Classes of Semi-binary Phylogenetic Networks encoded by μ-representations

Classes of Semi-binary Phylogenetic Networks encoded by μ-representations

Thesis

to obtain the degree of Master of Science,
as part of the master program Applied Mathematics
specializing in Discrete Mathematics and Optimization
at the faculty of Electrical Engineering, Mathematics, and Computer Science
(EEMCS),
at Delft University of Technology,
to be publicly defended on Thursday, September 14, 2023, at 14:00

by

Christopher Reichling

Student number: 4153685

Thesis Committee:

Dr. ir. L.J.J. van Iersel,	Technische Universiteit Delft
Dr. Y. Murakami,	Technische Universiteit Delft
Dr. A. Heinlein,	Technische Universiteit Delft

Contents

Summary

This thesis is on the subject of phylogenetic networks. These are schematic visualisations used mainly to investigate the evolutionary history of species, but which can be used for any set of distinguishable elements which have diverged from a common ancestor through some evolutionary process. The research specifically focuses on a way to encode these phylogenetic networks, called μ-representation, which enables researchers to efficiently compare networks in polynomial time.

The main contribution of this thesis lies in demonstrating that there are certain classes of phylogenetic networks for which the μ-representation or a modified version thereof serves as a unique encoding and can therefore be used to generate a metric for comparison. Additionally, it is shown that these results do not extend to some other classes of networks. Furthermore, this research shows that certain other information can be gained from analysing the μ-representation of a network, such as which nodes are adjacent to so-called bridges or cut-edges, and what the in-degrees of the nodes in the network are.

Acknowledgements

After completing my bachelor's degree in Applied Physics, I had decided not to continue my studies in that direction, and for a while, I felt somewhat lost. The last few years of studying Applied Mathematics have shown me that this is a scientific field I genuinely love. However, it was only when I started working on this thesis, that I realized how much I truly enjoy doing theoretical research and abstract mathematics (and that maybe I am also actually pretty good at it). This was in no small part due to the wonderful supervision I got from both Yuki and Leo. The enthusiasm with which they met all of my ideas and work (even when I was wrong) and the dedication they showed in collaborating with me on finding new results or solutions to problems was an endless source of motivation and joy for me. I am forever grateful to them for instilling so much confidence and pride in me at a time when I was still quite unsure of myself. I also really want to thank my wonderful girlfriend Eva for always supporting me and listening to all my ramblings and having so much patience, even though the subject is so far from her expertise. I want to thank my friend Wolf for our great discussions, helping me think through my problems and helping me stay disciplined by our weekly study sessions. And finally, I want to thank my parents Jantje and Antonie, for supporting me all the way through my academic journey and with each of the choices I made. I am very happy with what I've managed to accomplish and the way that this thesis has turned out. I hope that I've been able to make a meaningful contribution to the field. In any case this experience has inspired me to keep looking for interesting discoveries as I move forward on this journey.

Christopher Reichling
The Hague, August 2023

1 Introduction

1.1 Phylogenetic trees

Ever since Charles Darwin proposed the theory of evolution [7], phylogenetic trees have been used to represent the evolutionary history of a set of species. These phylogenetic trees are schematic visualisations of the paths via which species have evolved away from some common ancestor [2]. A phylogenetic tree is a connected directed acyclic graph. See Figure 1.1 for an example of a phylogenetic tree. At the root of the graph is the common ancestor. From this root, edges are directed outward, indicating genetic lineages. The paths branch at tree-nodes which correspond to speciation events. A speciation event is an evolutionary occurence at a point in time, when a group in a population has become so genetically isolated through mutation that it can be identified as a new species. At the furthest point from the root are the leaves. These are nodes which only have a single incoming edge and no outgoing edges and they represent the extant species, whose evolutionary history the tree is meant to depict. These phylogenetic trees can in fact be used to visualize the evolutionary history of any set of taxa, a set of distinguishable elements which have diverged from some common ancestor through a process of mutation. An interesting example is the history of words and languages.

These phylogenetic trees have aided in the classification and identification of species, and they have helped researchers investigate patterns of adaptation, evolutionary innovations, and the processes that drive speciation [6]. They contribute to understanding the origins and spread of infectious diseases, guiding efforts in disease surveillance, treatment, and prevention. They also inform conservation efforts by identifying genetically distinct populations and guiding the development of conservation strategies.

1.2 Phylogenetic networks

As useful as phylogenetic trees have been, in recent decades it has become clear that not all evolutionary histories can be accurately represented by a tree structure [8]. Whenever, the evolutionary history of a set of species involve reticulation events such as hybridization, horizontal gene transfer, and recombination, it is more accurately represented by a network. Such networks contain a different kind of node, called a reticulation, which has two or more edges directed into it, indicating an evolutionary event in which a new species was formed from the combination of two or more species. Reticulations therefore have in-degree at least 2 and out-degree 1. There are various ways to classify

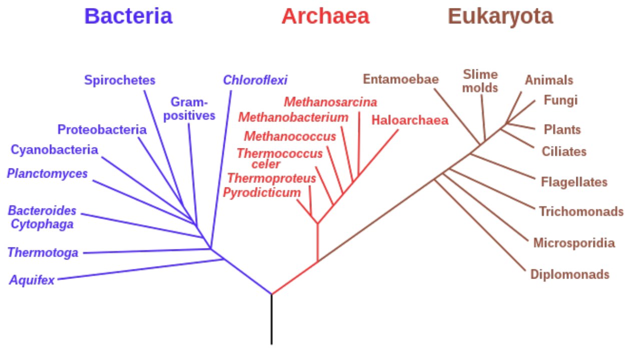


Figure 1.1: A phylogenetic tree showing the three life domains: bacteria, archaea, and eukaryota. The black branch at the bottom of the phylogenetic tree connects the three branches of living organisms to the last universal common ancestor.

phylogenetic networks. A network whose tree-nodes all have out-degree at most 2 and whose reticulations all have in-degree exactly 2 is called *binary*, while a network where tree-nodes have out-degree at most 2 and where reticulations can have in-degree greater than 2 is called *semi-binary*, a network without any restrictions on the out-degree of tree-nodes or the in-degree of reticulations is usually referred to as being *non-binary*. A reticulation with in-degree larger than 2 may indicate uncertainty in the order in which species have hybridized, or it may indicate that more than two species hybridized at the same time.

For most mathematical and algorithmic techniques, the full class of phylogenetic networks is too large. Therefore, several restricted classes of phylogenetic networks have been defined and studied. A network is called *tree-child* if none of its nodes have only reticulation children and a network is *stack-free* if no two reticulations are adjacent. A network is *reticulation-visible* if for each reticulation there exists a leaf such that all paths to this leaf visit the reticulation. These classes have mainly been defined for their nice properties, however, an intuitive biological argument can be made for tree-child networks as well. As long as a species does not go extinct it is highly unlikely that all of its surviving offspring is the result of hybridization. Tree-child networks are automatically stack-free, because if two reticulations are adjacent then one of them must have the other one as their only child. Reticulation-visible networks are stack-free as well. Moreover, any phylogenetic network can be made stack-free by iteratively identifying any two adjacent reticulations. More recently the class of *orchard* networks was introduced as a superclass of the class of tree-child networks with nice characteristics. A natural justification for this class is that orchard networks can be interpreted as trees with additional *horizontal* edges which correspond to hybridizations [10]. An example of a phylogenetic network can be seen in Figure 1.2.

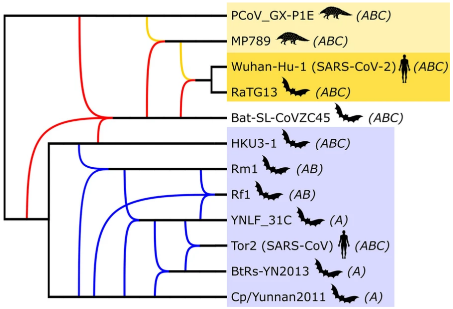


Figure 1.2: A phylogenetic network depicting a hypothetical evolutionary history of a set of corona-viruses from [12].

1.3 Metrics and encodings

Research has shown the scientific value and practical applications of phylogenetic networks [12]. As stated by Cardona et al. in [4], there are numerous practical algorithms for constructing phylogenetic networks from genetic data, yet there are few proposed metrics. A metric serves to quantify the dissimilarity between networks. Finding a metric for phylogenetic networks is necessary to be able to accurately compare and cluster phylogenetic networks. For instance, different reconstruction methods applied to the same genetic sequences, or a single method applied to different sequences, may yield different phylogenetic networks for a given set of species. Therefore, being able to compare the outcomes and determine whether they lie within some reasonable variation of the same network, becomes relevant, as mentioned by Cardona et al. in [4]. To find such a metric, it may be sufficient to find a suitable encoding. An encoding of a phylogenetic network is a way to represent a network by using a certain *building block*, as mentioned by Murakami in [11]. If no two networks within a certain class share the same set of a type of building blocks then that type of building blocks is called an encoding for networks in that class. In this case the networks in this class are uniquely determined by their building blocks. These encodings can then be used to compare networks within this class. Encodings can also serve as inspiration which may lead to algorithms for inferring networks in the specific class, if the building blocks can be generated from data. Moreover, encodings may give deeper insight into the structure and patterns of phylogenetic networks, ultimately deepening our understanding.

1.4 The μ-representation for encoding phylogenetic networks, to apply the symmetric difference as a metric

The μ-representation of a network contains the *path-multiplicity* vectors of its nodes, indicating how many paths there are from each node to each leaf. The μ-representation was originally proposed by Cardona et al. in [4], to serve as an encoding for tree-child networks. With this encoding it is possible to use the cardinality of the symmetric difference of the μ-representations of two networks in the class as a metric. The cardinality of the symmetric difference is per definition symmetric and non-negative, and it satisfies the triangle inequality. Furthermore the symmetric difference of a set with itself is the empty set, which has cardinality zero. However, for the positivity axiom for metrics to hold, namely that the distance between two distinct elements is always positive, we need an encoding result. It is necessary to show that, if two networks have the same μ-representation, then they are the same network.

Cardona et al. showed that encoding holds for tree-child networks. Building upon their work, Erdős, Semple and Steel sought to extend the application of μ-representations to a larger class of phylogenetic networks in [9], which they dubbed orchard networks. They proposed that it was possible to determine certain subgraphs called *cherries* and *reticulated cherries* by the μ-representation. Then, by a process of iteratively simplifying the network, they proposed it would be possible to find a sequence of these (reticulated) cherries from which one could uniquely reconstruct the network. It is important to note that they considered only binary networks. However, some of their findings were later refuted by Bai, Semple and Steel, in [1]. They showed that it is not possible to determine reticulated cherries by the μ-representation for general binary orchard networks. In that paper, Bai, Semple and Steel proposed a *stack-freeness* constraint within the class of orchard networks to establish the encoding result. While they aimed to show encoding holds for semi-binary stack-free orchard networks, their proof only works for networks which are binary, as Murakami showed in [11], by means of a counterexample for semi-binary networks. In [3], Cardona et al. proposed an extended μ-representation, which also takes into consideration the number of paths to reticulations from each node. In the paper they showed that this extended μ-representation is an encoding for binary orchard networks, lifting the stack-free condition. This modification however does not show encoding for semi-binary networks as originally proposed in [1], as the proof is restricted to networks which are binary. It is the aim of this thesis to find the correct restrictions such that encoding holds for the μ-representation on semi-binary orchard networks.

1.5 Contributions

In this thesis we show seven main results. First, we propose a modified μ-representation including the in-degrees of nodes, which is different from the extended μ-representation proposed by Cardona et al. in [3]. Theorem 1 states that this modified μ-representation encodes semi-binary stack-free orchard networks. With this theorem, we can define a metric given by the cardinality of the symmetric difference of the modified μ-representations. We also show that encoding does not hold for non-binary stack-free orchard networks even if the out-degrees are also fixed (Theorem 2).

Furthermore, we present a fundamental equation which governs the relationship between the in-degrees of reticulations and the μ-representation of a network. We prove that such an equation exists (Theorem 3), and show how this gives rise to a system of equations on the μ-vector of the root and the μ-vectors and in-degrees of reticulations. We furthermore show (Theorem 4), that for reticulation-visible networks with fixed reticulation set, the system of equations generated by Theorem 3 has a unique solution.

Then, we define a new class of networks called *strongly reticulation visible* networks, for which there is a *tree-path* (a path containing only tree-nodes including the trivial path for a tree-node) to a bridge from each child of a reticulation. A bridge is an edge which disconnects the network if cut. We show that a bridge and the lowest reticulation ancestor of that bridge in any network is uniquely determined by the μ-representation (Theorem 5 and Lemma 24). We shortly mention that this shows that strongly reticulation visible networks with the same μ-representations have nodes with the same in-degrees (Theorem 6). Finally we conclude (Theorem 7), that strongly reticulation visible semi-binary stack-free orchard networks are encoded in the space of semi-binary stack-free networks by the μ-representation. This means that the cardinality of the symmetric difference of the μ-representations gives a metric between these networks.

1.6 Overview of the thesis

In Section 2, we will go over all the necessary definitions, terms and concepts for understanding the results and the arguments put forth in the rest of the thesis. In the next section, Section 3, we propose our modified μ-representation, which includes the in-degrees and we will prove the first main result of this thesis. In Section 4, we discuss the difficulties in finding the in-degrees of reticulations. Then, we present a fundamental equation which governs the relationship between the in-degrees of reticulations and the μ-representation of a network. In Section 5, we show how the stability of a node is determined by the μ-representation under certain conditions and how this leads to the other main results of this thesis. In Section 6, we summarise the results of the thesis and discuss some further research questions.

2 Preliminaries

All of the results in this thesis relate to *directed acyclic graphs* (DAGs). A DAG is said to be *rooted* when it contains only a single root node of in-degree zero. A rooted DAG whose leaves are bijectively labeled by the elements of a finite set X, we will name an X-DAG. All phylogenetic networks are X-DAGs. From here on out we will identify the leaf nodes with the elements of the set X and no longer make a distinction between the two. Furthermore, we will assume edges are directed unless otherwise mentioned, and from now on, in all figures in this thesis, edges will be directed downward, such that the root is at the top and the leaves are at the bottom

2.1 Rooted directed acyclic graphs on a leaf set X

Let $\mathcal{N} = (V, E)$ be an X-DAG. The in-degree of a node $v \in V$ which we will denote $\delta^-(v)$ is equal to the number of edges which end in v. The out-degree of a node $v \in V$ which we will denote $\delta^+(v)$ is equal to the number of edges starting in v. The degree of a node is the sum of the in-degree and the out-degree. An X-DAG can contain several different types of nodes:

1. A single *root* ρ with in-degree $\delta^-(\rho) = 0$
2. *tree-nodes* v with in-degree $\delta^-(v) \leq 1$
3. *reticulations* r with in-degree $\delta^-(r) \geq 2$
4. and *leaves* a with out-degree $\delta^+(a) = 0$

Note that the root is a tree-node and leaves can be either tree-nodes or reticulations. Nodes which are not leaves are sometimes called *internal* nodes. The set of all reticulations contained in a given X-DAG $\mathcal{N}$ we will denote $R(\mathcal{N})$ or simply R when the X-DAG is obvious from the context. The *hybridization number* $h(\mathcal{N}) = \sum_{r_i \in R}(\delta^-(r_i) - 1)$ of an X-DAG $\mathcal{N}$ is the number of reticulation edges minus the number of reticulations. This can be used as an indicator for how much an X-DAG deviates from a tree.

A path $v_0 \rightsquigarrow v_k$ between two nodes $v_0, v_k \in V$ is a sequence of edges $v_0v_1, v_1v_2, ..., v_{k-1}v_k$ such that $v_iv_{i+1} \in E$ for $i \in \{0, 1, \ldots, k-1\}$. We say the path starts in v_0, ends or terminates in v_k and visits or passes through each node v_i. We assume all paths are directed unless otherwise specified. The *trivial path*, is the path from a node to itself, which contains no edges. We say a node

v_1 is an *ancestor* of another node v_2 if there is a path from v_1 to v_2, in this case v_2 is a *descendant* of v_1. In this case we may also say v_1 is *above* v_2 and v_2 is *below* v_1. If the path consists of a single edge, then we say v_1 is the *parent* of v_2, usually denoted p_{v_2} and v_2 is the *child* of v_1. We also consider the trivial path, therefore each node is both an ancestor and a descendant of itself. The number of paths from v_1 to v_2 we will denote $P_{v_1 v_2}$.

Given a directed edge $e = v_1 v_2$ we call v_2 the *head* of e and v_1 the *tail* of e. We say a node is below e if it is a descendant of v_1 and we say it is above e if it is an ancestor of v_2. We say two nodes are connected if there is an undirected path between them. We say a set of nodes is connected if every pair of nodes in the set is connected. We say a graph is connected if the set of its vertices is connected. An X-DAG is a connected graph.

A *tree-path* is a directed path $v_0 \rightsquigarrow v_k$, such that v_i is a tree-node for each $i \in \{0, 1, \ldots, k\}$. A tree-node v which has out-degree $\delta^+(v) = 1$ we shall call an *elementary* node. A path for which all but the start and end nodes are elementary nodes, we shall call an *elementary path.* The *height* of a node is the length of the longest path from the node to a leaf.

Two X-DAGs $\mathcal{N} = (V, E)$ and $\mathcal{N}' = (V', E')$ are said to be *isomorphic*, denoted by $\mathcal{N} \cong \mathcal{N}'$, when there exists a bijective function $f : V \to V'$ such that $f(a) = a$ for all $a \in X$ and $v_1 v_2 \in E \iff f(v_1) f(v_2) \in E'$ for all $v_1, v_2 \in V$.

2.2 The μ-representation

Given an X-DAG $\mathcal{N} = (V, E)$, the *μ-vector* of any node $v \in V$ is defined as follows: let $\mu(v) \in \mathbb{Z}^X$ be a vector such that the element indexed by leaf a, denoted $\mu_a(v)$, is equal to the number of paths from v to a. Note that $\mu(v)$ only contains non-negative integer elements and is never equal to the zero vector. Furthermore, as for each leaf there is only the trivial path from that leaf to itself, the μ-vectors of leaves are unit-vectors. With the exception of leaf nodes, the μ-vector of a node is always the sum of the μ-vectors of its children. The μ-representation of $\mathcal{N}$, denoted $\mu(\mathcal{N})$, is the multiset of all μ-vectors of nodes in V.

The main difference between a multiset and a set, is that a multiset can contain multiple instances of the same element. The number of instances of an element in a given multiset is called the multiplicity of that element in that multiset. For example, if the μ-representation $\mu(\mathcal{N})$ contains two instances of a vector $\mu(v)$, then we say $\mu(v)$ has multiplicity 2 in $\mu(\mathcal{N})$. We may shorten this to $\#\mu(v) = 2$, whenever the multiset containing $\mu(v)$ is implied. Usually the implied multiset is $\mu(\mathcal{N})$. Then, $\#\mu(v)$ denotes the multiplicity of $\mu(v)$ in $\mu(\mathcal{N})$.

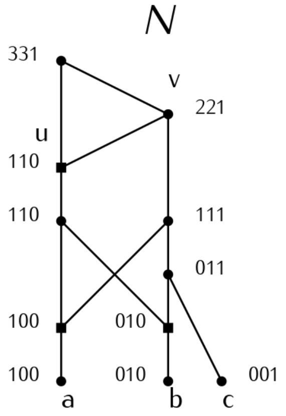


Figure 2.1: An example network $\mathcal{N}$, on leaves a, b and c, with μ-representation $\mu(\mathcal{N}) = \{(100, 2), (010, 2), (001, 1), (110, 2), (011, 1), (111, 1), (221, 1), (331, 1)\}$. Edges are directed downward. Nodes u and v have vectors $\mu(u) = 110$ with multiplicity 2 and $\mu(v) = 221$ with multiplicity 1.

In this thesis, the multisets considered will always be taken in the set of μ-vectors. If a given μ-vector $\mu(v)$ is not contained in a multiset $\mu(\mathcal{N})$, we may say $\mu(v)$ has multiplicity 0 in $\mu(\mathcal{N})$. The operation of removing a μ-vector $\mu(v)$ from $\mu(\mathcal{N})$ is then equivalent to lowering the multiplicity of $\mu(v)$ in $\mu(\mathcal{N})$ by 1. Clearly, the multiplicity of a μ-vector cannot be negative and a μ-vector which has multiplicity 0 in $\mu(\mathcal{N})$ cannot be removed from $\mu(\mathcal{N})$. The operation of adding a μ-vector $\mu(v)$ to a multiset $\mu(\mathcal{N})$ is equivalent to increasing the multiplicity of $\mu(v)$ in $\mu(\mathcal{N})$ by 1. We can say that $\mu(\mathcal{N})$ is generated by adding $\mu(v)$ for each node $v \in V$. Therefore, the multiplicity of a vector $\mu(v)$ in $\mu(\mathcal{N})$ is equal to the number of nodes in $\mathcal{N}$ with μ-vector equal to $\mu(v)$. We do not equate the nodes $v \in V$ with their μ-vectors because multiple nodes may have the same μ-vector. A set of at least two tree-nodes which have the same μ-vector we shall call *tree-clones*. A node which is part of a set of tree-clones, we shall call a tree-clone.

For example, the node u in Figure 2.1 has μ-vector 110, because there is exactly one path to leaf a, one path to leaf b, and there are no paths to leaf c starting in u. There are two instances of nodes with μ-vector 110, because the paths starting in the reticulation are in bijection with the paths starting in its child, by adding or deleting the edge between them. Therefore, $\mu(u)$ has multiplicity 2 in $\mu(\mathcal{N})$. The node v in Figure 2.1 has μ-vector 221, because there are 2 paths to leaf a, one via node u and one via the other child of v, and 2 paths to b and one path to c, starting in v. It should be clear from these examples why, with the exception of leaf nodes, the μ-vector of a node is always the sum of the μ-vectors of its children.

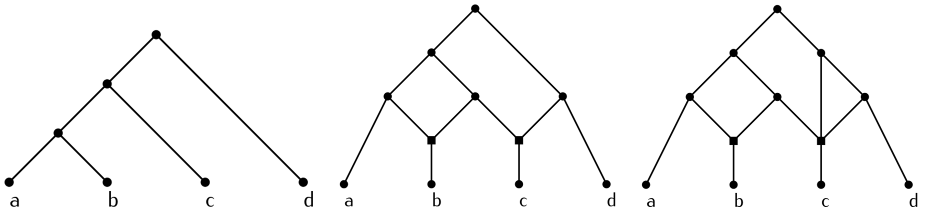


Figure 2.2: A binary phylogenetic tree, a binary phylogenetic network and a semi-binary phylogenetic network

2.3 Phylogenetic networks

A *phylogenetic network* is defined as an X-DAG without parallel arcs or elementary nodes, where the root must be a leaf or have out-degree greater than or equal to 2, reticulations have out-degree 1 and leaves are tree-nodes. A phylogenetic network in which all nodes except the root or the leaves have degree 3 is called binary. A phylogenetic network in which all tree-nodes except the root or the leaves have degree 3 but reticulations can have degree ≥ 3 is called semi-binary. We say a phylogenetic network is non-binary when there are no such added restrictions on the degrees of the nodes. A phylogenetic network which does not contain any reticulations is called a *phylogenetic tree*. See Figure 2.2 for some examples. A phylogenetic network $\mathcal{N}$ is said to be *stack-free* if no reticulation in $\mathcal{N}$ is the child of another reticulation. A phylogenetic network such that for each reticulation there is a leaf for which all paths from the root to this leaf pass through the reticulation is called *reticulation-visible*. The network in Figure 2.1 is binary and stack-free. However, it is not reticulation visible, because there are no leaves such that all paths from the root pass through u. All networks in Figure 2.2 are stack-free and reticulation visible. In Section 5, we will introduce the class of *strongly reticulation-visible networks* as the class of phylogenetic networks, in which there is a tree-path to a bridge from the child of each reticulation.

A *cherry* is an ordered pair of leaves (b, a) which have the same parent. A *reticulated cherry* is an ordered pair of leaves (b, a) such that the parent p_b of b is a reticulation and the parent of a is a tree-node p_a which is also the parent of p_b. A pair (b, a) which is either a cherry or a reticulated cherry is also called a *reducible pair*. *Suppressing* an elementary node is the action of deleting the node and adding an edge between the parent and the child of the node. To *reduce* a cherry in a network $\mathcal{N}$, we delete the leaf b and suppress its parent p_b if it has become elementary. To reduce a reticulated cherry in $\mathcal{N}$ we delete the edge $p_a p_b$ and suppress any nodes which have become elementary. In this way one always obtains another phylogenetic network as the result of reducing a reducible pair in a phylogenetic network.

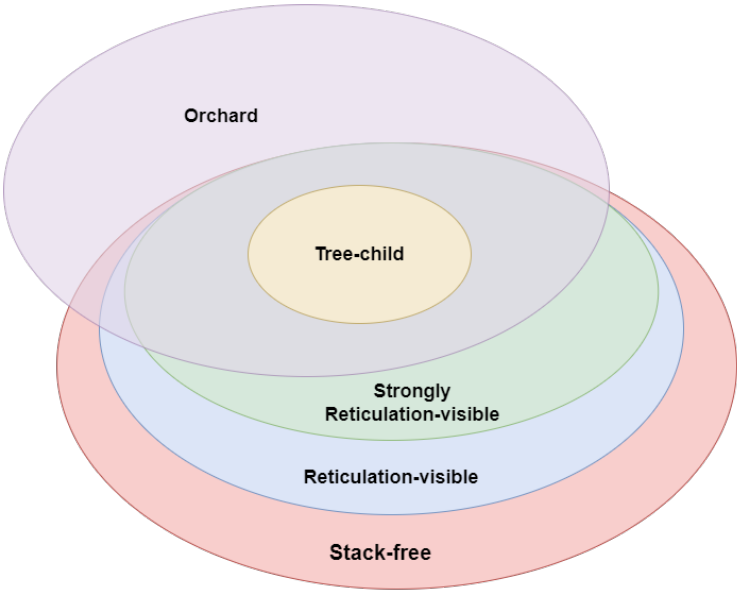


Figure 2.3: A venn diagram showing the relations between several different classes of phylogenetic networks

A network is called *orchard* if there exists a sequence $s_1 s_2 s_3 \dots s_i \dots s_n$, of ordered pairs, such that s_i is a reducible pair in the network after reducing each pair in the sequence up to s_{i-1} and the entire sequence reduces the network to a network on a single leaf. Note that in that case, each network generated by performing reductions s_1 up to s_i, is orchard with sequence $s_{i+1}, s_{i+2}, \dots, s_n$, see Corollary 4.2 in [9]. The network in Figure 2.1 is orchard with sequence $(b,c)(a,c)(b,a)(a,c)(c,a)$. It contains the reticulated cherry (b,c). The networks in Figure 2.1 are all orchard and contain the reducible pair (b,a). In the phylogenetic tree (b,a) is a cherry, while in the other networks (b,a) is a reticulated cherry. Bai, Semple and Steel showed in [1] Lemma 4.4, that orchard networks do not contain tree-clones. See Figure 2.3 for a visualization of the way the different classes of phylogenetic networks discussed in this thesis are related.

2.4 The symmetric difference

The symmetric difference between two sets S_1, S_2 is the set of elements from S_1 and S_2, which are not contained in both sets.

$$S_1 \triangle S_2 = (S_1 \cup S_2) \setminus (S_1 \cap S_2) \tag{2.1}$$

The cardinality of the symmetric difference, or the number of elements that are unique to either set, can be used as a measure for the difference between these two sets. For multisets the symmetric difference is defined somewhat differently. For instance, if we consider the μ-representations $\mu(\mathcal{N}_1)$ and $\mu(\mathcal{N}_2)$ of two networks $\mathcal{N}_1$ and $\mathcal{N}_2$, any μ-vector which has multiplicity i in $\mu(\mathcal{N}_1)$ and j in $\mu(\mathcal{N}_2)$, belongs to $\mu(\mathcal{N}_1)\triangle\mu(\mathcal{N}_2)$ with multiplicity $|i-j|$. The symmetric difference for multisets contains as many instances of a given element as how many more instances of that element are in one set when compared to the other. For the cardinality of the symmetric difference the following hold:

- $|\mu(\mathcal{N}_1)\triangle\mu(\mathcal{N}_2)| \geq 0$,
- $|\mu(\mathcal{N}_1)\triangle\mu(\mathcal{N}_2)| = 0$ if, and only if $\mu(\mathcal{N}_1) = \mu(\mathcal{N}_2)$,
- $|\mu(\mathcal{N}_1)\triangle\mu(\mathcal{N}_2)| = |\mu(\mathcal{N}_2)\triangle\mu(\mathcal{N}_1)|$, and
- $|\mu(\mathcal{N}_1)\triangle\mu(\mathcal{N}_3)| \leq |\mu(\mathcal{N}_1)\triangle\mu(\mathcal{N}_2)| + |\mu(\mathcal{N}_2)\triangle\mu(\mathcal{N}_3)|$.

The second condition is true, because a multiset can be uniquely represented by its multiplicity function, therefore if two multisets have equal multiplicity functions then they are the same multiset. This makes the cardinality of the symmetric difference a distance function or metric on multisets. For it to be a metric on phylogenetic networks however, we need a modified version of the second condition to hold:

- $|\mu(\mathcal{N}_1)\triangle\mu(\mathcal{N}_2)| = 0$ if, and only if $\mathcal{N}_1 \cong \mathcal{N}_2$.

Therefore, in this thesis we will be focusing on determining the conditions on $\mathcal{N}_1$ and $\mathcal{N}_2$ or on μ such that the following holds:

- $\mu(\mathcal{N}_1) = \mu(\mathcal{N}_2)$ if, and only if $\mathcal{N}_1 \cong \mathcal{N}_2$.

It is important to note here that there is always only one μ-representation belonging to a given network.

3 Encoding results for stack-free orchard networks by modified μ-representations

3.1 Introduction

In [1], Bai et al. propose ancestral profiles (which are equivalent to μ-representations) as an encoding for semi-binary stack-free orchard networks in the space of semi-binary stack-free networks. However, in [11] Murakami has shown by means of a counterexample, that μ-representations do not encode semi-binary stack-free orchard networks in the larger class. The question remains whether semi-binary stack-free orchard networks are encoded in their own class by their μ-representations. Bai et al. showed there are equivalent operations on μ-vectors for each type of cherry-reduction, Lemma 4.3 of [1]. Furthermore, it is possible to detect both regular and reticulated cherries by the corresponding μ-vectors, as shown by Erdhős et al. in [9] Corollary 3.2 and Bai et al. Lemma A.1 in [1]. However, while Erdhős assumed binary orchard networks, Bai et al. considered semi-binary stack-free orchard networks, but failed to mention how to determine which cutting method in Lemma 4.3 of [1] to use, which requires knowledge of the in-degree of the reticulation. Despite this, the rest of the arguments set out in [1] still hold. We propose that finding the in-degree of the reticulation which is part of a reticulated cherry would thus be sufficient to show encoding.

3.2 Why μ-representations do not encode semi-binary stack-free orchard networks in the space of semi-binary stack-free phylogenetic networks

In [1] it is proposed that semi-binary stack-free orchard networks are encoded by their μ-representation in the larger class of semi-binary stack-free networks. The proposition is supported by the following arguments: (reticulated) cherries are identifiable by the μ-representation of any semi-binary stack-free phylogenetic network and it is possible to compute the μ-representation of a network obtained by reducing (reticulated) cherries. However, the proof hinges on the claim that if the μ-representations of two networks are the same, then after reducing the same cherry in both networks the μ-representations of the resulting networks will also be the same. Murakami has shown in [11] by a counterexample that this is not true, see Figure 3.1. The two networks $\mathcal{N}$ and $\mathcal{N}'$ in his counterexample

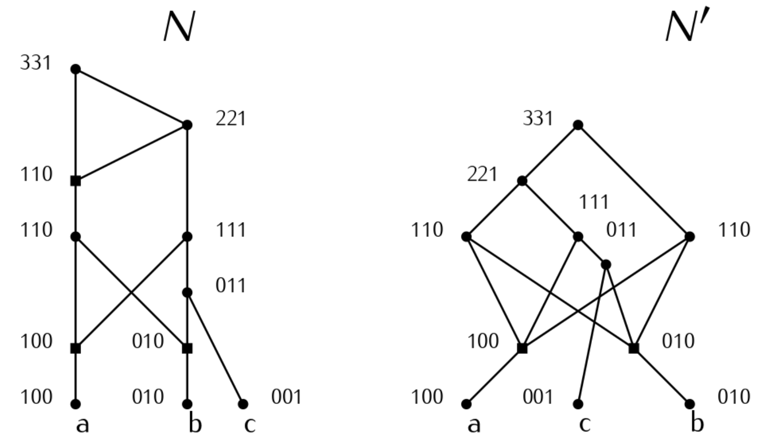


Figure 3.1: Two semi-binary stack-free phylogenetic networks $\mathcal{N}$ and $\mathcal{N}'$ with the same μ-representation, which are not isomorphic. A counterexample to Theorem 3.1 in [1] by Bai et al. put forth by Murakami in their thesis [11].

are both semi-binary stack-free and have the same μ-representation, therefore they both contain the same reticulated cherry (b, c). However, because the in-degrees of the reticulations above leaf b differ, the networks will not have the same μ-representation after reducing (b, c), which implies they are not isomorphic. To be precise, in $\mathcal{N}$ the reticulation above b should be suppressed after cutting, while in $\mathcal{N}'$ it should not be, which means the multiplicity of $\mu(b)$ after cutting will be 1 in $\mathcal{N}$, but it will remain 2 in $\mathcal{N}'$.

3.3 Preliminary lemmas

In this section we will show one of the main results of this thesis. This result is in a way a continuation and modification of previous propositions by Bai et al. [1] and Erdős et al. [9].

We will make use of a modified μ-representation. Let $\mathcal{N} = (V, E)$ be a semi-binary stack-free phylogenetic network.

Definition 1 *Given a μ-vector $\mu(v)$ of a node $v \in V$, the modified μ-vector is*

$$\bar{\mu}(v) = \mu_0(v) \oplus \mu(v)$$

where $\mu_0(v)$ is the in-degree of v, $\mu_0(v) = \delta^-(v)$. The modified μ-representation of a network $\mathcal{N}$, is the multiset $\bar{\mu}(\mathcal{N})$ of modified μ-vectors of nodes in $\mathcal{N}$.

We will also define two types of reticulated cherries:

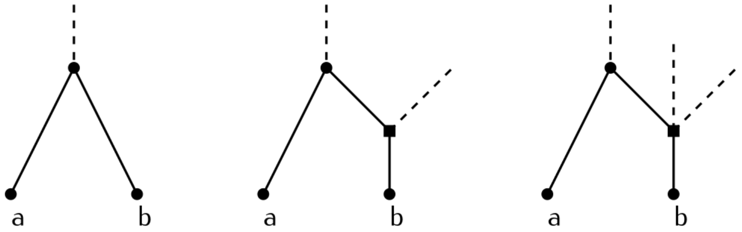


Figure 3.2: A cherry, a simple reticulated cherry and a complex reticulated cherry on the leaves a and b.

Definition 2

- *If (b, a) is a reticulated cherry such that the parent p_b of b is a reticulation with in-degree 2, then (b, a) is a simple reticulated cherry.*
- *If (b, a) is a reticulated cherry such that the parent p_b of b is a reticulation with in-degree greater than 2, then (b, a) is a complex reticulated cherry.*

See Figure 3.2, for examples of a cherry, a simple reticulated cherry and a complex reticulated cherry.

In this entire section, let $a, b \in X$ be leaves of $\mathcal{N}$. We will state some lemmas which will be used to prove the theorem about encoding by the modified μ-representation $\bar{\mu}(\mathcal{N})$.

Lemma 1 *Let a be a leaf. Then, $\mu(a)$ has multiplicity 1 or 2 in $\mu(\mathcal{N})$. If $\#\mu(a) = 1$, then its parent p_a is a tree-node with $\mu(p_a) > \mu(a)$, otherwise its parent is a reticulation with $\mu(p_a) = \mu(a)$.*

Proof: There can be no other tree-node v with $\mu(v) = \mu(a)$, because this tree-node would also be a leaf and the leaves are uniquely labeled by X. Therefore, if there is another node u with $\mu(u) = \mu(a)$ it must be a reticulation. A reticulation has the same μ-vector as its child and in a stack-free network the child of a reticulation is a tree-node, therefore the child of u must be a. A leaf has only a single parent, so there can be only one such reticulation. In conclusion, the number of nodes with the μ-vector $\mu(a)$ is either 1, if the parent of a is a tree-node or 2 if the parent of a is a reticulation. □

Lemma 2 *If $\#\mu(a) = 1$, then the parent of a is a tree-node and its μ-vector is minimal in the set $\{\mu(v) : \mu(v) > \mu(a), v \in V\}$.*

Proof: If $\#\mu(a) = 1$, then by Lemma 1, the parent of a must be a tree-node p_a with $\mu(p_a) > \mu(v)$. Furthermore, for any node v with $\mu(v) > \mu(a)$ there must be a path from v to a. Since a has a single parent, all paths to a from any node other than a must visit p_a. Therefore, each v with $\mu(v) > \mu(a)$ which

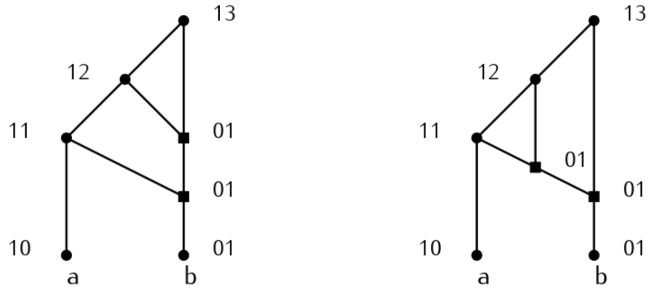


Figure 3.3: Two networks which are not stack-free. Although they have the same μ-representation, they are not isomorphic. In the first network (b, a) is a reticulated cherry while in the second network it is not. The first network is orchard, while the second network is not.

is not p_a must be an ancestor of p_a. This means that, $\mu(v) \geq \mu(p_a)$ for each such node v. In conclusion, the set $\{\mu(v) : \mu(v) > \mu(a)\}$ has a single minimal element, which is $\mu(p_a)$.

Now we will show that cherries and reticulated cherries are uniquely determined by $\mu(\mathcal{N})$.

Lemma 3 *The pair (b, a) is a cherry in $\mathcal{N}$ if, and only if, $\mu_a(v) = \mu_b(v)$ for each $\mu(v) \in \mu(\mathcal{N}) \setminus \{\mu(a), \mu(b)\}$.*

Note that the condition on the μ-vectors implies that $\mu(a)$ and $\mu(b)$ have multiplicity 1 in $\mu(\mathcal{N})$, because if for instance $\mu(b)$ has multiplicity greater than 1 in $\mu(\mathcal{N})$, then $\mu(\mathcal{N}) \setminus \{\mu(a), \mu(b)\}$ would still contain a vector $\mu(b)$, for which $\mu_a(b) = 0 \neq 1 = \mu_b(b)$.

Proof: For the first direction, let us assume the pair (b, a) is a cherry in $\mathcal{N}$ with parent node p. Then, for each node $v \in V \setminus \{a, b\}$ the number of paths from v to either a or b is equal to the number of paths from v to p, so $\mu_a(v) = P_{vp} = \mu_b(v)$.

For the other direction, we will use a proof by contradiction. Assume $\mu_a(v) = \mu_b(v)$ for each $\mu(v) \in \mu(\mathcal{N}) \setminus \{\mu(a), \mu(b)\}$. Now assume the pair (b, a) is not a cherry in $\mathcal{N}$. This means a and b must have different parents $p_a \neq p_b$. However, because $\mu_a(v) = \mu_b(v)$ for each $\mu(v) \in \mu(\mathcal{N}) \setminus \{\mu(a), \mu(b)\}$, we have that $\mu_b(p_a) = \mu_a(p_a) = 1$, therefore there is a path from p_a to b. This means p_a must be an ancestor of p_b. But also $\mu_a(p_b) = \mu_b(p_b) = 1$, therefore p_b must also be an ancestor of p_a. In acyclic graphs two nodes cannot be ancestors of each other unless they are the same node, therefore $p_a = p_b$, but this contradicts our assumption that (b, a) is not a cherry. □

Lemma 4 *The pair (b, a) is a reticulated cherry in $\mathcal{N}$ with b the reticulation leaf if, and only if, $\#\mu(a) = 1$, $\#\mu(b) = 2$, $\mu_b(v) \geq \mu_a(v)$ for each $\mu(v) \in \mu(\mathcal{N}) \setminus \{\mu(a), \mu(b)\}$ and $\mu(\mathcal{N})$ contains a vector $\mu(p_a) = \mu(a) + \mu(b)$.*

Proof: First let us assume (b, a) is a reticulated cherry in $\mathcal{N}$ with b the reticulation leaf. Then the parent of a is a tree node p_a and, by Lemma 1, $\mu(a)$ has multiplicity 1 in the multiset. Also, the parent of b is a reticulation p_b, therefore by Lemma 1, $\mu(b)$ has multiplicity 2 in the multiset. Furthermore, p_a is the parent of p_b and thus an ancestor of b. Therefore, for each path $v \rightsquigarrow p_a$ with $v \in V \setminus \{a, b\}$, there is at least one path $v \rightsquigarrow b$ via p_a. Furthermore, the number of paths from v to a equals the number of paths from v to p_a. This means that $\mu_b(v) \geq P_{vp_a} = \mu_a(v)$ for any node $v \in V \setminus \{a, b\}$. Finally, note that $\mu(p_a) = \mu(a) + \mu(p_b) = \mu(a) + \mu(b)$. This proves the first direction.

For the second direction, we will use proof by contradiction. Let us assume, $\mu(a)$ has multiplicity 1 in the multiset, $\mu(b)$ has multiplicity 2 in the multiset, $\mu_b(v) \geq \mu_a(v)$ for each $\mu(v) \in \mu(\mathcal{N}) \setminus \{\mu(a), \mu(b)\}$ and $\mu(\mathcal{N})$ contains $\mu(p_a) = \mu(a) + \mu(b)$. Now assume (b, a) is not a reticulated cherry. Note $\mu(p_a) > \mu(a)$ and the only μ-vectors $\mu(v)$ with $\mu(v) < \mu(p_a)$ are $\mu(a)$ and $\mu(b)$, thus $\mu(p_a)$ is minimal in $\{\mu(v) : \mu(v) > \mu(a), v \in V\}$. Then, by Lemma 2, $\mu(p_a)$ belongs to the parent of a. Furthermore, by Lemma 1, we know b has a reticulation parent p_b. Therefore the parent p_a of a is not a parent of p_b the parent of b, because otherwise (b, a) would be a reticulated cherry. But there must be a path from p_a to b because $\mu_b(p_a) \geq \mu_a(p_a) = 1$. This means there must be other nodes on the path from p_a to p_b. Let c then be the child of p_a, then $\mu(p_a) = \mu(a) + \mu(c) = \mu(a) + \mu(b)$. Subtracting $\mu(a)$ gives $\mu(c) = \mu(b)$. This means c is either the leaf b, which is not possible, or it is p_b, which we assumed it was not, or it is some other reticulation which has a child with the same μ-vector equal to $\mu(b)$. Its child cannot be the leaf b, because by assumption c is not p_b and its child cannot be p_b because the network is stack-free. Therefore, its child must be a tree-node with the same μ-vector as the leaf b, which is not itself leaf b or p_b. But, then $\#\mu(b) = 3$, which contradicts our assumption. □

Note that if (b, a) is a cherry or a reticulated cherry in $\mathcal{N}$, we say it is a cherry or reticulated cherry in $\mu(\mathcal{N})$ and $\bar{\mu}(\mathcal{N})$. Furthermore, if (b, a) is a reticulated cherry, it is simple if, and only if, $\bar{\mu}(\mathcal{N})$ contains $\bar{\mu}(p_b) = \{2\} \oplus \mu(b)$ ($\bar{\mu}_0(p_b) = 2$, for p_b the parent of b). Otherwise, it is complex. If (b, a) is a cherry or a reticulated cherry in $\mathcal{N}$ we say that (b, a) is a reducible pair in $\mathcal{N}$, in $\mu(\mathcal{N})$ and in $\bar{\mu}(\mathcal{N})$.

It is important to mention here that being able to identify reticulated cherries in the μ-representation is the reason for the stack-free restriction on the networks. The conditions in Lemma 4 are not sufficient to determine whether (b, a) is a reticulated cherry in general. In Figure 3.3 two networks are displayed which have the same μ-representation but are not isomorphic. One contains a reticulated cherry while the other one does not. In [3], Cardona et al. propose a different extended μ-representation which solves this issue of determining reticulated cherries in networks which contain stacks, thereby lifting the stack-free condition from the reticulated cherry lemma. While this works to prove encoding of binary orchard networks, it does not fix the issues with encoding semi-binary orchard networks. The networks in Figure 3.1 form a counterexample, they have the same extended μ-representation, while they are not isomorphic.

3.4 Reconstructing orchard networks

Now let us define cherry and reticulated cherry reductions in $\bar{\mu}(\mathcal{N})$. Recall that to remove a vector from a multiset means to lower the multiplicity by 1 to a minimum of 0. If a vector has multiplicity 0 in a multiset we say the multiset does not contain the vector. By removing element $\bar{\mu}_b(v)$ from $\bar{\mu}(v)$ we mean to change $\bar{\mu}(v)$ to $(\bar{\mu}_i(v))_{i\in\{0\}\cup X\setminus\{b\}}$, which means to project $\bar{\mu}(v)$ on $\mathbb{Z}^{\{0\}\cup X\setminus\{b\}}$. By subtracting $\bar{\mu}_a(v)$ from $\bar{\mu}_b(v)$, we mean the operation of changing $\bar{\mu}(v)$ to $[\bar{\mu}_0(v), \bar{\mu}_a(v), \bar{\mu}_b(v) - \bar{\mu}_a(v), \mu_c(v), \ldots]$.

Let (b, a) be a cherry in $\mathcal{N}$. We define the cherry reduction of (b, a) in $\bar{\mu}(\mathcal{N})$ as the following operations:

1. Remove $\bar{\mu}(b)$ from $\bar{\mu}(\mathcal{N})$.
2. Remove $\bar{\mu}(p_{ab}) = \{1\} \oplus (\mu(a) + \mu(b))$ from $\bar{\mu}(\mathcal{N})$.
3. For each $\bar{\mu}(v) \in \bar{\mu}(\mathcal{N})$, remove $\bar{\mu}_b(v)$ from $\bar{\mu}(v)$.

Note, that because (b, a) is a cherry, the parent p_{ab} of a and b will have $\bar{\mu}$-vector $\{1\} \oplus (\mu(a) + \mu(b))$ before reduction. Note also that none of the in-degrees of any nodes have changed. Now let (b, a) be a simple reticulated cherry in $\bar{\mu}(\mathcal{N})$ we define the simple reticulated cherry reduction of (b, a) as the following operations:

1. Remove $\bar{\mu}(p_a) = \{1\} \oplus (\mu(a) + \mu(b))$ from $\bar{\mu}(\mathcal{N})$.
2. Remove $\bar{\mu}(p_b) = \{2\} \oplus \mu(b)$ from $\bar{\mu}(\mathcal{N})$.
3. For each $\bar{\mu}(v) \in \bar{\mu}(\mathcal{N})$ subtract $\bar{\mu}_a(v)$ from $\bar{\mu}_b(v)$.

Note that tree-nodes have in-degree 1 and so, by Lemma 4, $\bar{\mu}(p_a)$ is the $\bar{\mu}$-vector of the parent of a, which should be suppressed when reducing (b, a). Furthermore, because (b, a) is a simple reticulated cherry, the parent p_b of b has in-degree 2 before reducing and should be suppressed as well. Note that the in-degrees of any nodes that are not suppressed have not changed. Finally, we define the complex reticulated cherry reduction as follows:

1. Remove $\bar{\mu}(p_a) = \{1\} \oplus (\mu(a) + \mu(b))$ from $\bar{\mu}(\mathcal{N})$.

2. Let $\bar{\mu}(p_b) \in \bar{\mu}(\mathcal{N})$ be the vector with $\bar{\mu}_0(p_b) > 1$ and $\bar{\mu}(p_b) = \bar{\mu}_0(p_b) \oplus \mu(b)$ and lower $\bar{\mu}_0(p_b)$ by 1.

3. For each $\bar{\mu}(v) \in \bar{\mu}(\mathcal{N})$ subtract $\bar{\mu}_a(v)$ from $\bar{\mu}_b(v)$.

For this reduction we keep the $\bar{\mu}$-vector of the parent p_b of b, because it is not suppressed when (b, a) is reduced in $\mathcal{N}$, because it has in-degree greater than 1 after reduction, but we do lower its in-degree by 1. Note that, by Lemma 1, in stack-free networks there can only be one non-leaf node with μ-vector equal to $\mu(b)$ and therefore $\bar{\mu}(p_b)$ has multiplicity 1 in $\bar{\mu}(\mathcal{N})$. Finally note that the in-degrees of any other nodes, besides p_b have not changed.

Lemma 5 *Let (b, a) be a reducible pair in $\bar{\mu}(\mathcal{N})$, the multiset generated by reducing (b, a) in $\bar{\mu}(\mathcal{N})$ is the $\bar{\mu}$-representation of the network generated by reducing (b, a) in $\mathcal{N}$.*

Proof: Let $\mathcal{N}' = (V', E')$ be the network generated from $\mathcal{N}$ by reducing (b, a) and let $\bar{\mu}'(\mathcal{N})$ be the multiset generated from $\bar{\mu}(\mathcal{N})$ by reducing (b, a). We will show that $\bar{\mu}(\mathcal{N}') = \bar{\mu}'(\mathcal{N})$.

First let us assume that (b, a) is a cherry in $\mathcal{N}$. Let p_{ab} be the parent of a and b in $\mathcal{N}$. Note that because b is deleted when reducing (b, a) in $\mathcal{N}$, the leaf set of $\mathcal{N}'$ is $X' = X \setminus \{b\}$, therefore the elements of $\bar{\mu}(\mathcal{N}')$ are elements of the space $\mathbb{Z}^{\{0\} \cup X \setminus \{b\}}$, as are the elements of $\bar{\mu}'(\mathcal{N})$. Furthermore, because $b \notin X'$, $\bar{\mu}(b)$ is not an element of $\bar{\mu}(\mathcal{N}')$ nor is it an element of $\bar{\mu}'(\mathcal{N})$ because the multiplicity of $\bar{\mu}(b)$ was 1 in $\bar{\mu}(\mathcal{N})$ before reduction. Moreover, the parent p_{ab} is suppressed, therefore $\bar{\mu}(\mathcal{N}')$ contains one less instance of $\bar{\mu}(p_{ab})$ as does $\bar{\mu}'(\mathcal{N})$. Also note that the in-degrees of all nodes in $\mathcal{N}'$ are equal to the in-degrees of the corresponding nodes in $\mathcal{N}$. Finally, for any nodes $v \in V'$, the number of paths to any leaves other than b is not changed by reducing (b, a). So given any leaf $c \neq b$, $\bar{\mu}_c(v)$ in $\bar{\mu}(\mathcal{N})$ is equal to $\bar{\mu}_c(v)$ in $\bar{\mu}(\mathcal{N}')$ and also $\bar{\mu}_c(v) = \bar{\mu}'_c(v)$. Therefore $\bar{\mu}(\mathcal{N}') = \bar{\mu}'(\mathcal{N})$.

Now let us assume that (b, a) is a simple reticulated cherry in $\mathcal{N}$, where b has a reticulation parent p_b. Note that p_a and p_b are suppressed when reducing (b, a) in $\mathcal{N}$, therefore $\bar{\mu}(\mathcal{N}')$ will contain one less instance of $\bar{\mu}(p_a)$ and of $\bar{\mu}(p_b)$ as does $\bar{\mu}'(\mathcal{N})$. Furthermore, because edge (p_a, p_b) is deleted when reducing (b, a) in $\mathcal{N}$, for any node $v \in V'$ the set of paths starting from v and ending in b will be the set of paths starting in v and ending in b in $\mathcal{N}$ minus the paths which visit p_a. Note that there is only one path from p_a to b, therefore the number of paths from v to b which visit p_a is equal to the number of paths from v to p_a, which is equal to $\bar{\mu}_a(v)$. Therefore $\bar{\mu}_b(v)$ in $\mathcal{N}'$ equals $\bar{\mu}_b(v) - \bar{\mu}_a(v)$ in $\mathcal{N}$. As the paths to other leaves remain unchanged by reducing (b, a) in $\mathcal{N}$ and no other elements of vectors have been changed by reducing (b, a) in $\bar{\mu}(\mathcal{N})$, we can conclude that $\bar{\mu}(\mathcal{N}') = \bar{\mu}'(\mathcal{N})$.

Finally, let us assume that (b, a) is a complex reticulated cherry in $\mathcal{N}$. The only difference as compared to the case where (b, a) was a simple reticulated cherry is the fact that when reducing (b, a) in $\mathcal{N}$ the parent p_b of b is not suppressed but its in-degree is lowered by 1. Therefore, $\bar{\mu}(p_b)$ is not removed from $\bar{\mu}(\mathcal{N})$ but $\bar{\mu}_0(p_b)$ is lowered by 1. All other arguments still hold, thus also in this case $\bar{\mu}(\mathcal{N}') = \bar{\mu}'(\mathcal{N})$. □

With these lemma's we have the following result for any two phylogenetic networks $\mathcal{N}_1 = (V_1, E_1)$ and $\mathcal{N}_2 = (V_2, E_2)$.

Theorem 1 *Let $\mathcal{N}_1$ be semi-binary stack-free orchard and let $\mathcal{N}_2$ be semi-binary stack-free. Then,*

$$\bar{\mu}(\mathcal{N}_1) = \bar{\mu}(\mathcal{N}_2) \iff \mathcal{N}_1 \cong \mathcal{N}_2 \tag{3.1}$$

Note that this means that semi-binary stack-free orchard networks are encoded by their modified μ-representation in the class of semi-binary stack-free networks.

Proof: Suppose we are given a semi-binary stack-free orchard network $\mathcal{N}_1$, and a semi-binary stack-free network $\mathcal{N}_2$ with $\bar{\mu}(\mathcal{N}_1) = \bar{\mu}(\mathcal{N}_2)$. Note that $\bar{\mu}(\mathcal{N}_1) = \bar{\mu}(\mathcal{N}_2)$ implies that also $\mu(\mathcal{N}_1) = \mu(\mathcal{N}_2)$. Then, because $\mathcal{N}_1$ is orchard it must contain a reducible pair of leaves (b, a). If the pair (b, a) is a cherry, then by Lemma 3 it must be a cherry in $\mu(\mathcal{N}_1)$. Therefore, it is also a cherry in $\mu(\mathcal{N}_2)$ and thus again by Lemma 3 it is a cherry in $\mathcal{N}_2$. In this case, if $\mathcal{N}_1'$ is the network generated by reducing (b, a) in $\mathcal{N}_1$, and $\bar{\mu}(\mathcal{N}_1')$ is the μ-representation of this network, then by Lemma 5, $\bar{\mu}'(\mathcal{N}_1) = \bar{\mu}(\mathcal{N}_1')$, where $\bar{\mu}'(\mathcal{N}_1)$ is the multiset generated by reducing (b, a) in $\bar{\mu}(\mathcal{N}_1)$. Thus, because there is only a single way of reducing a cherry in the $\bar{\mu}$-representation, we have that $\bar{\mu}'(\mathcal{N}_1) = \bar{\mu}'(\mathcal{N}_2)$ and by Lemma 5, $\bar{\mu}'(\mathcal{N}_2)$ is the $\bar{\mu}$-representation $\bar{\mu}(\mathcal{N}_2')$ of the network generated by reducing (b, a) in $\mathcal{N}_2$. To conclude, after reducing the cherry (b, a) in both $\mathcal{N}_1$ and $\mathcal{N}_2$, the two networks still have the same $\bar{\mu}$-representation.

Alternatively, if (b, a) is a reticulated cherry in $\mathcal{N}_1$ then by Lemma 4, it is a reticulated cherry in $\mu(\mathcal{N}_1)$. Therefore, by the same argument as before, it is a reticulated cherry in $\mathcal{N}_2$. Furthermore, because $\bar{\mu}(\mathcal{N}_1) = \bar{\mu}(\mathcal{N}_2)$, if (b, a) is simple in $\mathcal{N}_1$ then it is simple in $\mathcal{N}_2$ and otherwise it is complex in both networks. As for each type of reticulated cherry there is a single way of reducing it in the $\bar{\mu}$-representation, we again end up with two networks with the same modified $\bar{\mu}$-representation.

Moreover, because $\mathcal{N}_1$ is orchard, the network will still be orchard after reducing the pair (b, a). Therefore, it will again contain a reducible pair which is also a reducible pair in $\mathcal{N}_2$. It follows, that any sequence $S = s_1, s_2, \ldots, s_n$ of reducible pairs s_i, which reduces $\mathcal{N}_1$ (to a network on a single leaf), will also be a sequence of reducible pairs for $\mathcal{N}_2$. Furthermore, because $\mathcal{N}_1$ and $\mathcal{N}_2$ start out with the same set of leaves and each cherry reduction removes the same leaf from both networks, S will also reduce $\mathcal{N}_2$ to a network on a single leaf, and it will be the same leaf. We will show that $\mathcal{N}_1$ and $\mathcal{N}_2$ are isomorphic by an inductive proof. Let $\mathcal{N}_1^{(i)}$ and $\mathcal{N}_2^{(i)}$ be the networks generated from $\mathcal{N}_1$ and $\mathcal{N}_2$ by performing reductions s_1 up to s_i, and let $\mathcal{N}_1^{(0)} = \mathcal{N}_1$ and $\mathcal{N}_2^{(0)} = \mathcal{N}_2$. And let us assume that the networks $\mathcal{N}_1^{(i)}$ and $\mathcal{N}_2^{(i)}$ are isomorphic. This is true for the base case where $i = n$, such that $\mathcal{N}_1$ and $\mathcal{N}_2$ are both reduced to a network on a single leaf by the entire sequence $s_1, s_2, \ldots, s_n$. Now take the networks $\mathcal{N}_1^{(i-1)}$ and $\mathcal{N}_2^{(i-1)}$ generated by performing reductions $s_1, s_2, \ldots, s_{i-1}$. By Corollary 3 in [11], there is exactly one way to generate $\mathcal{N}_1^{(i-1)}$ and $\mathcal{N}_2^{(i-1)}$ from $\mathcal{N}_1^{(i)}$ and $\mathcal{N}_2^{(i)}$, respectively ($1a$ if s_i is a cherry and $2b$ if it is a reticulated cherry). From this it follows that, because $\mathcal{N}_1^{(i)}$ and $\mathcal{N}_2^{(i)}$ are isomorphic, we also have that $\mathcal{N}_1^{(i-1)}$ and $\mathcal{N}_2^{(i-1)}$ are isomorphic. Finally, because we have shown that the networks are isomorphic for $i = n$ and that they are isomorphic for $i = j - 1$ if they are isomorphic for $i = j$, we can conclude that they are isomorphic for $i = 0$. This means $\mathcal{N}_1$ and $\mathcal{N}_2$ are isomorphic. □

3.5 The $\bar{\mu}$-distance as a metric

By the definition as set out in chapter 2, the symmetric difference between two multisets is empty, if, and only if, they are the same multiset. Furthermore, because the μ-representation of a network is well-defined, if two networks are isomorphic their μ-representations are equal. If however, two networks have equal μ-representation, this does not necessarily mean they are isomorphic, see the examples in Figure 3.1 and Figure 3.3. Theorem 1 shows that given two semi-binary stack-free networks with equal modified μ-representations, if one of them is orchard, then they are isomorphic. This means that, in a sample of semi-binary stack-free networks, any orchard network can be compared to every other network in the sample using the cardinality of the symmetric difference of the modified μ-representations as a distance metric. So if we define the $\bar{\mu}$-distance $d_{\bar{\mu}}(\mathcal{N}_1, \mathcal{N}_2) = |\bar{\mu}(\mathcal{N}_1) \triangle \bar{\mu}(\mathcal{N}_2)|$, for $\mathcal{N}_1, \mathcal{N}_2$ semi-binary stack-free and $\mathcal{N}_2$ orchard, then $d_{\bar{\mu}}$ is a metric.

3.6 Non-binary stack-free orchard networks

In this section we will discuss whether our encoding results for the modified μ-representation extend to non-binary orchard networks. First, we suppose Lemma 1 regarding the parents of leaf nodes and the multiplicity of the μ-vectors of leaf nodes, and Lemma 2 regarding the μ-vector of the tree-node parent of a leaf node, hold for non-binary stack-free networks without any further modification. Similarly, we suggest Lemma 3 holds for non-binary stack-free networks and therefore cherries are uniquely determined by the μ-representation for non-binary stack-free networks.

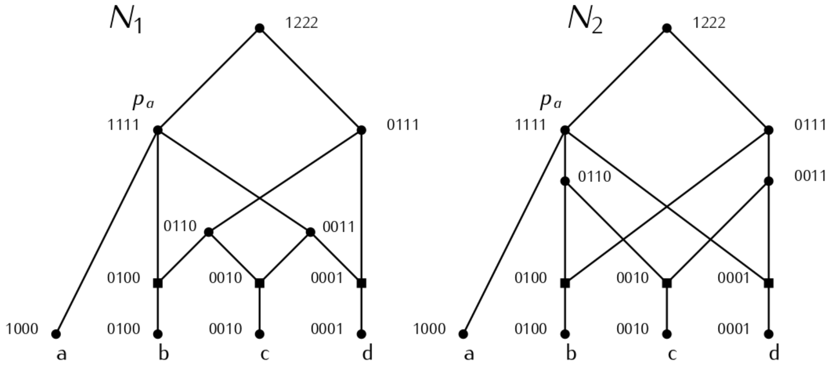


Figure 3.4: The networks $\mathcal{N}_1$ and $\mathcal{N}_2$ are both non-binary stack-free orchard with the same $\bar{\mu}$-representation and equal out-degrees, however they are non-isomorphic. In $\mathcal{N}_1$, (b, a) is a reticulated cherry, while in $\mathcal{N}_2$ it is not. Similarly in $\mathcal{N}_2$, (d, a) is a reticulated cherry, while in $\mathcal{N}_1$ it is not. This situation arises because the parent p_a of leaf a has out-degree 3. Notably p_a is the only node with degree greater than 3.

When trying to modify Lemma 4 for non-binary networks, we run into problems. When considering whether the leaf pair (b, a) is a reticulated cherry, we can no longer require the existence of $\mu(p_a) = \mu(a) + \mu(b)$, because the parent of a may have more children than just a and p_b. By Lemma 2 we can find the μ-vector of the parent of a, but if it is not equal to $\mu(a) + \mu(b)$, then it is impossible to determine whether there are any other nodes on the path $p_a \rightsquigarrow p_b$. Figure 3.4 displays two non-binary stack-free orchard networks with the same $\bar{\mu}$-representation, which are not isomorphic. Because the parent p_a of a has three children and its μ-vector $\mu(p_a)$ is equal to the sums of the μ-vectors of 2 different sets of 3 nodes ($1111 = 1000 + 0100 + 0011 = 1000 + 0110 + 0001$) of which all those that differ belong to reticulations, it is impossible to tell which set belongs to the children of p_a and therefore whether (b, a) is a reticulated cherry or not. As a consequence of the example given in Figure 3.4 we obtain the following:

Theorem 2 *Non-binary stack-free orchard networks are not encoded by their $\bar{\mu}$-representation.*

Note that nodes with the same $\bar{\mu}$-vectors in $\mathcal{N}_1$ and $\mathcal{N}_2$ also have the same out-degrees. This means that the logical extension of the modified μ-representation by adding the out-degrees of nodes does not lead to an encoding result for the class of non-binary stack-free orchard networks.

4 Determining the in-degrees of reticulations from μ-representations

4.1 Introduction

In [4], Cardona et al. showed that the μ-representation serves as an encoding for non-binary tree-child phylogenetic networks. However, we've shown that for more general networks, even if they are stack-free semi-binary, two non-isomorphic networks may have the same μ-representation. See Figure 3.1 for an example. We've also shown that two semi-binary stack-free networks which have the same μ-representation are isomorphic, as long as one is orchard and their nodes have the same in-degrees. We therefore wish to determine under which conditions the in-degrees are determined by the μ-representation. In the following sections we will first consider the lemmas which hold for tree-child networks, and show why they don't necessarily hold in the context of semi-binary stack-free orchard networks. We will then consider some new lemmas which do hold in the context of semi-binary stack-free orchard networks. Finally, we will propose an equation for the in-degrees in terms of the μ-representation of non-binary X-DAGs.

4.2 Identifying the parents of a node

One way we could try to determine the in-degree of a reticulation is to determine which nodes are parents. In [4] Cardona et al. showed that for tree-child networks it is possible to identify the children of any node by their μ-vectors. However for more general orchard networks part b of Lemma 5 does not hold. See the orchard network in Figure 4.1 for a counterexample. Note that the network in Figure 4.1 is stack-free, so adding this restriction does not make the lemma true. Note also that this network contains a node z such that $\mu(z) = \mu(y)+\mu(b) = \mu(x)+\mu(c)$. The issue in identifying the parents of the reticulation above b, is that it is not immediately evident from the μ-vectors which nodes are the children of z. However, in this particular case the issue is easily solved, because c is a leaf, we can find their parent. To show that this is true, we first define a partial order $\succeq$ on V_μ as in [4], such that $\hat{\mu}(x) = (\mu(x), i_x)$:

$$\begin{gathered}(\mu(x), i_x) \succeq (\mu(y), i_y) \iff \\ \mu(x) > \mu(y) \text{ with respect to the product partial order, or } \mu(x) = \mu(y) \text{ and } i_x < i_y\end{gathered} \tag{4.1}$$

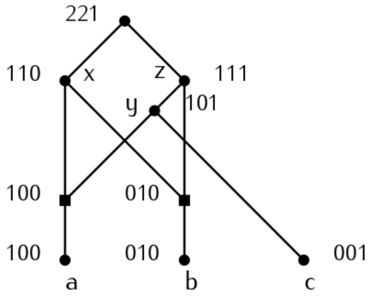


Figure 4.1: A semi-binary stack-free orchard network, which shows Lemma 5b of [4] does not hold in general for networks in this class.

Where we let i_v be the position of node v in the elementary path of all nodes with μ-vector $\mu(v)$. For stack-free networks in particular, such a path can only be two nodes long and as Murakami showed in [11], the multiplicity of any μ-vector in $\mu(\mathcal{N})$ for a semi-binary stack-free orchard network $\mathcal{N}$ is at most 2. So in this case if $\mu(u) = \mu(v)$ and $i_u < i_v$, then u is a reticulation whose only child is v. Let $\mathcal{N} = (V, E)$ be a stack-free semi-binary phylogenetic network. Then, we have the following lemma.

Lemma 6 *Let a be a leaf node, for any node $x \in V$ if $\hat{\mu}(x) \succ \hat{\mu}(a)$ and for all $v \in V$, $\hat{\mu}(v) \not\succ \hat{\mu}(a)$ or $\hat{\mu}(v) \succeq \hat{\mu}(x)$, then x is the parent of a.*

Proof: Take a node x which is not the parent of leaf node a. Assume $\hat{\mu}(x) \succ \hat{\mu}(a)$ and for all $v \in V$, $\hat{\mu}(v) \not\succ \hat{\mu}(a) \vee \hat{\mu}(v) \succeq \hat{\mu}(x)$. First note, $\hat{\mu}(x) \succ \hat{\mu}(a)$ implies $\mu_a(x) > 0$, therefore there is a path from x to a. Furthermore, because x is not a parent of a, the length of the path from x to a must be greater than 1. This means there must be a different node c on the path from x to a. Then, because c lies on a path to a, we must have that $\hat{\mu}(c) \succ \hat{\mu}(a)$. And because c is a descendant of x we have that $\hat{\mu}(c) \not\succeq \hat{\mu}(x)$. Which contradicts our assumptions. □

The previous lemma can be read as a modified version of Lemma 6 from [4]. While Lemma 6 in [4] considered all children of internal nodes, Lemma 6 above only considers parents of leaf nodes. We can extend Lemma 6 to include parents of reticulations with a leaf child.

Lemma 7 *Let a be a leaf and r_a its reticulation parent, such that $\mu(r_a) = \mu(a)$ and $\hat{\mu}(r_a) \succ \hat{\mu}(a)$. For any node $x \in V$, if $\hat{\mu}(x) \succ \hat{\mu}(r_a)$ and for all nodes $v \in V$, $\hat{\mu}(v) \not\succ \hat{\mu}(a)$ or $\hat{\mu}(v) \succeq \hat{\mu}(x)$. Then, x is a parent of r_a.*

Proof: Let x be any node which is not a parent of r_a and assume that $\hat{\mu}(x) \succ \hat{\mu}(r_a)$ and for all nodes $v \in V$, $\hat{\mu}(v) \not\succ \hat{\mu}(a)$ or $\hat{\mu}(v) \succeq \hat{\mu}(x)$. The first assumption, $\hat{\mu}(x) \succ \hat{\mu}(r_a)$ implies that $\mu(x) \geq \mu(r_a)$ and as $\mu(r_a) = \mu(a)$, this implies $\mu_a(x) > 0$. Therefore, there is a path from x to the leaf a. And as

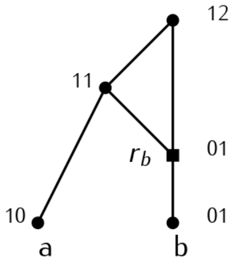


Figure 4.2: A semi-binary stack-free orchard network containing a triangle

r_a is the only parent of a this means that the path from x to a must go through r_a and therefore there is also a path from x to r_a. However, because x is not a parent of r_a, this means there must exist a different node c on the path from x to r_a. As c is therefore an ancestor of r_a we have $\hat{\mu}(c) \succ \hat{\mu}(r_a)$ and by extension $\hat{\mu}(c) \succ \hat{\mu}(a)$. However, because c is a descendant of x we also have $\hat{\mu}(c) \not\succeq \hat{\mu}(x)$, which leads to a contradiction. □

From Lemma 7 we can draw the following conclusion:

For every reticulation r with a leaf child, the set:
$M_r = \{x : \hat{\mu}(x) \succ \hat{\mu}(r)\}$ has minimal elements and all of them are parents of r.

Our example in Figure 4.1, shows that the set of minimal elements of M_r does not necessarily contain *all* parents of r. Take the reticulation above the leaf b (let's call it r_b). The node x is a minimal element of the set M_{r_b} but not z, because $\mu(z) \succ \mu(x)$, even though z is a parent of the reticulation. The number of minimal elements of M_r therefore gives a lower bound on the in-degree of reticulation r. Another example for which it is not satisfied with equality, is when we have a so-called triangle as in Figure 4.2. In this example the root is a parent of r_b the reticulation parent of leaf b, but it is not a minimal element of M_{r_b}. Clearly, for any reticulation r with a leaf child, the set M_r does contain all ancestors of r and in a semi-binary orchard network a tree-node can only have two children. Therefore, the reticulations for which this lower bound is not tight are summed up by these two cases:

- There exists a node $x \in M_r$ which has children r and y, where y is an ancestor of r.
- or, there exists a node $x \in M_r$ which has children y and z such that $\mu(y) + \mu(z) = \mu(r) + \mu(w)$ for some other node w in the network.

The node x in the second case we shall call an *ambiguous node*. The first case is easily solved, as long as the second one does not also hold for x. If for a node x, there exists exactly one node y such that $\mu(x) = \mu(r) + \mu(y)$. Then, we know x must be the parent of r and y, because if x is a tree-node and there are exactly two μ-vectors in the multiset whose sum is equal to $\mu(x)$ then x must be the parent of the corresponding nodes.

4.3 Preliminary lemmas

As we have shown in the last section, determining the parents of a reticulation is difficult. However, to show encoding for orchard networks we just need to determine the in-degree. Because the in-degree of a reticulation equals the number of paths from parents of a reticulation to that reticulation, the number of paths from the root could give us information. To show how this information can be used, we start first with a few lemmas for rooted directed acyclic graphs on a leaf set X (X-DAG).

In this section, we will state a few lemmas we will use to prove Theorem 3. Let $\mathcal{N} = (V, E)$ be a non-binary X-DAG and let all nodes be elements of V. From [5] we recall the following lemma.

Lemma 8 *Let there be a path $u \rightsquigarrow v$ from a node u to a node v. If $u \rightsquigarrow v$ is a tree-path, then every path $w \rightsquigarrow v$ ending in v is either contained in $u \rightsquigarrow v$ or contains $u \rightsquigarrow v$.*

Proof: First, let us assume there is a tree-path $u \rightsquigarrow v$ such that u, v and all other nodes on this path are tree-nodes. Now assume there is another path $w \rightsquigarrow v$ ending in v. Let w_i be the first node on the path $w \rightsquigarrow v$, such that all nodes $w_i, w_{i+1}, \ldots, v$ on the path $w \rightsquigarrow v$ are contained in $u \rightsquigarrow v$. Then w_i is a tree-node, which has only a single edge directed to it. Therefore, either $w_i = w$, in which case $w \rightsquigarrow v$ is contained in $u \rightsquigarrow v$, or $w_i = u$ in which case $u \rightsquigarrow v$ is contained in $w \rightsquigarrow v$. Otherwise, the parent of w_i is also in both paths, which contradicts our assumption on w_i. $\Box$

Lemma 9 *For any node v, there is either a tree-path from the root ρ to v or there is a lowest reticulation r, such that all paths $\rho \rightsquigarrow v$ pass through r.*

Proof: First note, if there is a tree-path from ρ to v, then by Lemma 8 there can be no other path $\rho \rightsquigarrow v$ and therefore v has no reticulation ancestors. Now assume there does not exist a tree-path from ρ to v. Then v has at least a single reticulation ancestor. Let r_l be the lowest reticulation ancestor above v, such that r_l has no other reticulation descendants which are ancestors of v. This r_l exists as we only consider finite graphs. Then either r_l is equal to v, in which case it is clear that all paths from ρ to v visit r_l, or there is a tree-path from the child c_l of r_l to v. Therefore, by Lemma 8, all paths from ancestors of c_l to v must contain $c_l \rightsquigarrow v$. Furthermore, c_l is a tree-node, therefore r must be its only parent and therefore all paths from ancestors of r to v which contain c_l must also contain r. The root is an ancestor of r. Thus, all paths $\rho \rightsquigarrow v$ pass through r. $\Box$

Corollary 1 *If there is a tree-path from the child of a reticulation r to a node v, then all reticulation ancestors of v are ancestors of r.*

Lemma 10 *Given a reticulation r with no reticulation ancestor, the number of paths from the root to r is equal to the the in-degree of r.*

$$P_{\rho r} = \delta^{-}(r) \tag{4.2}$$

Proof: Because r has no reticulation ancestor, for each parent of r there must be a tree-path from the root to the parent and so by Lemma 8, there is exactly one path from the root to each parent. The number of paths from the root to the reticulation r via a parent p_r is thus equal to the number of paths from the parent to r. Therefore, the total number of paths from the root to r is equal to the sum of paths from parents of r to r, which is precisely the in-degree of r. □

Lemma 11 *Given a leaf a, whose set of ancestors contains exactly one reticulation r, the in-degree of r is equal to the number of paths from the root to a.*

$$\mu_a(\rho) = \delta^{-}(r) \tag{4.3}$$

Proof: Because a has no other reticulation ancestor, either r is equal to a, or there must be a tree-path between the child of r and a, so by Lemma 8, there is exactly one path from r to a. Also note that by Lemma 9 there can be no path from the root to the leaf a which does not pass through the reticulation r. Furthermore, by Lemma 10 the number of paths from the root to r is equal to the in-degree of r. Therefore, the number of paths from the root to a, equals the number of paths from the root to r, which is exactly the in-degree of r. □

4.4 An equation relating the in-degrees of reticulations and μ-vectors

As we have seen in the previous section, when there is exactly one reticulation r above a certain leaf a, then we have $\mu_a(\rho) = \delta^{-}(r)$. The following theorem expands this to any number of reticulations and is one of the main results of this thesis.

Theorem 3 *Let $\mathcal{N}$ be a rooted directed acyclic graph on a leaf set X and let a be an element of X. Let R be the set of reticulations in V. Then,*

$$\mu_a(\rho) = \sum_{r_i \in R} (\delta^{-}(r_i) - 1)\mu_a(r_i) + 1.$$

Note, $\mu_a(r_j) = 1$ whenever r_j is the lowest reticulation above a. Also note that $\mu_a(r_i) = 0$ for any reticulation r_i which is not an ancestor of a, therefore these don't contribute to the sum.

Proof: We prove the theorem by induction on the hybridization number, $k = h(\mathcal{N})$. We start by considering the base case, $k = 0$.

Let us consider an X-DAG with $k = 0$. If the hybridization number is zero then the graph contains no reticulations. This means there is a tree-path from the root to each leaf and by Lemma 8 there is exactly one path from the root to each leaf, and thus $\mu_a(\rho) = 1$. This shows the equation holds for each leaf.

Now suppose the equation holds for each leaf of any X-DAG with hybridization number lower than k. Let $\mathcal{N}$ be an X-DAG with reticulation set R such that $h(\mathcal{N}) = k$. Without loss of generality let r_j be a highest reticulation in $\mathcal{N}$, which means r_j has no reticulation ancestors. We can decrease the in-degree of r_j by deleting an incoming edge ur_j. If u is an elementary node we delete all edges on the maximal elementary path which visits u, including the nodes which become isolated by doing so. When we delete any of the incoming edges ur_j in this way the resulting network $\mathcal{N}'$ is still an X-DAG. Furthermore, the in-degrees of any other reticulations in the network have not decreased, which means $\delta^-_{\mathcal{N}'}(r_i) = \delta^-_{\mathcal{N}}(r_i)$ for any $i \neq j$. And also the number of paths from any reticulation r to any leaf a has not changed so $\mu'_a(r) = \mu_a(r)$. Finally, the hybridization number of $\mathcal{N}'$ is equal to $k - 1$ and by the induction hypothesis we have,

$$\mu'_a(\rho) = \sum_{r_i \in R'} (\delta^-_{\mathcal{N}'}(r_i) - 1)\mu_a(r_i) + 1.$$

Now there are two cases to consider:

1. $\delta^-_{\mathcal{N}}(r_j) = 2$
2. $\delta^-_{\mathcal{N}}(r_j) > 2$.

In the first case, the in-degree of r_j after deleting an incoming edge becomes one, which means $r_j \notin R'$. In this case $\delta^-_{\mathcal{N}}(r_j) - 2 = 0$. Which gives us:

$$\begin{aligned}\mu'_a(\rho) &= \sum_{r_i \in R'} (\delta^-_{\mathcal{N}'}(r_i) - 1)\mu_a(r_i) + 1 \\ &= \sum_{r_i \in R \setminus \{r_j\}} (\delta^-_{\mathcal{N}}(r_i) - 1)\mu_a(r_i) + (\delta^-_{\mathcal{N}}(r_j) - 2)\mu_a(r_j) + 1\end{aligned}$$

In the second case $r_j \in R'$ and, because we did not decrease the in-degree of any other reticulations, $R = R'$. The only difference is, $\delta^-_{\mathcal{N}'}(r_j) = \delta^-_{\mathcal{N}}(r_j) - 1$.

Therefore, we have:

$$\begin{aligned}
\mu'_a(\rho) &= \sum_{r_i \in R'} ((\delta^-_{\mathcal{N}'}(r_i) - 1)\mu_a(r_i) + 1 \\
&= \sum_{r_i \in R} ((\delta^-_{\mathcal{N}'}(r_i) - 1)\mu_a(r_i) + 1 \\
&= \sum_{r_i \in R \setminus \{r_j\}} (\delta^-_{\mathcal{N}}(r_i) - 1)\mu_a(r_i) + (\delta^-_{\mathcal{N}'}(r_j) - 1)\mu_a(r_j) + 1 \\
&= \sum_{r_i \in R \setminus \{r_j\}} (\delta^-_{\mathcal{N}}(r_i) - 1)\mu_a(r_i) + (\delta^-_{\mathcal{N}}(r_j) - 2)\mu_a(r_j) + 1.
\end{aligned}$$

Note that after simplification the equation is the same for both cases. When we add the edge ur_j (or the maximal elementary path which visits u and ends in r_j) back to the network, we generate $\mathcal{N}$ from $\mathcal{N}'$. In doing so we increase $\mu'_a(\rho)$ by $\mu_a(r_j)$ for any leaf a. It is easy to see this is true, because r_j is chosen to be a highest reticulation in $\mathcal{N}$ and by Lemma 8 there is a single path from the root to r_j which uses the edge (u, r_j) and by definition there are $\mu_a(r_j)$ paths from r_j to any leaf a. This means that, for any leaf a

$$\begin{aligned}
\mu_a(\rho) &= \mu'_a(\rho) + \mu_a(r_j) \\
&= \sum_{r_i \in R \setminus \{r_j\}} (\delta^-(r_i)_{\mathcal{N}} - 1)\mu_a(r_i) + (\delta^-(r_j)_{\mathcal{N}} - 2)\mu_a(r_j) + 1 + \mu_a(r_j) \\
&= \sum_{r_i \in R \setminus \{r_j\}} (\delta^-(r_i)_{\mathcal{N}} - 1)\mu_a(r_i) + (\delta^-(r_j)_{\mathcal{N}} - 1)\mu_a(r_j) + 1 \\
&= \sum_{r_i \in R} (\delta^-(r_i)_{\mathcal{N}} - 1)\mu_a(r_i) + 1,
\end{aligned}$$

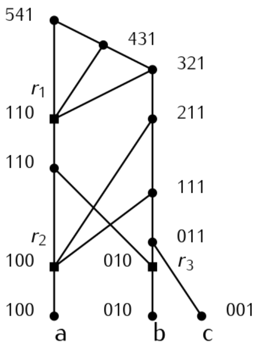
which means the equation in the theorem holds for any leaf of $\mathcal{N}$. Thus, we have shown that if the equation holds for any leaf of an X-DAG with hybridization number $k-1$, then it holds for any leaf of an X-DAG with hybridization number k. We had already shown the equation holds for any leaf for the base case $k = 0$. Therefore, we can conclude the equation holds for any leaf of any X-DAG, which proves the theorem. □

Note that this theorem holds for non-binary X-DAGs, as we never assumed a limit on the in-degree of reticulations or the out-degree of tree-nodes. Furthermore, the theorem does not have to be stated in terms of leaves or reticulations, but can be stated more directly as:

$$\mu(\rho) = \sum_{v_i \in V \setminus \{\rho\}} (\delta^-(v_i) - 1)\mu(v_i) + 1,$$

or,

$$0 = \sum_{v_i \in V} (\delta^-(v_i) - 1)\mu(v_i) + 1.$$



Equation for leaf a:

$$\mu_a(\rho) = 5 = (\delta^-(r_1) - 1)\underset{=1}{\mu_a(r_1)} + (\delta^-(r_2) - 1)\underset{=1}{\mu_a(r_2)} + 1$$

$$6 = \delta^-(r_1) + \delta^-(r_2)$$

Equation for leaf b:

$$\mu_b(\rho) = 4 = (\delta^-(r_1) - 1)\underset{=1}{\mu_b(r_1)} + (\delta^-(r_3) - 1)\underset{=1}{\mu_b(r_3)} + 1$$

$$5 = \delta^-(r_1) + \delta^-(r_3)$$

$$\delta^-(r_1) \geq 2, \delta^-(r_2) \geq 2, \delta^-(r_3) \geq 2$$

$$(\delta^-(r_1), \delta^-(r_2), \delta^-(r_3)) = (2, 4, 3) \vee (3, 3, 2)$$

Figure 4.3: A semi-binary stack-free orchard network for which the system of equations generated by applying Theorem 3 to each leaf does not have a unique solution. However, the solution $(\delta^-(r_1), \delta^-(r_2), \delta^-(r_3)) = (3, 3, 2)$ is the unique one that belongs to this network.

Theorem 3 provides us with a system of linear equations that govern the in-degrees of reticulations as a function of μ-vectors. We know this system of equations must have a solution as long as it belongs to a valid phylogenetic network. However, there are no guarantees yet that it has a unique solution. A network may contain more reticulations than there are linearly independent equations in the system. The network displayed in Figure 4.3 is semi-binary stack-free orchard, yet the system of equations generated by applying Theorem 3 to each leaf does not have a unique solution. Note that leaf c has no reticulation ancestors and therefore the equation for leaf c $(1 = 1)$ does not contribute. This means there is one equation for leaf a and one for leaf b, but three variables, the in-degrees of reticulations r_1, r_2 and r_3. Note that the lower bound for the in-degree of reticulations is 2, and therefore, there are only a finite number of solutions. In this case there are two solutions to the system of equations and Figure 4.3 shows $(\delta^-(r_1), \delta^-(r_2), \delta^-(r_3)) = (3, 3, 2)$ is the one belonging to this network.

4.5 In-degrees of stable reticulations

In the last section we presented a theorem which allows us to derive a system of equations which govern the in-degrees of reticulations as a function of the μ-vectors. This system of equations applies to every network with the same μ-representation and set of reticulations R. However, it is important to note the system of equations does not always have a unique solution. In this section we will introduce specific conditions under which it does have a unique solution. We can extend the results of the previous section to the class of networks called reticulation-visible. Recall the definition of a reticulation-visible network:

Definition 3 *An X-DAG with reticulation set R is called reticulation-visible if each reticulation $r \in R$ is stable, that is there exists a leaf $a \in X$ such that all paths from the root to a visit r.*

If we let $\mathcal{N}$ be a reticulation-visible X-DAG with reticulation set R the following is true. For $r \in R$, with leaf a below r such that all paths from the root to a visit r, and v any ancestor of r,

$$\mu_a(v) = P_{vr}\mu_a(r), \tag{4.4}$$

where P_{vr} is the number of paths from v to r. This is easy to see, as every path from v to a must be the product of a path from v to r and a path from r to a. With this we gain the following lemma.

Lemma 12 *Let $\mathcal{N}$ be a reticulation-visible X-DAG with reticulation set R. Let r_ℓ be any reticulation in R and let A be the set of ancestors of r_ℓ. Finally let $a \in X$ be a leaf such that all paths from the root to a visit r_ℓ. Then,*

$$\frac{\mu_a(\rho)}{\mu_a(r_\ell)} = \sum_{r \in A}(\delta^-(r) - 1)\frac{\mu_a(r)}{\mu_a(r_\ell)} + 1 \tag{4.5}$$

Proof: Let us generate $\mathcal{N}'$ from $\mathcal{N}$ by first attaching a new leaf a' to r_ℓ, by adding the edge (r_ℓ, a'), and then adjusting the μ-representation by adding a column for a', such that $\mu'(v) = \mu(v) \oplus \mu_{a'}(v)$. Note that this network is now an X'-DAG, where $X' = X \cup \{a'\}$. Then by Theorem 3 we have:

$$\mu'_{a'}(\rho) = \sum_{r \in R}(\delta^-(r) - 1)\mu'_{a'}(r) + 1. \tag{4.6}$$

Now note that r_ℓ is the lowest reticulation above a' by construction and therefore the only reticulations that contribute to the sum in Equation 4.6 are the reticulations in A. Furthermore, the number of paths from any ancestor v of a', which is not a', to a' are equal to the number of paths from v to r_ℓ, which is the same in $\mathcal{N}$ and $\mathcal{N}'$, which means by Equation 4.4,

$$\mu'_{a'}(v) = P_{vr_\ell} = \frac{\mu_a(v)}{\mu_a(r_\ell)}. \tag{4.7}$$

Because ρ is an ancestor of a', as are the elements of A, we can substitute Equation 4.7 in Equation 4.6 to get:

$$\frac{\mu_a(\rho)}{\mu_a(r_\ell)} = \sum_{r \in A}(\delta^-(r) - 1)\frac{\mu_a(r)}{\mu_a(r_\ell)} + 1.$$

□

For the following lemmas, let $\mathcal{N}_1$ and $\mathcal{N}_2$ be two reticulation-visible X-DAG's with $\mu(\mathcal{N}_1) = \mu(\mathcal{N}_2)$. Let both networks have the same reticulation set R and the same leaf set X. Finally, assume that for each reticulation $r \in R$, there exists a leaf $a \in X$ such that r is stable with respect to a in both networks.

Lemma 13 *Each reticulation $r \in R$ has the same reticulation descendants and the same reticulation ancestors in $\mathcal{N}_1$ and $\mathcal{N}_2$.*

Proof: Let $r \in R$ be stable for leaf a in both networks. Then all paths from the root to a visit r in both networks. Therefore, all reticulations $r_i \in R$, with $\mu(r_i) \geq \mu(a)$ must be on a path from the root to a which also visits r in both networks. Therefore, each $r_i \in R$, with $\mu(r_i) \geq \mu(a)$ must be either a descendant or an ancestor of r. If $\mu(r_i) \geq \mu(r)$ then it must be an ancestor in both networks, and if $\mu(r_i) \leq \mu(r)$ it must be a descendant in both networks. □

Lemma 14 *Let $H \subseteq R$ be the subset of R which contains only reticulations without other reticulation ancestors in either network. Then the reticulations in H have the same in-degrees in both networks.*

Proof: For any reticulation $r \in H$ the set A in Lemma 12 contains only r and therefore the in-degree of r is given directly by Equation 4.5, applied to the leaf for which the reticulation is stable in both networks. □

By a similar reasoning we obtain the following:

Lemma 15 *Let r_ℓ in R and let the in-degrees of the other ancestors of r_ℓ be the same in $\mathcal{N}_1$ and $\mathcal{N}_2$. Then the in-degree of r_ℓ is the same in $\mathcal{N}_1$ and $\mathcal{N}_2$.*

Proof: Given a reticulation $r_\ell \in R$, the set A in Equation 4.5 contains only r_ℓ and its other ancestors, which are the same in both networks. Now assume the in-degrees of the ancestors of r_ℓ are fixed, then the in-degree of r_ℓ is given by Equation 4.5, applied to the leaf for which the reticulation is stable in both networks. □

This leads to the following conclusion:

Theorem 4 *Let $\mathcal{N}_1$ and $\mathcal{N}_2$ be two reticulation-visible X-DAG's with $\mu(\mathcal{N}_1) = \mu(\mathcal{N}_2)$. Let both networks have the same reticulation set R and the same leaf set X. Finally, assume that for each reticulation $r \in R$, there exists a leaf $a \in X$ such that r is stable with respect to a in both networks. Then the in-degrees of the reticulations are the same in both networks.*

Proof: Let H be the set of highest reticulations in R, which is the same set for $\mathcal{N}_1$ and $\mathcal{N}_2$, by Lemma 13. By Lemma 14 the in-degrees of the reticulations in H are the same in both networks. Let S be the set of second highest reticulations in R, such that each reticulation $r \in S$ only has ancestors in H. By applying Lemma 15 the in-degrees of the reticulations in S are also the same in both networks. Now, in the same way, the in-degrees of the third highest reticulations are equal and so on and so forth. This process must terminate as we only consider finite graphs. Thus, by repeated application of Lemma 15, we see that the in-degrees of all reticulations in R are the same in both networks. □

5 Determining stable nodes from μ-representations

5.1 Introduction

Theorem 4 shows that if two reticulation-visible networks have equal μ-representations, identical reticulation sets, and every reticulation is stable for at least one common leaf in both networks, then the in-degrees of the reticulations are guaranteed to be equal in both networks. These conditions ensure that the same system of equations that govern the in-degrees of reticulations holds for both networks and it has a unique solution.

Therefore, to find the conditions under which the in-degrees of two networks with the same μ-representation are the same. We need to determine under which conditions the reticulation sets are equal and the reticulations are stable for common leaves. In the case of stack-free orchard networks the reticulations correspond to μ-vectors with multiplicity 2. However, even if they share the same μ-representation, the stability of reticulations with respect to specific leaves may still differ between the networks. Therefore, our focus will be on demonstrating the conditions under which the stability of a node with respect to a leaf a is determined by the μ-representation.

5.2 Preliminary lemmas

Let $\mathcal{N} = (V, E)$ be a phylogenetic network. Let all nodes be elements of V. Recall that a tree-clone is a tree-node for which there exists another tree-node with the same μ-vector.

Lemma 16 *Tree-clones are not stable.*

Proof: Let u, v be distinct tree-clones. Recall that for tree-nodes we have $\mu(u) > \mu(v)$ whenever $u \neq v$ and u is an ancestor of v. Therefore, because $\mu(u) = \mu(v)$, it is clear that u, v are neither ancestors or descendants of each other. This means a path never visits both u and v. Furthermore, because the μ-vectors are equal, we know that for each leaf, such that u is on a path to that leaf, v is also on a path to that leaf. By our previous statement these paths must be distinct and neither contains both nodes. Therefore, there are no leaves such that all paths to that leaf contain u, nor are there any leaves such that all paths to that leaf contain v. $\square$

Lemma 17 *A reticulation is stable with respect to a leaf if, and only if, its child is stable with respect to that leaf.*

Proof: In phylogenetic networks, all paths from the root to a leaf which visit a reticulation must also visit its child, as there is a single edge leaving a reticulation and a reticulation is not a leaf. □

Corollary 2 *Nodes with the same μ-vector as tree-clones are not stable.*

Either they are themselves tree-clones, or they are the reticulation parent of a tree-clone.

Lemma 18 *Let $\mu(\mathcal{N})$ be the μ-representation of a stack-free network and let $\mu(v) \in \mu(\mathcal{N})$ be a μ-vector, such that there does not exist a pair of tree-clones whose μ-vectors are equal to $\mu(v)$. Then $\mu(v)$ has multiplicity at most 2 in $\mu(\mathcal{N})$.*

Proof: Assume there are no tree-clones with μ-vectors equal to $\mu(v)$, but assume $\mu(v)$ has multiplicity more than 2. Then there must be at least 2 distinct reticulations $r_1, r_2 \in R$ with the same μ-vector $\mu(r_1) = \mu(r_2) = \mu(v)$. As reticulations have the same μ-vectors as their children and in a stack-free network the children of reticulations are tree-nodes, this means there exist two tree-nodes with μ-vectors equal to $\mu(v)$. However, as we assumed there are no tree-clones with μ-vectors equal to $\mu(v)$, we've reached a contradiction. □

Corollary 3 *Given a stack-free network $\mathcal{N}$, with μ-representation $\mu(\mathcal{N})$. If $\mu(v)$ has a multiplicity greater than 2 in $\mu(\mathcal{N})$ then all nodes $v \in V$ with μ-vector equal to $\mu(v)$ are not stable.*

Instead of showing directly when the μ-representation determines whether a μ-vector belongs to a stable node, we will first show some other results. A *bridge* or a *cut-edge* is an edge, for which it holds that if the edge would be deleted the number of connected components of the graph goes up. In the case of phylogenetic networks deleting a bridge makes the graph no longer connected.

Lemma 19 *An edge directed to a reticulation is never a bridge.*

Proof: There is a path from the root to any parent of a reticulation. A reticulation has at least two parents. If the edge from one of the parents to a reticulation is deleted, then there is still a path from the root to the reticulation via another parent. Therefore the reticulation and all its descendants are still connected to the root. All other nodes are still connected to the root as well, therefore the graph is still connected. □

Let the head of a bridge be called a *bridge-node.* Then, by the lemma above, bridge-nodes are tree-nodes. All leaves are automatically bridge-nodes, as they become isolated whenever the edge directed into them is deleted. The root is not a bridge-node because it has zero in-degree.

Lemma 20 *Let v_b be a tree-node other than the root. Then, v_b is a bridge-node if, and only if, all paths from ancestors of v_b to leaves below v_b pass through v_b.*

This lemma states, in other words, that a bridge-node is stable for all leaves below it. This implies that if v_b is a bridge-node, then nodes which lie on a path to a leaf below v_b are either ancestors or descendants of v_b.

Proof: Let us first assume that v_b is a bridge-node and there is a path $u \rightsquigarrow a$ from an ancestor u of v_b to a leaf a below v_b which does not pass through v_b. Then, after deleting the edge from the parent of v_b to v_b, there are still paths from v_b to all descendants of v_b, so all descendants of v_b are connected. Furthermore, there is still a path from the root to a, which is the product of a path from the root to u and the path $u \rightsquigarrow a$ which does not visit v_b. Therefore, a is still connected to the root and as all descendants of v_b are still connected to each other, this means all descendants are still connected to the root as well. Therefore, the graph is still connected. However, this contradicts our assumption that v_b is a bridge-node. Therefore, such a path cannot exist.

Now let us assume all paths from ancestors of v_b to leaves below v_b pass through v_b. As v_b is a tree-node, there is a single edge directed to v_b. Therefore, if the edge directed to v_b is deleted, there are no longer any paths from ancestors of v_b to leaves below v_b, except those that start in v_b. The root is an ancestor of v_b other than v_b. Therefore, if the edge directed to v_b is deleted, there are no longer any paths from the root to leaves below v_b, thus the graph is disconnected. This means v_b is a bridge-node. □

Corollary 4 *Let v_b be a bridge-node. Then, for $u \in V$*

- $\mu(u) \geq \mu(v_b) \iff \mu(u)$ *belongs only to ancestors of v*
- $\mu(u) < \mu(v_b) \iff \mu(u)$ *belongs only to descendants of v*

Previously these implications only held in one direction.

Lemma 21 *If there is a tree-path from a node v to a bridge-node v_b, then v is stable with respect to all leaves below v_b.*

Proof: Let there be a tree-path from a node v to a bridge node v_b. Then, by Lemma 8 all paths from ancestors of v to v_b pass through v. By Lemma 20, all paths from the root to leaves below v_b pass through v_b. This means that all paths from the root to leaves below v_b are a composition of a path from the root to v_b and a path from v_b to leaf below it. The root is an ancestor of v, therefore all paths from the root to v_b are a composition of a path from the root to v and the tree-path from v to v_b. Therefore all paths from the root to leaves below v_b are a composition of a path from the root to v, the tree-path from v to v_b and a path from v_b to a leaf below it. In conclusion all paths from the root to leaves below v_b pass through v, and therefore v is stable with respect to all leaves below v_b. □

Lemma 22 *If there is a tree-path from the child c_r of a reticulation r to a bridge-node v_b, then r is stable with respect to all leaves below v_b.*

Proof: Let us assume there is a tree-path from the child c_r of a reticulation r to a bridge-node v_b. By Lemma 21, c_r is stable with respect to all leaves below v_b. Then, by Lemma 17, r is stable with respect to all leaves below v_b. □

Let us define the set A_b as the subset of the ancestors of a bridge node v_b, whose μ-vectors have multiplicity 2 in $\mu(\mathcal{N})$. For any set $S \subseteq V$ let $\mu(S)$ be the multiset of μ-vectors of the nodes in S. Then $\mu(A_b)$ is exactly the set $\{\mu(v) : \mu(v) \geq \mu(v_b), \#\mu(v) = 2\}$ and by Corollary 4, each vector in $\mu(A_b)$ only belongs to ancestors of v_b.

Lemma 23 *If $\mu(v)$ minimal in $\mu(A_b)$, then it belongs to a reticulation and to the child of that reticulation.*

Proof: A μ-vector with multiplicity 2 either belongs to a pair of tree-clones or to a reticulation and its child. Now, if $\mu(v_1) \in \mu(A_b)$ belongs to a pair of tree-clones v_1, v_2 with $\mu(v_1) = \mu(v_2)$, then there cannot be a tree-path from either of them to a bridge node v_b, because by Lemma 16, tree-clones are not stable. Therefore, v_1 and v_2 must have reticulation descendants r_1, r_2, who are ancestors of v_b. Note that r_1 could be equal to r_2. If the multiplicity of either r_1 or r_2 is greater than 2, then their child must be a tree-clone, by the contrapositive of Lemma 18. By the same argument there must then be other reticulation descendants of v_1 and v_2 above v_b. We only consider finite graphs, and therefore, w.l.o.g. we can assume $\mu(r_1)$ and $\mu(r_2)$ have multiplicity 2 in $\mu(\mathcal{N})$. Then, $r_1, r_2 \in A_b$, with $\mu(r_1) < \mu(v_1)$ and $\mu(r_2) < \mu(v_2)$. Therefore, $\mu(v_1)$ is not minimal in A_b. We can conclude that if $\mu(r)$ is minimal in A_b, then r is a reticulation. □

Lemma 24 *There is a tree-path from the child of a reticulation r to a bridge-node v_b if, and only if, $\mu(r)$ is minimal in the set $\mu(A_b)$.*

Proof: For the first direction, assume there is a tree-path from the child of a reticulation r to a bridge v_b. Then, by Lemma 22, r is stable. Therefore, by Corollary 3, $\mu(r)$ has multiplicity 2 in $\mu(\mathcal{N})$. Furthermore, r is an ancestor of v_b, therefore $r \in A_b$. Also note, that by Lemma 8 and Lemma 9, r is the lowest reticulation above v_b. Then, by combining Lemma 9 and Lemma 23, all ancestors of v_b, with μ-vectors which have multiplicity 2, are ancestors of r. Thus, we have $\mu(v) \geq \mu(r)$, for $\mu(v) \in \mu(A_b)$. Therefore, $\mu(r)$ is minimal in A_b. This proves the first direction.

Now assume $\mu(r)$ is minimal in $\mu(A_b)$. By Lemma 23, $\mu(r)$ belongs to a reticulation r and its child c_r. Furthermore, because $\mu(c_r) = \mu(r) \in \mu(A_b)$, both r and c_r are ancestors of v_b. Therefore, there is a path from c_r to v_b. If the path from c_r to v_b is not a tree-path, then r is not the lowest reticulation

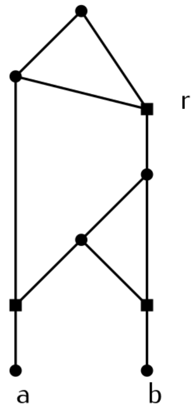


Figure 5.1: A phylogenetic network which is reticulation-visible but not strongly reticulation-visible. Reticulation r is stable with respect to leaf b, but there is no tree-path from the child of r to a bridge. The only bridges are the edges ending in a and b.

above v_b. Let r_ℓ be the lowest reticulation above v_b, then by Lemma 9, r_ℓ is a descendant of r. By Lemma 24, r_ℓ is stable. Therefore, by Corollary 3, $\#\mu(r_l) \leq 2$. The child of r_ℓ has the same μ-vector, therefore $\#\mu(r_l) = 2$. This means that $\mu(r_\ell) \in \mu(A_b)$ and because r_ℓ is a descendant of r, also $\mu(r) > \mu(r_\ell)$. This contradicts our assumption that $\mu(r)$ is minimal in $\mu(A_b)$. Therefore, r is the lowest reticulation above v_b, and the path from c_r to v_b is a tree-path. □

5.3 Strongly reticulation-visible networks

The lemmas in the previous section show that if we know that a given μ-vector belongs to a bridge-node, then we can find the μ-vector of the lowest reticulation above this bridge-node and the reticulation will be stable for all leaves below the bridge node. We therefore propose to consider the class of networks such that for each reticulation there is a tree-path from its child to a bridge. We will call this the class of *strongly reticulation-visible networks.* Note that all strongly reticulation-visible networks are reticulation-visible. But there are reticulation-visible networks which are not strongly reticulation-visible. See Figure 5.1 for an example. What remains to be shown is whether it is possible to know if a given μ-vector belongs to a bridge-node.

By Lemma 20, bridge-nodes are stable. Therefore, if v_b is a bridge-node, then by Corollary 3, $\mu(v_b) \leq 2$. Furthermore, by Lemma 16, $\mu(v_b)$ does not belong to a pair of tree-clones. All leaves are bridge-nodes, therefore all unit vectors in $\mu(\mathcal{N})$ belong to some bridge-nodes. Now let $\mathcal{N} = (V, E)$ be a semi-binary stack-free network such that all tree-nodes in V have at most 2 children.

Lemma 25 *If v_b is a bridge-node and c_1 and c_2 are its children, then either the μ-vectors of c_1 and c_2 are distinct and at most one has multiplicity greater than 1 in $\mu(\mathcal{N})$ or they are equal with multiplicity exactly 2.*

Proof: We will show the lemma holds by proving the contrapositive: if the μ-vectors of the two children of a tree node are distinct with multiplicity greater than 1 or are equal with multiplicity greater than 2, then their parent is not a bridge node. If the μ-vectors of two nodes c_1, c_2, who are children of the same node, are distinct with multiplicity greater than 1, then one of the following holds:

- c_1 and c_2 are reticulations,
- c_1 and c_2 are tree-clones with different μ-vectors,
- c_1 is a reticulation and c_2 is a tree-clone with a different μ-vector.

If their μ-vectors are equal with multiplicity greater than 2, one of these must hold:

- c_1 and c_2 are reticulations,
- c_1 and c_2 are tree-clones with the same μ-vector and there exists a third tree-clone c_3 with the same μ-vector as c_1 and c_2,
- c_1 is a tree-clone and c_2 is a reticulation with the same μ-vector.

Given a tree-node v_b, let us first assume v_b has two children c_1 and c_2 which are reticulations. Because c_1 and c_2 are reticulations they must have at least one other parent besides v_b. Neither c_1 is the parent of c_2 nor is c_2 the parent of c_1 because the network is stack-free. Note that, not both c_1 and c_2 can have a parent which is a descendant of the other one, because in that case there would be a cycle, from c_1 to the parent of c_2 to c_2 to the parent of c_1 to c_1. Furthermore, all descendants of v_b except v_b itself are descendants of c_1 or c_2. Therefore, either c_1 or c_2 has a parent which is not a descendant of v_b. But then there would be a path from the root to a leaf below v_b via this parent, which does not pass through v_b. Thus, by Lemma 20, this means v_b is not a bridge node.

For the second case let us assume v_b has two children c_1 and c_2 which are tree-clones, with different μ-vectors. In this case there exists a tree-clone c_1' with $\mu(c_1') = \mu(c_1)$ which has a parent which is not v_b. And there must be a tree-clone c_2' with $\mu(c_2') = \mu(c_2)$, which has a parent which is not v_b. It cannot be the case that the parent of c_1' is a descendant of c_2 and the parent of c_2' is a descendant of c_1. Because in that case, $\mu(c_1) = \mu(c_1') \leq \mu(c_2)$ and $\mu(c_2) = \mu(c_2') \leq \mu(c_1)$. Which means $\mu(c_1) = \mu(c_2)$ which contradicts our assumption. Nor can the parent of c_1' be a descendant of c_1, because then $\mu(c_1') < \mu(c_1)$, which contradicts our assumption. The same holds for c'_2 and c_2. This means that either, c'_1 or c'_2 must have a parent which is not a descendant of v_b. W.l.o.g. we can assume c'_1 has a parent which is not a descendant of v_b. Note that, $\mu(c'_1) < \mu(v_b)$ so there must be paths from c'_1 to leaves below v_b. This means that there is a path from the root to a leaf below v_b via c'_1 which does not visit v_b. Thus, v_b is not a bridge-node.

Third, let us assume v_b has one child c_1 which is a reticulation and a child c_2 which is a tree-clone, such that $\mu(c_1) \geq 2$, $\mu(c_2) \geq 2$ and $\mu(c_1) \neq \mu(c_2)$. If c_1 is not a descendant of c_2 then c_1 has another parent which is not a descendant of v_b. This means there must be a path from the root to a leaf below v_b via this parent, which does not pass through v_b. In this case v_b is not a bridge node. Alternatively, let us assume c_1 is a descendant of c_2. Then there must exist tree-clone v with $\mu(c_2) = \mu(v)$ and $\mu(v) > \mu(c_1)$, with a parent which is not a descendant of v_b. Because $\mu(v_b) > \mu(c_2) = \mu(v)$ there must be a path from v to a leaf below v_b and therefore, there must be a path from the root to a leaf below v_b via v, which does not pass through v_b. This means v_b is not a bridge node.

Fourth, let us assume that v_b has two children c_1 and c_2, which are tree-clones with $\mu(c_1) = \mu(c_2)$ and there exists another tree-clone c_3 with $\mu(c_3) = \mu(c_1) = \mu(c_2)$. Note that the parent of c_3 cannot be a descendant of c_1 or c_2, because c_1 and c_2 are tree-nodes and this would mean either $\mu(c_3) < \mu(c_1)$ or $\mu(c_3) < \mu(c_2)$, which we have assumed is not the case. Nor can the parent of c_3 be v_b, because v_b already has 2 children. Therefore, c_3 must have a parent which is not a descendant of v_b and, by the same argument as before, this implies v_b is not a bridge-node.

Finally, let us assume v_b has two children, c_1 which is a tree-node and c_2 which is a reticulation, with $\mu(c_2) = \mu(c_1)$. Then, c_2 must have at least one other parent, which is not v_b. This parent cannot be a descendant of c_1, because c_1 is a tree-node and therefore this would imply that $\mu(c_2) < \mu(c_1)$, which contradicts our assumption. Therefore, we can assume c_2 has a parent which is not a descendant of v_b. Which again implies that v_b is not a bridge-node. □

Lemma 26 *If $\mu(v_b)$ belongs to a bridge-node which is not a leaf then $\mu(\mathcal{N})$ contains exactly one pair of vectors $\mu(x), \mu(y)$ such that $\mu(v_b) = \mu(x) + \mu(y)$.*

Proof: We will show the lemma is true with a proof by contradiction, by showing that there cannot even be a second pair. Assume $\mu(v_b)$ is a bridge-node which is not a leaf and $\mu(\mathcal{N})$ contains at least two pairs $\mu(x), \mu(y)$ and $\mu(k), \mu(\ell)$ such that $\mu(v_b) = \mu(x) + \mu(y) = \mu(k) + \mu(\ell)$. W.l.o.g. we can assume $\mu(x)$ and $\mu(y)$ belong to the children x, y of v_b. By Lemma 25, we know that $\mu(k) \neq \mu(x)$, because otherwise $\mu(\ell) = \mu(y)$, in which case both $\#\mu(x) \geq 2$ and $\#\mu(y) \geq 2$, and if $\mu(x) = \mu(y)$ then $\#\mu(x) \geq 4$. The same goes for $\mu(k) \neq \mu(y)$, $\mu(\ell) \neq \mu(x)$ and $\mu(\ell) \neq \mu(y)$. Note also that none of x, y, k, ℓ are ancestors of v_b because their μ-vectors are each lower than $\mu(v_b)$. Now note that, by Corollary 4, $\mu(v_b) > \mu(k)$ and $\mu(v_b) > \mu(\ell)$ implies they are descendants of v_b. However, as we mentioned x, y are the children of v_b. This means k, ℓ must be descendants of x, y, which are not equal to x, y, because their μ-vectors differ. From this it follows that $\mu(x) + \mu(y) > \mu(k) + \mu(\ell)$. This contradicts our assumption that the sums were equal. □

Lemma 27 *If $\mu(v_t) \in \mu(\mathcal{N})$ belongs to a tree-clone, then one of the following is true:*

- *$\mu(\mathcal{N})$ contains exactly one pair $\mu(x), \mu(y)$ with $\mu(x) \neq \mu(y)$, such that $\mu(v_t) = \mu(x) + \mu(y)$. In which case $\#\mu(x) \geq 2$ and $\#\mu(y) \geq 2$, and if $\mu(x) = \mu(y)$, then $\#\mu(x) \geq 4$.*
- *$\mu(\mathcal{N})$ contains at least one more pair $\mu(k), \mu(\ell)$ which is distinct from $\mu(x), \mu(y)$, such that $\mu(v_t) = \mu(k) + \mu(\ell)$.*

Proof: If $\mu(v_t) \in \mu(\mathcal{N})$ belongs to a tree-clone, then there are at least two tree-nodes v_1 and v_2 with μ-vector equal to $\mu(v_1)$. By Lemma 1, unit-vectors do not belong to tree-clones, therefore v_1 and v_2 are not leaves. This means v_1 and v_2 each have two children. Let c_1 and c_2 be the children of v_1, and let c_3 and c_4 be the children of v_2. Note that c_1 must be distinct from c_2, and c_3 must be distinct from c_4, because phylogenetic networks do not contain parallel edges. We then have: $\mu(c_1) + \mu(c_2) = \mu(v_1) = \mu(v_2) = \mu(c_3) + \mu(c_4)$.

If the children of v_1 are the same as the children of v_2 then c_1 and c_2 both have two parents and are therefore reticulations. In that case, $\#\mu(c_1) \geq 2$ and $\#\mu(c_2) \geq 2$, and if $\mu(c_1) = \mu(c_2)$, then $\#\mu(c_1) \geq 4$, because c_1 cannot have the same child as c_2 because the network is stack-free.

If v_1 and v_2 share only one child, say $c_2 = c_3$, then c_1 and c_4 are distinct nodes. In this case, c_2 is a reticulation, and $\#\mu(c_2) \geq 2$. Then, either $\mu(c_1) = \mu(c_4) \neq \mu(c_2)$, so that $\#\mu(c_1) \geq 2$ as well. Or $\mu(c_1) = \mu(c_4) = \mu(c_2)$, which implies $\#\mu(c_1) \geq 4$, because neither c_1 nor c_4 can be the child of c_2, because the network is stack-free.

Finally, if all nodes c_1, c_2, c_3 and c_4 are distinct from each other. Then either, the sets $\{\mu(c_1), \mu(c_2)\}$ and $\{\mu(c_3), \mu(c_4)\}$ are not the same set, in which case $\mu(\mathcal{N})$ contains two distinct pairs, whose sum is $\mu(v_1)$. Or they are the same set, in which case we can say w.l.o.g. that $\mu(c_1) = \mu(c_3)$, which implies $\mu(c_2) = \mu(c_4)$. Which means that both $\#\mu(c_1) \geq 2$ and $\#\mu(c_2) \geq 2$, and if $\mu(c_1) = \mu(c_2)$ then $\#\mu(c_1) \geq 4$. □

Lemma 28 *Let $\mu(v_b) \in \mu(\mathcal{N})$ belong to a bridge-node v_b. Let I_b be the set of leaves below v_b, so that $\mu_i(v_b) = 0$ for $i \notin I_b$. For any $\mu(x) \in \mu(\mathcal{N})$ with $\mu(x) \not< \mu(v_b)$, it holds that $\mu(x) = P_{xv_b}\mu(v_b) + \mu'(x)$, where P_{xv_b} is a non-negative integer, equal to the number of paths from x to v_b, and μ' is a μ-vector such that $\mu'_i = 0$ for $i \in I_b$.*

Proof: Let $\mu(v_b) \in \mu(\mathcal{N})$ belong to a bridge-node. For any $\mu(x) \in \mu(\mathcal{N})$ with $\mu(x) \not< \mu(v_b)$, by Lemma 20, $\mu(x)$ does not belong to a descendant of the nodes with μ-vector $\mu(v_b)$. Therefore, all paths from x to leaves below v_b must visit v_b. Thus, if a is a leaf below v_b, then any path from x to a is the composition of a path from x to v_b and a path from v_b to a. It follows that for each such leaf a, $\mu_a(x) = P_{xv_b}\mu_a(v_b)$. From this it follows that $\mu(x) = P_{xv_b}\mu(v_b) + \mu'(x)$, where $\mu'(x)$ contains only the paths to leaves not below v_b, and therefore $\mu'(x)$ is a μ-vector such that $\mu'_i(x) = 0$ for $i \in I_b$. □

Lemma 29 *Let $\mu(v_b) \in \mu(\mathcal{N})$ belong to a bridge-node v_b. Then, for any $\mu(x), \mu(y) \in \mu(\mathcal{N})$ such that $\mu(x) < \mu(v_b)$, $\mu(\mathcal{N})$ does not contain $\mu(k), \mu(\ell)$ with $\mu(k) \not< \mu(v_b)$ and $\mu(\ell) \not< \mu(v_b)$, such that $\mu(k) + \mu(\ell) = \mu(x) + \mu(y)$.*

Proof: We will show this with a proof by contradiction. Let us assume that $\mu(v_b) \in \mu(\mathcal{N})$ belong to a bridge-node v_b and $\mu(\mathcal{N})$ contains some $\mu(x), \mu(y)$ such that $\mu(x) < \mu(v_b)$. Now let us assume, $\mu(\mathcal{N})$ contains $\mu(k), \mu(\ell)$ with $\mu(k) \not< \mu(v_b)$ and $\mu(\ell) \not< \mu(v_b)$, such that $\mu(k) + \mu(\ell) = \mu(x) + \mu(y)$. First note, that if $\mu(y) \leq \mu(v_b)$, then $\mu(y)$ belongs to a descendant of v_b and therefore $\mu(k) + \mu(\ell) = \mu(x) + \mu(y) \leq \mu(v_b)$. But this contradicts our assumption that $\mu(k) \not< \mu(v_b)$. Therefore, we can assume that $\mu(y) \not\leq \mu(v_b)$. Then, by Lemma 28:

$$\mu(k) = P_{kv_b}\mu(v_b) + \mu'(k)$$
$$\mu(\ell) = P_{\ell v_b}\mu(v_b) + \mu'(\ell)$$

and

$$\mu(y) = P_{yv_b}\mu(v_b) + \mu'(y).$$

Furthermore, from $\mu(x) + \mu(y) = \mu(k) + \mu(\ell)$ it follows:

$$\begin{aligned}\mu(x) &= \mu(k) + \mu(\ell) - \mu(y) \\ &= P_{kv_b}\mu(v_b) + \mu'(k) + P_{\ell v_b}\mu(v_b) + \mu'(\ell) - [P_{yv_b}\mu(v_b) + \mu'(y)] \\ &= (P_{kv_b} + P_{\ell v_b} - P_{yv_b})\mu(v_b) + \mu'(k) + \mu'(\ell) - \mu'(y).\end{aligned}$$

Then, from $\mu(x) < \mu(v_b)$ it follows that $P_{kv_b} + P_{\ell v_b} - P_{yv_b} = 0$, and $\mu'(k) + \mu'(\ell) - \mu'(y) = 0$. But then $\mu(x)$ is the zero vector, which is not possible because the zero vector is not a μ-vector and therefore not contained in $\mu(\mathcal{N})$. We have reached a contradiction. □

Lemma 30 *Given a μ-vector $\mu(v_b) \in \mu(\mathcal{N})$. If $\mu(\mathcal{N})$ contains $\mu(z) \not\leq \mu(v_b)$, $\mu(x) < \mu(v_b)$ and $\mu(y)$, such that $\mu(z) = \mu(x) + \mu(y)$. Then, $\mu(v_b)$ does not belong to a bridge-node.*

Proof: We will argue by contradiction. Assume $\mu(\mathcal{N})$ contains $\mu(v_b), \mu(z), \mu(x)$ and $\mu(y)$ as described in the lemma and let $\mu(v_b)$ belong to a bridge-node v_b. If $\mu(z)$ belongs to a reticulation r, then the child of r must be a tree-node with the same μ-vector, so w.l.o.g. we can assume $\mu(z)$ belongs to a tree-node z. Because $\mu(z) = \mu(x) + \mu(y)$, we know z is not a leaf, because $\mu(z)$ is not a unit vector. If $\mu(x), \mu(y)$ is the only pair in $\mu(\mathcal{N})$ which sum up to $\mu(z)$ then z must be the parent of x. If there are more pairs in $\mu(\mathcal{N})$ then by Lemma 29, each of those pairs must contain at least one μ-vector lower than $\mu(v_b)$. This means that in any case z will have a child with μ-vector lower than $\mu(v_b)$. However, $\mu(z) \not\leq \mu(v_b)$ implies that z is not a descendant of v_b. Then there must be a path from the root to a leaf below v_b via z which does not visit v_b. By Lemma 20, this means that v_b is not a bridge-node. Which contradicts our assumption. □

Theorem 5 *A non-unit vector $\mu(v_b)$ belongs to a bridge-node if, and only if,*

- *there is exactly one pair $\mu(k), \mu(\ell) \in \mu(\mathcal{N})$ such that $\mu(v_b) = \mu(k) + \mu(\ell)$, and*
- *$\mu(k)$ and $\mu(\ell)$ do not both have multiplicity greater than 1 in $\mu(\mathcal{N})$ unless $\mu(k) = \mu(\ell)$ with $\#\mu(k) = 2$, and*
- *$\mu(\mathcal{N})$ does not contain vectors $\mu(x), \mu(y), \mu(z)$, such that $\mu(z) \not\leq \mu(v_b)$, $\mu(z) = \mu(x) + \mu(y)$ and $\mu(x) < \mu(v_b)$.*

Proof: Assume the non-unit vector $\mu(v_b)$ belongs to a bridge-node. Then $\mu(v_b)$ belongs to a tree-node which is not a leaf and, by Lemma 26, there exists exactly one pair $\mu(k), \mu(\ell)$ with $\mu(v_b) = \mu(k) + \mu(\ell)$. Furthermore, by Lemma 25, the combined multiplicity of $\mu(k)$ and $\mu(\ell)$ in $\mu(\mathcal{N})$ is lower than or equal to 3. Finally, by Lemma 30, there does not exist $\mu(z) \not\leq \mu(v_b)$ such that $\mu(z) = \mu(x) + \mu(y)$ for $\mu(x) < \mu(v_b)$. This proves the first direction.

For the other direction, we will show the inverse holds. Let us assume the non-unit vector $\mu(v_b)$ does not belong to a bridge-node. Then, for each node v_b with μ-vector $\mu(v_b)$ there must be a path from the root to a descendant of v_b, which does not visit v_b. Note that for any reticulation r with μ-vector equal to $\mu(v_b)$, there must be a tree-node with the same μ-vector and r itself is not a bridge node. So w.l.o.g. it is enough to show that this holds for tree-nodes with μ-vector $\mu(v_b)$. Note that, for the μ-vector of the root $\mu(\rho) \not\leq \mu(v_b)$. Therefore, there must be a node z on the path from the root to a descendant of v_b which does not visit v_b, with $\mu(z) \not< \mu(v_b)$, and z is the parent of a node x, with $\mu(x) < \mu(v_b)$. Note that z cannot be a reticulation, because then $\mu(v_b) \not> \mu(z) = \mu(x) < \mu(v_b)$.

This means, that either $\mu(z) = \mu(v_b)$ or $\mu(z) \nleq \mu(v_b)$. If $\mu(z) = \mu(v_b)$, then $\mu(v_b)$ belongs to a tree-clone. In that case, there is at least one pair $\mu(k), \mu(\ell) \in \mu(\mathcal{N})$ which belong to the children of a node v_b, such that $\mu(v_b) = \mu(k) + \mu(\ell)$. Then, by Lemma 27, either there is more than one such pair, or $\#\mu(k) \geq 2$ and $\#\mu(\ell) \geq 2$, and if $\mu(k) = \mu(\ell)$ then $\#\mu(k) \geq 4$. This violates condition 1 or 2.

If $\mu(z) \nleq \mu(v_b)$, then $\mu(v_b)$ does not necessarily belong to a tree-clone. Note z is also not a leaf, as a leaf has no children. Therefore z is a tree-node with two children, one of which is x. This means, there exists a node y, the other child of z, such that $\mu(z) = \mu(x) + \mu(y)$. This violates condition 3. Now, we have shown one of the three conditions must be false. This proves the inverse statement. □

Let $\mathcal{N}_1$ and $\mathcal{N}_2$ be two semi-binary stack-free networks. Theorem 5 shows that if $\mu(\mathcal{N}_1) = \mu(\mathcal{N}_2)$ then the same μ-vectors will belong to bridge-nodes in both μ-representations. Lemma 24 shows that if $\mu(v_b)$ belongs to a bridge in both networks, then the same μ-vector belongs to the lowest reticulation above that bridge in both networks. Then, by Lemma 22, these reticulations will be stable with respect to the same leaves.

Theorem 6 *Let $\mathcal{N}_1$ and $\mathcal{N}_2$ be two strongly reticulation-visible semi-binary networks, with $\mu(\mathcal{N}_1) = \mu(\mathcal{N}_2)$. Then, $\bar{\mu}(\mathcal{N}_1) = \bar{\mu}(\mathcal{N}_2)$.*

Proof: As $\mathcal{N}_1$ and $\mathcal{N}_2$ are strongly reticulation-visible, there is a tree-path to a bridge node from the child of each reticulation. As mentioned above, the same μ-vectors belong to bridge-nodes in both networks, and the same μ-vectors belong to the lowest reticulation above those bridge-nodes. Therefore, the same μ-vectors in both μ-representations will belong to reticulations which are stable for the same set of leaves. To make this more clear, note the following. By Lemma 16, μ-vectors which belong to stable nodes cannot belong to tree-clones. Furthermore, by Corollary 3, μ-vectors with multiplicity greater than 3 do not belong to stable nodes. Moreover, as $\mathcal{N}_1$ and $\mathcal{N}_2$ are strongly reticulation-visible, all the reticulations in both networks are stable. Therefore, the μ-vectors with multiplicity greater than 3 in $\mu(\mathcal{N}_1)$ and $\mu(\mathcal{N}_2)$ do not belong to reticulations in either network. This means that the set of μ-vectors with multiplicity 2, for which the conditions in Theorem 5 hold is in bijection with the set of reticulations in both networks. In conclusion, the same μ-vectors belong to reticulations in $\mathcal{N}_1$ and $\mathcal{N}_2$, and they are stable for a common set of leaves. Thus, by Theorem 4, $\bar{\mu}(\mathcal{N}_1) = \bar{\mu}(\mathcal{N}_2)$. □

Theorem 7 *Let $\mathcal{N}_1$ and $\mathcal{N}_2$ be two semi-binary stack-free networks with $\mu(\mathcal{N}_1) = \mu(\mathcal{N}_2)$. Let $\mathcal{N}_1$ be strongly reticulation-visible and orchard. Then, $\mathcal{N}_1 \cong \mathcal{N}_2$.*

Proof: Because, $\mathcal{N}_1$ is orchard, it contains no tree-clones. Therefore, each μ-vector in $\mu(\mathcal{N}_1)$ has multiplicity at most 2. And the subset of $\mu(\mathcal{N}_1)$ of μ-vectors with multiplicity 2 is exactly the set of μ-vectors which belong to reticulations. Furthermore, because $\mathcal{N}_1$ is strongly reticulation-visible, these

μ-vectors belong to reticulations which are lowest above some bridge. By Theorem 5, the same μ-vectors correspond to bridge-nodes in $\mathcal{N}_1$ and $\mathcal{N}_2$. And by Lemma 24, every lowest reticulation - bridge-node pair is preserved in $\mathcal{N}_2$. As both μ-representations do not contain any vectors with multiplicity greater than 2, there are no other μ-vectors belonging to reticulations in $\mathcal{N}_2$. Note that, each μ-vector with multiplicity 2 belongs to exactly one reticulation in $\mathcal{N}_1$. Therefore, $\mathcal{N}_1$ and $\mathcal{N}_2$ have the same reticulation set R and each reticulation is stable for a common set of leaves in both networks. This means that, by Theorem 4, $\bar{\mu}(\mathcal{N}_1) = \bar{\mu}(\mathcal{N}_2)$. Therefore, by Theorem 1, $\mathcal{N}_1 \cong \mathcal{N}_2$. □

In the same way as in Section 3, Theorem 7 shows that in a sample of semi-binary stack-free networks, a single strongly reticulation-visible orchard network can be compared to every other network in the sample using the cardinality of the symmetric difference of the μ-representations. Therefore, we can apply the μ-distance $d_\mu(\mathcal{N}_1, \mathcal{N}_2) = |\mu(\mathcal{N}_1) \triangle \mu(\mathcal{N}_2)|$ between two semi-binary stack-free networks $\mathcal{N}_1, \mathcal{N}_2$ as a metric, as long as one of them is strongly reticulation-visible orchard.

6 Conclusions and Discussion

6.1 Main results

In this section we will outline and discuss the main contributions of this thesis, some of which are displayed in Table 6.1.

Given	class $\mathcal{N}_1$	class $\mathcal{N}_2$	Result	Theorem
$\bar{\mu}(\mathcal{N}_1) = \bar{\mu}(\mathcal{N}_2)$	SBSF orchard	SBSF	$\mathcal{N}_1 \cong \mathcal{N}_2$	Theorem 1
$\mu(\mathcal{N}_1) = \mu(\mathcal{N}_2)$	SB SRV	SB SRV	$\bar{\mu}(\mathcal{N}_1) = \bar{\mu}(\mathcal{N}_2)$	Theorem 6
$\mu(\mathcal{N}_1) = \mu(\mathcal{N}_2)$	SB SRV orchard	SBSF	$\mathcal{N}_1 \cong \mathcal{N}_2$	Theorem 7

Table 6.1: A table detailing some of the main results of this report. SB stands for semi-binary, SF stands for stack-free and SRV stands for strongly reticulation visible. Note that strongly reticulation visible networks are stack-free. The first row reads: "Given two networks $\mathcal{N}_1$ and $\mathcal{N}_2$ with equal modified μ-representation, if $\mathcal{N}_1$ is semi-binary stack-free orchard and $\mathcal{N}_2$ is semi-binary stack-free then they are isomorphic"

We've shown that semi-binary stack-free orchard networks are encoded in the space of semi-binary stack-free networks by a modified μ-representation (Theorem 1). This modified μ-representation, which we dubbed the $\bar{\mu}$-representation, contained the same path multiplicity vectors as the standard μ-representation as originally proposed by Cardona et al. in [4], but modified to include the in-degrees of nodes. With this result we've shown that given a semi-binary stack-free orchard network and a semi-binary stack-free network, the cardinality of the symmetric difference of the $\bar{\mu}$-representations of the networks gives a metric. We've also shown that this encoding result does not extend to non-binary stack-free orchard networks even if the out-degrees of nodes corresponding to μ-vectors are fixed (Theorem 2).

We proposed an equation governing the relationship between in-degrees of reticulations in a given X-DAG and the μ-vectors of those reticulations and the root, and we proved the correctness of this equation (Theorem 3). We showed that such an equation exists for each leaf, which induces a system of equations for any given network. We discussed how the system of equations for a given network does not necessarily have a unique solution. We then showed that for any network in the class of reticulation visible networks the system of equations does have a unique solution as long as the reticulation set is fixed and each reticulation is stable with respect to a fixed leaf (Theorem 4).

We then showed that the lowest reticulation above a bridge is stable and can be found by its μ-vector, as long as the μ-vector of the tail of the bridge is known (Lemma 24). Then, we proved that it is possible to determine whether a μ-vector belongs to the tail of a bridge (Theorem 5). We proposed the class of strongly reticulation visible networks, as the class of networks for which each reticulation is lowest above some bridge. For this class we proved that for any two networks with the same μ-representation, the μ-vectors belong to nodes with equal in-degrees. Therefore, they have the same modified μ-representation (Theorem 6). Finally, we concluded that strongly reticulation visible semi-binary stack-free orchard networks are encoded in the class of semi-binary stack-free networks by their μ-representation (Theorem 7). Which means that the cardinality of the symmetric difference of the μ-representation of two semi-binary stack-free, of which one is strongly reticulation visible orchard, gives a metric.

6.2 Discussion

In this section we will discuss some of the remaining open problems. For one, it is still unclear whether semi-binary stack-free orchard networks are in fact encoded in their own class by the μ-representation. Furthermore it may be interesting whether we could further modify the μ-representation to encode more classes. Another open problem is whether bridge-nodes are determined by the μ-representation in non-binary networks. We'll go into more detail on these problems below. A final remaining question is whether there are still more classes of networks, besides subclasses of orchard networks, which are encoded by the μ-representaton. We suggest an interesting research subject could be to try to define the class of networks which do not contain ambiguous nodes, which we defined in Section 4.2.

Tree-clone free networks

The question posed by Murakami in [11], after presenting their counterexample of the original theorem by Bai et al. is whether semi-binary stack-free orchard networks are actually encoded within their own class by the μ-representation. This question remains unanswered. Murakami focussed on the fact that their counterexample contains tree-clones and investigated the structure of tree-clone free networks. However, the example in Figure 6.1 shows two tree-clone free networks, which have the same μ-representation that are not isomorphic. The colored nodes in these networks are what we called ambiguous nodes, in Section 4.2. The μ-vectors of these nodes can be generated by taking the sum of two distinct pairs of nodes in the network.

If we refer to the colored nodes in Figure 6.1 by their color ($v_{red}, v_{blue}, v_{yellow}$), and the reticulation parents of the leaves we denote with an apostrophe (a', b', c', d') and the tree-nodes whose children are two of these reticulations as v_{ab}, v_{bc}, v_{bd}.

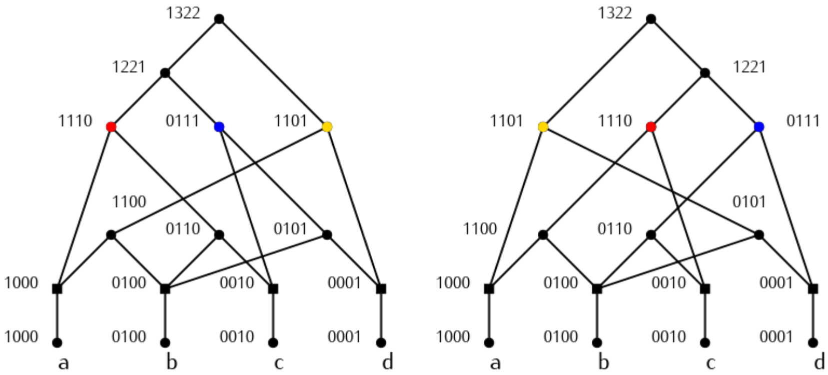


Figure 6.1: Two non-isomorphic semi-binary reticulation-visible tree-clone free networks with the same μ-representation. The coloured nodes have the same μ-vectors in both networks but their children have been permuted.

We obtain the following system of equations:

$$\begin{aligned}\mu(v_{red}) &= \mu(a') + \mu(v_{bc}) = \mu(c') + \mu(v_{ab})\\ \mu(v_{blue}) &= \mu(c') + \mu(v_{bd}) = \mu(d') + \mu(v_{bc})\\ \mu(v_{yellow}) &= \mu(d') + \mu(v_{ab}) = \mu(a') + \mu(v_{bd}).\end{aligned}$$

The μ-vectors of children of the colored nodes are simply in the middle column of these equations for the left network and the right column for the right network. These networks are semi-binary and reticulation visible, this makes them also stack-free. However, they are not orchard and there does not seem to be an obvious way to make them orchard while preserving the equality of μ-representations and the non-isomorphism. It remains to be seen whether this type of structure does not exist in orchard networks and whether other structures still exist which generate ambiguity.

Combining the extended and the modified μ-representations

In [3] Cardona et al. propose an extended μ-representation for encoding binary orchard networks. The extended μ-representation is generated by appending the number of paths to reticulations to the μ-vectors of the μ-representation. They show that with this extended μ-representation reticulated cherries can be found even in networks with stacks. Intuitively, if we combine both their extension and the modification we proposed in Section 3, the resulting encoding should be unique for semi-binary orchard networks. We therefore propose the following definition and conjecture.

Definition 4 *Let the twice extended μ-representation, denoted $\dot{\mu}(\mathcal{N})$, be the encoding of a phylogenetic network $\mathcal{N}$, which combines both the extended and modified μ-representations, by appending both the number of paths to reticulations and the in-degrees to the path multiplicity vectors.*

Conjecture 1 *Let $\mathcal{N}_1$, $\mathcal{N}_2$ be semi-binary orchard networks. Then,*

$$\dot{\mu}(\mathcal{N}_1) = \dot{\mu}(\mathcal{N}_2) \iff \mathcal{N}_1 \cong \mathcal{N}_2. \tag{6.1}$$

Non-binary strongly reticulation-visible networks

In this section we will discuss how the results from Section 5 regarding bridge-nodes and strongly reticulation visible networks extend to non-binary networks. The preliminary lemmas set forth in Section 5.2 hold for both semi-binary and non-binary networks as we never made any assumptions on the degrees of the nodes. However, in the entire Section 5.3 we only considered semi-binary networks. The first lemmas, Lemma 25, Lemma 26 and Lemma 27 all serve to show how to distinguish a bridge-node with multiplicity 2 from a pair of tree-clones. The example shown in Figure 6.2, shows that no similar lemmas hold for non-binary networks. The networks $\mathcal{N}_1$ and $\mathcal{N}_2$ are both strongly reticulation-visible with the same μ-representation, but in $\mathcal{N}_2$ the node v_b is a bridge-node with μ-vector 0220, but in $\mathcal{N}_1$ the nodes with μ-vector 0220 are tree-clones and therefore not stable. This example shows that it is not possible to determine whether a μ-vector with multiplicity 2 belongs to a stable reticulation and its bridge-node child or a pair of tree-clones in non-binary strongly reticulation-visible networks. It also shows that two non-binary strongly reticulation-visible networks with the same μ-representation do not necessarily have the same modified μ-representation as defined in Section 3. This means the μ-vectors do not necessarily belong to nodes with the same in-degrees in both networks.

This negative result does not mean that being strongly reticulation-visible is not relevant for non-binary networks. Remember that orchard networks do not contain any tree-clones. Therefore, if we focus on comparing orchard networks, we do not need to be able to distinguish bridge-nodes from tree-clones. If we look at Lemma 28, we see that it holds for non-binary networks. However, for Lemma 29 there does not seem to be an immediate non-binary analogue. Lemma 29 and Lemma 30, serve to establish a way to determine whether a descendant of a node v_b (besides v_b itself) has a parent which is not a descendant of that node, thereby proving that v_b is not a bridge-node. However, in non-binary networks any number of μ-vectors in $\mu(\mathcal{N})$ which together sum up to the μ-vector of a node v may belong to the children of v. Therefore, we can no longer assume that given a bridge-node v_b and a node z which is not a descendant of v_b, that the sum of the μ-vectors of the children of z which are descendants of v_b is actually smaller than $\mu(v_b)$. If we would modify Lemma 29 to consider tree-nodes which can have any number of children greater than 1,

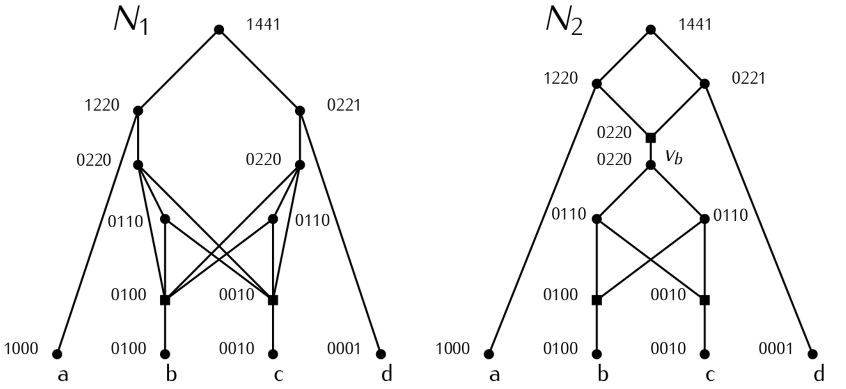


Figure 6.2: Two non-binary strongly reticulation-visible networks $\mathcal{N}_1$ and $\mathcal{N}_2$ with the same μ-representation. In $\mathcal{N}_2$ the edge between the reticulation with μ-vector 0220 and its child is a bridge and the node v_b is a bridge-node. In $\mathcal{N}_1$ the nodes with μ-vector 0220 are tree-clones and therefore not stable.

we would get the following:

Let $\mu(v_b) \in \mu(\mathcal{N})$ belong to a bridge-node and let $\mu(x) < \mu(v_b)$. Then, there do not exist S_1, S_2 subsets of V, with $\mu(v) \not< \mu(v_b)$ for all $v \in S_2$, such that:

$$\mu(x) + \sum_{u \in S_1} \mu(u) = \sum_{v \in S_2} \mu(v) \tag{6.2}$$

However, if we let S_1' be the subset of S_1 of nodes whose μ-vector is smaller than v_b. Then we would get:

$$\begin{aligned}\mu(x) + \sum_{v \in S_1'} \mu(v) &= \sum_{u \in S_2} \mu(u) - \sum_{w \in S_1 \setminus S_1'} \mu(w) \\ &= \left(\sum_{u \in S_2} P_{uv_b} - \sum_{w \in S_1 \setminus S_1'} P_{wv_b} \right) \mu(v_b) + \sum_{u \in S_2} \mu'(u) - \sum_{w \in S_1 \setminus S_1'} \mu'(w)\end{aligned}$$

but in this case we do not have for the left side $\mu(x) + \sum_{v \in S_1'} \mu(v) < \mu(v_b)$, thus we do not have $\left(\sum_{u \in S_2} P_{uv_b} - \sum_{w \in S_1 \setminus S_1'} P_{wv_b} \right) = 0$ for the right side. Therefore, we cannot use this argument to state that Equation 6.2 has no solution. Further research is needed to determine whether bridge-nodes are uniquely determined by the μ-representation in non-binary orchard networks. Even if they are, non-binary strongly reticulation-visible orchard networks are not encoded by their μ-representation, as the networks in Figure 3.4 are strongly reticulation-visible and serve as a counterexample.

Final Remarks

In this report we identified several classes of phylogenetic networks which are encoded by the μ-representation or a modified version thereof in the larger class of semi-binary stack-free networks. This (modified) μ-representation can be found in polynomial time for any X-DAG. It can be applied by researchers for comparing phylogenetic networks through the symmetric difference, as long as the results of the algorithmic techniques used to generate the phylogenetic networks are restricted to the class of semi-binary stack-free networks. Thus, allowing for uncertainty in the order of hybridization or hybridizations in which a larger number of species combine, but not sequential hybridizations with no evolution steps in-between. Therefore, if by application of different techniques or the same technique to different sets of gene-sequences multiple results are generated for the same set of species or taxa, these results can be compared in polynomial time using the μ-representation as long as one is strongly reticulation-visible and orchard or the modified μ-representation as long as one is orchard. Where before, using the μ-representation was only possible if all results were binary stack-free orchard networks or tree-child networks, which is a subclass of strongly reticulation-visible orchard networks. Furthermore, adapting and extending the proof, one can obtain a polynomial-time algorithm to construct a unique semi-binary strongly reticulation-visible orchard network from a given μ-representation (if it is consistent). Thus allowing for reconstruction of such a network if the μ-representation can be generated from genetic data.

Bibliography

[1] Allan Bai, Péter L Erdős, Charles Semple, and Mike Steel. “Defining phylogenetic networks using ancestral profiles”. *Mathematical Biosciences* 332 (2021).

[2] David Baum. “Reading a phylogenetic tree: the meaning of monophyletic groups”. *Nature Education* 1.1 (2008).

[3] Gabriel Cardona, Joan Carles Pons, Gerard Ribas, and Tomás Martínez Coronado. “Comparison of orchard networks using their extended μ-representation”. *arXiv preprint arXiv:2302.10015* (2023).

[4] Gabriel Cardona, Francesc Rosselló, and Gabriel Valiente. “Comparison of tree-child phylogenetic networks”. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 6.4 (2008), pp. 552–569.

[5] Gabriel Cardona, Francesc Rosselló, and Gabriel Valiente. “Tripartitions do not always discriminate phylogenetic networks”. *Mathematical Biosciences* 211.2 (2008), pp. 356–370.

[6] Surekha Challa and Nageswara Rao Reddy Neelapu. “Phylogenetic trees: applications, construction, and assessment”. *Essentials of Bioinformatics, Volume III: In Silico Life Sciences: Agriculture* (2019), pp. 167–192.

[7] Charles Darwin. *On the origin of species: A facsimile of the first edition.* Harvard University Press, 1964.

[8] W Ford Doolittle. “Phylogenetic classification and the universal tree”. *Science* 284.5423 (1999), pp. 2124–2128.

[9] Péter L Erdős, Charles Semple, and Mike Steel. “A class of phylogenetic networks reconstructable from ancestral profiles”. *Mathematical biosciences* 313 (2019).

[10] Leo van Iersel, Remie Janssen, Mark Jones, and Yukihiro Murakami. “Orchard networks are trees with additional horizontal arcs”. *Bulletin of Mathematical Biology* 84.8 (2022).

[11] Yukihiro Murakami. “On Phylogenetic Encodings and Orchard Networks”. PhD dissertation. Delft University of Technology, 2021.

[12] Rosanne Wallin, Leo Van Iersel, Steven Kelk, and Leen Stougie. “Applicability of several rooted phylogenetic network algorithms for representing the evolutionary history of SARS-CoV-2”. *BMC ecology and evolution* 21.1 (2021), pp. 1–14.