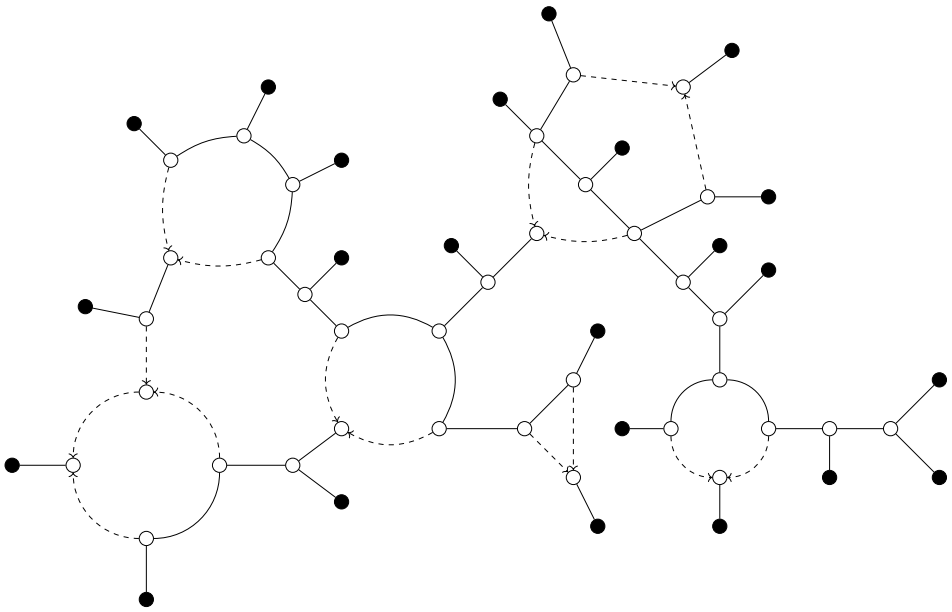
Reconstructing Phylogenetic Networks using Quarnet Puzzling

by

Simon Deuten

to obtain the degree of Bachelor of Science
at the Delft University of Technology,
to be defended publicly on Tuesday July 15, 2025 at 14:00 AM.



Student number:	5600243	
Project duration:	March 4, 2025 – July 15, 2025	
Thesis committee:	Dr. ir. L.J.J. van Iersel,	TU Delft, supervisor
	Dr. A. Heinlein,	TU Delft

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Abstract

Evolution can be modelled by semi-directed phylogenetic networks, partially directed graphs where directed edges represent reticulate evolutionary events. In this thesis, we present a polynomial-time algorithm that can reconstruct a level-2 semi-directed phylogenetic network from its quarnets (4-leaf subnetworks). First, we find a canonical form of a network by reducing our information to merely quarnet-splits. Here, we use the blob-tree of the network found with quarnet-splits to construct the canonical form of the network. Finally, we consider quartets, which give more information about level-1 and level-2 quarnets, to refine on the canonical form.

Summary

Evolution is a grand topic in various research fields, especially biology, linguistics, and genetics. Relations between different species can be described with a graph, where the leaves represent current species, and the edges indicate that a species is a direct descendant of another species. We call such a network of nodes and edges that represent evolutionary relationships a phylogenetic network. A special case of a well studied evolutionary network is a phylogenetic tree, where a node branches off into multiple different nodes, with directed edges representing the evolutionary direction, thus where nodes have at most one incoming directed edge. However, in reality it is common to have hybridisation between two different species, where two directional edges come together to one node, a reticulation node. This hybridisation gives a new level of complexity to a network. This can give us more insight into how evolution progresses and helps us better understand how and why various species can be formed.

In this thesis, we focus on the topic of constructing such a phylogenetic network with the information of quarnets. A quarnet is a phylogenetic network between four different species. Suppose that we have a set of species X. If we have a quarnet for every combination of four species in X, we can try to puzzle these quarnets together to form a full phylogenetic network between every species in X.

Phylogenetic networks can have different levels. A level of a network indicates the maximum amount of reticulation in a blob, a maximal sub-graph with no cut-edges. What has been previously researched in this field is the construction of level-0 and level-1 networks using information from quarnets. What is also known is that level-2 networks are encoded by quarnets, while level-3 and higher networks are not [3]. What that means is that, while a unique level-3 network cannot be constructed with only quarnets, a level-2 network can. In this thesis, we explore what the maximal version (canonical form) of a network is that can be found with limited information from the quarnets, mainly quarnet-splits.

Our research question is then if can we prove the following theorem: Given a phylogenetic network, we can find an algorithm that constructs the canonical form of the network using quarnet-splits/quartets of the network. Here we have two theorems, one for quarnet-splits and one for quartets. Our algorithm will make use of the blob-tree of a network, where all blobs of the network get contracted to a single node. Then the idea is, can we inflate those nodes back to their original form using the quarnet-splits/quartets? In this thesis, we will show that it is indeed possible for up to level-2 blobs and how this construction is done. Finally, we give an implementation for constructing the canonical form of a level-1 network using its quarnet-splits, which can later be expanded to level-2 networks with the algorithm given in this thesis.

Contents

1 Introduction

Phylogenetic networks describe evolutionary progress between various species, languages, and other taxa where evolution is present. Phylogenetic networks come in two forms. The first is a phylogenetic tree, where each node splits up into two new nodes. In a phylogenetic network, reticulation, or hybridization, is also present. That is, the evolutionary direction of two nodes is pointing towards the same node. In this thesis, we focus on the latter, as these networks are more complicated to work with, research on this networks are less common, and because phylogenetic trees are a sub-category of phylogenetic networks. Such a network is described by a binary semi-directed graph. This graph includes reticulation, which is represented by two directed reticulation edges pointing towards the reticulation node. All other edges apart from reticulation edges are undirected. The graph also only includes leaves and degree-3 nodes, where each leaf is a current taxa stored in the list of all taxa that are included in the graph. Such a network can be seen in Figure 1.1. In this thesis, we research the construction of such a network using smaller bits of information from the network, which will be explained later on. The important question is why this construction is useful and for whom it will be useful. Smaller pieces of information from a full network are easier to obtain than the full network on its own. And therefore, if we want to have the bigger picture of a phylogenetic network, it is useful to not have only a piece of the network, but the whole network itself. Therefore, combining this information to a big network is more useful to analyse, describe and understand evolutionary processes. This is a useful tool for especially scientists who work in the field of evolution, such as genetical evolution and biology researchers. But the idea of this construction may also be extended to other networks where relations do not have to be evolutionary, but a different form of connection that can be described with a graph. However, in this thesis we only focus on binary semi-directed phylogenetic networks.

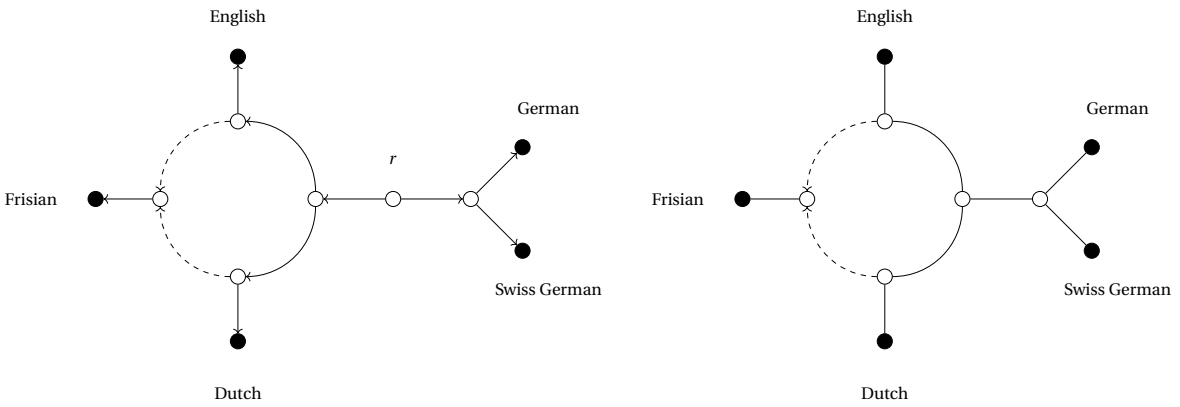


Figure 1.1: Example of a phylogenetic network, with the leaves being displayed as black dots, where each leaf represents a current language or dialect. The directed version of the phylogenetic network is on the left with its root r, and the semi-directed version of the phylogenetic network is on the right. [6]

A semi-directed phylogenetic network is a more simple and more achievable version of a phylogenetic network. In a directed phylogenetic network, as the name already tells us, all edges are directed, where there exists one node of degree two with two outgoing directed edges, which is called the root of the network. This is the better known version of a phylogenetic network, where evolution can be traced back to a root, unlike in a semi-directed phylogenetic network, where this is not possible. Such a network can be seen in Figure 1.2. We focus on semi-directed networks, as for a directed network, we have to include all directions of evolution, which is hard to find in practice. Also, a root is in most cases not identifiable, and as a root is always present in a directed phylogenetic network, it is better to first focus on semi-directed networks.

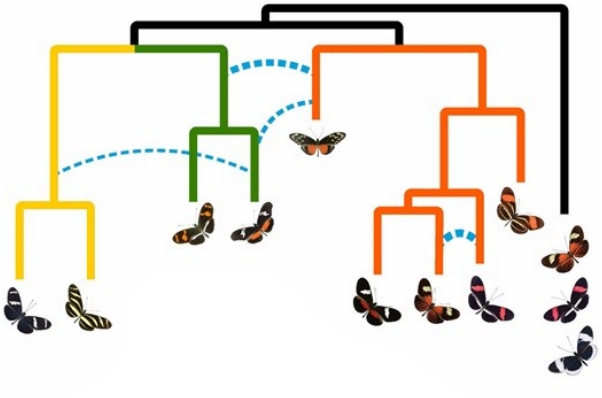


Figure 1.2: Example of a directed phylogenetic network with a root at the top (all non-dashed edges are directed downwards, dashed edges can point either way). Note that this image does not follow the direct definition of a directed phylogenetic network, as we do not know the direction of the reticulation edges. [5]

A list of quarnet-splits of a network will be the main source of information with which we will construct a semi-directed phylogenetic network. Quarnet-splits are a detail of a quarnet, which is a phylogenetic network on four leaves. A phylogenetic network on n leaves can be restricted to a network on four leaves, which is one quarnet of that network. This restriction will be explained in more detail in the following chapter. By restricting the network to all possible set of four leaves, all quarnets of the network can be obtained. However, in practice, it is of course easier to find the quarnets first, or only a piece of information from the quarnet, which is the quarnet-split. And as a phylogenetic network can be restricted to its quarnets, we will research if we can do the other way around, if such a network can also be constructed using its quarnets. In Figure 1.3, multiple unlabelled quarnets can be seen. A quarnet-split exists if there is a non-trivial cut-edge in the network that is not adjacent to a leaf. Thus, in this figure, the first five quarnets include a quarnet-split, and the last one does not. In the first five quarnets, there exists a quarnet-split between the top two leaves and the bottom two leaves, which are connected by the non-trivial cut-edge in the middle.
In this thesis, the closest version to a binary semi-directed phylogenetic network that can be obtained using the quarnet-splits of the network will be researched. This version is the canonical form of the network. This is as some networks can have the same exact set of quarnet-splits, therefore, multiple semi-directed phylogenetic networks have the same canonical form.

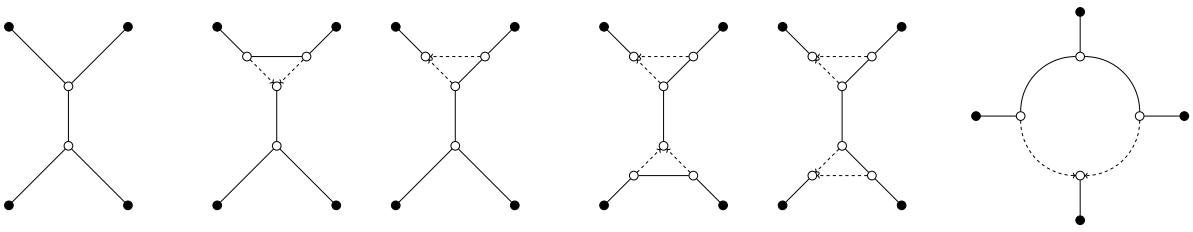


Figure 1.3: All possible quarnets of a semi-directed level-1 phylogenetic network.

Phylogenetic networks have levels. A level-0 network is a tree, and level-1 and higher networks have cycles included, which is a result from reticulation. The construction of level-0 and level-1 networks using quarnet-splits is already researched in previous works, where it was shown that level-0 networks can be fully found

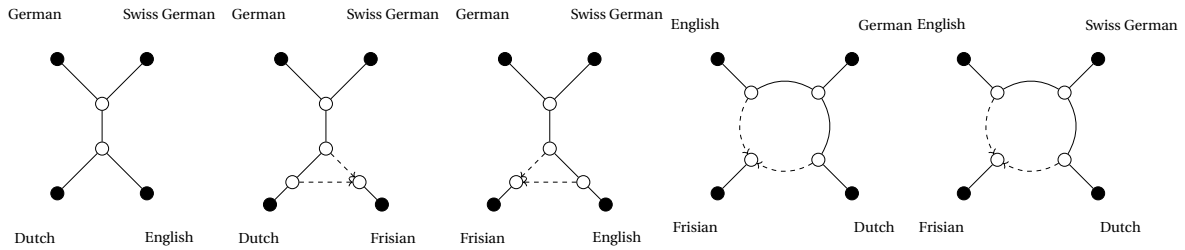


Figure 1.4: All quarnets of the example semi-directed phylogenetic network in Figure 1.1. Here can be seen that the network has three quarnet-splits: German,Swiss German|Dutch,English, German,Swiss German|Dutch,Frisian and German,Swiss German|English,Frisian.

with quarnet-splits, and that the canonical form of a network can be constructed using quarnet-splits [2]. Therefore, in this thesis, we focus on level-2 networks, and study whether such networks are achievable using quarnet-splits and if there exists an algorithm that can construct them. For this construction, we use the help of the blob-tree of a network. A blob in a phylogenetic network is a maximal subgraph that has no cut-edges. Then, the blob-tree of a network is obtained by contracting all blobs, thus these maximal subgraphs, to a single vertex. It has been proven that a blob-tree of any network of any level can be found using quarnet-splits [2]. As the blobs get contracted in a network, we will try to reverse this process and inflate the internal nodes in a blob-tree to full blobs.

The contribution of this thesis is as follows. First, we properly define all terms used in this thesis, e.g., semi-directed phylogenetic networks, quarnets and quarnet-splits, blob-tree, etc., which are found in Chapter 2. Then, we will guide the reader through the construction of the blob-tree of a network in Chapter 3. The following Chapter 4 will give an algorithm on constructing the canonical form of a level-1 network using its quarnet-splits. Then, in Chapter 5, the main contribution is given. Here, we properly define the canonical form of a level-2 network, and we prove the main theorem of the thesis, which is also given in that same chapter. In Chapter 6, we expand our information set to quartets, with which we will refine on the canonical form of a network. Lastly, an explanation of the implementation of the algorithm for constructing the canonical form of a level-1 network can be found in Chapter 7.

2

Preliminaries

This chapter will give an introduction to various terms and concepts that will be used throughout this thesis. These terms and definitions are mainly taken from article [2] and [3].

2.1. Phylogenetic Network

The network we want to construct is called a *(binary) semi-directed phylogenetic network*. A semi-directed phylogenetic network models the evolution history of, for example, various species. This network is represented by a semi-directed graph, where directed edges give the direction of the evolution. This network is based on a set of leaves $\mathcal{X} = \{x_1, x_2, ..., x_n\}$, where every leaf x_i in the network is an element of $\mathcal{X}$. This set can contain all sorts of elements, for example, $\mathcal{X} = \{1,2,3,4,5\}$, but also $\mathcal{X} = \{ape, horse, snake\}$. A semi-directed phylogenetic network is obtained from a *directed phylogenetic network*, where every edge in the graph is directed. A directed phylogenetic network also always includes a root, which is in most cases unidentifiable in practice, therefore, we only focus on semi-directed phylogenetic networks. Also, as we only consider binary networks in this thesis, from now on, we will mention this in definitions and theorems only.

Definition 1. (*Binary Directed Phylogenetic Network*) A binary directed phylogenetic network $\mathcal{N}$ on $\mathcal{X}$ is a graph with the following properties:

(i) $\mathcal{N}$ has no undirected edges

(ii) $\mathcal{N}$ has no directed cycles

(iii) each node has degree at most 3, indegree at most 2 and outdegree at most 2

(iv) there is a unique node with indegree 0 and outdegree 2, which is called the *root* of the network

(v) the nodes with outdegree 0 are leaves and are bijectively labelled by the elements from $\mathcal{X}$

Definition 2. (*Binary Semi-directed Phylogenetic Network*) A binary, semi-directed phylogenetic network $\mathcal{N}$ on $\mathcal{X}$ is obtained from a directed phylogenetic network on $\mathcal{X}$ by undirecting all edges that are no *reticulation edges* (edges pointing to a reticulation node) and suppressing the root of the network. Then, the network has the following properties:

(i) The network has no parallel edges.

(ii) Each node has either one or three adjacent nodes.

(iii) Each node has either two incoming directed edges, at most one outgoing directed edge, or no incident directed edges at all.

In Figure 1.1, we can see an example of a semi-directed phylogenetic network, with one reticulation node present. In this graph, the leaves are displayed as black dots, and the reticulation edges are displayed as dashed pointing edges, pointing towards the reticulation node. Think of a reticulation node as a hybridisation between two points, for example, a cross between two different species in the network. Hybridisation gives us

extra insight into how evolution goes to work, by not only splitting sorts into multiple different sorts, but also combining sorts into a new one. The presence of reticulation nodes in a phylogenetic network leads to the existence of *blobs*. A blob is a maximal sub-graph with at least two nodes and with no cut-edges within. The amount of incident edges to a blob decides the degree of that blob. Thus, a blob with m edges incident to it has a degree of m, which we call an *m-blob*. In this thesis, we only consider phylogenetic networks with leaves and blobs of degree at least 3, thus m-blobs with $m \geq 3$. Leaves in a phylogenetic network are not considered as blobs.
Let $\mathcal{N}$ be a phylogenetic network on $\mathcal{X}$. If removing all leaves in $\mathcal{N}$ results in a blob, then $\mathcal{N}$ is called *simple*.

Each m-blob with $m \geq 4$ has at least one reticulation node. The maximum number of reticulation nodes present in any blob in a phylogenetic network determines the level of the network. Suppose that the highest number of reticulation nodes in any blob in a network $\mathcal{N}$ is 3, then the network $\mathcal{N}$ is of level-3. The network is called *strict level-k* if the network is not level-$(k-1)$.
We say that a leaf is *below* a reticulation node, if there exists a semi-directed path starting from the reticulation node and ending at the leaf. We say that a leaf is *directly below* a reticulation node if this path does not contain other reticulation nodes.
In a blob, if a reticulation node has no other reticulation nodes in the blob below it, it is called a *sink* of the blob. A level-k blob can have up to k sinks.

A level-m network with $m \geq 1$ is complicated to get an idea of what the basic process of the construction is. Therefore, we first reduce the phylogenetic network to a *blob-tree*. This we achieve by contracting all individual blobs to single nodes. Then, the incident nodes of the blobs become the neighbours of those contracted nodes.

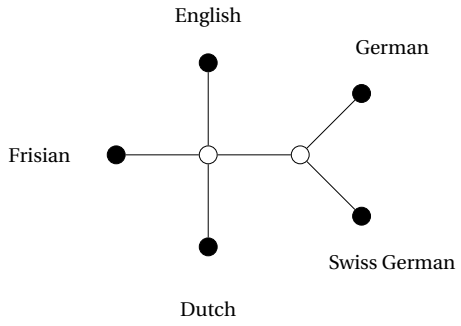


Figure 2.1: The corresponding blob-tree to the sample semi-directed phylogenetic network shown in Figure 1.1

In Figure 2.1, we can see an example of the result of contracting all the blobs of the phylogenetic network presented in Figure 1.1. This network had only one blob with 3 incident leaves, which is contracted to the internal node with 3 neighbouring leaves. Note that in the blob-tree, there are no reticulation edges and nodes, as these are always part of a blob, and therefore there are no cycles in the network, resulting in being a tree.

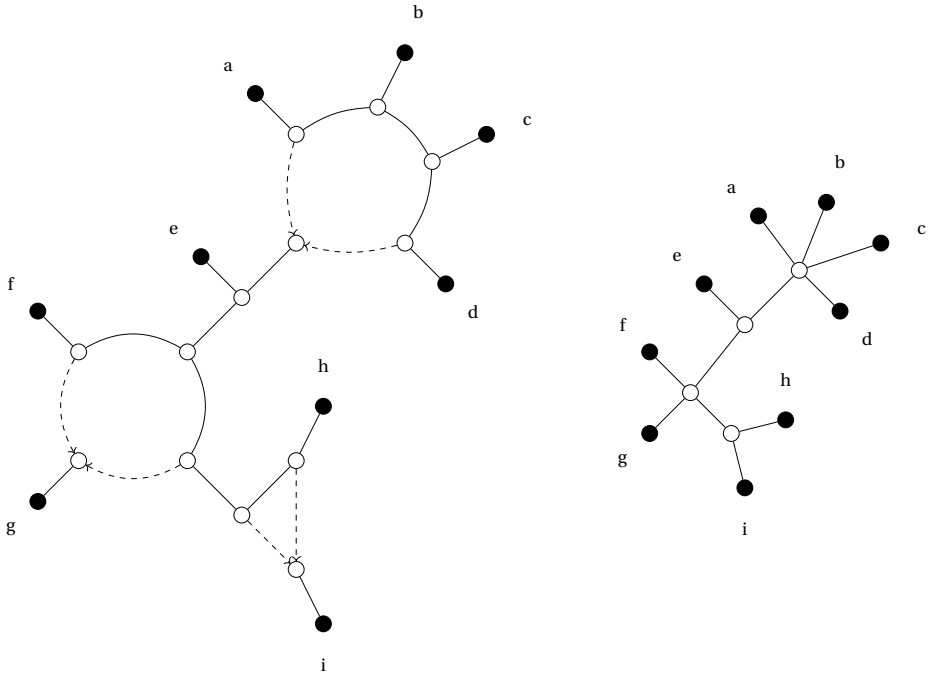


Figure 2.2: An example semi-directed phylogenetic network with multiple blobs on the left and its corresponding blob-tree on the right.

2.2. Quarnets and splits

We can restrict a network $\mathcal{N}$ on $\mathcal{X}$ to a smaller set of leaves $\mathcal{Y} \subset \mathcal{X}$. With this restriction, we get a smaller network $\mathcal{N}|_{\mathcal{Y}}$.
An *up-down path* in a semi-directed network is a path between leaves x_1 and x_2 with a length of k edges, where the first l edges are directed towards x_1 and the last $k-l$ edges are directed towards x_2. Undirected edges are considered as bidirected.

Definition 3. (Restriction). Given a semi-directed network $\mathcal{N}$ on $\mathcal{X}$ and some $\mathcal{Y} \subset \mathcal{X}$ with $|\mathcal{Y}| \geq 2$, the restriction of $\mathcal{N}$ to $\mathcal{Y}$ is the semi-directed network $\mathcal{N}|_{\mathcal{Y}}$ obtained from $\mathcal{N}$ by taking the union of all up-down paths between leaves in $\mathcal{Y}$, followed by exhaustively suppressing all 2-blobs and degree-2 vertices, and identifying parallel edges.

Suppose that we have the network $\mathcal{N}$ on $\mathcal{X}$ with a subset $\mathcal{Y} \subseteq \mathcal{X}$ with $|\mathcal{Y}| = 4$. Then, the restriction $\mathcal{N}|_{\mathcal{Y}}$ is called a *quarnet* of $\mathcal{N}$. For a level-1 network, there are six possible quarnets that can be obtained by taking a restriction on four leaves. These types of quarnets can be found in Figure 1.3.

The quarnets of a phylogenetic network are the pieces of information needed to construct a complete level-1 network. But before we get to that, first, a more incomplete version of the network is created using even less information. That information is the splits of the quarnets (*quarnet-splits*).

Definition 4. (Split) Given a network $\mathcal{N}$ on the set of leaves $\mathcal{X}$. Let $A \subset \mathcal{X}$ and $B = \mathcal{X} \setminus A$. Then $A|B$ is called a *split* of $\mathcal{N}$, if there exists a cut-edge in $\mathcal{N}$ that, with removal of this edge, disconnects the leaves A from B.

Given a quarnet, there exists a quarnet-split, if the split $A|B$ can be made with both $|A| = |B| = 2$, where both A and B are disjoint subsets of the set of leaves $\{a, b, c, d\}$ the quarnet is based on. For example, given the six types of quarnets in Figure 1.3, for the first five quarnets, there exists a quarnet-split between the upper two leaves and the lower two leaves, whereas the last quarnet, the quarnet-cycle, does not contain any quarnet-splits.

Definition 5. A *quarnet-split* of $\mathcal{N}$ is a split $A|B$ of a quarnet of $\mathcal{N}$ with $|A| = |B| = 2$.

Of course, a careful observer would note that even the quarnet-cycle contains four cut-edges, therefore four splits. However, as each leaf in a network has degree of one, these cut-edges are trivial, and therefore giving us no useful information. In this thesis, we only care about a split $A|B$ if both $|A| \geq 2$ and $|B| \geq 2$.

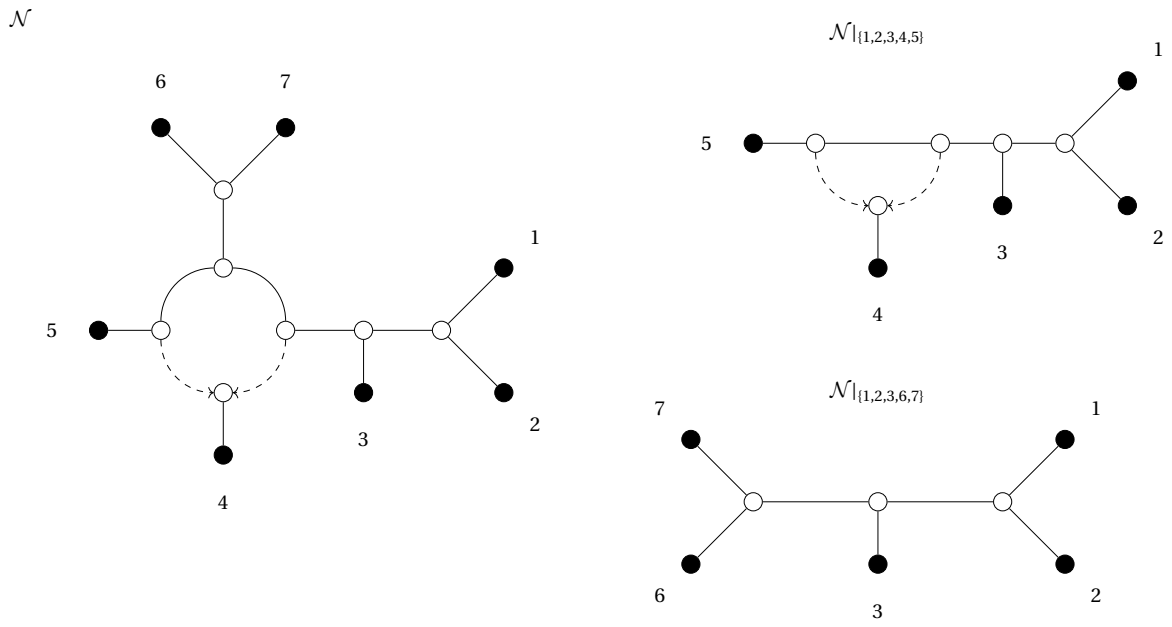


Figure 2.3: Example of two restrictions on network $\mathcal{N}$, one with the first five leaves, another with the set of leaves $\{1,2,3,6,7\}$.

A semi-directed phylogenetic network $\mathcal{N}$ on $\mathcal{X} = \{x_1, x_2, ..., x_n\}$ has exactly $\binom{n}{4}$ quarnets. $\mathcal{Q}^s(\mathcal{N})$ gives the list of all the quarnet-splits in $\mathcal{N}$, where every $q \in \mathcal{Q}^s(\mathcal{N})$ is written as $x_a x_b | x_c x_d$ for some $a, b, c, d \in \{1, 2, .., n\}$. We define $L(q)$ as the function that gives a set of the leaves in q, e.g., if $q = ab|cd$, then $L(q) = \{a, b, c, d\}$.

3

The Construction of a Blob-Tree

In this section, we guide the reader through the process of constructing a blob-tree $\mathcal{T}(\mathcal{N})$ of a network $\mathcal{N}$, with merely the quarnet-splits of $\mathcal{N}$ as information. The blob-tree is not necessary for the construction of a level-1 network, it is even less efficient, as the construction of a blob-tree costs $\mathcal{O}(n^3)$ time, while the more efficient way to construct a level-1 network requires only $\mathcal{O}(n\log(n))$ time, where n is the number of leaves in the network [2]. But in this thesis we want to focus on the construction of level-2 networks and possibly higher, wherefore the blob-tree of the network is a useful tool with the current construction. However, if the reader is interested in the optimised algorithm for level-1 networks, we refer to article [2]. The process and definitions in this chapter are based on the same article.

3.1. The initial tree

Let $\mathcal{N}$ be the network on the set of leaves $\mathcal{X}$, with its corresponding blob-tree $\mathcal{T}(\mathcal{N})$ that we want to construct. The idea of the process is iteratively attaching a new leaf to the tree. Of course, for that to work, we need a tree to start with. We know that all leaves need a degree of one, and that the degree of an internal node should be at least three. So, the biggest tree we can start with is a combination of the first three leaves of $\mathcal{X}$. We do this by attaching those three leaves to one internal node, as in Figure 3.1.

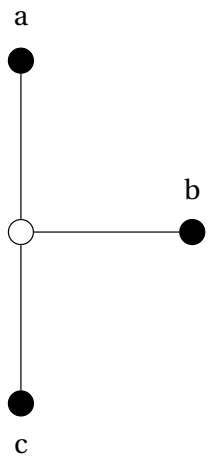


Figure 3.1: The initial tree $\mathcal{N}|_{\{a,b,c\}}$, where to blob-tree will be built on.

Now, from here on, we can try to attach a new leaf $x \in \mathcal{X}\setminus\mathcal{Y}$ to this tree, where $\mathcal{Y}$ is the set of leaves already present in the tree. This, we do based on the types of edges we have in the tree, where the type of an edge is based on the available quarnet-splits. There are three types of edges that can be present in a blob-tree. An edge can be a *strong stem edge*, a *weak stem edge*, where multiple weak edges form a *weak stem path*, or a directed edge.

3.2. Edge types

Each edge type is based on the splits it can induce with its removal. As we work with a tree, we know that each edge has to be a cut-edge. Thus, each edge induces a split $A|B$, with $A \cup B = \mathcal{Y}$. Now, we want to attach $x \in \mathcal{X} \backslash \mathcal{Y}$ to this tree. Based on the available quarnet-splits, we can determine if a cut-edge also induces the split $A|B \cup |\{x\}$ or $A \cup \{x\}|B$.

Lemma 1. [2] Given a semi-directed network $\mathcal{N}$ on $\mathcal{X}$, a non-trivial partition $A|B$ of $\mathcal{X}$ and any $a_1 \in A$, $b_1 \in B$, the following are equivalent:

(i) $A|B$ is a split in $\mathcal{N}$;

(ii) $a_1a_2|b_1b_2$ is a quarnet-split of $\mathcal{N}$ for all $a_2 \in A\backslash\{a_1\}, b_2 \in B\backslash\{b_1\}$.

Based on Lemma 1, we can determine if $A|B \cup |\{x\}$ is a split. If for all $a_1, a_2 \in A$ and for all $b_1 \in B$ the quarnet-split $a_1a_2|b_1x$ exists, we have the split $A|B \cup |\{x\}$. For the split $A \cup \{x\}|B$, the idea is the same, only mirrored. Now that we know how these splits can be found, we can define all edge types in the blob-tree.

Definition 6. [2] Given a semi-directed network $\mathcal{N}$ on $\mathcal{X}$ and $\mathcal{Y} \subset \mathcal{X}$, let uv be an edge in $\mathcal{T}(\mathcal{N}|_\mathcal{Y})$ (with u towards A and v towards B) inducing the split $A|B$ and $x \in \mathcal{X} \backslash \mathcal{Y}$. The edge uv is one of the following:

- a *strong stem edge (for x)*, if $A|B \cup |\{x\}$ and $A \cup \{x\}|B$ are splits in $\mathcal{N}|_{\mathcal{Y}\cup\{x\}}$;
- a *weak stem edge (for x)*, if neither $A|B \cup |\{x\}$ nor $A \cup \{x\}|B$ are splits in $\mathcal{N}|_{\mathcal{Y}\cup\{x\}}$;
- a *pointing edge (for x) with orientation uv*, if $A|B \cup |\{x\}$ is a split in $\mathcal{N}|_{\mathcal{Y}\cup\{x\}}$, but $A \cup \{x\}|B$ is not;
- a *pointing edge (for x) with orientation vu*, if $A|B \cup |\{x\}$ is not a split in $\mathcal{N}|_{\mathcal{Y}\cup\{x\}}$, but $A \cup \{x\}|B$ is;

An internal vertex v of $\mathcal{T}(\mathcal{N}|_\mathcal{Y})$ is a *stem vertex (for x)* if all its incident edges uv in $\mathcal{T}(\mathcal{N}|_\mathcal{Y})$ are pointing edges for x with orientation uv. A subtree of $\mathcal{T}(\mathcal{N}|_\mathcal{Y})$ is a *weak stem subtree (for x)* if it is a maximal subtree containing only weak stem edges for x.

3.3. Addition of a new leaf

The edge types give us information about the *stem* for x. For every blob-tree of a restriction, there is always a unique stem for the new leaf x that we want to attach. This can be either a strong stem edge, a stem vertex, or a weak stem subtree. Each type of stem has a unique way of attaching the new leaf to it, which is explained in Lemma 2. The proof for uniqueness and the processes have been proved in article [2], therefore, we refer to the article for a full proof.

Lemma 2. [2] Let $\mathcal{N}$ be a semi-directed network on $\mathcal{X}$ and let $\mathcal{Y} \subset \mathcal{X}$. Given the blob-tree $\mathcal{T}(\mathcal{N}|_\mathcal{Y})$ and some $x \in \mathcal{X} \backslash \mathcal{Y}$, there is a unique stem and the orientation of any pointing edge in $\mathcal{T}(\mathcal{N}|_\mathcal{Y})$ is towards this stem. Furthermore,

(i) if the stem is a strong stem edge, then the blob-tree $\mathcal{T}(\mathcal{N}|_{\mathcal{Y}\cup\{x\}})$ can be obtained from $\mathcal{T}(\mathcal{N}|_\mathcal{Y})$ by attaching x to this edge;

(ii) if the stem is a weak stem subtree, then the blob-tree $\mathcal{T}(\mathcal{N}|_{\mathcal{Y}\cup\{x\}})$ can be obtained from $\mathcal{T}(\mathcal{N}|_\mathcal{Y})$ by contracting this subtree to a single vertex and attaching x to it;

(iii) if the stem is a stem vertex, the blob-tree $\mathcal{T}(\mathcal{N}|_{\mathcal{Y}\cup\{x\}})$ can be obtained from $\mathcal{T}(\mathcal{N}|_\mathcal{Y})$ by attaching x to this vertex.

The idea of attaching a leaf to an edge could be wrongly interpreted, therefore we give a more detailed description of the process: we subdivide the strong stem edge, thus creating a new internal node between the two nodes the edge was connected to. Then, we add an edge between x and this new internal node, which we call the attachment of a node to another node, as is also done in (ii) and (iii).

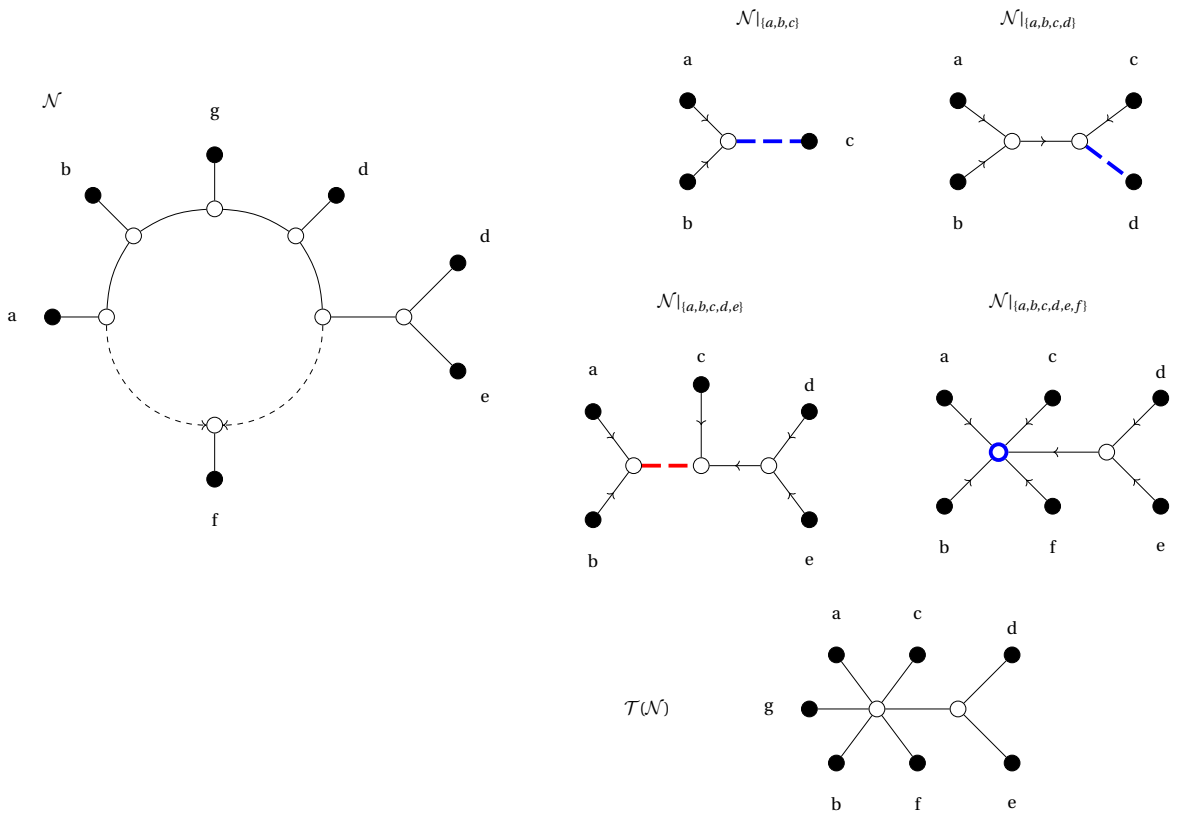


Figure 3.2: Process of the construction of the blob-tree $\mathcal{T}(\mathcal{N})$ using Definition 6 to determine the stem for the next leaf and using Lemma 2 to attach the next leaf. Here, the blue dashed edges indicate strong edges, red dashed edges indicate weak edges and a blue vertex represents a stem vertex.

4

Construction of Level-1 Network

With the help of the found blob-tree of a level-1 network $\mathcal{N}$, we can start to construct the network by 'inflating' the internal nodes. However, we cannot find the complete level-1 network, but only a canonical version of this. The canonical version of a network is the closest form to the complete network we can find using merely quarnet-splits.

In this chapter, we will show the process of constructing the canonical form of a level-1 network using information of the blob-tree of that same network.

4.1. Canonical form

In a level-1 network, each blob is either a vertex of degree three, or a k-cycle. Such a cycle is called a *large cycle* when $k \geq 5$. Furthermore, an edge in a cycle is called a *contraction edge* if an adjacent edge is a reticulation edge. Thus, every large cycle has exactly two contraction edges. The canonical form of a level-1 network $\mathcal{N}$ is defined as follows:

Definition 7. (Canonical form). Given a semi-directed level-1 network $\mathcal{N}$ on $\mathcal{X}$, the *canonical form* $\mathcal{N}^c$ of $\mathcal{N}$ is obtained by contracting every 3 and 4-cycle to a single vertex; and by contracting the cycle contraction edges of every large cycle.

This definition and why it is the closest form we can find to the complete network may not be immediately intuitive, but later in this chapter, this will be expanded on and will be more clear.

4.2. Inflating single internal nodes

This section will explain the process of 'inflating' internal nodes in the blob-tree to level-1 blobs. As we know from the definition of the canonical form, and as the canonical form is the best form we can construct with this method, only internal nodes with a degree of 5 should be considered for this process. First, we look at an example of a blob-tree $\mathcal{T}(\mathcal{N})$ with one internal node, and with eight adjacent leaves. This network $\mathcal{N}$ on $\mathcal{X} = \{a, b, c, d, e, f, g, h\}$ and its corresponding blob-tree $\mathcal{T}(\mathcal{N})$ are visualised in Figure 4.2.

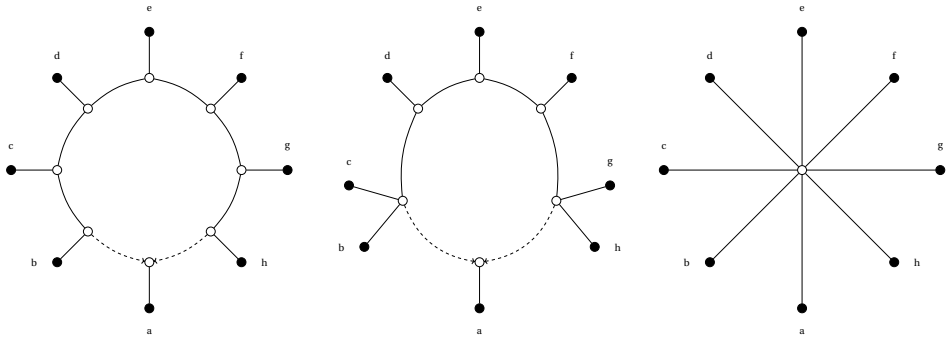


Figure 4.1: Network $\mathcal{N}$ on $\mathcal{X} = \{a, b, c, d, e, f, g, h\}$ on the left, its canonical form $\mathcal{N}^c$ in the middle and its corresponding blob-tree $\mathcal{T}(\mathcal{N})$ on the right.

Before we can begin the construction from the blob-tree to the canonical form of the network, we first have to know the reticulation node of this blob, which follows from the quarnet-splits of the network.

Lemma 3. Given a network $\mathcal{N}$ on $\mathcal{X}$ with $|\mathcal{X}| \geq 5$, such that there is only one internal blob in the network that is a k-cycle with $k \geq 5$. Then, the leaf under the reticulation-node is the only leaf not present in the quarnet-splits of $\mathcal{N}$.

Proof. Let $\mathcal{N}$ be a semi-directed network on $\mathcal{X}$ with one internal blob that is a k-cycle with $k \geq 5$. Let $x \in \mathcal{X}$ be the leaf under the reticulation-node. Then, for any $a, b, c \in \mathcal{X}$, the restriction $\mathcal{N}|_{\{x,a,b,c\}}$ is always a 4-cycle. As in a 4-cycle no non-trivial cut-edges are present, there exists no quarnet-split including x. For any $a, b, c, d \in \mathcal{X}\backslash\{x\}$, the restriction $\mathcal{N}|_{\{a,b,c,d\}}$ includes no reticulation-node, therefore it contains no cycle. So, there is at least one non-trivial cut-edge resulting in a quarnet-split including a, b, c and d. □

Given Lemma 3, with the help of the given quarnet-splits we can find the reticulation node of an internal node in a blob-tree that has at least 5 adjacent nodes. With this information, we can create the canonical form from the blob-tree. As blobs with at most 4 adjacent leaves are the same in their canonical form as in the blob-tree, we can ignore them in this stage. Let us take Figure 4.2 as an example network. Note that we can split up the canonical form into two parts. The first part is the reticulation node together with the leaf under this node. The second part is a tree of the resulting nodes and leaves. We can achieve these two parts by removing the reticulation edges and taking the two components as loose parts. So in this example, the first part is the reticulation node attached to leaf a, the second part is a tree on the leaves $\{b, c, d, e, f, g, h\}$. With careful observation, we can see that the second part is exactly the same as the blob-tree on the nodes $\{b, c, d, e, f, g, h\}$. Therefore, we can construct the second part in the same way as we would construct a blob-tree with these given nodes. Now that we have both parts, we can combine them by attaching a reticulation edge coming from both end-internal nodes of the second part into the reticulation-node of the first part, thus completing the canonical form.

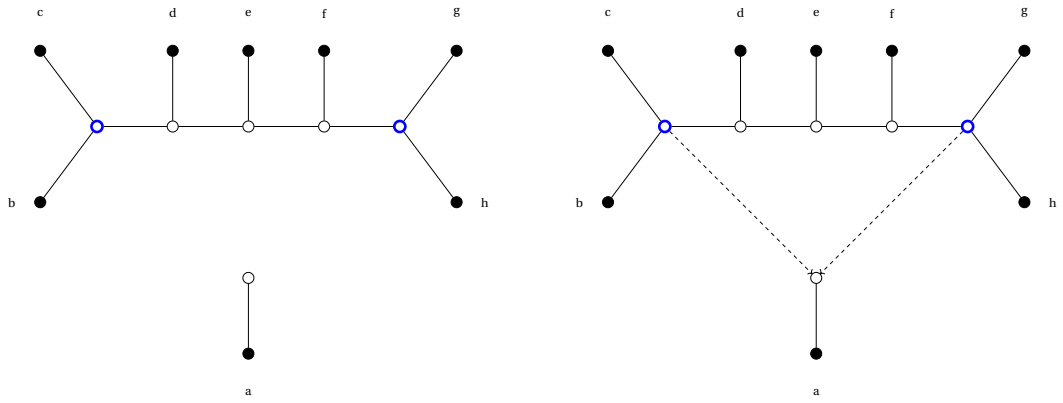


Figure 4.2: On the left, the two separate parts of the canonical form after removing the reticulation edges, with the end-internal nodes marked with blue. On the right, the two parts combined with two reticulation edges from the end-internal nodes to the reticulation node.

4.3. Inflating bigger blob-trees

In the previous section, we have seen how we can create the canonical form of a level-1 network with a single internal blob. However, a general level-1 phylogenetic network has more than one blob. Therefore, we have to give extra steps to the inflation process. For the inflation, we want to apply this to each internal node of the blob-tree individually. So, the first step will be to divide the blob-tree into multiple parts, where each part consists of one internal node and k adjacent leaves. The second step is to apply the inflation process to each individual internal node. The third step will be the glueing together of the resulting level-1 blobs.

4.3.1. Dividing the blob-tree

Let $\mathcal{T}(\mathcal{N})$ be the blob-tree of a network $\mathcal{N}$, and let v be an internal node in this blob-tree with k adjacent nodes. As a blob-tree contains no cycles, we can safely say that each edge vx_i induces a split. Following this, let X_i be the sub-graph containing x_i after this split. As v has k adjacent nodes, there exist k sub-graphs in this form, where we call each sub-graph X_i. In other words, there exists a partition $X_1|...|X_k|\{v\}$ of $\mathcal{T}(\mathcal{N})$,

where for each X_i there exists a $x \in X_i$ such that the edge $vx \in \mathcal{T}(\mathcal{N})$. Let u_i be a random leaf in X_i. Then, the blob-tree on $\mathcal{N}_{\{u_1,\ldots,u_k\}}$ will result in a tree with one internal node and k adjacent leaves. We can apply this process to each individual internal node in the blob-tree $\mathcal{T}(\mathcal{N})$, resulting in multiple components on which we can apply the inflation process.

4.3.2. Glueing blobs

Let $G = \mathcal{T}(\mathcal{N})$. After separating each internal node in G and after inflation, we can put these simple networks back together. We can do this, by replacing each internal node in G with its corresponding obtained simple network. Let u be this node, and $\mathcal{N}^c||_{\mathcal{Y}}$ the corresponding simple network. We remove u from G. Then, let X_i in G be the components resulting from this removal. For each leaf $x_i \in \mathcal{Y}$, if $x_i \in X_i$, then we replace the leaf $x_1 \in \mathcal{N}^c||_{\mathcal{Y}}$ with the component X_i, with the node $v \in X_i$ that was adjacent to u, now neighbouring the node $w \in \mathcal{N}^c||_{\mathcal{Y}}$ that was adjacent to x_i.

Algorithm 1 Algorithm for inflating a blob-tree of a level-1 phylogenetic network into the canonical form of this network, using its corresponding quarnet-splits.

Data: quarnet-splits $\mathcal{Q}^s(\mathcal{N})$ of a semi-directed level-1 network $\mathcal{N}$ on $\mathcal{X}$; the blob-tree $\mathcal{T}(\mathcal{N})$
Result: The canonical form $\mathcal{N}^c$ of a semi-directed phylogenetic network $\mathcal{N}$

Let G be $\mathcal{T}(\mathcal{N})$;
foreach *internal node u in G* **do**
    Isolate the node by taking a random leaf from each pendant subtree of u, resulting in the leaf-set $\mathcal{Y}$;
    if $|\mathcal{Y}| < 5$ **then**
        continue
    Using Lemma 3, we find the reticulation node and its reticulation leaf a of the simple network u will be inflated to;
    Construct the tree $\mathcal{C}$ with the leaves $\mathcal{Y} \backslash \{a\}$;
    Attach a to $\mathcal{C}$ using the weak edge path for a;
    Add $\mathcal{C}$ back to G by replacing u with $\mathcal{C}$ and substituting each leaf of $\mathcal{C}$ with its corresponding pendant subtree/subgraph;
return G

5

Level-2 Blobs

This chapter contains the main contribution to previous works on this topic. We will look at different versions of level-2 blobs, what distinguishes them from each other with only quarnet-splits as information, and how we can find the canonical form of level-2 networks using quarnet-splits and the corresponding blob-tree as information.

5.1. Types of level-2 blobs

All simple, strict level-2 networks on a leaf-set $\mathcal{X}$ can be constructed from a level-2 generator. This generator consists of two nodes u and v with three parallel arcs connecting them. The arcs can be subdivided into the paths P_1, P_2, P_3 such that at least two of these paths have a length of at least two, and such that for $i \in \{1,2,3\}$, for every $w \in V(P_i)$, w is adjacent to a leaf in $\mathcal{X}$. This generator can be classified into two different semi-directed level-2 generators. These generators consist of two reticulation nodes and, therefore, four reticulation edges. Simple, strict level-2 networks can be obtained from these generators by subdividing edges and attaching leaves to the resulting degree-2 nodes and to the degree-2 reticulation nodes. Both generators are visualised in Figure 5.1. [3]
For easier notation, let us call the generator on the left type-1 and the generator on the right type-2.

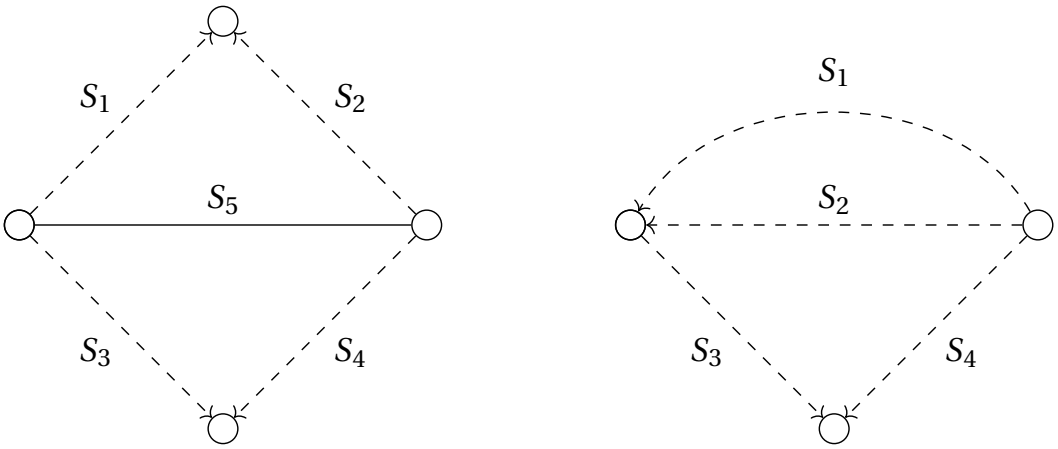


Figure 5.1: The two unique level-2 generators that represent all level-2 blobs. On the left, the generator of type-1, on the right, the generator of type-2.

With the knowledge of what a level-2 blob can look like, we can now try to obtain the structure of an unknown level-2 blob with information from the blob-tree and its corresponding quarnet-splits. In the same way that we inflated an internal node from a blob-tree to a level-1 blob, we can inflate an internal node to a level-2 blob. Unlike level-1 blobs, level-2 blobs appear as multiple different cases, where each has to be explored on its own.

5.2. Acquiring simple network cases

Each simple network case depends on the number of leaves under the reticulation nodes, the number of leaves attached to specific paths of the generators and the type of generator itself. These cases can be identified using the quarnet-splits on the leaves of the simple network. We will reuse the idea from the previous chapter of dividing a blob-tree into multiple components consisting of one internal node and a set $\mathcal{Y} \subset \mathcal{X}$ with $|\mathcal{Y}| = k$ with $k \geq 6$. Then, from the list of all quarnet-splits $\mathcal{Q}^s(\mathcal{N})$, we take the subset of all quarnets that include only leaves that are also present in $\mathcal{Y}$, which we will note as $\mathcal{Q}^s_{\mathcal{Y}}(\mathcal{N})$.

The cases of simple networks can firstly be distinguished by using the number of elements in $\mathcal{Y}$ that are not in any of the quarnet-splits in $\mathcal{Q}^s_{\mathcal{Y}}(\mathcal{N})$. In other 'words', $\#\{x : \forall q \in \mathcal{Q}^s_{\mathcal{Y}}(\mathcal{N}), x \notin L(q)\}$. For simple networks with 6 or more leaves, this number can be either zero, one or two. For easier notation, we name this number the *number of missing leaves*, and if a leaf is not present in any of these quarnet-splits, we call this a *missing leaf*.

5.3. Basic Case

Strict level-2 simple networks can have many different cases, where the canonical form and construction of the network depend on the number of leaves on each side of the generators. Before we look at all these different cases, we first take the basic easier case where each side on the generator has at least two leaves attached. By using Lemma 4, 6 and 8, we find that the number of leaves is either zero or one, where if the number is zero, the blob of the simple network is of type-1 and if the number is one, the blob of the simple network is of type-2. Note that, as each side has at least two leaves, the number of leaves in the simple network is at least 9. The type-2 generator has four sides with at least two leaves and one leaf under the sink. The type-1 generator has five sides with at least two leaves and one leaf under both sinks, which gives at least 12 leaves on a type-1 simple network.

5.3.1. Canonical form

Let $\mathcal{N}$ be a simple strict level-2 network generated from a type-1 or type-2 generator with each side in the generator having at least two leaves. Then, an edge that is part of the blob and adjacent to a reticulation edge is called a *cycle contraction edge*.

Definition 8. Let $\mathcal{N}$ be a simple strict level-2 network generated from a type-1 or type-2 generator with each side in the generator having at least two leaves. Then, the canonical form $\mathcal{N}^c$ is obtained by contracting each cycle contraction edge.

5.3.2. Construction of the canonical form

Given is $\mathcal{T}(\mathcal{N})$ of $\mathcal{N}$ on $\mathcal{X}$, which is one internal node with at least 9 leaves. Given is also the quarnet-splits $\mathcal{Q}^s(\mathcal{N})$.
Suppose that the number of missing leaves in $\mathcal{Q}^s(\mathcal{N})$ is zero. Using Lemma 4, we know that the blob must be of type-1. Then, using Lemma 5, we can find the two leaves x, y under the sinks. First, we run Algorithm 1 on the set of leaves $\mathcal{X}\backslash\{x, y\}$. As we know that the restriction of a type-1 simple network on the leaf-set without the leaves under the reticulation nodes results in a tree, the Algorithm returns a tree. The following construction is similar to the construction of a level-1 blob using the inflation method. Here, for both x, y, we determine the weak stem path. Let s_1, s_2 be the weak stem paths for x, y respectively. Then, we attach a reticulation node to the ends of s_1 with two reticulation edges pointing to the reticulation node. Finally, we attach x to this reticulation node. Then we do a similar process with y and s_2. Here, we have assumed that each side of the type-1 blob had at least two leaves. Without this assumption, the construction would be more complex, which will be discussed in detail in the next section.

Now suppose that the number of missing leaves in $\mathcal{Q}^s(\mathcal{N})$ is one, with x being the missing leaf. Here, for the sake of this section, we assume that the resulting simple network should be of type-2 with at least two leaves on each side of the generator, as with one missing leaf, the simple network could be of all types. First, we run Algorithm 1 on the set of leaves $\mathcal{X}\backslash\{x\}$. If we look at the restriction of a network of type-2 with at least two leaves on each side on the leaf-set without the missing leaf, which is under the sink, we see that this always results in a level-1 cycle blob with two pendant subtrees. Thus, the Algorithm returns such a network with a level-1 cycle blob with two pendant subtrees. Then, we attach a reticulation node to both ends of the pendant subtrees with reticulation edges pointing towards the reticulation node. Finally, we attach x to this

reticulation node.

5.4. Special cases

The construction of the basic case is similar to the construction of level-1 blobs. However, there are a lot of cases where the construction is more complicated, where different shaped simple networks have equivalent sets of quarnet-splits. Therefore, we look with more detail to each case that depends on the number of missing leaves. Here, the canonical form of the network also depends on the shape of the network, which has to be defined individually per case.

Definition 9. Let $\mathcal{N}$ on $\mathcal{X}$ be a semi-directed level-2 phylogenetic network. Then, the canonical form $\mathcal{N}^c$ is obtained by using the Definitions 10,11 and 12 for each blob, depending on the shape of the blob.

Theorem 1. Given the quarnet-splits of a semi-directed level-2 network $\mathcal{N}$ on $\mathcal{X} = \{x_1, x_2, ..., x_n\}$, there exists an algorithm that constructs the canonical form of $\mathcal{N}$.

The proof is given at the end of this chapter.

5.4.1. No missing leaves

If the number of missing leaves is zero for a simple level-2 network, there is only one general type of a simple level-2 network that can occur, which is proven in Lemma 4 below.

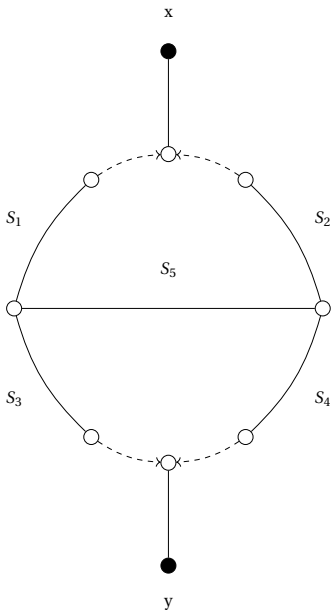


Figure 5.2: The only simple network on at least six leaves that can occur with no missing leaves.

In Figure 5.2, this type of a simple level-2 network can be seen. In this simple network, x, y are the leaves under the reticulation nodes and S_i represent the leaves attached to its corresponding edge. With this notation, we can state the following lemma.

Lemma 4. Let $\mathcal{N}$ be a level-2 semi-directed phylogenetic network on $\mathcal{X}$ with its corresponding quarnet-splits $\mathcal{Q}^s(\mathcal{N})$. Let $\mathcal{Y} \subseteq \mathcal{X}$ be such that $\mathcal{N}|_\mathcal{Y}$ is simple and $|\mathcal{Y}| \geq 6$, with its corresponding quarnet-splits $\mathcal{Q}^s_\mathcal{Y}(\mathcal{N})$. Then, the blob in the restriction is of type-1 and $|S_1|$ or $|S_2| \geq 2$ and $|S_3|$ or $|S_4| \geq 2$, if and only if $\forall x \in \mathcal{Y}, \exists q \in \mathcal{Q}^s_\mathcal{Y}(\mathcal{N})$, such that $x \in L(q)$.

Proof. Let $\mathcal{N}$ be a level-2 semi-directed phylogenetic network on $\mathcal{X}$. Let $\mathcal{Y} \subseteq \mathcal{X}$ be a subset of leaves such that $\mathcal{N}|_\mathcal{Y}$ is simple. Suppose that this is a type-1 level-2 blob and that $|S_1|$ or $|S_2| \geq 2$ and $|S_3|$ or $|S_4| \geq 2$. Given $a \in \mathcal{Y}$, we want to show that $\exists q \in \mathcal{Q}^s_\mathcal{Y}(\mathcal{N})$ *such that* $a \in L(q)$. If a, b, c, d are not under a reticulation node in $\mathcal{N}|_\mathcal{Y}$, the quarnet $\mathcal{N}|_{\{a,b,c,d\}}$ is level-0; therefore, there exists a non-trivial cut-edge in this quarnet. Thus,

for a,b,c,d, $\exists q \in \mathcal{Q}^s_{\mathcal{Y}}(\mathcal{N})$ *such that* $a,b,c,d \in L(q)$. Now, let $x,y \in \mathcal{Y}$ be under a reticulation node. Take the restriction $\mathcal{N}_{\mathcal{Y}\setminus\{y\}}$. As $|S_1|or|S_2| \geq 2$ and $|S_3|or|S_4| \geq 2$, the blob-tree $\mathcal{T}(\mathcal{N}_{\mathcal{Y}\setminus\{y\}})$ has more than one internal node, thus $\exists q \in \mathcal{Q}^s_{\mathcal{Y}}(\mathcal{N})$ *such that* $x \in L(q)$. This mirrors for y.

Let $\mathcal{N}$ be a level-2 semi-directed phylogenetic network on $\mathcal{X}$. Let $\mathcal{Y} \subseteq \mathcal{X}$ be a subset of leaves such that $\mathcal{N}|_{\mathcal{Y}}$ is simple. Suppose that $\forall a \in \mathcal{Y}, \exists q \in \mathcal{Q}^s_{\mathcal{Y}}(\mathcal{N})$ *such that* $a \in L(q)$. Then the blob cannot be level-1, as a level-1 k-blob with $k \geq 6$ always has a missing leaf. The blob cannot be a level-2 type-2 k-blob with $k \geq 6$, since this always has at least one missing leaf. Thus, it must be a level-2 type-1 blob. Using Figure 5.2, suppose that $|S_3|, |S_4| < 2$, then $\mathcal{T}(\mathcal{N}_{\mathcal{Y}\setminus\{y\}})$ has one internal blob, thus $\exists a \in \mathcal{Y}$ *such that* $\forall q \in \mathcal{Q}^s_{\mathcal{Y}}(\mathcal{N}), a \notin L(q)$. As seen before, $\mathcal{T}(\mathcal{N}_{\mathcal{Y}\setminus\{y\}})$ has more than one internal node when $|S_3|or|S_4| > 1$, therefore, it must be the case that $|S_3|$ or $|S_4|$ is greater than one. This mirrors for $|S_1|$ and $|S_2|$. □

Here, $|S_i|$ describes the number of leaves attached to this edge. As both leaves under the reticulation nodes are present in the quarnet-splits, these are more difficult to find. Therefore, we have to look at the combinations of quarnets that are available. We know that in a tree, a level-0 network, each two leaves are together present in at least one quarnet-split. Thus, in a level-k blob with at least 6 leaves, each two leaves that are not under a reticulation node are together in at least one quarnet-split. This is formalized in the next lemma.

Lemma 5. Let $\mathcal{N}$ be a level-2 semi-directed phylogenetic network on $\mathcal{X}$ with its corresponding quarnet-splits $\mathcal{Q}^s(\mathcal{N})$. Let $\mathcal{Y} \subseteq \mathcal{X}$ be such that $\mathcal{N}|_{\mathcal{Y}}$ is simple and $|\mathcal{Y}| = k \geq 6$, with its corresponding quarnet-splits $\mathcal{Q}^s_{\mathcal{Y}}(\mathcal{N})$ without missing leaves. Let $N_{\mathcal{Y}}(x) = \{y : \exists q \in \mathcal{Q}^s_{\mathcal{Y}}(\mathcal{N})$ such that $x,y \in L(q)\}$. Then for $x \in \mathcal{Y}$, $|N_{\mathcal{Y}}(x)| < k$ if and only if x is under a reticulation node.

Proof. As we know that $\mathcal{Q}^s_{\mathcal{Y}}(\mathcal{N})$ has no missing leaves and $|\mathcal{Y}| = k \geq 6$, the blob $\mathcal{B}$ in $\mathcal{N}|_{\mathcal{Y}}$ is strict level-2 of type-1, thus there are only two leaves x and y under separate reticulation nodes. Let $a,b,c,d \in \mathcal{Y}$ be not under a reticulation node in $\mathcal{N}|_{\mathcal{Y}}$. Then $\mathcal{N}|_{\{a,b,c,d\}}$ is always level-0. This holds for all $a,b,c,d \in \mathcal{Y}\setminus\{x,y\}$, and thus $|N_{\mathcal{Y}}(a)| \geq k-2$ for all $a \in \mathcal{Y}\setminus\{x,y\}$.
Using Figure 5.2, let x be one of the reticulation leaves and let $|S_3| \geq 2$. Let $a \in S_1 \cup S_2 \cup S_4 \cup S_5$ be arbitrary, and let $b,c \in S_3$ be arbitrary. Then the quarnet-split $ax|bc$ exists. This mirrors for $S_4 \geq 2$ and also for y. Thus, for any $a \in \mathcal{Y}\setminus\{x,y\}$, there exist quarnet-splits $q1, q2$ such that $a,x \in q1$ and $a,y \in q2$. This gives that $|N_{\mathcal{Y}}(a)| = k$ for all $a \in \mathcal{Y}\setminus\{x,y\}$.
Suppose x,y are the reticulation nodes. Let $a,b \in \mathcal{Y}\setminus\{x,y\}$ be arbitrary. As both x,y are under separate reticulation nodes, the restriction $\mathcal{N}|_{\{x,y,a,b\}}$ is a simple level-2 network, thus it has no non-trivial cut-edges, and therefore there is no quarnet-split q such that $x,y \in q$. Thus, $|N_{\mathcal{Y}}(x)| < k, |N_{\mathcal{Y}}(y)| < k$. □

First, we define the canonical form of a level-2 blob, such that there are no missing leaves, and hence the blob is of type 1 by Lemma 4. Let us take Figure 5.2 as a reference. If we remove the reticulation node above y, the remainder is a circular level-1 blob with one or two pendant subtrees (degree-2 nodes may be present). Then, in the cycle of the blob, if an edge is incident to a reticulation edge, then this is called a *cycle contraction edge*. Now we do the same, but with the removal of the reticulation node above x. An example of this can be found in Figure 5.3

Definition 10. Given a semi-directed level-2 network $\mathcal{N}$ on $\mathcal{X}$, and $\mathcal{Y} \subseteq \mathcal{X}$ such that $\mathcal{N}|_{\mathcal{Y}}$ is simple and $|\mathcal{Y}| \geq 6$. If $\mathcal{Q}^s_{\mathcal{Y}}(\mathcal{N})$ has zero missing leaves, then the canonical form of this blob is obtained by contracting all cycle contraction edges.

Now that we have defined the canonical form of a level-2 blob with no missing leaves, we can construct a simple strict level-2 network $\mathcal{N}$ with no missing leaves by its quarnet-splits $\mathcal{Q}^s(\mathcal{N})$. This construction is described in detail in Construction 1 below.

Construction 1. Using Lemma 5, we can find both leaves under the reticulation nodes, by searching for the leaves x, such that $N_{\mathcal{Y}}(x) < k$. Let x,y be the reticulation leaves. With this knowledge, we can start to build the level-2 blob. We do this by first constructing the blob without the reticulation nodes, which will result in a tree. In this tree, we determine the stem for x and y simultaneously, without yet attaching them. We do this using the same method as previously stated in Definition 6. Each stem can result in either a stem vertex or a weak stem path.

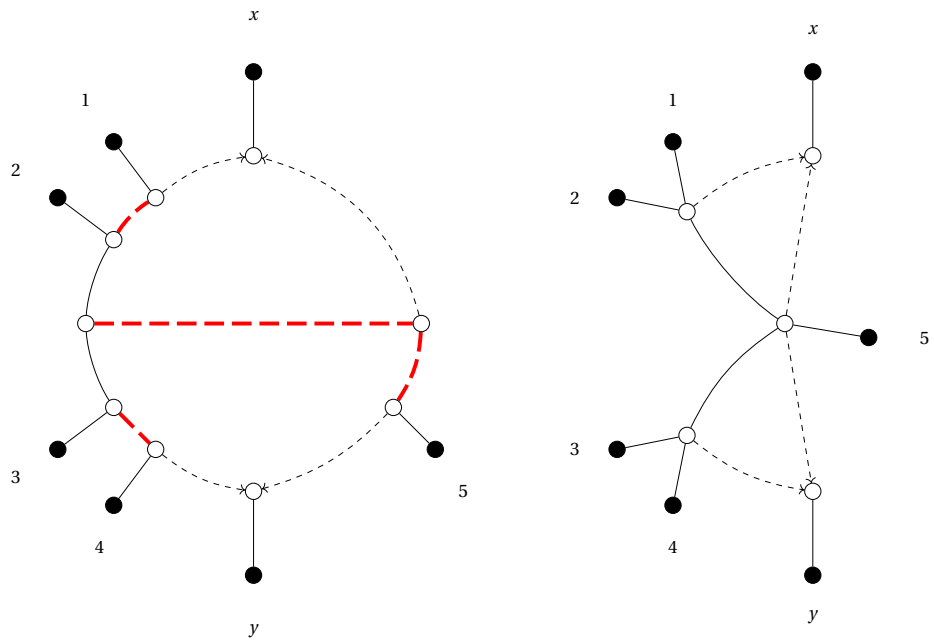


Figure 5.3: Let $\mathcal{N}$ be the network on the left. Then, as $\mathcal{Q}^s(\mathcal{N})$ has no missing leaves, the canonical form on the right is obtained as described in Definition 10. Here, the red thick dashed edges are the cycle contraction edges.

(1) Suppose that both x and y have a weak stem path, then, for both x and y, we add two reticulation edges from the ends of the corresponding weak stem path to a reticulation node directly above x or y, to which x or y is attached.

(2) Suppose that both x and y have a stem vertex. Then, there exists an edge between those two vertices. We subdivide this edge, resulting in a node u with its two stem vertex neighbours. Then, for both x and y, we add two reticulation edges, one outgoing from its corresponding stem vertex, one outgoing from the node u, and both directed to the reticulation node directly above x or y, to which x or y is attached.

(3) Suppose that x has a stem vertex and y has a weak stem path. Then, for x, we add a reticulation edge outgoing from its corresponding stem vertex, directed to the reticulation node directly above x. (a) Then, if there exists an internal node u with three adjacent internal non-reticulation nodes, we add a reticulation edge outgoing from u towards the reticulation node directly above x. (b) If this does not exist, we add a reticulation edge outgoing from the end of the weak stem path of y, where the end has only one adjacent leaf, to the reticulation node directly above x. For y, we add two reticulation edges, outgoing from both ends of the weak stem path, directed to the reticulation node directly above y. This mirrors for the case of x having a weak stem path and y having a stem vertex.

5.4.2. One missing leaf

In the case of one missing leaf, there are three types of blobs that can be considered. First, obviously from the previous section, this could be a level-1 blob. Second, both types of level-2 blobs can be the case, but again, only in specific cases within the type of blob.

For a level-2 blob of type-2 as shown in Figure 5.1, one leaf is missing if S_3 has more than 1 leaf attached to it, if $|S_4| \geq 2$, or if $|S_3| = 0$. For a level-2 blob of type-1 as shown in Figure 5.2, one leaf is missing if either

(i) $(|S_1| \geq 2 \vee |S_2| \geq 2) \wedge (|S_3|, |S_4| < 2)$

(ii) $(|S_3| \geq 2 \vee |S_4| \geq 2) \wedge (|S_1|, |S_2| < 2)$

Lemma 6. Let $\mathcal{N}$ be a level-2 semi-directed phylogenetic network on $\mathcal{X}$ with its corresponding quarnet-splits $\mathcal{Q}^s(\mathcal{N})$. Let $\mathcal{Y} \subseteq \mathcal{X}$ be such that $\mathcal{N}|_\mathcal{Y}$ is simple and $|\mathcal{Y}| \geq 6$, with its corresponding quarnet-splits $\mathcal{Q}^s_\mathcal{Y}(\mathcal{N})$. Then, $\mathcal{Q}^s_\mathcal{Y}(\mathcal{N})$ has one missing leaf, if and only if

1. $\mathcal{N}|_\mathcal{Y}$ is strict level-1.

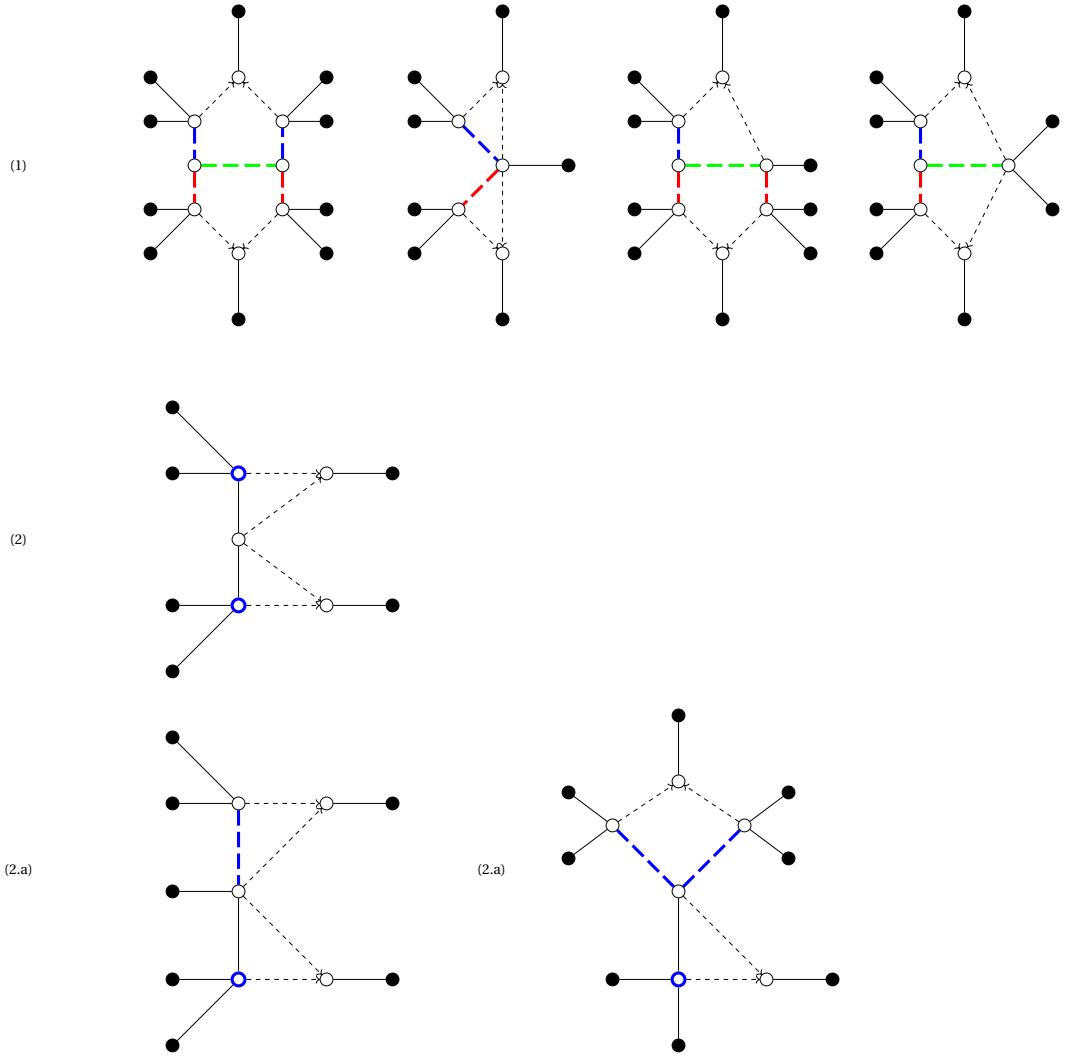


Figure 5.4: The obtained canonical forms after using Construction 1. Blue and red thick dashed edges are the weak stem paths, green thick dashed edges are overlapping weak stem path edges (so both red and blue), and blue nodes visualise the stem vertices. To each edge that is either blue, red or green, and arbitrary number of leaves can be attached.

2. $\mathcal{N}|_\mathcal{Y}$ is strict level-2 with its only blob being type-1, and x being below one reticulation node, and y being below the other reticulation node, such that the restriction $\mathcal{N}|_{\mathcal{Y}\setminus\{x\}}$ is simple and the restriction $\mathcal{Y}\setminus\{y\}$ is not simple (or the other way around).

3. $\mathcal{N}|_\mathcal{Y}$ is strict level-2 with its only blob being type-2 and x being under the sink, such that the restriction $\mathcal{N}|_{\mathcal{Y}\setminus\{x\}}$ is not simple.

Proof. As a simple network on at least six leaves cannot be level-0, we only look at strict level-1 and strict level-2 simple networks.

1. Suppose $\mathcal{N}|_\mathcal{Y}$ is level-1. As $\mathcal{N}|_\mathcal{Y}$ is simple, there is only one leaf x below the reticulation node. As $\mathcal{N}|_\mathcal{Y}$ is simple level-1, for any $a,b,c \in \mathcal{Y}$, the quarnet $\mathcal{N}|_{\{x,a,b,c\}}$ is a four-cycle, thus there is no $q \in \mathcal{Q}^s_\mathcal{Y}(\mathcal{N})$ such that $x \in L(q)$. For any $a,b,c,d \in \mathcal{Y}\setminus\{x\}$, as none of the leaves is below a reticulation node, there exists a $q \in \mathcal{Q}^s_\mathcal{Y}(\mathcal{N})$ such that $a,b,c,d \in L(q)$.

2. Suppose $\mathcal{N}|_\mathcal{Y}$ is strict level-2 with its only blob being type-1, and with x,y being the two leaves below two separate reticulation nodes, such that the restriction $\mathcal{N}|_{\mathcal{Y}\setminus\{x\}}$ is simple and the restriction $\mathcal{N}|_{\mathcal{Y}\setminus\{y\}}$ is not simple. Let $a,b \in \mathcal{Y}\setminus\{x,y\}$. Then, the quarnet $\mathcal{N}|_{\{x,y,a,b\}}$ is simple level-2, thus it has no non-trivial split. Let $a,b,c,d \in \mathcal{Y}\setminus\{x,y\}$. As none of the leaves is below a reticulation node, there exists a $q \in \mathcal{Q}^s_\mathcal{Y}(\mathcal{N})$ such that $a,b,c,d \in L(q)$. As $\mathcal{Y}\setminus\{y\}$ is not simple, for every $a \in \mathcal{Y}\setminus\{y\}$, there exists a $q \in \mathcal{Q}^s_\mathcal{Y}(\mathcal{N})$ such that

$a \in L(q)$. As $\mathcal{N}|_{\mathcal{Y}\setminus\{x\}}$ is simple level-1, there must exist one leaf that is not in any quarnet-split. As we have shown that every leaf $a \in \mathcal{Y}\setminus\{y\}$ is in a quarnet-split, this missing leaf has to be y. Therefore, y is the only leaf in $\mathcal{Y}$, such that there exists no $q \in \mathcal{Q}^s_{\mathcal{Y}}(\mathcal{N})$ such that $y \in L(q)$.
If both $\mathcal{N}|_{\mathcal{Y}\setminus\{x\}}$ and $\mathcal{N}|_{\mathcal{Y}\setminus\{y\}}$ were simple, then in both restrictions, there must be a missing leaf. But since all $a \in \mathcal{Y}\setminus\{x,y\}$ are not a missing leaf, both x and y should have that there is no $q_1, q_2 \in \mathcal{Q}^s_{\mathcal{Y}}(\mathcal{N})$ such that $x, y \in L(q_1) \cup L(q_2)$.
If both restrictions were non-simple, then in both restrictions there is no missing leaf. Therefore, $\mathcal{N}|_{\mathcal{Y}}$ would have no missing leaves.

3. Suppose $\mathcal{N}|_{\mathcal{Y}}$ is strict level-2 with its only blob being type-2, and with x being the leaf under the sink, such that the restriction $\mathcal{N}|_{\mathcal{Y}\setminus\{x\}}$ is not simple. As $\mathcal{N}|_{\mathcal{Y}\setminus\{x\}}$ is not simple, for all leaves $a \in \mathcal{Y}\setminus\{x\}$ there exists a $q \in \mathcal{Q}^s_{\mathcal{Y}}(\mathcal{N})$ such that $a \in L(q)$. Let $a, b, c \in \mathcal{Y}\setminus\{x\}$. As the quarnet $\mathcal{N}|_{\{x,a,b,c\}}$ contains all reticulation nodes in $\mathcal{N}|_{\mathcal{Y}}$, it is simple strict level-2, therefore it does not have a non-trivial split.
If the restriction $\mathcal{N}|_{\mathcal{Y}\setminus\{x\}}$ was simple level-1, there is a $y \in \mathcal{Y}\setminus\{x\}$, such that there is no $q \in \mathcal{Q}^s_{\mathcal{Y}}(\mathcal{N})$ such that $y \in L(q)$. As x also has no such quarnet-split, there would be at least two missing leaves.

□

Now we first define the canonical form of a level-2 type-2 blob with one missing leaf. In almost the same way as in the previous subsection about zero missing leaves, we find the cycle contraction edges by first neglecting the missing leaf, then in the resulting blob, the edges that are in the cycle and are incident to a reticulation edge are called cycle contraction edges. If this blob is an m-blob with $m < 5$, this is called a small cycle. With the missing leaf still neglected, the edges adjacent to a leaf and to a degree-2 internal node are called contraction edges. A reticulation edge adjacent to two reticulation nodes is also called a contraction edge.

For a level-2 blob of type-1, we do the exact same process to obtain the cycle contraction edges, the contraction edges, and the small cycles. Then the first version of the canonical form is again obtained by contracting all these edges and cycles. For this type of blob, the same cases of Definition 11.(5,6,7) can be obtained. The two additional cases (8,9) can also be found. Examples and visualisations can be found in Figure 5.5.

Definition 11. Given a semi-directed level-2 network $\mathcal{N}$ on $\mathcal{X}$, and $\mathcal{Y} \subset \mathcal{X}$ such that $\mathcal{N}_{\mathcal{Y}}$ is simple and $|\mathcal{Y}| \geq 6$. If $\mathcal{Q}^s_{\mathcal{Y}}(\mathcal{N})$ has one missing leaf, then the canonical form of this blob is obtained by contracting all cycle contraction edges, contraction edges, and small cycles, followed by one of the steps below corresponding to the result.

(1) A network with one reticulation node with three incoming reticulation edges. If there is no node with an outgoing reticulation edge and less than two adjacent leaves, we leave this as is, as this is the final canonical form of this case. If there exists a node with an outgoing reticulation edge and less than two adjacent leaves, we remove this reticulation edge. If these exist, then all degree-2 internal nodes dissolve. This is the final canonical form of this case. (This results in a level-1 blob)

(2) A level-2 blob, where if we leave out the missing leave ($\mathcal{N}_{\mathcal{X}\setminus\{x\}}$), multiple leaves are under the reticulation node and a possible pendant subtree. (a) If this pendant subtree exists, we leave this as is, as this is the final canonical form of this case. (b) If no such pendant subtree exists, we remove the reticulation edge outgoing from the blob and ingoing to the reticulation node directly above the missing leaf. Then we add a reticulation edge outgoing from the reticulation node inside the blob ingoing to the reticulation node directly above the missing leaf. This is the final canonical form.

(3) A level-2 blob, where if we leave out the missing leaf, there is one leaf under the reticulation node and where the blob has a pendant subtree, such that there exist two internal non-reticulation nodes of degree-4, such that both nodes have two adjacent leaves. This is the final canonical form.

(4) A level-2 blob, where if we leave out the missing leaf, there is one leaf under the reticulation node and where the blob has a pendant subtree, such that there exists only one internal node in the blob of degree-4 with two adjacent leaves. In addition to this, we add a reticulation edge going from the internal node with degree-4 and two adjacent leaves, to the reticulation node directly above the missing leaf. This is now the final canonical form of this case.

(5) A level-1 blob with no pendant subtrees. This is the final canonical form.

(6) A level-1 blob with one reticulation node and one internal node with three adjacent leaves and an outgoing reticulation node. This is the final canonical form.

(7) A level-1 blob with one reticulation node and one internal node with two adjacent leaves and no outgoing reticulation edges. This is the final canonical form.

(8) A level-2 blob where, if we leave out the missing leaf, we get a level-1 blob with one leaf under the reticulation node and where the blob has a pendant subtree, such that there exists only one internal node in the blob of degree-4 with two adjacent leaves. In addition to this, we add a reticulation node from the reticulation node inside the level-1 blob to the reticulation node directly above the missing leaf. This is the final canonical form.

(9) A level-2 blob where, if we leave out the missing leaf, we get a level-1 blob with one leaf under the reticulation node and where the blob has two pendant subtrees. This is the final canonical form.

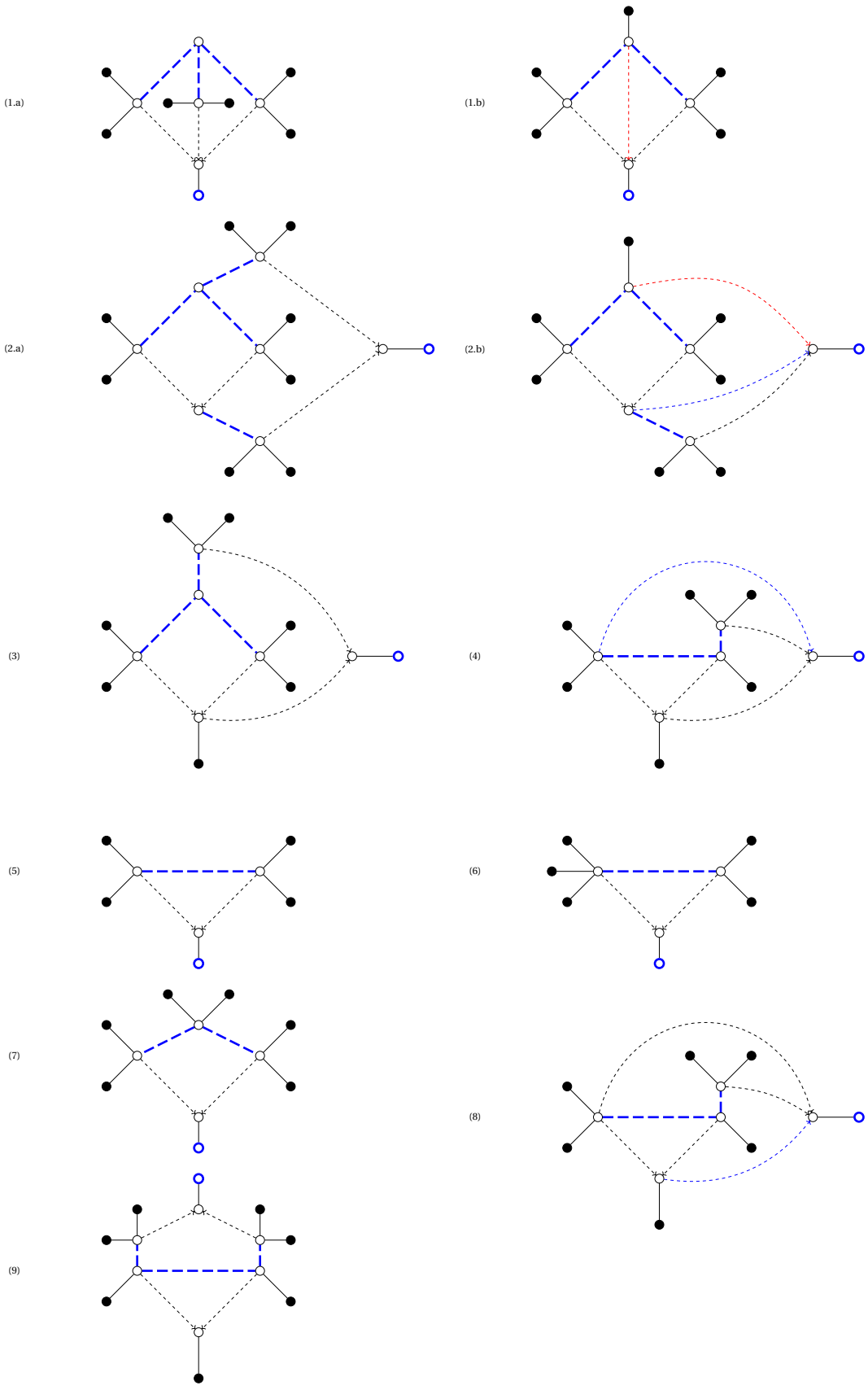


Figure 5.5: The cases of canonical forms of type-2 level-2 blobs as described in Definition 11 after contracting cycle contraction edges, contraction edges and small cycles. The red reticulation edges get removed, the blue reticulation edges are added to obtain the final canonical form. The blue regular edges indicate that an arbitrary number of leaves can be attached to this edge. The missing leaf is visualised as the blue node.

Note that the canonical form of a level-1 blob is the same as the canonical form of a level-2 blob in the case of (1.b) and (5). Therefore, when we want to obtain the final form of the network using the canonical form, we have to take into account that level-1 blobs can be turned into level-2 blobs. This principle also holds for level-2 blobs in the forms (4), (6), (7) and (8), where the canonical form can be turned into a type-1 or type-2 level-2 blob. For the forms (1.a), (2), (3) and (9), the level and the type of the blob result from the type of the canonical form ((1.a), (2) and (3) always result in a type-2 level-2 blob, (9) always results in a type-1 level-2 blob).

Lemma 7. Let $\mathcal{N}$ be a semi-directed phylogenetic network on $\mathcal{X}$ with its corresponding quarnet-splits $\mathcal{Q}^s(\mathcal{N})$. Let $\mathcal{Y} \subset \mathcal{X}$ be such that $\mathcal{N}|_\mathcal{Y}$ has one internal level-1 or level-2 blob and $|\mathcal{Y}| \geq 6$, with its corresponding quarnet-splits $\mathcal{Q}^s_\mathcal{Y}$. If there does not exist a $q \in \mathcal{Q}^s_\mathcal{Y}$ such that $x \in L(q)$, then x is under a reticulation node, and also the only leaf directly under this reticulation node.

Proof. Let a be above all reticulation nodes in $\mathcal{N}|_\mathcal{Y}$. If b, c, d are also above all reticulation nodes, there must be a $q \in \mathcal{Q}^s_\mathcal{Y}(\mathcal{N})$ such that $a \in L(q)$. If a is the only leaf above all reticulation nodes, then, as $k \geq 6$, there must be one reticulation node with at least three leaves $x1, x2, x3$ under it. Then, the restriction $\mathcal{N}|_{\{a,x1,x2,x3\}}$ cannot be a four-cycle, thus there must be a $q \in \mathcal{Q}^s_\mathcal{Y}(\mathcal{N})$ such that $a \in L(q)$.
Let a be under a reticulation node. Let b be directly under this same reticulation node. Note that this is only possible if $\mathcal{N}|_\mathcal{Y}$ is a simple level-2 network with a blob of type-2. As parallel edges cannot exist, there must be at least one leaf c above this reticulation node. As $k \geq 6$, there must be a leaf either d above or below this same reticulation node. Then, the restriction $\mathcal{N}|_{\{a,b,c,d\}}$ cannot be a four-cycle, thus there must be a $q \in \mathcal{Q}^s_\mathcal{Y}(\mathcal{N})$ such that $a \in L(q)$. □

Construction 2. Let x be the leaf under the reticulation node found. Then, the restriction $\mathcal{N}_{\mathcal{Y}\setminus\{x\}}$ is a phylogenetic network of level-1 or level-0. As we know how to construct a level-1 and level-0 network, we run the algorithm on the set of leaves $\mathcal{Y}\setminus\{x\}$.

1. If this construction results in a level-0 network with one internal node with three adjacent internal nodes, then there exist three internal nodes with one adjacent internal node. We add three reticulation edges, each outgoing from one of these three internal nodes, directed to the reticulation node directly above the missing leaf. (1.a)

2. If this construction results in a level-0 network with no internal nodes with three internal nodes, thus a tree (not binary, as nodes with 3 adjacent leaves can occur), its the same process as constructing a level-1 blob; thus, we attach the reticulation node with its corresponding reticulation leaf to the end internal nodes of this tree (internal nodes with one adjacent internal node). ((1.b),(5),(6),(7))

3. If this construction results in a level-1 network with multiple leaves under the reticulation node, then there exists at least one internal node with two adjacent leaves and no outgoing reticulation edges. (a) If there exist two, we add two reticulation edges outgoing from these nodes to the reticulation node directly above the missing leaf. (b) If there exists one, we add one reticulation edge outgoing from this node and one reticulation edge outgoing from the reticulation node in the found level-1 network, both directed to the reticulation node directly above the missing leaf. ((2.a),(2.b))

4. If this construction results in a level-1 network with one leaf under the reticulation node, if the level-1 blob has a pendant subtree and two internal nodes of degree-4 with two adjacent leaves and both an outgoing reticulation edge, then we add two reticulation edges, one outgoing from the end of the pendant subtree (the internal node of degree-3 with one adjacent internal node and no outgoing or incoming reticulation edges) and one outgoing from the reticulation node in the level-1 blob, both directed to the reticulation node directly above the missing leaf. (3)

5. If this construction results in a level-1 network with one leaf under the reticulation node, if the level-1 blob has a pendant subtree and only one internal node u of degree-4 with two adjacent leaves, we add three reticulation edges. One outgoing from the reticulation node in the level-1 blob, one outgoing from u, and one outgoing from the end of the pendant subtree (the internal node of degree-3 with one adjacent internal node and no outgoing or incoming reticulation edges), all three directed to the reticulation node directly above the missing leaf. ((4),(8))

6. If this construction results in a level-1 network with one leaf under the reticulation node, if the level-1 blob has two pendant subtrees, we add two reticulation node. Both outgoing from both ends of the pendant subtrees (the internal nodes of degree-3 with one adjacent internal node and no outgoing or incoming reticulation edges), both directed to the reticulation node directly above the missing leaf. (9)

5.4.3. Two missing leaves

In this subsection, we look at the last case, where the number of missing leaves is two. In this situation, we know that the blob must be a type-1 or type-2 level-2 blob. A type-1 m-blob of level-2 with $m \geq 6$ has two missing leaves if for $i \in \{1,2,3,4\}, |S_i| \leq 1$ (with taking Figure 5.1 as reference). A type-2 m-blob of level-2 with $m \geq 6$ has two missing leaves if $|S_3| = 1$ and if $|S_4| \leq 1$.

Lemma 8. Let $\mathcal{N}$ be a semi-directed phylogenetic network on $\mathcal{X}$ with its corresponding quarnet-splits $\mathcal{Q}^s(\mathcal{N})$. Let $\mathcal{Y} \subset \mathcal{X}$ be such that $\mathcal{N}|_\mathcal{Y}$ is simple and $|\mathcal{Y}| \geq 6$, with its corresponding quarnet-splits $\mathcal{Q}^s_\mathcal{Y}(\mathcal{N})$. Then, the number of missing leaves in $\mathcal{Q}^s_\mathcal{Y}(\mathcal{N})$ is two, if and only if either

1. the blob in $\mathcal{N}|_\mathcal{Y}$ is level-2 of type-1 and, using Figure 5.1, $|S_1|, |S_2|, |S_3|, |S_4| \leq 2$.
2. the blob in $\mathcal{N}|_\mathcal{Y}$ is level-2 of type-2 and, using Figure 5.1, $|S_3| = 1, |S_4| \leq 1$.

Proof. (1) Let x, y be under the reticulation nodes of a level-2 type-1 blob. If $|S_1|, |S_2|, |S_3|, |S_4| \leq 2$, both the restrictions $\mathcal{N}|_{\mathcal{Y}\setminus\{x\}}$ and $\mathcal{N}|_{\mathcal{Y}\setminus\{y\}}$ result in a simple level-1 network. As the reticulation leaf in a simple level-1 network is always a missing leaf, x is not in any of the quarnet-splits in $\mathcal{Q}^s_{\mathcal{Y}\setminus\{y\}}(\mathcal{N})$. This mirrors for y. Let $a, b \in \mathcal{Y}\setminus\{x, y\}$. Then, the quarnet $\mathcal{N}|_{\{x,y,a,b\}}$ is always a simple level-2 network, thus it has no quarnet-splits. Therefore, both x and y are not in any quarnet-splits.
(2) Let x, y be under the reticulation nodes of a level-2 type-2 blob. The only way this can exist is for one (x) to be on the side S_3, and one (y) to be under the sink of the blob. As $\mathcal{N}|_{\mathcal{Y}\setminus\{x,y\}}$ is level-0, any other leaf is in a quarnet-split. Let $a, b, c \in \mathcal{Y}\setminus\{x, y\}$. As any quarnet $\mathcal{N}|_{\{y,a,b,c\}}$ is a simple level-2 network, y is not in any of the quarnet-splits. As $|S_4| \leq 1$, any quarnet $\mathcal{N}|_{\{x,a,b,c\}}$ is a four-cycle, thus there is no quarnet-split with x in it.

If neither 1. and 2. would be the case, the network would fall under the case of no missing leaves or one missing leaf, what is given by the Lemmas 4 and 6. □

Definition 12. Let $\mathcal{N}|_\mathcal{Y}$ be simple, with $|\mathcal{Y}| \geq 6$ and with $\mathcal{Q}^s_\mathcal{Y}(\mathcal{N})$ having two missing leaves. Then, the canonical form is obtained as follows. Let x, y be these missing leaves. Take the restriction $\mathcal{N}|_{\mathcal{Y}\setminus\{x,y\}}$. In this restriction, there are exactly two internal nodes u, v with two adjacent leaves and one adjacent internal node. Then, we add two reticulation nodes outgoing from u and v directed to a reticulation node directly above x. We do the same with y. Finally, we add a regular edge between the two reticulation nodes.

See Figure 5.6 for visualisation below.

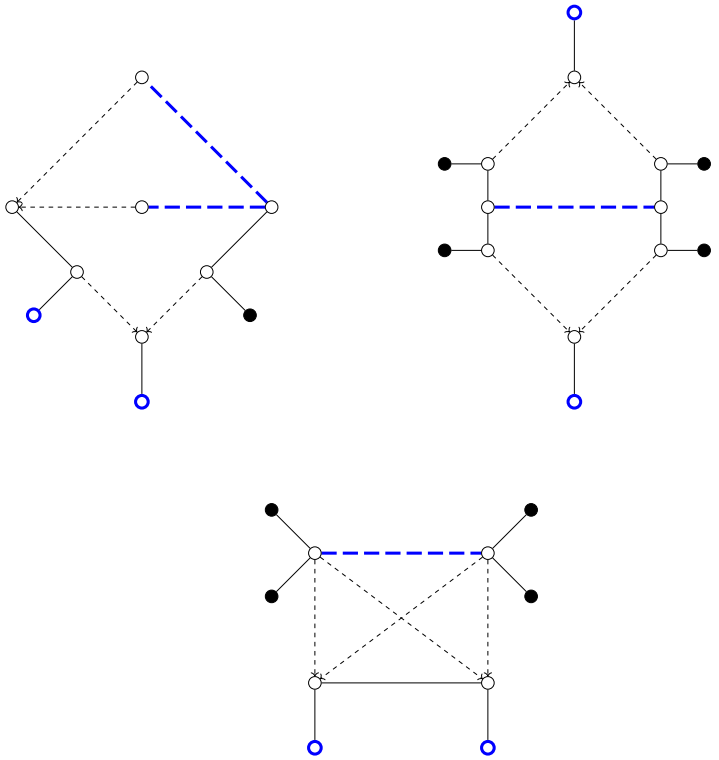


Figure 5.6: The two types of level-2 simple networks with two missing leaves on top, with their canonical form at the bottom. The missing leaves are indicated as blue nodes. On each blue thick dashed edge, an arbitrary number of leaves is attached. The black leaves on top are optional, if there are at least four leaves in the simple network that are not the missing leaves. The black leaves in the canonical form are always present in the way its visualised.

Construction 3. The construction of a level-2 blob with two missing leaves into its canonical form is quite similar to the previous constructions. First, we run the level-1 algorithm on the set of leaves $\mathcal{Y}\backslash\{x,y\}$ (where x and y are the missing leaves). If we observe the shape of the network $\mathcal{N}|_{\mathcal{Y}}$, we can see that only x and y are under a reticulation node, therefore the algorithm on $\mathcal{Y}\backslash\{x,y\}$ gives us a tree $\mathcal{T}$. As the missing leaves are not in any of the quarnet-splits $\mathcal{Q}^s_{\mathcal{Y}}(\mathcal{N})$ by definition, we find that the weak stem path includes all edges in $\mathcal{T}$ that are not adjacent to a leaf. Then, we attach two reticulation nodes to both ends of this weak stem path with two reticulation edges, and attach x and y to these reticulation nodes. Finally, we add an edge between the two reticulation nodes, indicating that using merely quarnet-splits, we do not know the type of the blob yet. This results in the canonical form that can be seen in Figure 5.6.

Algorithm 2 Algorithm for inflating a blob-tree of a level-2 phylogenetic network into the canonical form of this network, using its corresponding quarnet-splits.

Data: quarnet-splits $\mathcal{Q}^s(\mathcal{N})$ of a semi-directed level-2 network $\mathcal{N}$ on $\mathcal{X}$; the blob-tree $\mathcal{T}(\mathcal{N})$
Result: The canonical form $\mathcal{N}^c$ of a semi-directed phylogenetic network $\mathcal{N}$

Let G be $\mathcal{T}(\mathcal{N})$;
foreach *internal node u in G* **do**
  Isolate the node by taking a random leaf from each pendant subtree of u, resulting in the leaf-set $\mathcal{Y}$;
  if $|\mathcal{Y}| < 6$ **then**
    continue
  Determine the number of missing leaves nML from $\mathcal{Y}$ in $\mathcal{Q}_{\mathcal{Y}}$;
  if $nML == 0$ **then**
    Find the reticulation leaves r_1, r_2 using Lemma 5;
    Run Algorithm 1 on $\mathcal{Y}\backslash\{r_1, r_2\}$, resulting in $\mathcal{C}$;
    Determine the stems s_1 and s_2 for both r_1 and r_2 respectively in $\mathcal{C}$;
    Use Construction 1.(1-3), corresponding to the types of the stems s_1 and s_2;
  if $nML == 1$ **then**
    Let r_1 be the missing leaf;
    Run Algorithm 1 on $\mathcal{Y}\backslash\{r_1\}$, resulting in $\mathcal{C}$;
    Use Construction 2.(1-6), corresponding to the shape of $\mathcal{C}$;
  if $nML == 2$ **then**
    Let r_1 and r_2 be the missing leaves;
    Run Algorithm 1 on $\mathcal{Y}\backslash\{r_1, r_2\}$, resulting in $\mathcal{C}$;
    Use Construction 3;
  Add $\mathcal{C}$ back to G by replacing u with $\mathcal{C}$ and substituting each leaf of $\mathcal{C}$ with its corresponding pendant subtree/subgraph;
return G

Theorem (Theorem 1)**.** Given the quarnet-splits of a semi-directed level-2 network $\mathcal{N}$ on $\mathcal{X} = \{x_1, x_2, ..., x_n\}$, there exists an algorithm that constructs the canonical form of $\mathcal{N}$.

Proof. The algorithm begins with a blob-tree $\mathcal{T}(\mathcal{N})$ of $\mathcal{N}$, which is proven to be constructable using quarnet-splits for any level-k network. Then, we prove that every internal node can be inflated individually into its canonical form corresponding to its adjacent pendant trees. As no degree-5 nodes can be inflated, we skip these nodes. Let u be an internal degree-k node with $k \geq 6$. Then, u has k pendant subtrees. For each of this subtrees, we can find one random leaf. Thus we can isolate u to be one internal node with k adjacent leaves in the subset $\mathcal{Y}$. Then, using the definition of $\mathcal{Q}^s_{\mathcal{Y}}(\mathcal{N})$, we can find the quarnet-splits corresponding to the leaves $\mathcal{Y}$. Using $\mathcal{Q}^s_{\mathcal{Y}}(\mathcal{N})$, we can find the number of missing leaves. Using Lemma 4, Lemma 6 and Lemma 8, we can find the structure of the blob u will be inflated to based on the number of missing leaves.
If this number is zero, we can find the reticulation leaves using Lemma 5, and construct the canonical form of the simple network based on this information using Construction 1.
If this number is one, using Lemma 7, we can find the sink of the simple network. Removing this results in a level-1 network that can be constructed. Then, using Construction 2, we can find the canonical form of the simple network.
If this number is two, we know that removing the missing leaves from the network results in a level-0 network, that can be constructed. Then, using Construction 3, we can find the canonical form of the simple network.
Then, as inflating a singular node in a network to a blob does not affect any other nodes, we can glue the obtained simple network back to the original network, resulting in the canonical form of the network after inflating all degree-k nodes with $k \geq 6$. □

6

Addition of Quartets

In this final chapter, the focus is on obtaining the complete network $\mathcal{N}$ using its canonical form $\mathcal{N}^c$ and additional information from its quarnets $\mathcal{Q}(\mathcal{N})$. Here, we get the additional information of the quartet-trees of the quarnets of the network, which are generally harder to obtain then quarnet-splits.

6.1. Finalising level-1 networks

In the previous chapters, we only used quarnet-splits as an information source to build a network. However, as using merely quarnet-splits gives us only a canonical form of the network, we have to expand our source of information to achieve the complete binary semi-directed phylogenetic network. Therefore, we have to introduce a new term. As stated previously, there are multiple forms of level-1 quarnets. For now, the only information we use from the quarnets is if it has a non-trivial cut-edge, thus a split, or if the quarnet is a four-cycle, and if so, what is the reticulation node of this four-cycle. Thus, information on triangles is left out, so we contract all triangles in the quarnets, resulting in only quartet-trees and four-cycles.
Let q be a quarnet on $\{a,b,c,d\}$. If q is a quartet-tree, we denote q as its split, e.g., $ab|cd$. If q is a four-cycle, we denote q as a tuple (a,b,c,d) with respect to its circular order, and with the first term being under the reticulation node. Every four-cycle (a,b,c,d) can be split up into the two *quartets* $ab|cd$ and $ad|cb$, which will be the list of quartets.

6.1.1. Quartets

Suppose that we only have the quartet information of the four-cycles, that is, we know the circular order, but the reticulation is unknown. What is the maximal version of the level-1 phylogenetic network we can achieve? Let $\mathcal{N}$ be a simple level-1 phylogenetic network on $\mathcal{X}$ with $|\mathcal{X}| \geq 5$. Let $\mathcal{Q}^t(\mathcal{N})$ be the quartets of the network, and $\mathcal{N}^c$ the canonical form found with the quarnet-splits $\mathcal{Q}^s(\mathcal{N})$. As $|\mathcal{X}| \geq 5$, we know which leaf is under the reticulation node. Let this leaf be a. Removing a and the reticulation node above will result in a tree on the leaves $\mathcal{X}\backslash\{a\}$, which is the same as $\mathcal{N}^c||_{\mathcal{X}\backslash\{a\}}$. Using Lemma 1 with now using $\mathcal{Q}^t(\mathcal{N})$ instead of $\mathcal{Q}^s(\mathcal{N})$ (note that $\mathcal{Q}^s(\mathcal{N}) \subseteq \mathcal{Q}^t(\mathcal{N})$), and definition 6, we will find two strong stem edges for a that are adjacent to a leaf. We call these a *strong stem edge-pair*.

Lemma 9. Let $\mathcal{N}$ be a level-1 semi-directed network on $\mathcal{X}$ and let $\mathcal{Y} \subseteq \mathcal{X}$ be such that $\mathcal{N}|_{\mathcal{Y}}$ is simple and $|\mathcal{Y}| \geq 5$. Given is $\mathcal{N}^c||_{\mathcal{Y}}$ and $\mathcal{Q}^t(\mathcal{N})$. Let $a \in \mathcal{Y}$ be the leaf under the reticulation node in $\mathcal{N}^c||_{\mathcal{Y}}$. Then, $\mathcal{N}^c||_{\mathcal{Y}\backslash\{a\}} = \mathcal{N}|_{\mathcal{Y}\backslash\{a\}}$. Then $\mathcal{N}|_{\mathcal{Y}}$ is obtained by attaching two reticulation edges to the strong stem edge-pair for a in $\mathcal{N}|_{\mathcal{Y}\backslash\{a\}}$, directed to a reticulation node directly above a.

Proof. Let $\mathcal{N}$ be a strict level-1 semi-directed network on $\mathcal{X}$ and let $\mathcal{Y} \subseteq \mathcal{X}$ be such that $\mathcal{N}|_{\mathcal{Y}}$ is simple. Given is $\mathcal{N}^c||_{\mathcal{Y}}$ and $\mathcal{Q}^t(\mathcal{N})$. Let $a \in \mathcal{Y}$ be the leaf under the reticulation node in $\mathcal{N}^c||_{\mathcal{Y}}$. First, we want to show that $\mathcal{N}^c||_{\mathcal{Y}\backslash\{a\}} = \mathcal{N}|_{\mathcal{Y}\backslash\{a\}}$.
As $\mathcal{N}^c||_{\mathcal{Y}}$ is a simple level-1 network, there exists only one reticulation node with one leaf under it (if there would be more leaves under it, the network would not be simple). Thus, $\mathcal{N}^c||_{\mathcal{Y}\backslash\{a\}}$ is a network without reticulation nodes, thus level-0. As by definition, the level-0 network $\mathcal{N}^c||_{\mathcal{Y}\backslash\{a\}}$ is equal to its blob-tree, as there are no blobs to contract to a single vertex. As the canonical-form of a network is, in the case of this paper, built up from its blob-tree by inflating internal nodes, we can say that $\mathcal{T}(\mathcal{N}) = \mathcal{T}(\mathcal{N}^c)$. Thus, $\mathcal{T}(\mathcal{N}^c||_{\mathcal{Y}\backslash\{a\}}) =$

$\mathcal{T}(\mathcal{N}|_{\mathcal{Y}\setminus\{a\}})$. Since $\mathcal{N}|_{\mathcal{Y}\setminus\{a\}}$ is level-0, thus equal to its blob-tree, $\mathcal{N}^c||_{\mathcal{Y}\setminus\{a\}} = \mathcal{N}|_{\mathcal{Y}\setminus\{a\}}$.
Let $r1, r2$ be the two reticulation edges in $\mathcal{N}|_{\mathcal{Y}}$. If we remove a from $\mathcal{N}|_{\mathcal{Y}}$ and subsequently suppress all degree-2 nodes, there are two cut-edges $e1, e2$ in $\mathcal{N}|_{\mathcal{Y}\setminus\{a\}}$ as a result of this suppression. Both these edges are adjacent to a leaf, let these be $b1, b2$. Since the rest of the network stays the same, we can get $\mathcal{N}|_{\mathcal{Y}}$ from $\mathcal{N}|_{\mathcal{Y}\setminus\{a\}}$ by attaching two reticulation nodes from the edges $e'1, e'2$ which induce the same splits $x1|A$ and $x2|B$ as $e1$ and $e2$, to the reticulation node directly above a. Then, $\{x1, a\}|A$ and $\{x2, a\}|B$ are splits (the other way are trivial), thus $e'1$ and $e'2$ are a strong edge pair. □

By using lemma 9, we can expand all m-blobs with $m \geq 5$ of the canonical form of a network to its complete version. For inflating 3 and 4-blobs, we need the full quarnets as information, that is, knowledge of the reticulation node and the triangles in the quarnet. If we know this, the inflation of 3 and 4-blobs is easy to do. However, triangles and reticulation nodes in quarnets are harder to find in practice by previous studies. [1, 4] Therefore, we want to minimise the amount of information to find the maximal form possible of a phylogenetic network.

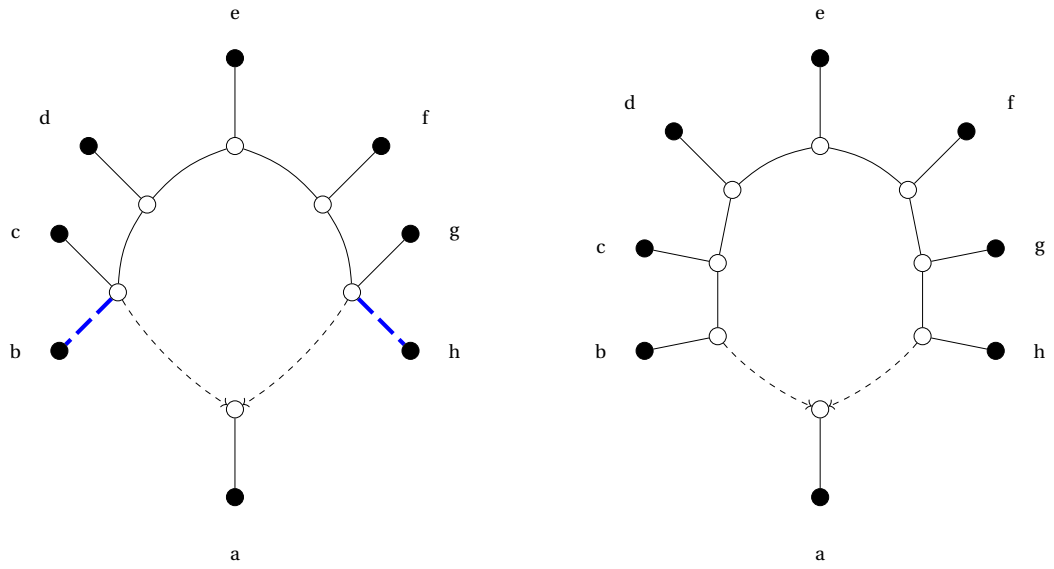


Figure 6.1: An example of using quarnets to refine a level-1 simple network. On the left, the canonical form of the simple network is illustrated, and on the right the results after refining. The blue dashed edges indicate the strong leaf edges.

6.2. Level-2 networks from quartets

In a similar way, we can upgrade strict level-2 blobs to a closer to its complete form using quartets. However, level-2 quarnets have additional possible structures that can occur. These are the quarnets generated from the level-2 generators, such that there are no parallel edges and no reticulation nodes without leaves under it. This also leads to different quartets from the strict level-2 quarnets. Each level-2 quarnet has either two or three quartets. We can find these by removing a random reticulation edge from each reticulation edge-pair. As there are always four reticulation edges, thus two pairs, there are $2^2 = 4$ combinations that can occur. By removing these edges, the quarnet turns into a level-0 quarnet, thus there must exist a non-trivial split in each combination. Dependent on the level-2 quarnet, this can result in two or three unique quartets.
As we still cannot determine the reticulation node of a quarnet using only information from the quartets, 3 and 4-blobs cannot be inflated with only quartet information. For that, we need to know the full quarnets. This is crucial for the completion of level-2 blobs, as these blobs can contain four-cycles and triangles, which will become clearer later in this chapter. Therefore, with only quartets as additional information, we still cannot find the complete desired binary form of every phylogenetic network. But we will show what blobs can be fully found, and what the maximal form is of non-complete blobs.

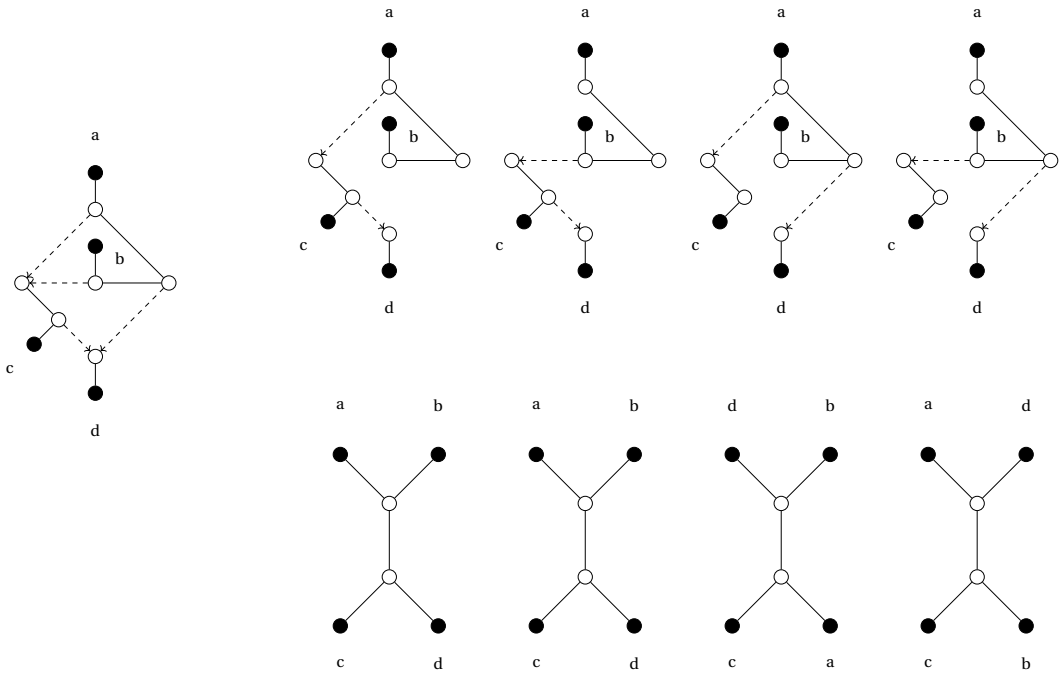


Figure 6.2: An example of a level-2 quarnet with its corresponding quartets. Here, this level-2 quarnet has all possible combinations of quartets $ab|cd$, $ac|bd$ and $ad|bc$.

6.2.1. Canonical form

In the same way as for level-1 networks, a complete level-2 network cannot always be found without using quarnets. Therefore, we have to define a canonical form $\mathcal{N}_t^c$ that can be found by using the quartets of a network $\mathcal{N}$.
In a level-2 network, a reticulation edge outgoing from a reticulation node, ingoing to a reticulation node, is called a contraction edge. A 3 or 4-cycle is called a *small cycle* if this cycle is a blob, or if this cycle does not include the sink of a level-2 blob.

Definition 13. Let $\mathcal{N}$ be a level-2 network on $\mathcal{X}$. Then, the canonical form $\mathcal{N}_t^c$ is obtained by contracting all contraction edges and all small cycles.

Theorem 2. Given the quartets $\mathcal{Q}^t(\mathcal{N})$ of a semi-directed level-2 network $\mathcal{N}$ on $\mathcal{X} = \{x_1, x_2, ..., x_3\}$, there exists an algorithm that constructs the canonical form $\mathcal{N}_t^c$.

6.2.2. No missing leaves

As quarnet-splits are still available, we still have knowledge about the number of missing leaves of a blob. Now we want to complete the canonical forms that we get from a level-2 blob with no missing leaves. These canonical forms are shown in figure 5.3. Here we note the leaves under the reticulation nodes as x and y.

Construction 4. 1. (a) If there is at least one overlapping edge from both weak stem paths for both reticulation leaves found with the quarnet-splits, then, when removing both reticulation leaves from the canonical form, we can determine a possible strong stem edge-pair for both reticulation leaves using quartet-splits. (a.1) Suppose such a pair exists for both leafs. If the strong edge-pairs are independent, we use lemma 9 to attach both reticulation leaves back. If both pairs share a common edge uv, we do the following. We first determine if the split $\{x, y\}|\mathcal{Y}\backslash\{x, y\}$ exists using the quartets. If so, we attach a new node w to uv, and attach the reticulation leaves to w and to the other corresponding edge in the strong edge-pairs that is not uv. If this does not exist, we attach both reticulation leaves back regularly using lemma 9, which will result in a degree-4 node in the network, thus the network would not be complete without full quarnet information.
(a.2) Suppose only one leaf x has a strong edge-pair. Then, we relocate the reticulation edges of this leaf to their corresponding strong edges. For the other leaf y, this leaf has at least one strong edge adjacent to a leaf. We relocate the reticulation edge adjacent to this strong edge to this same edge. The other

reticulation node begins from a node u adjacent an internal node v that is not in the weak stem path of y. Then, we relocate this reticulation edge to the edge uv. If both leaves have no strong edge-pair, we do the same process for both reticulation edge-pairs as the previous process corresponding to the leaf with no strong edge-pair. (b) If there is no overlapping edge, but only one overlapping node u with its only neighbouring leaf v, we remove the leaves x and y, determine their strong edge-pairs and attach them using lemma 9 to their corresponding edge pairs. Suppose that one of the edges in both strong edge-pairs is uv. We determine whether the split $\{x,y\}|\mathcal{Y}\backslash\{x,y\}$ exists using the quartets. If this indeed exists, then we extend the new node on the edge uv. If this does not exist, we leave it as is. (For a more accurate binary version, full quarnets are necessary).
(c) If there does not exist an overlapping edge and node, we do the following. First, we determine if the split $\{x,y\}|\mathcal{Y}\backslash\{x,y\}$ exists using quartets, and we determine a strong edge adjacent to leaf for both x,y. If the split exists, we do the following. There is a node u with two outgoing reticulation edges. Then, we relocate these reticulation edges to a new node v that we attach to u with the edge uv. The other reticulation are relocated to the corresponding adjacent strong edge. If the split does not exist, we do the following. There is a node u with two outgoing reticulation edges and two other adjacent edge ua and ub. Then we subdivide ua and ub, resulting in new nodes v and w. Then both reticulation edges adjacent to u are replaced to either v or w separately, in a way there will not exist any non-trivial cut-edge in the network. The other reticulation edges are relocated to their corresponding adjacent strong leaf-edge. Then, as u will remain as a degree-2 node, we can dissolve this node.

2. If both leaves have a stem vertex, we do the same process as in (1.c).

3. Suppose x has a stem vertex and y a weak stem path. (a) If by removal of the reticulation leaves the simple network results in a tree with no internal node with three adjacent internal nodes, we do the following. Let u be the node with two outgoing reticulation edges before this removal and v the stem vertex of x. First, we determine the strong edge with an adjacent leaf for both x,y. Then, we determine if the split $\{x,y\}|\mathcal{Y}\backslash\{x,y\}$ exists using quartets. If so, then we attach a new node w to uv, and relocate the reticulation nodes from u to w. The other reticulation edges are relocated to their corresponding adjacent strong edge. If the split does not exist, we subdivide the edge uv twice, such that the path $P=(u,p_1,p_2,v)$ exists. Let r_x,r_y be the reticulation edges outgoing from u corresponding to their reticulation leaves. Then, we relocate r_x to p_1 and r_y to p_2. The other reticulation edges are relocated to their corresponding adjacent strong leaf-edge.
(b) If by removal of the reticulation node the simple network results in a tree with one internal node u with three adjacent internal nodes, we do the following. Then, for y, we determine a strong edge-pair. For x, when determine the strong edges, there is one strong leaf-edge adjacent to a leaf and one strong edge on the weak stem path for y adjacent to u. We relocate the reticulation edge for x outgoing from u to this adjacent strong edge. The other reticulation edges are relocated to their corresponding adjacent strong leaf-edge.

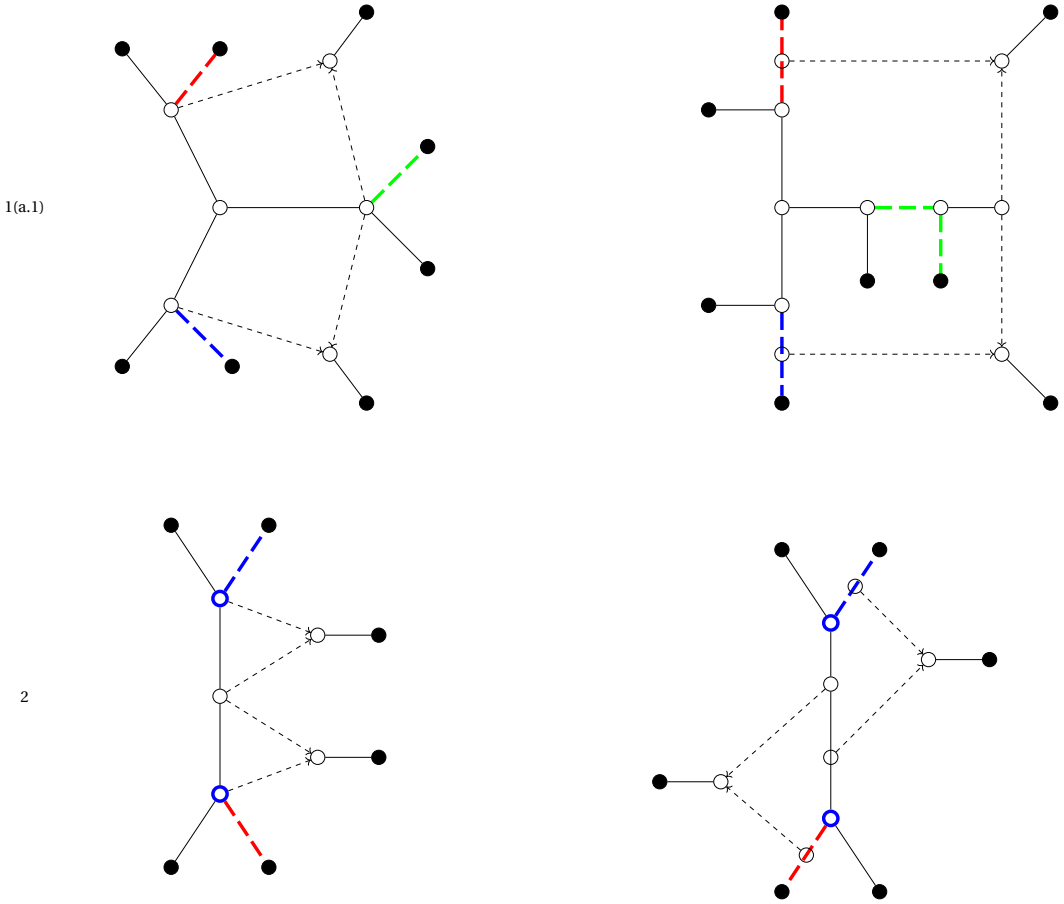


Figure 6.3: Examples of Construction 4, where blue and red edges display strong leaf edges and green edges represent an overlapping strong leaf edge for both reticulation leaves. The rest of the constructions follow a similar method.

6.2.3. One missing leaf

In this section, we will note the missing leaf as x.

Construction 5. 1. Suppose that if we remove the leaf x and its corresponding reticulation node with three incoming reticulation edges, we get a level-0 network with one internal node with three adjacent internal nodes. Then, for x, we determine a strong leaf-edge-group. This will result in three strong leaf-edges. Then, we relocate the reticulation edges to their corresponding adjacent strong leaf-edge. 2(1.a)

2. Suppose that if we remove the leaf x and its corresponding reticulation node, we get a level-1 network with a cycle-blob with two pendant subtrees with one of these subtrees below its reticulation node. Then, for x, we determine a strong leaf-edge-pair and relocate its reticulation edge to their corresponding adjacent strong leaf-edge. 2(2.a)

3. Suppose that if we remove the leaf x and its corresponding reticulation node, we get a level-1 network with a cycle-blob with one pendant subtree, which is below its reticulation node. Then, we determine a strong leaf-edge-group consisting of three strong leaf-edges. Two of these edges have a same adjacent edge u_1u_2. Let the other strong leaf-edge be v_1v_2. Then, for x, we attach its reticulation edges to u_1u_2 and v_1v_2. 2(2.b)

4. Suppose that if we remove the leaf x and its corresponding reticulation node, we get a level-1 network with a cycle-blob with one pendant subtree and only one leaf under the reticulation node, such that there are exactly three internal nodes with two adjacent leaves. Then, we determine a strong leaf-edge-pair for x and relocate its reticulation edges to their corresponding adjacent strong leaf-edge. 2(3)

5. Suppose that if we remove the leaf x and its corresponding reticulation node, we get a level-1 network with a cycle-blob with one pendant subtree and only one leaf y under the reticulation node, such that

there are exactly two internal nodes with two adjacent leaves. First, we relocate the reticulation edges for y as it is done in a level-1 network. Then, we determine the strong edges for x. If x has a strong leaf-edge-pair, we attach its reticulation edges to these edges. If x has three strong leaf-edges, two of these edges are both adjacent to an edge u_1u_2 and the other strong leaf-edge is v_1v_2. We attach its reticulation edges to u_1u_2 and v_1v_2. 2(4,8)

6. Suppose that the canonical form is level-1 with its reticulation leaf x, that is also the missing leaf. Remove the leaf x and its reticulation node. Then, we determine the strong edges for x. If there exists a strong leaf-edge-pair, we attach the reticulation edges of x to this edge-pair (2(5,6,7)). If the strong leaf-edge-group consists of three edges, we attach three reticulation edges for x to these edges, resulting in a reticulation node with three incoming edges. If the strong leaf-edge-group consists of four edges, two edges are both adjacent to an edge uv. We attach three reticulation edges for x to uv and the two strong leaf-edges non-adjacent to uv (2(1.b)).

7. Suppose that if we remove the leaf x and its corresponding reticulation node, we get a level-1 network with a cycle-blob with two pendant subtrees and only one leaf y under the reticulation node. First, we relocate the reticulation edges for y as we would in a level-1 network. Then, we determine a strong edge-pair for x, and relocate its reticulation edges to their corresponding adjacent strong leaf-edge. 2(9)

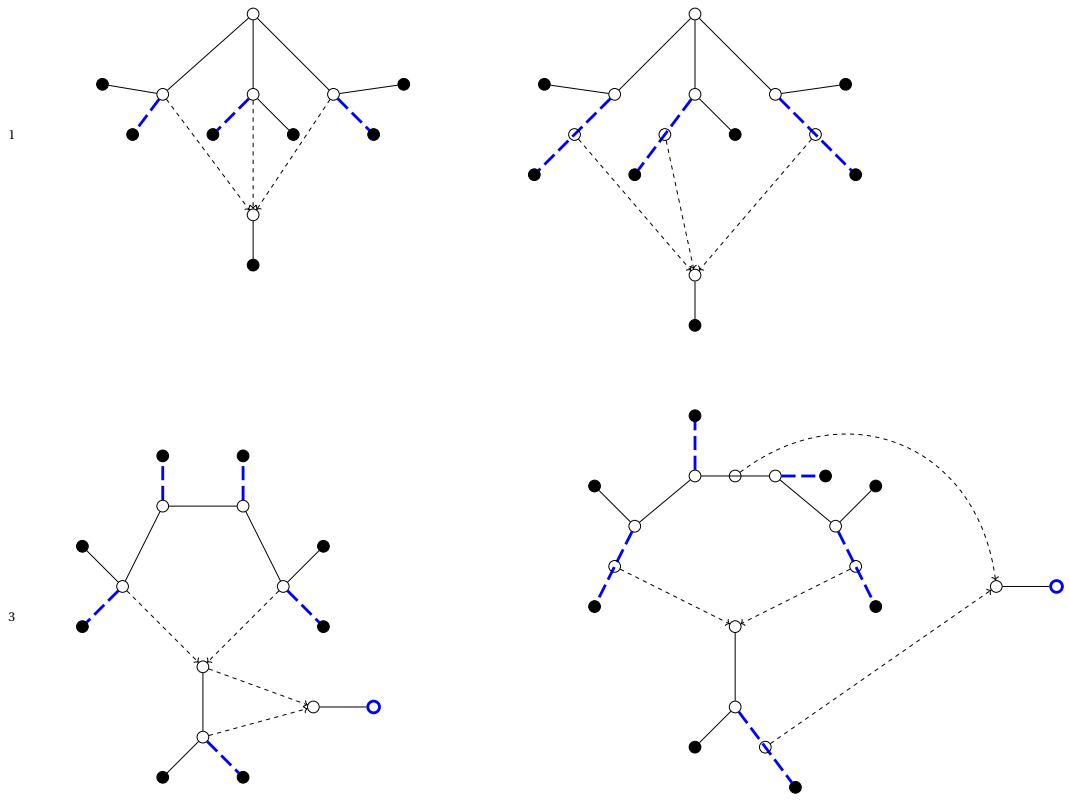


Figure 6.4: Examples of Construction 5, where blue edges display strong leaf edges, and the blue nodes represent the missing leaf. The rest of the constructions follow a similar method.

6.2.4. Two missing leaves

In this section, we will note the missing leaves as x, y.

Construction 6. First, we determine if the split $\{x, y\}|\mathcal{Y}\backslash\{x, y\}$ exists using quartets.
(a) If not, then for both x, y, we determine a strong leaf-edge-group. If the group is of size 2, we attach the reticulation edges as regularly. If the group is of size 3 or 4, we find the one or two edges adjacent to two strong leaf-edges and attach the reticulation edges to these edges.
(b) If the split exists, we again determine a strong leaf-edge-group for both x, y on the network without x and y, but also using all leaves in $\mathcal{Y}$ while determining the strong leaf-edges (thus also including x and y.

1. If both have the same exact leaf-edge-pair, we attach two new nodes u, v to both strong leaf-edges. Then, for both x, y, we attach two reticulation edges for x, y separately to u and v.

2. If one of x, y has a leaf-edge-group of size three, we attach the reticulation edges for both x, y as regularly to the strong leaf-edges.

3. If both leaf-edge-groups are of size two and share an edge, we attach the reticulation edges as regularly.

4. If both leaf-edge-groups are of size two and do not share an edge, we do the following. Suppose that only one strong leaf-edge-pair has an edge uv adjacent to both strong edges. The other strong leaf-edge-pair we call s_1, s_2. Then, we add reticulation edges from s_1, s_2 to the leaf with the strong leaf-edge-pair s_1, s_2, let this one be x. Then, for y, we add a reticulation edge from uv and from xw, where w is the reticulation node directly above x.

5. If both leaf-edge-pairs have an edge uv adjacent to both strong edges, we add two reticulation edges from u and v to a reticulation node w, with x, y both attached to w.

6. If both leaf-edge-groups consist of only one edge, we attach two reticulation edges from both strong leaf-edges to a reticulation node w, with x, y both attached to w.

1

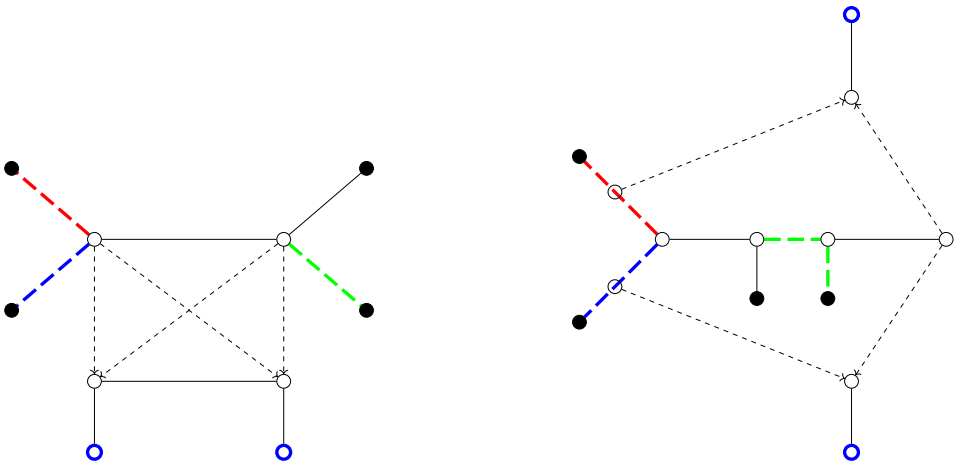


Figure 6.5: Examples of Construction 6, where blue and red edges display strong leaf edges and green edges represent an overlapping strong leaf edge for both reticulation leaves. The blue nodes represent the missing leaves. The rest of the constructions follow a similar method.

Algorithm 3 Algorithm for constructing the canonical form $\mathcal{N}_t^c$ by using its quartets and the canonical form $\mathcal{N}^c$ obtained with quarnet-splits.

Data: quartets $\mathcal{Q}^t(\mathcal{N})$ of a semi-directed level-2 network $\mathcal{N}$ on $\mathcal{X}$; the canonical network $\mathcal{N}^c$
Result: The canonical form $\mathcal{N}_t^c$ of a semi-directed phylogenetic network $\mathcal{N}$

Let G be $\mathcal{N}^c$;
foreach *internal blob* $\mathcal{B}$ *in* G **do**
  Isolate the blob by taking a random leaf from each pendant subtree of $\mathcal{B}$, resulting in the leaf-set $\mathcal{Y}$;
  if $|\mathcal{Y}| < 6$ **then**
    continue
  Determine the number of missing leaves nML from $\mathcal{Y}$ in $\mathcal{Q}_{\mathcal{Y}}$;
  if $nML == 0$ **then**
    Use Construction 4 on $\mathcal{B}$, resulting in $\mathcal{B}'$;
  if $nML == 1$ **then**
    Use Construction 5 on $\mathcal{B}$, resulting in $\mathcal{B}'$;
  if $nML == 2$ **then**
    Use Construction 6 on $\mathcal{B}$, resulting in $\mathcal{B}'$;
  Add $\mathcal{B}'$ back to G by replacing $\mathcal{B}$ with $\mathcal{B}'$ and substituting each leaf of $\mathcal{B}$ with its corresponding pendant subtree/subgraph;
return G

Theorem (Theorem 2)**.** Given the quartets $\mathcal{Q}^t(\mathcal{N})$ of a semi-directed level-2 network $\mathcal{N}$ on $\mathcal{X} = \{x_1, x_2, ..., x_3\}$, there exists an algorithm that constructs the canonical form $\mathcal{N}_t^c$.

Proof. Given is $\mathcal{Q}^t(\mathcal{N})$ of the network $\mathcal{N}$. As $\mathcal{Q}^s(\mathcal{N}) \subseteq \mathcal{Q}^t(\mathcal{N})$, we know that we can construct the canonical form $\mathcal{N}^c$. In the same way as in the proof of Theorem 1, we know that we can isolate a blob $\mathcal{B}$ and determine the number of missing leaves of the obtained simple network. Then, depending on the number of missing leaves, we apply Construction 4, 5 or 6 on the simple network, turning the isolated blob $\mathcal{B}$ into a more detailed blob $\mathcal{B}'$. Then, since changing a single blob to another blob does not affect the rest of the network, we can glue $\mathcal{B}'$ back into the network, replacing the blob $\mathcal{B}$. After changing all blobs in the network, we obtain the canonical network $\mathcal{N}_t^c$. □

7

Implementation

This chapter is dedicated to the implementation of previously discussed algorithms. As the implementation was not the main topic of this thesis, only Algorithm 1 has been fully implemented. But as now constructions and an algorithm exists for inflating level-2 blobs, this code can be extended. Unfortunately, this is for now a tedious task due to the amount of cases that exist. Therefore, it would be better to explore first if a more consistent algorithm exists that can construct level-2 networks with a less case-dependent construction, and if an iterative algorithm can be made in the same way it exists for level-1 networks.
The input of quarnet-splits is in a readable easy to write format, together with a list of all the leaves on top. See Figure 7.1 for an example input of the example network in the same figure. Running the code on this input will return the illustrated canonical form of a network with these quarnet-splits. The code returns it in a networkx format and displays the network separately in a plot.

The implementation together with example inputs can be found on GitHub (`https://github.com/simondeu/BEP`).

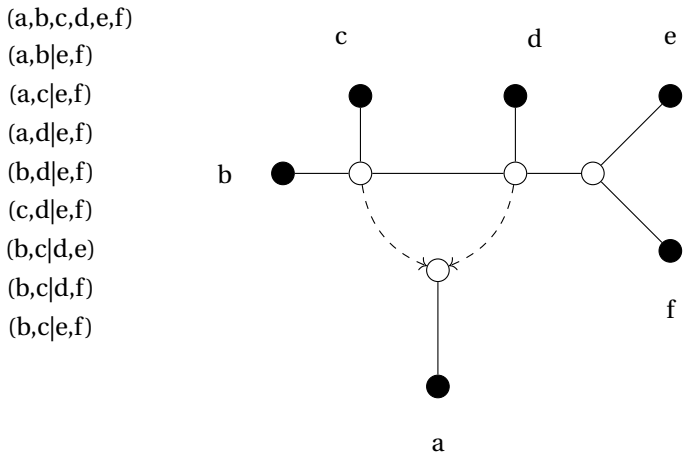


Figure 7.1: The list on the left is an example input of a set of leaves on the top and a list of quarnet-splits below. This input will result the canonical form on the right.

8

Discussion

As both theorems for finding the canonical form of a network have been proven for both the use of quarnet-splits and quartets, we can conclude that it is indeed possible to find a close to complete version of a level-2 network using information from its quarnets. Unfortunately, only a canonical form can be found, but by expanding the information to full quarnets, it would be possible to find the complete network. However, as finding complete quarnets may be significantly more difficult than finding quarnet-splits and quartets, we do not consider this detail in this thesis, which focusses on finding a maximal form with minimal information.
In this thesis, we worked with complete quarnet-split and quartet information. In reality, it is not always possible to find perfect quarnet-split information, as there is always a possibility that a quarnet-split is wrong or that a quarnet-split is missing. Therefore, it could be interesting to also include probability theory in this topic when constructing level-2 networks in practice. Other papers have already discussed and researched this problem for level-1 networks, expanding the results to level-2 networks is still open.
Now that it is known that level-2 networks can be constructed using information from quarnet-splits and quartets, the corresponding algorithm can be started to be optimised. Unfortunately, now the construction heavily depends on "if statements" as multiple different cases have to be constructed differently, and as completely different networks have equivalent quarnet-splits, thus also the same canonical form. What would be important for further research are the following questions. Can the network be constructed without the use of the blob-tree, and can the construction be done iteratively with adding leaves individually, in the same way a level-1 network can be constructed. Answering these questions could significantly optimise the process, which will help with constructing big networks that usually occur in practice.
Another direction for future research is to explore the construction of higher level networks. But, as previously stated, two level-3 networks or higher can have the exact same set of quarnets. With this, quarnet-split information will be even less helpful, and therefore, for constructing level-3 and higher networks, other information that is unique to a level-3 network should be considered. Another case that is not fully explored in this thesis is the inflation for m-blobs, for which $m \leq 5$. The reason is the following. For a strict level-2 simple network, in some cases, no quarnet-splits can be found. And without any quarnet-split information available for a simple network, we cannot construct anything. There exist only one or two cases for a strict level-2 simple network with less than 6 leaves where a quarnet-split is available, but as these are just one or two case of many, and as this quarnet-split does not provide information of if it is a level-1 or level-2 network, we did not consider inflating 5- or lower degree blobs. However, quartets could help with this issue, but this question is still left open for further research. Answering this question may be worth doing, as even small blobs may have a big impact on large networks.

Bibliography

[1] Travis Barton, Elizabeth Gross, Colby Long, and Joseph Rusinko. Statistical learning with phylogenetic network invariants, 2022. URL https://arxiv.org/abs/2211.11919.

[2] Martin Frohn, Niels Holtgrefe, Leo van Iersel, Mark Jones, and Steven Kelk. Reconstructing semi-directed level-1 networks using few quarnets. *Journal of Computer and System Sciences*, 152:103655, 2025. ISSN 0022-0000. doi: https://doi.org/10.1016/j.jcss.2025.103655. URL https://www.sciencedirect.com/science/article/pii/S0022000025000376.

[3] Katharina T. Huber, Leo van Iersel, Mark Jones, Vincent Moulton, and Leonie Veenema Nipius. When are quarnets sufficient to reconstruct semi-directed phylogenetic networks?, 2025. URL https://arxiv.org/abs/2408.12997.

[4] Samuel Martin, Vincent Moulton, and Richard M. Leggett. Algebraic invariants for inferring 4-leaf semi-directed phylogenetic networks. *bioRxiv*, 2023. doi: 10.1101/2023.09.11.557152. URL https://www.biorxiv.org/content/early/2023/09/19/2023.09.11.557152.

[5] Casey McGrath. Highlight: How a butterfly tree becomes a web. *Genome Biology and Evolution*, 13(7): evab143, 07 2021. ISSN 1759-6653. doi: 10.1093/gbe/evab143. URL https://doi.org/10.1093/gbe/evab143.

[6] Matthieu Willems, Etienne Lord, Louise Laforest, Gilbert Labelle, François-Joseph Lapointe, Anna Maria Di Sciullo, and Vladimir Makarenkov. Using hybridization networks to retrace the evolution of indo-european languages. *BMC Evolutionary Biology*, 16(1):180, 2016. ISSN 1471-2148. doi: 10.1186/s12862-016-0745-6. URL https://doi.org/10.1186/s12862-016-0745-6.