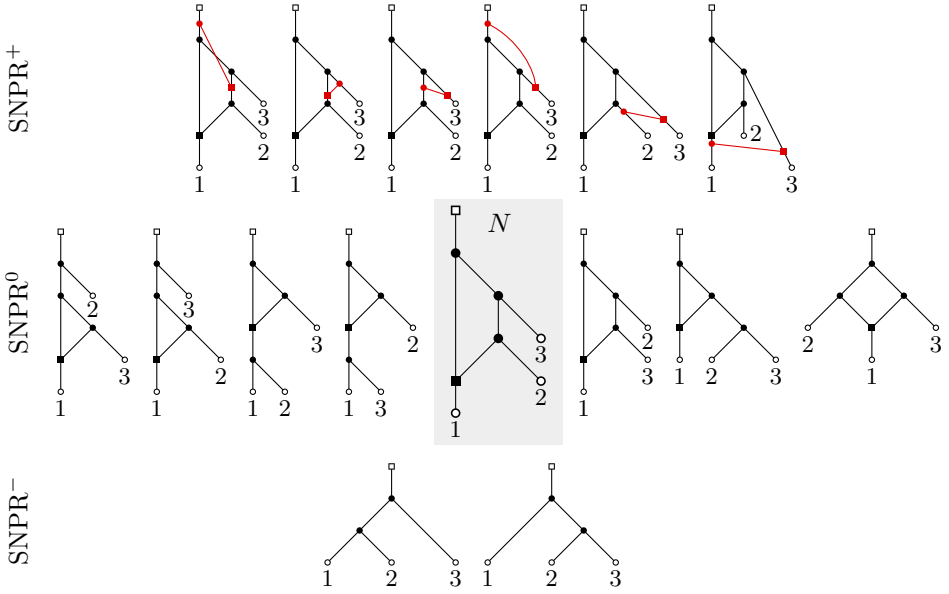


Figure 4.1: A tree-child network $N$ in the middle, with its SNPR tree-child neighbourhood around it. The top row shows the SNPR$^+$ neighbours, the middle row the SNPR$^0$ neighbours, and the bottom row the SNPR$^-$ neighbours.

When Robinson [Rob71] introduced the NNI operation on unrooted phylogenetic trees, he also showed that the size of the neighbourhood is $2n-6$. Furthermore, he also considered the NNI $k$-neighbourhood of a tree $T$; that is, all trees that have NNI-distance $k$ to $T$. In particular for $k = 2$ and $k = 3$, Robinson showed that the $k$-neighbourhood size of $T$ depends on the topology of $T$ and gave lower and upper bounds. Thirty years later, Allen and Steel [AS01] showed that the SPR neighbourhood of an unrooted tree has the simple formula $2(n-3)(n-7)$ but that the size of the TBR neighbourhood depends on the topology of the tree. Later Humphries and Wu [HW13] gave the exact formula $4\sum_{A|B\in S}|A||B| - (4n-2)(n-3)$ for the size of the TBR neighbourhood, where $S$ is the set of all nontrivial splits[1] of the tree. In addition, they proved that this is maximised by caterpillars and minimised by balanced trees. Also on unrooted phylogenetic trees, de Jong et al. [dJMS16] considered the problem of finding neighbours that are two or more operations away under NNI, SPR, and the so-called Robinson-Foulds metric. Baskowski et al. [BMSW15] solved the neighbourhood problem for NNI, SPR, and TBR on unrooted phylogenetic trees that are restricted to a circular ordering of its leaves. A question related to the neighbourhood problem was considered by Caceres et al. [CCLSJ13], who showed that a shortest NNI-walk through the SPR neighbourhood of a tree takes $\Theta(n^2)$ more steps than the number of trees in that neighbourhood.

Concerning rooted phylogenetic trees, Song [Son03] solved the SPR neighbourhood problem by first constructing a recursive formula from which he then derived a closed formula. Moreover, he gave sharp upper and lower bounds for the neighbourhood size. We look at these results more closely below. Song [Son06] used the same approach for ranked phylogenetic trees[2] to find a formula for the neighbourhood size under SPR, gave a sharp upper bound and conjectured a sharp lower bound. Related to this, Gavryushkin et al. [GWMI18] looked at NNI neighbourhoods of discrete time-trees[3].

In this chapter we study the neighbourhood problem for spaces under NNI and SNPR. We give exact formulas for classes of relatively low complexity. In particular, we look at the classes of trees, tree-child networks, and normal networks. An important property of a network in either of these classes is that each vertex and each edge is uniquely identifiable. Because of this property the complexity of handling isomorphism between neighbours arising from different operations is manageable. However, as we will see, the formulas for neighbourhood sizes are still rather long and thus hard to comprehend. Therefore we also give bounds for the extreme values of these formulas. Tree-based networks or general phylogenetic networks do not have this property. We therefore refrain from finding formulas for these network classes. Nevertheless, we still look at bounds of the neighbourhood size for a phylogenetic network under NNI, SNPR, and PR.

We first look at the class of phylogenetic trees (Section 4.2). We use the simple case of trees to illustrate our counting scheme to find formulas. Roughly speaking, in this scheme we first count the number of possible operations and then factor in double counting of neighbours and the operations that yield the starting network again. We then study the class of tree-child networks in Section 4.3. This section is the main work of this chapter and we use several results thereof for the neighbourhood problem on the other classes. In particular, we use the classification of redundant operations for the class of normal networks (Section 4.4). We then discuss why it is harder to find exact formulas for the

[1] A split $A|B$ of an unrooted phylogenetic tree with leave set $\mathcal{X}$ is a biparition $(A, B)$ of $\mathcal{X}$ such that there is a cut-edge of $T$ separating $A$ and $B$.

[2] Ranked or totally-ordered phylogenetic trees are phylogenetic trees together with a total order of the inner tree vertices.

[3] Discrete time trees are phylogenetic trees where all vertices are vertices are assigned positive real numbers and where the different time periods between two vertices may take only a finite number of values.

neighbourhood problem for other classes of phylogenetic networks. Lastly, we give bounds on the neighbourhood size of a phylogenetic network in $\mathcal{N}_n$ (Section 4.5). Before we start with the neighbourhood of a tree, we make several definitions.

*Remark.* Section 4.3 on the neighbourhood of tree-child networks and other parts of this chapter appeared in the paper "The SNPR neighbourhood of tree-child networks" [Kla18].

## 4.1 Preliminaries

This section contains definitions, notation, and observations used throughout this chapter. First, we introduce a notation for SNPR and NNI operations, so that we can distinguish two operations on the same network. We also look at different properties of operations. We then define the neighbourhood problem formally. Next, we look at an important property of tree-child networks that keeps the complexity of the neighbourhood problem for this class within reasonable bounds. We further count how many edges of a certain type a tree-child network contains, define functions to count descendant edges, and define special structures of networks.

**Operation types.** Let $N \in \mathcal{N}_n$. For an $\text{SNPR}^0$ that prunes the edge $e$ and regrafts it to the edge $f$, we write $(e, f)$. For an $\text{SNPR}^+$ that adds a new edge from the edge $f$ to the edge $e$, we write $(e, f)$. For an $\text{SNPR}^-$ that removes the edge $e$, we simply write $e$. Let $\Theta^{\text{SNPR}}(N)$ denote the multiset of all SNPR operations on $N$. If $N$ is in the class $\mathcal{C}_n$, let $\Theta^{\text{SNPR}}_{\mathcal{C}}(N)$ denote the subset of $\Theta^{\text{SNPR}}(N)$ of operations that result in a network in $\mathcal{C}_n$. If, say, $\mathcal{C}_n = \mathcal{TC}_n$, then we call an operation $\theta \in \Theta^{\text{SNPR}}_{\mathcal{C}}(N)$ a *tree-child respecting* operation. The definitions for other types of operations and classes are analogous.

For an $\text{NNI}^0$ operation on the edges $e$, $f$, and $g$ with axis $e = (u, v)$ and where $f$ is incident to $u$ and $g$ is incident to $v$, we write $(f, e, g)$. We do not need a notation for $\text{NNI}^+$ and $\text{NNI}^-$ operations.

Let $\theta \in \Theta^{\text{SNPR}}(N)$. Let $\theta(N)$ denote the network obtained by applying $\theta$ to $N$. The operation $\theta$ is *trivial* if $\theta(N) = N$. Two distinct operations $\theta, \theta' \in \Theta^{\text{SNPR}}(N)$ on $N$ are *redundant* if $\theta(N) = \theta'(N)$. We call a set of pairwise redundant operations a *redundancy set*. For example, all trivial operations on $N$ form a redundancy set. We call a redundancy set *nontrivial* if it is not the set of trivial operations. As the following observations show, trivial operations are not uncommon, but are also not possible for every type of operation.

**Observation 4.1.**
*Let $N \in \mathcal{N}_n$ for $n \geq 2$. Let $u$ be an inner tree vertex of $N$ with incident edge $e = (u, v)$. Then $e$ is part of at least two trivial* $\text{SNPR}^0$ *operations $(e, f)$ and $(e, f')$.*

*Proof.* Let $p$ be the parent of $u$ and $w \neq v$ the second child of $u$. To find two trivial $\text{SNPR}^0$ that prune $e$, we choose $f = (p, u)$ and $f' = (u, w)$. When $e$ gets pruned and thus $u$ suppressed, then $f$ and $f'$ are merged into an edge $\tilde{f}$. Regrafting $e$ to $\tilde{f}$ yields again $N$. Hence, $(e, f)(N) = (e, f')(N) = N$. □

**Observation 4.2.**
*Vertical rearrangement operations are nontrivial.*

**The neighbourhood problem.** Let $N \in \mathcal{TC}_n$. We define the SNPR *tree-child neighbourhood* of $N$, denoted by $U^{\text{SNPR}}_{\mathcal{TC}}(N)$, as

$$U^{\text{SNPR}}_{\mathcal{TC}}(N) := \{N' \mid N' \in \mathcal{TC}_n, \exists \theta \in \Theta^{\text{SNPR}}(N) \colon \theta(N) = N' \text{ and } N' \neq N\}.^4$$

An SNPR *tree-child neighbour* of $N$ is a network $N' \in U_{\mathcal{TC}}^{\mathrm{SNPR}}(N)$. Definitions for other types of operations and classes are analogous. The *neighbourhood problem* for $N \in \mathcal{C}_n^{op}$ for a rearrangement operation $op$ is the problem of determining $|U_{\mathcal{C}}^{op}(N)|$.

**Number of automorphisms.** Let $N \in \mathcal{N}_n$. Assume there exists an automorphism on $N$ that fixes the leaf set of $N$ and that maps an edge $e$ to an edge $e' \neq e$. This implies that, in some sense, $e$ and $e'$ are indistinguishable. Now consider two $\mathrm{SNPR}^0$ operations $\theta$ and $\theta'$ on $N$ that prune $e$ and $e'$ respectively and regraft them to an edge $f$. Then, in some cases, $\theta(N) = \theta'(N)$ and thus $\theta$ and $\theta'$ are redundant. Such indistinguishable sets of edges make counting neighbours of a network harder, because for each operation $\theta = (e, f)$ one has to consider whether $e$ and $f$ are indistinguishable from other edges and if thus $\theta$ would be redundant to other operations. The next lemma shows that tree-child networks have no such sets of indistinguishable edges. The lemma is a reformulation of a result by McDiarmid et al. [MSW15, Lemma 5.1].

**Lemma 4.3.**
*Let $N \in \mathcal{TC}_n$. There is exactly one automorphism on $N$ that fixes the leaf set of $N$.*

Lemma 4.3 implies that every vertex and every edge of a tree-child network is uniquely identifiable, for example by its set of descendant edges. Since by Table 2.1 phylogenetic trees and normal networks are also tree-child networks, we get the following corollary.

**Corollary 4.4.**
*Let $N$ be a tree or a normal network.*
*There is exactly one automorphism on $N$ that fixes the leaf set of $N$.*

**Structures.** Let $N \in \mathcal{TC}_n$. In the following we define certain subgraphs of $N$, which we call structures, that are determining factors of whether and which operations on $N$ are trivial, redundant, and respect a class. Figure 4.2 accompanies the description of these structures.

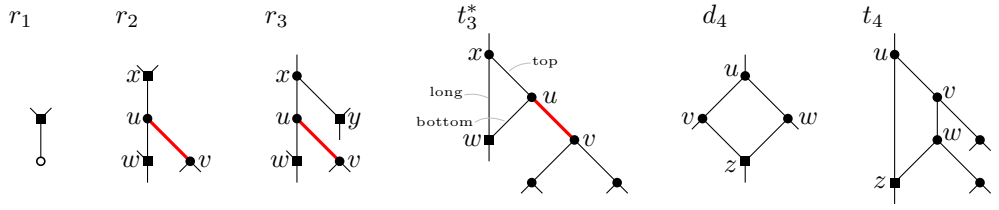


Figure 4.2: An $r_1$, $r_2$, and $r_3$ structure, a tree-branching triangle $t_3^*$, a diamond $d_4$, a trapezoid with outgoing tree edges $t_4$. The critical edges are highlighted (bold red).

We define $r_1$ as the number of reticulations in $N$ that have a leaf as child. An $r_2$ structure of $N$ is a path of length two from a reticulation $x$ via a vertex $u$ to a reticulation $w$. An $r_3$ structure consists of four vertices $x, y, u, w$ with edges $(x, y)$, $(x, u)$ and $(u, w)$ where $y$ and $w$ are reticulations. We refer to these three edges as the underlying path of the structure. Note that in both an $r_2$ and an $r_3$ structure, since $N$ is tree child, both $u$ and its second child $v$ are tree vertices. We relax the notation to let $r_2$ and $r_3$ also denote the number of these structures in $N$.

[4]The neighbourhood is denoted by $U$, since it is sometimes referred to as the *unit* neighbourhood compared to the $k$-neighbourhood that contains all networks that have distance at most $k$ to $N$.

Note that an $r_3$ structure with $y = w$ is a triangle. More formally, a *triangle* of $N$ consist of three vertices $x, u, w$ with the edges $(x, u)$, $(x, w)$ and $(u, w)$. We call the edge $(x, u)$ the *top side*, the edge $(x, w)$ the *long side*, and the edge $(u, w)$ the *bottom side* of the triangle. Let $t_3$ denote the number of triangles in $N$. Let $v \neq w$ be the second child of $u$. If $v$ is incident to three pure tree edges, we call it a *tree-branching triangle*. Let $t_3^*$ denote the number of tree-branching triangles. Note that every tree-branching triangle of $N$ is included in the counts $r_3$, $t_3$, and $t_3^*$.

For an $r_2$ structure, an $r_3$ structure, or a triangle, with the notation from above, we call the tree edge $(u, v)$ the *critical* edge of this structure. See again Figure 4.2, where critical edges are highlighted, and note how pruning them yields a vertex without a tree child. These edge will be important when we consider tree-child respecting SNPR operations.

A *diamond* of $N$ is an underlying four-cycle consisting of the edges $(u, v)$, $(u, w)$, $(v, z)$ and $(w, z)$. A *trapezoid* is an underlying four-cycle consisting of the edges $(u, v)$, $(v, w)$, $(w, z)$ and $(u, z)$. In both underlying four-cycles $z$ is a reticulation. Important for us are trapezoids where the outgoing edges of the four-cycle at $v$ and $w$ are pure tree edges. Let $d_4$ denote the number of diamonds and let $t_4$ denote the number of trapezoids with two outgoing pure tree edges.

**Number of edges.** Let $N = (V, E) \in \mathcal{TC}_{n,r}$. Let $m$ denote the size of $E$. Let $E_R \subset E$ denote the set of reticulation edges, let $E_T \subseteq E$ denote the set of tree edges, let $E_{T^*} \subseteq E_T$ denote the set of non-critical tree edges, and let $E_{PS} \subset E$ denote the set of pure tree edges with a sibling pure tree edge. Also recall that we defined phylogenetic networks to have a root with outdegree one.

**Observation 4.5.**
*Let $N = (V, E) \in \mathcal{TC}_n$ with $r$ reticulations. Then*

*(i)* $|E| = m = 2n + 3r - 1$;

*(ii)* $m^2 = 4n^2 - 4n + 9r_2 - 6r + 12nr + 1$;

*(iii)* $|E_R| = 2r$;

*(iv)* $|E_T| = 2n + r - 1$;

*(v)* $|E_{T^*}| = m - 3r - r_2 - r_3 = 2n - r_2 - r_3 - 1$;

*(vi)* $|E_{PS}| = m - 5r - 1 = 2n - 2r - 2$.

Recall that an edge $(x, y)$ is a descendant edge of an edge $(u, v)$ if $x = v$ or if $x$ is a descendant of $v$. Let $N = (V, E) \in \mathcal{N}_n$ and $e \in E$. Let $\alpha, \delta \colon E \to \mathbb{N}$ be functions that map an edge to its number of ancestor or descendant edges, respectively, i.e.

$$\alpha(e) := |\{f : f \in E \text{ is ancestor of } e\}|$$

and

$$\delta(e) := |\{f : f \in E \text{ is descendant of } e\}|.$$

Let $\alpha_T$ and $\delta_T$ be the restrictions of $\alpha$ and $\delta$ that only count ancestor and descendant edges that are tree edges, respectively.

## 4.2 Trees

In this section we give formulas for the size of the NNI and the SNPR neighbourhood of a (rooted) phylogenetic tree. While these neighbourhood problems have been solved before within $\mathcal{T}_n$, we extend these solutions to $\mathcal{N}_n$. We start with the NNI neighbourhood problem.

### 4.2.1 NNI neighbourhood

For unrooted phylogenetic trees, Robinson [Rob71] showed that the $\mathrm{NNI}^0$ neighbourhood size of a tree $T$ is equal to twice the number of inner edges of $T$ and thus $2n-6$. Rooted phylogenetic trees have $n-2$ inner edges and it is well known that like in the unrooted case each of these edges induces two different $\mathrm{NNI}^0$ operations. Since a tree has no reticulations, it has no $\mathrm{NNI}^-$ neighbours. Recall that an $\mathrm{NNI}^+$ adds an edge $(u', v')$ between two incident edges and thus creates a triangle. If the reticulation $v'$ subdivides a tree edge $(u, v)$ then it makes no difference whether $u'$ subdivides the incoming edge or the second outgoing edge of $u$. Either choice will result in the same network. Note that the root edge cannot be subdivided by a reticulation introduced by an $\mathrm{NNI}^+$. Thus a tree has as many $\mathrm{NNI}^+$ neighbours as it has edges minus one, which is $2n-2$ by Observation 4.5. All together, these observations give the following results on NNI neighbourhoods of a tree.

**Theorem 4.6.**
*Let $T \in \mathcal{T}_n$ with $n \geq 2$.*
*The neighbourhoods of $T$ under* NNI *operations have the sizes*

*(i)* $|U_{\mathcal{N}}^{\mathrm{NNI}^0}(T)| = |U_{\mathcal{T}}^{\mathrm{NNI}}(T)| = 2n-4$,

*(ii)* $|U_{\mathcal{N}}^{\mathrm{NNI}^-}(T)| = 0$,

*(iii)* $|U_{\mathcal{N}}^{\mathrm{NNI}^+}(T)| = 2n-2$, *and*

*(iv)* $|U_{\mathcal{N}}^{\mathrm{NNI}}(T)| = 4n-6$.

Note that each neighbourhood in Theorem 4.6 only depends on the size of the tree. Also note that an $\mathrm{NNI}^+$ neighbour of a tree is a level-1 and a tree-child network, but not a normal network. Hence, the NNI neighbourhood of a tree in $\mathcal{N}_n$ is the same as in $\mathcal{TC}_n$ (and other superclasses) but not as in $\mathcal{NN}_n$ and $\mathcal{TP}_n$.

### 4.2.2 SNPR neighbourhood

We use this section to illustrate how the SNPR neighbourhood size depends on the topology of a tree and that this can (partly) be represented by the number of ancestors or descendants of edges. While Song [Son03] used a recursive method to obtain a formula for the $\mathrm{SNPR}^0$ neighbourhood size, we use a direct counting scheme, following Humphries and Wu [HW13]. This scheme consists of three steps. First, we determine the number of possible operations, which in our case is the size of $\Theta_{\mathcal{T}}^{\mathrm{SNPR}}(T)$. Second, we subtract from this the number of trivial operations, and third, correct for double counting of neighbours due to redundancies. For the rest of this section let $T \in \mathcal{T}_n$.

**Counting SNPR operations.** Our first step is to count the number of SNPR on $T$. We distinguish between $\mathrm{SNPR}^0$ and $\mathrm{SNPR}^+$, and disregard $\mathrm{SNPR}^-$ since there are none for $T$. Recall that an $\mathrm{SNPR}^0$ $(e, f) \in \Theta^{\mathrm{SNPR}^0}(T)$ prunes the edge $e$ at its tail vertex and regrafts it to the edge $f$. By the definition of $\mathrm{SNPR}^0$, the edge $f$ cannot be $e$ or a descendant of

$e$. The operation $(e, f)$ would otherwise induce a cycle, as illustrated by the trees $T$ and $T_\delta$ in Figure 4.3. Recall that an SNPR$^+$ $(e, f) \in \Theta^{\text{SNPR}^+}(T)$ adds an edge from $f$ to $e$. Furthermore, recall that $\delta(e)$ is the number of descendant edges of $e$.

**Lemma 4.7.**
*Let $n \geq 2$. Let $T = (V, E) \in \mathcal{T}_n$. The number of* SNPR$^0$ *on $T$ is*

$$|\Theta^{\text{SNPR}^0}(T)| = 4n^2 - 6n + 2 - \sum_{e \in E} \delta(e), \tag{4.1}$$

*and the number of* SNPR$^+$ *on $T$ is*

$$|\Theta^{\text{SNPR}^+}(T)| = 4n^2 - 4n + 1 - \sum_{e \in E} \delta(e). \tag{4.2}$$

*Proof.* Let $m = |E|$. First we count SNPR$^0$. Let $\Theta^{\text{SNPR}^0}(T, e)$ denote all SNPR$^0$ on $T$ that prune $e$. By our observation that the edge $f$ of an SNPR$^0$ operation $(e, f)$ can be any edge that is not a descendant of $e$ or $e$ itself we get the equation

$$|\Theta^{\text{SNPR}^0}(T, e)| = m - 1 - \delta(e).$$

The total number of SNPR$^0$ operations on $T$ is then

$$|\Theta^{\text{SNPR}^0}(T)| = |\bigcup_{e \in E} \Theta^{\text{SNPR}^0}(T, e)| = m^2 - m - \sum_{e \in E} \delta(e) = 4n^2 - 6n + 2 - \sum_{e \in E} \delta_e.$$

The last step follows from Observation 4.5. This proves Equation (4.1).

Next we count SNPR$^+$. This case is similar to the previous one with the only difference that for an SNPR$^+$ $(e, f)$ it is possible that $f = e$. By Observation 4.5 there are thus $m = 2n - 1$ more SNPR$^+$ than SNPR$^0$. This proves Equation (4.2). □

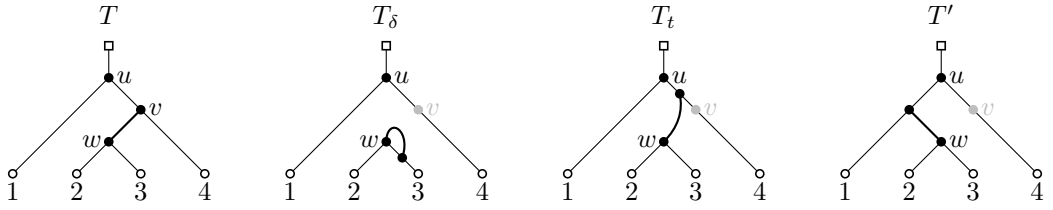


Figure 4.3: Illustration of different possibilities for regrafting when pruning the edge $(v, w)$ in the phylogenetic tree $T$. The tree $T_\delta$ shows that we cannot regraft $(., w)$ to a descendant edge; the tree $T_t$ shows that regrafting to the edge $(u, v)$ (or $(v, 4)$) yields a trivial SNPR; the tree $T'$ is a neighbour of $T$.

**Counting trivial operations.** Our second step is to count all trivial SNPR on $T$. Recall that an operation $\theta$ is trivial if $\theta(T) = T$. By Observation 4.2 there are no trivial SNPR$^+$. Let $(e, f) \in \Theta^{\text{SNPR}^0}(T)$. By Observation 4.1 we know that for each $e$ there are at least two choices of $f$ such that $(e, f)$ is trivial. For $e = (u, v)$ this is the case when $f \neq e$ is incident to $u$. See again Figure 4.3. We prove in Lemma 4.14 that there are no further trivial operations on $T$. Hence, for each $e \in E(T)$ that is not the root edge of $T$ there are two trivial SNPR$^0$ and thus the total number of trivial operations in $\Theta^{\text{SNPR}^0}(T)$ is

$$2(m - 1) = 4n - 4. \tag{4.3}$$

**Counting redundancies.** Our third step is to count nontrivial redundancies of SNPR operations on $T$. Recall that two operations are redundant if they yield the same tree. Humphries and Wu [HW13] showed that for unrooted trees every nontrivial redundancy set has size four and corresponds to an $\mathrm{NNI}^0$. In Lemma 4.8 we state that a nontrivial redundancy set has only size three in the corresponding statement for rooted trees. Figure 4.4 illustrates where this change from size four to three comes from.

**Lemma 4.8.**
*Let $T \in \mathcal{T}_n$ and let $\theta, \theta' \in \Theta^{\mathrm{SNPR}^0}(T)$ be distinct, nontrivial, and redundant with $\theta(T) = T'$. Then there exists an $\mathrm{NNI}^0$ operation $\sigma \in \Theta^{\mathrm{NNI}^0}(T)$ such that $\sigma(T) = T'$. Furthermore, every nontrivial redundancy set of $\Theta^{\mathrm{SNPR}^0}(T)$ has size three.*

We prove Lemma 4.8 in Section 4.3.1.

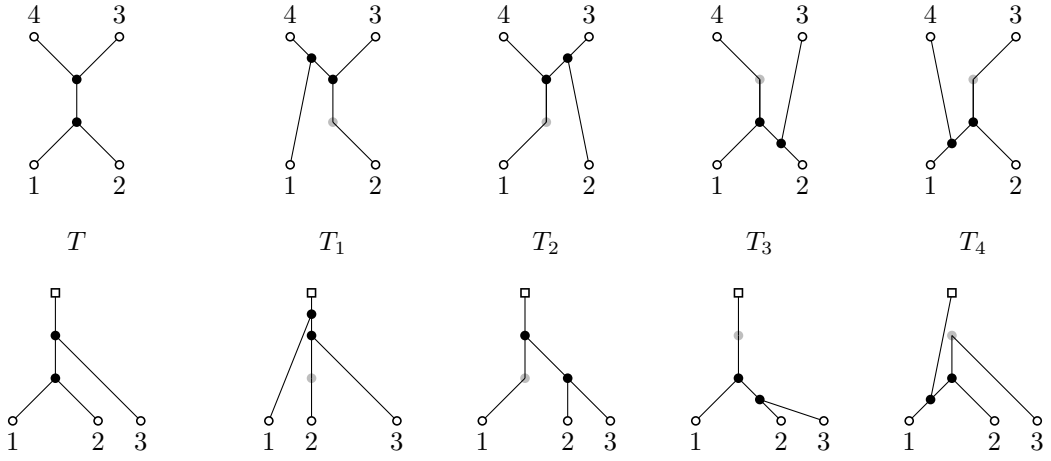


Figure 4.4: The top row illustrates how four different, but redundant SPR operations correspond to an NNI operation on an unrooted tree. The bottom row illustrates how one of these SPR operations has no rooted $\mathrm{SNPR}^0$ as counterpart.

**Lemma 4.9.**
*Let $n \geq 2$. Let $T \in \mathcal{T}_n$.*
*There are $2n-4$ nontrivial redundancy sets of $\mathrm{SNPR}^0$ operations in $\Theta^{\mathrm{SNPR}^0}(T)$, each with size three, and $2n-2$ nontrivial redundancy sets of $\mathrm{SNPR}^+$ operations in $\Theta^{\mathrm{SNPR}^+}(T)$, each with size two.*

*Proof.* The first part follows from Lemma 4.8 and Theorem 4.6. The second part follows from the proof of Proposition 4.19, which shows that redundancies of $\mathrm{SNPR}^+$ (for a tree) only arise from $\mathrm{NNI}^+$, and Theorem 4.6. □

**SNPR neighbourhood size.** The neighbourhood size of $T$ can now be determined from the number of SNPR on $T$ as counted in Lemma 4.7 by subtracting the trivial SNPR as counted in Equation (4.3), and by picking only one operation of every redundancy set of nontrivial SNPR as counted in Lemma 4.9.

**Theorem 4.10.**
*Let $T \in \mathcal{T}_n$ with $n \geq 2$.*
*The* SNPR *neighbourhoods of $T$ have the sizes*

*(i)* $|U_{\mathcal{N}}^{\mathrm{SNPR}^0}(T)| = |U_{\mathcal{T}}^{\mathrm{SNPR}}(T)| = 4n^2 - 14n + 14 - \sum_{e \in E} \delta(e)$,

*(ii)* $|U_{\mathcal{N}}^{\mathrm{SNPR}^-}(T)| = 0$,

*(iii)* $|U_{\mathcal{N}}^{\mathrm{SNPR}^+}(T)| = 4n^2 - 6n + 3 - \sum_{e \in E} \delta(e)$, *and*

*(iv)* $|U_{\mathcal{N}}^{\mathrm{SNPR}}(T)| = 8n^2 - 20n + 17 - 2\sum_{e \in E} \delta(e)$.

Note that not every $\mathrm{SNPR}^+$ neighbour of $T$ is a tree-child network. Therefore, the formulas of Theorem 4.10 do not describe the same neighbourhoods as those in Proposition 4.19 and Theorem 4.20 when applied to a tree.

Also note that for a tree $T$, we have the equation

$$\sum_{e \in E} \delta(e) = \sum_{e \in E} \alpha(e). \tag{4.4}$$

Equation (4.4) follows from the observation that if an edge gets counted $k$ times as a descendant, then it has $k$ ancestors. Having Equation (4.4) means that we can either count the descendants or the ancestors of the edges of $T$ for the formulas in Theorem 4.10.

**Equivalence to Song's formula.** Song [Son03] already gave a closed formula for $|U_{\mathcal{T}}^{\mathrm{SNPR}}(T)| = |U_{\mathcal{N}}^{\mathrm{SNPR}^0}(T)|$ derived from a recursive formula based on a so-called cherry sequence. To state his formula we need the following definitions. For a vertex $v$ let $\alpha(v)$ be the number of ancestor vertices of $v$. Let $V_I$ be the inner vertices of $T$ and let $v \in V_I$. Then Song defined the function $\gamma(v) = \max\{\alpha(v) - 2, 0\}$. Song's formula for the SNPR neighbourhood of a rooted phylogenetic tree is

$$|U_{\mathcal{T}}^{\mathrm{SNPR}}(T)| = 4n^2 - 18n + 20 - 2\sum_{v \in V_I} \gamma(v).$$

We show how to transform this formula with $\gamma$ on $V_I$ to the formula in Theorem 4.10 with $\delta$ on $E$. First, note that every inner vertex $v$ has two outgoing edges $e$ and the only edge that is not covered in this way is the root edge $e_\rho$. Let $v \in V_I$ with outgoing edges $e$ and $e'$. Then $2\gamma(v) = \alpha(e) + \alpha(e') - c$. If $v$ is the child of the root then $c = 2$, otherwise $c = 4$. Furthermore, note that $\alpha(e_\rho) = 0$. In total we get that

$$2\sum_{v \in V_I} \gamma(v) = \sum_{e \in E} \alpha(e) - 2m + 4 = \sum_{e \in E} \alpha(e) - 4n + 6.$$

Lastly, we use Equation (4.4) to get the equivalence

$$4n^2 - 18n + 20 - 2\sum_{v \in V_I} \gamma(v) = 4n^2 - 14n + 14 - \sum_{e \in E} \alpha(e) = 4n^2 - 14n + 14 - \sum_{e \in E} \delta(e).$$

**Bounds.** Since the size of the SNPR neighbourhood of a phylogenetic tree depends not only on its size but also on its topology, we now look at lower and upper bounds. Song [Son03, Proposition 4.1] showed that the minimum and maximum $\mathrm{SNPR}^0$ neighbourhood size are achieved by caterpillars and balanced trees, respectively. Song's result also implies that the caterpillar minimises and that a balanced tree maximises $\sum_{e \in E} \delta(e)$. The minimum and maximum $\mathrm{SNPR}^+$ (and SNPR) neighbourhood size of $T$ are thus also achieved by caterpillars and balanced trees, respectively. Moreover, Song [Son03, Corollary 4.2] gave the exact lower bound with $3n^2 - 13n + 14$ and the exact upper bound with $4n^2 - 16n + 16 - 2\sum_{i=1}^{n-2} \lfloor \log_2(i+1) \rfloor$ for the $\mathrm{SNPR}^0$ neighbourhood size of $T$. Thus, for a caterpillar we have $\sum_{e \in E} \delta(e) = n^2 - n$ and for a balanced tree we have $\sum_{e \in E} \delta(e) = 2\sum_{i=1}^{n-2} \lfloor \log_2(i+1) \rfloor + 2n - 2$. Applying these observations to the formulas of Theorem 4.10 gives the following corollary.

**Corollary 4.11.**
*Let $T \in \mathcal{T}_n$ with $n \geq 4$. Then*

$$n^2 - 13n + 14 \leq |U_{\mathcal{N}}^{\mathrm{SNPR}^0}(T)| \leq 4n^2 - 16n + 16 - 2\sum_{i=1}^{n-2} \lfloor \log_2(i+1) \rfloor$$

$$3n^2 - 5n + 3 \leq |U_{\mathcal{N}}^{\mathrm{SNPR}^+}(T)| \leq 4n^2 - 8n + 5 - 2\sum_{i=1}^{n-2} \lfloor \log_2(i+1) \rfloor$$

$$6n^2 - 18n + 17 \leq |U_{\mathcal{N}}^{\mathrm{SNPR}}(T)| \leq 8n^2 - 24n + 21 - 4\sum_{i=1}^{n-2} \lfloor \log_2(i+1) \rfloor$$

*with the lower bound achieved by caterpillars and the upper bound achieved by balanced trees.*

## 4.3 Tree-child networks

In the last section we saw that applying any NNI or SNPR to a tree yields a tree in $\mathcal{T}_n$ or a network in $\mathcal{N}_n$. This is different for the space of tree-child networks, where we have to check whether an operation is tree-child respecting. Furthermore, recognising trivial and redundant operations becomes more complex as there are more structures to consider. Fortunately, the fact that vertices and edges are uniquely identifiable in tree-child networks keeps the complexity of this task within manageable bounds. As we will see, the formulas for neighbourhood sizes are nevertheless lengthy. We therefore also give bounds for these formulas that depend only on the size of the network.

The study of SNPR on tree-child networks is the main part of this chapter. We already used results thereof for trees and will use them again for normal network. Most proofs in the section are rather technical. The proof about the redundancies of tree-child respecting SNPR, however, uses a nice way to reduce the problem of identifying redundancies from a global problem (on the network) down to a local problem.

We start with the neighbourhood problem under SNPR. Throughout this section let $N \in \mathcal{TC}_n$.

### 4.3.1 SNPR neighbourhood

We first consider $\mathrm{SNPR}^0$, then $\mathrm{SNPR}^+$ and $\mathrm{SNPR}^-$. Towards formulas for neighbourhood sizes, we use the counting scheme we illustrated on trees. Thus we first count tree-child respecting operations, then trivial operations, and then redundancies.

**Counting tree-child respecting $\mathrm{SNPR}^0$ operations.** Let $\theta = (e, f) \in \Theta_{\mathcal{TC}}^{\mathrm{SNPR}^0}(N)$ and $e = (u, v)$. For $\theta$ to be tree-child respecting, neither pruning $(u, v)$ nor regrafting it to $f$ can yield a non-leaf vertex without a tree child. Roughly speaking and as we show in the following lemma, this implies that if $e$ is a critical edge, then there are only limited options for $f$, and also that $e$ and $f$ cannot both be reticulation edges. Figure 4.5 illustrates the cases where $e = (u, v)$ is a critical edge.

**Lemma 4.12.**
*Let $N \in \mathcal{TC}_n$ and $(e, f) \in \Theta^{\mathrm{SNPR}^0}(N)$. Let $e = (u, v)$ and $f$ not be a descendant of $e$. Then $N' = (e, f)(N)$ is a tree-child network if and only if one of the following cases holds:*

*(i) $e$ is a reticulation edge and $f$ is not a reticulation edge;*

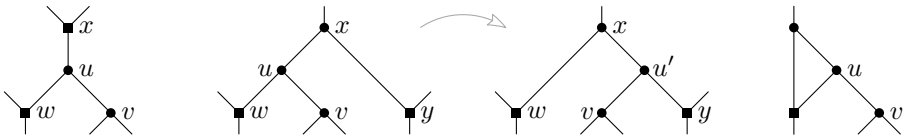


Figure 4.5: Illustration of the cases $r_2$, $r_3$, and triangle, where there are only trivial SNPR$^0$ operations that prune $(u, v)$ or, in the case of $r_3$, exactly one nontrivial SNPR$^0$ as shown. This is formalised in Lemma 4.12

*(ii)* *$e$ is a pure tree edge that is not critical;*

*(iii)* *$e$ is a critical edge and $f$ is incident to $u$;*

*(iv)* *$e$ is a critical edge of an $r_3$ structure with underlying path $w, u, x, y$ such that $u$ and $y$ are the children of $x$ and $f = (x, y)$;*

*(v)* *$e$ is a critical edge of a triangle and $f$ is the long side of the triangle.*

*Proof.* We prove this by considering the different types of $e = (u, v)$. By the definition of an SNPR$^0$ operation, $u$ cannot be a reticulation. Thus, $e$ cannot be a pure reticulation edge or an impure tree edge. Let $e$ be an impure reticulation edge, i.e. let $v$ be a reticulation. Then, since $N \in \mathcal{TC}_n$, the sibling $w$ of $v$ with shared parent $u$ is a tree vertex. Thus after pruning $e$ and suppressing $u$, the parent of $u$ has, in $N'$, the vertex $w$ as tree child. Hence, a reticulation edge can always be pruned. Now, if $f = (x, y)$ is a reticulation edge, then the new vertex $u'$ in $N'$, resulting from the subdivision of $f$, has the two children $v$ and $y$, which are both reticulations. Thus, if $e$ is a reticulation edge, then $f$ cannot be a reticulation edge. If $f$ is not a reticulation edge (Item (i)), then, in $N'$, the new vertex $u'$ has the tree child $y$, the vertex $x$ has the tree child $u'$, and all other vertices stay unaffected.

Next, let $e$ be a pure tree edge. If $e$ is not critical (Item (ii)), then either the sibling of $v$ or the sibling of $u$ is a tree vertex. Without loss of generality let $w$, the sibling of $v$, be a tree vertex. Then, after pruning $e$ and suppressing $u$, the parent $x$ of $u$ has $w$ as a tree child in $N'$. Since $v$ is a tree vertex, regrafting to any edge $f$ does not create a non-leaf vertex without tree child. Hence, $N'$ is tree child.

If $e$ is critical and $f$ incident to $e$ (Item (iii)), as $f$ is not a descendant edge of $e$, then $N' = N$ and $N'$ is thus tree child. If $e$ is the critical edge of an $r_2$ structure, then clearly $f$ being incident to $e$ is the only option for $N'$ to be tree child. If $e$ is the critical edge of an $r_3$ structure, then after pruning $e$ and suppressing $u$, the parent $x$ of $u$ has the two reticulations $y$ and $w$ as children if and only if $e$ is not regrafted to an incident edge and if $f \neq (x, y)$ (Item (iv)). In the case that the $r_3$ structure is a triangle, $f = (x, y)$ implies that $f$ is the long side of the triangle (Item (v)). Since we covered all types of $e$, the described choices of $e$ and $f$ cover all tree-child respecting SNPR$^0$ operations on $N$. □

We now know when SNPR$^0$ operations respect the tree-child property. We can thus continue counting them. Recall that $E_R$ denotes all reticulation edges and that $E_{T^*}$ denotes all pure non-critical tree edges. Furthermore, recall that $\delta_T(e)$ counts only descendant edges of $e$ that are tree edges.

**Lemma 4.13.**
*Let $N \in \mathcal{TC}_n$. Then the number of tree-child respecting* SNPR$^0$ *operations on $N$ is*

$$|\Theta_{\mathcal{TC}}^{\mathrm{SNPR}^0}(N)| = 4n^2 + 10nr - 2n(r_2 + r_3) - 6n + 2r^2 - 3r(r_2 + r_3) - 5r + 4r_2 + 5r_3 + 2 - \sum_{e \in E_{T^*}} \delta(e) - \sum_{e \in E_R} \delta_T(e).$$

*Proof.* Following Lemma 4.12, we prove this by distinguishing the different types of the pruned edge $e$. We use the fact that $N$ has $m = 2n + 3r - 1$ edges. First, any reticulation edge $e = (u, v)$ can be regrafted to any non-reticulation edge that is not descendant of $e$. Hence, there are the following many such operations:

$$2r(m - 2r) - \sum_{e \in E_R} \delta_T(e) = 4nr + 2r^2 - 2r - \sum_{e \in E_R} \delta_T(e) \tag{4.5}$$

Equation (4.5) uses $\delta_T(e)$ instead of $\delta(e)$, since we would otherwise double count the forbidden operations of regrafting to an edge that is reticulation edge and descendant of the pruned edge.

If $e \in E_{T^*}$, i.e. a pure non-critical tree edge, then $e$ can be pruned and regrafted to every edge not $e$ itself or a descendant of $e$. Hence, with Observation 4.5, there are the following many such operations:

$$\begin{aligned}(m - 3r - r_2 - r_3)(m - 1) - \sum_{e \in E_{T^*}} \delta(e) \\ = 4n^2 + 6nr - 2n(r_2 + r_3) - 6n - 3r(r_2 + r_3 + 1) + 2r_2 + 2r_3 + 2 - \sum_{e \in E_{T^*}} \delta(e)\end{aligned} \tag{4.6}$$

If $e$ is the critical edge of an $r_2$ or $r_3$ structure (including triangles), then there are only 2 or 3 operations, respectively. Hence, there are the following many such operations:

$$2r_2 + 3r_3 \tag{4.7}$$

Adding Equations (4.5) to (4.7) together, the lemma follows. □

**Counting trivial SNPR$^0$ operations.** Pruning an edge and regrafting it at the same edge is a trivial SNPR$^0$ operation. Another trivial SNPR$^0$ operation $(e, f)$ arises for every triangle where $e$ is the triangle's critical edge and $f$ is its long side. Furthermore, the reticulation edges of a triangle induce a trivial operation each, as the proof of the following lemma shows.

**Lemma 4.14.**
*Let $N \in \mathcal{TC}_n$. Then there are $4n + 4r + 3t_3 - 4$ trivial operations in $\Theta_{\mathcal{TC}}^{\mathrm{SNPR}^0}(N)$.*

*Proof.* Let $(e, f) \in \Theta_{\mathcal{TC}}^{\mathrm{SNPR}^0}(N)$ with $e = (u, v)$ and $f = (x, y)$. The operation $(e, f)$ can be trivial in three ways. First, if $f$ is incident to $e$ at $u$. The root edge and edges $(u, v)$ with a reticulation $u$ are not prunable. Therefore, there are $m - r - 1$ prunable edges and $2(m - r - 1)$ trivial tree-child respecting SNPR$^0$ operations.

Second, $f$ is isomorphic to the edge $g$ created by pruning $e$ and suppressing $u$. However, this can only happen if $f$ and $g$ are parallel edges, since by Lemma 4.3 there are no pairs of isomorphic edges in a tree-child network. This means that the critical edge of a triangle gets pruned. Thus, there are $t_3$ many trivial tree-child respecting SNPR$^0$ operations of that type.

Third, let $f$ be neither of the above. Let the edges of $N$ be labelled and then in $N' = (e, f)(N)$ let all labels be as in $N$ except those affected $(e, f)$. Let the regrafted edge have the label $e$. Then, since $N' = N$, there has to be an edge $e' = (u', v')$ in $N'$ that is, without label, the same edge as $e$ in $N$. By the choice of $f$, this cannot be $e$. The edges $e$ and $e'$ got, so to say, swapped. Then, since by Lemma 4.3 every vertex is unique, $v = v'$ follows. The edges $e$ and $e'$ are thus reticulation edges. For $N' = N$, clearly, $e$ and $e'$ have to be the reticulation edges of a triangle: If we prune the long side of a triangle and regraft

it to the critical edge of the triangle, it results again in $N$. An equivalent operation exists for the bottom edge of the triangle. Hence, there are $2t_3$ such trivial tree-child respecting $\text{SNPR}^0$ operations.

Furthermore, these three cases do not overlap and we thus counted all trivial tree-child respecting $\text{SNPR}^0$ operations on $N$. Since $2(m - r - 1) + 3t_3 = 4n + 4r + 3t_3 - 4$, the lemma follows. □

**Counting redundant $\text{SNPR}^0$ operations.** In Lemma 4.8 in Section 4.2.2 we stated that $\text{SNPR}^0$ operations on trees are redundant only if they correspond to an $\text{NNI}^0$ operation. This result is also important for tree-child networks. Before we fully restate and proof Lemma 4.8, we make some general observations on how an $\text{SNPR}^0$ redundancy can occur.

Figure 4.6 illustrates how three $\text{SNPR}^0$ operations correspond to an $\text{NNI}^0$ operation with axis $(x, u)$. There, the siblings $v$ and $w$ are children of $u$ and their uncle $y$ is a child of $x$. Now, the three redundant $\text{SNPR}^0$ operations could be described as follows. First, $((u, v), (x, y))$ prunes $v$ and regrafts it as sibling of $y$. Second, $((x, y), (u, v))$ prunes $y$ and regrafts it as sibling of $v$. Third, $((u, w), (p, x))$ prunes $w$ and regrafts it above $x$, thus making $v$ and $y$ siblings. In general, to find redundancies of $\text{SNPR}^0$ operations, we can fix two vertices that stand in a certain relation in $N'$, but not yet in $N$. Then, to create this relation, say making $v$ and $y$ siblings, we can either regraft one as sibling of the other or alter the path between them. We formalise this in Lemma 4.15, after we describe the initial situation more precisely.

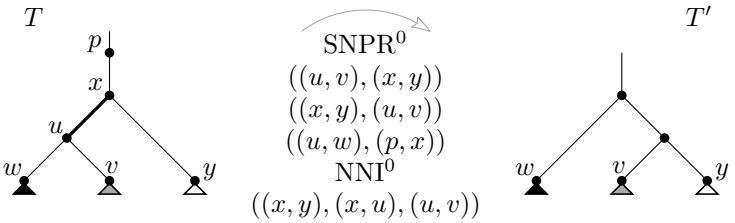


Figure 4.6: Correlation of an NNI operation with a pure inner tree edge as axis and three SNPR operation, all being pairwise redundant.

Let $N, N' \in \mathcal{TC}_n$ be neighbours with $N' = \theta(N), \theta \in \Theta_{\mathcal{TC}}^{\text{SNPR}^0}(N)$. Let the vertices in both $N$ and $N'$ be labelled and let $\theta$ preserve these labels, except of course for removed or new vertices. Also let $v$ and $y$ be distinct vertices in $N$ and $N'$ with the same labels and such that neither is ancestor of the other in both $N$ and $N'$. We now say that $v$ and $y$ are in a *desired relation* if one of the following holds:

- The vertex $v$ is a sibling, an uncle, or a nephew of $y$ in $N'$ via a path $P'$, but $v$ is in a different relation to $y$ in $N$.
- The vertex $v$ is an uncle or a nephew of $y$ in $N'$ via a path $P'$ and in $N$ via a path $P \neq P'$.

In the second condition, $P \neq P'$ means that the labels of the vertices on $P$ differ for a least one vertex from the labels of the vertices on $P'$.

**Lemma 4.15.**
*Let $N, N' \in \mathcal{TC}_n$ and $\theta \in \Theta_{\mathcal{TC}}^{\text{SNPR}^0}(N)$ with $\theta(N) = N' \neq N$. Let $v$ and $y$ be in a desired relation via a path $P'$ in $N'$.*
*Then there are only the following possibilities of how $\theta$ operates on $N$ to yield $N'$:*

*(i) an incoming edge of $v$ or $y$ gets pruned and regrafted such that $v$ becomes sibling, uncle or nephew, respectively, of $y$;*

*(ii) an incoming edge of the parent of $v$ or $y$ gets pruned and regrafted such that $v$ becomes uncle or nephew, respectively, of $y$;*

*(iii) an edge $e = (u, w)$ with $u$, but not $w$, being on a path connecting $v$ and $y$, gets pruned yielding $P'$;*

*(iv) an edge gets regrafted to a path connecting $v$ and $y$ yielding $P'$.*

*Proof.* The existence of $v$ and $y$ in $N'$ after applying $\theta$ to $N$ means that $\theta$ does not prune an outgoing edge of $v$ or $y$. For $\theta$ to yield the desired relation and path $P'$ in $N'$, $\theta$ can either alter an existing path between $v$ and $y$ by one vertex, i.e. (iii) or (iv), or prune an edge of an existing path between $v$ and $y$ and regraft it such that a desired path $P'$ gets created, i.e. (i) or (ii). □

Applying Lemma 4.15 means that we can consider an $\mathrm{SNPR}^0$ operation $\theta$, find two vertices $v$ and $y$ in the resulting network $N'$ that are in desired relation, and then check whether other $\mathrm{SNPR}^0$ operations corresponding to one of the possibilities listed in the lemma exist that are redundant to $\theta$. We can now prove Lemma 4.8, restated here for convenience.

**Lemma 4.8** (restated)**.**
*Let $T \in \mathcal{T}_n$ and let $\theta, \theta' \in \Theta^{\mathrm{SNPR}^0}(T)$ be distinct, nontrivial, and redundant with $\theta(T) = T'$. Then there exists an $\mathrm{NNI}^0$ operation $\sigma \in \Theta^{\mathrm{NNI}^0}(T)$ such that $\sigma(T) = T'$. Furthermore, every nontrivial redundancy set of $\Theta^{\mathrm{SNPR}^0}(T)$ has size three.*

*Proof of Lemma 4.8.* The lemma states that if two nontrivial $\mathrm{SNPR}^0$ $\theta$ and $\theta'$ are redundant on a phylogenetic tree, then there is an $\mathrm{NNI}^0$ that is redundant to them. Let $v$ and $y$ be two distinct vertices that are not siblings in $T$, but that, under preserving of labels by $\theta$, are siblings in $T'$. Then $v$ and $y$ are in a desired relation. With the cases (i) and (iii) of Lemma 4.15 we see that $\theta$ and $\theta'$ correspond to an $\mathrm{NNI}^0$ and, moreover, that a third $\mathrm{SNPR}^0$ is redundant to them. Hence, the redundancy set containing $\theta$ has size three. □

We now count the number of tree-child respecting $\mathrm{SNPR}^0$ that we can discard due to redundancy. In the proof of the following proposition, we will see that redundancies only arise from few different sources; for example from $\mathrm{NNI}^0$ operations with the axis being a pure inner tree edge. The other sources of redundancies are operations that create a triangle, the reticulation edges of a triangle (see Figure 4.7), and the existence of tree-branching triangles, diamonds, and $t_4$ trapezoids (see Figures 4.8 to 4.10). We note that an $\mathrm{NNI}^0$ on a phylogenetic network with the axis being a reticulation edge or an impure tree edge does not correspond to a redundancy set of $\mathrm{SNPR}^0$.

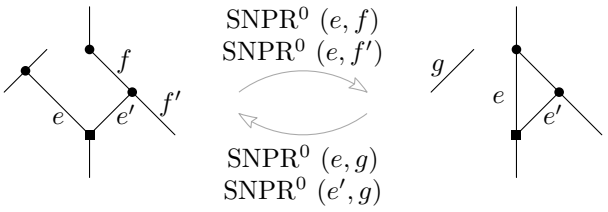


Figure 4.7: Redundant $\mathrm{SNPR}^0$ operations due to the creation of a triangle and due to the reticulation edges of a triangle.

**Proposition 4.16.**
*Let $N \in \mathcal{TC}_n$.*
*Then the number of nontrivial redundant* $\mathrm{SNPR}^0$ *of* $\Theta_{\mathcal{TC}}^{\mathrm{SNPR}^0}(N)$ *minus the number of redundancy sets of nontrivial* $\mathrm{SNPR}^0$ *of* $\Theta_{\mathcal{TC}}^{\mathrm{SNPR}^0}(N)$ *is*

$$2n(2+t_3) + r(2+t_3) + 4r_1 - 2r_3 - 8t_3 + t_3^* + 3d_4 + t_4 - 8 - \sum_{e \in E_{\bar{t}_3}} \delta_T(e).$$

*Proof.* Let $\theta = (e, f) \in \Theta_{\mathcal{TC}}^{\mathrm{SNPR}^0}(N)$ such that $\theta(N) = N' \neq N$. Let $e = (u, v)$, $f = (x, y)$.

The operations we want to count are those that we want to *discard* when counting neighbours. To count all the operations we can discard, we go through the sources of redundancy one by one and determine the sizes of the corresponding redundancy sets. In order to find all sources, the idea of this proof is to consider when and where $\theta$ can be redundant with other operations. To cope with all possibilities, we fix a reticulation including a cycle for which this reticulation is the lowest vertex (i.e. descendant of all other vertices on the cycle). Then, when applying $\theta$, it can be distinguished whether the size of this cycle gets decreased, increased, or whether only the order of edges with start vertex on the cycle gets altered. Therefore in the following case distinction, let $[c \to c']$ denote the change from a cycle of size $c$ to size $c'$. Note that there can be no cycle of size 1 or 2. A cycle size of 0 means either no cycle, i.e. a tree, or that no cycle is under consideration or of any influence to redundancies of $\theta$.

One source of redundancy that we already identified are $\mathrm{NNI}^0$ operations with a pure inner tree edge as axis. A phylogenetic network has $n + r_1 - 2$ pure inner tree edges (all edges minus any incident to reticulations, leaves, or the root) each inducing two $\mathrm{NNI}^0$ operations. However, if the axis is part of an $r_3$ structure, then one of the possible $\mathrm{NNI}^0$ for this axis is not tree-child respecting. Also, if it is part of a triangle, the operation is either trivial or not tree-child respecting. There are thus $2(n + r_1 - 2) - r_3 - t_3$ $\mathrm{NNI}^0$ operations of interest, each with an $\mathrm{SNPR}^0$ redundancy set of size three. Therefore, we discard the following many nontrivial tree-child respecting $\mathrm{SNPR}^0$ operations:

$$4n + 4r_1 - 2r_3 - 2t_3 - 8 \tag{4.8}$$

We freely use Lemma 4.3 throughout the remainder of this proof.

$[0 \to 0]$ If no cycle is involved, the part where the $\mathrm{SNPR}^0$ operations make changes is tree-like and there are thus only tree edges. It follows thus from Lemma 4.8 that the redundancy comes from an $\mathrm{NNI}^0$. Hence, these redundancies are covered by Equation (4.8).

$[3 \to 3]$ A triangle with fixed reticulation has only one shape and thus cannot be transformed into another one with a single SNPR operation.

In the following two cases, we will see redundancies due to the reticulation edges of triangles. We count the $\mathrm{SNPR}^0$ operations we discard afterwards.

$[3 \to 4]$ A triangle can be transformed into a cycle of size four either by pruning one of its edges and regrafting it to an edge outside of the triangle, thus including this edge as third outgoing edge, or by regrafting an edge from outside to the triangle. We will see that considering only the latter case also covers the former case.

Let the edge $e = (u, v)$ have distance at least two to the triangle and let $f = (x, y)$ be an edge of the triangle. Assuming $e$ has distance greater than two, it is clear (for example with the analysis of Lemma 4.15) that there can only be a redundancy,

if $e$ is the reticulation edge of another triangle and $f$ the top side of the triangle. However, no SNPR$^0$ pruning an edge of the fixed triangle or incident to it can be redundant to this operation and thus any redundancy would be accredited to the other triangle. Therefore, assuming now that $e$ has distance two to the triangle, the following cases can be distinguished.

(i) $u$ **is parent of triangle,** $f$ **is long side of triangle.** Requiring that $e$ is a tree edge, the triangle gets transformed into a diamond. Using the analysis of Lemma 4.15 with $y$ and $v$ as siblings in $N'$ yields that there are four redundant SNPR$^0$ operations, as illustrated by Figure 4.8. Three SNPR$^0$ operations can be associated to the NNI$^0$ operation $(e, c, f)$ where $c$ is the incoming edge of the triangle. Also, pruning one of the two reticulation edges of the triangle and regrafting it to $e$ is redundant to doing the same with the other. We note that the SNPR$^0$ operation $(f, e)$ corresponds to both redundancies.

(ii) $u$ **is sibling of reticulation of triangle,** $f$ **is long side of triangle.** Requiring that both $e$ and its sibling edge are tree edges (and thus that the triangle is a tree-branching triangle), this transforms the triangle again into a diamond (see again Figure 4.8). This time the analysis, again with $y$ and $v$ as siblings, yields a redundancy set of size two, namely regrafting $e$ and its sibling edge $e'$ to $f$. This means that each tree-branching triangle of $N$ induces a redundancy set of size two. We thus discard one SNPR$^0$ operation per such triangle:

$$t_3^* \tag{4.9}$$

(iii) $u$ **is parent of triangle,** $f$ **is top side of triangle.** Without a requirement on $e$, this transforms the triangle into a trapezoid (see Figure 4.9). Like in (i), the analysis with $v$ being uncle of $y$, yields again an NNI$^0$ operation redundancy with an overlap of a triangle reticulation edges redundancy. Furthermore, this can coincide with a transformation of another triangle into a trapezoid of the next case.

(iv) $u$ **is sibling of reticulation of triangle,** $f$ **is top side of triangle.** This requires that the sibling edge of $e$ is a tree edge and transforms the triangle into a trapezoid (see again Figure 4.9). The analysis yields the same as in the previous case. If the sibling edge of $e$ is not a tree edge, we would have an $r_3$ structure and $N'$ would not be tree child.

Furthermore, if $e$ is a tree edge, the case is equivalent to $f$ being the bottom side of the triangle.

(v) $u$ **is parent of triangle,** $f$ **is bottom side of triangle.** The analysis yields that there is no redundancy of SNPR$^0$ operations here.

$[3 \to c, c \geq 5]$ Since it is not possible to add two outgoing edges to a triangle by regrafting them to the triangle with a single SNPR$^0$ operation, the size can only be increased by pruning an edge of the triangle and regrafting it to an edge $f$ at the desired distance. This yields the same neighbour for the two reticulation edges, but different cycles for the top side of the triangle and one of its reticulation edges.

From the last two cases, we know that two SNPR$^0$ operations $\theta$ and $\theta'$ that prune the two different reticulation edges of a triangle and regraft it to the same edge are always redundant. In cases, where $\theta$ and $\theta'$ are also redundant to SNPR$^0$ operations corresponding to an NNI$^0$ operation, either $\theta$ or $\theta'$ also corresponds to that NNI$^0$ operation. In any case,

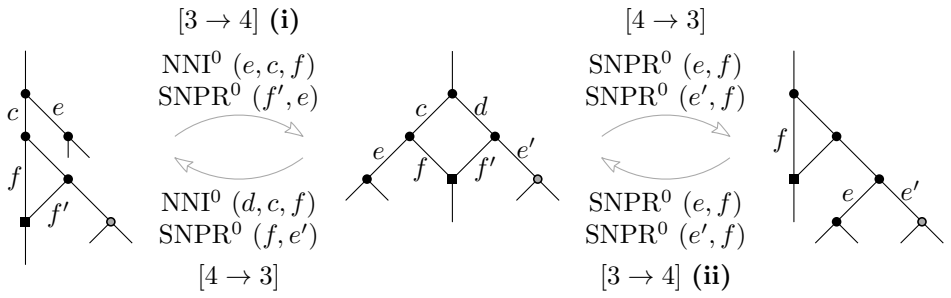


Figure 4.8: Redundant operations that transform triangles into diamonds and vice versa, covering parts of the cases $[3 \to 4]$ and $[4 \to 3]$.

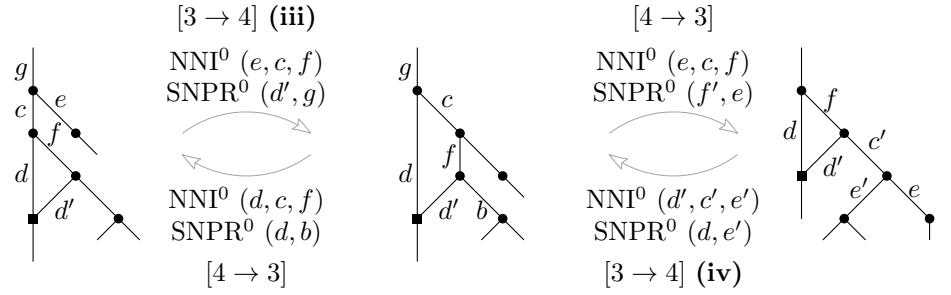


Figure 4.9: Redundant operations that transform triangles into trapezoids and vice versa, covering parts of the cases $[3 \to 4]$ and $[4 \to 3]$.

without loss of generality, we can discard all (nontrivial) $\text{SNPR}^0$ operations that prune an edge $e \in E_{t_3^-}$, i.e the bottom side of a triangle:

$$2nt_3 + rt_3 - 4t_3 - \sum_{e \in E_{t_3^-}} \delta_T(e) \tag{4.10}$$

$[4 \to 3]$ This is basically the analysis of $[3 \to 4]$ backwards. See again Figures 4.8 and 4.9 for illustrations.

$[c \to 3, c \geq 5]$ To create a triangle with a specific reticulation, one way is to prune one of its reticulation edges and to regraft it to an edge incident to the other reticulation edge. This is the reverse of $[3 \to c]$ and yields two redundancy sets of size two.

The second possibility is to to prune the parent edge of one of the reticulation edges and regraft it to the other reticulation edge. Again, as seen in $[3 \to c]$, this is not redundant to the other way or other operations.

The last two cases covered the creation of triangles. With Equation (4.9) we accounted for redundancies from a tree-branching triangle to a diamond. With Equation (4.11) we do the reverse:

$$d_4 \tag{4.11}$$

Furthermore, as the reverse of Equation (4.10), each reticulation edge that is not part of a triangle corresponds to two redundant $\text{SNPR}^0$ operations. In the case that a four cycle is created, this can coincide with a redundancy due to an $\text{NNI}^0$ operation. Like before, we can discard one of the two operations:

$$2r - 2t_3 \tag{4.12}$$

$[4 \to 4]$ To change a cycle of size four into another cycle of size four, one can either change the order of the outgoing edges of a $t_4$ trapezoid, which is then equivalent to Case $[0 \to 0]$, or transform a $t_4$ trapezoid into a diamond or vice versa (see Figure 4.10). Applying Lemma 4.15 for any of the directions with two appropriate vertices yields redundancy sets of size four. We see that three edges correspond to an $\mathrm{NNI}^0$ operation. We have thus already counted a neighbour and can discard the fourth $\mathrm{SNPR}^0$ operation. We note that there are two different transformations from a diamond to a $t_4$ trapezoid distinguished by the order of the resulting trapezoid's outgoing edges. Hence, we discard the following many $\mathrm{SNPR}^0$ operations:

$$2d_4 + t_4 \tag{4.13}$$

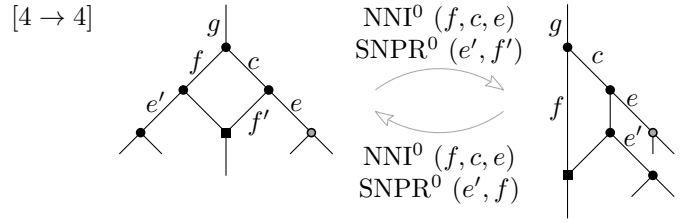


Figure 4.10: Redundant operations that transform diamonds into $t_4$ trapezoids and vice versa, illustrating the case $[4 \to 4]$.

$[c \to c+1, c \geq 4]$ Adding a branch to a cycle of size at least four, and thus increasing its size by one, is, by using Lemma 4.15, only possible if the operations correspond to an $\mathrm{NNI}^0$.

$[c \to c+x, c \geq 4, x \geq 2]$ Unlike in the case $[3 \to 5]$ there is obviously no redundancy of any edges of the cycle anymore.

$[c+1 \to c, c \geq 4]$ This is the reverse of $[c \to c+1]$ and there are thus only redundancies due to an $\mathrm{NNI}^0$.

$[c+x \to c, c \geq 4, x \geq 2]$ As the reverse of the case $[c+x \to c]$, there are no redundancies in this case.

We have now covered all cases of transformations of a cycle into another one. We have further identified the different sources of redundancies and discarded nontrivial tree-child respecting $\mathrm{SNPR}^0$ operations accordingly. The statement follows by adding Equations (4.8) to (4.13) together. □

**$\mathrm{SNPR}^0$ tree-child neighbourhood size.** The $\mathrm{SNPR}^0$ tree-child neighbourhood of $N$ is determined by the number of tree-child respecting $\mathrm{SNPR}^0$ operations on $N$ (Lemma 4.13), from which trivial operations are subtracted (Lemma 4.14), and operations that yield redundant neighbours are discarded (Proposition 4.16).

**Theorem 4.17.**
*Let $N \in \mathcal{TC}_n, n \geq 2$. The $\mathrm{SNPR}^0$ tree-child neighbourhood $U_{\mathcal{TC}}^{\mathrm{SNPR}^0}(N)$ of $N$ has size*

$$\begin{aligned}|U_{\mathcal{TC}}^{\mathrm{SNPR}^0}(N)| &= 4n^2 + 10nr - 2n(r_2 + r_3 + t_3) - 14n + 2r^2 - r(3r_2 + 3r_3 + t_3) - 11r \\ &\quad - 4r_1 + 4r_2 + 7r_3 + 5t_3 - t_3^* - 3d_4 - t_4 + 14 \\ &\quad - \sum_{e \in E_{T^*}} \delta(e) - \sum_{e \in E_R} \delta_T(e) + \sum_{e \in E_{\bar{t}_3}} \delta_T(e).\end{aligned}$$

Table 4.1 lists the values for the parameters of the tree-child network $N$ from Figure 4.1. Applying these values to Theorem 4.17 we get that $N$ has, just as depicted, seven different $\mathrm{SNPR}^0$ tree-child neighbours.

Table 4.1: Parameters for the network $N$ from Figure 4.1, which has an $\mathrm{SNPR}^0$ tree-child neighbourhood of size seven.

| parameter | description | value for $N$ |
|---|---|---|
| $n$ | # leaves | 3 |
| $r$ | # reticulations | 1 |
| $r_1$ | # reticulations with leaf as child | 1 |
| $r_2$ | # $r_2$ structures | 0 |
| $r_3$ | # $r_3$ structures | 0 |
| $t_3$ | # triangles | 0 |
| $t_3^*$ | # tree-branching triangles | 0 |
| $d_4$ | # diamonds | 0 |
| $t_4$ | # $t_4$ trapezoids | 1 |
| $\sum_{e\in E_{T^*}}\delta(e)$ | # descendant edges of pure non-critical tree edges | 13 |
| $\sum_{e\in E_R}\delta_T(e)$ | # descendant tree edges of reticulation edges | 2 |
| $\sum_{e\in E_{\bar{t}_3}}\delta_T(e)$ | # descendant tree edges of triangle bottom sides | 0 |

Note that if $N$ is a phylogenetic tree, the formula from Theorem 4.17 becomes the formula from Theorem 4.10.

**$\mathrm{SNPR}^+$ and $\mathrm{SNPR}^-$ tree-child neighbourhood size.** We now consider $\mathrm{SNPR}^+$ and $\mathrm{SNPR}^-$ operations and count again the number of such operations first. For this, recall that $E_{PS}$ denotes the set of edges that are pure tree edges with a sibling pure tree edge.

**Lemma 4.18.**
*Let $N \in \mathcal{TC}_n$ with $n \geq 2$. Then*

$$|\Theta_{\mathcal{TC}}^{\mathrm{SNPR}^+}(N)| = 4n^2 - 2nr - 8n - 2r^2 + 2r + 4 - \sum_{e\in E_{PS}} \delta_T(e), \text{ and}$$

$$|\Theta_{\mathcal{TC}}^{\mathrm{SNPR}^-}(N)| = 2r.$$

*Proof.* For an $\mathrm{SNPR}^+$ operation $(e, f)$, which adds an edge from $f$ to $e$, for $(e, f)(N)$ to be tree child, $e$ has to be a pure tree edge with a sibling pure tree edge. This implies that $e \neq e_\rho$ and that $e$ cannot be incident to a reticulation or sibling edge of a reticulation edge. Otherwise, if $e$ would be incident to a reticulation, this would yield a pure reticulation edge, and if $e$ would be the sibling edge of a reticulation edge, this would yield a vertex with two reticulations as children. In either case $(e, f)(N)$ would not be tree child. Thus every reticulation induces a set of five unsuitable edges, consisting of the three edges incident to it and their two sibling edges. Since $N \in \mathcal{TC}_n$, clearly these sets are disjoint for every pair of reticulations of $N$. There are thus $m - 5r - 1$ choices for $e$.

Next, by the definition of an $\mathrm{SNPR}^+$ operation, the edge $f$ cannot be a descendant of $e$. Furthermore, $f$ cannot be a reticulation edge or $e$. Otherwise, if $f$ would be a reticulation edge, this would yield a vertex with two reticulations as children, and if $f = e$,

the operation would create a parallel edge. In either case $(e, f)(N)$ would not be tree child. Clearly any other choice of $f$ is fine. For any feasible choice of $e$, there are thus $m - 2r - 1 - \delta_T(e)$ choices of $f$. With the $\delta_T(e)$ summing up to $\sum_{e \in E_{PS}} \delta_T(e)$ over all choices of $e$ and $(m-5r-1)(m-2r-1) = 4n^2-2nr-8n-2r^2+2r+4$ the first statement follows.

Concerning $\Theta_{\mathcal{TC}}^{\mathrm{SNPR}^-}(N)$, we know by Theorem 3.13 that removing a reticulation edge of a tree-child network yields again a tree-child network. There are thus $2r$ tree-child respecting $\mathrm{SNPR}^-$ operations on $N$. □

We already noted that $\mathrm{SNPR}^+$ and $\mathrm{SNPR}^-$ operations are never trivial. However, for both of these types of operations redundancies might exist. Like for most SNPR redundancies, these redundancies are equivalent to $\mathrm{NNI}^+$ and $\mathrm{NNI}^-$ operations, as the proof of the following proposition shows.

**Proposition 4.19.**
*Let $N \in \mathcal{TC}_n$ with $n \geq 2$.*
*The $\mathrm{SNPR}^+$ tree-child neighbourhood $U_{\mathcal{TC}}^{\mathrm{SNPR}^+}$ of $N$ has size*

$$|U_{\mathcal{TC}}^{\mathrm{SNPR}^+}(N)| = 4n^2 - 2nr - 10n - 2r^2 + 4r + 6 - \sum_{e \in E_{PS}} \delta_T(e),$$

*and the $\mathrm{SNPR}^-$ tree-child neighbourhood $U_{\mathcal{TC}}^{\mathrm{SNPR}^-}$ of $N$ has size*

$$|U_{\mathcal{TC}}^{\mathrm{SNPR}^-}(N)| = 2r - t_3.$$

*Proof.* This proof uses the concept of the proof of Proposition 4.16. For the first part, we assume that the considered $\mathrm{SNPR}^+$ operations are tree-child respecting.

$[0 \to 3]$ Let $f = (u, v)$ be a tree edge with $v$ having two outgoing pure tree edges $e = (v, w)$ and $e' = (v, y)$. Then a reticulation and a triangle can be added by the $\mathrm{SNPR}^+$ operation $(e, f)$, which adds an edge from $f$ to $e$. This is however redundant to the $\mathrm{SNPR}^+$ operation $(e, e')$. It follows by the uniqueness of $e$ that there are no further redundant $\mathrm{SNPR}^+$ operations. Furthermore, note that these operations are redundant to an $\mathrm{NNI}^+$.

$[0 \to c, c \geq 4]$ Similar to $[0 \to 3]$, the edge $e$ that gets subdivided for the new reticulation is unique. However, for a new cycle of size at least 4, there are no two edges that can be chosen interchangeably to be subdivided for the source of the new reticulation edge to yield the same network $N'$.

There are $n - r - 1$ pairs of siblings of pure tree edges in $N$. To account for redundancy, we can thus discard $2(n - r - 1)$ $\mathrm{SNPR}^+$ operations. The first part follows then from Lemma 4.18.

$[3 \to 0]$ If a triangle gets removed, removing one of the reticulation edge of the triangle is redundant to removing the other. Since no reticulations are isomorphic in $N$, there can be no further reticulation edges in $N$ that if removed would yield the same network. These $\mathrm{SNPR}^-$ operations are thus equivalent to an $\mathrm{NNI}^-$ operation of the respective triangle.

$[c \to 0, c \geq 4]$ The reticulation edges of a cycle of size at least four are neither isomorphic nor can they change roles like in triangles. Thus removing one of the reticulation edges cannot be redundant to removing the other.

There are $t_3$ many NNI$^-$ operations in $N$. Discarding one SNPR$^-$ operation per triangle, the second part follows again from Lemma 4.18. □

We can again consider the tree-child network $N$ from Figure 4.1. Since $N$ has one reticulation, but no triangles, $N$ has two SNPR$^-$ neighbours. Using Table 4.1 and the fact that $\sum_{e\in E_{PS}}\delta_T(e) = 2$, we get that $N$ has six SNPR$^+$ tree-child neighbours.

**SNPR tree-child neighbourhood size.** To obtain the total size of the SNPR tree-child neighbourhood of a tree-child network $N$ we can now add together the sizes of the SNPR$^0$, SNPR$^+$ and SNPR$^-$ neighbourhoods (Theorem 4.17 and Proposition 4.19).

**Theorem 4.20.**
*Let $N \in \mathcal{TC}_n$ with $n \geq 2$.*
*The* SNPR *tree-child neighbourhood $U^{\mathrm{SNPR}}_{\mathcal{TC}}$ of $N$ has size*

$$\begin{aligned}|U^{\mathrm{SNPR}}_{\mathcal{TC}}(N)| = {} & 8n^2 + 8nr - 2n(r_2 + r_3 + t_3) - 24n - r(3r_2 + 3r_3 + t_3) - 5r \\ & - 4r_1 + 4r_2 + 7r_3 + 4t_3 - t_3^* - 3d_4 - t_4 + 20 \\ & - \sum_{e\in E_{T^*}} \delta(e) - \sum_{e\in E_R} \delta_T(e) - \sum_{e\in E_{PS}} \delta_T(e) + \sum_{e\in E_{t_3^-}} \delta_T(e).\end{aligned}$$

**Bounds.** The formula for the SNPR tree-child neighbourhood of a tree-child network depends on a lot of parameters. It is therefore of interest to see how small and big a neighbourhood can get in terms of $n$.

**Proposition 4.21.**
*Let $n \geq 2$. Then*

$$\begin{aligned} n - 1 \leq \min_{N\in\mathcal{TC}_n} \{|U^{\mathrm{SNPR}}_{\mathcal{TC}}(N)|\} \leq \frac{3}{2}n^2 - \frac{7}{2}n + 2, \text{ and} \\ 8n^2 - \mathcal{O}(n\log_2 n) \leq \max_{N\in\mathcal{TC}_n} \{|U^{\mathrm{SNPR}}_{\mathcal{TC}}(N)|\} < 16n^2 - 38n + 26.\end{aligned}$$

*Proof.* We first establish a lower bound for the minimum neighbourhood size of a tree-child network. Let $N \in \mathcal{TC}_n$ with $n \geq 2$. By Proposition 4.19, we know that $|U^{\mathrm{SNPR}^-}_{\mathcal{TC}}(N)| = 2r - t_3$. Each reticulation gives rise to two different SNPR$^-$ operations with redundancy sets of size at most two. Furthermore, a reticulation edge can be added from the root edge $e_\rho$ to every other pure tree edge that is not sibling edge of a reticulation edge. There are $2n-2r-2$ such edges. Note that $t_3 \leq n-1$. There are thus at least $2n-2r-2+2r-t_3 = 2n-t_3-2 \geq n-1$ SNPR tree-child neighbours of $N$. This is sharp for a tree-child network with $n = 2$ and $r = 1$.

Next, we look at an upper bound for the minimum neighbourhood size of a tree-child network. For this we consider a family of tree-child networks, where each has a relative small neighbourhood. Let $N_r \in \mathcal{TC}_n, n \geq 2$ be a chain of $r$ triangles, where each triangle (except for the top one) is a child of the triangle above, like $N_4$ in Figure 4.11. Since $N_r$ has $n-1$ reticulations, $N_r$ has no SNPR$^+$ neighbours. Removing a reticulation edge from one of the triangles corresponds to one of $r = n-1$ different SNPR$^-$ neighbours of $N_r$. Concerning SNPR$^0$ operations, the only prunable non-critical edges are the long sides of the triangles (and the bottom sides, which however behave redundantly and are thus ignored). Since these are reticulation edges, they can, when pruned, only be regrafted to tree edges that are not their descendants. For the long side of the triangle closest to the root three such edges exist, which however correspond to trivial operations. For the long

side of the triangle below six such edges exist, of which again three yield trivial operation. Thus, in total there are

$$3+6+9+12+\ldots+(n-2)3=\sum_{i=1}^{n-2}3i=\frac{3}{2}(n-2)(n-1)=\frac{3}{2}n^2-\frac{9}{2}n+3$$

$\mathrm{SNPR}^0$ neighbours. All together, $N_r$ has $\frac{3}{2}n^2-\frac{7}{2}n+2$ SNPR neighbours.

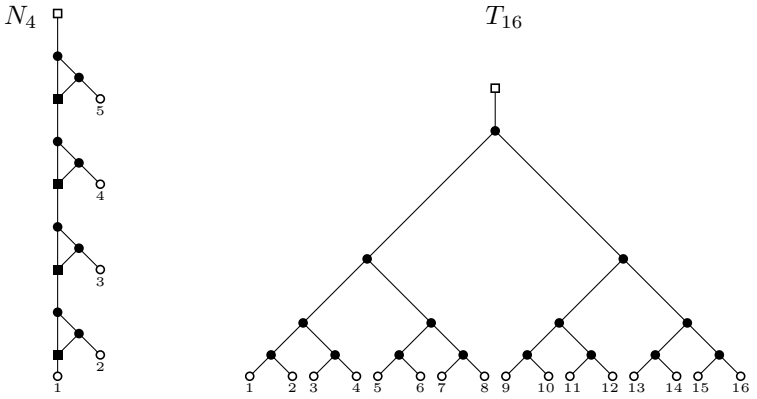


Figure 4.11: Tree-child networks with small and big neighbourhoods: The tree-child network $N_4$ has $\frac{3}{2}5^2-\frac{7}{2}5+2=22$ SNPR neighbours. The balanced tree $T_{16}$ has at least $8\cdot 16^2-4\cdot 16\log_2 16-18\cdot 16+14=1518$ SNPR neighbours.

For a lower bound of the maximum SNPR tree-child neighbourhood size of a tree-child network, we consider the balanced tree $T_n$ on $n$ leaves, as illustrated by $T_{16}$ in Figure 4.11. The formula for the SNPR neighbourhood (Theorem 4.20) is then

$$8n^2-24n+20-\sum_{e\in E}\delta(e)-\sum_{e\in E_{PS}}\delta_T(e).$$

For simplicity, we now assume that $n=2^k$ with $k\geq 1$. Then, for the first sum we have

$$\sum_{e\in E}\delta(e)=\sum_{i=1}^{\log_2 n}i2^i=2n\log_2 n-2n+2.$$

The second sum only differs from the first by the fact the root edge $e_\rho$ is not in $E_{PS}$ and thus

$$\sum_{e\in E_{PS}}\delta_T(e)=2n\log_2 n-2n+2-\delta(e_\rho)=2n\log_2 n-4n+4.$$

In total this yields that, for $n=2^k$, $T_n$ has $8n^2-4n\log_2 n-18n+14$ SNPR neighbours. With no restriction on $n$, the tree $T_n$ has at least $8n^2-\mathcal{O}(n\log_2 n)$ SNPR neighbours.

For an upper bound of the maximum SNPR neighbourhood size, we estimate bounds for the various parameters. We thus assume that the parameters $r_1, r_2, r_3, t_3, d_4$, and $t_4$ are zero. Concerning the sums $-\sum_{e\in E_{T^*}}\delta(e)-\sum_{e\in E_R}\delta_T(e)-\sum_{e\in E_{PS}}\delta_T(e)+\sum_{e\in E_{\bar{b}_3}}\delta_T(e)$ of the SNPR neighbourhood formula, we observe that the first and third sum are at best zero, that the second sum is at best $2r$ and that the last sum is at best half of the second. Therefore, the sums account for only $-r$ in our estimate. Assuming that $r=n-1$ and thus maximal, we get that for any tree-child network $N$ the SNPR tree-child neighbourhood has size at most $16n^2-38n+26$. □

### 4.3.2 NNI neighbourhood

We now look at the NNI neighbourhood of a tree-child network. We recall the definition of an $\mathrm{NNI}^0$ operation $\theta = (f, e, g) \in \Theta^{\mathrm{NNI}^0}(N)$. Let the axis $e = (u, v)$. Then by definition, $g$ is incident to $v$ and $f$ is incident to $u$. The operation now acts differently depending on the type of $e$. If $e$ is a tree edge (pure or impure), then $g$ gets pruned and regrafted to $f$. If $e$ is a reticulation edge, there are two different ways. First, $f$ can get pruned and regrafted to $g$. It is then necessary that $f$ is not an ancestor of $g$, since $\theta$ would otherwise create a cycle. Second, if $f = (u, w)$ and $g = (v, x)$, then both $f$ and $g$ can be removed and the edges $(u, x)$ and $(v, w)$ added. We then say that $w$ and $x$ get *swapped*. Note that this cannot be achieved with an $\mathrm{SNPR}^0$.

We need two further variables describing $N$. Let $t_1$ denote the number of transitive edges in $N$. Also let $r_2^*$ denote the number of $r_2$ structure where the reticulation edge is not a transitive edge.

**NNI tree-child neighbourhood.** Let $\theta = (f, e, g) \in \Theta^{\mathrm{NNI}^0}(N)$. The following lemma states when $\theta$ is not tree-child respecting.

**Lemma 4.22.**
*Let $N \in \mathcal{TC}_n$ and $(f, e, g) \in \Theta^{\mathrm{NNI}^0}(N)$.*
*Then $N' = (f, e, g)(N)$ is not a tree-child network if and only if one of the following cases holds:*

*(i) $e$ is the pure tree edge where $\{f, e, g\}$ forms the path of an $r_3$ structure;*

*(ii) $e$ is the impure tree edge of an $r_2$ structure;*

*(iii) $e$ is the reticulation edge af an $r_2$ structure;*

*(iv) $e$ is the reticulation edge of an $r_3$ structure that is incident to the lower reticulation of this structure and $(f, e, g)$ is an $\mathrm{SNPR}^0$.*

*Proof.* Let $e = (u, v)$. Note that $e$ has to be inner edge. First, assume that $e$ is a pure tree edge. Let $f = (u, w)$ and $g = (v, x)$. It follows straightforward that, if $\{f, e, g\}$ forms an $r_3$ structure, then $N'$ is not tree child (Case (i)). See for example Figure 4.12 (a). Otherwise either $f$ or $g$ is a tree edge. Without loss of generality, assume $f$ is a tree edge and $\tilde{v}$ is the vertex subdividing $f$ to regraft $g$. Then, in $N'$, the vertex $u$ has $\tilde{v}$ as tree child, $\tilde{v}$ has $w$ as tree child and every other vertex keeps its former tree child. Hence, $N'$ is tree child. The case where $g$ but not $f$ is a tree edge works analogously.

Next, assume that $e$ is an impure tree edge. Let $f = (w, u)$ and $g = (v, x)$. If $e$ is the impure tree edge of an $r_2$ structure, then $N'$ is not tree child (Case (ii)). See Figure 4.12 (b). Thus both child edges of $e$ are pure tree edges. Let $h = (v, y) \neq g$ be the second child edge of $e$. Then, in $N'$, the vertex $u$ has $y$ as tree child and the new vertex $\tilde{v}$ has $x$ as tree child. Every other vertex keeps its tree child and thus $N'$ is tree child.

Last, assume that $e$ is an impure reticulation edge. Let $f = (u, w)$. By the definition of an $\mathrm{NNI}^0$ operation $f$ cannot be an ancestor of $g$, so $e$ is not a transitive edge. Assume $e$ is in an $r_2$ structure (Case (iii)) or in an $r_3$ structure and incident to its lower reticulation (Case (iv)). This implies that $f$ is a critical edge. This, if $\theta$ prunes $f$ and regrafts to $g$, then, in $N'$, the parent of $u$ has no tree child. See Figure 4.12 (c) and (d). Note that this is an $\mathrm{SNPR}^0$ operation, but the following is not. If however, $g = (v, x)$ and $w$ and $x$ get swapped by $\theta$, then $N'$ is tree child. If neither is the case, $\theta$ is a legal operation and, in $N'$, the parent of $u$ has a tree child, the new vertex $\tilde{u}'$ has $w$ as tree child and is tree child of another vertex, while every other vertex keeps its tree child. Thus, $N'$ is tree child. □

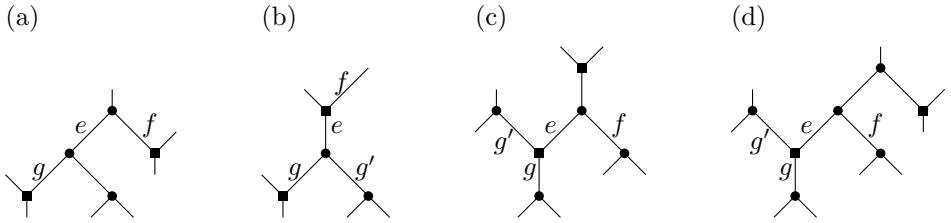


Figure 4.12: Illustration of Lemma 4.22, showing cases where the $\mathrm{NNI}^0$ operation $(f, e, g)$ on a tree-child network may not be tree-child respecting.

Next, we count tree-child respecting $\mathrm{NNI}^0$ operations on $N$.

**Lemma 4.23.**
*Let $N \in \mathcal{TC}_n$ with $n \geq 2$. Then*

$$|\Theta^{\mathrm{NNI}^0}_{\mathcal{TC}}(N)| = 2n + 10r - 2r_1 - 4r_2 - 2r_2^* - 3r_3 - 3t_1 + t_3 - 4.$$

*Proof.* Let $\theta = (f, e, g) \in \Theta^{\mathrm{NNI}^0}_{\mathcal{TC}}(N)$. Let $e = (u, v)$. We count the operations based on the type of the axis $e$. First, assume $e$ is a pure inner tree edge, then it has two outgoing edges $g = (v, x)$ and $h = (v, y)$ that can be regrafted to the edge $f = (u, w) \neq e$. There are $n + r_1 - 2$ pure inner tree edges. By Lemma 4.22, the edges $\{f, e, g\}$ may not form an $r_3$ structure. Hence there are $2(n + r_1 - 2) - r_3$ such $\mathrm{NNI}^0$ operation.

Second, assume $e$ is an impure inner tree edge. There are $r - r_1$ many such edges in $N$. Unless $e$ is part of an $r_2$ structure, both its incident tree edges are pure and can thus be pruned and regrafted to both of $e$'s incident reticulation edges. Hence, there are $4(r - r_1 - r_2)$ such $\mathrm{NNI}^0$ operations.

Third, assume $e$ is one of the $2r$ reticulation edges of $N$. Then there can be three different $\mathrm{NNI}^0$ with $e$ as axis. For such an operation to be tree-child respecting, by Lemma 4.22, we have to account for transitive edges, $r_2$ structures, and $r_3$ structures. If $e$ is a transitive edge, then there is no $\mathrm{NNI}^0$ with $e$ as axis. If $e$ is the reticulation edge of an $r_2$ structure, then two of the possible operations do not result in a tree-child network. However, we only count $r_2^*$ structures to not double count bad cases. Also, if $e$ is the lower reticulation edge of an $r_3$ structure, then again two of the possible operations do not result in a tree-child network. However, there is the exception of the $r_3$ structure being a triangle. Then one operation is actually trivial and thus tree-child respecting. Hence, there are $6r - 2r_2^* - 2r_3 - 3t_1 + t_3$ such $\mathrm{NNI}^0$ operations. The lemma follows from adding the formulas for all three types. □

Next, we count trivial $\mathrm{NNI}^0$ operations and redundancy sets. By Observation 4.2 there are no trivial $\mathrm{NNI}^+$ and $\mathrm{NNI}^-$ operations.

**Lemma 4.24.**
*Let $N \in \mathcal{TC}_n$.*
*Then there are $2t_3$ trivial operations and $2d_4 + t_4$ nontrivial redundancy sets of size two in $\Theta^{\mathrm{NNI}^0}_{\mathcal{TC}}(N)$.*

*Proof.* The proof of Lemma 4.14 describes the cases in which an $\mathrm{SNPR}^0$ on $N$ is trivial. Checking for each of these cases whether the $\mathrm{SNPR}^0$ is also an $\mathrm{NNI}^0$, we get that $(f, e, g) \in \Theta^{\mathrm{NNI}^0}_{\mathcal{TC}}(N)$ is trivial if and only if either $g$ is the critical edge of a triangle and $e$ and $f$ are edges of this triangle or $f$ is the critical edge of a triangle and $e$ and $g$ are edges of this triangle. There are thus two trivial $\mathrm{NNI}^0$ operations per triangle. It is easy to see that a tree-child respecting $\mathrm{NNI}^0$ operations that is not an $\mathrm{SNPR}^0$ operations is nontrivial.

The proof of Proposition 4.16 identifies all sources of nontrivial redundancy sets of tree-child respecting $\text{SNPR}^0$ operations on $N$. We can thus check for each of these cases whether there are two or more different tree-child respecting $\text{NNI}^0$ in such a redundancy set. This gives us that every diamond yields two redundancy sets of size two (for a transformation into a $t_4$ trapezoid) and every $t_4$ trapezoid yields one redundancy set of size two (for a transformation into a diamond). It is easy to see that a tree-child respecting $\text{NNI}^0$ operation that is not an $\text{SNPR}^0$ operations is not redundant with another tree-child respecting $\text{NNI}^0$ operation. □

**Theorem 4.25.**
*Let $N \in \mathcal{TC}_n$ with $n \geq 2$.*
*Then the tree-child neighbourhoods of $N$ under* NNI *operations have the following sizes:*

*(i)* $|U^{\text{NNI}^0}_{\mathcal{TC}}(T)| = 2n + 10r - 2r_1 - 4r_2 - 2r_2^* - 3r_3 - 2d_4 - 3t_1 - t_3 - t_4 - 4$,

*(ii)* $|U^{\text{NNI}^-}_{\mathcal{TC}}(T)| = t_3$,

*(iii)* $|U^{\text{NNI}^+}_{\mathcal{TC}}(T)| = 2n - 2r - 2$, *and*

*(iv)* $|U^{\text{NNI}}_{\mathcal{TC}}(T)| = 4n + 8r - 2r_1 - 4r_2 - 2r_2^* - 3r_3 - 2d_4 - 3t_1 - t_4 - 6$.

*Proof.* Concerning $\text{NNI}^0$, (i) follows from Lemma 4.23 and Lemma 4.24. Concerning $\text{NNI}^-$, (ii) follows directly from Lemma 4.3. Concerning $\text{NNI}^+$, Lemma 4.18 tells us that a reticulation can only be added to a pure tree edge with a pure tree sibling edge. Like for trees, each such edge gives then rise to one different $\text{NNI}^+$ neighbour. Thus, (iii) follows from Observation 4.5, which provides us with the number of these edges in $N$. Lastly, (iv) follows from (i) to (iii). □

An easy way to obtain upper bounds for the $\text{NNI}^0$ and the NNI neighbourhood size is setting all variables except for $n$ and $r$ in the respective formulas in Theorem 4.25 to zero.

We close this section with the example shown Figure 4.13. There, a tree-child network $N$ with its parameters is given. Applying the formulas from Theorem 4.25, we get that $N$ has seventeen $\text{NNI}^0$ neighbours, one $\text{NNI}^-$ neighbour, and four $\text{NNI}^+$ neighbours in $\mathcal{TC}_7$. In total, $N$ has thus twenty-two NNI tree-child neighbours.

## 4.4 Normal networks

For normal networks, we only consider the SNPR neighbourhood. In the previous section on tree-child networks we have shown when SNPR operations are tree-child respecting, when they are trivial, and when they are redundant. Since normal networks are also tree-child networks, we will make use of these results to count SNPR neighbours in the space of normal networks $\mathcal{NN}_n$. We have to integrate, of course, that normal networks do not contain transitive edges. Recall that a transitive edge is an edge $(u, v)$ for which a path from $u$ to $v$ exists that is disjoint from $(u, v)$. For example, triangles and trapezoid contain a transitive edge. Therefore, a normal network contains neither. Also note that only reticulation edges can be transitive edges. We make the following observations.

**Lemma 4.26.**
*Let $N \in \mathcal{NN}_n$. Let $v$ be a reticulation of $N$ with incoming edges $e$ and $f$. Let $e'$ and $f'$ be the sibling edges of $e$ and $f$ respectively.*
*The parents of $v$ are tree vertices and neither parent is ancestor of the other. Moreover, $e'$ is not an ancestor of $f$ and $e$ is not a descendant of $f'$.*

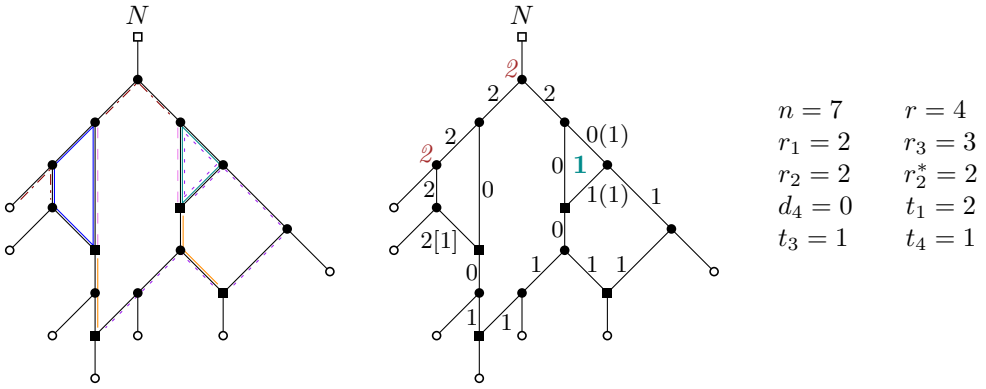


Figure 4.13: A tree-child network and its parameters. On the left, the different structures are highlighted - the triangle with green, the trapezoid with blue, the $r_2$ structures with orange lines, the $r_3$ structures with purple dotted lines, the transitive edges with pink dashed lines, and the sibling pairs of pure tree edges with dark-red dash-dotted lines. In the middle, the black numbers next to inner edges indicate how many $\text{NNI}^0$ operations each edge contributes as axis (trivial ones in round brackets, redundant ones in square brackets), the green, bold 1 indicates the $\text{NNI}^-$ operation, and the dark-red, cursive 2's the $\text{NNI}^+$ operations.

*Proof.* Since $N$ is tree child, the parents $u$ and $u'$ of $v$ are tree vertices. If, say, $u$ was ancestor of $u'$, then $N$ would contain the transitive edge $(u, v)$. The second statement follows directly from the first. □

We extend the notation from Section 4.1 with structures and sets specific to normal networks. Like for tree-child networks there are also critical edges for normal networks. These are present in $r_2$ and $r_3$ structures, but also in the following structure. Let $N \in \mathcal{NN}_n$. A $d_2$ structure of $N$ is a length two path $P$ from a tree vertex $x$ via a tree vertex $u$ to a reticulation $w$ such that there also exists a path $P'$ from $x$ to $w$ that is disjoint from $P$. The critical edge of a $d_2$ structure is the sibling edge $(u, v)$ of the edge $(u, w)$. Figure 4.14 illustrates both of these definitions. Note that pruning the critical edge of a $d_2$ structure creates a transitive edge. We define $E_S$ as the set of pure tree edges in $N$ that are not critical edges of an $r_2$, an $r_3$, or a $d_2$ structure. Also, let $E_{RP}$ contain each edge that is a parent edge of a reticulation edge. Similarly, let $E_{TP}$ contain each edge that is a parent edge of a pair of pure tree edges

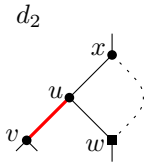


Figure 4.14: A $d_2$ structure in a normal network with the critical edge highlighted in red.

We need another function besides $\alpha$ and $\delta$ to describe the SNPR neighbourhood precisely. Let $\gamma_R \colon E \to \mathbb{N}$ be the function that maps from an edge to the number of reticulation edges that are not descendant edges of $e$ but partner reticulation edges of descendant reticulation edges of $e$.

### 4.4.1 SNPR neighbourhood

We first consider $\mathrm{SNPR}^0$, then $\mathrm{SNPR}^+$ and $\mathrm{SNPR}^-$. With the first lemma we count the number of normal respecting $\mathrm{SNPR}^0$ on $N$.

**Lemma 4.27.**
*Let $N \in \mathcal{NN}_n$. The number of normal respecting $\mathrm{SNPR}^0$ operations on $N$ is*

$$\begin{aligned}|\Theta_{\mathcal{NN}}^{\mathrm{SNPR}^0}(N)| &= 4n^2 + 10nr - 2n(r_2 + r_3 + d_2) - 6n + 2r^2 - 3r(r_2 + r_3 + d_2) - 7r \\ &\quad + 4r_2 + 5r_3 + 4d_2 + 2 - \sum_{e \in E_S}(\delta(e) + \gamma_R(e)) - \sum_{e \in E_R}\alpha_T(e) - \sum_{e \in E_{RP}}\delta_T(e).\end{aligned}$$

*Proof.* Let $(e, f) \in \Theta_{\mathcal{NN}}^{\mathrm{SNPR}^0}(N)$. We distinguish the following cases by the type of $e = (u, v)$. Assume $e$ is a reticulation edge. Since $N$ is normal and thus contains no pure reticulation edge, we know that $e$ is prunable. There are thus $2r$ choices for $e$. Let $e' = (u, v')$ be the partner reticulation edge of $e$. By Lemma 4.12 we know that $f$ cannot be a reticulation edge. This already limits the choices of $f$ to $(m-2r)$ edges. Furthermore, to not create a transitive edge, we know by Lemma 4.26 that in the resulting network neither of the parents of $u$ is an ancestor of the other. This implies that $f$ cannot be an ancestral edge of $e'$, since otherwise $e$ would become a transitive edge. Moreover, $f$ can also not be a descendant edge of $v'$, since otherwise $e'$ in $N'$ would be a transitive edge. This is equivalent to $f$ not being a descendant edge of the parent edge of $e'$. Note that this includes the requirement that $f$ cannot be a descendant of $e$. Combining the choices for $e$ and $f$, the number of $\mathrm{SNPR}^0$ operations that prune a reticulation edge is

$$\begin{aligned}&2r(m - 2r) - \sum_{e \in E_R}\alpha_T(e) - \sum_{e \in E_{RP}}\delta_T(e) \\ &= 4nr + 2r^2 - 2r - \sum_{e \in E_R}\alpha_T(e) - \sum_{e \in E_{RP}}\delta_T(e).\end{aligned} \tag{4.14}$$

Recall from Observation 4.5 that $m = 2n + 3r - 1$. We use $\alpha_T$ and $\delta_T$ to not double count reticulation edges.

Next, assume $e \in E_S$; that is, $e$ is a non-critical tree edge. Since the number of pure tree edges of $N$ is $m - 3r$, the number of choices for $e$ is $|E_S| = m - 3r - r_2 - r_3 - d_2$. By our definition of non-critical tree edges for normal networks, we know that we can prune $e$ without creating a transitive edge. However, regrafting $e$ to a reticulation edge $f = (x, y)$ where $y$ but not $f$ is a descendant of $e$ would create a transitive edge. Note that we defined $\gamma_R$ to count exactly such edges $(x, y)$ with respect to $e$. In addition, $f$ cannot be $e$ or a descendant of $e$. Combining the choices for $e$ and $f$, the number of $\mathrm{SNPR}^0$ operations that prune a non-critical tree edge is

$$\begin{aligned}&(m - 3r - r_2 - r_3 - d_2)(m - 1) - \sum_{e \in E_S}(\delta(e) + \gamma_R(e)) \\ &= (2n - 1 - r_2 - r_3 - d_2)(2n + 3r - 2) - \sum_{e \in E_S}(\delta(e) + \gamma_R(e)) \\ &= 4n^2 + 6nr - 2n(r_2 + r_3 + d_2 + 3) - 3r(r_2 + r_3 + d_2 + 1) \\ &\qquad +2(r_2 + r_3 + d_2 + 1) - \sum_{e \in E_S}(\delta(e) + \gamma_R(e)).\end{aligned} \tag{4.15}$$

A critical edge of an $r_2$ and a $d_2$ structure of $N$ can only be regrafted trivially. A critical edge of an $r_3$ structure can be regrafted to only one edge nontrivially. Therefore, the

number of $\mathrm{SNPR}^0$ operations that prune a critical edge is

$$2r_2 + 3r_3 + 2d_2. \tag{4.16}$$

Adding Equations (4.14) to (4.16) together completes the proof. □

The following lemma follows from Lemma 4.14 by disregarding triangles.

**Lemma 4.28.**
*Let $N \in \mathcal{NN}_n, n \geq 2$. There are $4n + 4r - 4$ trivial operations in $\Theta_{\mathcal{NN}}^{\mathrm{SNPR}^0}(N)$.*

Next, we count the operations we disregard due to redundancies.

**Lemma 4.29.**
*Let $N \in \mathcal{NN}_n, n \geq 2$.*
*The number of nontrivial redundant $\mathrm{SNPR}^0$ of $\Theta_{\mathcal{NN}}^{\mathrm{SNPR}^0}(N)$ minus the number of redundancy sets of nontrivial $\mathrm{SNPR}^0$ of $\Theta_{\mathcal{NN}}^{\mathrm{SNPR}^0}(N)$ is*

$$4n + 4r_1 - 2r_3 - 4d_2 - 8.$$

*Proof.* By Proposition 4.16 we know that redundancies of $\mathrm{SNPR}^0$ are tied to $\mathrm{NNI}^0$ on pure tree edges, and to triangles, diamonds and trapezoids. A normal network contains neither triangles nor trapezoids. The redundant operations for diamonds are not normal respecting. Therefore, we only have to count the number of $\mathrm{NNI}^0$ on pure tree edges that are normal respecting. The number of pure inner tree edges in $N$ is $(n + r_1 - 2)$. However, the pure tree edge of an $r_3$ structure only allows one $\mathrm{NNI}^0$; the pure tree edge of a $d_2$ structure allows none. Hence, there are $2(n + r_1 - 2) - r_3 - 2d_2$ $\mathrm{NNI}^0$ that each induce a redundancy set of size three by Lemma 4.8. Counting two operations per redundancy set, the statement follows. □

We subtract from the number of normal respecting $\mathrm{SNPR}^0$ (Lemma 4.27) the number of trivial $\mathrm{SNPR}^0$ (Lemma 4.28) and the number of operations we can discard per redundancy set (Lemma 4.29). This gives us the size of the $\mathrm{SNPR}^0$ normal neighbourhood of $N$.

**Proposition 4.30.**
*Let $N \in \mathcal{NN}_n, n \geq 2$.*
*Then the $\mathrm{SNPR}^0$ normal neighbourhood $U_{\mathcal{NN}}^{\mathrm{SNPR}^0}$ of $N$ has size*

$$\begin{aligned}|U_{\mathcal{NN}}^{\mathrm{SNPR}}(N)| = {} & 4n^2 + 10nr - 2n(r_2 + r_3 + d_3) - 14n + 2r^2 - 3r(r_2 + r_3 + d_e) - 11r \\ & - 4r_1 + 4r_2 + 7r_3 + 8d_2 + 14 \\ & - \sum_{e \in E_S} (\delta(e) + \gamma_R(e)) - \sum_{e \in E_R} \alpha_T(e) - \sum_{e \in E_{RP}} \delta_T(e).\end{aligned}$$

Next, we look at the $\mathrm{SNPR}^+$ neighbourhood and the $\mathrm{SNPR}^-$ neighbourhood of $N$. Recall that $E_{TP}$ is the set of edges that are parent edge of a pair of pure tree edges.

**Lemma 4.31.**
*Let $N \in \mathcal{NN}_n$ with $n \geq 2$.*
*Then the $\mathrm{SNPR}^+$ normal neighbourhood $U_{\mathcal{NN}}^{\mathrm{SNPR}^+}$ of $N$ has size*

$$|U_{\mathcal{NN}}^{\mathrm{SNPR}^+}(N)| = 4n^2 - 2nr - 6n - 2r^2 + 2 - \sum_{e \in E_{PS}} \alpha_T(e) - 2 \sum_{e \in E_{TP}} \delta_T(e),$$

*and the $\mathrm{SNPR}^-$ normal neighbourhood $U_{\mathcal{NN}}^{\mathrm{SNPR}^-}$ of $N$ has size*

$$|U_{\mathcal{NN}}^{\mathrm{SNPR}^-}(N)| = 2r.$$

*Proof.* We start with the SNPR$^+$ neighbourhood. Let $(e, f) \in \Theta_{\mathcal{NN}}^{\text{SNPR}^+}(N)$. Let $(e, f)(N) = N'$ with the newly added edge $(u', v')$. We know from tree-child networks (see Lemma 4.18) that $e$ has to be a pure tree edge with a sibling pure tree edge; that is, $e \in E_{PS}$. By Observation 4.5 there are thus $2n - 2r - 2$ choices for $e$. On the other hand, we know that $f$ has to be a tree edge, that $f$ is not $e$, and that $f$ cannot be a descendant of $e$. These requirements ensure that $N'$ is a tree-child network. In addition, to ensure that $N'$ is normal, the operation $(e, f)$ cannot add a transitive edge. By Lemma 4.26, this is the case if the new reticulation $v'$ is a child of two tree vertices where neither is ancestor of the other. This implies that $f$ cannot be an ancestral edge of $e$. Furthermore, $f$ can also not be the sibling edge of $e$ or a descendant of it, since otherwise the partner reticulation edge of $(u', v')$ in $N'$ would be a transitive edge. In other words, $f$ cannot be a descendant of the parent edge of $e$. There are $(2n + r - 1)$ choices for $f$ to be a tree edge from which we subtract the number of ancestral tree edges of $e$ and the number of descendant tree edges of the parent edge of $e$. Combining the choices for $e$ and $f$, we get that the number of normal respecting SNPR$^+$ on $N$ is

$$\begin{aligned}&(2n - 2r - 2)(2n + r - 1) - \sum_{e \in E_{PS}} \alpha_T(e) - 2 \sum_{e \in E_{TP}} \delta_T(e)\\ = {} & 4n^2 - 2nr - 6n - 2r^2 + 2 - \sum_{e \in E_{PS}} \alpha_T(e) - 2 \sum_{e \in E_{TP}} \delta_T(e).\end{aligned}$$

Since by Proposition 4.19 two SNPR$^+$ operations on a tree-child network are only redundant if they create a triangle, we know that there are no redundancies for SNPR$^+$ operations on normal networks. The number of operations above equals thus the number of SNPR$^+$ normal neighbours of $N$.

Applying an SNPR$^-$ to a normal network yields again a normal network (see Theorem 3.20). Concerning redundancies, we know by Proposition 4.19 that two SNPR$^-$ on a tree-child network are only redundant if they remove the two reticulation edges of a triangle. Since normal networks do not contain triangles, we know that each SNPR$^-$ on one of the $2r$ reticulation edges of $N$ gives a different SNPR$^-$ normal neighbour of $N$. □

We can now add the SNPR$^0$, the SNPR$^+$, and the SNPR$^-$ normal neighbourhood sizes from Proposition 4.30 and Lemma 4.31 together, which gives us the SNPR normal neighbourhood size of a normal network. Recall that $E_S$ is the set of pure tree edges that are not critical edges, that $E_R$ is the set of reticulation edges, that $E_{RP}$ is the set of parent edges of reticulation edges, that $E_{PS}$ is the set of pure tree edges with a sibling pure tree edge, and that $E_{TP}$ is the set of edges with two tree edges as child.

**Theorem 4.32.**
*Let $N \in \mathcal{NN}_n$ with $n \geq 2$.*
*Then the* SNPR *normal neighbourhood $U_{\mathcal{NN}}^{\text{SNPR}}$ of $N$ has size*

$$\begin{aligned}|U_{\mathcal{NN}}^{\text{SNPR}}(N)| = {} & 8n^2 + 8nr - 2n(r_2 + r_3 + d_2) - 20n - 3r(r_2 + r_3 + d_2) - 9r\\ & - 4r_1 + 4r_2 + 7r_3 + 8d_2 + 16\\ & - \sum_{e \in E_S} (\delta(e) + \gamma_R(e)) - \sum_{e \in E_R} \alpha_T(e) - \sum_{e \in E_{RP}} \delta_T(e) - \sum_{e \in E_{PS}} \alpha_T(e) - 2 \sum_{e \in E_{TP}} \delta_T(e).\end{aligned}$$

## 4.5 Other network classes

We now look at the neighbourhood problem for other classes of phylogenetic networks. In particular, we consider whether the method from the previous sections can also be used

to obtain exact formulas for neighbourhood sizes of other classes. For this, we recall two properties of tree-child networks that were particular helpful to find such formulas. First, the tree-child property is a local property that can be easily checked. For example, with only a few structures and edge types it was easy to describe which edges can be pruned and to which edges can be regrafted such that operations are tree-child respecting. Second, every vertex and every edge in a tree-child network is uniquely identifiable. This was one of our main tools when counting trivial operations and redundancy sets. However, the number of parameters needed in the formulas for tree-child or normal neighbourhoods is still large.

**Level-1 networks.** Huber et al. [HLMW16] solved the NNI neighbourhood problem of unrooted level-1 networks without parallel edges. For an unrooted level-1 network $N$ with $b_3$ blobs of size three, $b_4$ blobs of size four, $b$ blobs in total, $l$ links (edges incident to two blobs), and $t$ inner vertices not contained in a cycle, they showed that the NNI level-1 neighbourhood has a size of

$$2n - 6 + 6b - 5b_3 - 2b_4 - 2l + t.$$

Gambette et al. [GvIJ+17] noted that rooted level-1 networks "do not permit a simple formula". However, we can give upper bounds on the sizes of NNI neighbourhoods of a rooted level-1 network using links.

Let $N$ be a rooted level-1 network (with possibly parallel edges); i.e. $N \in \mathcal{LV}_{1,n}$. A *link* of $N$ is an edge that is incident to two different blobs of $N$. Let $l_p$ (resp. $l_i$) denote the number of links of $N$ that are pure (resp. impure) tree edges. Note that an $\mathrm{NNI}^0$ operations with a link as axis is not level-1 respecting. We use this observation to give the following bounds for the NNI neighbourhoods of $N$. Recall that $t_1$ denotes the number of transitive edges in $N$.

**Lemma 4.33.**
*Let $N \in \mathcal{LV}_{1,n}$ with $n \geq 2$.*
*Then the neighbourhoods of $N$ under* NNI *operations have sizes*

*(i)* $|U^{\mathrm{NNI}^0}_{\mathcal{LV}_1}(N)| \leq 2n + 10r - 2r_1 - 4l_i - 2l_p - 3t_1 - 4$,

*(ii)* $|U^{\mathrm{NNI}^-}_{\mathcal{LV}_1}(N)| = t_3$,

*(iii)* $|U^{\mathrm{NNI}^+}_{\mathcal{LV}_1}(N)| \leq 2n - 2$, *and*

*(iv)* $|U^{\mathrm{NNI}}_{\mathcal{LV}_1}(N)| \leq 4n + 10r - 4r_1 - 4l_i - 2l_p - 3t_1 - 6$.

*Proof.* Concerning the $\mathrm{NNI}^0$ neighbourhood of $N$, we count possible operations based on the type of the axis $e$ of an operation. If $e$ is a pure inner tree edge and not a link, then there are two $\mathrm{NNI}^0$ on $e$. There are $n + r_1 - l_p - 2$ such edges in $N$. If $e$ is an impure inner tree edge and not a link, then there are four $\mathrm{NNI}^0$ on $e$. There are $r - r_1 - l_i$ such edges in $N$. If $e$ is a reticulation edge and not a transitive edge, then there are three $\mathrm{NNI}^0$ on $e$. There are $2r - t_1$ such edges in $N$. The bound now follows from adding the numbers for the three possible types of $e$ together.

The result on the $\mathrm{NNI}^-$ neighbourhood of $N$ follows from Corollary 4.4. The upper bound on the $\mathrm{NNI}^+$ neighbourhood of $N$ follows from the number of pure tree edges with a pure tree sibling edge in $N$. It is only an upper bound, since if at least one of those two edges is in a blob, then the resulting network is a level-2 network. The upper bound on the NNI neighbourhood follows from (i) to (iii). □

To derive exact formulas it is necessary to identify all trivial operations and redundancy sets. The results form Lemma 4.24 on trivial and redundant NNI operations on tree-child networks could partially be integrated into the bounds in Lemma 4.33. However, for example redundant operations that transform a triangle into a diamond or a trapezoid in tree-child networks may not be level-1 respecting if they involve a link. Furthermore, parallel edges are another source for trivial and redundant operations. So while with some effort exact formulas could be obtained, we restrain from this here to avoid introducing even more parameters.

Next, consider the SNPR neighbourhood of a level-1 network. Note that being a level-1 network is not a local property. Therefore, when regrafting or adding an edge, one cannot simply check in a neighbourhood of a few edges to see if the operation would be level-1 respecting. A formula would need to describe that a pruned edge cannot be regrafted or an edge added such that two blobs merge. Furthermore, parallel edges make it also harder to count redundancies. Hence, level-1 networks miss the two properties we described above that made it comprehensible for us to find exact formulas for the SNPR neighbourhoods of tree-child and normal networks.

**Tree-sibling networks.** The tree-sibling property is like the tree-child property a local and easy to check property. Furthermore, it can be shown that vertices and edges in a tree-sibling network are unique. A rigor analysis of structures determining tree-sibling respecting operations, trivial, and redundant operations would thus yield a formula similar to the one for the tree-child neighbourhood.

**Reticulation-visible and tree-based networks.** Consider the network $N$ in Figure 4.15, which is a reticulation-visible and tree-based network. Note that $N$ contains non-unique vertices and edges, for example the edges $f$ and $g$. This gives us the following observation.

**Observation 4.34.**
*Let $N \in \mathcal{RV}_n$ with $n \geq 2$*
*There can be more than one automorphism on $N$ that fixes the leaf set of $N$.*

For $N$ in Figure 4.15, consider two SNPR$^0$ that prune the leaf 3 and where one regrafts to $f$ and the other to $g$. Since the edges are indistinguishable, the resulting networks from these operation would be the same. However, if we consider an SNPR$^0$ operation that prunes the edge $e$, then regrafting to $f$ makes the operation trivial, but regrafting to $g$ does not. Therefore, edges that are in general indistinguishable in $N$ may not be indistinguishable for all operations.

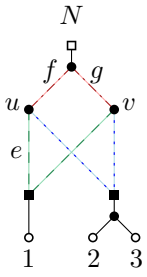


Figure 4.15: A reticulation-visible (and tree-based) network $N$ with non-unique vertices and edges. The vertices $u$ and $v$ are isomorphic and so are the three groups of edges with the same colour and style.

Both these observations add complexity to the problem of counting trivial and redundant

operations. A formula for the neighbourhood size under SNPR would have to capture how groups of indistinguishable edges relate to other edges. This does not seem to be possible with parameters for some fixed structures and by counting the number of ancestors and descendants of edges. Furthermore, like with the level-1 property, the reticulation-visible and tree-based property of a network are not local but global properties. We conclude that reticulation-visible or tree-based networks do not permit a simple formula for the neighbourhood size.

**Phylogenetic networks.** We now consider the space of all phylogenetic networks. Note that every operation is by default respecting the class $\mathcal{N}_n$. The problems due to indistinguishable edges and vertices discussed above still exist, however. We therefore consider only bounds for the neighbourhood sizes under NNI, SNPR, and also PR.

Gambette et al. [GvIJ+17] considered $\mathrm{NNI}^0$ on rooted phylogenetic networks and gave a sharp upper bound on the neighbourhood size that depends on the number of edges of different types (pure and impure tree and reticulation edges). With our notation this bound is

$$|U_{\mathcal{N}}^{\mathrm{NNI}^0}(N)| \leq 2n + 10r - 2r_1 - 4.$$

We can improve this bound slightly by factoring in that transitive edges do not give rise to $\mathrm{NNI}^0$ operations. Furthermore, we give bounds for the other NNI neighbourhoods.

**Proposition 4.35.**
*Let $N \in \mathcal{N}_n$ with $n \geq 2$.*
*Then the neighbourhoods of $N$ under* NNI *operations have sizes*

*(i)* $|U_{\mathcal{N}}^{\mathrm{NNI}^0}(N)| \leq 2n + 10r - 2r_1 - 3t_1 - 4$,

*(ii)* $|U_{\mathcal{N}}^{\mathrm{NNI}^-}(N)| \leq t_3$,

*(iii)* $|U_{\mathcal{N}}^{\mathrm{NNI}^+}(N)| \leq 2n + 4r - 2$, *and*

*(iv)* $|U_{\mathcal{N}}^{\mathrm{NNI}}(N)| \leq 4n + 14r - 2r_1 - 3t_1 + t_3 - 6$.

*Proof.* We discussed above how to derive the bound on the $\mathrm{NNI}^0$ neighbourhood of $N$. The bound on the $\mathrm{NNI}^-$ neighbourhood of $N$ follows from the definition of $\mathrm{NNI}^-$. The bound on the $\mathrm{NNI}^+$ neighbourhood of $N$ follows from twice the number of inner vertices of $N$. The bound on the NNI neighbourhood of $N$ follows from (i) to (iii). □

We give upper bounds on the SNPR neighbourhoods of a phylogenetic network by simply considering all possible operations. Francis et al. [FHMW18] also did this to obtain an upper bound on the neighbourhood size for unrooted phylogenetic networks and their generalisation of SPR.

**Proposition 4.36.**
*Let $N \in \mathcal{N}_n$ with $n \geq 2$.*
*Then the neighbourhoods of $N$ under* SNPR *operations have sizes*

*(i)* $|U_{\mathcal{N}}^{\mathrm{SNPR}^0}(N)| \leq 4n^2 + 10nr - 10n + 6r^2 - 7r + 4 - \sum_{e \in E} \delta(e)$,

*(ii)* $|U_{\mathcal{N}}^{\mathrm{SNPR}^-}(N)| \leq 2r$,

*(iii)* $|U_{\mathcal{N}}^{\mathrm{SNPR}^+}(N)| \leq 4n^2 + 12nr - 4n + 9r^2 - 6r + 1 - \sum_{e \in E} \delta(e)$, *and*

*(iv)* $|U_{\mathcal{N}}^{\mathrm{SNPR}}(N)| \leq 8n^2 + 22nr - 14n + 15r^2 - 13r + 5 - 2\sum_{e \in E} \delta(e)$.

*Proof.* For the upper bound on the $\mathrm{SNPR}^0$ neighbourhood of $N$, note that we can prune any edge $e$ that is not an outgoing edge of a reticulation and then regraft it to an edge $f$ that is not $e$ or a descendant of $e$. We can also ignore the two trivial $\mathrm{SNPR}^0$ that regraft to edges incident to the tail of $e$. There are thus $(m-r)(m-3) - \sum_{e\in E}\delta(e)$ potential neighbours. The given bound then follows from $m = 2n + 3r - 1$ (Observation 4.5).

For the upper bound on the $\mathrm{SNPR}^+$ neighbourhood of $N$, note that for an operation $(e, f) \in \Theta_{\mathcal{N}}^{\mathrm{SNPR}^+}(N)$ the edge $f$ cannot be a descendant of $e$. However, every other combination of $e$ and $f$ gives a potential neighbour. The result thus follows from expanding $m^2 - \sum_{e\in E}\delta(e)$.

For the upper bound on the $\mathrm{SNPR}^-$ neighbourhood of $N$, note that there are $2r$ different reticulation edges that can be removed. The last statement follows from the first three. □

Note that while counting $\mathrm{SNPR}^-$ neighbours was easy for tree-child networks, it is harder in the general case. Assume that a network $N$ contains a chain of pairs of parallel edges. Then removing any reticulation edge from one of these pairs yields the same $\mathrm{SNPR}^-$ neighbour. That means that a network with $r > 0$ reticulation may have only a single $\mathrm{SNPR}^-$ neighbour.

We can also consider the PR neighbourhood of a phylogenetic network. This adds potential neighbours to the bounds in Proposition 4.36 that come from head $\mathrm{PR}^0$ operations.

**Proposition 4.37.**
*Let $N \in \mathcal{N}_n$ with $n \geq 2$.*
*Then the neighbourhoods of $N$ under* PR *operations have sizes*

*(i)* $|U_{\mathcal{N}}^{\mathrm{PR}^0}(N)| \leq 4n^2 + 14nr - 10n + 12r^2 - 15r + 4 - \sum_{e\in E}\ \delta(e) - \sum_{e\in E_R}\ \alpha(e)$,

*(ii)* $|U_{\mathcal{N}}^{\mathrm{PR}^-}(N)| \leq 2r$,

*(iii)* $|U_{\mathcal{N}}^{\mathrm{PR}^+}(N)| \leq 4n^2 + 12nr - 4n + 9r^2 - 6r + 1 - \sum_{e\in E}\ \delta(e)$, *and*

*(iv)* $|U_{\mathcal{N}}^{\mathrm{PR}}(N)| \leq 8n^2 + 26nr - 14n + 21r^2 - 21r + 5 - 2\sum_{e\in E}\ \delta(e) - \sum_{e\in E_R}\ \alpha(e)$.

*Proof.* We add to the formulas from Proposition 4.36 that the $2r$ reticulation edges can be pruned and then regrafted to any edge that is not an ancestor of the pruned edge or the edge itself. We also ignore the two trivial operations that regraft to incident edges of the pruned edge. This adds $2r(2n + 3r - 4) - \sum_{e\in E_R}\alpha(e)$ possible $\mathrm{PR}^0$ neighbours. □

## 4.6 Concluding remarks

In this chapter we have studied the neighbourhood problem under SNPR and NNI. We started with trees and then increased the complexity with tree-child and normal networks. In the previous section, we looked at the problem for level-1 networks and general phylogenetic networks. For trees, tree-child networks, and normal networks we derived exact formulas for the SNPR neighbourhood. Furthermore, we also found formulas for the NNI neighbourhood of a tree and of a tree-child network. Our method to derive these formulas was a three-step counting scheme. In the first step, we counted all possible operations that respect the current class. In the second step, we subtracted all trivial operations, and in the third step, we counted redundancies of operations. This last step becomes the most challenging with increasing complexity.

It is interesting to look back and see how the complexity of the neighbourhood problem changes from trees to networks. For this, we identify three key factors that determine the formulas for the $\mathrm{SNPR}^0$ neighbourhood. These are the size, the sum of descendants of

edges, and the occurrence of certain small subgraphs. If we first look at a tree in $\mathcal{T}_n$, then its size given by $n$ determines how many edges can be pruned and its sum of descendants of edges tells us to how many edges the pruned edges cannot be regrafted. Thirdly, every inner edge gives rise to redundancies of SNPR$^0$ operations, which correspond to the two possible NNI$^0$ on this edge. With the formula for the SNPR tree-child neighbourhood of a network in $\mathcal{TC}_{n,r}$, we can consider how these factors generalise to networks. First, the size is of course now given by $n$ and $r$. Second, we have to count again the sum of descendants but have to take care which edges can actually be pruned and whether the resulting network will be tree-child. We found that the latter is determined by the types of edges and by the occurrence of small structures within $N$, like triangles. Lastly, accounting for redundancies becomes more complex, though interestingly, we found that many redundancies are still tied to NNI. However, there are also sources of redundancies independent of NNI operations. For example, we found that SNPR$^0$ on the two reticulation edges of a triangle yield the same neighbours. In total the occurrence of seven different structures in a tree-child network affect its neighbourhood size. Beyond tree-child networks, if we look at a level-1 network it becomes more challenging to even count level-1 respecting SNPR, since no operation may join two blobs. On the other hand, for a general phylogenetic network, which does not have the property that every vertex and edge is uniquely identifiable, the main challenge is to count redundancies.

The problem of accounting for redundancies becomes even more complex when considering PR instead of SNPR. On top of the redundancies between tail PR$^0$ come redundancies between head and tail PR$^0$. Recall that the proof of classification of sources of SNPR redundancies in a tree-child network (Proposition 4.16) was based on fixing reticulations and cycles. This approach may or may not be extendable to redundancies with PR$^0$. Nevertheless, we gave upper bounds on the PR neighbourhood of a phylogenetic network in Proposition 4.37.

# 5. Shortest paths

We have seen in Chapter 4 how to find and count the neighbours of a network. In other words, we have looked at what networks can be reached with a path of length one. In Chapter 3 we have seen that most classes of phylogenetic networks form metric spaces under SNPR and PR. This means that there exists a path between any two networks in each of these spaces. Among these paths are shortest paths, which are the paths that define the SNPR- or PR-distance of the two networks. These are the paths we study in this chapter. In the following paragraphs, when we mention the PR-distance, the same holds for the SNPR-distance.

It is NP-hard to compute the SNPR-distance [BLS17, Theorem 7.2] and, as we will see in this chapter, also to compute the PR-distance. So far no method has been proposed to compute the PR-distance of two networks $N$ and $N'$ in $\mathcal{N}_n$. One approach could be to start an exhaustive search from $N$ until it reaches $N'$. An improvement would be to start one search from $N$ and one from $N'$ simultaneously and see when they meet. As we have seen, the PR neighbourhood of a network can be in $\Omega(n^2)$ (Theorem 4.20) and the PR-distance of two networks can be in $\Omega(n+r)$. Hence, an exhaustive search may have to consider an exponentially growing number of possible paths. We are therefore interested in what assumptions we can make about shortest paths between $N$ and $N'$ that may either guide the search or restrict the search space. For example, if we know that both $N$ and $N'$ are in tier $r$, does this imply that there is a shortest path from $N$ to $N'$ in $\mathcal{N}_n$ that is fully contained in tier $r$? We will see that this is not the case.

We start with the PR-distance of a tree $T$ and a network $N$. One main relation between $N$ and $T$ is whether $N$ displays $T$. Recall that we then write $T \in D(N)$. Among several other relations, we look at shortest paths between $T$ and $N$ when $T$ is in $D(N)$, when $T$ is in $D(N')$ where we know the PR-distance of $N$ and $N'$, or when we know something about the PR-distance of $T$ to a tree $T' \in D(N)$. The main result will be a characterisation of the PR-distance of $T$ and $N$ in terms of the PR-distance of $T$ to the trees in $D(N)$. This result will allow us to find a fixed-parameter tractable algorithm to compute the PR-distance of $T$ and $N$. We extend several results from Bordewich et al. [BLS17] and Klawitter and Linz [KL19] from SNPR to PR.

We then look at shortest paths between two networks $N$ and $N'$ where we know the number of reticulations of $N$ and $N'$. As mentioned above, we are particularly interested in whether $\mathcal{N}_{n,r}$ is an isometric subgraph of $\mathcal{N}_n$. Restricting this question to particular classes, we also look at whether the distance of two networks in a particular class differs from their distance in the overall space of networks. In other words, we ask whether these

classes are isometric subgraph of $\mathcal{N}_n$. We answer this negatively for the classes $\mathcal{TC}_n$, $\mathcal{NN}_n$, $\mathcal{RV}_n$, $\mathcal{TB}_n$, and $\mathcal{LV}_{k,n}$.

*Remark.* The content of this chapter concerning the SNPR-distance is part of the joint work with Simone Linz called "On the Subnet Prune and Regraft Distance" [KL19].

## 5.1 Tree to network

Let $N \in \mathcal{N}_n$ with $r$ reticulations and let $T \in \mathcal{T}_n$. In this section we look at bounds on the PR-distance of $T$ and $N$. Recall that $T$ is displayed by $N$, written $T \in D(N)$, if it is has an embedding into $N$, meaning there is a subdivision of $T$ that is a subgraph of $N$.

Note that since a $\mathrm{PR}^-$ removes only a single reticulation, it follows that $r$ is a lower bound on the distance of $T$ and $N$. Consider the case where $N$ is tree-based with base tree $T$. Then $T$ has an embedding that covers all vertices of $N$ except for $r$ disjoint reticulation edges. It is easy to see that there is then a $\mathrm{PR}^-$-sequence of length $r$ from $N$ to $T$ that removes these $r$ reticulation edges one at a time. Bordewich et al. [BLS17] have shown that there is also such a path if $T$ is only displayed by $N$. Hence, we have the following results.

**Lemma 5.1** ([BLS17, Lemma 7.4])**.**
*Let $N \in \mathcal{N}_n$ with $r$ reticulations. Let $T \in D(N)$. Then $\mathrm{d}_{\mathrm{SNPR}}(T, N) = r$.*

**Corollary 5.2.**
*Let $N \in \mathcal{N}_n$ with $r$ reticulations. Let $T \in D(N)$. Then $\mathrm{d}_{\mathrm{PR}}(T, N) = r$.*

Next, consider the case where $N$ is a tree, that is, $N = T' \in \mathcal{T}_n$. The main question is then whether $\mathrm{d}_{\mathrm{PR}}(T, T')$ is the same in $\mathcal{T}_n$ and in $\mathcal{N}_n$ or, in other words, if $\mathcal{T}_n$ is an isometric subgraph of $\mathcal{N}_n$ under PR. Bordewich et al. [BLS17] have shown that this is the case for SNPR. We rephrase their proof to show that it also holds for PR.

**Theorem 5.3** ([BLS17, Proposition 7.1])**.**
*The class of phylogenetic trees $\mathcal{T}_n$ is an isometric subgraph of the class of phylogenetic networks $\mathcal{N}_n$ under* SNPR*. Moreover, for every $T, T' \in \mathcal{T}_n$, every shortest path from $T$ to $T'$ in $\mathcal{N}_n^{\mathrm{SNPR}}$ is fully contained in $\mathcal{T}_n$.*

**Theorem 5.4.**
*The class of phylogenetic trees $\mathcal{T}_n$ is an isometric subgraph of the class of phylogenetic networks $\mathcal{N}_n$ under* PR*. Moreover, for every $T, T' \in \mathcal{T}_n$, every shortest path from $T$ to $T'$ in $\mathcal{N}_n^{\mathrm{PR}}$ is fully contained in $\mathcal{T}_n$.*

*Proof.* Let $\mathrm{d}_{\mathcal{T}}$ and $\mathrm{d}_{\mathcal{N}}$ denote the PR-distance in $\mathcal{T}_n$ and $\mathcal{N}_n$, respectively. To prove the statement, it suffices to show that $\mathrm{d}_{\mathcal{T}}(T, T') = \mathrm{d}_{\mathcal{N}}(T, T')$ for every pair $T, T' \in \mathcal{T}_n$. Note that $\mathrm{d}_{\mathcal{T}}(T, T') \geq \mathrm{d}_{\mathcal{N}}(T, T')$ holds by definition. To prove the converse, let $\sigma = (T = N_0, N_1, \ldots, N_k = T')$ be a shortest PR-sequence from $T$ to $T'$. Consider the following 2-colouring of the edges of each $N_i$, for $i \in \{0, \ldots, k\}$. Colour all edges of $N_0$ blue. For $i \in \{1, \ldots, k\}$ preserve the colouring of $N_{i-1}$ to a colouring of $N_i$ for all edges except those affected by the PR. In particular, an edge that gets added or moved is coloured red, an edge resulting from a vertex suppression is coloured blue if the two merged edges were blue, and red otherwise, and edges resulting from a subdivision are coloured like the subdivided edge.

Let $F_i$ be the graph obtained from $N_i$ by removing all red edges. We claim that $F_i$ is a forest with at most $k + 1$ components. Since $F_0 = T$, the statement holds for $i = 0$. If $N_i$ is obtained from $N_{i-1}$ by a $\mathrm{PR}^+$, then $F_i = F_{i-1}$ since no blue component gets split.

If $N_i$ is obtained from $N_{i-1}$ by a $\mathrm{PR}^0$, then clearly at most one component gets split no matter whether an edge gets pruned at its tail or its head. The same is the case when $N_i$ is obtained by a $\mathrm{PR}^-$. Note that $F_k$ is conceptually equivalent to an agreement forest (which gets defined precisely in the next chapter) for $T$ and $T'$ and thus $\mathrm{d}_{\mathcal{T}}(T,T') \leq k = \mathrm{d}_{\mathcal{N}}(T,T')$ by Theorem 2.1 of Bordewich and Semple [BS05]. Furthermore, if $\sigma$ would use a $\mathrm{PR}^+$, then the forest $F_k$ would contain at most $k$ components. However, then $\mathrm{d}_{\mathcal{T}}(T,T') < k$; a contradiction. □

Note that the previous two theorems imply that the classic SPR-distance of two trees in $\mathcal{T}_n$ equals their SNPR- and PR-distance in $\mathcal{N}_n$. Therefore and since computing the SPR-distance of two trees is NP-hard [BS05], the following corollary is an immediate consequence of Theorem 5.4.

**Corollary 5.5.**
*Computing the* PR*-distance of an arbitrary pair of networks in $\mathcal{N}_n$ is NP-hard.*

With the next lemmata we show that if two networks $N$ and $N'$ have a certain distance $k$, then they display trees that also have at most distance $k$. This and the previous results can be paraphrased to say that moving in higher tiers of $\mathcal{N}_n$ is not faster than in $\mathcal{T}_n$ in terms of displayed trees.

**Lemma 5.6** ([BLS17, Proposition 7.7])**.**
*Let $N, N' \in \mathcal{N}_n$ such that $\mathrm{d}_{\mathrm{SNPR}}(N,N') = k$. Let $T \in D(N)$.*
*Then there exists a phylogenetic tree $T' \in D(N)$ such that $\mathrm{d}_{\mathrm{SNPR}}(T,T') \leq k$.*

**Lemma 5.7.**
*Let $N, N' \in \mathcal{N}_n$ such that $\mathrm{d}_{\mathrm{PR}}(N,N') = k$. Let $T \in D(N)$.*
*Then there exists a phylogenetic tree $T' \in D(N)$ such that $\mathrm{d}_{\mathrm{PR}}(T,T') \leq k$.*

*Proof.* We extend the proof of Lemma 5.6 from SNPR to PR. The proof is by induction on $k$. If $k = 0$, then the statement trivially holds. Suppose that $k = 1$. If $T \in D(N')$, then set $T' = T$, and we have $\mathrm{d}_{\mathrm{PR}}(T,T') = 0 \leq 1$. So assume otherwise, namely that $T \notin D(N')$. Note that if $N'$ has been obtained from $N$ by a $\mathrm{PR}^+$, then $N'$ displays $T$. Therefore we distinguish whether $N'$ has been obtained from $N$ by a $\mathrm{PR}^0$ or $\mathrm{PR}^-$.

Suppose that $N'$ has been obtained from $N$ by a $\mathrm{PR}^0$. If this is a tail $\mathrm{PR}^0$, then Bordewich et al. [BLS17] have shown that there is a tree $T' \in D(N')$ with $\mathrm{d}_{\mathrm{PR}}(T,T') = 1$. So assume otherwise, namely that a head $\mathrm{PR}^0$ prunes the reticulation edge $e = (u,v)$ at $v$ of $N$. Let $e' = (u',v)$ be the partner edge of $e$. Fix a subdivision $S$ of $T$ that is a subgraph of $N$. Since $N'$ does not display $T$, the edge $e$ is contained in $S$. The edge $e'$, on the other hand, is not in $S$, since $S$ is a tree. Let $\bar{e}$ be the edge of $T$ that is subdivided into the path $P$ of $S$ that contains $e$. Let $w$ be the start vertex of $P$. Then let $P_1$ be the subpath of $P$ in $S$ from $w$ to $v$. Also let $P_2$ be a path in $N$ that starts at a vertex $w'$ of $S$, that contains no edge of $S$, but that contains $e'$, and that ends at $v$. Note that such $P_2$ has to exist since there are at least two paths in $N$ from the root to $v$. See also Figure 5.1 for an illustration. Note that $S$ without $P_1$ but with $P_2$ is a tree in $N$. Moreover, $P_2$ and $P \setminus P_1$ also exist in $N'$. We can thus obtain $S'$ from $S$ by removing $P_1$, adding $P_2$, suppressing $v$, and subdividing the edge to which $e$ gets regrafted. Note that $S'$ is a tree in $N'$. See again Figure 5.1. Furthermore, we can prune the edge $\bar{e}$ in $T$ and regraft it to obtain a tree $T'$ such that $S'$ provides an embedding of $T'$ into $N$ and $N'$. More precisely, we regraft the pruned $\bar{e}$ to the edge of $N$ that is mapped to the path that contains $w'$. Hence, we have that $T' \in D(N')$ and $\mathrm{d}_{\mathrm{PR}}(T,T') = 1$. The case where $N'$ has been obtained from $N$ by a $\mathrm{PR}^-$ works analogously.

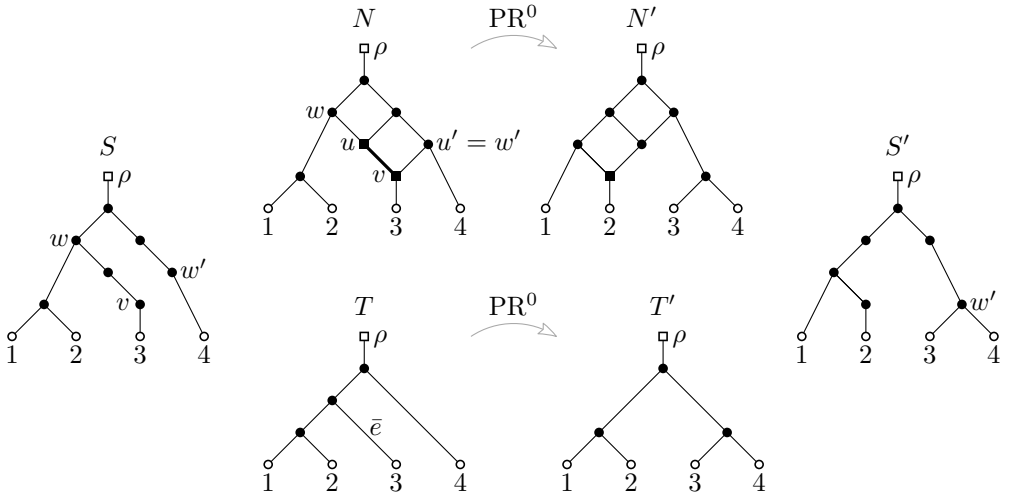


Figure 5.1: An example for the proof of Lemma 5.7. When a head $\mathrm{PR}^0$ prunes the edge $e = (u, v)$ in $N$ to obtain $N'$, then in a tree $T \in D(N)$ we can use a $\mathrm{PR}^0$ that prunes an edge $\bar{e}$ to obtain a tree $T'$ in $D(N')$. For the embeddings $S$ and $S'$ of $T$ and $T'$ into $N$ and $N'$, respectively, this means that the path from $w$ to leaf 3 is replaced with the path from $w'$ to leaf 3.

Now suppose that $k \geq 2$ and the hypothesis holds for any two networks with PR-distance at most $k-1$. Let $N'' \in \mathcal{N}_n$ such that $\mathrm{d_{PR}}(N, N'') = k-1$ and $\mathrm{d_{PR}}(N'', N') = 1$. Thus by induction there are trees $T''$ and $T'$ such that $T'' \in D(N'')$ with $\mathrm{d_{PR}}(T, T'') \leq k-1$ and $T' \in D(N')$ with $\mathrm{d_{PR}}(T'', T') \leq 1$. It follows that $\mathrm{d_{PR}}(T, T') \leq k$, thereby completing the proof of the lemma. □

With Corollary 5.2 and Lemma 5.7 (resp. Lemma 5.1 and Lemma 5.6) we can get further bounds on the PR-distance (resp. SNPR-distance) of trees.

**Lemma 5.8.**
*Let $T \in \mathcal{T}_n$ and $N \in \mathcal{N}_n$ with $\mathrm{d_{PR}}(T, N) = k$ (resp. $\mathrm{d_{SNPR}}(T, N) = k'$).*
*Then $\mathrm{d_{PR}}(T, T') \leq k$ (resp. $\mathrm{d_{SNPR}}(T, T') \leq k'$) for each $T' \in D(N)$.*

*Proof.* Since $\mathrm{d_{PR}}(N, T) = k$ and by Lemma 5.7, for every tree $T' \in D(N)$ there exists a tree $T'' \in D(T)$ with $\mathrm{d_{PR}}(T', T'') \leq k$. As $T$ is the only tree displayed by $T$, it follows that $T'' = T$ and thus $\mathrm{d_{PR}}(T', T) \leq k$. □

**Lemma 5.9.**
*Let $N \in \mathcal{N}_n$ with $r$ reticulations. Let $T, T' \in D(N)$. Then $\mathrm{d_{PR}}(T, T') = \mathrm{d_{SNPR}}(T, T') \leq r$.*

*Proof.* Note that $\mathrm{d_{PR}}(T, N) = \mathrm{d_{SNPR}}(T, N) = r$ by Corollary 5.2. With Lemma 5.8 we then get that every tree $T'' \in D(N)$ has distance at most $r$ to $T$. This also holds for $T'$ and thus $\mathrm{d_{PR}}(T, T') \leq r$. □

Note that the situation of Lemma 5.9 is related to the Hybridisation Number problem, where we are given two trees $T$ and $T'$ and look for the minimum number of reticulations $r$ needed such that there is a network $N$ that displays both $T$ and $T'$ and that has only $r$ reticulations. As has already been known [BGMS05] and as follows from Lemma 5.9, we get that if $\mathrm{d_{PR}}(T, T') = r$, then $N$ needs at least $r$ reticulations.

The main result of this section is the following theorem that characterises the distance between a phylogenetic tree and a phylogenetic network.

**Theorem 5.10.**
*Let $T \in \mathcal{T}_n$. Let $N \in \mathcal{N}_n$ with $r$ reticulations. Then*

$$\mathrm{d_{PR}}(T, N) = \mathrm{d_{SNPR}}(T, N) = \min_{T' \in D(N)} \mathrm{d_{PR}}(T, T') + r.$$

*Proof.* Let $T^* \in D(N)$ such that $\mathrm{d_{PR}}(T, T^*) \leq \mathrm{d_{PR}}(T, T')$ for each $T' \in D(N)$. Then, by Corollary 5.2 and Theorem 5.4, it follows that

$$\mathrm{d_{PR}}(T, N) \leq \mathrm{d_{PR}}(T, T^*) + \mathrm{d_{PR}}(T^*, N) = \min_{T' \in D(N)} \mathrm{d_{PR}}(T, T') + r. \tag{5.1}$$

We next show that

$$\mathrm{d_{PR}}(T, N) \geq \min_{T' \in D(N)} \mathrm{d_{PR}}(T, T') + r.$$

Suppose that $\mathrm{d_{PR}}(T, N) = k$. Let $\sigma = (T = N_0, N_1, N_2, \ldots, N_k = N)$ be a PR-sequence from $T$ to $N$. For each $i \in \{1, 2, \ldots, k\}$, consider the two networks $N_{i-1}$ and $N_i$ in $\sigma$. If $N_i$ has been obtained from $N_{i-1}$ by applying a $\mathrm{PR}^+$, then $D(N_{i-1}) \subseteq D(N_i)$. Furthermore, regardless of the PR used to obtain $N_i$ from $N_{i-1}$, Lemma 5.6 implies that, for each tree $T_{i-1} \in D(N_{i-1})$, there exists a tree $T_i$ in $D(N_i)$ such that $\mathrm{d_{PR}}(T_{i-1}, T_i) \leq 1$. It is now straightforward to check that we can construct a sequence $S = (T_0, T_1, T_2, \ldots, T_k)$ of phylogenetic trees on $\mathcal{X}$ from $\sigma$ that satisfies the following properties.

(i) For each $i \in \{0, 1, \ldots, k\}$, we have $T_i \in D(N_i)$.

(ii) For each $i \in \{1, 2, \ldots, k\}$, if $N_i$ has been obtained from $N_{i-1}$ by applying an $\mathrm{SNPR}^+$ operation, then $T_i = T_{i-1}$.

(iii) For each $i \in \{1, 2, \ldots, k\}$, we have $\mathrm{d_{PR}}(T_{i-1}, T_i) \leq 1$.

By construction and since $\sigma$ contains at least $r$ $\mathrm{PR}^+$, there exists a subsequence of $S$ of length at moth $k - r$ that is a PR-sequence from $T_0$ to $T_k$. Hence, we have $\mathrm{d_{PR}}(T, T_k) \leq k - r$. Moreover, as $T_k \in D(N)$, it follows from Lemma 5.1 that $\mathrm{d_{PR}}(T_k, N) = r$ and thus

$$\begin{aligned} \min_{T' \in D(N)} \mathrm{d_{PR}}(T, T') + r &\leq \mathrm{d_{PR}}(T, T_k) + \mathrm{d_{PR}}(T_k, N) \\ &= k - r + r = k = \mathrm{d_{PR}}(T, N). \end{aligned} \tag{5.2}$$

The same holds for SNPR. Combining Inequalities 5.1 and 5.2 establishes the theorem. □

Given Theorems 5.4 and 5.10 and that $\mathrm{d_{PR}}(T, T') = \mathrm{d_{SNPR}}(T, T') = \mathrm{d_{SPR}}(T, T')$, it is worth noting that the problem of computing the PR-distance between a phylogenetic network and a phylogenetic tree can be reduced to computing the SPR-distance between pairs of trees. Calculating the SPR-distance between two phylogenetic trees is a well understood problem and several exact algorithms exist (e.g. [BS05,WBZ16]). Furthermore, this problem is known to be fixed-parameter tractable with the SPR-distance itself as parameter [BS05, Theorem 3.4]. This means that there exists an algorithm to compute $k = \mathrm{d_{SPR}}(T, T') = \mathrm{d_{PR}}(T, T')$ in $f(k)p(n)$ time where $f$ is a computable function that only depends on $k$ and $p$ is a polynomial function. Note that replacing $k$ by a function $f'(k)$ or calling such an algorithm as a black-box at most $f'(k)$ times, yields again a fixed-parameter tractable algorithm in $k$. We use this observation to establish the following theorem.

**Theorem 5.11.**
*Let $T \in \mathcal{T}_n$ and $N \in \mathcal{N}_n$. Then computing $\mathrm{d_{PR}}(T, N) = \mathrm{d_{SNPR}}(T, N)$ is fixed-parameter tractable when parameterised by $\mathrm{d_{PR}}(T, N)$.*

*Proof.* Let $d = \mathrm{d_{PR}}(T, N)$ and let $r$ be the number of reticulations of $N$. By Lemma 5.8 we know that $k = \mathrm{d_{SPR}}(T, T') = \mathrm{d_{PR}}(T, T') \leq d$ for all $T' \in D(N)$. From the observation before the theorem, it follows that computing $\mathrm{d_{SPR}}(T, T')$ is also fixed-parameter tractable when parameterised by $d$. Next, note that $|D(N)| \leq 2^r \leq 2^d$, since we know by Theorem 5.10 that $r \leq d$. Again, by the observation above, computing $\mathrm{d_{SPR}}(T, T')$ for at most $2^d$ trees $T' \in D(N)$ is still fixed-parameter tractable when parameterised by $d$. By Theorem 5.10 $\mathrm{d_{PR}}(T, N)$ can be computed by computing $\mathrm{d_{SPR}}(T, T')$ for each $T' \in D(N)$. Taken together, this implies that computing $\mathrm{d_{PR}}(T, N)$ is fixed-parameter tractable. □

## 5.2 Network to network

Next we analyse properties of shortest SNPR- and PR-sequences that connect a pair of phylogenetic networks. Let $\sigma = (N = N_0, N_1, \ldots, N_k = N')$ be a PR-sequence from $N$ to $N'$. We say that $\sigma$ *horizontally traverses* tier $r$ if $\sigma$ contains two networks $N_{i-1}$ and $N_i$ with $i \in \{1, 2, \ldots, k\}$ such that both have $r$ reticulations; in other words, $N_i$ and $N_{i-1}$ are $\mathrm{PR}^0$ neighbours.

Let $N, N' \in \mathcal{N}_n$ with $r$ and $r'$ reticulations, respectively. Without loss of generality, we may assume that $r \leq r'$. From a computational viewpoint and in trying to shrink the search space when computing $\mathrm{d_{PR}}(N, N')$, it would be favourable if there always exists a shortest PR-sequence connecting $N$ and $N'$ that traverses exactly one tier horizontally. In particular, if $r < r'$, it would have positive implications for computing $\mathrm{d_{PR}}(N, N')$ if all $\mathrm{PR}^0$ operations could be pushed to be the beginning or the end of a shortest PR-sequence from $N$ to $N'$. Note that we have seen with Theorem 5.10 that this is possible for a tree and a network. On the other hand, if $r = r'$, then the existence of a shortest PR-sequence from $N$ to $N'$ whose networks all belong to tier $r$ would allow us to compute $\mathrm{d_{PR}}(N, N')$ by considering only tier $r$. In what follows, we present several results showing that the existence of a shortest PR-sequence with such properties cannot be guaranteed. In addition, at the end of the section we give two bounds on the distance of $N$ and $N'$.

**Lemma 5.12.**
*Let $n \geq 4$. Let $N, N' \in \mathcal{N}_n$ with $r$ and $r'$ reticulations, respectively, such that $1 \leq r < r'$. Then there does not necessarily exist a shortest* SNPR*-sequence from $N$ to $N'$ that traverses at most one tier horizontally.*

*Proof.* To prove the statement, we show that every shortest SNPR-sequence for the two phylogenetic networks $N$ and $N'$ that are depicted in Figure 5.2 traverses at least two tiers horizontally.

We start by observing four differences between $N$ and $N'$:

- Leaf 1 is a descendant of a reticulation in $N$, but not in $N'$.
- Leaves 1 and 4 form a cherry in $N'$, but not in $N$.
- Leaves 2 and 3 form a cherry in $N'$, but not in $N$.
- Leaves 2 and 3 are descendants of two reticulations in $N'$, but not in $N$.

Since $N'$ has one more reticulation than $N$, at least one $\mathrm{SNPR}^+$ is required to transform $N$ into $N'$. Also note that an $\mathrm{SNPR}^+$ cannot create a cherry. Furthermore, note that an $\mathrm{SNPR}^0$ on $N$ (or a network derived from $N$ by an $\mathrm{SNPR}^+$) can create at most one cherry. Therefore, to transform $N$ into $N'$ at least three SNPR are necessary and thus $\mathrm{d_{SNPR}}(N, N') > 2$. Consequently, referring back to the networks shown in Figure 5.2,

$$\sigma = (N = N_0, N_1, N_2, N_3 = N')$$

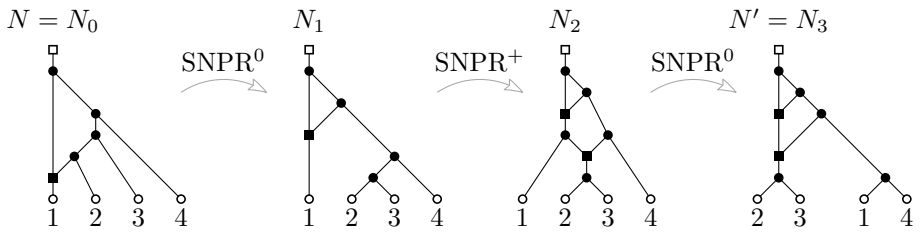


Figure 5.2: For the two networks $N$ and $N'$ shown, every shortest SNPR-sequence between them traverses two tiers horizontally.

is a shortest SNPR-sequence from $N$ to $N'$ that horizontally traverses tier 1 and tier 2.

To establish the statement, it is therefore sufficient to show that there exists no SNPR-sequence, say

$$\sigma^* = (N, M, M', N'),$$

such that $M$ can be obtained from $N$ by an SNPR$^+$, or $N'$ can be obtained from $M'$ by an SNPR$^+$. Note that a sequence that uses an SNPR$^+$ (or an SNPR$^-$) to transform $M$ into $M'$ would either be covered by one of these two cases or would be a sequence that traverses two tiers horizontally like $\sigma$. We thus proceed by distinguishing the first two cases.

First, assume that $\sigma^*$ exists and that $M$ has been obtained from $N$ by an SNPR$^+$. Then $M$ and $N'$ have the same four differences as listed above for $N$ and $N'$ with the exception that either leaf 2 or 3 (but not both) is possibly a descendant of two reticulations in $M$. Suppose that $M$ is indeed obtained from $N$ by (i) subdividing the incoming edge of leaf 1 with a new vertex $u$, subdividing the edge directed into 2 with a new vertex $v$, and adding the new edge $(u, v)$, or (ii) subdividing the incoming edge of leaf 1 with a new vertex $u$, subdividing the edge directed into 3 with a new vertex $v$, and adding the new edge $(u, v)$. Then $M$ would equal either the network $M_1$ or $M_2$ shown in Figure 5.3. In both cases, it requires two SNPR to transform $M$ into a network, say $M^*$, in which leaf 1 is not a descendant of any reticulation and leaves 2 and 3 are descendants of two reticulations. One such $M^*$ is shown in Figure 5.3. However, $M^* \neq N'$ and, so, it would take in total at least three SNPR operations to transform $M$ into $N'$. Now, suppose that $M$ is obtained from $N$ by an SNPR$^+$ other than (i) or (ii). With similar observations as above we note that again at least three SNPR operations are necessary to transform $M$ into $N'$. Hence, we conclude that $M$ has not been obtained from $N$ by an SNPR$^+$.

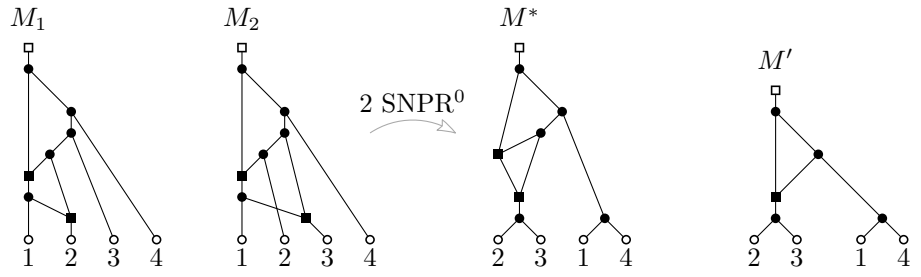


Figure 5.3: Networks in an SNPR-sequences from $N$ to $N'$ of Figure 5.2 for the proof of Lemma 5.12.

Second, assume that $\sigma^*$ exists and that $N'$ has been obtained from $M'$ by an SNPR$^+$ or, equivalently, $M'$ has been obtained from $N'$ by an SNPR$^-$. Then $M'$ is as shown in Figure 5.3 since each of the three SNPR$^-$ operations that can be applied to $N'$ results in the same network $M'$. Because of the aforementioned differences between $N$ and $N'$ that

are also differences between $N$ and $M'$ with the exception that 2 and 3 are descendants of only a single reticulation in $M'$, it takes at least three SNPR operations to transform $N$ into $M'$. Consequently, $N'$ has not been obtained from $M'$ in $\sigma^*$ by an SNPR$^+$.

Lastly, since neither $M$ nor $N'$ has been obtained from $N$ and $M'$, respectively, by an SNPR$^+$, it follows that $\sigma^*$ cannot be chosen so that no tier is horizontally traversed. This completes the proof. □

We can extend Lemma 5.12 to PR if we add one reticulation to our counterexample.

**Lemma 5.13.**
*Let $n \geq 4$. Let $N, N' \in \mathcal{N}_n$ with $r$ and $r'$ reticulations, respectively, such that $2 \leq r < r'$. Then there does not necessarily exist a shortest* PR*-sequence from $N$ to $N'$ that traverses at most one tier horizontally.*

*Proof.* We extend the proof of Lemma 5.12 from SNPR to PR based on the example shown in Figure 5.4. Again, we claim that the sequence $\sigma = (N = N_0, N_1, N_2, N_3 = N')$ is a shortest PR-sequence from $N$ to $N'$ and that there is no shortest PR-sequence that starts or ends with a PR$^+$. The proof of the former is analogous to the proof for SNPR. Moreover, the same is the case for the proof that no shortest PR-sequence from $N$ to $N'$ starts with a PR$^+$.

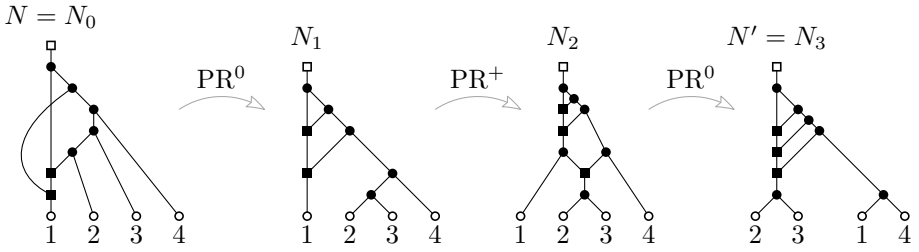


Figure 5.4: For the two networks $N$ and $N'$ shown, every shortest PR-sequence between them traverses two tiers horizontally.

It remains to show that no shortest PR-sequence from $N$ to $N'$ ends with a PR$^+$. Assume otherwise and let $\sigma^* = (N, M, M', N')$ be such a sequence. Then $M'$ can be obtained from $N'$ by a PR$^-$ and is as shown in Figure 5.5. We further know from the proof of Lemma 5.12 that there is no length two PR-sequence from $N$ to $M'$ that does not use a head PR$^0$. So assume first that $M$ is obtained from $N$ by a head PR$^0$ $\theta$. Only one of the four reticulation edges of $N$ could be pruned by $\theta$ to result in a network $M$ where leaf 1 is not child of a reticulation and such that the leaves 1 and 4 could be joined into a cherry with one further PR$^0$. See for example $M_1$ in Figure 5.5. However, we note that such $M$ (or $M_1$) does not have PR-distance one to $M'$. Assume therefore that $M$ is obtained from $M'$ by a head PR$^0$. With similar arguments we get that such $M$ (like $M_2$ in Figure 5.5) does not have PR-distance one to $N$. Therefore, a suitable $M$ cannot be obtained from $N$ or $M'$ by a head PR$^0$.

Since there is no PR-sequence from $N$ to $N'$ that starts or ends with a PR$^+$, it follows that $\sigma^*$ cannot exist and that each shortest PR-sequence from $N$ to $N'$ traverses two tiers horizontally. □

Bordewich et al. [BLS17, Proposition 7.5] have shown that

$$\mathrm{d}_{\mathrm{SNPR}}(N, N') \leq \min\{\mathrm{d}_{\mathrm{SNPR}}(T, T') \colon T \in D(N) \text{ and } T' \in D(N)\} + r + r',$$

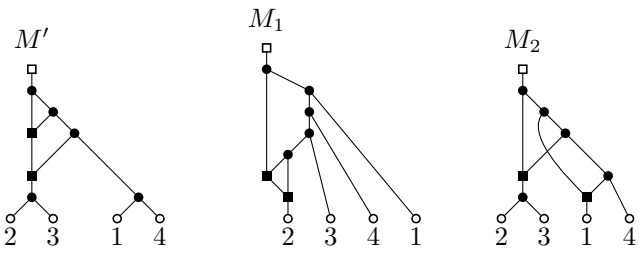


Figure 5.5: Networks in an SNPR-sequences from $N$ to $N'$ of Figure 5.2 for the proof of Lemma 5.12.

since there is always a path from $N$ to $N'$ via $\mathcal{T}_n$. If $N$ and $N'$ display the same tree $T$, then $r + r'$ gives an upper bound on the distance of $N$ and $N'$. We now prove that this bound can be sharp, even for PR.

**Lemma 5.14.**
*Let $r \geq 2$ and $n \geq 2r + 2$. There exist $\bar{N}_r, \bar{N}'_r \in \mathcal{N}_n$ with $r$ reticulations such that every shortest* SNPR- *and* PR-*sequence from $\bar{N}_r$ to $\bar{N}'_r$ contains a phylogenetic tree.*

*Proof.* To prove the statement, we show that every shortest PR-sequence

$$\sigma = (\bar{N}_r = N_0, N_1, \ldots, N_k = \bar{N}'_r)$$

connecting the two phylogenetic networks $\bar{N}_r$ and $\bar{N}'_r$ depicted in Figure 5.6 has length $2k$, for each $i \in \{1, 2, \ldots, r\}$, $N_i$ is obtained from $N_{i-1}$ by a PR$^-$, and for each $i \in \{r+1, r+2, \ldots, 2r\}$, $N_i$ is obtained from $N_{i-1}$ by a PR$^+$. Since $\bar{N}_r$ and $\bar{N}'_r$ both have $r$ reticulations, this implies that $\sigma$ contains a phylogenetic tree. Note that $\sigma$ exists because we can transform $\bar{N}_r$ into $\bar{N}'_r$ by removing each reticulation edge in $\{e_1, e_2, \ldots, e_r\}$ with a PR$^-$ and then adding each edge in $\{e'_1, e'_2, \ldots, e'_r\}$ with a PR$^+$. In addition, note that $\sigma$ is an SNPR-sequence.

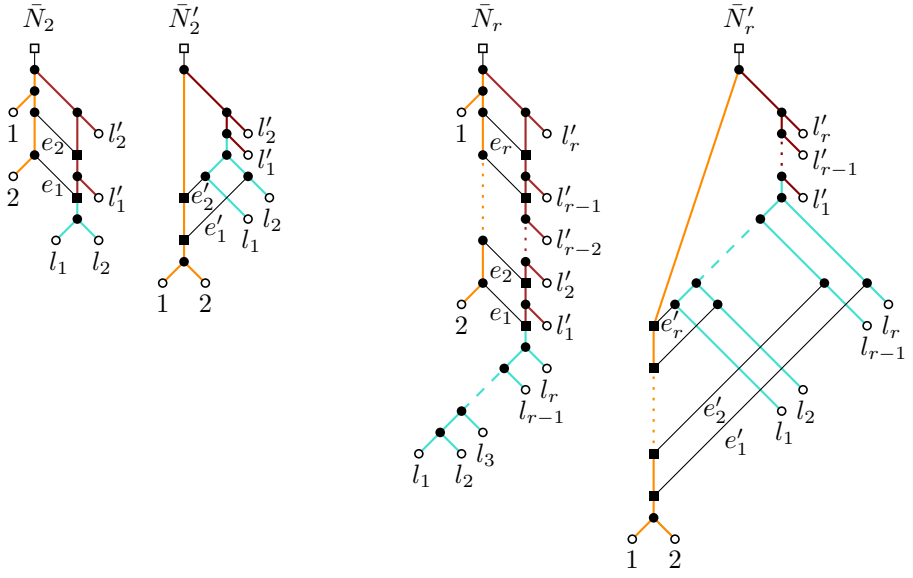


Figure 5.6: Construction that is used in the proof of Lemma 5.14 to show that, for each $r \geq 2$, there exist two phylogenetic networks $\bar{N}_r$ and $\bar{N}'_r$ such that every shortest SNPR- and PR-sequence from $\bar{N}_r$ to $\bar{N}'_r$ contains a phylogenetic tree.

We pause to observe three properties of $\bar{N}'_r$ that are crucial for the remainder of this proof:

(P1) For each $i \in \{1, 2, \ldots, r\}$, the leaf $l_i$ is sibling of a reticulation.

(P2) Leaves 1 and 2 form a cherry, and are descendants of all reticulations.

(P3) There exists a directed path $(\rho, w, v_1, v_2, \ldots, v_r)$, where $\rho$ is the root, $w$ is the child of $\rho$, and each $v_i$ with $i \in \{1, 2, \ldots, r\}$ is a reticulation.

To illustrate, for $r = 2$, the networks $\bar{N}_2$ and $\bar{N}'_2$ are shown in Figure 5.6.

Now assume that there exists a PR-sequence

$$\sigma^* = (\bar{N}_r = M_0, M_1, M_2, \ldots, M_{k'} = \bar{N}'_r)$$

from $\bar{N}_r$ to $\bar{N}'_r$ of length $k' \leq 2r$ that is distinct from $\sigma$. Let

$$O^* = (o_1, o_2, \ldots, o_{k'})$$

be the sequence obtained from $\sigma^*$ such that for each $i \in \{1, 2, \ldots, k'\}$ the following holds:

- $o_i = 0$ if $M_i$ is obtained from $M_{i-1}$ by a $\mathrm{PR}^0$,
- $o_i = +$ if $M_i$ is obtained from $M_{i-1}$ by a $\mathrm{PR}^+$, or
- $o_i = -$ if $M_i$ is obtained from $M_{i-1}$ by a $\mathrm{PR}^-$.

Let $m$ be the number of elements in $O^*$ that are equal to $-$.

**Case 1.** Assume that $m > r$. Since $\bar{N}_r$ and $\bar{N}'_r$ both have $r$ reticulations, $O^*$ contains exactly $m$ elements that are equal to $+$. Hence, $k' \geq 2m > 2r$; a contradiction.

**Case 2.** Assume that $m < r$. Again, since $\bar{N}_r$ and $\bar{N}'_r$ both have $r$ reticulations, $O^*$ contains exactly $m$ elements that are equal to $+$. Thus, with $k' \leq 2r$, it follows that $O^*$ contains at most $2(r-m)$ elements that are equal to 0. Let $i$ be an element in $\{1, 2, \ldots, k'\}$ such that $o_i = +$. Then, the number of leaves in $\{l_1, l_2, \ldots, l_r\}$ that are siblings of different reticulations in $M_{i-1}$ and $M_i$ differs by at most one. Therefore, we need at least $k_1 \geq r-m$ $\mathrm{PR}^0$ operations to obtain a network from $\bar{N}_r$ that satisfies (P1). Similarly, the number of vertices on a path that consists only of reticulations in $M_{i-1}$ and $M_i$ differs by at most one. Therefore, we need at least $k_2 \geq r - m$ $\mathrm{PR}^0$ operations to obtain a network from $\bar{N}_r$ that satisfies (P3).

Let $i \in \{1, 2, \ldots, k'\}$ such that $o_i = 0$. Assume that the number of leaves in $\{l_1, l_2, \ldots, l_r\}$ that are siblings of reticulations in $M_i$ is greater than this number in $M_{i-1}$. Then, a tail $\mathrm{PR}^0$ operation to obtain $M_i$ from $M_{i-1}$ either regrafts such a leaf $l_j$ as sibling to an incoming edge of a reticulation or regrafts a reticulation edge to the incoming edge of such a leaf. Alternatively, a head $\mathrm{PR}^0$ to obtain $M_1$ form $M_{i-1}$ regrafts a reticulation edge to the sibling edge of the incoming edge of such a leaf $l_j$. Therefore, in either case, this operation cannot increase the number of vertices that lie on a directed path of reticulations in $M_i$ compared to $M_{i-1}$. Similarly, if the number of vertices that lie on a directed path of reticulations in $M_i$ is greater than that number in $M_{i-1}$, then the number of leaves in $\{l_1, l_2, \ldots, l_r\}$ that are siblings of reticulations is not greater in $M_i$ than in $M_{i-1}$. Again, a $\mathrm{PR}^0$ operation cannot change both values for these networks at the same time. Overall, we observe that the $k_1$ $\mathrm{PR}^0$ used to satisfy property (P1) affect the leaves $l_j$ and reticulation edges, whereas the $k_2$ $\mathrm{PR}^0$ used to satisfy property (P3) affect the leaves $l'_j$ and (possibly) leaf 1. It follows that $k_1 = k_2 = (r - m)$ and, so, $k' = 2r$.

Lastly, to see that $M_{k'}$ does not satisfy property (P2), observe that neither the $k_1 + k_2$ $\mathrm{PR}^0$ operations nor the $2m$ $\mathrm{PR}^-$ and $\mathrm{PR}^+$ operations that are used to satisfy (P1) and (P3) result in a network that simultaneously satisfies (P2). Hence, it follows that at least one additional $\mathrm{PR}^0$ is needed to transform $\bar{N}_r$ into $\bar{N}'_r$; thereby contradicting that $k' \leq 2r$.
**Case 3.** Assume that $m = r$. Since $\bar{N}_r$ and $\bar{N}'_r$ both have $r$ reticulations and $k' \leq 2r$, it follows that $k' = 2r$. We complete the proof by showing that, for each $i \in \{1, 2, \ldots, r\}$, we have $o_i = -$ and, for each $i \in \{r+1, r+2, \ldots, 2r\}$, we have $o_i = +$. Assume that, for some $i \leq r$, we have $o_i = +$. Choose $i$ to be as small as possible. Let $v$ be the unique reticulation in $M_i$ that is not a reticulation in $M_{i-1}$. Then $v$ does not have leaves 1 and 2 as descendants and a leaf in $\{l_1, l_2, \ldots, l_r\}$ as a sibling of a reticulation. Now, as $O^*$ does not contain an element equal to 0, there exists an element $o_j = -$ with $j > i$ such that $M_j$ does not contain the reticulation edge that was added in transforming $M_{i-1}$ into $M_i$. In turn, this implies that the remaining $r - 1$ $\mathrm{PR}^+$ cannot transform $\bar{N}_r$ into a network that satisfies (P1) and (P3). Hence, if $m = r$, then $\sigma^*$ first uses $r$ $\mathrm{PR}^-$ and then $r$ $\mathrm{PR}^+$ like $\sigma$.
Combining all three cases establishes the statement. □

The statement of Lemma 5.14 requires $\bar{N}_r$ and $\bar{N}'_r$ to have at least two reticulations. Nevertheless, using a slightly different construction than that for $\bar{N}_r$ and $\bar{N}'_r$, Figure 5.7 shows two phylogenetic networks that both have one reticulation such that every shortest PR-sequence connecting these two networks contains a phylogenetic tree. While omitting a formal proof, we note that this claim can be verified by following the same ideas as in the proof of Lemma 5.14.

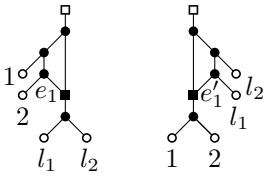


Figure 5.7: Two phylogenetic networks with one reticulation such that every shortest SNPR- and PR-sequence connecting them contains a phylogenetic tree.

Lemma 5.14 also implies that to compute $\mathrm{d}_{\mathrm{PR}}(N, N')$ it may be necessary to consider the space of all phylogenetic networks with at most $r$ reticulations. However, for SNPR even this can sometimes be insufficient. More precisely, we show that there are networks $N$ and $N'$ with $r$ reticulation such that every shortest SNPR sequence from $N$ to $N'$ contains a network with more than $r$ reticulations.

**Lemma 5.15.**
*Let $n \geq 2$, $r \geq 3$, and let $N, N' \in \mathcal{N}_n$ with $r$ reticulations.*
*There does not necessarily exist a shortest* SNPR*-sequence from $N$ to $N'$ such that each network in the sequence has at most $r$ reticulations.*

*Proof.* To establish the lemma, we show that every shortest SNPR-sequence that connects the two phylogenetic networks $N$ and $N'$ as depicted in Figure 5.8 contains a network with four reticulations. First observe that $\mathrm{d}_{\mathrm{SNPR}}(N, N') \geq 2$ and, so, the SNPR-sequence $(N, M, N')$ is of minimum length.

We complete the proof by showing that there exists no SNPR-sequence $(N, M', N')$ such that $M'$ is obtained from $N$ by an $\mathrm{SNPR}^-$ or $\mathrm{SNPR}^0$. Towards a contradiction, assume that $M'$ is obtained from $N$ by an $\mathrm{SNPR}^-$. Clearly, leaf 1 is a child of a reticulation in $M'$. Moreover, as $M'$ has two reticulations, it follows that $N'$ is obtained from $M'$ by an $\mathrm{SNPR}^+$ and that leaf 1 is still a child or descendant of a reticulation in $N'$; a contradiction.

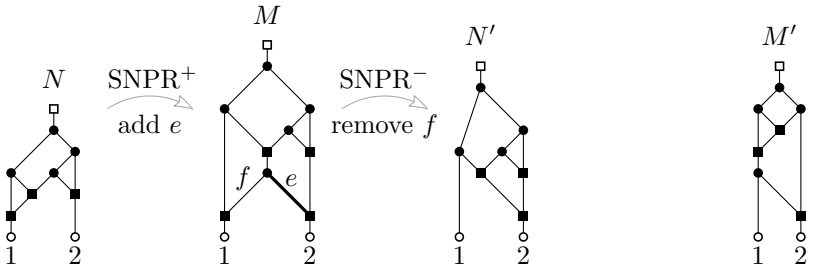


Figure 5.8: Every shortest SNPR-sequence from $N$ to $N'$, which both have three reticulations, contains a network with four reticulations. This example is used in the proof of Lemma 5.15

Now assume that $M'$ is obtained from $N$ by an $\mathrm{SNPR}^0$. If leaf 1 is a child of a reticulation in $M'$, then $\mathrm{d}_{\mathrm{SNPR}}(M', N') > 1$. We may therefore assume that leaf 1 is not a child of a reticulation in $M'$. Hence, $M'$ is the network that is shown on the right-hand side of Figure 5.8 in which all three reticulations lie on a directed path. It now follows that $\mathrm{d}_{\mathrm{SNPR}}(M', N') > 1$ because it requires at least two SNPR to transform $M'$ into a network in which not all three reticulations are on a path and where leaf 1 is not a descendant of any reticulation; again a contradiction. □

Note that the example used in Lemma 5.15 does not work for PR, since the networks $N$ and $N'$ have PR-distance one via a head $\mathrm{PR}^0$. In fact, we show that a $\mathrm{PR}^+$ directly followed by a $\mathrm{PR}^-$ can always be substituted with one or two $\mathrm{PR}^0$.

**Lemma 5.16.**
*Let $N, N' \in \mathcal{N}_{n,r}$ such that there is a* PR*-sequence $(N, M, N')$ that starts with a* $\mathrm{PR}^+$*. Then there is a* $\mathrm{PR}^0$*-sequence from $N$ to $N'$ of length at most two.*

*Proof.* Let the $\mathrm{PR}^+$ from $N$ to $M$ add the edge $e$ and let the $\mathrm{PR}^-$ from $M$ to $N'$ remove the edge $f$. If $f$ is also an edge of $N$, then it is straightforward to move $f$ to $e$ with two $\mathrm{PR}^0$. Otherwise, let $f'$ be the edge that gets subdivided when adding $e$ into $f$ and another edge. Depending on whether $f'$ gets subdivided by a tree vertex or a reticulation, $N'$ can be obtained from $N$ with a head or tail $\mathrm{PR}^0$ that prunes $f'$, respectively. □

In the previous section, we have seen with Lemma 5.7 that the distance of $N$ to a tree that it displays it at most $r$. We now generalise this to when $N'$ displays $N$.

**Lemma 5.17.**
*Let $N, N' \in \mathcal{N}_n$ with $r$ and $r'$ reticulations, respectively, such that $N'$ displays $N$. Then* $\mathrm{d}_{\mathrm{PR}}(N, N') = \mathrm{d}_{\mathrm{SNPR}}(N, N') = r' - r$.

*Proof.* Let $l = r' - r$ and let $k = \mathrm{d}_{\mathrm{PR}}(N, N')$. Note that $k \geq l$ since at least $l$ $\mathrm{PR}^+$ are needed from $N$ to $N'$ to increase the number of reticulations to $r'$. We now prove that $k \leq l$ by induction on $l$. If $l = 0$, then $N = N'$ and the statement holds.

Suppose that $l \geq 1$ and that the lemma holds whenever a network in $\mathcal{N}_n$ displays $N$ and has fewer than $r'$ reticulations. Let $S$ be a subdivision of $N$ that is a subgraph of $N'$; i.e. $S$ represents an embedding of $N$ into $N'$. Then there are $l$ reticulations in $N'$ such that not both its incoming edges are covered by $S$. Otherwise $N$ would have more than $r = r' - l$ reticulations. Among these $l$ reticulations of $N'$, let $v$ be the one closest to the root. Let $u$ and $u'$ be the parents of $v$. Then either $u$ or $u'$ or both are tree vertices. To see this, note that if $u$ and $u'$ were reticulations and covered by $S$, then both incoming edges of $v$ would be covered by $S$. That would be a contradiction to our choices of $v$. If

one of the parents was a reticulation and not covered by $S$, then we would have chosen that parent instead of $v$. So assume without loss of generality that $u$ is a tree vertex and that the edge $(u, v)$ is not covered by $S$. Remove $(u, v)$ with a $\mathrm{PR}^-$ from $N'$ to obtain a network with $l-1$ reticulations that still displays $N$. The statement now follows from the induction hypothesis. □

Next, Lemma 5.7 gives us a lower bound on the distance of $N$ and $N'$ with regards to the trees they display.

**Lemma 5.18.**
*Let $N, N' \in \mathcal{N}_n$. Then $\mathrm{d_{SNPR}}(N, N') \geq \mathrm{d_{PR}}(N, N') \geq \max_{T \in D(N)} \min_{T' \in D(N')} \mathrm{d_{PR}}(T, T')$.*

*Proof.* Note that the first inequality holds by the definition of SNPR. We prove the second one. Let $k = \mathrm{d_{PR}}(N, N')$. By Lemma 5.7 we know that for every tree $T \in D(N)$ there is a tree $T' \in D(N')$ with $\mathrm{d_{PR}}(T, T') \leq k$. Hence, the minimum distance of any tree in $D(N)$ to a tree in $D(N')$ provides a lower bound for $k$. □

## 5.3 Isometric relations between classes

In Figure 2.3 we have seen inclusion relationships between different classes of phylogenetic networks. Then in Chapter 3 we have proven that all these classes form metric spaces under SNPR and PR. We now look at the isometric relationships of these spaces. In particular, we want to know whether moving between two networks of a certain class might be faster if we leave this class. We can also rephrase this into the question of whether adding constraints to networks increases the distance.

Above we have seen and proven that $\mathcal{T}_n$ is an isometric subgraph of $\mathcal{N}_n$ under SNPR and PR. Concerning higher tiers of $\mathcal{N}_n$, however, we get the following theorem from the results of the previous section.

**Theorem 5.19.**
*Let $n \geq 4$ and $r \geq 1$.*
*Then $\mathcal{N}_{n,r}$ is not an isometric subgraph of $\mathcal{N}_n$ under* SNPR *and* PR*. Moreover, for $r \geq 3$ and* SNPR*, $\mathcal{N}_{n,r}$ is not an isometric subgraph of the class of phylogenetic networks in $\mathcal{N}_n$ that have at most $r$ reticulations.*

*Proof.* The first statement follows from Lemma 5.14 for $r \geq 2$ and from Figure 5.7 for $r = 1$. The second statement follows from Lemma 5.15. □

We now show that allowing parallel edges can reduce the distance between two networks (following the proof of Proposition 16 by the author and Linz [KL19]).

**Lemma 5.20.**
*Let $n \geq 3$. Let $\mathcal{N}_n^*$ be the class of phylogenetic networks in $\mathcal{N}_n$ that do not contain parallel edges. Then $\mathcal{N}_n^*$ is not an isometric subgraph of $\mathcal{N}_n$ under* SNPR *and* PR*.*

*Proof.* To establish the statement, we give an explicit examples of two networks $N$ and $N'$ in $\mathcal{N}_n^*$ whose SNPR- and PR-distance in $\mathcal{N}_n^*$ is greater than in $\mathcal{N}_n$. For this let $N$ and $N'$ be the networks shown in Figure 5.9. Then $\sigma = (N, M, N')$ is a length two SNPR- and PR-sequence from $N$ to $N'$. Note that $N'$ can be obtained from $N$ by swapping the leaves 1 and 3. Since leaf 3 is the child of a reticulation in $N$, it cannot be pruned in $N$. The sequence $\sigma$ thus prunes the edge incident to leaf 1 to regraft it above leaf 3, which then

enables the edge incident to leaf 3 to be pruned and regrafted to the former position of leaf 1.

Towards a contradiction, assume that there exists a PR-sequence $\sigma^* = (N, M', N')$ distinct from $\sigma$. Suppose $\sigma^*$ does not start by pruning the edge incident to leaf 1. Then leaf 1 has to be moved or a triangle above it constructed from $M'$ to $N'$. Furthermore, the edge incident to leaf 3 cannot be pruned in $N$, so leaf 3 has to be moved from $M'$ to $N'$. However, making both these changes is not possible with a single PR. Therefore, $\sigma$ is the unique length two SNPR- and PR-sequence in $\mathcal{N}_n$ that connects $N$ and $N'$. Hence, as $M$ contains a pair of parallel edges, we have that the SNPR- and PR-distance of $N$ and $N'$ in $\mathcal{N}_n$ is two, but at least three in $\mathcal{N}_n^*$. □

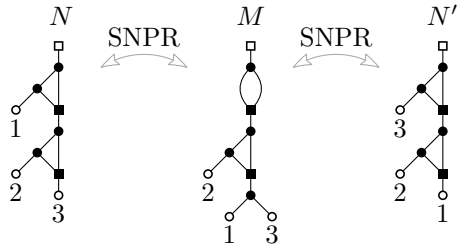


Figure 5.9: Two networks $N, N'$ without parallel edges for which the only shortest SNPR- and PR-sequence in $\mathcal{N}_n$ goes through $M$, which contains a pair of parallel edges. This example is used in the proof of Lemma 5.20.

We can use the same line of argument as in Lemma 5.20 on the examples in Figure 5.10 to prove the following relations.

**Theorem 5.21.**
*Let $n \geq 5$. Then under* SNPR *and* PR

*(i) $\mathcal{NN}_n$ is not an isometric subgraph of $\mathcal{TC}_n$ and $\mathcal{N}_n$,*

*(ii) $\mathcal{TC}_n$ is not an isometric subgraph of $\mathcal{TS}_n$ and $\mathcal{N}_n$,*

*(iii) $\mathcal{RV}_n$ is not an isometric subgraph of $\mathcal{TB}_n$ and $\mathcal{N}_n$,*

*(iv) $\mathcal{TS}_n$ is not an isometric subgraph of $\mathcal{TB}_n$ and $\mathcal{N}_n$, and*

*(v) $\mathcal{TB}_n$ is not an isometric subgraph of $\mathcal{N}_n$.*

Note that while Francis and Steel [FS15] allow tree-based networks to have edges in parallel, it is easily seen that one can add a single edge to each of the three networks depicted in Figure 5.10 (v) to obtain an example showing that the class of tree-based networks without parallel edges on $n$ leaves is not an isometric subgraph of $\mathcal{N}_n$ either.

Next we look at level-$k$ networks.

**Lemma 5.22.**
*Let $n \geq 4$ and $i \geq 1$.*
*Then $\mathcal{LV}_{i,n}$ is not an isometric subgraph of $\mathcal{LV}_{i+1,n}$ and $\mathcal{N}_n$ under* SNPR *and* PR.

*Proof.* To establish the statement for level-1 networks, we give an explicit example of two networks $N$ and $N'$ in $\mathcal{LV}_{1,n}$ whose SNPR- and PR-distance is three in $\mathcal{LV}_{1,n}$ but only two in $\mathcal{LV}_{2,n}$. For this let $N$ and $N'$ be the networks that are shown in Figure 5.9. Then $\sigma = (N, M, N')$ is a length two SNPR- and PR-sequence from $N$ to $N'$. It is straightforward

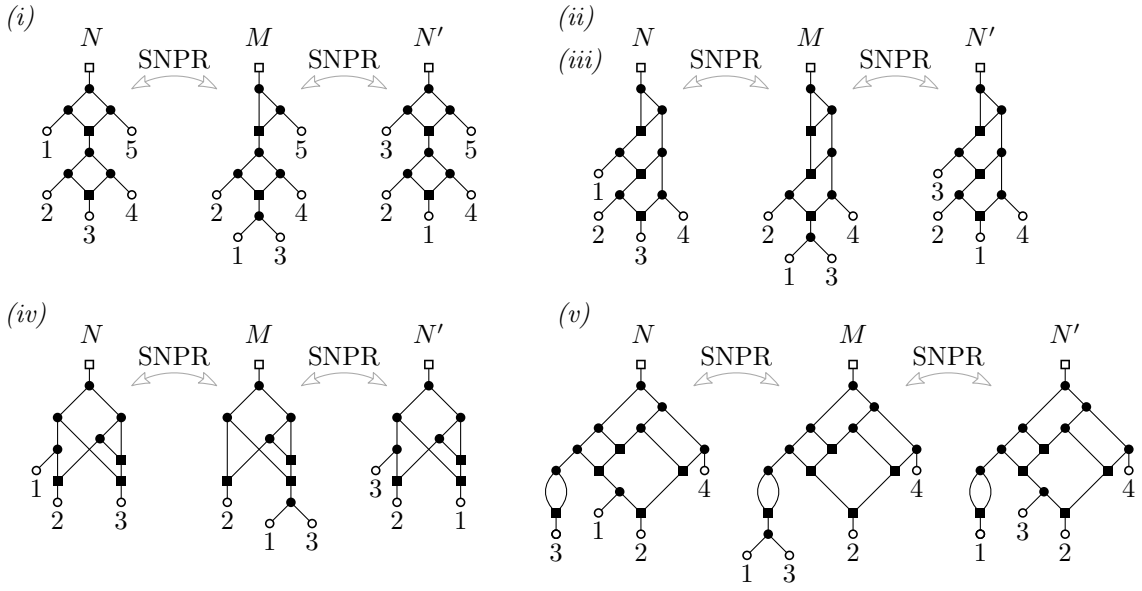


Figure 5.10: Examples to prove the statements of Theorem 5.21 with the same numbering. In each example, the two networks $N$ and $N'$ are of the class of (i) normal, (ii) tree-child, (iii) reticulation-visible, (iv) tree-sibling, or (v) tree-based networks, and only differ by interchanging the labels 1 and 3. The only shortest SNPR- and PR-sequence between $N$ and $N'$ is through $M$, which is not in this class.

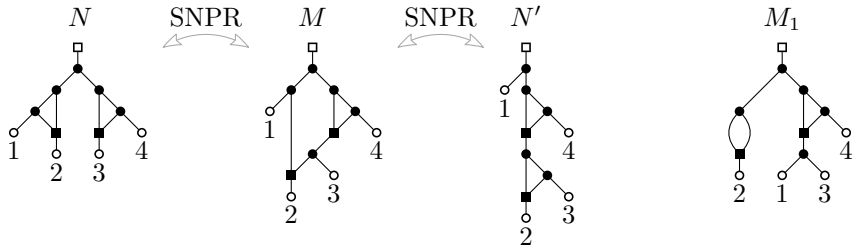


Figure 5.11: Two networks $N, N' \in \mathcal{LV}_{1,n}$ for which the only shortest SNPR- and PR-sequence in $\mathcal{N}_n$ goes through $M \notin \mathcal{LV}_{1,n}$. This example is used in the proof of Lemma 5.22.

to check that $\mathrm{d_{PR}}(N, N') > 1$ and hence $\sigma$ is a shortest path. Note that $M$ is not a level-1 network.

We claim that there is no PR-sequence of length two from $N$ to $N'$ within $\mathcal{LV}_{1,n}$. Towards a contradiction, assume the contrary, namely that there exists such a PR-sequence $\sigma^* = (N, M', N')$. Suppose $\sigma^*$ starts with a $\mathrm{PR}^-$ and thus removes a triangle. However, then the $\mathrm{PR}^+$ from $M'$ to $N'$ cannot add a triangle that is incident to both leaf 2 and leaf 3. Next, suppose $\sigma^*$ starts with a $\mathrm{PR}^+$ that does not create a level-2 network. However, then the $\mathrm{PR}^-$ applied to $M'$ can only remove one of the three blobs of $M'$ and the resulting network can thus not have a triangle that is incident to both leaf 2 and 3. Last, suppose that $\sigma^*$ starts with a $\mathrm{PR}^0$ that does not create a level-2 network. Note that it takes at least two $\mathrm{PR}^0$ to create a triangle that is incident to both leaf 2 and 3 as in $N'$, since leaf 1 has to, so to say, be pruned from the triangle above leaf 2 and and leaf 3 attached to it. Therefore the $\mathrm{PR}^0$ from $N$ to $M'$ has to work towards that. Going through all possible $\mathrm{PR}^0$, we find that the only suitable $\mathrm{PR}^0$ prunes leaf 1 and attaches it to the edge incident

leaf 3. Then $M'$ is as $M_1$ in Figure 5.11. (Note that a $\mathrm{PR}^0$ on $N$ that regrafts leaf 1 to another edge, would result in a network $M'$ where leaf 3 cannot be moved.) As a result, leaf 3 is prunable in $M' = M_1$ and can be moved to the pair of parallel edges above leaf 2 with another $\mathrm{PR}^0$. This creates the desired triangle. However, the resulting network is not $N'$. The sequence $\sigma^*$ can thus not avoid a level-2 network $M'$. Hence, we have that the SNPR- and PR-distance of $N$ and $N'$ is two in $\mathcal{LV}_{2,n}$, but at least three in $\mathcal{LV}_{1,n}$.

Lastly, note that the example can easily be extended to higher levels and $n$. □

We close this section with two remarks. Firstly, note that, for high enough $r$, Theorem 5.21 and Lemma 5.22 also hold for the tiers $r$ of the respective classes, that is, $\mathcal{C}_{n,r}$ is not an isometric subgraph of $\mathcal{N}_{n,r}$. Secondly, the networks presented in this chapter may seem rather small. However, they can be regarded as skeletons of larger networks with the same properties. For instance, in all examples that we used to establish the results of this chapter, there is a leaf that can be replaced with a subtree or a subnetwork. Furthermore, some edges can be subdivided to add further reticulation edges or subtrees to obtain larger networks with the same properties.

## 5.4 Concluding remarks

In this chapter we have looked at shortest paths in the space of phylogenetic networks under SNPR and PR between trees and networks. Using the sets of trees displayed by a network, we gave several bounds on the distance of two trees or a tree and a network. One interpretation of these results is that it is not faster to walk in higher tiers from one network $N$ to another network $N'$ than it is from the displayed trees of $N$ to the displayed trees of $N'$. Furthermore, we have seen that $\mathcal{T}_n$ is an isometric subgraph of $\mathcal{N}_n$ under PR. This implies that it is $\mathcal{NP}$-hard to compute the PR-distance in $\mathcal{N}_n$. On the positive side, we characterised the distance of a tree $T$ and a network $N$ in terms of the trees displayed by $N$. This allowed us to use fixed-parameter tractable algorithm for the computation of the SPR-distance of two trees as a black-box for a fixed-parameter tractable algorithm for the computation of the PR-distance of $T$ and $N$.

Looking at the distance of two networks in tier $r$, we found that an exhaustive algorithm that wants to compute PR-distance in $\mathcal{N}_n$ may have to include all tiers below tier $r$, including trees, into the search space. In addition, we showed that for SNPR even the tier above may have to be included. This means that the tiers $\mathcal{N}_{n,r}$ with $r \geq 1$ are not isometric subgraphs of $\mathcal{N}_n$ under SNPR and PR. There remain several open problems. First, are there two networks $N$ and $N'$ with $r$ reticulations such that every shortest PR-sequence from $N$ to $N'$ traverses tier $r+1$? We have seen that the PR-distance of such networks has to be at least three. Second, are there two networks $N$ and $N'$ with $r$ reticulations such that every shortest SNPR-sequence from $N$ to $N'$ traverses tier $r+l$ where $l$ is at least two? Also, how large can $l$ be?

Further negative results showed that the most popular classes of phylogenetic networks are not isometric subgraphs of $\mathcal{N}_n$. Moreover, this is not even the case if we restrict it to the tiers of the class and $\mathcal{N}_{n,r}$. This means that when considering two networks of a certain class in tier $r$, one has to decide which PR-distance one wants to consider: the PR-distance in that class, in $\mathcal{N}_{n,r}$, or in $\mathcal{N}_n$. This raises the question whether there is a class of phylogenetic networks for which the SNPR- or the PR-distance in that class equals the distance in $\mathcal{N}_n$.

# 6. Agreement graph and distance

In this chapter we look at how the notion of agreement forests on phylogenetic trees can be generalised for phylogenetic networks.

In the previous chapter we have seen that working with sequences of rearrangement operations can be challenging. For unrooted trees, Hein et al. [HJWZ96] proposed the use of agreement forest for this problem. Agreement forests have been adopted to rooted trees and work as follows. Consider the two trees $T$ and $T'$ in Figure 6.1 and the SPR-sequence from $T$ to $T'$. Each of the SPR prunes one edge and then regrafts it again. If we only carry out these prunings but do not regraft the edges, we end up with a forest $F$ consisting of multiple smaller phylogenetic trees. Now, reversing these prunings by regrafting the edges again we end up with $T$ and, similarly, regrafting the edges like in the SPR-sequence we end up with $T'$. Hence we note that $T$ and $T'$ "agree" on $F$, which is thus called an agreement forest of $T$ and $T'$.

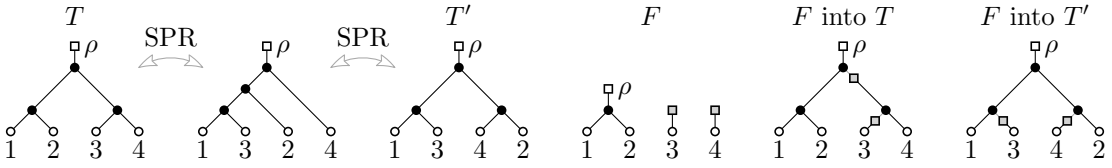


Figure 6.1: A length two SPR-sequence from $T$ to $T'$ that first prunes leaf 3 and then leaf 4. This yields the agreement forest $F$ of $T$ and $T'$. On the right is shown how $F$ can be embedded into $T$ and $T'$, where a grey edges show which pairs of vertices get identified by the embedding.

A special case occurs when, in an SPR-sequence, an edge gets regrafted to the root edge and then its resulting sibling edge gets pruned. See for example the SPR-sequence shown in Figure 6.2. In this case, the agreement forest $F$ of $T$ and $T'$ contains the isolated vertex labelled $\rho$.

Recall that a graph $G$ consisting of multiple components has an embedding into another graph $H$ if the components of $G$ have pairwise edge-disjoint embeddings into $H$. In addition we require that an embedding maps labelled vertices of $G$ to vertices with the same label in $H$. Looking again at the examples in Figures 6.1 and 6.2 we note that the agreement forest $F$ has embeddings into $T$ and into $T'$ such that all their edges are covered. Our definition of an agreement forest and its generalisation below is based on this observation.

An agreement forest $F$ of $T$ and $T'$ with the minimum number of components among all

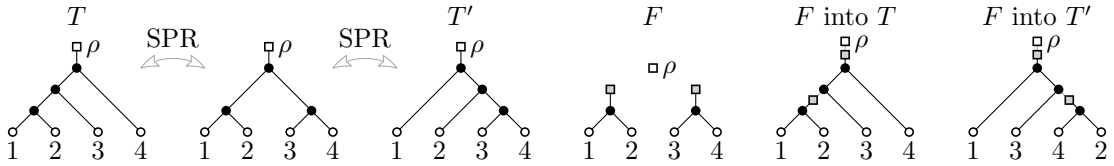


Figure 6.2: A length two SPR-sequence from $T$ to $T'$ that first prunes the cherry $\{1, 2\}$ and then the cherry $\{3, 4\}$. This yields the agreement forest $F$ of $T$ and $T'$. Note that $F$ contains the isolated vertex labelled $\rho$. On the right is shown how $F$ can be embedded into $T$ and $T'$.

agreement forests of $T$ and $T'$ is called a *maximum agreement forest* (MAF). Note that in the examples above each SPR causes one component to split. Hence, Hein et al. [HJWZ96] claimed that in the unrooted case the number of components minus one would equal the SPR-distance of two unrooted trees. This was then corrected by Allen and Steel [AS01], who showed that this is not the case for SPR, but for TBR on unrooted trees. Later Bordewich and Semple [BS05] proved that, for the rooted case, if $F$ is a MAF, then its number of components characterises the SPR-distance of $T$ and $T'$. More precisely, for a maximum agreement forest $F$ of $T$ and $T'$ with $k$ components, we define $\mathrm{m}(T, T') = k - 1$.

**Theorem 6.1** (Bordewich and Semple [BS05, Theorem 2.1])**.**
*Let $T, T' \in \mathcal{T}_n$. Then*

$$\mathrm{d_{SPR}}(T, T') = \mathrm{m}(T, T').$$

Bordewich and Semple proved Theorem 6.1 by converting a shortest SPR-sequence into an agreement forest and, conversely, deriving a SPR-sequence from a maximum agreement forest.

This characterisation of the SPR-distance turned out to be a practical tool. Instead of having to work with countless intermediate trees in possible SPR-sequences from $T$ to $T'$, one can argue about the distance of $T$ and $T'$ based on a single graph – a MAF. For example, using MAFs it has been shown that the following reduction rules preserve the SPR-distance (or the TBR-distance in the unrooted case). We can replace common pendant subtrees of $T$ and $T'$ with a new leaf and replace chains of common leaves that occur identically in both trees with three new leaves. In other words, if $T$ and $T'$ agree on a pendant subtree (or a chain), then this subtree is fully contained in one of the trees for a MAF of $T$ and $T'$. Using the reduction rules as kernelisation, fixed-parameter tractable algorithms for the problem of computing the SPR-distance parameterised with its natural parameter have been developed [AS01, BS05]. Similarly, a divide-and-conquer approach for the computation of the SPR-distance has been developed that uses MAFs [LS11]. Furthermore, MAFs have been used for exact and approximation algorithms [HJWZ96, RSW07, BMS08, BSJ09, Wu09, WZ09, SFYW16, SvZvdS16, CHW17]. Beyond two phylogenetic trees, MAFs have also been generalised for non-binary trees and for sets of more than two trees, again to develop fixed-parameter tractable algorithms and approximation algorithms [Cha05, RSW07, vIKLS14, CFS15, CSW16]. With slight modification, agreement forests have also been used for results on the Hybridisation Number problem [LS09, HL18]. Another problem related to MAF is the Maximum Agreement Subtree problem (MAST), which ask for the largest common subtree of a set of given phylogenetic trees [SW93, AK97, CFCH$^+$00, Mar18]. This problem has also been extended to the Maximum Agreement Subnetwork problem for two networks [CJSS05].

Here we are interested in how we can generalise MAFs to model a PR-sequence between two networks $N$ and $N'$ and whether such a generalisation can characterise the

PR-distance. Just like for an SPR-sequence, we thus look at what happens when we only consider the pruning part, but not the regrafting part of a PR-sequence. In addition, we find a way to model that $N$ and $N'$ might have a different number of reticulation and thus also a different number of edges. In this chapter we develop such a generalisation of MAFs for networks. Based on this we define a metric called agreement distance and then consider whether it characterises the PR-distance. Using the results from the previous chapter, we are able to show that this is the case for the distance of a tree and a network. However, we show that the agreement distance does not in general characterise the PR-distance of two networks. Nevertheless, we prove that it still bounds the PR-distance and thus the SNPR-distance with constant factors.

*Remark.* This chapter is based on "The agreement distance of rooted phylogenetic networks" [Kla19] except for Section 6.3.2, which is derived from the joint work with Simone Linz "On the Subnet Prune and Regraft Distance" [KL19].

## 6.1 Agreement graph

In this section we define maximum agreement graphs for two phylogenetic networks $N$ and $N'$. The main idea is to find a graph that can be obtained from both $N$ and $N'$ with a minimum number of prunings. Throughout this section, let $N, N' \in \mathcal{N}_n$ with $r$ and $r'$ reticulations, respectively. Without loss of generality, assume that $r' \geq r$ and let $l = r' - r$ be the difference in the number of reticulations of $N$ and $N'$. Further recall that we assume that a path in a directed graph is a directed path and contains at least one edge.

**Prunings and sprouts.** Let $G$ be a directed graph. Let $u$ be a vertex of $G$ that is either labelled or has degree three. Let $(u, v)$ be an edge of $G$. Recall that a *pruning* of $(u, v)$ at $u$ is the process of deleting $(u, v)$ and adding a new edge $(\bar{u}, v)$, where $\bar{u}$ is a new (unlabelled) vertex. If $u$ is now an indegree one, outdegree one vertex, then we suppress $u$. Note that a pruning does not remove a label from $u$. The definition for a pruning of the edge $(u, v)$ at $v$ is analogous. We mostly apply a pruning to a phylogenetic network or a graph derived from a phylogenetic network. Therefore, the restriction that $u$ is either labelled or has degree three can be understood as $u$ being either the root $\rho$, a (labelled) leaf, or an internal vertex.

A *sprout* of $G$ is an unlabelled degree one vertex of $G$. For example, applying a pruning to a phylogenetic network yields a graph with exactly one sprout. A *t-sprout* (resp. *h-sprout*) is a sprout that is the tail (resp. head) of its incident edge.

**Agreement embeddings.** Let $(u, v)$ be an edge of $N$ with $u$ either a labelled vertex; i.e. the root $\rho$, or a degree-three vertex. Consider a graph $G$ obtained from $N$ by pruning $(u, v)$ at $u$. Then $G$ has exactly one sprout $\bar{u}$, and $n + 1$ labelled vertices of which $n$ are bijectively labelled by $\mathcal{X}$ and one with $\rho$. We can distinguish three cases. If $u$ is $\rho$ of $N$, then $G$ contains an isolated vertex labelled $\rho$, say, $\bar{u}'$. If $u$ is a reticulation in $N$, then $G$ contains an indegree two, outdegree zero vertex, say, $\bar{u}'$. If $u$ is a inner tree vertex in $N$, then $u$ gets suppressed in the process of the pruning. In the first two cases, we get a canonical embedding of $G$ into $N$ that is a bijection of the edges of $G$ to the edges of $N$ and a surjection of the vertices of $G$ to the vertices of $N$. Only $\bar{u}$ and $\bar{u}'$ of $G$ get mapped to $u$ of $N$. In the third case, we obtain such an embedding for a subdivision of $G$ (which reverses the suppression of $u$) into $N$. The case for pruning $(u, v)$ at $v$ is similar. Together the three cases motivate the following definition.

Let $G$ be a directed graph. We say $G$ has an *agreement embedding* into $N$ if there exists

an embedding of $G$ into $N$ with the following properties.

- The pairwise edge-disjoint embeddings of the components of $G$ into $N$ together cover all edges of $N$.
- At most two vertices of $G$ are mapped to the same vertex of $N$. In this case, one of these two vertices of $G$ is a sprout and the other is either a labelled isolated vertex, or an indegree two, outdegree zero vertex, or an indegree zero, outdegree two vertex.
- For each labelled vertex $v$ of $N$, there exists exactly one vertex $\bar{v}$ with the same label in $G$ and $\bar{v}$ is mapped to $v$.

Note that if $G$ has an agreement embedding into $N$, then $G$ has $n+1$ labelled vertices of which $n$ are bijectively labelled by $\mathcal{X}$ and one with $\rho$. Furthermore, note that to every inner tree vertex of $N$ either a tree vertex, or a t-sprout, or an h-sprout and an outdegree two, indegree zero vertex of $G$ gets mapped. The situation is similar for a reticulation, a leaf, and the root.

**Lemma 6.2.**
*Let $G$ be a directed graph and $N \in \mathcal{N}_n$.*
*Then $G$ has an agreement embedding into $N$ if and only if $G$ can be obtained from $N$ by a sequence of prunings.*

*Proof.* If $G$ can be obtained from $N$ by a sequence of prunings, then an agreement embedding of $G$ into $N$ follows naturally. So assume that $G$ has an agreement embedding into $N$. Then $G$ can be constructed from a sequence of prunings as follows. Assume that $G$ contains a t-sprout $\bar{u}$. If $\bar{u}$ is mapped to a vertex $u$ and the edge $(\bar{u}, \bar{v})$ is mapped to the path from $u$ to $v$ in $N$, with $w$ the child of $u$ in this path, then prune the edge $(u, w)$ at $u$. This covers either of the cases of when $\bar{u}$ is mapped to $\rho$, or to the same reticulation to which a degree two vertex of $G$ is mapped, or to a tree vertex of $N$ that lies on a path to which an edge of $G$ is mapped. In either case, applying this pruning also either creates the isolated vertex $\rho$, a degree two vertex, or suppresses a vertex, respectively. A pruning for an h-sprout works analogously. We can find such a pruning for each sprout of $G$. Now consider the case where we have identified that we want to prune the edge $e = (u, v)$ at $u$ and the edge $f = (u, w)$ at $w$. Let $p$ be the parent of $u$. If we now prune $e$ at $u$, then the edges $f$ and $(p, u)$ are removed when suppressing $u$ and a new edge $f' = (p, w)$ added. In the resulting graph, we cannot prune $f$, but instead now want to prune $f'$ at $w$. Further note that since $G$ has an agreement embedding into $N$, no two edges have to be pruned at the same vertex. Hence, we can apply one pruning after the other on the edges identified in $N$ or on the edges they get extended to by preceding prunings. As noted, this does not only create the sprouts, but also the labelled, isolated vertices and degree-two vertices and shrinks the path of $N$ to which edges of $G$ get mapped to edges. Hence, this sequence of prunings results in $G$. □

**Agreement graphs.** Recall that we assume that $N'$ has $l$ more reticulations than $N$. Let $G$ be a directed graph with connected components $S_1, \ldots, S_k$ and $E_1, \ldots, E_l$ such that the $E_1, \ldots, E_l$ each consist of a single directed edge. Then $G$ is an *agreement graph* of $N$ and $N'$ if

- $G$ without $E_1, \ldots, E_l$ has an agreement embedding into $N$, and
- $G$ has an agreement embedding into $N'$.

For such an agreement graph, we refer to each $S_i$ as an *agreement subgraph* and to each $E_j$ as a *disagreement edge*. A *maximum agreement graph* $G$ for $N$ and $N'$ is an agreement graph for $N$ and $N'$ with the minimum number of sprouts. Figures 6.3 and 6.4 give two examples of maximum agreement graphs.

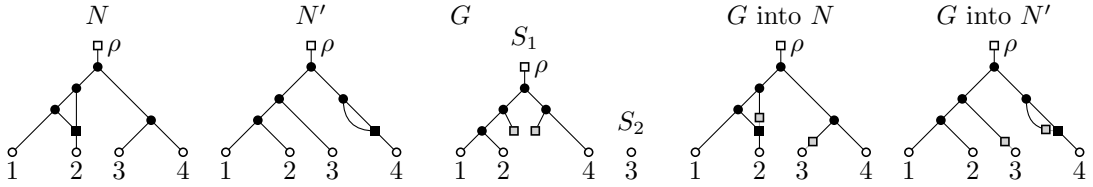


Figure 6.3: A maximum agreement graph $G$ for $N$ and $N'$ with its agreement embeddings into $N$ and $N'$ shown on the right. Note that the agreement subgraph $S_2$ consists of a labelled isolated vertex.

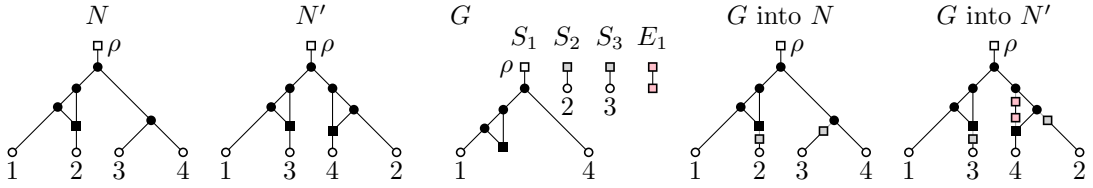


Figure 6.4: A maximum agreement graph $G$ for $N$ and $N'$ with its agreement embeddings into $N$ and $N'$ shown on the right. Note that the disagreement edge $E_1$ is only used for $N'$.

An *agreement forest* $F$ for two trees $T$ and $T'$ is an agreement graph for $T$ and $T'$ where each agreement subgraph $S_i$ for $i \in \{2, \dots, k\}$ is a phylogenetic tree with an unlabelled root and $S_1$ is either a phylogenetic tree (with the root labelled $\rho$) or an isolated vertex labelled $\rho$. Note that an agreement forest $F$ contains no h-sprouts and that thus in the respective agreement embeddings of $F$ into $T$ and $T'$ a sprout of $F$ is mapped either to the root $\rho$ or to a subdivision vertex of an edge of another agreement subgraph. See again Figures 6.1 and 6.2 for examples. On the other hand, considering shortest PR-sequences between $N$ and $N'$ in the examples in Figures 6.3 and 6.4 shows why in general in an agreement embedding of an agreement graph $G$ a sprout may have to be mapped to the same vertex as a labelled isolated vertex ($\rho$ or a leaf of $N$ and $N'$) or an unsuppressible degree-two vertex of $G$.

Before we show that maximum agreement graphs induce a metric on $\mathcal{N}_n$, we establish further notation and terminology to ease talking about agreement embeddings and agreement graphs. We use $\bar{e}, \bar{f}, \bar{u}, \bar{v}$ if we refer to edges or vertices of an agreement graph and $e, f, u, v$ for edges and vertices of $N$ or $N'$. If we use symbols like $\bar{u}$ and $u$ in the same context, then $\bar{u}$ is usually mapped to $u$ by the agreement embedding under consideration.

Let $G = (V_G, E_G)$ be a graph with an agreement embedding into a network $N = (V_N, E_N)$. We say a sprout $\bar{u} \in V_G$ is *attached to* $\bar{e} \in E_G$ *in* $N$ if $\bar{u}$ is mapped to a vertex $u \in V_N$ that is an internal vertex of the path to which $\bar{e}$ is mapped. Similarly, we say $\bar{u} \in V_G$ is *attached to* $\bar{x} \in V_G$ *in* $N$ if $\bar{u}$ and $\bar{x}$ are mapped to the same vertex $x \in V_N$. We say an edge $\bar{e} = (\bar{u}, \bar{v}) \in E_G$ is *attached to* $\bar{f} \in E_G$ *in* $N$ if either $\bar{u}$ or $\bar{v}$ is a sprout and attached to $\bar{f}$. Note that $\bar{e}$ being attached to $\bar{f}$ does not imply $\bar{f}$ being attached to $\bar{e}$. Considering the example in Figure 6.3 and the agreement embedding of $G$ into $N$, note that one sprout is attached to the incoming edge of leaf 2 in $N$ and another sprout is attached to the isolated vertex labelled 3 in $N$.

**Embedding changes.** Note that a graph $G$ may have several agreement embeddings into $N$ or $N'$. We now describe how, in some cases, an agreement embedding can be changed into another one. For this, let $\bar{u}$ and $\bar{v}$ be two t-sprouts of $G$ with outgoing edges $\bar{e} = (\bar{u}, \bar{w})$ and $\bar{f} = (\bar{v}, \bar{z})$, respectively, such that $\bar{u}$ is attached to $\bar{f}$ in $N$. Let $\bar{e}$ be mapped to the path $P = (y, \dots, w)$ in $N$ and let $\bar{f}$ be mapped to the path $P' = (x, \dots, y, \dots, z)$ in $N$. Then an *embedding change* of $G$ into $N$ with respect to $\bar{u}$ and $\bar{v}$ is the change of the embedding such that $\bar{e}$ is mapped to the path $(x, \dots, y, \dots, w)$ formed by a subpath of $P'$ and the path $P$, and such that $\bar{f}$ is mapped to the subpath $(y, \dots, z)$ of $P'$. See Figure 6.5 for an example. The definition for h-sprouts is analogous.

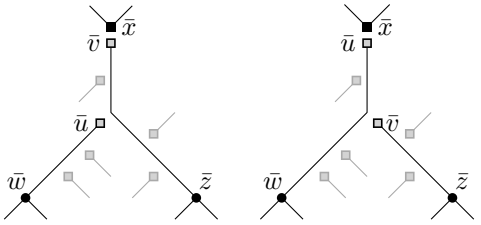


Figure 6.5: An embedding change with respect to $\bar{u}$ and $\bar{v}$.

We now use embedding changes to show that an agreement embedding of $G$ into $N'$ can be changed into an agreement embedding with some nice properties.

**Lemma 6.3.**
*Let $N, N' \in \mathcal{N}_n$ with $r$ and $r' > r$ reticulations, respectively. Let $G$ be a maximum agreement graph for $N$ and $N'$.*
*Then there exist an agreement embedding of $G$ into $N'$ such that*

- *no sprout of an agreement subgraph is attached to a disagreement edge, and*
- *at least one disagreement edge is not attached to any other disagreement edge, and*
- *a disagreement edge $E_i$ of $G$ may only be attached to a disagreement edge $E_j$ of $G$ if $j < i$.*

*Proof.* Fix an agreement embedding of $G$ into $N'$. Assume that this embedding does not fulfill the first property. Then let $\bar{u}$ be a sprout of an agreement subgraph of $G$ that is attached to a disagreement edge $(\bar{v}, \bar{w})$ in $N'$. Without loss of generality, assume that $\bar{u}$ is a t-sprout. Apply an embedding change with respect to $\bar{u}$ and $\bar{v}$. If $\bar{v}$ was attached to another disagreement edge $(\bar{x}, \bar{y})$ in $N'$, then repeat this step with $\bar{u}$ and $\bar{x}$. Otherwise $\bar{u}$ is now attached to a vertex or an edge of an agreement subgraph. This process terminates since the vertex $u$ to which $\bar{u}$ gets mapped gets closer to the root in $N'$ with every step. Note that the embedding change of $\bar{u}$ and $\bar{v}$ may cause a sprout $\bar{z}$ that was previously attached to $(\bar{v}, \bar{w})$ in $N'$ to now be attached to the edge incident to $\bar{u}$ in $N'$. However, since this edge is an edge of an agreement subgraph, this does not negatively effect the first property. Therefore, every sprout that was previously attached to an edge of an agreement subgraph is still so after each step. Hence, the sprouts of agreement subgraphs can be handled one after the other and without negatively affecting property three.

Next, assume that the current embedding fulfills the first, but not the third property. Let $E_i = (\bar{u}, \bar{v})$ be a disagreement edge of $G$, starting with $E_i = E_1$. If $\bar{u}$ and $\bar{v}$ are each attached to a vertex or an edge of an agreement subgraph or to a disagreement edge $E_j$ with $j < i$ in $N'$, then proceed with $E_{i+1}$. Otherwise, without loss of generality, assume that $\bar{u}$ is attached to a disagreement edge $E_j = (\bar{x}, \bar{y})$ with $j > i$. Apply an embedding change with respect to $\bar{u}$ and $\bar{x}$. The same arguments as above show that eventually $\bar{u}$

is attached in $N'$ in a good way. Since the embedding change does not affect a sprout of any $E_m$ with $m < i$ or of an agreement subgraph, this process does not affect the first property or the previously handled disagreement edges. Therefore, the $E_i$'s can be handled one after the other. Apply analogous steps, if necessary, to $\bar{v}$ before proceeding with $E_{i+1}$. The process terminates after $E_i = E_l$ has been handled. Finally, note that the third property implies the second. □

Next, we show how to prune a particular edge of $G$ such that the resulting graph is still an agreement graph for $N$ and $N'$.

**Lemma 6.4.**
*Let $N, N' \in \mathcal{N}_n$. Let $G$ be an agreement graph of $N$ and $N'$. Let $\bar{e} = (\bar{u}, \bar{v})$ be an edge of $G$. Then $G$ can be transformed into a graph $G'$ such that*

- *$\bar{u}$ (or $\bar{v}$) is a sprout in $G'$,*
- *$G'$ contains at most one sprout more than $G$, and*
- *$G'$ is an agreement graph for $N$ and $N'$.*

*Proof.* We prove this for a t-sprout $\bar{u}$. The proof for an h-sprout works analogously. If $\bar{u}$ is already a sprout, then there is nothing to do. If $\bar{u}$ is labelled $\rho$ or $\bar{u}$ has degree three, then obtain $G'$ by pruning the edge $\bar{e}$ at $\bar{u}$. So assume that $\bar{u}$ is an indegree zero, outdegree two vertex. Consider the agreement embedding of $G$ into $N$. Let $\bar{u}$ be mapped to $u$ in $N$. Since $u$ has degree three in $N$, there is an h-sprout $\bar{w}$ mapped to $u$ in $N$. This and the following process are illustrated in Figure 6.6. Then identify $\bar{w}$ with $\bar{u}$; i.e. regraft $\bar{w}$ to $\bar{u}$, and then prune $\bar{e}$ from $\bar{u}$. Let $G''$ be the resulting graph. In the agreement embedding of the resulting graph into $N$, the new sprout $\bar{u}$ is now attached to an edge $\bar{f} = (\bar{x}, \bar{y})$. To get $\bar{w}$ back, restart this case distinction with the goal to prune $\bar{f}$ at $\bar{y}$. Note that this process terminates since the number of degree two vertices in $G''$ is one less than in $G$ and thus at some point one of the first two cases has to apply. Let $G'$ be the resulting graph when the process has terminated. Then $G'$ contains the sprout $\bar{u}$ with incident edge $\bar{e}$ and contains at most one sprout more than $G$. That is because before the case distinction got restarted, the sprout $\bar{w}$ got removed first. Clearly, $G'$ has an agreement embedding into $N$. If one of the first two cases applied, then it is also easy to show that $G'$ has an agreement embedding into $N'$. Otherwise, note that in the agreement embedding of $G$ into $N'$ an h-sprout $\bar{w}'$ is attached to the degree-two vertex $\bar{u}$. For the agreement embedding of $G'$ into $N'$, this sprout $\bar{w}'$ extends the same way as $\bar{w}$ got extended in the embedding into $N$ (see again Figure 6.6).

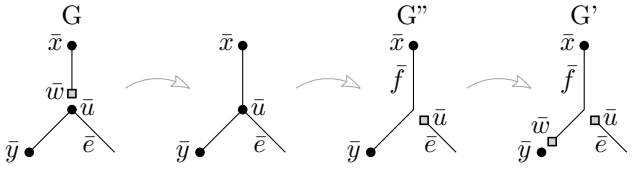


Figure 6.6: For the proof of Lemma 6.4, how to prune the edge $\bar{e}$ at a degree-two vertex $\bar{u}$. First, regraft $\bar{w}$; second, prune $\bar{e}$ at $\bar{u}$; third, reobtain the sprout $\bar{w}$.

□

## 6.2 Agreement distance

In this section we show that maximum agreement graphs induce a metric on $\mathcal{N}_n$. Let $N, N' \in \mathcal{N}_n$ and let $l$ be the difference in number of reticulations of $N$ and $N'$. Let $G$ be a maximum agreement graph for $N$ and $N'$ with $l$ disagreement edges. Let $s$ be the total number of sprouts in the agreement subgraphs of $G$. Then define the *agreement distance* $\mathrm{d_{AD}}$ of $N$ and $N'$ as

$$\mathrm{d_{AD}}(N, N') = s + l.$$

This is well defined since $l$ is fixed by $N$ and $N'$, and since $s$ is minimum over all agreement graphs for $N$ and $N'$ by the choice of $G$.

**Theorem 6.5.**
*The agreement distance* $\mathrm{d_{AD}}$ *on* $\mathcal{N}_n$ *is a metric.*

*Proof.* We have to show that $\mathrm{d_{AD}}$ is symmetric, non-negative, that for all $M, M' \in \mathcal{N}_n$ $\mathrm{d_{AD}}(M, M') = 0$ if and only if $M = M'$, and that $\mathrm{d_{AD}}$ satisfies the triangle inequality. Let $N$, $N'$, and $l$ be as above. First note that the agreement distance is symmetric and non-negative by definition. Second, if $N = N'$, then $G = N$ is a maximum agreement graph for $N$ and $N'$ with zero sprouts and zero disagreement edges and thus $\mathrm{d_{AD}}(N, N) = 0$. Now let $G$ be a maximum agreement graph for $N$ and $N'$ with zero sprouts and zero disagreement edges; i.e. $\mathrm{d_{AD}}(N, N') = 0$. Together with the fact that $N$ and $N'$ are internally binary, this implies that every unlabelled vertex of $N$ and $N'$ gets covered by a degree three vertex of $G$. Thus $G$ has to consist of a single connected component and has an agreement embedding into both $N$ and $N'$ without subdivisions. This in turn implies that $N = G = N'$.

Next, we prove that the agreement distance satisfies the triangle inequality. For this let $N, N', N'' \in \mathcal{N}_n$ with $r$, $r'$, and $r''$ reticulations, respectively. Without loss of generality, assume that $r \leq r''$. Let $G'$ (resp. $G''$) be a maximum agreement graph for $N$ and $N'$ (resp. $N'$ and $N''$) with $s'$ sprouts in its agreement subgraphs and $l'$ disagreement edges (resp. $s''$ and $l''$). For the triangle inequality to hold, we have to show that

$$\mathrm{d_{AD}}(N, N'') \leq d = \mathrm{d_{AD}}(N, N') + \mathrm{d_{AD}}(N', N'') = s' + s'' + l' + l''.$$

For this, we construct an agreement graph $G$ for $N$ and $N''$ with $s$ sprouts in its agreement subgraphs and $l$ disagreement edges such that $s + l \leq d$. Note that $G$ does not have to be a maximum agreement graph. Also note that $l$ is fixed by $N$ and $N''$. The main idea for the construction of $G$ is to merge $G'$ and $G''$ in terms of the prunings they represent in $N$, $N'$ and $N''$. Containing, so to say, sprouts from both $G'$ and $G''$ and the right amount of disagreement edges, finding agreement embeddings of $G$ into $N$ and $N''$ will become easy. We first consider the restricted cases of when $N$, $N'$ and $N''$ either have the same number of reticulations or only differ in the number of reticulations.

**Case I** – $l' = l'' = 0$**.** In this case, by Lemma 6.2 both $G'$ and $G''$ can be obtained from $N'$ by applying $s'$ and $s''$ prunings, respectively. We now apply all these prunings to $N'$ to construct $G$ in the following way. Like in Lemma 6.2, we identify to which edges of $N'$ this prunings correspond and whether they prune at the tail or the head of the edge. Apply the $s'$ prunings of $G'$ to $N'$ to obtain, of course, $G'$. Next, to apply the $s''$ prunings (in $N'$) of $G''$ to $G'$, we have to identify which edges to prune in $G'$.

Assume, without loss of generality, that we want to prune $e = (u, v)$ at $u$ in $N'$. Further assume $G'$ contains an edge $\bar{e} = (\bar{u}, \bar{y})$ such that $\bar{u}$ is mapped to $u$ and $\bar{e}$ to a path containing $e$. With Lemma 6.4 prune $\bar{e}$ at $\bar{u}$ and obtain a graph $\bar{G}$. Note that $\bar{G}$ has an agreement embedding into $N$ and $N'$. Next, assume $G'$ contains an edge $\bar{e}' = (\bar{x}, \bar{y})$ such that $\bar{e}'$ is

mapped to a path containing $e$ and $u$ as internal vertex. Then, $G'$ contains a sprout $\bar{w}$ that is mapped to $u$ (and thus attached to $\bar{e}'$ in $N'$). If $\bar{w}$ is an h-sprout, prune $\bar{e}'$ at $\bar{x}$ with Lemma 6.4 and obtain a graph $\bar{G}$. Note that $\bar{G}$ has an agreement embedding into $N$ and $N'$. So assume otherwise, namely that $\bar{w}$ is a t-sprout. Let $\bar{w}$ have the incident edge $(\bar{w}, \bar{z})$. Subdivide $\bar{e}'$ with a new vertex $\bar{u}$ and identify $\bar{w}$ with $\bar{u}$. Prune $\bar{e} = (\bar{u}, \bar{y})$ at $\bar{u}$ and then use Lemma 6.4 to prune $(\bar{x}, \bar{z})$ at $\bar{x}$ to reobtain $\bar{w}$. Let $\bar{G}$ be the resulting graph. Note that $\bar{G}$ has an agreement embedding into $N'$. Furthermore, apply an embedding change with respect to $\bar{u}$ and $\bar{w}$ to see that $G$ still has an agreement embedding into $N$. Repeat this process (now using $\bar{G}$ instead of $G'$) for each of the $s''$ sprouts of $G''$. Let $G$ be the resulting graph, which by construction has an agreement embedding into $N'$ and $N$. Furthermore, $G$ has at most $s \leq s' + s''$ sprouts.

Lastly, we have to show that $G$ has an agreement embedding into $N''$. Consider the agreement embeddings of $G$ and $G''$ into $N'$. Let $\bar{u}$ be a sprout of $G$ obtained for a sprout $\bar{u}''$ of $G''$. If $\bar{u}$ and $\bar{u}''$ are mapped to the same vertex $u$ of $N''$, then it is straightforward to handle $\bar{u}$ when obtaining the agreement embedding of $G$ into $N''$. On the other hand, $\bar{u}$ could "reach beyond" $u$, that is, its incident edge is mapped to a path containing $u$ as internal vertex. This case might be reduced to the former with an embedding change of $G$ into $N'$. Otherwise, we know that $\bar{u}''$ is attached to a degree two vertex $\bar{x}''$ in $N'$. Furthermore, there is then also a sprout $\bar{w}''$ of $G''$ that is attached to $\bar{x}$ in the agreement embedding of $G''$ into $N''$. Let $\bar{w}$ be the sprout of $G$ obtained for the sprout $\bar{w}''$. Using the agreement embedding of $G''$ into $N''$ to obtain the agreement embedding of $G$ into $N''$, we then let the sprout $\bar{w}$ "reach beyond" $\bar{x}''$ in the same way as $\bar{u}$ does in the agreement embedding of $G$ into $N'$ (see also Figure 6.6). To conclude, note that with $s + l = s \leq s' + s'' = s' + s'' + l' + l''$ the triangle inequality holds in this case.

**Case II.a** – $s' = s'' = 0$ **and** $r < r' < r''$**.** In this case, $N'$ can be seen as $N$ plus $l'$ reticulation edges and $N''$ can be seen as $N'$ plus $l''$ reticulation edges. Thus, $N''$ can also be seen as $N$ plus $l' + l''$ reticulation edges. Therefore $G$ consisting of $N$ and $l = l' + l''$ disagreement edges is a desired agreement graph for $N$ and $N''$ showing that the triangle inequality holds in this case.

**Case II.b** – $s' = s'' = 0$ **and** $r < r' > r''$**.** Fix agreement embeddings of $G'$ and $G''$ into $N'$. Colour all edges to which a disagreement edge of $G'$ is mapped orange and to which a disagreement edge of $G''$ is mapped green. Intuitively, edges that are now both green and orange in $N'$ are neither in $N$ nor in $N''$. We now align the agreement embeddings of $G'$ (and $G''$) such that a disagreement edge is mapped to either edges that are all orange or all green-orange (resp. all green or all green-orange). Note that a disagreement edge is mapped to a path that starts at a tree vertex and ends at a reticulation. Furthermore, if such a path contains an internal vertex $v$, then the sprout of another disagreement edge is mapped to $v$. Therefore, to align the agreement embeddings as described above, we can apply a sequence of simple embedding changes to the sprouts of disagreement edges as illustrated in Figure 6.7 (i) and (ii) (the rules for h-sprouts and swapped colours are analogous). We can further align those disagreement edges of $G'$ and $G''$ that are mapped to green-orange edges with rule (iii) in Figure 6.7. Now let $k'$ be the number disagreement edges of $G'$ (and thus also of $G''$) that are mapped to green-orange edges.

Obtain a new $N'$ from $N'$ by removing all green-orange edges from $N'$, obtain new $G'$ and $G''$ from $G'$ and $G''$ by removing $k'$ disagreement edges. Note that $G''$ has now $k = l'' - k'$ disagreement edges. Clearly, $G'$ (resp. $G''$) has still an agreement embedding into $N$ and $N'$ (resp. $N'$ and $N''$). Then, in $N'$, if a vertex is incident to an uncoloured edge $e$, an orange edge, and a green edge, then colour $e$ red. Such a colouring is illustrated in Figure 6.8. Next and as long as possible, while a vertex is incident to an uncoloured edge $e$, a red edge and a green or orange edge, colour $e$ red. Obtain $S$ from $N'$ by removing all coloured edges

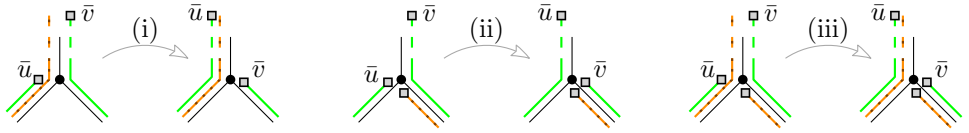


Figure 6.7: For Case II.b, embedding changes of $G''$ (green) into $N'$ with respect to $\bar{u}$ and $\bar{v}$ to align the embeddings of disagreement edges of $G'$ (orange with dots) and $G''$ in $N'$.

and suppressing indegree one, outdegree one vertices. Removing the red edges prevents $S$ from having sprouts. Let $G$ be the graph consisting of $S$ and $l$ disagreement edges and $k = l'' - k'$ connected components $F_i$ consisting of a single directed edge. We claim that $G$ is an agreement graph for $N$ and $N''$.

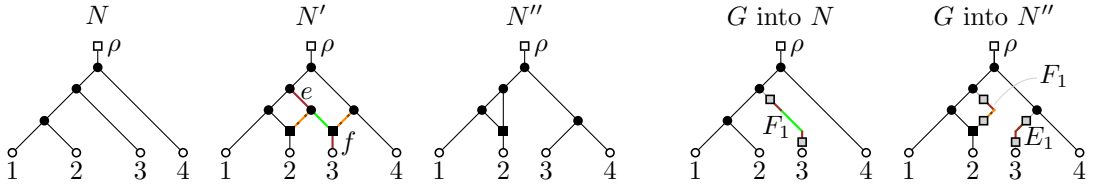


Figure 6.8: For Case II.b, $N$ (resp. $N''$) can be obtained from $N'$ by removing the orange (with dots) (resp. green) edges. Embedding $G$ into $N$, the agreement subgraph $F_1$ has to cover not only the green edge, but also the red edges ($e$ and $f$), which got removed from $N'$ when obtaining $G$ because a disagreement edge of both $N$ and of $N''$ were incident to them.

We construct an agreement embedding of $G$ into $N$. The embedding of $S$ into $N$ is given by the embeddings of $S$ and $N$ into $N'$. Let $E_i$ be a disagreement edge of $G''$. Let $P$ be the green path in $N'$ that corresponds to $E_i$. If an edge of $P$ caused the creation of a red edge $e$, extend $P$ by $e$ if possible; that is, if $P$ would still be a directed path. Next and as long as possible, if $e$ caused another red edge $e'$, extend $P$ by $e'$ if possible. Then embed an $F_i$ into $N$ in the way that $P$ is embedded onto $N$ in the embedding of $N$ into $N'$. The colours of the edges ensure that this is possible. See again Figure 6.8 for an example. Furthermore, note that this construction eventually covers all green and red edges. Hence, we constructed an agreement embedding of $G$ into $N$. Finding an agreement embedding of $G$ into $N''$ works analogously but also uses the disagreement edges of $G$ besides the $F_i$. Since $l = l' - l''$, we get $s + l = 2k + l =\leq 2l'' + l = l' + l''$, and thus the triangle inequality also holds in this case.

**Case II.c** – $s' = s'' = 0$ **and** $r > r' < r''$**.** In this case, $N$ and $N''$ can be obtained from $N'$ by adding $l'$ and $l'' = l + l'$ reticulation edges, respectively. Consequently, $N'$ together with $l$ disagreement edges and $l'$ further connected components that consists of a single directed edge gives an agreement graph for $N$ and $N''$. Since $l = l'' - l'$, we get $s + l = 2l' + l = l' + l''$, and thus the triangle inequality also holds in this case.

**Case III.a** – $r \leq r' \leq r''$**.** Assume agreement embeddings of $G'$ and $G''$ with nice properties as in Lemma 6.3. We now combine Case I and Case II.b to obtain $G$. Let $H$ be the graph $G''$ without its disagreement edges. Note that $H$ has an agreement embedding into $N'$ and has $s''$ sprouts. Like in Case I, obtain a graph $R$ from $H$ by applying $s'$ prunings in the way the $s'$ sprouts of $G'$ are attached to vertices in $N'$. Note that $R$ has an agreement embedding into $N'$ and has at most $s' + s''$ sprouts. Then like in Case II.b, obtain a graph $S$ from $R$ by removing all paths from $R$ to which disagreement edges of $G'$ are mapped. Again, handle conflicts between a sprout of a disagreement edge of $G'$

and a sprout of $R$ like the red edges in Case II.b. Now let $G$ be the graph consisting of $S$ and $l = l' + l''$ disagreement edges. Note that $S$ and thus $G$ have at most $s' + s''$ sprouts (ignoring those in the disagreement edges). Hence, $s + l \leq d$. Constructing agreement embeddings of $G$ works again by combining the mechanisms from Case I and Case II.b.

The two cases for when $r \leq r' \geq r''$ and $r \geq r' \leq r''$ can be handled similarly to Case III.a together with the ideas from Case II.b and Case II.c. We give a brief outline of how $G$ can be constructed.

**Case III.b** – $r \leq r' \geq r''$**.** Let $S$ be the graph obtained from $N'$ by removing all paths to which the disagreement edges of $G'$ and $G''$ are mapped (like in Case II.b) and by applying the prunings of $G'$ and $G''$ in the way they embed into $N'$ (like in Case I). Again, in this process we have to take care of cases where two sprouts are mapped to the same vertex. Then the graph $G$ consisting of $S$ and $k \leq l''$ additional directed edges and $l$ disagreement edges is an agreement graph for $N$ and $N''$ with at most $s' + s'' + 2l''$ sprouts in agreement subgraphs and $l = l' - l''$ disagreement edges. Hence, $s + l \leq d$.

**Case III.c** – $r \geq r' \leq r''$**.** Let $S$ be the graph obtained from $N'$ by applying the prunings of $G'$ and $G''$ in the way they embed into $N'$ (like in Case I). Then the graph $G$ consisting of $S$ and $l'$ additional directed edges and $l$ disagreement edges is an agreement graph for $N$ and $N''$ with at most $s' + s'' + 2l'$ sprouts in agreement subgraphs and $l = l'' - l'$ disagreement edges. Hence, $s + l \leq d$.

This concludes the proof. □

## 6.3 Relation to rearrangement distances

In this section we look at the relation of the agreement distance to the SPR-, SNPR-, and PR-distance. We distinguish the cases of distances between two trees, between a tree and a network, and between two networks.

### 6.3.1 Tree to tree

First we prove that the agreement distance, if restricted to $\mathcal{T}_n$, equals the SPR-distance. As a consequence, we get that the agreement distance is NP-hard to compute.

**Proposition 6.6.**
*The agreement distance on $\mathcal{T}_n$ is equivalent to the* SPR*-distance.*

*Proof.* Let $T, T' \in \mathcal{T}_n$. Let $G$ be a maximum agreement graph for $T$ and $T'$ with components $S_1, \ldots, S_m$. We distinguish whether $G$ contains an h-sprout or not.

Assume $G$ does not contain an h-sprout. Then $G$ is a maximum agreement forest for $T$ and $T'$. Therefore, $\mathrm{d_{AD}}(T, T') = m - 1$, that is, it equals the number of components of $G$ minus one. Furthermore, by removing sprouts and their incident edges from $G$ we obtain a forest $F$ that is a maximum agreement forest for $T$ and $T'$ under the definition of Bordewich and Semple [BS05]. Hence, the statement follows from Theorem 2.1 by Bordewich and Semple [BS05].

Now assume $G$ contains $k$ h-sprouts. We now show how to derive a maximum agreement graph $G'$ for $T$ and $T'$ without h-sprouts. Assume that $G$ contains an h-sprout $\bar{u}$ that is a child of a degree two vertex $\bar{v}$. Note that in the agreement embedding of $G$ into $T$ and $T'$ there is another h-sprout attached to $\bar{v}$. Thus, deleting $(\bar{u}, \bar{v})$ from $G$ creates a new t-sprout $\bar{v}$ such that $G$ is still a maximum agreement graph for $T$ and $T'$ (see Figure 6.9 (a)). So assume that $G$ contains no such h-sprout. Hence, $G$ contains $k$ h-sprouts that are adjacent to degree three vertices, to $\rho$ or a t-sprout. Then since a tree does not contain reticulations, note that $G$ also contains $k$ vertices with indegree zero but outdegree either

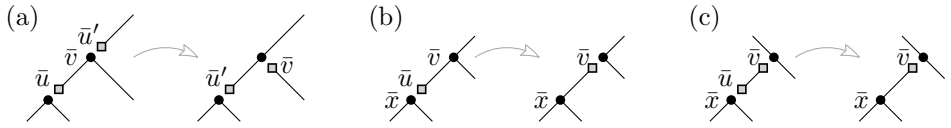


Figure 6.9: How to convert h-sprouts from a maximum agreement graph $G$ for two trees to t-sprouts for Proposition 6.6, when the h-sprout $\bar{u}$ is child of (a) a degree two vertex, (b) a degree three vertex, or (c) a t-sprout, respectively.

zero (a labelled leaf of $T$) or two. That is because in the agreement embedding of $G$ into $T$ and $T'$ the $k$ h-sprouts have to get mapped to such $k$ vertices. Let $M$ be the set of those vertices. Now, firstly, remove from $G$ the $k$ h-sprouts and their incident edges and suppress resulting degree two vertices. If this results in an unlabelled, isolated vertex, remove it too. This does not create any new sprouts since by assumption no h-sprout was incident to a degree two vertex. Secondly, add $k$ edges connecting each vertex in $M$ with a new t-sprout (see Figure 6.9 (b) and (c)). Let $G'$ be the resulting graph. Note that $G'$ contains either the same number of sprouts as $G$ or less if an h-sprout was adjacent to a t-sprout in $G$. (Note that if the latter case applies, then $G$ was actually not a maximum agreement graph.) Figure 6.9 also shows how to derive agreement embeddings of $G'$ into $T$ and $T'$ from the agreement embeddings of $G$. Hence $G'$ is a maximum agreement graph for $T$ and $T'$ without h-sprouts and the claim follows from the previous case. □

Let $G$ be a maximum agreement graph of $T, T' \in \mathcal{T}_n$. From the proof of Proposition 6.6 we learn that we may assume that $G$ contains no h-sprout. Therefore, each agreement subgraph of $G$ contains at most one sprout and the agreement subgraph containing the vertex labelled $\rho$ contains no sprout. Hence, $\mathrm{d_{AD}}(T, T')$ equals the number of components of $G$ minus one and thus also $\mathrm{m}(T, T')$.

Bordewich and Semple [BS05, Theorem 2.2] have shown that computing the SPR-distance of two phylogenetic trees is NP-hard. Together with Proposition 6.6 this implies the following corollary.

**Corollary 6.7.**
*Computing the agreement distance of an arbitrary pair of networks in $\mathcal{N}_n$ is NP-hard.*

### 6.3.2 Tree to network

Building on the results in Section 5.1 we now look at the distances of a phylogenetic tree $T$ and a phylogenetic network $N$. We start with the case when $T$ is displayed by $N$.

**Lemma 6.8.**
*Let $N \in \mathcal{N}_n$ with $r$ reticulations. Let $T \in D(N)$. Then*

$$\mathrm{d_{SNPR}}(T, N) = \mathrm{d_{PR}}(T, N) = \mathrm{d_{AD}}(T, N) = r.$$

*Proof.* By Lemma 5.1 and Corollary 5.2, we have that $\mathrm{d_{SNPR}}(T, N) = \mathrm{d_{PR}}(T, N) = r$ and know that there exists a $\mathrm{PR}^+$-sequence $\sigma = (T = N_0, N_1, \ldots, N_r = N)$ that transforms $T$ into $N$. Using $\sigma$, we now prove that $G = \{T = T_\rho, E_1, \ldots, E_r\}$ is an agreement graph for $T$ and $N$. The proof is by induction on $r$. If $r = 0$, then $T = N$ and the claim trivially holds. Next, let $e$ be the edge added from $N_{i-1}$ to $N_i$ for $i = \{1, \ldots, r\}$. Note that $G_{i-1} = \{T, E_1, \ldots, E_{i-1}\}$ has an agreement embedding into $N_i$. Extending this embedding by mapping $E_i$ of $G_i = \{T, E_1, \ldots, E_i\}$ to $e$, we get that $G_i$ is an agreement graph of $T$ and $N_i$. This is illustrated in Figure 6.10. Therefore, $G$ is an agreement graph

for $T$ and $N$ with $r$ disagreement edges and no sprouts in its agreement subgraph. Hence

$$r = \mathrm{d_{PR}}(T, N) \geq \mathrm{d_{AD}}(T, N).$$

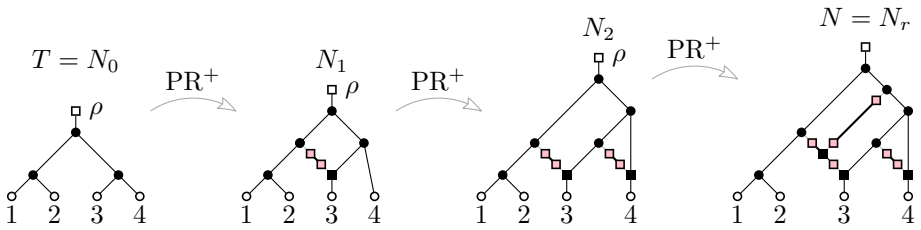


Figure 6.10: An example of how to obtain an agreement embedding into $N$ of an agreement graph $G = \{T, E_1, \ldots, E_r\}$ for $T$ and $N$ for the proof of Lemma 6.8.

To establish the other direction, let $G$ be a maximum agreement forest for $N$ and $T$. Recall that, by definition, $G$ contains $r$ disagreement edges and at least one agreement subgraph. Thus,

$$\mathrm{d_{AD}}(T, N) \geq r = \mathrm{d_{PR}}(T, N).$$

This completes the proof of the lemma. □

In Theorem 5.10 we characterised the distance of $T$ and $N$ in terms of the distance of $T$ to the trees displayed by $N$. We now use this characterisation to establish the following result.

**Theorem 6.9.**
*Let $T \in \mathcal{T}_n$, $N \in \mathcal{N}_n$. Then*

$$\mathrm{d_{SNPR}}(T, N) = \mathrm{d_{PR}}(T, N) = \mathrm{d_{AD}}(T, N).$$

*Proof.* Let $r$ be the number of reticulations in $N$. Note that $\mathrm{d_{SNPR}}(T, N) = \mathrm{d_{PR}}(T, N)$ by Theorem 5.10. We first show that $\mathrm{d_{AD}}(T, N) \leq \mathrm{d_{PR}}(T, N)$. By Theorem 5.10, there exists a phylogenetic tree $T'$ that is displayed by $N$ such that

$$\mathrm{d_{PR}}(T, N) = \mathrm{d_{PR}}(T, T') + \mathrm{d_{PR}}(T', N) = \mathrm{d_{PR}}(T, T') + r.$$

Hence, we have $\mathrm{d_{AD}}(T, T') = \mathrm{d_{PR}}(T, T') = \mathrm{d_{PR}}(T, N) - r$, where the first equality follows from Theorem 5.4 and Proposition 6.6. Moreover, by Lemma 6.8, we have $\mathrm{d_{AD}}(T', N) = \mathrm{d_{PR}}(T', N) = r$. Let $G'$ be a maximum agreement graph for $T$ and $T'$, and let $G''$ be a maximum agreement graph for $T'$ and $N$. We know by Lemma 6.8 that such an $G''$ exists and that $T' \in G''$. Now, let

$$G = G' \cup (G'' - \{T'\}).$$

Since $G'$ has an agreement embedding into $T'$ and since $T'$ has an agreement embedding into $N$, we get an agreement embedding of $G'$ into $N$. This embedding covers all edges of $N$, except those to which the disagreement edges of $G''$ get mapped. Since $G$ contains both $G'$ and the disagreement edges of $G''$, it follows that $G$ is an agreement graph for $T$ and $N$. Note that $G'$ has $\mathrm{d_{PR}}(T, T')$ sprouts in agreement subgraphs but no disagreement edges and that $G''$ has no agreement subgraphs but $r = \mathrm{d_{PR}}(T', N)$ disagreement edges. Hence,

$$\mathrm{d_{AD}}(T, N) \leq \mathrm{d_{PR}}(T, T') + \mathrm{d_{PR}}(T', N) = \mathrm{d_{PR}}(T, N).$$

We next show that $\mathrm{d_{PR}}(T, N) \leq \mathrm{d_{AD}}(T, N)$. Let $G = \{S_1, S_2, \ldots, S_k, E_1, , E_2, \ldots, E_r\}$ be a maximum agreement graph for $T$ and $N$. Assume for now that the agreement subgraphs of $G$ contain no h-sprout. The proof is by induction on $d = \mathrm{d_{AD}}(T, N)$. If $d = 0$, then $G = T = N$ and thus $\mathrm{d_{PR}}(T, N) = 0$. Now assume that inequality holds for all pairs of a phylogenetic tree and a phylogenetic network with agreement distance at most $d - 1$. If $r = 0$, then $N$ is a phylogenetic tree and $G = \{S_1, S_2, \ldots, S_k\}$. Then it follows from Lemma 6.8 that $\mathrm{d_{PR}}(T, N) = \mathrm{d_{AD}}(T, N)$.

We may therefore assume that $r > 0$. By Lemma 6.3 we can assume that no sprout of an agreement subgraph of $G$ is attached to a disagreement edge $E_i$ and that no disagreement edge is attached to $E_r = (\bar{u}, \bar{v})$. Then $E_r$ is mapped to an edge $e = (u, v)$ of $N$. Since the agreement subgraphs do not contain h-sprouts, it follows that the h-sprouts of the disagreement edges are mapped to the $r$ reticulations of $N$. Therefore, the edge $e$ is a reticulation edge. Furthermore, $u$ is an inner tree vertex, since neither the agreement subgraphs $S_i$ nor the disagreement edges contain any vertices with indegree two. Let $G' = G \setminus \{E_R\}$ and let $N'$ be the network obtained from $N$ by deleting $(u, v)$ and suppressing resulting degree two vertices. Then $G'$ is a maximum agreement graph for $T$ and $N'$. Since $\mathrm{d_{AD}}(T, N') < d$, it follows from the induction hypothesis that $\mathrm{d_{PR}}(T, N') \leq \mathrm{d_{AD}}(T, N')$. Furthermore, by construction, $N$ can be obtained from $N'$ by a single $\mathrm{PR}^+$. Taken together, this implies that

$$\mathrm{d_{PR}}(T, N) \leq \mathrm{d_{PR}}(T, N') + 1 \leq \mathrm{d_{AD}}(T, N') + 1 = \mathrm{d_{AD}}(T, N).$$

Now assume that an agreement subgraph of $G$ contains an h-sprout. We show how to derive a maximum agreement graph $G'$ of $T$ and $N'$ without this property, so that we can apply the previous case. By Lemma 6.3 we can assume again that no sprout of an agreement subgraph is attached to a disagreement edge. If the $r$ h-sprouts of disagreement edges of $G$ are mapped to reticulations of $N$, then the embedding of $G$ without its disagreement edges into $N$ is a tree. Then we obtain $G'$ from $G$ as in the proof of Proposition 6.6. Otherwise, there is an h-sprout of an agreement subgraph mapped to a reticulation of $N$.

We obtain $G'$ from $G$ via the following three steps. First, if an h-sprout is child of a degree two vertex, then apply the change shown in Figure 6.9 (a) like in the proof of Proposition 6.6 to reduce the number of h-sprouts by one. Repeat this procedure as long as this case applies. Recall that this maintains the agreement embeddings into $T$ and $N$. Every h-sprout of an agreement subgraph is now the child of a degree three vertex, the root, or a t-sprout. Therefore, in the second step, it is possible to prune all $m$ edges incident to h-sprouts of agreement subgraphs at their tail. If an h-sprout is adjacent to a t-sprout, then this has no effect. Let $\bar{w}$ be a vertex that is mapped to the same vertex $v$ as an h-sprout in $T$. Note that such $v$ is either a tree vertex or a leaf Then after the second step there is an h-sprout that is incident to a t-sprout and that is mapped to $v$, both in the agreement embedding into $T$ and into $N$. Similarly, for each reticulation $x$ of $N$, there is an h-sprout that is incident to a t-sprout and that is mapped to $x$. We can thus change the naming of components such that the disagreement edges are mapped to the $r$ components whose h-sprout is mapped to a reticulation. Hence, in the third step, merging all pairs consisting of an h-sprout $\bar{v}$ and a vertex $\bar{w}$ that are mapped to the same vertex in $T$ yields an agreement embedding of the resulting graph $G'$ into $T$ and $N$. Note that this step regrafts at least $m$ edges. Therefore, $G'$ contains at most as many sprouts as $G$ and its only h-sprouts belong to disagreement edges. Hence, $G'$ is a maximum agreement graph for $T$ and $N$ as required. This concludes the proof. □

### 6.3.3 Network to network

After we have shown that the agreement distance equals the PR-distance on $\mathcal{T}_n$ and the PR-distance of a tree and a network, we now consider its relation to the PR- and SNPR-distance on $\mathcal{N}_n$. We start on a positive note concerning the neighbourhoods of a phylogenetic network under PR and the agreement distance.

**Lemma 6.10.**
*Let $N, N' \in \mathcal{N}_n$. Then $\mathrm{d_{AD}}(N, N') = 1$ if and only if $\mathrm{d_{PR}}(N, N') = 1$.*

*Proof.* Assume $\mathrm{d_{PR}}(N, N') = 1$. Depending on whether $N'$ can be obtained from $N$ by applying a $\mathrm{PR}^0$ or a $\mathrm{PR}^+$ operation, obtain a maximum agreement graph $G$ by either mimicking the pruning or adding a disagreement edge to $N$. In either case, it follows that $\mathrm{d_{AD}}(N, N') = 1$.

Now assume $\mathrm{d_{AD}}(N, N') = 1$ and let $G$ be a maximum agreement graph for $N$ and $N'$. If $G$ contains a disagreement edge, then it is easy to see that $\mathrm{d_{PR}}(N, N') = 1$. So assume $G$ contains a single sprout $\bar{u}$. If $\bar{u}$ is attached to a vertex $\bar{x}$ of $G$ in the agreement embedding into $N$, then it has to be attached to $\bar{x}$ also in the agreement embedding into $N'$. However, then $N = N'$, which is a contradiction to $\mathrm{d_{AD}}(N, N') = 1$. If, on the other hand, $\bar{u}$ is attached to an edge of $G$ in the agreement embedding into $N$ (and thus into $N'$), then finding a $\mathrm{PR}^0$ that transforms $N$ into $N'$ is straightforward. It follows that $\mathrm{d_{PR}}(N, N') = 1$. □

Consider the two networks $N$ and $N'$ shown in Figure 6.11. Observe that $\mathrm{d_{PR}}(N, N') = 3$ (which can be shown with an exhaustive search), but that $\mathrm{d_{AD}}(N, N') = 2$. Intuitively, the differences arises from the fact that no $\mathrm{PR}^0$ can prune, from $N$ or $N'$, any of the two sprouts of the shown maximum agreement graph $G$ and regraft it without creating a directed cycle. Nor is there a shortest PR-sequence of length two that uses $\mathrm{PR}^+$ and $\mathrm{PR}^-$ operations. This shows that, in general, the agreement distance and the PR-distance differ on $\mathcal{N}_n$. Since allowing only tail $\mathrm{PR}^0$ (like SNPR) or not allowing parallel edges increases the distance in general, it follows that the agreement distances also differs from the SNPR-distance and distances of other generalisations of SPR. Furthermore, by Lemma 5.14 there exist pairs of phylogenetic networks with $r \geq 1$ reticulations for which every shortest PR- or SNPR-sequence contains a phylogenetic tree. This implies that along such a sequence reticulation edges get removed and added again. Therefore, and even if the PR-distance (or SNPR-distance) and the agreement distance would be the same for such a pair, an agreement graph can in general not fully model every shortest PR- and SNPR-sequence. On the upside, however, we prove now that the agreement distance gives a lower and upper bound for the PR-distance with constant factors. We start with the lower bound.

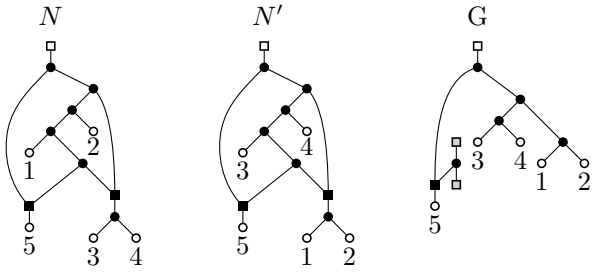


Figure 6.11: Two phylogenetic networks $N$ and $N'$ with $\mathrm{d_{PR}}(N, N') = 3$, but $\mathrm{d_{AD}}(N, N') = 2$ as the maximum agreement graph $G$ shows.

**Theorem 6.11.**
*Let $N, N' \in \mathcal{N}_n$. Then $\mathrm{d_{AD}}(N, N') \leq \mathrm{d_{PR}}(N, N')$.*

*Proof.* Given $N$ and $N'$ with PR-distance $d = \mathrm{d_{PR}}(N, N')$, we construct an agreement graph $G$ of $N$ and $N'$ with $s$ sprouts in the agreement subgraphs and $l$ disagreement edges such that $s + l \leq d$. Let $N$ and $N'$ have $r$ and $r'$ reticulations, respectively. Without loss of generality, assume that $r' \geq r$ and let $l = r' - r$. The proof is now by induction on $d$. If $d = 0$, then $G = N$ is as desired. If $d = 1$, the statement follows from Lemma 6.10. Now assume that for each pair of phylogenetic networks $M, M' \in \mathcal{N}_n$ with PR-distance at most $d' < d$ for some arbitrary but fixed $d > 1$ there exists an agreement graph of $M$ and $M'$ proving that $\mathrm{d_{AD}}(M, M') \leq d'$.

Fix a PR-sequence of length $d$ from $N$ to $N'$. Let $N'' \in \mathcal{N}_n$ be the network of that sequence such that $\mathrm{d_{PR}}(N, N'') = d-1$ and $\mathrm{d_{PR}}(N'', N') = 1$. By the induction hypothesis there exists an agreement graph $G'$ for $N$ and $N''$ showing that $\mathrm{d_{AD}}(N, N'') \leq d - 1$. We distinguish whether $N'$ is obtained from $N''$ by a $\mathrm{PR}^0$, a $\mathrm{PR}^+$, or a $\mathrm{PR}^-$ operation.

First, assume that $N'$ can be obtained from $N''$ by pruning the edge $e = (u, v)$ at $u$. Assume $G'$ contains an edge $\bar{e} = (\bar{u}, \bar{y})$ such that $\bar{u}$ is mapped to $u$ and $\bar{e}$ to a path containing $e$. With Lemma 6.4 prune $\bar{e}$ at $\bar{u}$ and obtain $G$. Then use the agreement embedding of $G$ into $N''$ to obtain an agreement embedding of $G$ into $N'$. Next, assume $G'$ contains an edge $\bar{e}' = (\bar{x}, \bar{y})$ such that $\bar{e}'$ is mapped to a path containing $e$ and $u$ as internal vertex. Then, $G'$ contains a t-sprout $\bar{w}$ that is mapped to $u$ (and thus attached to $\bar{e}'$ in $N''$). The vertex $\bar{w}$ cannot be an h-sprout, because $u$ is a tree vertex and the previous case does not apply. Let $\bar{w}$ have the incident edge $(\bar{w}, \bar{z})$. Subdivide $\bar{e}'$ with a new vertex $\bar{u}$ and identify $\bar{w}$ with $\bar{u}$. Prune $\bar{e} = (\bar{u}, \bar{y})$ at $\bar{u}$ and then use Lemma 6.4 to prune $(\bar{x}, \bar{z})$ at $\bar{x}$ to reobtain $\bar{w}$. Let $G$ be the resulting graph, which has now an agreement embedding into $N'$. Considering the embedding of $G$ into $N''$, apply an embedding change with respect to $\bar{u}$ and $\bar{w}$ to see that $G$ still has an agreement embedding into $N$. In either case, since $G$ contains at most one sprout more than $G'$, it follows that $\mathrm{d_{AD}}(N, N') \leq \mathrm{d_{AD}}(N, N'') + 1 \leq d$. The case where $N'$ is obtained from $N''$ by pruning an h-sprout works analogously.

Second, assume that $N'$ has been obtained from $N''$ by a $\mathrm{PR}^-$ that removed the edge $e = (u, v)$. Note that then $G'$ contains $l + 1$ disagreement edges. Assume $G'$ contains a disagreement edge $E_j = (\bar{x}, \bar{y})$ that maps to a path $P$ that contains $e$ in the agreement embedding of $G'$ into $N''$. Note that $u$ is a tree vertex and $v$ a reticulation. Therefore, if $P$ contains $u$ as internal vertex, then a t-sprout $\bar{w}$ is attached to $E_j$ in $N''$ and is mapped to $u$. Apply an embedding change with regards to $\bar{w}$ and $\bar{x}$. Handle the case where $P$ contains $v$ as internal vertex analogously. Then $E_j$ is mapped precisely to $e$. Hence, obtain $G$ from $G'$ by removing $E_j$. The agreement embedding of $G$ into $N$ is then the same as of $G'$ and the agreement embedding of $G$ into $N'$ is derived from that of $G'$ into $N''$ by removing $E_j$. Now assume that $e$ is not covered by a disagreement edge of $G'$. Let $\bar{e} = (\bar{x}, \bar{y})$ be the edge of $G'$ that covers $e$. With Lemma 6.4 prune $\bar{e}$ at $\bar{x}$ and $\bar{y}$ such that the resulting graph $G''$ has at most two sprouts more than $G'$ and an agreement embedding into both $N$ and $N''$. Consider $\bar{e}$ now a disagreement edge of $G''$ and consider a disagreement edge of $G''$ an agreement subgraph. Then apply the previous case to obtain $G$. In either case, $G$ contains one disagreement edge less and at most two sprouts more in its agreement subgraphs and therefore $\mathrm{d_{AD}}(N, N') \leq \mathrm{d_{AD}}(N, N'') + 2 - 1 \leq d$.

Lastly, assume $N'$ has been obtained from $N''$ by a $\mathrm{PR}^+$. If $l > 0$, obtain $G$ from $G'$ by adding one disagreement edge. If $l = 0$, then $G'$ contains one disagreement edge. Thus obtain $G$ from $G'$ by considering this disagreement edge an agreement subgraph. In either case, it is straightforward to find agreement embeddings of $G$ into $N$ and $N'$. Since $G$ contains either one disagreement edge more or two sprouts more but one disagreement

edge less, it follows again that $\mathrm{d_{AD}}(N, N') \leq d$. This completes the proof. □

Let $N, N' \in \mathcal{N}_n$ with a maximum agreement graph $G = (V_G, E_G)$. Fix agreement embeddings of $G$ into $N$ and $N'$ and assume that they fulfill the properties of Lemma 6.3. In the proof of the upper bound we will construct a PR-sequence based on agreement embeddings of $G$ along this sequence. To ease talking about PR operations on networks along the sequence based on vertices and edges of $G$ we define the following terminology. Let $\bar{u} \in V_G$ be a t-sprout with outgoing edge $\bar{e} = (\bar{u}, \bar{v}) \in E_G$. Let $e = (u, v)$ be the first edge on the path in $N$ to which $\bar{e}$ is mapped. *Pruning* $\bar{u}$ in $N$ then means that the edge $e$ gets pruned at $u$. *Regrafting* $\bar{u}$ to an edge $\bar{f} \in E_G$ in $N$ then means that $e$ gets regrafted to the edge $f \in E_N$ that is the first edge on the path to which $\bar{f}$ is mapped. Let $\bar{x}$ be a indegree two, outdegree zero vertex or the singleton labelled $\rho$ of $G$. *Regrafting* $\bar{u}$ to a vertex $\bar{x} \in V_G$ in $N$ then means that $e$ gets regrafted to the edge $f \in E_N$ that is the outgoing edge of the vertex $x$ to which $\bar{x}$ is mapped. The terminology for h-sprouts is analogously defined. More precisely, the differences for an h-sprout $\bar{u}$ are that the edge $\bar{e}$ is the incoming edge of $\bar{u}$, and that $f$ is the last edge of the respective path to which $\bar{f}$ is mapped or the incoming edge of the tree vertex $x$.

We say a sprout $\bar{u}$ is *prunable* (with respect to $N$) if it is attached to an edge $\bar{e}$ in $N$ and *unprunable* if it is attached to a vertex $\bar{x}$ in $N$. Let $\bar{u}$ be a sprout that is attached to an edge $\bar{f}$ (or vertex $\bar{x}$) in $N'$. We say the sprout $\bar{u}$ is *blocked* if regrafting it to $\bar{f}$ (or $\bar{x}$) in $N$ would create a directed cycle; otherwise we call it *unblocked*. This implies that there is at least one sprout $\bar{v} \in V_G$ on the path from $\bar{u}$ to $\bar{f}$ (or $\bar{x}$) in the embedding of $G$ into $N$. We call such a sprout $\bar{v}$ *blocking*. See Figure 6.12 (a) and (b) for examples.

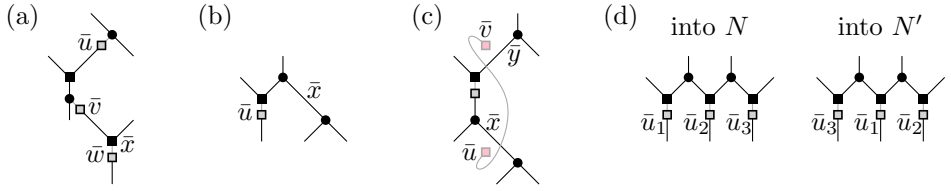


Figure 6.12: Embeddings of $G$ into $N$ (and $N'$ in (d)). In (a), the sprout $\bar{u}$ is prunable, but blocked by the blocking sprout $\bar{v}$ if $\bar{u}$ is supposed to take the place of $\bar{w}$. In (b), $\bar{u}$ is unprunable, but unblocked. In (c), the disagreement edge $(\bar{u}, \bar{v})$ is not addable since $\bar{y}$ is ancestor of $\bar{x}$. In (d), the sprouts $\bar{u}_1$, $\bar{u}_2$, and $\bar{u}_3$ form a replacing cycle.

Let $E_i = (\bar{u}, \bar{v})$ be a disagreement edge and $\bar{x}$ and $\bar{y}$ be the vertices or edges to which $\bar{u}$ and $\bar{v}$, respectively, are attached to in $N'$. If $\bar{x}$ or $\bar{y}$ is a disagreement edge $E_j$, then $E_i$ cannot be added to $N$ before $E_j$. Furthermore, if $\bar{y}$ is an ancestor of $\bar{x}$ in the embedding into $N$, adding $E_i$ to $N$ would create a directed cycle. Therefore we call a disagreement edge $E_i = (\bar{u}, \bar{v})$ *addable* if $\bar{y}$ is not an ancestor of $\bar{x}$ in $N$ and neither $\bar{x}$ nor $\bar{y}$ is a disagreement edge. For example, the edge $(\bar{u}, \bar{v})$ in Figure 6.12 (c) is not addable.

If $\bar{u}$ is a sprout attached to a vertex $\bar{x}$ in $N$, then there is a sprout $\bar{v}$ that is attached to $\bar{x}$ in $N'$. We say that $\bar{v}$ *takes the place* of $\bar{u}$. This allows us to define a *replacing sequence* $(\bar{u}_1, \ldots, \bar{u}_k)$ of sprouts such that $\bar{u}_i$ takes the place of $\bar{u}_{i+1}$ with regards to $N$ and $N'$. If furthermore $\bar{u}_k$ takes the place of $\bar{u}_1$, then we call it a *replacing cycle*. See Figure 6.12 (d) for an example. Note that in a replacing sequence the sprout $\bar{u}_1$ can be the sprout of a disagreement edge.

**Theorem 6.12.**
*Let $N, N' \in \mathcal{N}_n$. Then $\mathrm{d_{PR}}(N, N') \leq 3\,\mathrm{d_{AD}}(N, N')$.*

*Proof.* Let $N, N' \in \mathcal{N}_n$ with $r$ and $r'$ reticulations, respectively. Without loss of generality, assume that $r' \geq r$ and let $l = r' - r$. Let $G$ be a maximum agreement graph for $N$ and $N'$. Let $S_1, \ldots, S_k$ be the agreement subgraphs of $G$ and $E_1, \ldots, E_l$ be the disagreement edges of $G$. Fix agreement embeddings of $G$ into $N$ and into $N'$. For the embedding into $N'$, assume that it fulfills the properties of Lemma 6.3. That is, no sprout of an agreement subgraph $S_i$ is attached to a disagreement edge $E_j$, that at least one disagreement edge (if one exists) is not attached to any other disagreement edge, and that $E_i$ may be attached to $E_j$ only if $j < i$.

Let $d = \mathrm{d_{AD}}(N, N')$. To prove the statement we show how to construct a PR-sequence

$$\sigma = (N = N_0, N_1, \ldots, N_m = N')$$

with $m \leq 3d$. While $G$ has an agreement embedding into $N$ and $N'$, it may not have an agreement embedding for several $N_i$, $i \in \{1, \ldots, m-1\}$. However, starting at $N = N_0$, we preserve the mapping of vertices and edges of $G$ to vertices and paths of $N_{i-1}$ to $N_i$ with each step. Furthermore, along the sequence we map disagreement edges of $G$ to newly added edges. In some cases, it is necessary to add edges to $N_{i-1}$ to obtain $N_i$ with a $\mathrm{PR}^+$ to which no disagreement edge will be mapped. We call such edges *shadow edges*. From each $N_{i-1}$ to an $N_i$ we only prune edges at a vertex in $N_i$ to which a sprout and its incident edge are mapped, or add a disagreement edge, or add or alter a shadow edge. We describe any change of $G$, or of the embeddings of $G$ into $N_i$ or $N'$ explicitly.

To keep track of the length $m$ of $\sigma$, we credit every PR operation either to a sprout or to a disagreement edge. When we obtain $N_m = N'$, each sprout and each disagreement edge will have a credit of at most three and, hence, $m \leq 3d$. Now, assume $\sigma$ has been constructed up to $N_{i-1}$.

To obtain $N_i$ we apply the first applicable case of those described below to a sprout or to a disagreement edge. Overall the strategy is to first handle easy cases, that is prunable, unblocked sprouts (Case (A) and (A')) and addable disagreement edges (Case (B) and (B')). Then Case (C), (C') and (C") handle unprunable, unblocked sprouts. With Case (D) prunable, blocking sprouts are moved "aside" to make them non-blocking and Case (D') adds disagreement edges whose h-sprouts starts a replacing sequence of h-sprouts. After exhaustively applying Case (D) and (D'), we can prove that there always exists a prunable sprout (if any sprouts are left). A particular sprout (resp. disagreement edge) is subject of at most one application of Case (D) (resp. (D')) and one other case.

**(A) Prunable, unblocked sprout to non-shadow edge.** If there is a prunable, unblocked sprout $\bar{u}$ in $N_{i-1}$, then obtain $N_i$ by pruning $\bar{u}$ in $N_{i-1}$ and regrafting it to the edge $\bar{f}$ or vertex $\bar{x}$ to which $\bar{u}$ is attached in $N'$. This step gives $\bar{u}$ a credit of one operation. If $\bar{u}$ is regrafted to a vertex $\bar{x}$, let $\bar{v}$ be the sprout that is attached to $\bar{x}$ in $N_i$ (i.e. $\bar{u}$ takes the place of $\bar{v}$). Apply an embedding change of $G$ into $N$ with respect to $\bar{u}$ and $\bar{v}$. This whole step is illustrated in Figure 6.13. Note that $\bar{u}$ is now attached either to the same edge $\bar{f}$ or the same vertex $\bar{x}$ in both $N_i$ and $N'$. Therefore, for the rest of the proof, fix $\bar{u}$ to $\bar{f}$ or identify $\bar{u}$ with $\bar{x}$, respectively, in $G$. As a result, $\bar{u}$ with a credit of only one is now not a sprout anymore and thus not subject of another case.

**(B) Addable disagreement edge without shadow edge.** If there exists an addable disagreement edge $E_j$ for $N_{i-1}$, then obtain $N_i$ by adding $E_j$ to $N_{i-1}$ with a $\mathrm{PR}^+$. This step gives $E_j$ a credit of one operation. If a sprout of $E_j$ is attached to a vertex in $N'$, then apply again embedding changes of $G$ into $N_i$ like in Case (A). Note that $E_j$ is now attached to the same vertices or edges in both $N_i$ and $N'$. Therefore, merge the sprouts of $E_j$ with the vertices or edges they are attached to in $G$. As a result, $E_j$ with a credit of only one is now no disagreement edge anymore, but an edge of an agreement subgraph $S_{j'}$ of $G$. It will therefore not get any further credit.

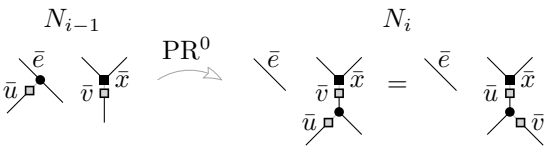


Figure 6.13: Illustration of Case (A) where a prunable unblocked sprout $\bar{u}$ gets regrafted to a vertex $\bar{x}$, and the subsequent embedding change with regards to $\bar{u}$ and $\bar{v}$.

**(C) Sprout at root, add shadow edge.** If there is an unprunable t-sprout $\bar{v}$ attached to the root $\rho$ in $N_{i-1}$, then there is another t-sprout $\bar{u}$ that is attached to the root in $N'$. Assume that $\bar{u}$ is a sprout of a disagreement edge $(\bar{u}, \bar{w})$ in $N'$, but that Case (B) does not apply. Then $\bar{w}$ must be attached to another disagreement edge in $N'$. This however can be changed with embedding changes (like in Lemma 6.3) such that $(\bar{u}, \bar{w})$ becomes addable and Case (B) applies. Therefore assume $\bar{u}$ is a sprout of an agreement subgraph. Since Case (A) does not apply and the root is an ancestor of $\bar{u}$, it follows that $\bar{u}$ is an unprunable, but unblocked t-sprout in $N_{i-1}$. Let $\bar{y}$ be the indegree two, outdegree zero vertex to which $\bar{u}$ is attached in $N_{i-1}$. We now obtain $N_i$ from $N_{i-1}$ by adding and attaching a shadow edge $(\bar{w}, \bar{z})$ from the outgoing edge of $\bar{u}$ to the incoming edge of leaf 1 with a $\mathrm{PR}^+$. After an embedding change of $G$ into $N_i$ with respect to $\bar{w}$ and $\bar{u}$, the sprout $\bar{u}$ becomes prunable. Give $\bar{u}$ a credit of one and apply Case (A) to obtain $N_{i+1}$. In total, $\bar{u}$ gets a credit of two and in $N_{i+1}$ and $N'$ no sprout is attached to the root anymore. This whole step is illustrated in Figure 6.15. As mentioned above, the embedding of $G$ into $N_{i+1}$ does not cover all edges anymore, since no edge is mapped to the shadow edge.

**(C') Sprout at leaf, add shadow edge.** This case is analogous to Case (C) but for h-sprouts. Here, if there is an unprunable h-sprout $\bar{v}$ attached to a leaf $l$ in $N_{i-1}$, then there is another unprunable, unblocked h-sprout $\bar{u}$ that takes the place of $\bar{v}$. Then obtain $N_i$ again by adding a shadow edge from the outgoing edge of $\rho$ to the incoming edge of $\bar{u}$. After applying an embedding change, obtain $N_{i+1}$ by pruning $\bar{u}$ and attaching it to the incoming edge $\bar{f}$ of $\bar{v}$. After another embedding change, merge $\bar{u}$ with the leaf $l$. If $l = 1$ and there is a shadow edge $(\bar{w}, \bar{z})$ attached to $\bar{f}$, then attach $\bar{u}$ above $\bar{z}$ to $\bar{f}$. This way, $\bar{z}$ is attached to the incoming edge of $l = \bar{u}$ and not to the incoming edge of $\bar{v}$ after the embedding change.

**(A') Prunable, unblocked sprout to shadow edge.** If after the previous two cases, there is again a prunable, unblocked sprout $\bar{u}$, apply Case (A) again. However, if in this process $\bar{u}$ gets regrafted to a shadow edge incident to the vertex $\bar{x}$, then remove the shadow edge with a $\mathrm{PR}^-$ after the embedding change of Case (A). This results in a total credit of two for $\bar{u}$ – one for the $\mathrm{PR}^0$ to move $\bar{u}$ and one for the $\mathrm{PR}^-$.

**(B') Addable disagreement edge with shadow edge.** Similarly, if there is now an addable disagreement edge $E_j = (\bar{u}, \bar{v})$, apply Case (B) in the following way. Assume that $\bar{v}$ of $E_j$ is supposed to get regrafted to a vertex $\bar{y}$ with an incoming shadow edge $\bar{f} = (\bar{w}, \bar{z})$. Then apply a $\mathrm{PR}^0$ to $N_{i-1}$ to prune $\bar{f}$ at $\bar{w}$ and regraft it where $\bar{u}$ is supposed to be attached. Then again, if $\bar{u}$ is supposed to be attached to a vertex $\bar{x}$ with an outgoing shadow edge $\bar{f}'$, remove $\bar{f}'$ with a $\mathrm{PR}^-$ operation after an embedding change. This step is illustrated in Figure 6.14. The case where only $\bar{u}$ is supposed to be attached to a vertex with an incident shadow edge but not $\bar{v}$ is handled analogously. If there is no shadow edge involved for either $\bar{u}$ or $\bar{v}$, then Case (B) directly applies. In either case, the total credit for $E_j$ is at most two.

The next case is used to decrease the number of blocking sprouts.

**(D) Blocked and blocking, but prunable sprout.** Let $\bar{u}$ be a prunable, blocked sprout that is blocking another sprout in $N_{i-1}$. Then obtain $N_i$ from $N_{i-1}$ by pruning $\bar{u}$

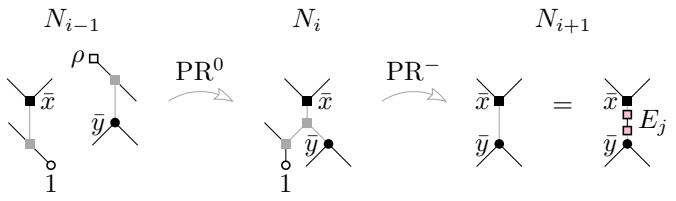


Figure 6.14: Illustration of Case (B') with two shadow edges.

and regrafting it to the outgoing edge of $\rho$ if $\bar{u}$ is a t-sprout, or to the incoming edge of leaf 1 otherwise. Note that $\bar{u}$ is now not blocking any other sprout in $N_i$. This step gives $\bar{u}$ a credit of one. Later on, $\bar{u}$ will get one or two more credit, depending on whether Case (A) or (A') will apply to it.

**(D') Non-addable disagreement edges attached to vertex.** Let $E_j = (\bar{u}, \bar{v})$ be a non-addable disagreement edge for which $\bar{v}$ is attached to a vertex $\bar{x}$ in $N'$. That means that a replacing sequence of h-sprouts starts with $\bar{v}$ of $E_j$ – we change this now. Obtain $N_i$ from $N_{i-1}$ by adding an edge $(\bar{u}, \bar{v})$ from the outgoing edge of $\rho$ to the incoming edge of $\bar{x}$. Identify $E_j$ with this new edge and then, after an embedding change, merge $\bar{v}$ with $\bar{x}$. The vertex $\bar{u}$ is now a non-blocking and prunable, but blocked t-sprout with a credit of one (just like the sprouts of Case (D)). Note that, after Case (D') does not apply anymore, there can be no replacing sequence of h-sprouts that starts with a sprout of a disagreement edge left. (We do not, maybe even cannot, do the analogous for disagreement edges that start a replacing sequence of t-sprouts.)

Applying Case (A) or Case (A') may now start with a sprout that has already a credit of one. However, as in both cases the credit is increased by at most two, the credit will afterwards be at most three.

So far we have applied Case (A) and (B) until not further possible. Then Case (C) and (C') are applied at most once and $n$ times, respectively. We then apply Cases (A), (A'), (B), (B') as long as possible. If then applicable we apply Case (D) or (D') and repeat this loop. Next, we show that if neither of the previous cases applies but there are still sprouts in $N_{i-1}$ that there is then at least one unprunable, unblocked sprout in $N_{i-1}$.

**Existence of unblocked sprout.** Assume that there exists a replacing cycle $\tau$ of, without loss of generality, t-sprouts in $N_{i-1}$. Then note that for a t-sprout to be blocked the vertex or edge it will be attached to has to be a descendant. Since phylogenetic networks are acyclic, the sprouts in $\tau$ cannot all replace a descendant. Therefore one of the sprouts has to be an unblocked sprout.

Next, assume that there is no replacing cycle in $N_{i-1}$. If no unprunable t-sprout $\bar{u}$ exists, then the h-sprout with no ancestor h-sprout in $N_{i-1}$ is an unblocked sprout. So assume otherwise and let $\bar{u}$ be an unprunable t-sprout with no descendant t-sprout in $N_{i-1}$. If $\bar{u}$ is unblocked, we are done; so assume otherwise. This means that the vertex or edge to which $\bar{u}$ is supposed to be regrafted is a descendant of $\bar{u}$ in $N_{i-1}$. Thus, by the choice of $\bar{u}$, it can only be blocked by an h-sprout $\bar{v}$. Since Case (D) moved prunable, blocking sprouts aside, $\bar{v}$ has to be unprunable. If $\bar{v}$ is unblocked, we are done; so assume otherwise. Then there is a replacing sequence $\tau = (\bar{v}_1, \ldots, \bar{v}_m)$ with $\bar{v} = \bar{v}_i$ for some $i \in \{2, \ldots, m\}$. Note that $\bar{v}_1$ is prunable since Case (D') does not apply and since there are no replacing cycles anymore and thus $\bar{v} \neq \bar{v}_1$. Since further Case (D) does not apply, $\bar{v}_1$ is also not a blocking sprout. Assuming that there is no unblocked sprout in $\tau$, we know that for every $1 \leq j < i$ the h-sprouts $\bar{v}_1$ to $\bar{v}_j$ are all descendants of $\bar{v}_{j+1}$ to $\bar{v}_i$ and thus also of $\bar{u}$. Since $\bar{v}_1$ is blocked, there has to be an unprunable h-sprout $\bar{v}'$ blocking $\bar{v}_1$. Note that $\bar{v}'$ is a descendant of $\bar{v}_2$ and thus not in $\tau$. The situation with $\bar{v}'$ is now the same as with $\bar{v}$

and the chain of descendants of h-sprouts below $\bar{u}$ contains now $\bar{v} = \bar{v}_i, \ldots, \bar{v}_2, \bar{v}'$. Finally, we either find an unprunable h-sprout in the replacing sequence $\tau' \neq \tau$ that contains $\bar{v}'$ or the chain of descendants of h-sprouts below $\bar{u}$ grows longer with h-sprouts $\bar{v}'_2$ and $\bar{v}''$. Since $N_{i-1}$ is finite this chain cannot grow indefinitely and thus at some point we find an unblocked h-sprout.

**(C") Unprunable, unblocked sprout.** If there is an unprunable, unblocked sprout $\bar{u}$ in $N_{i-1}$ that is attached to the edge $\bar{f}$ or a vertex $\bar{x}$ in $N'$ that has no shadow edge attached in $N_{i-1}$, then use the same procedure as in Case (C) or (C') to obtain $N_i$ and then $N_{i+1}$. This gives $\bar{u}$ a credit of two, before it gets merged with $\bar{x}$ or $\bar{f}$. This step is illustrated in Figure 6.15.

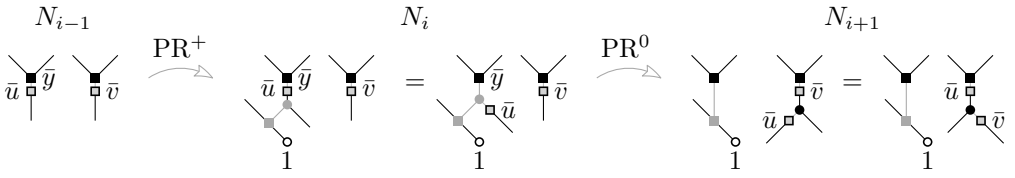


Figure 6.15: Illustration of Case (C) and Case (C") where an unprunable, unblocked sprout $\bar{u}$ is moved to the vertex $\bar{x}$ with two PR operations and two embedding changes.

If there is an unprunable, unblocked sprout $\bar{u}$ in $N_{i-1}$ that is attached to a vertex $\bar{x}$ in $N'$ that has a shadow edge attached in $N_{i-1}$, then apply the process shown in Figure 6.16 to obtain $N_i$ and $N_{i+1}$. This gives $\bar{u}$ a credit of two, before it gets merged with $\bar{x}$. Note that this moves the shadow edge from $\bar{x}$ to the vertex to which $\bar{u}$ was attached to in $N_{i-1}$.

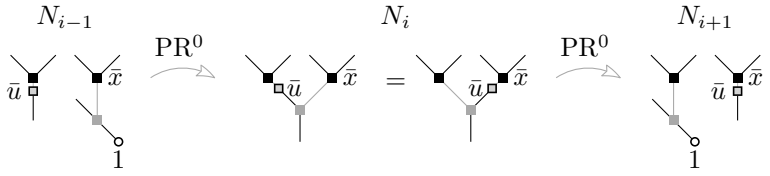


Figure 6.16: Illustration of Case (C") where an unprunable, unblocked sprout $\bar{u}$ is moved to a vertex $\bar{x}$ with an incident shadow edge.

Since prunable sprouts cannot block after they got moved aside, since disagreement edges cannot block either (by the properties of the agreement embedding into $N'$), and since the number of unprunable sprouts is decreased stepwise, the whole process resolves all sprouts and disagreement edges. Hence, $N_m = N'$. Since every sprout and every disagreement edge got a credit of at most three, it follows that $m \leq 3d$. This concludes the proof. □

We prove a relation between the PR-distance and the SNPR-distance.

**Lemma 6.13.**
*Let $N, N' \in \mathcal{N}_n$. Then $\mathrm{d}_{\mathrm{PR}}(N, N') \leq \mathrm{d}_{\mathrm{SNPR}}(N, N') \leq 2\,\mathrm{d}_{\mathrm{PR}}(N, N')$.*

*Proof.* The first inequality follows from the definitions of PR and SNPR. For the second inequality, let $d = \mathrm{d}_{\mathrm{PR}}(N, N')$ and $\sigma = (N = N_0, N_1, \ldots, N_d = N')$ be a PR-sequence from $N$ to $N'$ of length $d$. Then we can construct an SNPR-sequence $\sigma^* = (N = M_0, M_1, \ldots, M_k = N')$ with $k \leq 2d$ as follows. Assume we have constructed $\sigma^*$ up to $M_{j-1} = N_{i-1}$. Then, if $N_i$ is obtained from $N_{i-1}$ by a tail PR$^0$ or a PR$^+$ or a PR$^-$, then apply the same operation to $M_{j-1}$ to obtain $M_j$. So assume, otherwise; i.e. $N_i$ is obtained from $N_{i-1}$ by a head PR$^0$. Let $e = (u, v)$ be the edge that gets pruned at $v$ and

$f$ be the edge that gets subdivided to regraft $e$. Obtain $M_i$ from $M_{i-1}$ with the SNPR$^+$ that subdivides $e$ with a new vertex $u'$, subdivides $f$ with a new vertex $v'$, and adds the edge $(u', v')$. Next, obtain $M_{i+1}$ from $M_i$ by removing $(u', v)$ and suppressing the resulting degree two vertices. Then clearly $M_{i+1} = N_i$. Since at most two SNPR operations are needed per PR, it follows that $k \leq 2d$. □

The following corollary is a direct consequence of Theorems 6.11 and 6.12 and Lemma 6.13.

**Corollary 6.14.**
*Let $N, N' \in \mathcal{N}_n$. Then*

$$\mathrm{d_{AD}}(N, N') \leq \mathrm{d_{PR}}(N, N') \leq 3\,\mathrm{d_{AD}}(N, N')$$

*and*

$$\mathrm{d_{AD}}(N, N') \leq \mathrm{d_{SNPR}}(N, N') \leq 6\,\mathrm{d_{AD}}(N, N').$$

## 6.4 Concluding remarks

In this chapter we defined maximum agreement graphs for two rooted binary phylogenetic networks. Like maximum agreement forests for trees, a maximum agreement graph models how the two networks agree on subgraphs derived from a minimum number of prunings. If the two networks have different numbers of reticulations, then agreement graphs also model how they disagree on that. Based on this, we defined the agreement distance on phylogenetic networks. We then showed that agreement distance is a metric. Looking at the relation of the agreement distance to distances induced by rearrangement operations, we proved that the agreement distance equals the SPR-distance of two phylogenetic trees. What is more, the agreement distance agrees also with the SNPR- and PR-distance of a tree and a network. In general, for phylogenetic networks, the agreement distance is a lower bound on the SNPR- and PR-distance. Furthermore, it bounds both the SNPR- and PR-distance from above by a factor of at most three and six, respectively. These upper bounds might not be tight. For example, for the PR-distance the bound might be closer to twice the agreement distance. This thought is also motivated by the fact that the neighbourhoods of a network under PR and the agreement distance are the same.

While the agreement distance is still NP-hard to compute, it avoids problems of shortest SNPR- or PR-sequences that we have seen in the previous chapter. In particular, while for such a shortest sequence it might matter at which step of the sequence a reticulation edge is added, an agreement graph has simply as many disagreement edges as needed. Furthermore, while a sequence might traverse networks with more or less reticulations than the start and target network, this is also irrelevant for agreement graphs. Moreover, we have seen that there are multiple SNPR- and PR-distances of two networks based on whether we chose the distance within the class of the networks, like tree-child networks, in $\mathcal{N}_{n,r}$, or in $\mathcal{N}_n$. This is by definition not the case for the agreement distance. We therefore hope that it is easier to find exact and approximation algorithms for the agreement distance than for the PR-distance, just as it has been more fruitful to work with agreement forests than with shortest SPR-sequences.

Beyond rooted binary phylogenetic networks it is interesting to see whether agreement graphs and the agreement distance can be generalised to multifurcating phylogenetic networks or even to directed graphs in general. For unrooted phylogenetic trees, Allen and Steel [AS01] have shown that unrooted agreement forests characterise the TBR-distance. This imposes the questions whether agreement graphs can also be defined for unrooted phylogenetic networks and how they would relate to generalisations of the (unrooted) SPR and the TBR operation.

# 7. Conclusions

In this thesis we have studied several problems concerning spaces of phylogenetic networks. In Chapter 3 we proved that most classes of phylogenetic networks and their tiers are connected under SNPR and PR, and, for some classes, also under NNI. In the affirmative cases, this established that the distance induced by the rearrangement operations are metrics. We also gave asymptotic bounds on the diameters of these spaces. Here, we found that if tight bounds are known for diameters under SNPR or PR, then they are linear in the size of the network, like they are for the space of phylogenetic trees.

In Chapter 4 we saw that the size of the SNPR neighbourhood of a normal or tree-child network depends not only on its size and topology but also on the occurrences of certain subgraphs. In particular, we found that the neighbour size is asymptotically quadratic in the size of the network, like the size of the SPR neighbourhood of a tree. However, compared to trees, counting neighbours is more challenging for networks since different operations can lead to the same neighbours. We have seen that this problem of redundant operations becomes even harder for networks where vertices and edges are not necessarily uniquely identifiable, for example, like in reticulation-visible, tree-based, or general phylogenetic networks.

In Chapter 5 we looked at the behaviour of shortest paths under SNPR and PR in the space of phylogenetic networks. We saw that the space of phylogenetic trees is an isometric subgraph of the space of phylogenetic networks under SNPR and PR. Furthermore, we showed that there is always a shortest path from a tree to a network that can be divided into two parts where the first part uses only horizontal operations while the second part uses only vertical operations. This allowed us to characterise the distance of a tree and a network in terms of the trees displayed by the network. As a consequence, their distance can be computed with a fixed-parameter tractable algorithm. Concerning the distances between two networks, we found that shortest paths may have bad properties. For example, there are pairs of networks with the same number of reticulations such that every shortest path between them contains a tree. Furthermore, for two networks of a certain class, one has to decided which space to consider for their distance, because most spaces do not embedded isometrically into each other under SNPR and PR.

In Chapter 6 we introduced maximum agreement graphs as generalisation of maximum agreement forests. We showed that they induce a metric, the agreement distance, on phylogenetic networks. Even though the agreement distance does not characterise the SNPR- and PR-distance, it still bounds them with constant factors.

We conclude that many results on trees can be lifted to networks. However, at this stage

they are still less precise. For example, the bounds on diameters of spaces of phylogenetic networks can be sharpened. Furthermore, several problems remain open. While we were able to use the fixed-parameter tractable algorithm for two trees to compute the distance for a tree and a network, there is no known algorithm to compute the agreement distance or a rearrangement distance of two networks yet. Here, maximum agreement graphs could be of help.

# Bibliography


[AK97] A. Amir and D. Keselman, "Maximum Agreement Subtree in a Set of Evolutionary Trees: Metrics and Efficient Algorithms," *SIAM Journal on Computing*, vol. 26, no. 6, pp. 1656–1669, 1997. doi:10.1137/S0097539794269461

[ALC+14] A. R. Amaral, G. Lovewell, M. M. Coelho, G. Amato, and H. C. Rosenbaum, "Hybrid Speciation in a Marine Mammal: The Clymene Dolphin (Stenella clymene)," *PLOS ONE*, vol. 9, no. 1, pp. 1–8, 2014. doi:10.1371/journal.pone.0083645

[AS01] B. L. Allen and M. Steel, "Subtree Transfer Operations and Their Induced Metrics on Evolutionary Trees," *Annals of Combinatorics*, vol. 5, no. 1, pp. 1–15, 2001. doi:10.1007/s00026-001-8006-8

[Bar04] M. Baroni, "Hybrid phylogenies: a graph-based approach to represent reticulate evolution," Ph.D. dissertation, University of Canterbury, New Zealand, 2004. http://ir.canterbury.ac.nz/bitstream/10092/4803/1/baroni_thesis.pdf

[BGMS05] M. Baroni, S. Grünewald, V. Moulton, and C. Semple, "Bounding the Number of Hybridisation Events for a Consistent Evolutionary History," *Journal of Mathematical Biology*, vol. 51, no. 2, pp. 171–182, 2005. doi:10.1007/s00285-005-0315-9

[BHK+14] R. Bouckaert, J. Heled, D. Kühnert, T. Vaughan, C.-H. Wu, D. Xie, M. A. Suchard, A. Rambaut, and A. J. Drummond, "BEAST 2: A Software Platform for Bayesian Evolutionary Analysis," *PLOS Computational Biology*, vol. 10, no. 4, pp. 1–6, 2014. doi:10.1371/journal.pcbi.1003537

[Bic12] D. R. Bickner, "On normal networks," Ph.D. dissertation, Iowa State University, 2012, http://gradworks.umi.com/3511361.pdf. http://gradworks.umi.com/35/11/3511361.html

[BLS17] M. Bordewich, S. Linz, and C. Semple, "Lost in space? Generalising subtree prune and regraft to spaces of phylogenetic networks," *Journal of Theoretical Biology*, vol. 423, pp. 1–12, 2017. doi:10.1016/j.jtbi.2017.03.032

[BMS08] M. Bordewich, C. McCartin, and C. Semple, "A 3-approximation algorithm for the subtree distance between phylogenies," *Journal of Discrete Algorithms*, vol. 6, no. 3, pp. 458–471, 2008. doi:10.1016/j.jda.2007.10.002

[BMSW15] S. Baskowski, V. Moulton, A. Spillner, and T. Wu, "Neighborhoods of Trees in Circular Orderings," *Bulletin of Mathematical Biology*, vol. 77, no. 1, pp. 46–70, 2015. doi:10.1007/s11538-014-0049-1

[Bor03] M. Bordewich, "The Complexity of Counting and Randomised Approximation," Ph.D. dissertation, University of Oxford, 2003. http://community.dur.ac.uk/m.j.r.bordewich/papers/Bordewich2003-a.pdf

[BS05] M. Bordewich and C. Semple, "On the Computational Complexity of the Rooted Subtree Prune and Regraft Distance," *Annals of Combinatorics*, vol. 8, no. 4, pp. 409–423, 2005. doi:10.1007/s00026-004-0229-z

[BS18] ——, "A universal tree-based network with the minimum number of reticulations," *Discrete Applied Mathematics*, vol. 250, pp. 357–362, 2018. doi:10.1016/j.dam.2018.05.010

[BSJ09] M. L. Bonet and K. St. John, "Efficiently Calculating Evolutionary Tree Measures Using SAT," in *Theory and Applications of Satisfiability Testing - SAT 2009*, O. Kullmann, Ed. Springer, 2009, pp. 4–17. doi:10.1007/978-3-642-02777-2_3

[BSJMA06] M. L. Bonet, K. St. John, R. Mahindru, and N. Amenta, "Approximating subtree distances between phylogenies," *Journal of Computational Biology*, vol. 13, no. 8, pp. 1419–1434, 2006. doi:10.1089/cmb.2006.13.1419

[BSS06] M. Baroni, C. Semple, and M. Steel, "Hybrids in Real Time," *Systematic Biology*, vol. 55, no. 1, pp. 46–56, 2006. doi:10.1080/10635150500431197

[CCLSJ13] A. J. J. Caceres, J. Castillo, J. Lee, and K. St. John, "Walks on SPR Neighborhoods," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 10, no. 1, pp. 236–239, 2013. doi:10.1109/TCBB.2012.136

[CFCH+00] R. Cole, M. Farach-Colton, R. Hariharan, T. Przytycka, and M. Thorup, "An $O(n \log n)$ Algorithm for the Maximum Agreement Subtree Problem for Binary Trees," *SIAM Journal on Computing*, vol. 30, no. 5, pp. 1385–1404, 2000. doi:10.1137/S0097539796313477

[CFS15] J. Chen, J.-H. Fan, and S.-H. Sze, "Parameterized and approximation algorithms for maximum agreement forest in multifurcating trees," *Theoretical Computer Science*, vol. 562, pp. 496–512, 2015. doi:10.1016/j.tcs.2014.10.031

[Cha05] F. Chataigner, "Approximating the Maximum Agreement Forest on k trees," *Information Processing Letters*, vol. 93, no. 5, pp. 239–244, 2005. doi:10.1016/j.ipl.2004.11.004

[CHW17] Z.-Z. Chen, Y. Harada, and L. Wang, "A New 2-Approximation Algorithm for rSPR Distance," in *Bioinformatics Research and Applications*, Z. Cai, O. Daescu, and M. Li, Eds. Springer, 2017, pp. 128–139. doi:10.1007/978-3-319-59575-7_12

[CJSS05] C. Choy, J. Jansson, K. Sadakane, and W.-K. Sung, "Computing the maximum agreement of phylogenetic networks," *Theoretical Computer Science*, vol. 335, no. 1, pp. 93–107, 2005. doi:10.1016/j.tcs.2004.12.012

[CLRS09] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. MIT Press, 2009. https://mitpress.mit.edu/books/introduction-algorithms-third-edition

[CLRV08] G. Cardona, M. Llabrés, F. Rosselló, and G. Valiente, "A distance metric for a class of tree-sibling phylogenetic networks," *Bioinformatics*, vol. 24, no. 13, pp. 1481–1488, 2008. doi:10.1093/bioinformatics/btn231

[CLRV09a] ——, "Metrics for Phylogenetic Networks I: Generalizations of the Robinson-Foulds Metric," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 6, no. 1, pp. 46–61, 2009. doi:10.1109/TCBB.2008.70

[CLRV09b] ——, "Metrics for Phylogenetic Networks II: Nodal and Triplets Metrics," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 6, no. 3, pp. 454–469, 2009. doi:10.1109/TCBB.2008.127

[CLRV09c] ——, "On Nakhleh's Metric for Reduced Phylogenetic Networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 6, no. 4, pp. 629–638, 2009. doi:10.1109/TCBB.2009.33

[CPR15] G. Cardona, J. C. Pons, and F. Rosselló, "A reconstruction problem for a class of phylogenetic networks with lateral gene transfers," *Algorithms for Molecular Biology*, vol. 10, no. 1, p. 28, 2015. doi:10.1186/s13015-015-0059-z

[CPS19] G. Cardona, J. C. Pons, and C. Scornavacca, "Generation of Tree-Child phylogenetic networks," *arXiv preprint arXiv:1902.09015*, 2019. arXiv:1902.09015

[CRV09] G. Cardona, F. Rosselló, and G. Valiente, "Comparison of Tree-Child Phylogenetic Networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 6, no. 4, pp. 552–569, 2009. doi:10.1109/TCBB.2007.70270

[CSW16] J. Chen, F. Shi, and J. Wang, "Approximating Maximum Agreement Forest on Multiple Binary Trees," *Algorithmica*, vol. 76, no. 4, pp. 867–889, 2016. doi:10.1007/s00453-015-0087-6

[DGH11] Y. Ding, S. Grünewald, and P. J. Humphries, "On agreement forests," *Journal of Combinatorial Theory, Series A*, vol. 118, no. 7, pp. 2059–2065, 2011. doi:10.1016/j.jcta.2011.04.013

[Die17] R. Diestel, *Graph Theory*, 5th ed. Springer, 2017. doi:10.1007/978-3-662-53622-3

[dJMS16] J. V. de Jong, J. C. McLeod, and M. Steel, "Neighborhoods of Phylogenetic Trees: Exact and Asymptotic Counts," *SIAM Journal on Discrete Mathematics*, vol. 30, no. 4, pp. 2265–2287, 2016. doi:10.1137/15M1035070

[Dun14] M. Dunn, "Language phylogenies," in *The Routledge Handbook of Historical Linguistics*, C. Bowern and B. Evans, Eds. Routledge, 2014, ch. 7. https://www.routledgehandbooks.com/doi/10.4324/9781315794013.ch7

[EOZN19] R. A. L. Elworth, H. A. Ogilvie, J. Zhu, and L. Nakhleh, *Advances in Computational Methods for Phylogenetic Networks in the Presence of Hybridization.* Springer, 2019, pp. 317–360. doi:10.1007/978-3-030-10837-3_13

[ESS19] P. L. Erdős, C. Semple, and M. Steel, "A class of phylogenetic networks reconstructable from ancestral profiles," *Mathematical Biosciences*, vol. 313, pp. 33–40, 2019. doi:10.1016/j.mbs.2019.04.009

[Fel04] J. Felsenstein, *Inferring phylogenies.* Sinauer Associates, 2004, vol. 2.

[FGH+18] M. Fischer, M. Galla, L. Herbst, Y. Long, and K. Wicke, "Non-binary treebased unrooted phylogenetic networks and their relations to binary and rooted ones," *arXiv preprint arXiv:1810.06853*, 2018. arXiv:1810.06853

[FGM19] M. Fuchs, B. Gittenberger, and M. Mansouri, "Counting phylogenetic networks with few reticulation vertices: tree-child and normal networks," *Australasian Journal of Combinatorics*, vol. 73, no. 2, pp. 385–423, 2019. https://ajc.maths.uq.edu.au/pdf/73/ajc_v73_p385.pdf

[FHM18] A. Francis, K. T. Huber, and V. Moulton, "Tree-Based Unrooted Phylogenetic Networks," *Bulletin of Mathematical Biology*, vol. 80, no. 2, pp. 404–416, 2018. doi:10.1007/s11538-017-0381-3

[FHMW18] A. Francis, K. T. Huber, V. Moulton, and T. Wu, "Bounds for phylogenetic network space metrics," *Journal of Mathematical Biology*, vol. 76, no. 5, pp. 1229–1248, 2018. doi:10.1007/s00285-017-1171-0

[FS15] A. R. Francis and M. Steel, "Which Phylogenetic Networks are Merely Trees with Additional Arcs?" *Systematic Biology*, vol. 64, no. 5, pp. 768–777, 2015. doi:10.1093/sysbio/syv037

[FSS18] A. Francis, C. Semple, and M. Steel, "New characterisations of tree-based networks and proximity measures," *Advances in Applied Mathematics*, vol. 93, pp. 93–107, 2018. doi:10.1016/j.aam.2017.08.003

[GBP09] P. Gambette, V. Berry, and C. Paul, "The Structure of Level-k Phylogenetic Networks," in *Combinatorial Pattern Matching*, G. Kucherov and E. Ukkonen, Eds. Springer, 2009, pp. 289–300. doi:10.1007/978-3-642-02441-2_26

[GDL+10] S. Guindon, J.-F. Dufayard, V. Lefort, M. Anisimova, W. Hordijk, and O. Gascuel, "New Algorithms and Methods to Estimate Maximum-Likelihood Phylogenies: Assessing the Performance of PhyML 3.0," *Systematic Biology*, vol. 59, no. 3, pp. 307–321, 2010. doi:10.1093/sysbio/syq010

[GEL03] D. Gusfield, S. Eddhu, and C. Langley, "Efficient reconstruction of phylogenetic networks with constrained recombination," in *Proceedings of the 2003 IEEE Bioinformatics Conference.* IEEE Computer Society Press, 2003, pp. 363–374. doi:10.1109/CSB.2003.1227337

[GGL+15] P. Gambette, A. D. M. Gunawan, A. Labarre, S. Vialette, and L. Zhang, "Locating a Tree in a Phylogenetic Network in Quadratic Time," in *Research in Computational Molecular Biology*, T. M. Przytycka, Ed. Springer, 2015, pp. 96–107. doi:10.1007/978-3-319-16706-0_12

[GJ79] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, 1979.

[Gus14] D. Gusfield, *ReCombinatorics: The Algorithmics of Ancestral Recombination Graphs and Explicit Phylogenetic Networks.* MIT Press, 2014.

[GvIJ+17] P. Gambette, L. van Iersel, M. Jones, M. Lafond, F. Pardi, and C. Scornavacca, "Rearrangement moves on rooted phylogenetic networks," *PLOS Computational Biology*, vol. 13, no. 8, pp. 1–21, 2017. doi:10.1371/journal.pcbi.1005611

[GWMI18] A. Gavryushkin, C. Whidden, and F. A. Matsen IV, "The combinatorics of discrete time-trees: theory and open problems," *Journal of Mathematical Biology*, vol. 76, no. 5, pp. 1101–1121, 2018. doi:10.1007/s00285-017-1167-9

[GZ15] A. D. Gunawan and L. Zhang, "Bounding the Size of a Network Defined By Visibility Property," *arXiv preprint arXiv:1510.00115*, 2015. arXiv:1510.00115

[Hay16] M. Hayamizu, "On the existence of infinitely many universal tree-based networks," *Journal of Theoretical Biology*, vol. 396, pp. 204–206, 2016. doi:10.1016/j.jtbi.2016.02.023

[HDRCB08] G. Hickey, F. Dehne, A. Rau-Chaplin, and C. Blouin, "SPR Distance Computation for Unrooted Trees," *Evolutionary Bioinformatics*, vol. 4, p. EBO.S419, 2008. doi:10.4137/EBO.S419

[Hen18] M. Hendriksen, "Tree-based unrooted nonbinary phylogenetic networks," *Mathematical Biosciences*, vol. 302, pp. 131–138, 2018. doi:10.1016/j.mbs.2018.06.005

[HJWZ96] J. Hein, T. Jiang, L. Wang, and K. Zhang, "On the complexity of comparing evolutionary trees," *Discrete Applied Mathematics*, vol. 71, no. 1, pp. 153–169, 1996. doi:10.1016/S0166-218X(96)00062-5

[HL18] D. H. Huson and S. Linz, "Autumn Algorithm — Computation of Hybridization Networks for Realistic Phylogenetic Trees," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 2, pp. 398–410, 2018. doi:10.1109/TCBB.2016.2537326

[HLMW16] K. T. Huber, S. Linz, V. Moulton, and T. Wu, "Spaces of phylogenetic networks from generalized nearest-neighbor interchange operations," *Journal of Mathematical Biology*, vol. 72, no. 3, pp. 699–725, 2016. doi:10.1007/s00285-015-0899-7

[HMSW16] K. T. Huber, V. Moulton, M. Steel, and T. Wu, "Folding and unfolding phylogenetic trees and networks," *Journal of Mathematical Biology*, vol. 73, no. 6, pp. 1761–1780, 2016.doi:10.1007/s00285-016-0993-5

[HMW16] K. T. Huber, V. Moulton, and T. Wu, "Transforming phylogenetic networks: Moving beyond tree space," *Journal of Theoretical Biology*, vol. 404, pp. 30–39, 2016. doi:10.1016/j.jtbi.2016.05.030

[HRS10] D. H. Huson, R. Rupp, and C. Scornavacca, *Phylogenetic Networks: Concepts, Algorithms and Applications.* Cambridge University Press, 2010. doi:10.1093/sysbio/syr055

[HS10] D. H. Huson and C. Scornavacca, "A Survey of Combinatorial Methods for Phylogenetic Networks," *Genome Biology and Evolution*, vol. 3, pp. 23–35, 2010. doi:10.1093/gbe/evq077

[HW13] P. J. Humphries and T. Wu, "On the Neighborhoods of Trees," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 10, no. 3, pp. 721–728, 2013. doi:10.1109/TCBB.2013.66

[Jan18] R. Janssen, "Heading in the right direction? Using head moves to traverse phylogenetic network space," *arXiv preprint arXiv:1810.09795*, 2018. arXiv:1810.09795

[JJE+18] R. Janssen, M. Jones, P. L. Erdős, L. van Iersel, and C. Scornavacca, "Exploring the tiers of rooted phylogenetic network space using tail moves," *Bulletin of Mathematical Biology*, vol. 80, no. 8, pp. 2177–2208, 2018. doi:10.1007/s11538-018-0452-0

[JK19] R. Janssen and J. Klawitter, "Rearrangement operations on unrooted phylogenetic networks," *arXiv preprint arXiv:1906.04468*, 2019. arXiv:1906.04468

[JM18] R. Janssen and Y. Murakami, "Solving phylogenetic network containment problems using cherry-picking sequences," *arXiv preprint arXiv:1812.08065*, 2018. arXiv:1812.08065

[JvI18] L. Jetten and L. van Iersel, "Nonbinary tree-based phylogenetic networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 1, pp. 205–217, 2018. doi:10.1109/TCBB.2016.2615918

[KL19] J. Klawitter and S. Linz, "On the Subnet Prune and Regraft Distance," *Electronic Journal of Combinatorics*, vol. 26, pp. 329–355, 2019. https://www.combinatorics.org/ojs/index.php/eljc/article/view/v26i2p3

[Kla18] J. Klawitter, "The SNPR neighbourhood of tree-child networks," *Journal of Graph Algorithms and Applications*, vol. 22, no. 2, pp. 329–355, 2018. doi:10.7155/jgaa.00472

[Kla19] ——, "The agreement distance of rooted phylogenetic networks," *Discrete Mathematics & Theoretical Computer Science*, vol. 21, no. 3, 2019. https://dmtcs.episciences.org/5487

[LS09] S. Linz and C. Semple, "Hybridization in Nonbinary Trees," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 6, no. 1, pp. 30–45, 2009. doi:10.1109/TCBB.2008.86

[LS11] ——, "A Cluster Reduction for Computing the Subtree Distance Between Phylogenies," *Annals of Combinatorics*, vol. 15, no. 3, p. 465, 2011. doi:10.1007/s00026-011-0108-3

[LSJS13] S. Linz, K. St. John, and C. Semple, "Counting Trees in a Phylogenetic Network Is #P-Complete," *SIAM Journal on Computing*, vol. 42, no. 4, pp. 1768–1776, 2013. doi:10.1137/12089394X

[LTZ96] M. Li, J. Tromp, and L. Zhang, *Some notes on the nearest neighbour interchange distance*. Springer, 1996, pp. 343–351. doi:10.1007/3-540-61332-3_168

[Mar18] A. Markin, "On the Extremal Maximum Agreement Subtree Problem," *arXiv preprint arXiv:1812.06951*, 2018. arXiv:1812.06951

[MMM+17] J. I. Meier, D. A. Marques, S. Mwaiko, C. E. Wagner, L. Excoffier, and O. Seehausen, "Ancient hybridization fuels rapid cichlid fish adaptive radiations," *Nature communications*, vol. 8, p. 14363, 2017. doi:10.1038/ncomms14363

[MMS16] S. Mark, J. C. McLeod, and M. Steel, "A navigation system for tree space," *Journal of Graph Algorithms and Applications*, vol. 20, no. 2, pp. 247–268, 2016. doi:10.7155/jgaa.00392

[MSW15] C. McDiarmid, C. Semple, and D. Welsh, "Counting Phylogenetic Networks," *Annals of Combinatorics*, vol. 19, no. 1, pp. 205–224, 2015. doi:10.1007/s00026-015-0260-2

[Nie06] R. Niedermeier, *Invitation to fixed-parameter algorithms.* Oxford University Press, 2006, see page 21.

[Pag93] R. D. M. Page, "On Islands of Trees and the Efficacy of Different Methods of Branch Swapping in Finding Most-Parsimonious Trees," *Systematic Biology*, vol. 42, no. 2, pp. 200–210, 1993. doi:10.2307/2992542

[PSS19] J. C. Pons, C. Semple, and M. Steel, "Tree-based networks: characterisations, metrics, and support trees," *Journal of Mathematical Biology*, vol. 78, no. 4, pp. 899–918, 2019. doi:10.1007/s00285-018-1296-9

[RH03] F. Ronquist and J. P. Huelsenbeck, "MrBayes 3: Bayesian phylogenetic inference under mixed models," *Bioinformatics*, vol. 19, no. 12, pp. 1572–1574, 2003. doi:10.1093/bioinformatics/btg180

[Rob71] D. F. Robinson, "Comparison of labeled trees with valency three," *Journal of Combinatorial Theory, Series B*, vol. 11, no. 2, pp. 105–119, 1971. doi:10.1016/0095-8956(71)90020-7

[RSW07] E. M. Rodrigues, M.-F. Sagot, and Y. Wakabayashi, "The maximum agreement forest problem: Approximation algorithms and computational experiments," *Theoretical Computer Science*, vol. 374, no. 1, pp. 91–110, 2007. doi:10.1016/j.tcs.2006.12.011

[RW07] L. H. Rieseberg and J. H. Willis, "Plant Speciation," *Science*, vol. 317, no. 5840, pp. 910–914, 2007. doi:10.1126/science.1137729

[Sch70] E. Schröder, "Vier kombinatorische Probleme," *Zeitschrift für Mathematik und Physik*, vol. 15, pp. 361–376, 1870.

[Sem16] C. Semple, "Phylogenetic Networks with Every Embedded Phylogenetic Tree a Base Tree," *Bulletin of Mathematical Biology*, vol. 78, no. 1, pp. 132–137, 2016. doi:10.1007/s11538-015-0132-2

[SFYW16] F. Shi, Q. Feng, J. You, and J. Wang, "Improved approximation algorithm for maximum agreement forest of two rooted binary phylogenetic trees," *Journal of Combinatorial Optimization*, vol. 32, no. 1, pp. 111–143, 2016. doi:10.1007/s10878-015-9921-7

[SJ17] K. St. John, "Review Paper: The Shape of Phylogenetic Treespace," *Systematic Biology*, vol. 66, no. 1, pp. e83–e94, 2017. doi:10.1093/sysbio/syw025

[Son03] Y. S. Song, "On the combinatorics of rooted binary phylogenetic trees," *Annals of Combinatorics*, vol. 7, no. 3, pp. 365–379, 2003. doi:10.1007/s00026-003-0192-0

[Son06] ——, "Properties of Subtree-Prune-and-Regraft Operations on Totally-Ordered Phylogenetic Trees," *Annals of Combinatorics*, vol. 10, no. 1, pp. 147–163, 2006. doi:10.1007/s00026-006-0279-5

[SOW96] D. L. Swofford, G. J. Olsen, and P. J. Waddell, "Phylogenetic Inference," in *Molecular Systematics*, D. M. Hillis, C. Moritz, and B. K. Mable, Eds. Sinauer Associates, 1996, ch. 11, pp. 407–514.

[SS03] C. Semple and M. A. Steel, *Phylogenetics*. Oxford University Press, 2003, vol. 24.

[SvZvdS16] F. Schalekamp, A. van Zuylen, and S. van der Ster, "A Duality Based 2-Approximation Algorithm for Maximum Agreement Forest," in *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), I. Chatzigiannakis, M. Mitzenmacher, Y. Rabani, and D. Sangiorgi, Eds., vol. 55. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016, pp. 70:1–70:14. http://drops.dagstuhl.de/opus/volltexte/2016/6214

[SW93] M. Steel and T. Warnow, "Kaikoura tree theorems: Computing the maximum agreement subtree," *Information Processing Letters*, vol. 48, no. 2, pp. 77–82, 1993. doi:10.1016/0020-0190(93)90181-8

[TKR09] V. Trifonov, H. Khiabanian, and R. Rabadan, "Geographic Dependence, Surveillance, and Origins of the 2009 Influenza A (H1N1) Virus," *New England Journal of Medicine*, vol. 361, no. 2, pp. 115–119, 2009. doi:doi.org/10.1056/NEJMp0904572

[TN05] C. M. Thomas and K. M. Nielsen, "Mechanisms of, and Barriers to, Horizontal Gene Transfer between Bacteria," *Nature Reviews Microbiology*, vol. 3, no. 9, pp. 711–721, 2005. doi:10.1038/nrmicro1234

[vIK11] L. van Iersel and S. Kelk, "Constructing the Simplest Possible Phylogenetic Network from Triplets," *Algorithmica*, vol. 60, no. 2, pp. 207–235, 2011. doi:10.1007/s00453-009-9333-0

[vIKLS14] L. van Iersel, S. Kelk, N. Lekic, and L. Stougie, "Approximation algorithms for nonbinary agreement forests," *SIAM Journal on Discrete Mathematics*, vol. 28, no. 1, pp. 49–66, 2014. doi:10.1137/120903567

[WBZ13] C. Whidden, R. G. Beiko, and N. Zeh, "Fixed-Parameter Algorithms for Maximum Agreement Forests," *SIAM Journal on Computing*, vol. 42, no. 4, pp. 1431–1466, 2013. doi:10.1137/110845045

[WBZ16] ——, "Fixed-Parameter and Approximation Algorithms for Maximum Agreement Forests of Multifurcating Trees," *Algorithmica*, vol. 74, no. 3, pp. 1019–1054, 2016. doi:10.1007/s00453-015-9983-z

[Wil10] S. J. Willson, "Properties of Normal Phylogenetic Networks," *Bulletin of Mathematical Biology*, vol. 72, no. 2, pp. 340–358, 2010. doi:10.1007/s11538-009-9449-z

[WMI15] C. Whidden and F. A. Matsen IV, "Quantifying MCMC Exploration of Phylogenetic Tree Space," *Systematic Biology*, vol. 64, no. 3, pp. 472–491, 2015. doi:10.1093/sysbio/syv006

[WMI17] ——, "Ricci–Ollivier curvature of the rooted phylogenetic subtree–prune–regraft graph," *Theoretical Computer Science*, vol. 699, pp. 1–20, 2017. doi:10.1016/j.tcs.2017.02.006

[Wu09] Y. Wu, "A practical method for exact computation of subtree prune and regraft distance," *Bioinformatics*, vol. 25, no. 2, pp. 190–196, 2009. doi:10.1093/bioinformatics/btn606

[WWK+17] A. Wagner, R. J. Whitaker, D. J. Krause, J.-H. Heilers, M. van Wolferen, C. van der Does, and S.-V. Albers, "Mechanisms of gene flow in archaea," *Nature Reviews Microbiology*, vol. 15, no. 8, pp. 492–502, 2017. doi:10.1038/nrmicro.2017.41

[WZ09] C. Whidden and N. Zeh, "A Unifying View on Approximation and FPT of Agreement Forests," in *Algorithms in Bioinformatics*, S. L. Salzberg and T. Warnow, Eds. Springer, 2009, pp. 390–402. doi:10.1007/978-3-642-04241-6_32

[YBN13] Y. Yu, R. M. Barnett, and L. Nakhleh, "Parsimonious Inference of Hybridization in the Presence of Incomplete Lineage Sorting," *Systematic Biology*, vol. 62, no. 5, pp. 738–751, 07 2013. doi:10.1093/sysbio/syt037

[YDLN14] Y. Yu, J. Dong, K. J. Liu, and L. Nakhleh, "Maximum likelihood inference of reticulate evolutionary histories," *Proceedings of the National Academy of Sciences*, vol. 111, no. 46, pp. 16 448–16 453, 2014. doi:10.1073/pnas.1407950111

[Zha16] L. Zhang, "On Tree-Based Phylogenetic Networks," *Journal of Computational Biology*, vol. 23, no. 7, pp. 553–565, 2016. doi:10.1089/cmb.2015.0228