New algorithms and mathematical tools for phylogenetics beyond trees

Guillaume E. Scholz

School of Computing Sciences

University of East Anglia

A thesis submitted for the degree of

Doctor of Philosophy

April 2018

“Prépare toi, petit garçon
Elle s’ra longue, l’expédition”
Les cowboys fringants.

I would like to dedicate this thesis to Mr. B. Rinckel,
who highly inspired a growing young boy
some fifteen years ago.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification at this or any other university or other institute of learning. I certify that this thesis, and the research to which it refers, are the product of my own work, and that any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged.

List of Publications

- P. Gambette, K. T. Huber and G. E. Scholz. Uprooted phylogenetic networks. *Bulletin of Mathematical Biology* (2017) 79(9): 2022-2048.

- K. T. Huber and G. E. Scholz. Beyond representing orthology relations with trees. *Algorithmica* (2018) 80(1): 73-103.

- K. T. Huber, V. Moulton and G. E. Scholz. Three-way symbolic tree-maps and ultrametrics. *Journal of Classification*, in press.

- G. E. Scholz, A.-A. Popescu, M. I. Taylor, V. Moulton and K. T. Huber. OSF-Builder: A new tool for reconstructing and representing phylogenetic histories involving introgression, *submitted.*

Abstract

Phylogenetic trees and networks are mathematical structures for representing the evolutionary history of a set of taxa. The need for methods to build such structures from various type of data, as well as the need to understand the story these data may tell, give rise to exciting new challenges for mathematics and computer sciences. This thesis presents some recent advances in both these directions. It features new mathematical methodology for reconstructing phylogenetic networks, and new computational tools for inferring complex evolutionary scenarios. These come with a thorough analysis, assessing their attractiveness in terms of their theoretical properties. It expands on previous results, which are themselves briefly reviewed, and conclude with potentially interesting further research questions.

Preface

29$^{\text{th}}$ of February. Cold morning in the cobbled streets of Strasbourg. A hot chocolate, and a remark from the friend in front of me: “I never imagined one of us would work in Phylogenetics”. She was right, who could have guessed when, back in high school, Phylogenetics meant nothing more to me than “the easiest chapter of the Biology class”. I would not have associated the topic with Mathematics then. Perhaps with Logic. But what are Mathematics, if not Logic?

This thesis does not provide an answer to this question. It does, however, highlight some of the several links between Phylogenetics and Mathematics, as I have been led to discover and study them over the past three years.

For these three exciting years, I would like to thank first my primary supervisor, Dr. Katharina Huber. For everything. I add a special thank also to Prof. Vincent Moulton. If I am allowed to acknowledge a city, I would like to thank the inspiring streets and places of Norwich. A fine city, indeed. And of course, a big thank for the amazing NoBoG group, for all these evenings spent moving cubes.

Finally, I would like to send a few *mercis* back to France. Merci Laurent, for my first steps into the vast world of research. Merci Yann, Charlotte, and all those of you who frequently take me out of this world, in words exchanged or in time spent together (and Yann, merci for Figure 1(ii) as well.). Merci papa, maman and Jeanne, for always being to me such a peaceful river. And merci Marine, for all of the above and much more, for your help and advices, and for allowing me to take a part of you with me wherever I am.

Guillaume.
Borgarnes, September 2017.

Contents

Figures

Introduction

From a high level, a phylogenetic network is a picture. To biologists, genealogists, even linguists, it is a graphical representation of the interrelationships governing a set of taxa, such as individuals, genes, organisms, or languages, among others. It can come equiped with directions, in which case it is commonly interpreted as some kind of illustration of the common evolutionary history of a set of taxa. Figure 1(i) depicts such a network, that originally appeared in [49] in the context of studying the complex evolutionary relationships governing a set of five distinct bread wheat lineages. But a phylogenetic network is more than a simple picture. It is a complex system of lines, nodes and tips, an object which scientists more commonly call a graph, and it can be studied from a purely theoretical point of view. As such, it is highly attractive for both mathematicians and computer scientists.

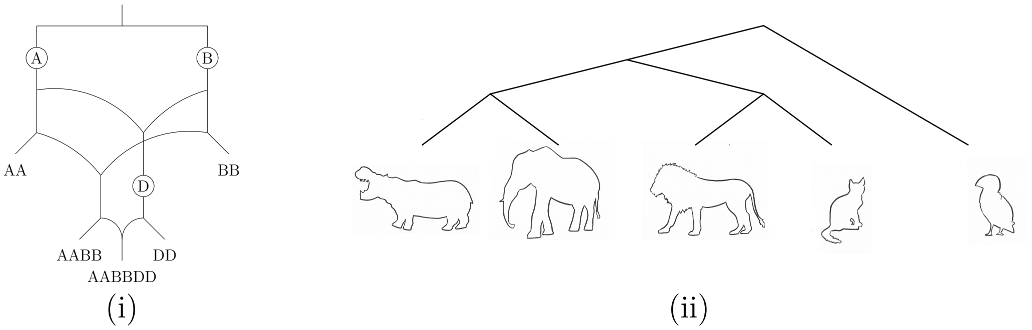


Figure 1: (i) A phylogenetic network depicting the evolutionary relationships between five lineages of bread wheat, indicated by AA, BB, DD, AABB, and AABBDD (see [49]). (ii) A phylogenetic tree on 5 species: hippopotamus, elephant, lion, cat and puffin. The four mammals are grouped together against the bird. Among the mammals, we distinguish between two groups, the herbivores (hippopotamus and elephant) and the carnivores (lion and cat).

One of the first examples of a phylogenetic network is depicted in Charles Darwin's *On the Origin of Species by Means of Natural Selection, or the Preservation* *of Favoured Races in the Struggle for Life*, published in 1859, and is in facts a tree. From a structural point of view, it differs from a "proper" phylogenetic network by the addition of the extra requirements that it does not allow for two distinct paths to merge into a single one.

The idea of a phylogenetic tree is closely related to the main theory defended by Darwin in that book, that is, that the diversity of life on earth as can be observed today results from a complex chain of evolutionary events. Following this idea, it then makes sense to try and obtain this chain by comparing taxa with each other other, the idea being that closely related species should not differ too much. The toy example in Figure 1(ii) depicts such a tree for five animal species, based on simple morphological characters.

Nowadays, with the significant increase of interest in molecular biology, helped by decisive technological breakthroughs, the quality and quantity of data from which a phylogenetic tree may be constructed have become more and more complex, revealing more and more incongruences. One of the consequences of this is that it is not always possible to represent data in the form of a phylogenetic tree. These incongruences may be genuine signals of complex evolutionary events, ranging from hybridization, to introgression and horizontal gene transfer, where part of the genome of an organism moves to the gene pool of another. Or they may be noisy signals, induced for example by the technologies used to generate the data, the large amount of data to treat, or due to missing information. Whatever the reason, the resulting conflicts in the data need to be taken into account one way or another.

Therefore, challenges for mathematicians and computer scientists are numerous. This thesis addresses some of these challenges, both from a mathematical and a computational perspective. To help develop a feel for them, we next classify them into two main types.

Inference of information from phylogenetic trees and networks. Apart from providing a graphical representation, one of the main interest in using phylogenetic trees or networks is to reveal new insights into the data set for which they have been constructed. Clearly, there is a strong link between these new insights and the theoretical properties of such structures, thus making their study as abstract combinatorial objects an important task in phylogenetics.

Phylogenetic trees and networks also provide information when put in perspective with each other, that can not be observed from investigating each on their own. For example, if the evolution of a set A of taxa is represented by a phylogenetic tree T, and the evolution of a set B of taxa represented by a phylogenetic tree T', what can these trees, taken together, tell us about the common evolutionary history of A and B? This question naturally extends to sets of more than two trees. The purpose may be, for example to build a single tree reconciling all the trees of a set of tree (see the "reconstruction problem" below), or to construct some sort of consensus. Indeed, as we shall see in Chapter 2, phylogenetic trees or networks can sometimes be "put together", in a certain well-defined sense, to reconstruct a larger one. However, we shall see in Chapter 5 that there are other ways to reconcile two (or more) trees than to build a bigger tree from them.

Reconstruction of phylogenetic trees and networks. This is one of the "classic" problems in phylogenetics, and is concerned with the translation of some data on a given set of taxa into a phylogenetic tree or network representing these data. This task can be summarized as follows. A phylogenetic tree or network N provides information about the set of taxa it is inferred from. Does the knowledge of that information, without the previous knowledge of the tree or network used to infer it, allow for recovering N? This task is, for example, the starting point of the work presented in Chapter 3.

Most of the time the answer to this question takes the form of an algorithm, which takes as input some data, and generates a phylogenetic tree or network that in some way displays these data, should one exist. Such an algorithm is often based on a *characterization*, which provides conditions on these data for them to be representable using the desired structure. Many of these algorithms also have a software implementation, so that they can be used without any previous knowledge of their inner-workings.

The difference in treatment made here between phylogenetic trees and networks is not incidental, and two reasons stand for favoring trees over networks, if possible. As mentioned before, phylogenetic trees are much simpler in structure, which makes them, in some circumstances, both more suitable as a representation of complex relationships, and easier to deal with from a purely theoretical point of view. Split systems, which play a central role in Chapter 4, are a perfect illustration of this simplicity. Without embarking on too much detail here, it suffices to say that a split system is closely related to a certain type of distance between the taxa considered. The question of the representability of a split system by a phylogenetic tree has been answered in [10], together with a characterization of split systems that can be represented by a tree (Theorem 1.2.3), as this does not hold for all split systems. Thus the question becomes, how and under which conditions a split system can be represented by a phylogenetic network, if that split system does not satisfy the aforementioned characterization. The work that has been realized in the context of trees forms a solid starting point to address this question, which can then be thought of as a generalization of the previous results, going one step further in complexity. Chapter 2 also adopts the same generalization pattern.

As the information inference and the reconstruction problems are closely entangled with each other, so are the distinct chapters of this thesis. It can however be divided into two main parts, which, although not independent, deal with different aspects of phylogeny research. Each of Chapters 2 to 5 is based on a research paper that has appeared (or is under peer review) in a journal in the past three years (see below for the references). Details about these papers, including their availability and the respective contribution of their authors, can be found in the introduction of the respective chapters. We next provide details about the various chapters that make up this thesis.

Chapter 1 presents a review of relevant definitions and results that will play a role in this thesis. As such, it should not be seen as an exhaustive review of the literature, but as a tool to put the findings of the subsequent chapters into context. In particular, unless stated otherwise explicitly, all definitions and terminology introduced there will also be used for the remainder of this thesis.

Chapters 2, 3 and 4 are based on [41], [39] and [29] respectively. All three of them concern reconstruction problems, and introduce novel approaches to tackle them. In all three cases, the starting point is a variation of the general notion of a distance.

Chapter 5 is based on [54]. It presents a new type of reconciliation problem, which can be summarized as follows: Given a set of phylogenetic trees representing the evolution of some sets of species, and a further tree representing the evolution of the alleles of some genes carried by these species, how can we infer "introgression", that is, the gene flow within these distinct sets of species.

Chap. 1

Phylogenetics and Mathematics: a brief overview

1.1 Basic definitions

In this section, we first review required basic concepts for directed and undirected graphs, and then elaborate on them in the context of phylogenetic trees and networks. Unless stated otherwise, the terminology follows [43].

1.1.1 Directed and undirected graphs

We start by reviewing the basic definitions for undirected graphs, and then move on to directed ones.

Undirected graphs. An *undirected graph* (or graph for short) is a pair $G = (V, E)$, where V is a finite set of *vertices*, and E is a finite set of *edges* $e = \{u, v\}$, where u and v are two distinct elements of V (see Figure 1.1(i) for a first example). For convenience, we sometimes write $V(G)$ and $E(G)$ for V and E respectively.

For a graph $G = (V, E)$ and $u, v \in V$, we say that an edge $e = \{u, v\} \in E$ is *incident* with u and v. We also say that two edges are *adjacent* if they share a vertex. The *degree* of v is defined as the number of edges in E that are incident with v. We call v *isolated* if v has degree 0, and *internal* if v has degree at least two. If v has degree 1, we say that v is a *leaf* and we denote by $L(G)$ the set of leaves of G. Finally, a *subgraph* $G' = (V', E')$ of $G = (V, E)$ is a graph such that $V' \subseteq V$ and $E' \subseteq E$.

Two graphs $G = (V, E)$ and $G' = (V', E')$ are said to be *isomorphic* if there exists a bijection $\phi : V \to V'$ such that for any two vertices $v, u \in V$, we have that $\{u, v\} \in E$ if and only if $\{\phi(u), \phi(v)\} \in E'$. If G and G' are not isomorphic, we

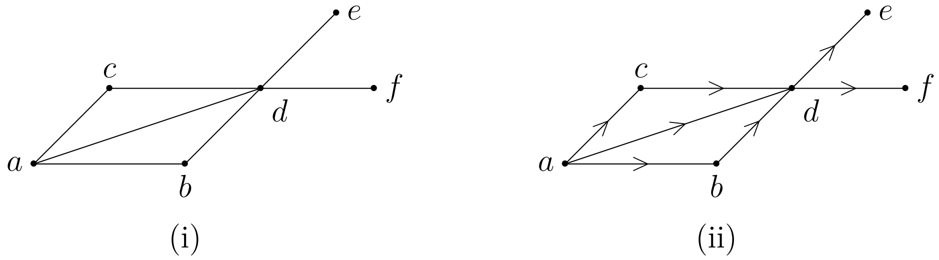


Figure 1.1: (i) An undirected graph (ii) A directed graph. In both cases, black dots represent vertices of the graph, and lines joining them are edges, whose direction in (ii) are given by arrows. In both graphs vertices e and f are leaves, since they have degree one. The vertex d has degree 5 in both (i) and (ii). It has in-degree 3 and out-degree 2 in (ii).

simply say that G and G' are *different*. By abuse of terminology, we say that a graph is *unique* (subject to certain properties), if it is unique up to isomorphism.

The notion of a *path* turns out to be fundamental to the definition of phylogenetic trees and networks, our main objects of interest. Suppose $G = (V, E)$ is a graph. A path P is a sequence of $k + 1 \geq 1$ distinct vertices $v_0, \dots, v_k \in V$ such that for all $0 \leq i \leq k - 1$, the vertices v_i and v_{i+1} are joined by an edge $e_i \in E$. We say that P *connects* v_0 and v_k, and for all i, that the edge e_i is *lying on* P. We call the vertices v_2 to v_{k-1} *internal vertices* of P. If, in addition, v_k and v_0 are connected by an edge, we say that the sequence $v_0, \dots, v_k, v_0$ is a *cycle* of G.

The *length* of a path P is defined as the number of edges lying on P. More generally, we can associate to each edge e of a graph $G = (V, E)$ a *length* $\omega(e)$, where $\omega : E \to \mathbb{R}_{>0}$. In that case, the pair (G, ω) is said to be a *weighted graph* and the *length* of a path P in G is defined as the sum of the length of the edges lying on P. Note that we can always consider an unweighted graph as a weighted graph whose edges all have length one.

If any pair of distinct vertices of a graph can be connected by a path, we say that this graph is *connected*. Following this idea, we define a *connected component* of a graph to be a maximal connected subgraph.

Operations on undirected graphs. Various operations have been defined for graphs. To review some of them, suppose for the remainder that $G = (V, E)$ is a graph. The *edge-deletion* operation simply consists of removing an edge e from E. The related *vertex-deletion* operation removes not only a vertex v from V, but also all edge from E that are incident with v. Clearly, these operations give rise to subgraphs of G. If G is connected, we call an edge e of G a *cut-edge* if its deletion disconnects G. Similarly, a vertex v of G enjoying this property is called a *cut-vertex*.

We can also *subdivide* an edge $e = \{u, v\} \in E$, by adding a new vertex w to V, and replacing the edge $\{u, v\}$ by two edges $\{u, w\}$ and $\{v, w\}$. This operation is reversible, as we can *suppress* a vertex w of G of degree 2 by first deleting w in the above sense, and then adding a new edge joining the two vertices that were adjacent to w. If G is weighted, when removing a degree two vertex w, we assign to the newly created edge the sum of the length of the two edges incident to w. Similarly, when subdividing an edge e of G, we need to ensure that the sum of the weight assigned to the two newly created edges is precisely the length of e. This allows us to talk about the *addition* of a new edge to a graph $G = (V, E)$, which can either be done by adding a new edge to E, or by subdividing two edges of G with new vertices u and v, and adding to E the edge $\{u, v\}$.

A further operation is that of *collapsing* an edge $e = \{v, w\}$ of E. In that case, we remove e from E, and merge v and w in V. In other word, we add a new vertex u to V, remove v and w from V, and define every edge incident to v or w to be incident to u instead. Note that collapsing edges of G does not result in a subgraph of G. The reverse operation is called *vertex expansion*. It works by replacing a vertex u of V by a pair of new vertices v and w, adding the edge $\{v, w\}$ to E, and defining all edges in E adjacent with u to be adjacent either with v or with w instead. Note that although collapsing an edge can only be performed in an unique way, this is not the case for the vertex expansion operation. Indeed, for each edge e adjacent to some vertex v of G we want to expand, we have to decide to which of the two new vertices e should be adjacent.

Directed graphs. Given a graph $G = (V, E)$, we can assign to each edge $e \in E$ a *direction*. Indeed, since e is given by an subset of two elements V, we can define an ordering on that subset. Motivated by this, we say that a *directed graph* $G = (V, E)$ consists of a finite set of *vertices* V, and a finite set of *arcs* (i. e. directed edges) E. We denote an arc $e \in E$ as (v, w) and say that e is *directed* from v to w (see Figure 1.1(ii)). We refer to v as the *tail* of e, and to w as the *head* of e. Moreover, we say that v is a *parent* of w and that w is a *child* of v.

Let $G = (V, E)$ be a directed graph. We call the undirected graph $U(G)$ obtained from G by ignoring the directions of the edges of E and suppressing resulting degree-2 vertices the *underlying graph* of G. Through this transformation, most of the definitions previously introduced for undirected graph also apply within the directed framework. However, directed graphs also induce specific terminology, related to the extra information they contain. For a vertex $v \in V$, we call the number of arcs of E directed to v the *in-degree* of v, and the number of arcs directed away from v the *out-degree* of v. Clearly, the degree of v is equal to the sum of its in-degree and its out-degree. We also say that a path $v_0, \ldots v_k$, $k \geq 1$ of G is a *directed path* if for all $0 \leq i \leq k-1$, the edge e_i is directed from v_i to v_{i+1}. By extension, we call a sequence $v_0, \ldots, v_k, v_0$ a *directed cycle*, if $v_0, \ldots, v_k$ is a directed path and $(v_k, v_0) \in E$.

1.1.2 Phylogenetic trees and networks

As already mentioned, the graph-theoretical notion of a path allows us to define phylogenetic trees and networks, both of which are central to phylogenetics. From now on, let $X = \{x_1, \ldots, x_n\}$ be a finite set of size $n \geq 3$. In a phylogenetic context, the elements of X are called *taxa* (singular *taxon*), and represent, for example, species, individuals or genes.

Phylogenetic networks have been defined and studied both as undirected and directed graphs (see [33, 43]). We first consider the undirected ones, and then turn our attention to the directed ones.

Unrooted phylogenetic networks. We say that an undirected graph N is an *(unrooted) phylogenetic network* on X if N satisfies the three following assumptions:

(N1) N is connected.

(N2) The leaves of N are the elements of X.

(N3) N does not contain any vertex of degree 2.

A special case of an unrooted phylogenetic network is an *unrooted phylogenetic tree*, that is, an unrooted phylogenetic network that does not contain any cycle. Phylogenetic trees enjoy some additional properties, making them particularly attractive. For example, any two vertices of a tree T are connected by exactly one path. If we remove an edge from T, we necessarily obtain a disconnected graph (i. e. all edges are cut-edge), and if we add a new edge to T, we obtain a cycle.

Reflecting the assumption that an evolutionary process is assumed to generate two distinct species, we say that an unrooted phylogenetic network N is *binary* if every internal vertex of N has degree 3. Note that we can always transform a non-binary unrooted phylogenetic network into a binary one by a succession of vertex-expansion operations (see Section 1.1.1). We call this operation a *resolution* of N.

Rooted phylogenetic networks. A *rooted (phylogenetic) network* on X is a directed graph N such that:

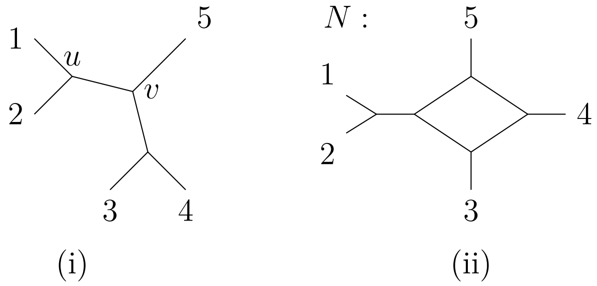


Figure 1.2: (i) An unrooted phylogenetic tree on $X = \{1, 2, 3, 4, 5\}$ (ii) An unrooted phylogenetic network on X that is not a tree. Both graphs in (i) and (ii) are binary. Note that we do not always represent vertices with a dot, as we did in Figure 1.1, when there is no need to do so.

(R1) The underlying graph $U(N)$ is an unrooted phylogenetic network, that is, $U(N)$ satisfies (N1) – (N3).

(R2) N contains exactly one vertex ρ of in-degree 0.

(R3) N does not contain any directed cycles.

(R4) N does not contain any vertex of in-degree and out-degree 1, or of out-degree 0 and in-degree greater than one.

For N a rooted phylogenetic network, the vertex ρ defined in (R2) is called the *root* of N (see Figure 1.3[1]). As in the unrooted case, a rooted phylogenetic network N whose underlying graph does not contain a cycle is called a *rooted phylogenetic tree*. If N is a rooted phylogenetic tree, the directed path from ρ_N to any leaf of N is unique, whereas this may not be true in general.

We say that a rooted phylogenetic network N is *binary* if ρ_N has out-degree two, and all its other internal vertices have degree three. Note that if N is binary, then so is $U(N)$. Contrary to a binary rooted phylogenetic tree, where each internal vertex that is not the root has necessarily in-degree one and out-degree two, an internal vertex of a binary rooted phylogenetic network can either have in-degree one and out-degree two, or vice versa. For N a rooted phylogenetic network, we call a vertex of in-degree zero or one a *tree-vertex*, and a vertex of in-degree greater than one a *hybrid vertex*. For example, the rooted phylogenetic networks depicted in Figure 1.3(ii) and (iii) both contain exactly one hybrid vertex, the parent of 4

[1] In Figure 1.3 and in subsequent figures, we draw rooted phylogenetic networks with their root at the top, and the arcs are implicitly assumed to be directed downwards, away from the root.

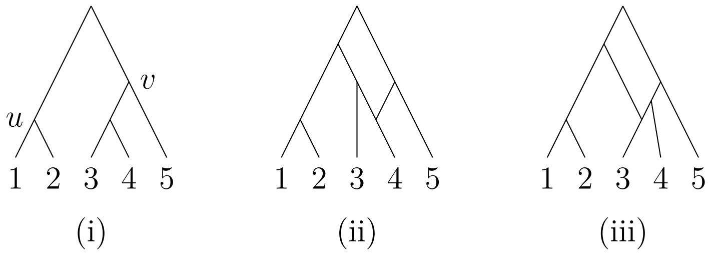


Figure 1.3: (i) A rooted phylogenetic tree on $X = \{1, 2, 3, 4, 5\}$. (ii) and (iii) Two distinct rooted phylogenetic networks with the same underlying graph, that is the unrooted phylogenetic network depicted in Figure 1.2(ii).

and 3 respectively. As is easy to see, a rooted tree does not contain any hybrid vertices.

Within an evolutionary context, the idea of ancestorship is central. This notion is captured for a rooted phylogenetic tree T on X as follows: A vertex v of T is called an *ancestor* of a leaf $x \in X$ if v lies on the path from the root ρ_T of T to x. We also say that x is an *offspring* of v, and we denote the set of offsprings of v by $C(v)$ (sometimes also called the *cluster* induced by v). If Y is a subset of X of size $k \geq 2$, we say that a vertex v of T is the *last common ancestor* of Y if v is an ancestor of all elements of Y, and no child of v enjoys this property. Note that such a vertex always exists, and that it is necessarily unique. We put $v = lca_T(Y)$, where we may omit the index if the tree T we are referring to is clear from the context. Note that if $Y = \{y_1, \ldots, y_k\}$, we usually write $lca(y_1, \ldots, y_k)$ rather than $lca(\{y_1, \ldots, y_k\})$, to avoid overcomplicated notation. For example, in Figure 1.3(i), we have $lca(1, 2) = u$ and $lca(3, 4, 5) = v$.

Although a similar formalization of ancestorship might be attractive for rooted phylogenetic networks that are not also phylogenetic trees, the definition of a last common ancestor of a subset Y of X does not easily carry over. Indeed, for N a rooted phylogenetic network on X and $Y \subseteq X$, there may be more than one vertex of N satisfying the condition for being a last common ancestor of Y. In the network N depicted in Figure 1.4, for example, both the vertices v ad w satisfy that condition for the leaves 2 and 4. To tackle this problem, the notion of a stable ancestor was introduced in [42], where a vertex v of N is called a *stable ancestor* of a leaf $x \in X$ if v lies on every path from ρ_N to x. We can then consider the *last stable common ancestor* of a subset $Y \subseteq X$, which is unique and corresponds to the last common ancestor of Y if N is a tree. Continuing with the example of the network N depicted in Figure 1.4, the last stable common ancestor of the leaves 2 and 4 is the root of N.

Alternatively, we can consider the notion of a lowest common ancestor, where we say that a vertex v of N is a *lowest common ancestor* for a subset $Y \subseteq X$ if for all $x \in Y$, there exists a path from the root to x such that v is the last vertex that is common to all these paths. Here again, such a vertex is unique and corresponds to the last common ancestor of Y in the case N is a tree, but it may not be unique otherwise.

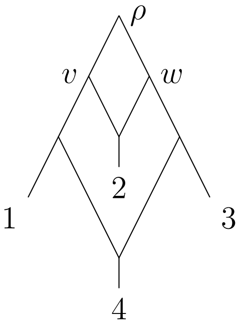


Figure 1.4: A rooted phylogenetic network. The vertex v is the lowest common ancestor of $\{1, 2\}$, whereas the last stable common ancestor of $\{1, 2\}$ is ρ. The lowest common ancestor of $\{2, 4\}$ is not unique, since both v and w satisfy the property of a lowest common ancestor for the set $\{2, 4\}$.

Subtree and subnetwork. Given a phylogenetic network N on X, there is sometimes a need to extract some substructures from N. These substructures are called *subnetworks* (or *subtrees* in case N is a phylogenetic tree), as they are themselves phylogenetic networks on some subset of X. Subnetworks can be of two different types:

If N is a rooted phylogenetic network and v is a vertex of N, we call the *subnetwork of N rooted at v* the subgraph N_v of N that consists of all edges and vertices that can be reached from v in N via a directed path. Clearly, N_v is a rooted phylogenetic network on the set $C(v)$ of offsprings of v.

The second type does not require N to be rooted, and does not, in general, lead to a subgraph of N. Considering a subset Y of X, we define the *subnetwork of N induced by Y* in two steps. First, we delete all edges and vertices of N that do not lie on a (undirected) path between two elements of Y. Then, we successively suppress resulting degree two vertices. As we shall see, this latter type of subgraph leads to the key notions of a "triplet" (Section 1.4.1) and of a "trinet" (see Section 1.4.2).

Rooting an unrooted phylogenetic network. As seen above, rooted and unrooted phylogenetic network are closely related to each other, since due to Property

(R1), the underlying graph $U(N)$ of a rooted phylogenetic network N is always an unrooted phylogenetic network. Conversely, it is possible to assign directions to the edges of an unrooted phylogenetic network N, in order to get a rooted phylogenetic network N_r satisfying $U(N_r) = N$.

To do this, we first need to define a root ρ for the network N_r. This can be done either by declaring an internal vertex of N to be ρ, or by subdividing an edge of N and defining the newly created vertex to be ρ. The latter is usually preferred in case we want to transform a binary unrooted phylogenetic network into a binary rooted one, as in that case, the created root has out-degree two. Once the root is defined, we attribute a direction to each edge of N, making sure that we do not conflict with Properties (R2) and (R3). If N is a tree, there is only one way to do so, that is, direct all edges "away from the root". Otherwise, as shown in [27], there may be as many as $2^{|X|/2}$ different way to assign directions to the edges once the root has been defined.

As an example, the rooted phylogenetic tree in Figure 1.3(i) is obtained by subdividing the edge $\{u, v\}$ of the unrooted tree in Figure 1.2(i), by introducing a new vertex ρ, and directing the edges away from ρ. On the other hand, both the rooted phylogenetic networks (ii) and (iii) in Figure 1.3 are obtained by subdividing the same edge of the unrooted phylogenetic network in Figure 1.2(ii), but the way the edges have been directed in both networks is different.

A second way to transform a rooted phylogenetic networks into an unrooted phylogenetic networks, and back, is the *Combinatorial Farris transform*, introduced in [22] in the context of phylogenetic trees (see [17] for a review of its use and properties). For N an unrooted phylogenetic network, we first choose a leaf $x \in X$. We then remove x from N, and define the vertex u adjacent to x in N to be the root, assigning directions to the edges in the way described above. This process leads to a rooted phylogenetic network on $X - \{x\}$. The reverse operation is the following: For N a rooted phylogenetic network on X with root ρ, we first add a new vertex $r \notin X$, and the arc (r, ρ). By forgetting about the direction of the arcs, we obtain an unrooted phylogenetic network N' on $X \cup \{r\}$.

1.1.3 The variety of phylogenetic networks

The need to compare structures of the same type or to evaluate how "different" they are from each other is essential in many areas of mathematics, and phylogenetics makes no exception. To make this more precise it is necessary to define what is meant by writing, for two phylogenetic networks N and N', that $N = N'$. To this aim, we expand the notion of an isomorphism between graphs to phylogenetic networks. We say that two phylogenetic networks N and N' have the *same topology* is they are isomorphic as graphs, according to the definition given in Section 1.1.1. We say that they are *isomorphic* if in addition, they have the same set of leaves X and this isomorphism is the identity on X. If this is the case, we write "$N = N'$".

Reflecting the various biological processes that phylogenetic networks aim to model, a number of different types of such networks have been introduced in the literature. We focus here on three approaches, all of which induce particular types of networks that will be of interest in the following chapters.

k-nested networks. The underlying idea of this approach is to look at the complexity, in a sense to be defined, of some substructures contained in a phylogenetic network. Put differently, the aim is to capture the complexity of the non-treelike parts of a phylogenetic network, known as blobs (or blocks in the rooted case). A *blob* of an unrooted phylogenetic network N is defined in [27] as a maximal connected subgraph of N that does not contain any cut-vertex. The *level* of a blob B is then defined as the minimal number of edges that need to be removed from B in order to obtain a graph that does not contain any cycle. Thus, a blob of level-0 is a cut-edge, and blobs of level-1 are isolated cycles, that is, cycles that do not share an edge with a further cycle. If N is a rooted phylogenetic network, we simply call a subgraph B of N a *block* of N if $U(B)$ is a blob of $U(N)$, and define the level of B in N as the level of $U(B)$ in $U(N)$.

For $k \in \mathbb{N}$ and N an unrooted (*resp.* rooted) phylogenetic network, we say that N is a *k-nested network* if the blobs (*resp.* the blocks) of N have level at most k. We say that N is a *level-k* network if N is a k-nested network such that no two blobs (*resp.* blocks) of N of level greater than zero share a vertex. In a binary context, the definitions of a k-nested network and of a level-k network are equivalent. For example, the rooted phylogenetic networks depicted in Figures 1.3(ii) and (iii) and their underlying graph depicted in Figure 1.2(ii) are level-1 (and 1-nested) networks, and the rooted phylogenetic network depicted in Figure 1.4 is a level-2 (and a 2-nested) network. Note that a phylogenetic network N is a level-0 network (or equivalently, a 0-nested network) if and only if N is a phylogenetic tree. Note also that if N is a rooted, binary phylogenetic network, the level of a block B of N is precisely the number of hybrid vertices of B.

As we shall see, 1-nested and level-1 networks (also known in the rooted case as *galled trees*, see [33]) play a key role in Chapters 2 and 4. In such networks, the only blocks are cut-edges and isolated cycles, both structurally simple to understand. In fact, it is possible to define 1-nested and level-1 networks without going for the more general notion of a blob. Indeed, a 1-nested network can be seen as a phylogenetic network in which two cycles do not share any edge, whereas level-1 networks are phylogenetic networks in which two cycles do not share any vertex.

Tree-child and tree-sibling networks. A second idea is to look at hybrid vertices, since such vertices represent the main difference between phylogenetic trees and phylogenetic networks that are not trees. Although the block approach can apply both to the rooted and the unrooted case, this one can only be used in the rooted one. The main idea is to check “how close” the hybrid vertices are to each other. Roughly speaking, this is done by looking, for a hybrid vertex h of a rooted phylogenetic network N, at the nature (hybrid or tree-vertices) of the vertices of N sharing a parent with h.

To do so, for an internal vertex v of a rooted phylogenetic network N, the authors of [12] and [11] consider two situations:

- If v has at least one child that is a tree vertex, we say that v is a *tree-child vertex*.

- If v is not the root and has a parent u with a child other than v that is a tree-vertex, we say that v is a *tree-sibling vertex*.

From these definitions, we can define two types of rooted phylogenetic networks. Let N be such a network. If all non-leaf vertices of N are tree-child, we say that N is a *tree-child network*. If all hybrid vertices of N are tree-sibling, we say that N is a *tree-sibling network*. Clearly, a tree-child network is a tree-sibling network, whereas the converse is not necessarily true. Phylogenetic trees, as well as 1-nested networks, are examples of tree-child (and thus, tree-sibling) networks.

Coming back to the underlying idea of these definitions, we remark that in a tree-child network N, all vertices sharing a parent with a hybrid vertex are tree-vertices. On the opposite, if N is a rooted phylogenetic network that is not tree-sibling, then there exists a hybrid vertex h of N such that all vertices sharing a parent with h are also hybrid vertices.

Tree-based networks. Finally, the third approach of interest to us was introduced in [26], in the form of the question “Which networks can be obtained from a tree by adding new arcs to it?”. To answer this question, the authors formally define a rooted phylogenetic network N as a *tree-based networks* if there exists a binary phylogenetic tree T on the same set of leaves as N such that N can be obtained from T by carrying out the following three operations (see Figure 1.5):

(a) Introduce any number of new vertices by subdividing arcs of T.

(b) Define new arcs between these newly created vertices, avoiding the creation of directed cycles.

(c) Suppress any resulting vertices with in-degree and out-degree 1.

The phylogenetic tree T is said to be a *support tree* for N. However, that tree need not be unique. In fact, for the tree-based phylogenetic network N on

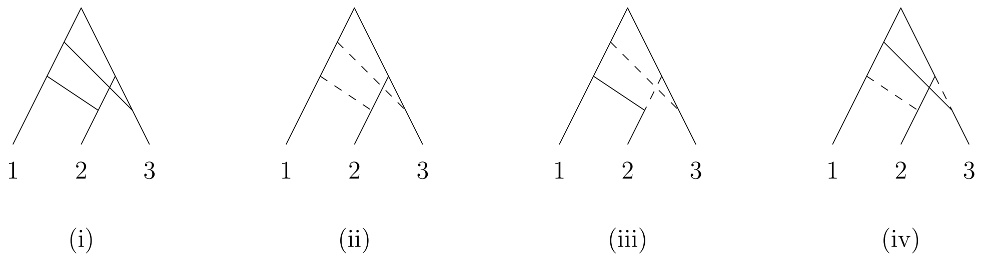


Figure 1.5: (i) A tree-based phylogenetic network N on $X = \{1,2,3\}$. (ii)-(iv) Three distinct phylogenetic trees on $X = \{1,2,3\}$ to which new arcs are added (dashed) so that the resulting phylogenetic network is N.

$\{1,2,3\}$ depicted in Figure 1.5(i), all three binary phylogenetic trees on $\{1,2,3\}$ are a support tree. This observation gives rise to the following notion. We say that a tree-based network N on X is a *universal tree-based network* on X if all binary phylogenetic trees on X are support-trees for N. In [34] a method to build, for any $n \geq 3$, a universal tree-based phylogenetic network on n leaves is presented. Moreover, the way the method works also establishes that there are infinitely many such networks for a given number of leaves. Since these networks may be overly complicated, the authors of [7] propose a method to build such a network with a minimal number of hybrid vertices.

The definition of a tree-based network has been extended in [45] to the nonbinary case. This is done by dropping the requirement for the support tree T to be binary, and by replacing step (b) of the construction by:

(b') Define new arcs between existing vertices, avoiding the creation of directed cycles.

In particular, new arcs may be incident to vertices of T, rather than being only allowed to be incident to vertices created in step (a).

1.2 Distances and splits

In this section, we turn our attention to metrics on the set X, and some more general extensions of this notion.

1.2.1 Trees and distances

Formally speaking, a *metric* on a set Z is a function $d : Z^2 \rightarrow \mathbb{R}_{\geq 0}$ satisfying:

(M1) For any $x, y \in Z$, we have $d(x, y) = 0$ if and only if $x = y$ (Identity and separation).

(M2) For any $x, y \in Z$, we have $d(x, y) = d(y, x)$ (Symmetry).

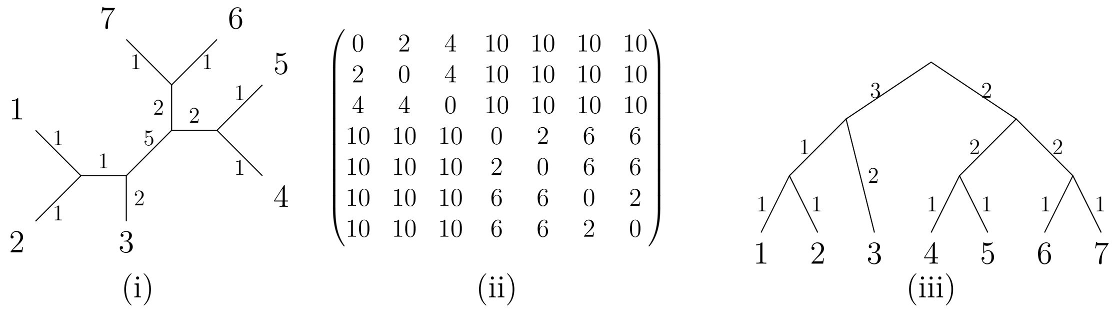
(M3) For any $x, y, z \in Z$, we have $d(x, z) \leq d(x, y) + d(y, z)$ (Triangle inequality).

If instead of (M1) d satisfies the weakened version (M1)' given by:

(M1)' For all $x \in Z$, we have $d(x, x) = 0$,

then d is said to be a *pseudo-metric*. Note that we shall sometimes use the word "distance" for metrics or pseudo-metrics, although strictly speaking, a distance does not need to satisfy Properties (M1) and (M3).

The use of metrics in a phylogenetic context is motivated by the observation that any unrooted weighted phylogenetic tree $\mathcal{T} = (T, \omega)$ on X trivially induces a distance $d_{\mathcal{T}}$ on its leaf set X by taking, for each pair $x, y \in X$, the distance $d_{\mathcal{T}}(x, y)$ between x and y to be the length of the (unique) path between x and y in T. This distance is known as the *phyletic distance* of $\mathcal{T}$ (see e. g. [23]). For example, in the unrooted tree depicted in Figure 1.6(i), the distance between the leaves 1 and 3 is $1+1+2 = 4$, and the distance between the leaves 2 and 5 is $1+1+5+2+1 = 10$. Thus, it is of interest to understand under which conditions a given distance d on X can be represented by a weighted phylogenetic tree $\mathcal{T}$ on X.



$$\begin{pmatrix} 0 & 2 & 4 & 10 & 10 & 10 & 10 \\ 2 & 0 & 4 & 10 & 10 & 10 & 10 \\ 4 & 4 & 0 & 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 0 & 2 & 6 & 6 \\ 10 & 10 & 10 & 2 & 0 & 6 & 6 \\ 10 & 10 & 10 & 6 & 6 & 0 & 2 \\ 10 & 10 & 10 & 6 & 6 & 2 & 0 \end{pmatrix}$$

(i) (ii) (iii)

Figure 1.6: (i) An unrooted weighted phylogenetic tree $\mathcal{T} = (T, \omega)$ on $X = \{1, \ldots, 7\}$. (ii) The distance $d = d_{\mathcal{T}}$ induced by $\mathcal{T}$, presented in terms of a distance-matrix $(d(i, j))_{i,j \in X}$. (iii) An ultrametric tree inducing the same distance d on X.

A metric d for which there exists a weighted phylogenetic tree $\mathcal{T}$ satisfying $d = d_{\mathcal{T}}$ is said to be *tree-like*. As is easy to see, any four leaves x, y, z and u of the phylogenetic tree T depicted in Figure 1.2(i) satisfy the following inequality, known as the *four-point condition*:

$$d_{\mathcal{T}}(x, y) + d_{\mathcal{T}}(z, u) \leq \max\{d_{\mathcal{T}}(x, z) + d_{\mathcal{T}}(y, u), d_{\mathcal{T}}(x, u) + d_{\mathcal{T}}(y, z)\}.$$

As it turns out, this definition is at the heart of a characterization of tree-like metrics:

Theorem 1.2.1 ([10])**.** *Let d be a metric on X. Then d is tree-like if and only if d satisfies the four-point condition.*

As is easy to see, the distance $d_{\mathcal{T}}$ induced by a rooted, weighted tree $\mathcal{T} = (T, \omega)$ is independent of the location of the root in T. More precisely, for $(U(T), \omega_0)$ the underlying weighted tree of $\mathcal{T}$ we have that $d_{\mathcal{T}} = d_{(U(T),\omega_0)}$.

However, a particular type of weighted tree allows for getting around this non-uniqueness problem. We say that a weighted, rooted tree (T, ω) is an *ultrametric tree* if the length of the path from the root of T to a leaf is the same for all leaves of T. For example, the tree depicted in Figure 1.6(iii) is an ultrametric tree, since the length of the path from its root to any of its leaves is 5. This gives rise to the following definition. We call a metric d on X an *ultrametric* on X if there exists an ultrametric tree $\mathcal{T} = (T, \omega)$ on X such that $d = d_{\mathcal{T}}$. As in the case of tree-like metrics, ultrametrics can be characterized in term of an inequality. We say that a metric d on X satisfies the *three-point condition* if the following holds for all $x, y, z \in X$:

$$d(x, y) \leq \max\{d(x, z), d(y, z)\}.$$

In other words, the two larger distances between two of three leaves x, y and z are equal. Clearly, a metric satisfying the three-point condition also satisfies the four-point condition, and thus, is tree-like, whereas the converse is not true in general. Indeed, the distance $d_{\mathcal{T}}$ induced by the unrooted phylogenetic tree $\mathcal{T}$ depicted in Figure 1.6(i) where all edges are given weight 1 is tree-like, but does not satisfy the three-point condition, as $\max\{d_{\mathcal{T}}(1, 3), d_{\mathcal{T}}(3, 4)\} = 4 \leq d_{\mathcal{T}}(1, 4) = 5$.

This property provides the rooted analogue of Theorem 1.2.1:

Theorem 1.2.2 ([55])**.** *Let d be a metric on X. Then d is an ultrametric on X if and only if d satisfies the three-point condition.*

Perhaps not surprisingly, various algorithm have been developed to recover a weighted phylogenetic tree (T, ω) from a metric. These include the Unweighted Pair Group Method using Arithmetic averages (known as UPGMA, see [56]) for ultrametrics, and the Neighbor-Joining (NJ, [53]), for metrics in general. Both methods take as input a distance d on X and are based on an *agglomerative* process. Basically, this means that starting with n clusters, each of them containing a single element of X, the two clusters that are, in some sense, the closest to each other are merged, until a single cluster containing all elements of X is obtained. The distance between clusters is derived from the distance given on X, and is updated at each step, to reflect the merging of clusters.

1.2.2 The Split Equivalence Theorem

The fact that the path between two leaves of a phylogenetic network may not be unique makes it difficult to extend this approach beyond phylogenetic trees. However, tree-like distances enjoy a property that turns out to be useful in that respect. To be able to state that property, we next introduce the notion of a "split" of X.

A *split* $S = \{A, B\}$ on X is a *bipartition* of X into two non-empty subsets $A, B \subsetneq X$, that is, $A \cup B = X$ and $A \cap B = \emptyset$. We write $S = A|B$, or $S = A|\overline{A}$. Note that the role of the two sets that make up a split is symmetric, so we have $A|B = B|A$. If $A = \{x_1, \dots, x_k\}$ and $B = \{x_{k+1}, \dots, x_n\}$ for some $1 \leq k \leq n-1$ we usually write $x_1 \dots x_k|x_{k+1} \dots x_n$ rather than $\{x_1, \dots, x_k\}|\{x_{k+1}, \dots, x_n\}$.

We denote by $\Sigma(X) = \{A|\overline{A} : A \subsetneq X, A \neq \emptyset\}$ the set of all splits of X, and we call a subset Σ of $\Sigma(X)$ a *split system* on X. Note that $|\Sigma(X)| = 2^{|X|-1} - 1$. Finally, for $S = A|B$ a split on X and $x \in X$, we define the set $S(x)$ of S as:

$$S(x) = \begin{cases} A \text{ if } x \in A \\ B \text{ if } x \in B \end{cases}$$

There exists a direct link between the notion of a split and an unrooted phylogenetic tree. Indeed, a split $S = A|B$ of X is said to be *displayed* by an unrooted phylogenetic tree T on X if there exists an edge e of T whose deletion disconnects T into two connected components whose set of leaves are respectively A and B. Denoting the split induced by an edge e of T by S_e, we call the set $\Sigma(T) = \{S_e : e \in E(T)\}$ the *split system* induced by T. We say that a split system Σ on X is *displayed* by T if $\Sigma \subseteq \Sigma(T)$, and that it is *represented* by T if $\Sigma = \Sigma(T)$. For example, the split system

$$\Sigma = \{1|2345, 2|1345, 12|345, 123|45, 3|1245, 4|1235, 5|1234\}.$$

on $X = \{1, 2, 3, 4, 5\}$ is the split system represented by the phylogenetic tree T depicted in Figure 1.7.

Note that if T is an unrooted phylogenetic tree, then for all elements $x \in X$, the deletion of the edge of T incident to x induces the split $S_x = \{x\}|X - \{x\}$. Such splits are called *trivial splits* of X.

The notion of a split system can be extended to a *weighted split system* (Σ, α), by including a map $\alpha : \Sigma \to \mathbb{R}_{\geq 0}$. Moreover, a split S on X trivially induces a pseudo-metric $\delta_S : X \times X \to \{0, 1\}$ by putting, for $x, y \in X$, $\delta_S(x, y) = 0$ if $S(x) = S(y)$, and $\delta_S(x, y) = 1$ otherwise. Thus, a weighted split system (Σ, α) on X induces a pseudo-metric $d_{(\Sigma,\alpha)}$ given by:

$$\begin{array}{rcl} d_{(\Sigma,\alpha)} : X^2 & \to & \mathbb{R}_{\geq 0} \\ (x, y) & \mapsto & \displaystyle\sum_{S \in \Sigma} \alpha(S)\delta_S(x, y). \end{array}$$

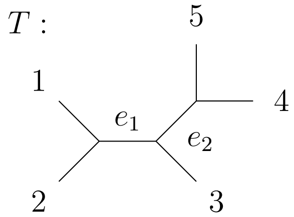


Figure 1.7: A phylogenetic tree on $X = \{1,2,3,4,5\}$. The edges e_1 and e_2 respectively induce the splits $12|345$ and $123|45$.

That $d_{(\Sigma,\alpha)}$ satisfies Properties (M1') and (M2) of a pseudo-metric is straightforward to observe. To see that $d_{(\Sigma,\alpha)}$ also satisfies the triangle inequality (M3), it suffices to observe that for any split $S \in \Sigma(X)$ and any three elements $x, y, z \in X$, we have $\delta_S(x,z) \leq \delta_S(x,y) + \delta_S(y,z)$. Note also that $d_{(\Sigma,\alpha)}$ is a metric if and only if for all distinct $x, y \in X$, there exists a split $S \in \Sigma$ such that $S(x) \neq S(y)$. As is easy to see, if (T,ω) is a weighted unrooted phylogenetic tree, and $(\Sigma(T),\alpha)$ is such that for all edge e of T, we have $\alpha(S_e) = \omega(e)$, then the distance $d_{(\Sigma(T),\alpha)}$ coincides with the phyletic distance $d_{(T,\omega)}$.

In [3], it has been shown that for all metrics d on X, there exists a weighted split system (Σ_d, α_d) and a *residual term* $d_0 \in \mathbb{R}$ such that $d = d_{(\Sigma_d,\alpha_d)} + d_0$. Referring to that expression as a *decomposition* of d, we say that d is *totally-decomposable* if it admits a decomposition with residual term $d_0 = 0$, that is, if d of the type $d_{(\Sigma,\alpha)}$ for a weighted split system (Σ,α). Such metrics have been introduced and studied in [3] (see also [14] for more about metrics decomposition).

Note that two distinct weighted split systems (Σ,α) and (Σ',α') may induce the same distance, that is, $d_{(\Sigma,\alpha)} = d_{(\Sigma',\alpha')}$. This is the case, for example, for the splits systems (Σ_1,α_1) and (Σ_2,α_2) on $X = \{1,2,3,4\}$, where $\Sigma_1 = \{12|34, 13|24, 14|23\}$, $\Sigma_2 = \{1|234, 2|341, 3|412, 4|123\}$, and α_i assigns weight one to all splits in Σ_i, $i \in \{1,2\}$. The distance d induced by both these weighted split systems is given, for $x, y \in X$, by:

$$d(x,y) = \begin{cases} 0 \text{ if } x = y \\ 2 \text{ if } x \neq y \end{cases}$$

The natural question arising from these observations is the following: Given a non-empty split system $\Sigma \subseteq \Sigma(X)$, under which conditions does there exist an unrooted phylogenetic tree T on X representing Σ? Due to the above observation, it is not possible to consider directly the induced distance d_Σ. Indeed consider again the distance $d_{(\Sigma_1,\alpha_1)}$ induced by the split system (Σ_1,α_1). Although $d_{(\Sigma_1,\alpha_1)}$ satisfies the four-point condition reviewed in Section 1.2.1, the phylogenetic tree

T representing $d_{(\Sigma_1,\alpha_1)}$ does not satisfy $\Sigma(T) = \Sigma_1$ (in fact, we have $\Sigma(T) = \Sigma_2$). However, a direct answer to that question, known as the *Split Equivalence Theorem*, was given in [10].

To state this theorem, we require a further notion for splits. We say that two distinct splits $S_1 = A_1|B_1$ and $S_2 = A_2|B_2$ on X are *compatible* if one of the four following intersections:

$$A_1 \cap A_2 \ ; \ A_1 \cap B_2 \ ; \ B_1 \cap A_2 \ ; \ B_1 \cap B_2$$

is empty. Equivalently, two distinct splits S_1 and S_2 on X are compatible if and only if there exist $A_1 \in S_1$ and $A_2 \in S_2$ such that $A_1 \subset A_2$. More generally, a split system Σ is said to be *compatible* if any two distinct splits in Σ are compatible. If moreover, there exists no split $S \in \Sigma(X) - \Sigma$ such that $\Sigma \cup \{S\}$ is compatible, then Σ is said to be *maximal compatible.* The Split Equivalence Theorem can then be stated as follows:

Theorem 1.2.3 ([10]). *Let Σ be a split system on a set X that contains all trivial splits on X. Then, there exists a (unique) phylogenetic tree T such that $\Sigma = \Sigma(T)$ if and only if Σ is compatible. Moreover, T is binary if and only if Σ is maximal compatible.*

To obtain the phylogenetic tree postulated in Theorem 1.2.3, Meacham's Tree Popping approach ([50]) may be used. This method works as follows:

Starting with T initialized as a *star tree*, that is, the unrooted phylogenetic tree on X whose n leaves are joined to a unique interior vertex, the method proceeds in the form a succession of vertex expansion operations (see Section 1.1). While $\Sigma(T) \neq \Sigma$, we consider a split $S = A|B \in \Sigma - \Sigma(T)$, and the necessarily unique interior vertex v of T such that all the connected components obtained from T by deletion of v have their set of leaves contained in either A or B. Then, we expand v to a new edge $\{v^A, v^B\}$ in the unique way such that $S_{\{v^A,v^B\}} = S$. Figure 1.8 illustrates this process for the split system represented by the phylogenetic tree T depicted in Figure 1.7.

1.2.3 Split-networks

As mentioned before, tree-like metrics are totally-decomposable, but the converse is true only if the metric admits a decomposition in term of a compatible split system. A straightforward consequence of Theorem 1.2.3 is that a totally-decomposable metric admits at most one such decomposition. The authors of [3] extend this property, by introducing the notion of weak compatibility as follows: For S_1, S_2 and S_3 three distinct split on X, we say that $S_1 = A_1|B_1$, $S_2 = A_2|B_2$ and $S_3 = A_3|B_3$ are *weakly compatible* if $A_1 \cap A_2 \cap A_3 = \emptyset$ or $A_i \subseteq A_j$ holds for

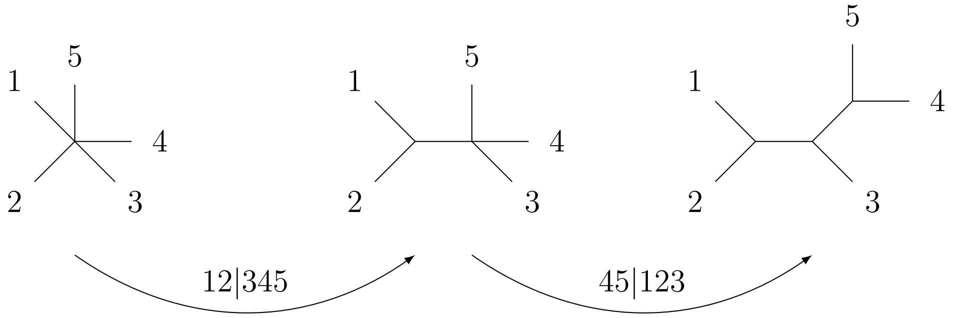


Figure 1.8: Building the phylogenetic tree T depicted in Figure 1.7 from the split system $\Sigma(T)$. See text for the description of Meacham's Tree Popping approach.

some distinct $i, j \in \{1, 2, 3\}$. As in the case of compatibility, we say that a split system Σ on X is *weakly compatible* if any three splits in Σ are weakly compatible. Clearly, if S_1 and S_2 are two compatible splits on X, then for any split S_3 on X, the splits S_1, S_2 and S_3 are weakly compatible. Thus, a compatible split system is also weakly compatible. Moreover, we have:

Theorem 1.2.4 ([3]). *Let d be a totally-decomposable metric on a set X. Then there exists a unique weighted split system (Σ, α) satisfying $d = d_{(\Sigma,\alpha)}$ such that Σ is weakly compatible.*

Note that Theorem 1.2.4 allows us to consider the question of the representability of a totally-decomposable metric d that is not tree-like by considering the unique weighted split system $\mathcal{S}_d$ it induces, rather than the metric d itself. We next review popular approaches that have been developed in order to deal with such split systems. We group these approaches by two main ideas.

Split-graphs. Consider a graph G and a map σ from the edge set $E(G)$ of G to a finite set C of *colors*. Such a map is said to be *isometric* if for any pair of vertices u and v of G, all edges on a shortest path between u and v have a different color, and the set of such colors is the same for all shortest paths between u and v. Following [20], we formally define a *split-graph* (G, σ) to be a *bipartite* undirected connected graph G, that is, a graph with no cycle of odd length, together with an isometric labelling $\sigma : E(G) \to C$ of its edges. The key property of such graphs is the following:

Theorem 1.2.5. *Let (G, σ) be a split graph, and let C denote the set of values taken by σ on $E(G)$. For all $c \in C$, the graph G' obtained from G by deletion of all edges e in $E(G)$ satisfying $\sigma(e) = c$ consists of exactly two connected components.*

A *split network* N on X is a split graph (G, σ) some of whose vertices are labelled by elements of X. Thus, any color $c \in C$ directly corresponds to a split S_c of X, that is, the split obtained by removing all edges e of G satisfying $\sigma(e) = c$. The split system $\Sigma(N) = \{S_c : c \in C\}$ is called the split system displayed by N. In the context of split networks, the set of edges sharing the same color is more important than the color itself so, in general, the coloring of the edges is not explicitly mentioned. Whenever possible, we draw split networks in such a way that edges associated to a same split are parallel (see Figure 1.9), so that they can easily be identified.

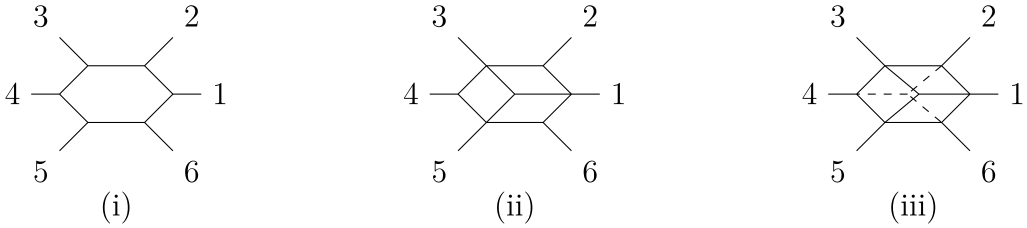


Figure 1.9: Three distinct split networks on $X = \{1, 2, 3, 4, 5, 6\}$ representing the same split system $\Sigma = \{123|456, 234|561, 345|612\} \cup \{x|X - \{x\} : x \in X\}$. The networks in (i) and (ii) are outerplanar, and the network in (iii) is the Buneman graph $\mathcal{B}(\Sigma)$ of Σ (see Section 4.3.1 for more on this particular type of split network).

Note that a split network is not a phylogenetic network in the sense of Section 1.1.2, as non-leaf vertices may be labelled by elements of X, and two or more elements of X may label the same vertex. In fact, a split network N is a phylogenetic network in the sense of Section 1.1.2 if and only if the split system $\Sigma(N)$ contains all trivial splits on X. Similarly, N is a phylogenetic tree if and only if $\Sigma(N)$ is compatible and contains all trivial splits on X, in which case the notions of displaying a split or a split system boil down to the definition of the respective concepts introduced in Section 1.2.2.

From a combinatorial point of view, split networks are interesting since for any split system Σ on X, there exists a network N such that $\Sigma = \Sigma(N)$. However, as suggested by Figure 1.9, such a network is in general non-unique. Among the several split-networks representing a given split system Σ, *Buneman graphs* are of particular interest, due to their attracive combinatorial properties as well as their links with other mathematical structures (see e. g. [16] and Section 4.3.1 for some of them).

Splits networks also allow one to take into account the weight of splits. Indeed, if (Σ, α) is a weighted split system, we can consider the weighted network

(N, ω), where $\Sigma = \Sigma(N)$ and ω assigns to each edge e of N the weight $\alpha(S)$ of the split S it corresponds to. By definition, any shortest path between two elements x, y of X in N is in bijection with a subset $\Sigma_{x,y}$ of $\Sigma(N)$, which is precisely the set of splits S of Σ satisfying $S(x) \neq S(y)$ (or equivalently, $\delta_S(x, y) = 1$). Thus, we have that the distance $d_{(N,\omega)}$ defined by assigning to all pairs $x, y \in X$ the length of a shortest path between x and y in N is precisely the distance $d_{(\Sigma,\alpha)}$. In particular, that distance does not depend on the choice of N. Consequently, any totally-decomposable distance can be represented by a split-network.

Outerplanar networks. In the study of split systems, circular ones play a special role due to the properties enjoyed by their split-network representation. Following [3], we say that a split system Σ on X is *circular* if there exists a circular ordering $x_0, x_1, \ldots x_n = x_0$ of the elements of X such that for all split $S = A|B$ in Σ, A and B are *intervals* for that ordering, that is, sets of consecutive elements of X. As an example, the split system Σ defined in Figure 1.9 is circular, for the lexicographical ordering on $X = \{1, \ldots, 6\}$.

Note that a circular split system is weakly compatible (although the converse is not true in general), but is not necessarily compatible. Moreover, circular split systems are linked to a particular type of network, called outerplanar. A network, whether it is a split-network or a phylogenetic network, is said to be *outerplanar* if it can be drawn in the Euclidian plane in such a way that:

(O1) No two edges cross.

(O2) The leaves lie *outside* the network.

A network satisfying Property (O1) only is called *planar*. As an example, phylogenetic trees and level-1 networks are both outerplanar. The link between outerplanar networks and circular split systems was highlighted in [20] in term of the following result:

Theorem 1.2.6 ([20])**.** *Let Σ be a split system on a set X. There exists an outerplanar split network N such that $\Sigma(N) = \Sigma$ if and only if Σ is circular.*

Theorem 1.2.6 is attractive, as it guarantees the existence of a structurally simple split-network representing a circular split system Σ. The question how such a split network can be built from a circular split system Σ was answered in [9]. The authors of that paper propose a method, called Neighbor-Net, which is an adaptation of the Neighbor-Joining algorithm for distances (described in Section 1.2.1). Indeed, Neighbor-Net does not take as input a split system but a distance d on X. If d is totally-decomposable and if the underlying split system $\mathcal{S}_d$ of d is circular, then the algorithm builds an outerplanar split-network (N, ω)

such that $d = d_{(N,\omega)}$ in the sense defined above.

Minimal cuts. As it turns out, minimal cuts of graphs have recently been introduced as an alternative way to display split systems in terms of an unrooted phylogenetic network ([3, 27]). If N is an unrooted phylogenetic network on X, a *minimal cut* of N is the deletion of a set-inclusion minimal set E_0 of edge of N that disconnects N. Since a minimal cut induces a split of X, a split system $\Sigma(N)$ can be defined by considering all minimal cuts of N. As in the case of split networks, we have that if N is a tree, then $\Sigma(N)$ is the split system induced by N in the sense defined in Section 1.2.2.

Although the minimal cut approach is less well understood than the split-network one, some attractive properties hold in case the split system Σ considered is circular. As a first result, we have:

Theorem 1.2.7 ([27])**.** *Let Σ be a split system on a set X. There exists a level-1 network N on X such that $\Sigma \subseteq \Sigma(N)$ if and only if Σ is circular.*

As suggested by Theorem 1.2.7, for a given split system Σ on X, there exists in general no phylogenetic network N such that $\Sigma = \Sigma(N)$. Thus, it is of interest to characterize those split systems for which this equality holds. In case of level-1 network, such a characterization is given in [8]. This characterization is closely related to the notion of incompatibility, and uses the concept of an "intersection" between two split. For two distinct splits $S_1 = A_1|B_1$ and $S_2 = A_2|B_2$ of X such that $A_1 \cap A_2$ is non empty, we say that the split $A_1 \cap A_2|B_1 \cup B_2$ is an *intersection* between S_1 and S_2.

As is easy to see, two distinct splits S_1 and S_2 on X can have up to four intersections. More precisely, S_1 and S_2 have three intersections if they are compatible, two of which being S_1 and S_2 themselves, and four distinct intersections if they are incompatible. Split systems that can be represented by a level-1 network were characterized in [8] as follows:

Theorem 1.2.8 ([8])**.** *Let Σ be a split system on a set X containing all trivial splits. There exists a level-1 network N on X satisfying $\Sigma(N) = \Sigma$ if and only if for any pair of incompatible splits $S_1 = A_1|B_1$ and $S_2 = A_2|B_2$ in Σ, the four intersections between S_1 and S_2 belong to Σ, whereas the split $(A_1 \cap A_2) \cup (B_1 \cap B_2)|(A_1 \cap B_2) \cup (B_1 \cap A_2)$ does not.*

In [29], which forms the basis of Chapter 4, we revisit Theorems 1.2.7 and 1.2.8. In addition, we highlight a link between the split-network approach and the minimal cut approach in case the split system Σ is circular.

1.3 Beyond metrics

In Section 1.2.1, a metric d is defined as a map with domain a set Z^2 and range in $\mathbb{R}_{\geq 0}$. Both these requirements can be weakened, independently from each other, thus leading to generalizations of the notion of a metric.

1.3.1 Symbolic distances

In the study of evolution, internal vertices of a phylogenetic network may be seen as evolutionary events. For a rooted phylogenetic tree T on X and two taxa x and y in X, we can consider them to have arisen from a hypothetical last common ancestor in T, until an evolutionary event leads them to follow different paths. For example, for a family $\mathcal{G}$ of genes, such an event can be a *speciation*, that is, a mutation that leads over time to distinct species, or a *genome duplication*. The study of such relations is known as *gene homology*. In particular, two genes of $\mathcal{G}$ are said to be *orthologs* if they have arisen from a common ancestor through a speciation event, and *paralogs* if they have been separated over time by a duplication event. Recently, ortholog/paralogs detection have been a topic of interest to biologists, and methods have been developed to infer, for a pair of genes, whether they are involved in an orthology or a paralogy relation (see e. g. [47] for an overview). Some of these methods rely on sets of more than two genes to infer this information, which, as we shall see, will turn out to be of interest to us. This is the case, for example, in [57], where the notion of a *cluster of orthologous genes* is introduced. Others do not restrict to speciation and duplication events only, but consider a wider set of events (see e. g. [51]). The existence of such methods motivates the introduction of maps derived from distances but which do not necessarily take numerical values.

If T is a rooted phylogenetic tree on X such that for each of its internal vertices, the evolutionary event it corresponds to is known, we can associate to each pair of elements of X the events corresponding to their last common ancestor in T. More formally, suppose T is a rooted phylogenetic tree on X. Let $V_{int}(T)$ denote the set of internal vertices of T, and let M be a nonempty set that contains all permissible evolutionary events. Then, a map $t : V_{int}(T) \to M$ induces a map $\delta_{(T,t)} : X^2 \to M \cup \{\odot\}$ defined as follows (see Figure 1.10 for an example):

$$\begin{array}{rcl} \delta_{(T,t)} : X^2 & \to & M \cup \{\odot\} \\ (x,y) & \mapsto & \begin{cases} t(lca_T(x,y)) & \text{if } x \neq y \\ \odot & \text{if } x = y \end{cases} \end{array} \tag{1.1}$$

The symbol $\odot$ is not directly related to an evolutionary event, but is necessary as a sort of “neutral element” for technical reasons. We refer to the pair (T,t) as a *labelled tree*, and to the map $\delta_{(T,t)}$ as the map *represented* by that labelled tree.

Moreover, a labelled tree (T, t) is said to be *discriminating* if for all arcs (u, v) of T such that v is not a leaf, we have $t(u) \neq t(v)$.

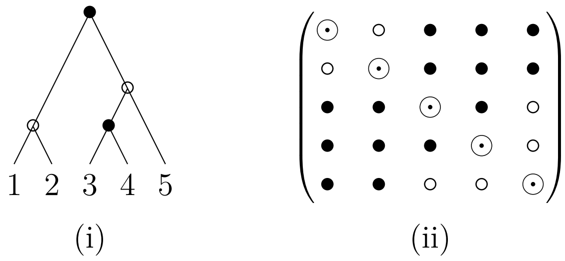


Figure 1.10: (i) A rooted phylogenetic tree T on $X = \{1, 2, 3, 4, 5\}$ together with a labelling map t of its internal vertices in terms of the set $\{\bullet, \circ\}$. (ii) The map δ represented by (T, t), presented in terms of a symbolic distance-matrix $\mathcal{M} = (\delta(x, y))_{x,y \in X}$ on X.

Maps such as the one in Expression 1.1 are known as symbolic distances. Formally, a *symbolic distance* on X is a map $\delta : X^2 \to M$, where M is a set of size at least two, satisfying the following two properties:

(S1) There exists an element $\odot \in M$ such that for all $x, y \in X$, we have $\delta(x, y) = \odot$ if and only if $x = y$.

(S2) For any $x, y \in X$, we have $\delta(x, y) = \delta(y, x)$.

Clearly, Properties (S1) and (S2) are natural analogues of Properties (M1) and (M2) respectively for (real-valued) metrics, in the sense that they are equivalent if Y is the set $\mathbb{R}_{\geq 0}$ (taking the neutral element $\odot$ to be 0). However, note that if $M = \mathbb{R}_{\geq 0}$, then δ is not necessarily a distance in the usual sense, as it need not satisfy the triangle inequality.

In order to better understand the link between labelled trees and symbolic distances, the notion of a symbolic ultrametric was introduced in [6]. A *symbolic ultrametric* on X is a symbolic distance on X satisfying the following two additional properties:

(U1) For any three elements x, y and z in X, at least two of the three values $\delta(x, y)$, $\delta(x, z)$ and $\delta(y, z)$ coincide.

(U2) There exists no four elements x, y, z and u in X such that:

$$\delta(x, y) = \delta(y, z) = \delta(z, u) \neq \delta(y, u) = \delta(u, x) = \delta(x, z)$$

holds (see Figure 1.11(ii) for a graphical representation of that condition).

For example, the map represented by the symbolic distance-matrix in Figure 1.10(ii) is a symbolic ultrametric. Note that (U1) may be viewed as the natural analogue of the three-point condition (see Section 1.2.1) for ultrametrics. More precisely, an ultrametric $d : X^2 \to \mathbb{R}_{\geq 0}$ satisfies (U1) and (U2), and thus, considering numbers as symbols, can be seen as a symbolic ultrametric. The converse is not true in general, as a symbolic ultrametric with image set in $\mathbb{R}_{\geq 0}$ need not be an ultrametric, for two reasons. First, as in the case of symbolic distances mentioned above, such a map does not necessarily satisfy the triangle inequality. Second, the three point conditions requires, in addition to the correspondence postulated by (U1), an inequality to be satisfied, whereas no order is required on the image set of a symbolic ultrametric.

Symbolic ultrametrics were characterized in [6] as follows:

Theorem 1.3.1 ([6])**.** *Let X and M be two sets of size 2 or more, and let $\delta : X^2 \to M$ be a map. There exists a rooted phylogenetic tree T on X together with a map $t : V_{int}(T) \to M$ such that $\delta = \delta_{(T,t)}$ if and only if δ is a symbolic ultrametric.*

Note that for a given symbolic ultrametric δ, the labelled tree (T, t) satisfying $\delta_{(T,t)} = \delta$, is not necessarily unique. However, the following holds:

Theorem 1.3.2 ([6])**.** *Let X and M be two sets of size 2 or more. If $\delta : X^2 \to M$ is a symbolic ultrametric, then there exists a unique discriminating labelled tree (T, t) satisfying $\delta_{(T,t)} = \delta$.*

Interestingly, an equivalent of Theorem 1.3.1 and the associated uniqueness result appeared in [31] in the context of game theory (see also [32] for more details). Within this context, the leaves of the tree T are seen as end of game situations, the label set M corresponds to a set of players, and a directed path from the root of T to a leaf is a sequence of plays.

In [31], a symbolic map on X is seen as an edge-colored unrooted graph H_δ with vertex-set X and edge set $\{\{x, y\} \in X^2, x \neq y\}$, where the color associated to an edge $\{x, y\}$ corresponds to the value $\delta(x, y)$. This representation is interesting, as it allows one to vizualize conditions (U1) and (U2) in terms of two “forbidden patterns”, which we depict in Figure 1.11.

As is easy to see, a symbolic map $\delta : X^2 \to M$ is a symbolic ultrametric if and only if H_δ does not contain a subgraph isomorphic to either Δ or Π. More precisely, δ satisfies (U1) (*resp.* (U2)) if and only if H_δ does not contain a subgraph isomorphic to Δ (*resp.* Π).

An algorithm aimed at building the unique discriminating labelled tree (T, t) postulated by Theorem 1.3.2 from a given symbolic ultrametric δ on X is presented in [35]. Called Bottom-Up, this algorithm is agglomerative. In order to describe it, we first require some more terminology.

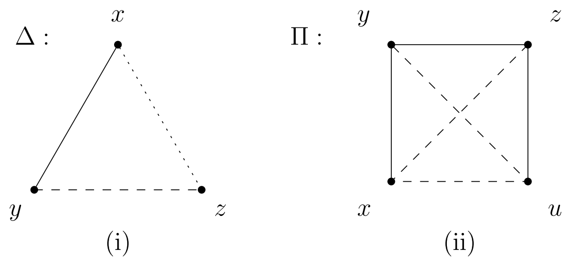


Figure 1.11: (i) An edge-colored graph Δ on $X = \{x, y, z\}$. (ii) An edge-colored graph Π on $X = \{x, y, z, u\}$. Colors are represented in terms of different edge styles (plain, dashed and dotted).

Suppose we have a map $\delta : X^2 \to M$. For $x \in X$ and $m \in M$, we put $N_m(x) = \{y \in X : \delta(x, y) = m\}$ and $N_m[x] = N_m(x) \cup \{x\}$. Armed with these notations, we define an undirected graph $G(\delta)$ as follows: The vertex set of $G(\delta)$ is X, and two vertices x and y are joined by an edge if there exists an element $m \in M$ such that $N_m[x] = N_m[y]$. Finally, we denote by $\pi(\delta)$ the set whose elements are the vertex sets of the connected components of $G(\delta)$, and by $\pi_2(\delta)$ the set of elements of $\pi(\delta)$ of size at least 2.

For example, for $X = \{1, 2, 3, 4, 5\}$ and $M = \{\odot, \circ, \bullet\}$, consider the map $\delta : X^2 \to M$ given by the distance-matrix in Figure 1.10(ii). We have $N_\circ[1] = \{1, 2\} = N_\circ[2]$ and $N_\bullet[3] = \{1, 2, 3, 4\} = N_\bullet[4]$. Then, the vertex set of the graph $G(\delta)$ is X and its edge set $\{\{1, 2\}, \{3, 4\}\}$. Consequently, $\pi(\delta) = \{\{1, 2\}, \{3, 4\}, \{5\}\}$ and $\pi_2(\delta) = \{\{1, 2\}, \{3, 4\}\}$.

The Bottom-Up algorithm starts with initializing T as $n = |X|$ isolated vertices, each of which labelled by an element of X. The main observation on which the algorithm is based is the fact that a connected component of size two or more in $G(\delta)$ corresponds to a *pseudo-cherry* in the tree we are looking for, that is, a set of two or more leaves sharing the same parent. This is an extension of the notion of a *cherry*, that is a pair of leaves sharing the same parent, to the non-binary case.

Suppose $Y \in \pi_2(\delta)$. Then, we define a new vertex v, and add to T the arcs (v, x) for all $x \in Y$. If $\pi_2(\delta)$ is empty, we return the statement that δ is not a symbolic ultrametric. Next, we define $t(v) = \delta(x, y)$, where the choice of x and y in Y is of no relevance since by definition of Y, δ has the same value on any pair of distinct elements of Y.

Once this is done for all $Y \in \pi_2(\delta)$, we update δ by identifying every connected component in the tree built thus far with one of its leaves. We then iterate that process until we obtain a labelled tree or the statement that δ is not a symbolic ultrametric. Note that in the case where a labelled tree (T, t) is returned, this tree represents δ but may not be discriminating. In that case, we collapse all internal arcs (u, v) of T for which $t(u) = t(v)$ holds into a new node w, and put $t(w) = t(v)$. As is easy to see, this operation does not modify the symbolic ultrametric represented by the resulting labelled tree.

In Chapter 2, which is based on [41], we propose an extension of Theorem 1.3.1 and of the Bottom-Up algorithm to the space of level-1 networks.

1.3.2 Dissimilarities

Rather than considering a distance between two elements, it may sometimes be of interest to consider an equivalent taking into account three elements or more. Indeed, such an equivalent may turn out to be more accurate than metrics, as they can potentially capture more information (see e. g. [24, 52]). To formalize this idea, let $1 \leq k \leq |X|$. We denote by $\binom{X}{k}$ the set of subsets of X of size exactly k, and by $\binom{X}{\leq k}$ the set of subsets of X of size k or less. We call a map $d : \binom{X}{k} \to \mathbb{R}_{\geq 0}$ a *k-way dissimilarity* (or a *k-dissimilarity* for short). For $\{x_1, \dots, x_k\} \in \binom{X}{k}$, we shall write $d(x_1, \dots, x_k)$ rather than $d(\{x_1, \dots, x_k\})$, where the order of the elements $x_1, \dots, x_k$ is of no relevance. Clearly, a distance is a particular type of 2-dissimilarity. We remark in passing that, since the symmetry required by Property (M2) trivially holds for 2-dissimilarities, a metric can alternatively be defined as a 2-dissimilarity d with image in $\mathbb{R}_{>0}$, satisfying the triangle inequality (M3).

The following relationship between phylogenetic trees and k-dissimilarities was observed in [52]. For (T, ω) a weighted phylogenetic tree on X (rooted or unrooted), and $Y \in \binom{X}{k}$, let (T_Y, ω_Y) denote the weighted subtree induced by Y and let $d^k_{(T,\omega)}(Y)$ denote the sum of the length of all edges of (T_Y, ω_Y). For example, if (T, ω) is the phylogenetic tree depicted in Figure 1.6(i), then $d^3_{(T,\omega)}(1, 2, 3) = 5$, and $d^4_{(T,\omega)}(4, 5, 6, 7) = 8$. Clearly, for $k = 2$, the map $d^2_{(T,\omega)}$ obtained this way coincides with the phyletic distance $d_{(T,\omega)}$ defined in Section 1.2.1, as the subtree induced by two leaves x, y of T is precisely the (unique) path between x and y in T. As in the case of metrics (Section 1.2.1), we call a k-dissimilarity d on X *tree-like* if there exists a weighted tree (T, ω) on X such that $d = d^k_{(T,\omega)}$. As a first result, we have:

Theorem 1.3.3 ([52]). *Let (T, ω) be an unrooted weighted phylogenetic tree on X and let $k \geq 2$. If $2k + 1 \leq |X|$, then (T, ω) is uniquely determined by the map $d^k_{(T,\omega)}$.*

This means that if d is a tree-like k-dissimilarity, the unrooted weighted phylogenetic tree (T, ω) satisfying $d = d^k_{(T,\omega)}$ is unique. To decide whether a given k-dissimilarity d on X is tree-like or not, the authors of [36] propose to look at the restrictions of d to some carefully chosen subsets of X. For d a k-dissimilarity on X and Y a subset of X of size at least k, we denote by $d|_Y$ the restriction of d to $\binom{Y}{k}$. Then, we have:

Theorem 1.3.4 ([36]). *Let d be a k-dissimilarity on X, with $2 \leq k \leq \frac{|X|}{2}$. Then, d is tree-like if and only if $d|_Y$ is tree-like for all subsets Y of X of size $2k$.*

In particular, Theorem 1.3.4 suggests that tree-like k-dissimilarities can be characterized using properties involving $2k$ elements or less. This is clearly the case for $k = 2$ as the two conditions for a 2-dissimilarity to be tree-like (see Section 1.2.1) are the triangle inequality (M3), involving three elements of X, and the four-point condition, which involves four elements. The authors of [36] also note that it is impossible to go below this limit of $2k$, as for all $k \geq 3$, it is possible to find a k-dissimilarity d on X that is not tree-like, but is such that $d|_Y$ is tree-like for all subsets Y of X of size $2k - 1$.

Interestingly, Theorem 1.3.4 has a rooted equivalent, involving ultrametric trees. If d is a tree-like k-dissimilarity such that there exists an ultrametric tree (T, ω) satisfying $d^k_{(T,\omega)} = d$, we call d *equidistant*. As shown in [36], Theorem 1.3.4 also holds when replacing "tree-like" by "equidistant":

Theorem 1.3.5 ([36]). *Let d be a k-dissimilarity on X, with $2 \leq k \leq \frac{|X|}{2}$. Then, d is equidistant if and only if $d|_Y$ is equidistant for all subsets Y of X of size $2k$.*

In Chapter 3, which is based on [39], we combine this approach with the result of the previous section on symbolic distances to study two different types of symbolic 3-dissimilarities, and the relationship they enjoy with labelled phylogenetic trees.

1.4 Decomposition into smaller structures

It may sometimes be useful to study the structure of a graph G in terms of smaller graphs induced by G. In the context of phylogenetics, the idea manifests itself in terms of trying to decompose a phylogenetic network N into smaller networks, in such a way that such networks contain enough information to allow one to recover N. In the case of rooted phylogenetic networks, the simplest meaningful such graphs induced by a network are triplets and trinets, both of which we define next.

1.4.1 Triplets

Triplets may be thought of as the fundamental building blocks for rooted phylogenetic networks. For x, y and z three distinct elements of X, a *triplet* τ on $\{x, y, z\}$

is a binary rooted phylogenetic tree on $\{x, y, z\}$. If, say, z is a child of the root of τ whereas x and y are not (as is the case of the triplet τ_2 in Figure 1.14), we write $\tau = xy|z$ (or, equivalently, $z|xy$) to capture the structure of the triplet. A rooted phylogenetic network N on X is said to *display* a triplet $\tau = xy|z$ if there exists a vertex v_0 of N and two disjoints directed paths respectively from v_0 to z and from v_0 to a lowest common ancestor of $\{x, y\}$. More generally, a rooted phylogenetic network N on X is said to *display* a set of triplets C on X if N displays all triplets in C.

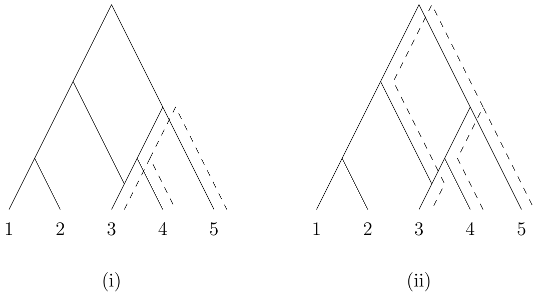


Figure 1.12: The leaves 3,4 and 5 of the rooted phylogenetic network N on $X = \{1, 2, 3, 4, 5\}$ depicted in Figure 1.3(iii) induce both the triplets $34|5$ ((i), dashed) and $45|3$ ((ii), dashed).

We denote by $C(N)$ the set of triplets displayed by a rooted phylogenetic network N on X. Note that three elements $\{x, y, z\} \subseteq X$ may be the leaf set of more than one triplet in $C(N)$ (see Figure 1.12 for an example), although this is impossible if N is a phylogenetic tree. In the same way, there might be subsets $\{x, y, z\} \subseteq X$, such that there exists no triplet $\tau \in C(N)$ with leaf set $\{x, y, z\}$. However, there are no such sets if N is binary. A collection C of triplets on X satisfying the property that for all distinct $x, y, z \in X$, there exists at least one triplet τ in C whose leaf set is $\{x, y, z\}$ is said to be *dense*[1]. As we shall see, this property turns out to be helpful for the purpose of reconstructing phylogenetic networks from triplets.

A method to build a rooted phylogenetic tree T *displaying* a given set C of triplets on X, that is, a phylogenetic tree T satisfying $C \subseteq C(T)$, is presented in [1]. Called BUILD, it takes as input a collection C of triplets on X, such that no two triplets in C have the same set of leaves. The algorithm starts by initializing T as a single vertex ρ, to which a set $S(\rho) = X$ is associated. Essentially, the methode proceeds by successively adding children to a vertex v of the tree T thus far constructed.

[1] Note that this use of the word "dense", which appears in [44] and has been widely used since (see e. g.[43, 58, 59]), bears no link with the well-known notion of density in topology.

For every vertex v of T of out-degree 0, we consider the associated set $S(v) \subseteq X$. If $|S(v)| = 2$, we call x_1 and x_2 its two elements. We then add two children x_1 and x_2 to v. If $|S(v)| \geq 3$, we consider the *Aho graph* $\pi(v)$, whose vertices are the elements in $S(v)$, in which two vertices x and y in $\pi(v)$ are joined by an edge if there exists a further element $z \in S(v)$ such that the triplet $xy|z$ belongs to C. If $\pi(v)$ consists of $m \geq 2$ connected components $\pi_1, \ldots, \pi_m$, the algorithm adds m children $v_1, \ldots, v_m$ to v, and defines, for all $1 \leq i \leq m$, the set $S(v_i)$ as the vertex set of π_i. This process is illustrated in Figure 1.13 for the triplet set $C = \{12|4, 45|3, 35|1\}$ on $X = \{1, 2, 3, 4, 5\}$.

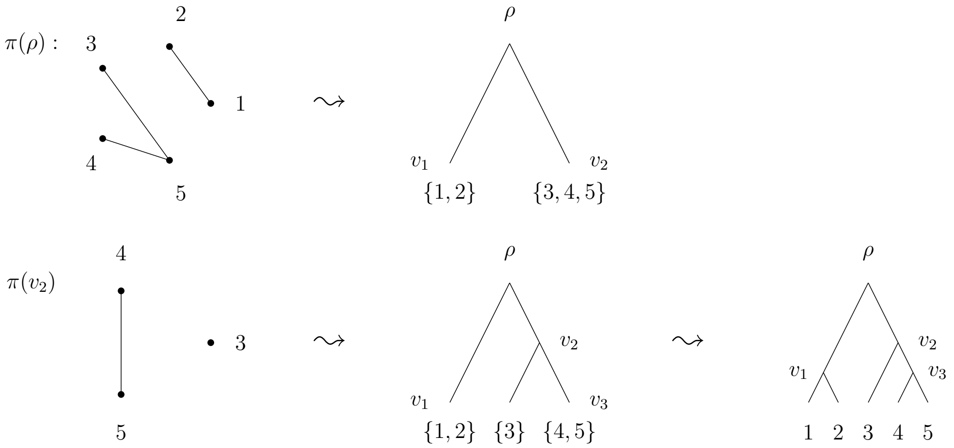


Figure 1.13: The different steps carried out by BUILD to construct a phylogenetic tree on $X = \{1, 2, 3, 4, 5\}$ displaying the triplets $12|4$, $45|3$ and $35|1$. See text for the detailed description of the algorithm.

As is easy to see, BUILD stops either if at some stages, the Aho graph $\pi(v)$ constructed along the way for some vertex v consists of a single connected component, or if a tree whose leaves l all satisfy $|S(l)| = 1$ is constructed. In the first case, there exists no phylogenetic tree displaying all triplets in C, whereas in the latter, the phylogenetic tree on X obtained by identifying each leaf l with the unique element of $S(l)$ displays all triplets in C. As it turns out, the BUILD algorithm also provides a characterization for the uniqueness of the tree displaying a set of triplets. Indeed, we have:

Theorem 1.4.1 ([1])**.** *Let C be a collection of triplets on a set X. We have:*

(i) There exists a rooted phylogenetic tree T on X such that $C \subseteq C(T)$ if and only if BUILD *returns a phylogenetic tree on X.*

(ii) There exists a unique rooted phylogenetic tree T on X such that $C \subseteq C(T)$ if and only if BUILD *returns a binary phylogenetic tree on X.*

Triplets are interesting to us due to their link with symbolic ultrametrics defined in Section 1.3.1. Indeed, for a given symbolic ultrametric δ on X, we can associate to δ the set $R(\delta)$ of triplets given by:

$$R(\delta) = \{xy|z : \delta(x,z) = \delta(y,z) \neq \delta(x,y)\}.$$

It was shown in [55] that there exists a rooted phylogenetic tree on X displaying all triplets in $R(\delta)$. Moreover, if T is the phylogenetic tree returned by BUILD when applied to $R(\delta)$, there exists a map t from $V_{int}(T)$ to the image set of δ such that (T,t) is the unique discriminating tree representing δ. The map t can then be trivially recovered from δ.

The problem of reconstructing a phylogenetic network from a collection of triplets C seems to be more complex than for phylogenetic trees. Indeed, it was shown in [60] that this problem is NP-hard. However, it turns out that the property of density of the input triplet set plays a key role in the simplification of this reconstruction problem. Indeed, if C is dense, it is possible to build in time $O(n^3)$ a level-1 network N satisfying $C(N) = C$, if such a network exists ([44]). A similar result, for level-2 networks, is given in [58].

1.4.2 Encoding properties and trinets

As it turns out, triplets are often too limited to capture the complexity of a rooted phylogenetic network, in the sense that two non-isomorphic phylogenetic networks may display the same collection of triplets (see [28]). This can be seen, for example, in Figure 1.14, where the phylogenetic networks τ_3 and τ_4, although different, both display precisely the triplets $xy|z$ and $zy|x$.

To help overcome this problem, the notion of a *trinet*, that is, a phylogenetic network on a set X of size 3, was introduced in [38]. In Figure 1.14, we represent the 14 possible 1-nested trinets as they appear in [38]. For N a phylogenetic network on X, we say that a trinet τ on $\{x,y,z\} \subseteq X$ is *displayed* by a phylogenetic network N if τ is the subnetwork of N induced by the leaves x, y and z (see Section 1.1.2).

As is easy to see, if N induces a trinet τ that is also a triplet, then N also displays τ as a triplet in the above sense. Moreover, if N is a phylogenetic tree, the converse also holds. More precisely, if T is a phylogenetic tree on X, we have that T displays a triplet $xy|z$ if and only if $xy|z$ is the subtree of T induced by the leaves x, y, z

As we shall see, trinets τ_1, τ_2 and τ_3 play a key role in Chapter 2. Note also that according to the definitions given in Section 1.1.3, the trinets τ_1 to τ_{12} are also level-1 trinets. This is not the case for τ_{13} and τ_{14}, since they both contain a vertex belonging to two distinct cycles. Finally, we can see that 5 of these trinets $(\tau_1, \tau_{11}, \tau_{12}, \tau_{13}$ and $\tau_{14})$ are not binary. However, it is interesting to note that binary networks display only binary trinets.

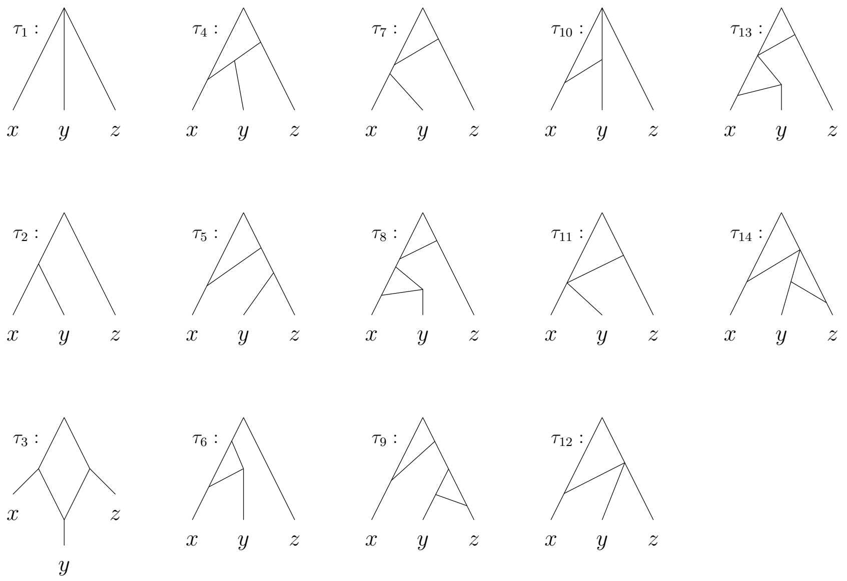


Figure 1.14: The fourteen 1-nested trinets up to a relabelling of their leaves.

The authors of [38] established the following result:

Theorem 1.4.2 ([38]). *1-nested networks are uniquely determined by the set of their induced trinets, and can be reconstructed from that set in polynomial time.*

This result has been extended in [61] to the space of level-2 networks and tree-child networks. However, it turns out that this result does not hold in general. Indeed, two non-isomorphic phylogenetic networks on 4 leaves are presented in [40], both of which display the same set of trinets.

In Chapter 2, where the idea of network reconstruction from trinets is used, we restrict ourselves to the space of level-1 networks. Thus, the uniqueness property postulated by Theorem 1.4.2 holds, which, as we shall see, will turn out to be useful.

Chap. 2

On symbolic 3-dissimilarities and labelled level-1 networks

Adapted from:

> K. T. Huber and G. E. Scholz. Beyond representing orthology relations with trees. *Algorithmica* (2018) 80(1): 73-103.

My personal contribution to this work has been the development of the algorithms, as well as their implementation in Python. I also established the results presented along the way, and I have written the first draft of the paper.

This chapter addresses the question of the representability of an orthology relation, formalized in terms of a symbolic 3-dissimilarity, by a level-1 phylogenetic network. We introduce the algorithm Network-Poping, aimed at building such a network representing a given symbolic 3-dissimilarity, should one exist, and provide a characterization of those dissimilarities that admit a representation in the form of a level-1 network.

2.1 Introduction

This chapter is based on [41], an original research work on symbolic 3-dissimilarities and the newly introduced concept of a labelled level-1 network. The minimal prerequists for understanding it are the notions discussed in Section 1.3.1. The starting point of this work is Theorem 1.3.1, characterizing symbolic ultrametrics in terms of two conditions (U1) and (U2). Let suppose we have got some biological data on a set of taxa X in the form of a symbolic distance (or symbolic 2-dissimilarity) $\delta : \binom{X}{2} \to M$, where M represents a set of some evolutionary events. Theorem 1.3.1 ensures that if δ satisfies conditions (U1) and (U2), there is a way to "represent" δ by a labelled phylogenetic tree (see Section 1.3.1 for more on this). However, it is generally too much to hope for that such a map δ inferred from real biological data satisfies that characterization. The question we want to answer here is, what can we do in that case?

To try to overcome this problem, two approaches may be considered. The first one is to assume that the data are "biased" or "noisy", and to try to correct the map δ in such a way that they satisfy the characterization of Theorem 1.3.1. As shown in [48], however, this often leads to NP-Complete problems.

We focus here on the second approach. Rather than trying to modify the data, we are looking for structures other than phylogenetic trees to represent them. As phylogenetic networks stand as a natural extension of phylogenetic trees, the first task is to extend the notion of a symbolic distance being represented by a tree to the notion of a symbolic distance being represented by a network.

As we have seen in Section 1.3.1, this notion of representability is directly related to the concept of a lowest common ancestor. However, although the lowest common ancestor of a set Y of taxa is unique in a tree, this is not necessarily the case for phylogenetic networks in general (see Section 1.1.2 and Figure 1.4 for an example of this non-uniqueness). Thus, we have to restrict ourselves to phylogenetic networks for which this uniqueness property holds.

As it turns out, level-1 networks satisfy this property (Lemma 2.2.1). Although they are not the only networks in that case, they stand as a good starting point, due to the relative simplicity of their non tree-like structures, which, by abuse of terminology, we call cycles in the following (see Section 2.2 for a formal definition).

A further problem that quickly arises is the question of uniqueness. Consider the symbolic distance $\delta : \binom{\{1,2,3\}}{2} \rightarrow M = \{\bullet, \times, \blacksquare\}$, defined by $\delta(1,2) = \bullet$, $\delta(1,3) = \blacksquare$ and $\delta(2,3) = \times$. This is clearly not a symbolic ultrametric, as it does not satisfy condition (U1) of Theorem 1.3.1, thus it cannot be represented by a labelled phylogenetic tree. However, all three level-1 networks depicted in Figure 2.1 satisfy that for any pair $x, y \in \{1,2,3\}$, the symbol assigned to lowest common ancestor of x and y coincide with $\delta(x,y)$. Clearly, this is not suitable from a uniqueness perspective. To tackle this problem, the key observation is that the symbol assigned to the lowest common ancestor of all three leaves is different in all three networks of Figure 2.1. For this reason, symbolic 3-dissimilarities appear as an interesting alternative to symbolic distances.

As it turns out, distinct level-1 networks on a set X may also induce the same 3-dissimilarity δ. This is for example the case of networks $\mathcal{N}_1$, $\mathcal{N}_2$ and $\mathcal{N}_3$ in Figure 2.1. Mimicking again the case of labelled phylogenetic trees, which requires the trees to be discriminating to ensure such a uniqueness, we introduce three mild properties, up to which this uniqueness holds (Corollary 2.4.5).

This chapter takes the form of a step-by-step development of an algorithm,

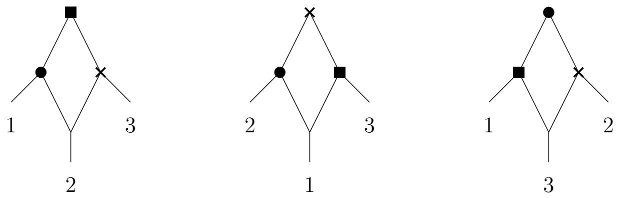


Figure 2.1: Three distinct level-1 representations of the same 2-dissimilarity $\delta : \binom{\{1,2,3\}}{2} \to M = \{\bullet, \times, \blacksquare\}$ defined by taking lowest common ancestors of pairs of leaves.

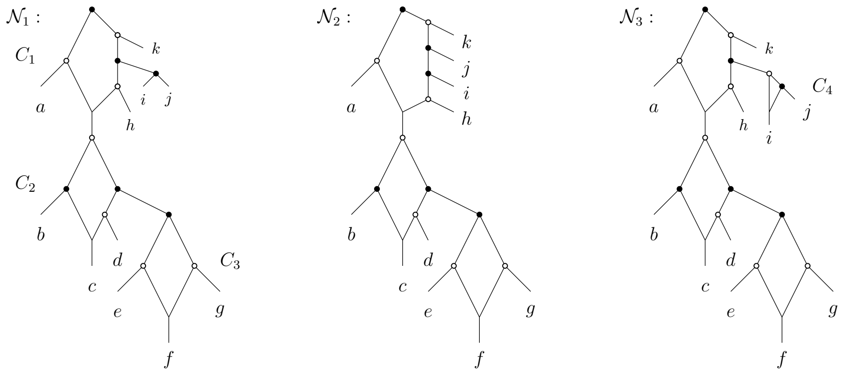


Figure 2.2: Three distinct level-1 representations of the same symbolic 3-dissimilarity δ on X={a,...,k}, with image set $M = \{\circ, \bullet\}$. In all three cases the underlying phylogenetic network is a level-1 network.

called Network-Popping aimed at building a unique labelled level-1 network from the 3-dissimilarity it induces, satisfying these three properties. Along the way, we characterize level-1 representable symbolic 3-dissimilarities δ on a set X in terms of eight natural properties (P1) – (P8) enjoyed by δ (Theorem 2.4.1). Furthermore, we characterize such dissimilarities in terms of level-1 representable symbolic 3-dissimilarities on subsets of X of size $|X| - 1$ (Theorem 2.4.8). Within a Divide-and-Conquer framework the resulting speed-up of algorithm Network-Popping might allow it to also be applicable to large datasets.

As we shall see, Network-Popping takes as input a 3-dissimilarity on some set X and is guaranteed to find, in $O(|X|^6)$-time, a level-1 representation for it if such a representation exists. For this, it relies on four further algorithms, among which is the Bottom-Up algorithm, introduced in [35] and described in Section 1.3.1.

This chapter is organized as follows. We present in Section 2.2.1 some basic terminology and results on level-1 networks, including Lemma 2.2.1 on the uniqueness of the notion of a lowest common ancestor in such a network. In Section 2.2.2, we formally define a labelled level-1 network and explore its relationships with symbolic 3-dissimilarities. We then introduce in Section 2.2.3 the crucial concept of a δ-trinet associated to a symbolic 3-dissimilarity and state Property (P1).

In Section 2.3.1, we present algorithm Find-Cycles as well as Properties (P2) and (P3). In Section 2.3.2, we introduce and analyze algorithm Build-Cycle. Furthermore, we state Properties (P4) – (P6). In Section 2.3.3, we present algorithms Vertex-Growing and Network-Popping. For the convenience of the reader, we illustrate all these algorithms by means of the level-1 networks depicted in Figure 2.2 and the symbolic 3-dissimilarity they induce.

We address in Section 2.4.1 the uniqueness question discussed above (Corollary 2.4.5), and we present algorithm Transform which allows to obtain, from a labelled level-1 network N, the unique labelled level-1 network N' returned by Network-Popping when applied to the symbolic 3-dissimilarity induced by N. As part of this we establish Theorem 2.4.1 which includes stating Properties (P7) and (P8). In Section 2.4.2, we establish Theorem 2.4.8.

In this chapter, unless stated otherwise, X denotes a finite set of size $n \geq 3$, M denotes a finite set of symbols of size at least two and $\odot$ denotes a symbol not already contained in M.

2.2 Preliminaries

We start here by formally introducing the notions of a labelled phylogenetic network and of a symbolic 3-dissimilarity. We collect relevant basic terminology and present some first results.

2.2.1 Rooted level-1 networks

By abuse of terminology, we call a subgraph H of a directed graph G a *cycle* of G if the induced subgraph $U(H)$ of $U(G)$ is a cycle. We start with introducing further terminology on cycles, which will be used throughout this chapter. Let N be a level-1 network on X. For C a cycle of N we denote by $h(C)$ the unique hybrid vertex of C (which we shall refer to as the *hybrid of* C) and we call the unique tree vertex v of C such that there exists two arc-disjoints directed paths from v to $h(C)$ the *top-vertex* of C, denoted by $r(C)$.

In the following, we say that a vertex w is *below* a vertex v in N if there exists a directed path from v to w. Having said that, we denote the set of all elements of X below $r(C)$ by $R(C)$ and the set of all elements of X below $h(C)$ by $H(C)$. By definition, we have $H(C) \subsetneq R(C)$. Moreover, for any leaf $x \in R(C) - H(C)$, we denote by $v_C(x)$ the last ancestor of x in C, that is, the unique vertex v of C such that x is below v but x is not also below a child of v contained in C. Note that $v_C(x)$ is the parent of x if and only if x is incident with a vertex in C.

Last-but-not-least, we call the vertex sets of the two arc-disjoint directed paths from $r(C)$ to $h(C)$ the *sides* of C. Denoting these two paths by P_1 and P_2, respectively, we say that two leaves x and y in $R(C) - H(C)$ *lie on the same side* of C if the vertices $v_C(x)$ and $v_C(y)$ are both interior vertices of the same path $P_i, i \in \{1, 2\}$, and that they *lie on different sides* otherwise. For example, denoting the unique cycle in the network on $X = \{1, \ldots, 7\}$ depicted in Figure 2.3 by C, we have $R(C) = \{1, 2, 3, 4, 5, 6\}$ and $H(C) = \{3, 4\}$. Furthermore, the sides of C are the paths $\{r(C), v_C(1), v_C(2), h(C)\}$ and $\{r(C), v_C(5), h(C)\}$, and the leaves 1 and 2 lie on the same side of C whereas the leaves 1 and 5 lie on different sides of C.

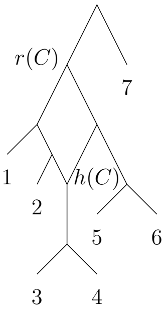


Figure 2.3: A level-1 network on $X = \{1, \ldots, 7\}$ with a single cycle C, of which we indicate the vertices $r(C)$ and $h(C)$.

As mentioned in Section 1.1.2, the notion of a lowest common ancestor is not well-defined for phylogenetic networks in general. However the situation changes in case the network in question is a level-1 network, as the following central result shows.

Lemma 2.2.1. *Let N be a level-1 network on X and assume that $Y \subseteq X$ such that $|Y| \geq 2$. Then there exists a unique interior vertex $v_Y \in V(N)$ such that $Y \subseteq C(v_Y)$ and $Y \nsubseteq C(v')$, for all children $v' \in V(N)$ of v_Y. Furthermore, there exists two distinct elements $x, y \in Y$ such that v_Y satisfies $\{x, y\} \subseteq C(v_Y)$ and $\{x, y\} \nsubseteq C(v')$for all children $v' \in V(N)$ of v_Y.*

Proof. Note first that if N is a level-1 network, and v and w are two vertices of N such that $C(v) \cap C(w) \neq \emptyset$, we either have $C(v) \subset C(w)$[1], which means that

[1] or the symmetric relation, $C(w) \subset C(v)$.

there exists a path from w to v, or there exists a hybrid vertex h of N satisfying $C(h) = C(v) \cap C(w)$.

Now, let $Y \subseteq X$, and assume by contradiction that there exists two distinct vertices v and w satisfying the definition of a lowest common ancestor for Y. This implies that the set $C(v) \cap C(w)$ is nonempty, since it contains Y. Suppose first that $C(v) \subseteq C(w)$. Since there exists a path from w to v, and a path from v to any element of Y, all vertices v' on the path from w to v satisfies $Y \subset C(v')$. In particular, w has a child satisfying this property, which is impossible since w is a lca for Y.

Thus, we must have $Y \subset C(h)$, where h is an hybrid vertex of N. Again, there exists a path from h to any element of Y, and since $C(h) \subseteq C(w)$, there exists a path from w to y. For the same reason, this implies that w cannot be a lca for Y.

Now, let $v = lca_N(Y)$ and assume by contradiction that for all pair $\{x, y\} \subsetneq Y$, $lca_N(x, y) \neq v$. This means that for each of these pairs, v has a child v' satisfying $\{x, y\} \subseteq C(v')$. Then, there exists at least as many distinct hybrid vertices reachable through a path from v that do not cross any other hybrid vertex as there are elements in Y, which is impossible since N is a level-1 network. □

Continuing with the terminology of Lemma 2.2.1, we refer to v_Y as the *lowest common ancestor of* Y *in* N, denoted by $lca_N(Y)$. As in the case of a phylogenetic tree, we write $lca(Y)$ rather than $lca_N(Y)$ if the network N we are referring to is clear from the context.

2.2.2 Labelled level-1 networks

Let N be a level-1 network on X. As in Section 1.3.1, we denote the set of interior vertices of N by $V_{int}(N)$. Moreover, we denote the set of interior vertices of N that are not hybrid vertices of N by $V_{int}(N)^-$.

A *labelled (phylogenetic) network (on* X*)* is a pair $\mathcal{N} = (N, t)$ consisting of a phylogenetic network N on X and a labelling map $t : V_{int}(N)^- \to M$. If N is a level-1 network then $\mathcal{N}$ is called a *labelled level-1 network*. To improve clarity of exposition we use calligraphic font to denote a labelled phylogenetic network.

Suppose $\mathcal{N} = (N, t)$ is a labelled level-1 network on X such that the vertices in $V_{int}(N)^-$ are labelled in terms of M. Then, we denote by $\delta_{\mathcal{N}} : \binom{X}{\leq 3} \to M \cup \{\odot\}$ the symbolic 3-dissimilarity[1] on X induced by $\mathcal{N}$ given by $\delta_{\mathcal{N}}(Y) = t(lca(Y))$ if $|Y| \neq 1$, and $\delta_{\mathcal{N}}(Y) = \odot$ otherwise. For $\mathcal{N}' = (N', t')$ a further labelled level-1 network on X, we say that $\mathcal{N}$ and $\mathcal{N}'$ are *isomorphic* if N and N' are isomorphic as phylogenetic networks, and $\delta_{\mathcal{N}} = \delta_{\mathcal{N}'}$.

[1] Note that $\delta_{\mathcal{N}}$ is not exactly a 3-dissimilarity as defined in Section 1.3.2, as it takes input in $\binom{X}{\leq 3}$ and not in $\binom{X}{3}$.

Conversely, suppose δ is a symbolic 3-dissimilarity on X. In view of Lemma 2.2.1, we call a labelled level-1 network $\mathcal{N} = (N, t)$ on X a *level-1 representation* of δ if $\delta = \delta_{\mathcal{N}}$. For ease of terminology, we sometimes say that δ is *level-1 representable* if the labelled network we are referring to is of no relevance to the discussion.

As is straightforward to see, any labelled network $\mathcal{N} = (N, t)$ that contains an arc e both of whose end-vertices have the same label induces the same symbolic 3-dissimilarity as the labelled network obtained from $\mathcal{N}$ by collapsing e. From a uniqueness point of view this is clearly undesirable. We therefore call a level-1 representation of δ *semi-discriminating* if N does not contain an arc (u, v) such that $t(u) = t(v)$ except for when there exists a cycle C of N with $|V(C) \cap \{u, v\}| = 1$. For example, all three labelled level-1 networks depicted in Figure 2.2 are level-1 representations of the same symbolic 3-dissimilarity δ. However $\mathcal{N}_1$ and $\mathcal{N}_3$ are semi-discriminating whereas $\mathcal{N}_2$ is not as the parents of j and i belong to the same cycle, are joined by an arc, and have same label.

Note that in case N is a phylogenetic tree on X the definition of a semi-discriminating labelled level-1 network to that of a discriminating labelled tree as defined in Section 1.3.1.

Clearly, it is too much to hope for that any symbolic 3-dissimilarity δ has a level-1 representation. The question therefore becomes: Which symbolic 3-dissimilarities have such a representation?

2.2.3 δ-triplets, δ-tricycles, and δ-forks

To make a first inroad into the aforementioned question, we next investigate the links between symbolic 3-dissimilarities and trinets (see Section 1.4.2). As we shall see, these turn out to be of fundamental importance for our algorithm Network-Popping (see Section 2.3.3) as well as for our analysis of its properties. Perhaps not surprisingly, trinets on their own are not strong enough to uniquely determine labelled level-1 networks in the sense that any two level-1 representations of a symbolic 3-dissimilarity must be isomorphic. To see this, suppose $|X| = 3$ and consider the symbolic 3-dissimilarity $\delta : \binom{X}{\leq 3} \to \{A, \odot\}$ that maps X and every 2-subset of X to A. Then the labelled network (τ_1, t) where t maps the unique vertex in $V_{int}(\tau_1)^-$ to A is a level-1 representation of δ and so is the labelled network (τ_4, t'), where every vertex in $V_{int}(\tau_4)^-$ is mapped to A by t'. Note that similar arguments may also be applied to the level-1 representations involving the trinets τ_4 to τ_{12} depicted in Figure 1.14.

To be able to state the next result (Lemma 2.2.2), we say that a symbolic 3-dissimilarity δ satisfies the *Helly-type property* if, for any three elements $x, y, z \in X$, we have $\delta(x, y, z) \in \{\delta(x, y), \delta(x, z), \delta(y, z)\}$. This is inspired by the notion of a *Helly family* (see e. g. [16]), that is, a collection $\mathcal{S}$ of sets such that for all collections $\mathcal{S}' \subseteq \mathcal{S}$ with $\cap_{S \in \mathcal{S}'} S \neq \emptyset$, there exists $S_1, S_2 \in \mathcal{S}'$ such that $\cap_{S \in \mathcal{S}'} S = S_1 \cap S_2$. In other words the intersection between the elements of $\mathcal{S}'$ corresponds the intersection between two elements of $\mathcal{S}'$, the same way the value of δ on $\{x, y, z\}$ corresponds to the value of δ on two out of these three elements if $\delta(x, y, z) \in \{\delta(x, y), \delta(x, z), \delta(y, z)\}$. Note that in the following, we sometimes also refer to the Helly-type property as Property (P1).

Lemma 2.2.2. *Suppose δ is a symbolic 3-dissimilarity on a set $X = \{x, y, z\}$ taking values in $M \cup \{\odot\}$. Then there exists a level-1 representation $\mathcal{N}$ of δ if and only if δ satisfies the Helly-type property. In that case $\mathcal{N}$ can be (uniquely) chosen to be (up to permutation of the leaves of the underlying level-1 network N) isomorphic to one of the trinets τ_1, τ_2 and τ_3 depicted in Figure 1.14.*

Proof. Suppose first that $\mathcal{N} = (N, t)$ is a level-1 representation of δ. Then, in view of Lemma 2.2.1, $\delta(x, y, z) \in \{\delta(x, y), \delta(x, z), \delta(y, z)\}$ must hold. Conversely, suppose that $\delta(x, y, z) \in E := \{\delta(x, y), \delta(x, z), \delta(y, z)\}$ holds. By analyzing the size of E it is straightforward to show that one of the situations indicated in the rightmost column of Table 2.1 must apply.

$\lvert\{\delta(x, y), \delta(x, z), \delta(y, z)\}\rvert$	$\delta(x, y, z) = \ldots$	N
1	$\delta(x, y) = \delta(x, z) = \delta(y, z)$	fork
3	$\delta(y, z)$	$x\vert\vert yz$
2	$\delta(y, z) \neq \delta(x, y) = \delta(x, z)$	$x\vert\vert yz$
2	$\delta(x, y) = \delta(x, z)$	$x\vert yz$

Table 2.1: For $\delta : \binom{X}{\leq 3} \to M^{\odot}$ a symbolic 3-dissimilarity we list all labelled trinets on $X = \{x, y, z\}$ in terms of the size of E.

With defining a labelling map $t : V_{int}(N)^{-} \to M \cup \{\odot\}$ in the obvious way using the second column of that table, it follows that $\mathcal{N}$ is a level-1 representation for δ. □

Interestingly, all of trinets τ_1 through to τ_{12} can be labelled in such a way that line 1 in Table 2.1 is satisfied. Similarly, all of trinets τ_4 through to τ_{11} and τ_2 can be labelled so that line 4 in Table 2.1 is satisfied. However only trinet τ_3 can be labelled so that lines 2 or 3 in that table hold. Reflecting our assumption that the amount of non-treelike signals in a dataset is small, we evoke parsimony regarding the number of cycles for the four cases discussed in Table 2.1 and focus from now on on the trinets τ_1, τ_2 and τ_3. We shall refer to them as *fork* on $X = \{x, y, z\}$, *triplet* $z|xy$, and *tricycle* $y||xz$, respectively.

Armed with Lemma 2.2.2, we make the following central definition. Suppose that $|Y| = 3$, that δ is a symbolic 3-dissimilarity on Y, and that $\mathcal{N} = (N, t)$ is the level-1 representation of δ found using Table 2.1. Then we call $\mathcal{N}$ a δ*-fork* if N is a fork on Y, a δ*-triplet* if N is a triplet on Y, and a δ*-tricycle* if N is a tricycle on Y, and we collectively refer to all three of them as a δ*-trinet*. As is easy to see all δ-trinets are semi-discriminating.

By abuse of terminology, we shall now refer for a symbolic 3-dissimilarity δ on X and any 3-subset $Y \subseteq X$ to a $\delta|_Y$-trinet as a δ-trinet.

For example, consider δ the symbolic 3-dissimilarity on $X = \{x, y, z, u\}$ induced by the labelled level-1 network depicted in Figure 2.4(i). Since $\delta(x, y) = \delta(y, z) \neq \delta(x, y) = \delta(x, y, z)$, Table 2.1 implies that $\delta|_{\{x,y,z\}}$ can be represented by a δ-tricycle, which we depict in Figure 2.4(ii). Similarly, we have that $\delta(x, y, u) = \delta(x, u) = \delta(y, u) \neq \delta(x, y)$, so by Table 2.1, $\delta|_{\{x,y,u\}}$ can be represented by a δ-triplet, which we depict in Figure 2.4(iv). Note that the labelled tricycle in Figure 2.4(iii) is also a representation of $\delta|_{\{x,y,u\}}$, but, from what precedes, it is not a δ-tricycle.

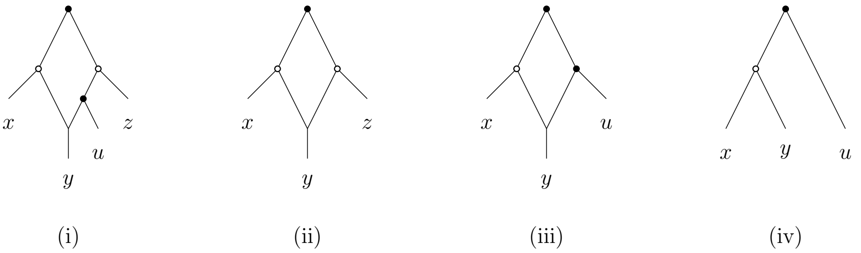


Figure 2.4: (i) A labelled level-1 network $\mathcal{N}$ on $X = \{x, y, z, u\}$. (ii) and (iv) Semi-discriminating level-1 representations of $\delta_{\mathcal{N}}$ restricted to $\{x, y, z\}$ and $Y = \{u, x, y\}$, respectively. (iii) A level-1 representation of $\delta_{\mathcal{N}}|_Y$ in the form of a labelled trinet that is not a $\delta_{\mathcal{N}}$-trinet.

2.3 Three steps for a reconstruction

Armed with these preliminaries results, we now turn our attention to the construction of the algorithm itself. We build it step by step, bearing in mind the following question: Given a symbolic 3-dissimilarity δ that is level-1 representable, how can we build a representation of δ?

2.3.1 Recognizing cycles

In this section, we introduce and analyze algorithm Find-Cycles (see Algorithm 1 for a pseudo-code version). Its purpose is to recognize cycles in a level-1 representation of a symbolic 3-dissimilarity δ if such a representation exists. As we shall see, this algorithm relies on Property (P1) and a certain graph $\mathcal{C}(\delta)$ that can be canonically associated to δ. Along the way, we also establish two further crucial properties enjoyed by a level-1 representable symbolic 3-dissimilarity.

Suggested by Property (U2), the following property is of interest to us where δ denotes again a symbolic 3-dissimilarity on X:

(P2) For all $x, y, z, u \in X$ distinct for which $\delta(x, y) = \delta(y, z) = \delta(z, u) \neq \delta(z, x) = \delta(x, u) = \delta(u, y)$ holds there exists exactly one subset $Y \subseteq \{x, y, z, u\}$ of size 3 such that a tricycle on Y underlies a level-1 representation of $\delta|_Y$.

As a first result, we obtain:

Lemma 2.3.1. *Suppose δ is a level-1 representable symbolic 3-dissimilarity on X. Then δ satisfies the Helly-type property as well as Property (P2).*

Proof. Note first that Property (P1) is a straightforward consequence of Lemma 2.2.1.

To see that Property (P2) holds, note first that since δ is level-1 representable there exists a labelled level-1 network (N, t) such that $\delta(Y) = t(lca(Y))$, for all subsets $Y \subsetneq X$ of size 2 or 3. Suppose $x, y, z, u \in X$ distinct are such that $\delta(x, y) = \delta(y, z) = \delta(z, u) \neq \delta(z, x) = \delta(x, u) = \delta(u, y)$. To see that there exists some $Y \subseteq Z := \{x, y, z, u\}$ for which $(N|_Y, t|_Y)$ is a δ-tricycle, assume for contradiction that there exists no such set Y. By Theorem 1.3.1, N cannot be a phylogenetic tree on X and, so, N must contain at least one cycle C. Without loss of generality, we may assume that $x \in H(C)$, and y lies on one of the two sides of C. By assumption $\delta(y, z) \neq \delta(x, z)$ and so either z and y lie on opposite sides of C, or z and y lie on the same side of C and $v_C(y)$ lies on the directed path from $r(C)$ to $v_C(z)$. As can be easily checked, either one of these two cases yields a contradiction since then $\delta(z, u) \neq \delta(x, u) = \delta(y, u)$ cannot hold for u, as required.

To see that there can exist at most one such tricycle on Z, assume for contradiction that there exist two tricycles τ and τ' with $L(\tau) \cup L(\tau') \subseteq Z$. Then $|L(\tau) \cap L(\tau')| = 2$. Choose $x, y \in L(\tau) \cap L(\tau')$. Note that the assumption on the elements of Z implies that x or y must be below the hybrid vertex of one of τ and τ' but not the other. Without loss of generality we may assume that y is below the hybrid vertex of τ but not below the hybrid vertex of τ'. Then y must lie on a side of the unique cycle C' of τ'. But this is impossible since the unique cycle of τ and C' are induced by the same cycle of N. □

We remark in passing that the proof of uniqueness in the proof of Lemma 2.3.1 combined with the structure of a level-1 network, readily implies the following result.

Lemma 2.3.2. *Suppose that δ is a symbolic 3-dissimilarity on X that is level-1 representable by a labelled network (N, t) and that $x, y, z \in X$ are three distinct* *elements such that $x||yz$ is a δ-tricycle. Let C denote the unique cycle in N such that $x \in H(C)$ and $y, z \in R(C) - H(C)$, and let $x' \in X$. If $x'||yz$ is a δ-tricycle then $x' \in H(C)$ and if $x||x'z$ is a δ-tricycle then $x' \in R(C)$ and x' and y lie on the same side of C.*

To better understand the structure of a symbolic 3-dissimilarity δ, we next associate to δ a graph $\mathcal{C}(\delta)$ defined as follows. The vertices of $\mathcal{C}(\delta)$ are the δ-tricycles and any two δ-tricycles τ and τ' are joined by an edge if $|L(\tau) \cap L(\tau')| = 2$. For example, consider the symbolic 3-dissimilarity $\delta_{\mathcal{N}_1}$ induced by the labelled level-1 network $\mathcal{N}_1$ pictured in Figure 2.2. Then the graph presented in Figure 2.5 is $\mathcal{C}(\delta_{\mathcal{N}_1})$.

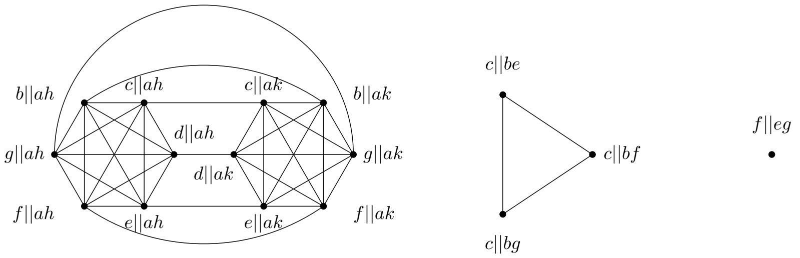


Figure 2.5: The graph $\mathcal{C}(\delta_{\mathcal{N}_1})$, where $\delta_{\mathcal{N}_1}$ is the symbolic 3-dissimilarity induced by the labelled level-1 network $\mathcal{N}_1$ in Figure 2.2.

The example in Figure 2.5 suggests the following property for a symbolic 3-dissimilarity $\delta : \binom{X}{\leq 3} \to M \cup \{\odot\}$ to be level-1 representable:

(P3) If τ and τ' are δ-tricycles contained in the same connected component of $\mathcal{C}(\delta)$, then
$$\delta(L(\tau)) = \delta(L(\tau')).$$

We collect first results concerning Property (P3) in the next proposition.

Proposition 2.3.3. *Suppose $\delta : \binom{X}{\leq 3} \to M \cup \{\odot\}$ is a symbolic 3-dissimilarity. If δ is level-1 representable or $|M| = 2$ holds then Property (P3) must hold. In particular, if $\mathcal{N}$ is a level-1 representation for δ then there exists a canonical injective map from the set of connected components of $\mathcal{C}(\delta)$ to the set of cycles of the level-1 network underlying $\mathcal{N}$.*

Proof. Suppose first that δ is level-1 representable. Let $\mathcal{N} = (N, t)$ denote a level-1 representation of δ. Then $\delta = \delta_{\mathcal{N}}$. Since $\delta_{\mathcal{N}}(x, y, z) = t(r(C))$ holds for all cycles C of N, and any $x \in H(C)$ and any $y, z \in R(C)$ that lie on different sides of C, Property (P3) follows.

Suppose next that $|M| = 2$. It suffices to show that Property (P3) holds for any two adjacent vertices of $\mathcal{C}(\delta)$. Suppose τ and τ' are two such vertices and that $x, y, z \in X$ are such that $\tau = x||yz$. Then there exists some $u \in X$ such that either $\tau' = u||yz$ or $\tau' = x||ru$ where $r \in \{y, z\}$. Without loss of generality we may assume that $r = y$. In view of Table 2.1, we clearly have $\delta(x, y) \neq \delta(x, y, z) = \delta(y, z)$. Since, in addition, $\delta(u, y, z) = \delta(y, z)$ holds in the former case it follows that $\delta(L(\tau)) = \delta(L(\tau'))$. In the latter case, we obtain $\delta(x, y, u) \neq \delta(x, y)$ and thus, $\delta(L(\tau)) = \delta(L(\tau'))$ follows in this case too as $|M| = 2$. The claimed injective map is a straightforward consequence of Lemma 2.3.2. □

Algorithm Find-Cycles exploits the injection mentioned in Proposition 2.3.3 by interpreting for a symbolic 3-dissimilarity δ a connected component C of $\mathcal{C}(\delta)$ in terms of two sets H_C and R'_C. Note that if C' is a cycle in the level-1 network underlying a level-1 representation of δ (if such a representation exists!), the sets $H(C')$ and H_C coincide and $R'_C \subseteq R(C')$ holds.

```
   Input: A symbolic 3-dissimilarity $\delta$ on $X$.
   Output: An integer $m \geq 0$ and $m$ pairs of subsets $(H_i, R'_i)$ of $X$, $1 \leq i \leq m$, or the
           statement "$\delta$ is not level-1 representable".
1  if $\delta$ satisfies Property (P1) then
2      Build the graph $\mathcal{C}(\delta)$;
3      Denote by $m$ the number of connected components of $\mathcal{C}(\delta)$;
4      for $i \in \{1, \dots, m\}$ do
5          Let $K_i$ denote a connected component of $\mathcal{C}(\delta)$;
6          set $H_i = \{x \in X :$ there exist $y, z \in X$ such that $x||yz$ is a vertex of $K_i\}$;
7          set $R'_i = H_i \cup \{y \in X :$ there exist $x, z \in X$ such that $x||yz$ is a vertex of $K_i\}$;
8      end
9      return $m, (H_1, R'_1), \dots, (H_m, R'_m)$;
10 end
11 else
12     return $\delta$ is not level-1 representable;
13 end
```

Algorithm 1: Find-Cycles – Property (P1) is checked in Line 1.

For example, for the symbolic 3-dissimilarity $\delta_{\mathcal{N}_1}$ induced by the labelled network $\mathcal{N}_1$ depicted in Figure 2.2, algorithm Find-Cycles returns the three pairs $(bcdefg, abcdefgk)$, $(c, bcefg)$ and (f, efg) where we write $x_1 \dots x_{|A|}$ for a set $A = \{x_1, \dots, x_{|A|}\}$.

2.3.2 Constructing cycles

We next turn our attention toward reconstructing a structurally very simple level-1 representation of a symbolic 3-dissimilarity (should such a representation exist).

For this, we use algorithm Build-Cycle which takes as input a symbolic 3-dissimilarity δ and a pair returned by Find-Cycles when given δ.

To state Build-Cycle, we require further terminology. Suppose N is a level-1 network. Then we say that N is *partially resolved* if all vertices in a cycle of N have degree three. Note that partially resolved level-1 networks may have interior vertices not contained in a cycle that have degree greater than three. Thus such networks need not be binary. If, in addition to being partially resolved, N is such that it contains a unique cycle C such that every non-leaf vertex of N is a vertex of C then we call N *simple*.

Algorithm Build-Cycle (see Algorithm 2 for a pseudo-code version) relies on a further graph called the TopDown graph associated to a symbolic 3-dissimilarity δ. For (H, R') a pair returned by algorithm Find-Cycle when given δ and $x \in H$ and $S \subseteq R'$, that graph essentially orders the vertices of S. Thus, for each connected component K of $\mathcal{C}(\delta)$, Build-Cycle computes a level-1 representation of δ corresponding to K (should such a representation exist).

We start with presenting a central observation concerning labelled level-1 networks.

Lemma 2.3.4. *Suppose $\mathcal{N} = (N, t)$ is a labelled level-1 network, and C is a cycle of N. Suppose also that $x, y, z \in X$ are three elements such that $x \in H(C)$, $y, z \in R(C) - H(C)$ and $t(v_C(z)) = t(r(C)) \neq t(v_C(y))$. Then, $v_C(z)$ lies on the directed path from $v_C(y)$ to $h(C)$ if and only if $y|xz$ is a $\delta_{\mathcal{N}}$-triplet.*

Proof. Put $\delta = \delta_{\mathcal{N}}$. Suppose first that $v_C(z)$ lies on the directed path from $v_C(y)$ to $h(C)$. Then $lca(x, y, z) = lca(x, y) = lca(y, z) = v_C(y)$ and $lca(x, z) = v_C(z)$. Hence, $\delta(x, y, z) = \delta(x, y) = \delta(y, z) = t(v_C(y)) \neq t(v_C(z)) = \delta(x, z)$. By Table 2.1, $y|xz$ is a δ-triplet.

Conversely, suppose that $y|xz$ is a δ-triplet. Then, by Table 2.1, we have $\delta(x, y, z) = \delta(x, y) = \delta(y, z) \neq \delta(x, z)$. Since $\delta(x, y) = t(v_C(y))$ and $\delta(x, z) = t(v_C(z))$, it follows that $\delta(x, y, z) = t(v_C(y)) \neq t(v_C(z))$. But then y and z must lie on the same side of C as otherwise $\delta(y, z) = t(r(C))$ follows which is impossible by assumption on x, y and z. Thus, either $v_C(y)$ must lie on a directed path P from $v_C(z)$ to $h(C)$ or $v_C(z)$ must lie on a directed path P' from $v_C(y)$ to $h(C)$. However $v_C(y)$ cannot be a vertex on P as otherwise $lca(y, z) = v_C(z)$ holds and, so, $\delta(y, z) = \delta(x, z)$ follows, which is impossible. Thus $v_C(z)$ must be a vertex on P'. □

With $\mathcal{N}$ and C as in Lemma 2.3.4, it follows from Lemma 2.3.2, that whenever algorithm Find-Cycles is given $\delta_{\mathcal{N}}$ as input, it returns a pair (H, R') such that $H = H(C)$ and $R' = H(C) \cup \{y \in R(C) : t(v_C(y)) \neq t(r(C))\}$. Moreover giving (H, R') and $\delta_{\mathcal{N}}$ as input to algorithm Build-Cycle, Lemma 2.3.4 implies that Build-Cycle finds all elements $z \in R(C) - R'$ for which there exists some $y \in R'$

such that $v_C(z)$ lies on the path from $v_C(y)$ to $h(C)$. However it should be noted that if $z \in R(C) - H(C)$ is such that $t(v) = t(r(C)) = t(v_C(z))$ holds for all vertices v on the path from $r(C)$ to $v_C(z)$ then the information captured by $\delta_{\mathcal{N}}$ for x, y, and z is in general not sufficient to decide if z and y lie on the same side of C or not. In fact, it is easy to see that, in general, $z \in R(C)$ need not even hold.

We now turn our attention to the aforementioned TopDown graph associated to a symbolic 3-dissimilarity δ on X which is defined as follows. Suppose that $S \subsetneq X$, and that $x \in X - S$. Then the vertex set of the *TopDown graph* $TD(S, x)$ is S and two elements $u, v \in S$ distinct are joined by a direct edge (u, v) if $u|vx$ is a δ-triplet. For example, consider again the dissimilarity $\delta_{\mathcal{N}_1}$ induced by the labelled level-1 network $\mathcal{N}_1$ depicted in Figure 2.2. Then $TD(\{d, e, f, g\}, c)$ is the graph depicted in Fig 2.6(a). In fact, $\{d, e, f, g\}$ is a side of the cycle of N_1 indicated by C_2.

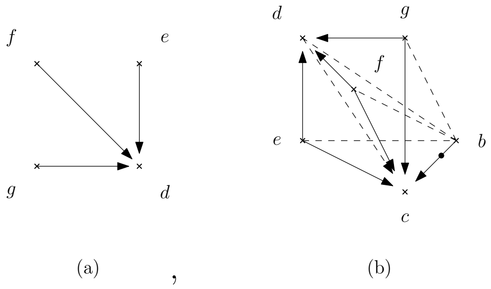


Figure 2.6: For $\delta_{\mathcal{N}_1}$ the symbolic 3-dissimilarity induced by the labelled network $\mathcal{N}_1$ pictured in Figure 2.2, we depict in (a) the TopDown graph $TD(\{d, e, f, g\}, c)$ and in (b) the CheckLabels graph $CL(\{c\}, \{b\}, \{d, e, f, g\})$ which we formally introduce in Section 2.4.1. In both graphs, the vertices are indicated by a cross. In the latter graph the value assigned to two vertices under $\delta_{\mathcal{N}_1}$ is indicated in terms of dashed and non-dashed edges (ignoring directions for the moment) . See text for details.

Rather than continuing with our analysis of algorithm Build-Cycle we break for the moment and illustrate it by means of an example. For this we return again to the symbolic 3-dissimilarity $\delta_{\mathcal{N}_1}$ on $X = \{a, \dots, k\}$ induced by the labelled level-1 network $\mathcal{N}_1$ depicted in Figure 2.2. Suppose $(c, bcefg)$ is a pair returned by algorithm Find-Cycle and $c||be$ is the δ-tricycle chosen in line 2 of Build-Cycle. Then $H = \{c\}$, $S'_b = \{b\}$ and $S'_e = \{e, f, g\}$ (lines 3 and 4), and $S_b = \{b\}$ and $S_e = \{d, e, f, g\}$ (lines 8 and 9). The graph $TD(S_e, c)$ is depicted in Figure 2.6(a). It implies that for the cycle C associated to the pair $(c, bcefg)$ in a level-1 representation of $\delta_{\mathcal{N}_1}$, we must have $v_C(e) = v_C(f) = v_C(g)$ and that one of the two sides of C is $\{d, e, f, g\}$. Since $|S_b| = 1$, the other side of C is $\{b\}$ (lines

Input: A symbolic 3-dissimilarity δ on X that satisfies Property (P1) and a pair (H, R') returned by algorithm Find-Cycle when given δ.
Output: Either a labelled simple level-1 network (C, t) on a partition of a subset X' of X such that $R' \subseteq X'$ and $H(K) = H$ holds for the unique cycle K of C, or the statement "δ is not level-1 representable".

```
1  set $rep=0$;
2  Choose a $\delta$-tricycle $x||yz$, where $x \in H$ and $y, z \in R' - H$;
3  set $S'_y = \{u \in R' : x||uz \text{ is a } \delta\text{-tricycle}\}$;
4  set $S'_z = \{u \in R' : x||yu \text{ is } \delta\text{-tricycle}\}$;
5  Initialize $C$ as a graph with three vertices respectively labelled by $r(C)$, $h(C)$ and $H$, and
   the arc $(h(C), H)$;
6  if for all $x' \in H$, $y' \in S'_y$ and $z' \in S'_z$, $x'||y'z'$ is a $\delta$-tricycle and $\delta(x,y,z) = \delta(x',y',z')$
   then
7      set $t(r(C)) = \delta(x,y,z)$;
8      set $S_y = S'_y \cup \{u \in X - R' : \text{ there exists } u' \in S'_y \text{ such that } u'|ux \text{ is a } \delta\text{-triplet}\}$;
9      set $S_z = S'_z \cup \{u \in X - R' : \text{ there exists } u' \in S'_z \text{ such that } u'|ux \text{ is a } \delta\text{-triplet}\}$;
10     if for all $u_1 \in S_y, u_2 \in S_z, \delta(u_1, u_2) = t(r(C))$ then
11         for $i \in \{y, z\}$ do
12             set $v_l = r(C)$;
13             if $TD(S_i, x') = TD(S_i, x'')$ for all $x', x'' \in H$ and $TD(S_i, x)$ does not
               contain a directed cycle then
14                 set $G = TD(S_i, x)$;
15                 set $rep=rep+1$;
16                 while $V(G) \neq \emptyset$ do
17                     Add a new child $v$ to $v_l$;
18                     set $C(v) = \{u \in S_i : u \text{ has in-degree } 0 \text{ in } G\}$;
19                     Delete from $G$ all vertices in $C(v)$;
20                     if for all $u, u' \in C(v)$, $x', x'' \in H \cup V(G)$, $\delta(u, x') = \delta(u', x'')$ then
21                         Choose some $u \in C(v)$;
22                         set $t(v) = \delta(x, u)$;
23                         Add the leaf $C(v)$ as a child of $v$;
24                         set $v_l = v$;
25                     end
26                     else
27                         Remove all vertices from $G$;
28                         set $rep=rep-1$;
29                     end
30                 end
31                 Add the arc $(v_l, h(C))$;
32             end
33         end
34     end
35 end
36 if $rep=2$ then
37     return $C$;
38 end
39 else
40     return $\delta$ is not level-1 representable;
41 end
```

Algorithm 2: Build-Cycle – The set R' is the set $H \cup S_y \cup S_z$, Property (P4) is checked in Lines 6, 10, and 20, and Properties (P3), (P6), (P7) and (P8) are checked in Lines 6, 13, 10 and 20, respectively.– See text for details.

11 to 33).

Continuing with our analysis of algorithm BUILD-CYCLE, we remark that the fact that the TopDown graph $TD(S_e, c)$ in the previous example is non-empty is not a coincidence. In fact, it is easy to see that the graph G defined in line 14 of BUILD-CYCLE is non-empty whenever δ is level-1 representable. Thus, the graph C returned by algorithm BUILD-CYCLE cannot contain multi-arcs. Note however that there might be tricycles induced by C of the form $x||uz$ with $u \in R' - S'_y$ as, for example, $\delta(x,z) = \delta(x,y) = \delta(z,y) = \delta(x,u)$ might hold and thus $x||uz$ is not a δ-tricycle. Note that similar reasoning also applies to S'_z and the extensions of S'_y and S'_z to S_y and S_z defined in lines 8 and 9, respectively. Also note that the sets S_y and S_z are dependent on the choice of the δ-tricycle in line 2. However, line 6 ensures that the labelled simple level-1 network returned by algorithm BUILD-CYCLE is independent of the choice of that δ-tricycle.

To establish Proposition 2.3.6 which ensures that algorithm BUILD-CYCLE terminates, we next associate to a directed graph G a new graph $P(G)$ by successively removing vertices of in-degree zero and their incident arcs until no such vertices remain. As a first almost trivial observation concerning that graph we have the following straightforward result whose proof we again omit.

Lemma 2.3.5. *Let G be a directed graph. Then $P(G)$ is nonempty if and only if G contains a directed cycle.*

Given as input to algorithm BUILD-CYCLE a symbolic 3-dissimilarity δ that satisfies Property (P1) and a pair (H, R') returned by algorithm FIND-CYCLE for δ we have:

Proposition 2.3.6. *Algorithm* BUILD-CYCLE *terminates.*

Proof. As is easy to check the only reason for algorithm BUILD-CYCLE not to terminate is the while loop initiated in its line 16. For $i = 1, 2$, this while loop works by successively removing vertices of in-degree 0 (and their incident arcs) from the graph $TD(S_i, x)$, and terminates if the resulting graph, i. e. $P(TD(S_i, x))$, is empty. Since line 13 ensures that this loop is entered if and only if $TD(S_i, x)$ does not contain a directed cycle, Lemma 2.3.5 implies that BUILD-CYCLE terminates. □

It is straightforward to see that when given a level-1 representable symbolic 3-dissimilarity δ such that the underlying level-1 network is in fact a simple level-1 network the labelled network returned by algorithm BUILD-CYCLE satisfies the following three additional properties (where we use the notations introduced in algorithm BUILD-CYCLE).

(P4) For $i \in \{y, z\}$, we have $S'_i = \{u \in S_i : \delta(u, x) \neq \delta(y, z)\}$ and $S_y \cap S_z = S_y \cap H = S_z \cap H = \emptyset$.

(P5) For all $u, v \in R := H \cup S_y \cup S_z$ and all $w \in X - R$, we have $\delta(u, w) = \delta(v, w)$.

(P6) For all $u, u' \in H$ and $i \in \{y, z\}$, the graphs $TD(S_i, u)$ and $TD(S_i, u')$ are isomorphic and do not contain a directed cycle.

Since the quantities on which these properties are based also exist for general symbolic 3-dissimilarities we next study Properties (P4) - (P6) for such dissimilarities. As a first consequence of Property (P4) combined with Properties (P1) and (P2), we obtain a sufficient condition under which the TopDown graph $TD(S_i, x)$ considered in algorithm BUILD-CYCLE does not contain a directed cycle (lines 13). For convenience, we employ again the notation used in Algorithm 2.

Proposition 2.3.7. *Suppose that $\delta : \binom{X}{\leq 3} \to M \cup \{\odot\}$ is a symbolic 3-dissimilarity that satisfies Properties (P1), (P2) and (P4), that (H, R') is a pair returned by algorithm* FIND-CYCLES *when given δ, and that x, y and z are as specified as in line 2 of algorithm* BUILD-CYCLE*. Then the following hold for $i = y, z$.*
(i) If $TD(S_i, x)$ contains a directed cycle then it contains a directed cycle of size 3.
(ii) $TD(S_i, x)$ does not contain a directed cycle of length 3 whenever $|M| = 2$ holds.

Proof. (i) By symmetry, it suffices to show the proposition for $i = y$. Suppose $TD(S_y, x)$ contains a directed cycle. Over all such cycles in $TD(S_y, x)$, choose a directed cycle C of minimal length. If $|V(C)| = 3$, then the statement clearly holds.

Suppose for contradiction for the remainder that $|V(C)| \geq 4$. Suppose $a, b, c, d \in V(C)$ are such that (a, b), (b, c), (c, d) are three arcs in C. We next distinguish between the cases that $|V(C)| \geq 5$ and that $|V(C)| = 4$.

Suppose $|V(C)| \geq 5$. Then since $a, c \in S_y$, Lemma 2.3.2 combined with the minimality of C implies that we either have a δ-fork on $\{a, c, x\}$ or the δ-triplet $ac|x$. Hence, $\delta(x, a) = \delta(x, c)$ holds in either case. Note that similar arguments also imply that $\delta(x, b) = \delta(x, d)$. Since $|V(C)| \geq 5$, the arcs (a, d) and (d, a) cannot be contained in $TD(S_y, x)$ and, using again similar arguments as before, $\delta(x, a) = \delta(x, d)$ must hold. In combination, we obtain $\delta(x, a) = \delta(x, b)$ which is impossible in view of (a, b) being an arc in $TD(S_y, x)$ and thus $\delta(x, a) \neq \delta(x, b)$.

Suppose $|V(C)| = 4$. By the minimality of C, neither (b, d) (d, b), (a, c) nor (c, a) can be an arc in $TD(S_y, x)$. Using similar arguments as in the previous case, it follows that $\delta(x, b) = \delta(x, d)$ and $\delta(x, a) = \delta(x, c)$. Combined with the facts that (a, b), (b, c), (c, d) are arcs in C and that (d, a) must also be an arc in C as $|V(C)| = 4$, it follows that with $A := \delta(c, d)$ and $B := \delta(b, c)$ we have

$$A = \delta(x, c) = \delta(x, a) = \delta(a, b) \neq \delta(x, b) = \delta(x, d) = \delta(d, a) = \delta(b, c) = B. \quad (2.1)$$

Note that, $\delta(a,c) \in \{A,B\}$ must also hold as otherwise $|\{\delta(a,c),\delta(a,b),\delta(b,c)\}| = 3$ and so, in view of Table 2.1, $\delta|_{\{a,b,c\}}$ would be level-1 representable by a δ-tricycle on $\{a,b,c\}$. But then $H \cap \{a,b,c\} \neq \emptyset$ which is impossible in view of Property (P4). Similarly, one can show that $\delta(b,d) \in \{A,B\}$. By combining a case analysis as indicated in Table 2.1 with Equation 2.1, it is straightforward to see that each of the four detailed combinations of $\delta(a,c)$ and $\delta(b,d)$ in that table yields a contradiction in view of Property (P2).

(ii) By symmetry, it suffices to assume $i = y$. Let $|M| = 2$ and assume for contradiction that $TD(S_y, x)$ contains a directed cycle C of size 3. Let s, u, v denote the 3 vertices of C such that (s,u), (u,v) and (v,s) are the three arcs of C. Then $\delta(u,x) \neq \delta(s,x) \neq \delta(v,s) = \delta(v,x) \neq \delta(u,v) = \delta(u,x)$ must hold. Since $|M| = 2$, this is impossible. □

2.3.3 Constructing a level-1 representation

In this section, we present algorithm Network-Popping which allows us to decide if a symbolic 3-dissimilarity is level-1 representable or not. If it is, then Network-Popping is guaranteed to find a level-1 representation in polynomial time.

Network-Popping takes as input a symbolic 3-dissimilarity δ on X and employs a top-down approach to recursively construct a semi-discriminating level-1 representation for δ (if such a representation exists). For l a leaf whose label set is of size at least two and constructed in one of the previous steps it essentially works by either replacing l with a labelled simple level-1 network or a labelled phylogenetic tree. To compute those networks algorithms Find-Cycle and Build-Cycle are used, and to construct such trees algorithm Vertex-Growing is employed. At the heart of the latter lie Proposition 2.3.9 and algorithm Bottom-Up introduced in [35] and described in Section 1.3.1.

To be able to state algorithm Vertex-Growing, we require again further terminology. Following [55], we call a collection $\mathcal{H}$ of non-empty subsets of X a *hierarchy on* X if $A \cap B \in \{A, B, \emptyset\}$ holds for any two sets $A, B \in \mathcal{H}$. The proof of the following result is straightforward and thus omitted.

Lemma 2.3.8. *Let N be a level-1 network with cycles $C_1, C_2, \ldots, C_k$, $k \geq 1$. Then, $\mathcal{H}_N = \{R(C_1), R(C_2), \ldots, R(C_k)\}$ is a hierarchy on X.*

Suppose $\mathcal{A}$ is a set of non-empty subsets of X. Then we define a relation $\sim_{(X,\mathcal{A})}$ on X by putting $x \sim_{(X,\mathcal{A})} y$ if there exists some $A \in \mathcal{A}$ such that $x, y \in A$, for all $x, y \in X$. Note first that $\sim_{(X,\mathcal{A})}$ is clearly an equivalence relation whenever $\mathcal{A}$ is a hierarchy. In addition, suppose that $\mathcal{A}$ is such that the partition X' of X induced by $\sim_{(X,\mathcal{A})}$ has size two or more. If $\delta : \binom{X}{\leq 3} \to M \cup \{\odot\}$ is a symbolic 3-dissimilarity such that for any two sets $Y, Y' \in X'$ we have $\delta(x,y) = \delta(x',y')$ for all $x, x' \in Y$ and $y, y' \in Y'$, then we associate to δ the map $\hat{\delta}$ given by

$$\begin{array}{rcl} \hat{\delta} : \binom{X'}{\leq 2} & \to & M \cup \{\odot\} \\ \{Y_1, Y_2\} & \mapsto & \begin{cases} \odot & \text{if } Y_1 = Y_2, \\ \delta(y_1, y_2), \text{ where } y_1 \in Y_1, y_2 \in Y_2 & \text{otherwise.} \end{cases} \end{array}$$

Note that $\hat{\delta}$ is clearly well-defined and a symbolic 2-dissimilarity on X'. Associating to a level-1 representation $\mathcal{N} = (N,t)$ of δ the set $\mathcal{R} := \{R(C) : C$ is a cycle of $N\}$, we have the following result as an immediate consequence.

Proposition 2.3.9. *Suppose $\mathcal{N}$ is a labelled level-1 network on X and X' is the partition of X induced by the relation $\sim_{(X,\mathcal{R})}$ on X. If $|X'| \geq 2$ then $\widehat{\delta_{\mathcal{N}}}$ is well defined and satisfies Properties (U1) and (U2). In particular, $\widehat{\delta_{\mathcal{N}}}$ is a symbolic ultrametric on X'.*

Proof. Put $\mathcal{N} = (N,t)$ and $\delta' = \widehat{\delta_{\mathcal{N}}}$. Note first that for all $x, y \in X$, Lemma 2.3.8 implies that there exists some $R \in \mathcal{R}$ such that $x, y \in R$ if and only if there exists $R' \in \mathcal{R}' := \{R \in \mathcal{R} : R$ is set-inclusion maximal in $\mathcal{R}\}$ such that $x, y \in R'$. Let T_N denote the tree obtained from N by first collapsing for every cycle C of N with $R(C) \in \mathcal{R}'$ all vertices below or equal to $r(C)$ into a vertex and then labelling that vertex by $R(C)$. Put $t_N := t|_{V(T_N)}$. Then (T_N, t_N) is clearly a labelled phylogenetic tree on X'. Since $\mathcal{N}$ is a labelled level-1 network, it follows that (T_N, t_N) is a symbolic discriminating representation of $\widehat{\delta_{\mathcal{N}}}$. In view of Theorem 1.3.1, the proposition follows. □

Input: A symbolic 3-dissimilarity δ on a set X, a subset $Y \subseteq X$, and a hierarchy $\mathcal{S}$ of proper subsets of Y.
Output: A discriminating symbolic representation on the partition of Y induced by $\sim_{(Y,\mathcal{S})}$ or the statement "There exists no discriminating symbolic representation".

```
1 Let $Y'$ denote the partition of $Y$ induced by $\sim_{(Y,\mathcal{S})}$;
2 Apply the Bottom-Up algorithm to the symbolic ultrametric $\hat{\delta}$ induced by $\delta$ on $Y'$, as
  considered in Proposition 2.3.9;
3 if Bottom-Up returns a labelled tree $\mathcal{T}$ then
4     return $\mathcal{T}$;
5 end
6 else
7     return There exists no discriminating symbolic representation. ;
8 end
```

Algorithm 3: Vertex-Growing – Property (P2) is checked in Line 3.

To illustrate algorithm Vertex-Growing consider again the symbolic 3-dissimilarity $\delta_{\mathcal{N}_1}$ induced by the labelled level-1 network on $X = \{a \ldots, k\}$ depicted in Figure 2.2. Let $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$ denote the three labelled simple level-1 networks returned by algorithm Build-Cycle when given $\delta_{\mathcal{N}_1}$ such that $L(\mathcal{M}_1) = X$, $L(\mathcal{M}_2) = \{b, \dots, g\}$ and $L(\mathcal{M}_3) = \{e, f, g\}$. Then the partition of X found in line 1 of algorithm Vertex-Growing when given $\delta_{\mathcal{N}_1}$ and $\mathcal{R} = \{L(\mathcal{M}_i) : 1 \leq i \leq 3\}$ is X itself, since any two leaves of X are in relation with respect to $\sim_{(X,\mathcal{R})}$. Thus, the discriminating symbolic representation returned by Bottom-Up is a single leaf.

Armed with the algorithms Find-Cycles, Build-Cycle, and Vertex-Growing, we next present a pseudo-code version of algorithm Network-Popping (Algorithm 4).

To be able to establish in Proposition 2.3.11 that algorithm Network-Popping returns a semi-discriminating level-1 representation for a symbolic 3-dissimilarity (if such a representation exists), we require the following technical result.

Proposition 2.3.10. *Let δ be a symbolic 3-dissimilarity on X satisfying Property (P1), and assume that* Network-Popping *returns a labelled level-1 network $\mathcal{N}$ on X when given δ as input. Then the restrictions $\delta|_{\binom{X}{\leq 2}}$ and $\delta_{\mathcal{N}}|_{\binom{X}{\leq 2}}$ of δ and $\delta_{\mathcal{N}}$ to $\binom{X}{\leq 2}$, respectively, coincide if and only if δ and $\delta_{\mathcal{N}}$ coincide.*

Proof. Put $\mathcal{N} = (N, t)$. Also, put $\delta' = \delta|_{\binom{X}{\leq 2}}$ and $\delta'_{\mathcal{N}} = \delta_{\mathcal{N}}|_{\binom{X}{\leq 2}}$. Clearly, if δ and $\delta_{\mathcal{N}}$ coincide then $\delta' = \delta'_{\mathcal{N}}$ must hold.

Conversely, assume that $\delta' = \delta'_{\mathcal{N}}$. Let $Z = \{a, b, c\} \in \binom{X}{3}$ and put $m = \delta(Z)$. Note that since $\mathcal{N}$ is clearly a level-1 representation of $\delta_{\mathcal{N}}$, Lemma 2.3.1 implies that $\delta_{\mathcal{N}}$ also satisfies Property (P1). Further note that, up to permuting the elements in Z, we either have (i) a δ-fork on Z, (ii) $a|bc$ is a δ-triplet, or (iii) $a||bc$ is a δ-tricycle.

If Case (i) holds then $\delta(a, b) = \delta(a, c) = \delta(b, c) = m$. Since, by assumption, $\delta(Y) = \delta_{\mathcal{N}}(Y)$ for all $Y \in \binom{X}{2}$, we also have $\delta_{\mathcal{N}}(a, b) = \delta_{\mathcal{N}}(a, c) = \delta_{\mathcal{N}}(b, c) = m$. Hence, $\delta_{\mathcal{N}}(Z) = m = \delta(Z)$ as δ satisfies Property (P1).

If Case (ii) holds then $m = \delta(a, b) = \delta(a, c) \neq \delta(b, c)$. Assume for contradiction that $\delta_{\mathcal{N}}(Z) \neq m$. Then, since $\delta_{\mathcal{N}}$ satisfies Property (P1) it follows that $\delta_{\mathcal{N}}(Z) = \delta_{\mathcal{N}}(b, c)$. By Table 2.1, $a||bc$ must be a $\delta_{\mathcal{N}}$-tricycle. Hence, there must exist a cycle C in N such that $a \in H(C)$, b and c are contained in $R(C)$ but lie on different sides of C, and $t(r(C)) = \delta_{\mathcal{N}}(Z)$. Since algorithm Network-Popping completes by returning $\mathcal{N}$ it follows that C is constructed in the while-loop starting in line 16 of algorithm Build-Cycle. But then the condition in line 6 of Build-Cycle has to be satisfied which implies that $t(r(C)) = \delta(Z)$ in view of line 7 of that algorithm. Hence, $m \neq \delta_{\mathcal{N}}(Z) = t(r(C)) = \delta(Z) = m$ which is impossible.

If Case (iii) holds then the while-loop initiated in line 16 of algorithm Build-Cycle implies that there must exist a cycle C in N such that $t(r(C)) = \delta(Z) = m$.

```
Input: A symbolic 3-dissimilarity $\delta$ on $X$.
Output: A semi-discriminating level-1 representation $\mathcal{N} = (N, t')$ of $\delta$, if such a
        representation exists, or the statement "$\delta$ is not level-1 representable".
1  Initialize $N$ as an unique vertex $v$, labelled by $X$;
2  set $r = 1$;
3  Use FIND-CYCLES($\delta$) to obtain $m \geq 0$ pairs $(H_i, R'_i)$ of subsets $H_i$ and $R'_i$ of $X$,
   $1 \leq i \leq m$;
4  if for all $i \in \{1, \dots, m\}$, BUILD-CYCLE($\delta; H_i, R_i$) returns a labelled simple level-1
   network $(C_i, t_i)$ as described in that algorithm then
5      put $R_i = R(C_i)$, and $\mathcal{R} = \{R_1, \dots, R_m\}$;
6      if for all $i \in \{1, \dots, m\}$, and all $y, z \in R_i$, and $x \notin R_i$, we have $\delta(x, y) = \delta(x, z)$ then
7          while there exists a leaf $l$ of $N$ whose label set $V_l \subseteq X$ has two or more elements
           AND $r \neq 0$ do
8              if there exists $i \in \{1, \dots, m\}$ such that $V_l = R_i$ then
9                  identify $l$ with the root of the labelled simple level-1 network
                   corresponding to $R_i$ and replace $N$ with the resulting labelled level-1
                   network;
10             end
11             else
12                 put $\mathcal{S}_l = \{R \in \mathcal{R} : R \subseteq V_l\}$;
13                 if VERTEX-POPPING($\delta, V_l, \mathcal{S}_l$) returns a discriminating symbolic
                   representation $\mathcal{T} = (T, t)$ then
14                     identify $l$ with the root of $T$ and replace $N$ with the resulting
                       labelled level-1 network;
15                 end
16                 else
17                     set $r = 0$;
18                 end
19             end
20         end
21     end
22 end
23 if $r = 1$ AND $N$ is not $v$ then
24     return $\mathcal{N} := (N, t')$ where $t'$ is canonically obtained by combining the maps $t$ and $t_i$,
       $1 \leq i \leq m$;
25 end
26 else
27     return $\delta$ is not level-1 representable;
28 end
```

Algorithm 4: NETWORK-POPPING – Property (P5) is checked in Line 6.

Since $\mathcal{N}$ is returned by algorithm Network-Popping when given δ and $\mathcal{N}$ is clearly a level-1 representation for $\delta_{\mathcal{N}}$ it follows that $\delta_{\mathcal{N}}(Z) = t(r(C)) = m = \delta(Z)$. □

As a first result concerning algorithm Network-Popping, we have

Proposition 2.3.11. *Suppose δ is a symbolic 3-dissimilarity on X, and* Network-Popping *applied to δ returns a labelled level-1 network $\mathcal{N}$. Then $\delta = \delta_{\mathcal{N}}$. In particular, $\mathcal{N}$ is a level-1 representation for δ.*

Proof. Put $\mathcal{N} = (N, t)$. In view of Proposition 2.3.10, it suffices to show that $\delta(a, b) = \delta_{\mathcal{N}}(a, b)$ holds for all $a, b \in X$ distinct. Let a and b denote two such elements. We distinguish between the cases that either (i) there exists a cycle C of N such that $v_C(a) \neq v_C(b)$, or (ii) that no such cycle exists.

Assume first that Case (i) holds. Then a and b lie either on the same side of C, or one of a and b is below the hybrid $h(C)$ of C and the other lies on a side of C, or a and b lie on different sides of C. If a and b lie on the same side of C or one of them is below $h(C)$ then we may assume without loss of generality that there exists a directed path in C from $v_C(a)$ to $v_C(b)$. Then line 22 of algorithm Build-Cycle implies $t(v_C(a)) = \delta(a, b)$. Since $lca(a, b) = v_C(a)$, it follows that $\delta_{\mathcal{N}}(a, b) = t(v_C(a)) = \delta(a, b)$, as required.

If a and b lie on different sides of C then $x||ab$ is a δ-tricycle, for x as in line 2 of algorithm Build-Cycle. Since that algorithm completes, line 7 of that algorithm implies $\delta(a, b) = t(r(C))$. But then $\delta_{\mathcal{N}}(a, b) = t(r(C)) = \delta(a, b)$, as $\mathcal{N}$ is returned by Network-Popping.

For the remainder, assume that Case (ii) holds, that is, there exists no cycle C of N such that $v_C(a) \neq v_C(b)$. Consider the vertex $v_0 \in V(N)$ defined as follows: if the path from the root ρ_N of N to $lca(a, b)$ does not contain a vertex that is also contained in a cycle of N, then put $v_0 = \rho_N$. Otherwise let v_0 denote the last vertex on a directed path from ρ_N to $lca(a, b)$ such that v_0 belongs to a cycle Z of N. Note that $v_0 = lca(a, b)$ holds if $lca(a, b)$ is also contained in Z. Put $V = C(v_1)$ where v_1 is the unique child of v_0 not contained in Z, and let V' denote the partition of V induced by $\sim_{(V, \mathcal{S}_{v_0})}$ where for any vertex $w \in V(N)$ the set $\mathcal{S}_w$ is defined as in line 12 of algorithm Network-Popping. Let $R_a, R_b \in V'$ such that $a \in R_a$ and $b \in R_b$. Then line 5 of Network-Popping implies $\hat{\delta_{\mathcal{N}}}(R_a, R_b) = \delta_{\mathcal{N}}(a, b)$ and $\hat{\delta}(R_a, R_b) = \delta(a, b)$. Since $\mathcal{N}$ is returned by Network-Popping when given δ, line 12 of that algorithm implies $\hat{\delta}(R_a, R_b) = \hat{\delta_{\mathcal{N}}}(R_a, R_b)$. Consequently, $\delta_{\mathcal{N}}(a, b) = \delta(a, b)$ holds in this case too. □

We conclude this section with some remarks concerning the runtime of algorithm Network-Popping. Suppose X and δ are as in the description of that algorithm. Then the runtime of Network-Popping manifests itself through

(i) pairwise comparisons between δ-tricycles (construction of the graph $\mathcal{C}(\delta)$) and δ-triplets (Algorithm Bottom-Up), respectively, and (ii) comparisons between elements x of X and (a) δ-tricycles containing x to determine the pair (H, R') associated to a given connected component of $\mathcal{C}(\delta)$ and (b) δ-triplets to obtain the TopDown graph associated to a given connected component of $\mathcal{C}(\delta)$. Since the number of δ-tricycles and of δ-triplets is bounded by the number $\frac{n(n-1)(n-2)}{6}$ of 3-subsets of X and the number of δ-tricycles and of δ-triplets containing a given element $x \in X$, respectively, is bounded by the number $\frac{(n-1)(n-2)}{2}$ of 2-subsets of $X - \{x\}$, it follows that the runtime of Network-Popping is $O(n^6)$.

2.4 Encoding and characterization properties

Now that we are ensured the correctness of Algorithm Network Popping, we can derive from its structure and the previous results some properties of level-1 representable symbolic 3-dissimilarities, and of their representations.

2.4.1 Uniqueness of the output

As is easy to see, there exist symbolic 3-dissimilarities that although they satisfy Properties (P1) - (P6) are not level-1 representable. The reason for this is that such 3-dissimilarities need not satisfy the assumptions of lines 10 and 20 in algorithm Build-Cycle. A careful analysis of that algorithm suggests however two further properties for a symbolic 3-dissimilarity to be level-1 representable. To state them, we next associate to a symbolic 3-dissimilarity its CheckLabels graph. Suppose Y_0, Y_1, and Y_2 are three pairwise disjoint subsets of X such that for all $x, x' \in Y_0$ and all $i = 1, 2$, the graphs $TD(Y_i, x)$ and $TD(Y_i, x')$ are isomorphic (which is motivated by Property (P6)). Then we denote by $CL(Y_0, Y_1, Y_2)$ the *CheckLabels graph* associated to δ, Y_0, Y_1, and Y_2 defined as follows. The vertex set of $CL(Y_0, Y_1, Y_2)$ is $Y_0 \cup Y_1 \cup Y_2$. Any pair $(u, v) \in Y_1 \times Y_2$ is joined by an (undirected) edge $\{u, v\}$, any pair $(u, v) \in (Y_1 \cup Y_2) \times Y_0$ is joined by a directed edge (u, v), and two elements $u, v \in Y_i$, $i = 1, 2$, are joined by a directed edge (u, v) if there exists a direct path from u to v in $TD(Y_i, x)$. Finally, to each edge of $CL(Y_0, Y_1, Y_2)$ with end vertices u and v or directed edge of that graph with tail u and head v, we assign the label $\delta(u, v)$. We illustrate the CheckLabels graph in Figure 2.6(b) for the network $\mathcal{N}_1$ depicted in Figure 2.2.

Using the terminology of algorithm Build-Cycle it is straightforward to observe that the following two properties are implied by Build-Cycle's lines 10 and 20 whenever its input symbolic 3-dissimilarity is level-1 representable:

(P7) All undirected edges of $CL(H, S_y, S_z)$ have the same label;

(P8) For all vertices u of $CL(H, S_y, S_z)$, all directed edges in $CL(H, S_y, S_z)$ with tail u have the same label.

As indicated in Table 2.2, Properties (P1) - (P8) are independent of each other. As we shall see, they allow us to characterize level-1 representable symbolic 3-dissimilarities (Theorem 2.4.1).

Prop.	X	M	δ
(P1)	$\{x, y, z\}$	$\{D, S\}$	$\delta(x, y) = \delta(x, z) = \delta(y, z) = D$; $\delta(x, y, z) = S$.
(P2)	$\{x, y, z, u\}$	$\{D, S\}$	$\delta(x, y, z) = \delta(y, z, u) = \delta(x, y) = \delta(y, z) = \delta(z, u) = D$; $\delta(Y) = S$ otherwise.
(P3)	$\{x_1, x_2, y, z\}$	$\{D, S_1, S_2\}$	$\delta(x_i, y, z) = S_i$, $i \in \{1, 2\}$; $\delta(Y) = D$ otherwise.
(P4)	$\{x, y, z, u\}$	$\{D, S\}$	$\delta(x, y, u) = \delta(x, u) = \delta(y, z) = \delta(x, y, z) = D$; $\delta(Y) = S$ otherwise.
(P5)	$\{1, \dots, 5\}$	$\{D, S\}$	$\delta(1, 4) = S$; $\delta(Y) = \delta_{\mathcal{N}_5}(Y)$ otherwise.
(P6)	$\{1, \dots, 6\}$	$\{D, S\}$	$\delta(3, 6) = \delta(2, 3, 6) = D$; $\delta(Y) = \delta_{\mathcal{N}_6}(Y)$ otherwise.
(P7)	$\{1, \dots, 5\}$	$\{D, S\}$	$\delta(2, 4) = \delta(2, 3, 4) = \delta(1, 2, 4) = \delta(2, 4, 5) = S$; $\delta(Y) = \delta_{\mathcal{N}_7}(Y)$ otherwise.
(P8)	$\{1, \dots, 5\}$	$\{D, S\}$	$\delta(3, 5) = \delta(3, 4, 5) = D$; $\delta(Y) = \delta_{\mathcal{N}_8}(Y)$ otherwise.

Table 2.2: For sets X and M and δ a symbolic 3-dissimilarity on X as indicated, the property stated in the first column of each row holds whereas the remaining seven properties do not. For $i = 5, 6, 7, 8$, the networks $\mathcal{N}_i$ are depicted in Figure 2.7.

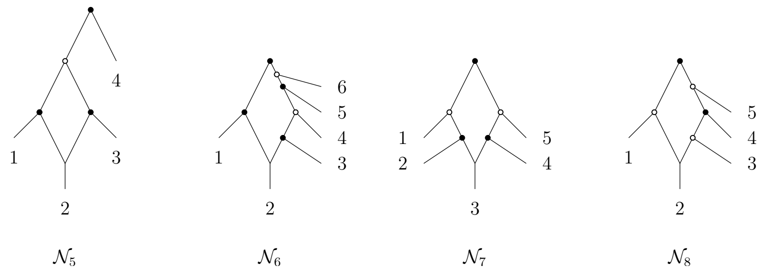


Figure 2.7: The networks $\mathcal{N}_i$, $i = 5, 6, 7, 8$, considered in Table 2.2

Theorem 2.4.1. *Let δ be a symbolic 3-dissimilarity on X. Then the following statements are equivalent (where in (iii)-(v) the input to algorithm* Network-Popping *is δ):*

(i) δ is level-1 representable.
(ii) δ satisfies conditions (P1) - (P8).
(iii) Network-Popping *returns a labelled level-1 network which is unique up to isomorphism.*
(iv) Network-Popping *returns a level-1 representation for δ.*
(v) Network-Popping *returns a semi-discriminating level-1 representation for δ.*

Proof. (i) ⇒ (ii): This is an immediate consequence of Lemma 2.3.1, Proposition 2.3.3, the remark preceding Proposition 2.3.7 and the observation preceding Table 2.2.

(ii) ⇒ (iii): Assume that δ satisfies Properties (P1) - (P8). Then algorithm Find-Cycles first constructs the graph $\mathcal{C}(\delta)$ and then finds for each connected component K of $\mathcal{C}(\delta)$ the pair (H_K, R'_K). Since algorithm Build-Cycle relies on Properties (P3), (P4), (P6) - (P8) being satisfied, it follows that Build-Cycle constructs for each pair (H_K, R'_K), K a connected component of $\mathcal{C}(\delta)$, a labelled simple level-1 network as specified in the output of Build-Cycle. By construction, the labelled graph $\mathcal{N} = (N, t)$ returned by algorithm Network-Popping is clearly a labelled phylogenetic network. Since, in view of the while loop of that algorithm starting at line 7, no two cycles in N can share a vertex it follows that N is in fact a level-1 network. Proposition 2.3.10 combined with the observation that in none of our four algorithms we have to break a tie implies that $\mathcal{N}$ is unique up to isomorphism.

(iii) ⇒ (iv): This is trivial in view of Proposition 2.3.11.

(iv) ⇒ (v): Suppose algorithm Network-Popping returns a level-1 representation $\mathcal{N}$ for δ. To see that $\mathcal{N}$ is in fact semi-discriminating, note that algorithms Vertex-Growing and Build-Cycle return a discriminating symbolic representation and a discriminating level-1 representation for its input symbolic 3-dissimilarity, respectively. In combination it follows that $\mathcal{N}$ must be semi-discriminating.

(v) ⇒ (i): This is trivial. □

As suggested by the two semi-discriminating level-1 representations $\mathcal{N}_1$ and $\mathcal{N}_3$ for $\delta_{\mathcal{N}_1}$ depicted in Figure 2.2, the output of algorithm Network Popping when given a level-1 representable symbolic 3-dissimilarity δ need not be the labelled level-1 network that induced δ. To help clarify the relationship between both networks, we require further terminology.

Suppose that (N, t) is a labelled level-1 network. Then we say that a cycle C of N is *weakly labelled* if there exists at least one vertex v on either side of C such that $t(v) \neq t(r(\mathcal{C}))$. More generally, we call a labelled level-1 network (N, t) *weakly labelled* if every cycle of N is weakly labelled. For example, the labelled level-1 network $\mathcal{N}_2$ pictured in Figure 2.2 is weakly labelled (but not semi-discriminating) whereas the network $\mathcal{N}_3$ depicted in Figure 2.2 is semi-discriminating but not weakly labelled.

Armed with this definition, we can characterize weakly labelled cycles as follows.

Lemma 2.4.2. *Let $\mathcal{N} = (N, t)$ be a labelled level-1 network, and let C be a cycle of N. Then C is weakly labelled if and only if there exists some $x \in H(C)$ and $y, z \in R(C) - H(C)$ lying on different sides of C such that $x||yz$ is a $\delta_{\mathcal{N}}$-tricycle. Moreover, $x'||yz$ is a $\delta_{\mathcal{N}}$- tricycle, for all $x' \in H(C)$.*

Proof. Put $\delta = \delta_{\mathcal{N}}$. Assume first that there exists some $x \in H(C)$ and leaves $y, z \in R(C) - H(C)$ that lie on two different sides of C such that $x||yz$ is a δ-tricycle. Then $\delta(x, y, z) = \delta(z, y) = t(r(\mathcal{C}))$. Also $\delta(x, y) = t(v_C(y))$ and $\delta(x, z) = t(v_C(z))$. In view of Table 2.1, $\delta(x, y, z) \notin \{\delta(x, y), \delta(x, z)\}$ and, so, $t(v_C(i)) \neq t(r(C))$, for $i = y, z$.

Conversely, suppose C is weakly labelled. Let $v_1, v_2 \in V(C)$ denote two vertices of N that lie on different directed paths from $r(C)$ to $h(C)$ such that $t(r(C)) \notin \{t(v_1), t(v_2)\}$. Suppose $y, z \in X$ are such that $v_C(y) = v_1$ and $v_C(z) = v_2$. Then $x||yz$ must be a δ-tricycle, for all $x \in H(C)$. Indeed, $\delta(x, y) = t(v_1)$ and $\delta(x, z) = t(v_2)$ holds. Since $\delta(y, z) = \delta(x, y, z) = t(r(C)) \notin \{\delta(x, y), \delta(x, z)\}$, Table 2.1 implies that $x||yz$ is a δ-tricycle.

The remainder of the lemma follows from the fact that, for all $x' \in H(C)$, we have $\delta(x', y, z) = \delta(x, y, z)$, $\delta(x, y) = \delta(x', y)$ and $\delta(x, z) = \delta(x', z)$. □

As a consequence, we can strengthen Proposition 2.3.3 to the following characterization.

Theorem 2.4.3. *If $\mathcal{N} = (N, t)$ is a labelled level-1 network, the connected components of $\mathcal{C}(\delta_{\mathcal{N}})$ are in 1-1 correspondence with the weakly labelled cycles of N.*

Implied by Theorem 2.4.3, we have:

Corollary 2.4.4. *Let δ be a level-1 representable symbolic 3-dissimilarity on X, and let $\mathcal{N} = (N, t)$ be the level-1 representation of δ returned by algorithm* Network-Popping *when applied to δ. Then $\mathcal{N}$ is weakly labelled if and only if, for any level-1 representation $\mathcal{N}' = (N', t')$ of δ, the number of cycles in N equals the number of weakly labelled cycles in N'. In particular, the number of cycles in N is minimal.*

The next algorithm, Transform (Algorithm 5), is not directly part of Network-Popping. Its aim is to build, from a network $\mathcal{N}$, a semi-discriminating, weakly labelled and partially resolved level-1 network $\mathcal{N}' = (N', t')$ such that $\delta_{\mathcal{N}} = \delta_{\mathcal{N}'}$, in time linear in the number of vertices of $\mathcal{N}$. We have:

```
Input: A labelled level-1 network $\mathcal{N} = (N, t)$ on $X$.
Output: A semi-discriminating, weakly labelled, partially resolved level-1 network
        $\mathcal{N}' = (N', t')$ such that $\delta_{\mathcal{N}} = \delta_{\mathcal{N}'}$.
set $\mathcal{N}' = \mathcal{N}$;
while $\mathcal{N}'$ is not semi-discriminating or not weakly labelled or not partially resolved do
    Collapse all arcs $(u, v)$ satisfying $t'(u) = t'(v)$ and such that either $u$ and $v$ belong to
    the same cycle of $N'$ or do not belong to a cycle;
    for All vertices $v$ of a cycle $C$ of degree 4 or more do
        Define a new child $w$ of $v$;
        set $t'(w) = t'(v)$;
        if $v = r(C)$ then
            Redefine the children of $v$ in $C$ as children of $w$;
        end
        else
            Redefine the children of $v$ outside of $C$ as children of $w$;
        end
    end
    for All cycles $C$ of $N'$ such that $(r(C), h(C))$ is an arc of $N'$ do
        Remove the arc $(r(C), h(C))$;
    end
    Suppress all vertices of degree 2;
end
```

Algorithm 5: Transform

Corollary 2.4.5. *Suppose $\mathcal{N}$ is a labelled level-1 network and $\mathcal{N}'$ is the level-1 representation for $\delta_{\mathcal{N}}$ returned by algorithm* Network-Popping. *Then $\mathcal{N}'$ is isomorphic with the labelled level-1 network returned by algorithm* Transform *when given $\mathcal{N}$ as input. In particular, $\mathcal{N}$ and $\mathcal{N}'$ are isomorphic if and only if $\mathcal{N}$ is semi-discriminating, weakly labelled, and partially resolved. Furthermore, if δ is a level-1 representable symbolic 3- dissimilarity, then there exists an unique representation of δ that is semi-discriminating, weakly labelled, and partially resolved.*

2.4.2 Characterizing level-1 representability

In this section, we present a characterization of level-1 representable symbolic 3-dissimilarities on X in terms of level-1 representable symbolic 3-dissimilarities on subsets of X of size $|X|-1$ (Theorem 2.4.8). Combined with the fact that algorithm Network-Popping has polynomial run time, this suggests that Network-Popping might lend itself to studies involving large data sets using a Divide-and-Conquer approach. At the heart of the proof of our characterization lies the following technical lemma which concerns the question under what circumstances the restriction of a level-1 representable symbolic 3-dissimilarity δ on X is itself level-1 representable. Central to its proof is the fact that $|X| \neq 4$ since, in general, a symbolic 3-dissimilarity δ on a set X of size 4 need not be level-1 representable but the restriction of δ to any subset of size 3 is level-1 representable. An example for this is furnished by the symbolic 3-dissimilarity δ on $X = \{x, y, z, u\}$, given by $\delta(x, y, z) = \delta(y, z, u) = \delta(x, y) = \delta(y, z) = \delta(z, u) \neq \delta(x, z) = \delta(x, u) = \delta(y, u) = \delta(x, z, u) = \delta(x, y, u)$.

Using the assumptions and definitions for the elements x, y, and z, and the sets H, S_z, and S_y made in algorithm Build-Cycle, we have the following result.

Lemma 2.4.6. *Suppose δ is a symbolic 3-dissimilarity on X satisfying Properties (P1), (P2), (P4), and (P6), $x||yz$ is the δ-tricycle chosen in line 2 of algorithm* Build-Cycle, *and $i \in \{y, z\}$. If $u, w \in S_i$ are joined by a direct path from u to w in $TD(S_i, x)$, then either (u, w) is a directed edge of $TD(S_i, x)$ or there exists $v \in S_i$ such that both directed edges (u, v) and (v, w) are contained in $TD(S_i, x)$.*

Proof. By symmetry, we may assume $i = y$. Suppose there exists a directed path $v_0 = u, v_1, \dots, v_k, v_{k+1} = w$, some $k \geq 0$, from u to w in $TD(S_y, x)$ and that (u, w) is not a directed edge on that path. Then $k \geq 1$ and, so, $v_1 \notin \{u, w\}$. It suffices to show that (v_1, w) is a directed edge of $TD(S_y, x)$.

Observe first that, in view of Property (P6), (w, u) is not a directed edge in $TD(S_y, x)$ as otherwise $TD(S_y, x)$ would contain a directed cycle. Combined with the definition of S_y it follows that either $x|uw$ is a δ-triplet or we have a δ-fork on $\{x, u, w\}$. In either case, $\delta(u, x) = \delta(w, x)$ holds. Since (u, v_1) is a directed edge in $TD(S_y, x)$, we also have that $xv_1|u$ is a δ-triplet. Hence, $\delta(v_1, x) \neq \delta(x, u) = \delta(w, x)$ and so we cannot have a δ-fork on $\{x, w, v_1\}$. Since, in view of Property (P4), we cannot have a δ-tricycle on $\{x, w, v_1\}$ either $\delta(w, v_1) = \delta(w, x)$ or $\delta(w, v_1) = \delta(v_1, x)$ follows.

If the first equality holds, then $v_1x|w$ is a δ-triplet and, so, (w, v_1) is a directed edge in $TD(S_y, x)$. Consequently, the directed path $v_1, \dots, v_k, w$ concatenated with that edge forms a directed cycle in $TD(S_y, x)$, which is impossible in view of Property (P6) holding. Thus, $\delta(w, v_1) = \delta(v_1, x)$ must hold. Consequently, $wx|v_1$ is a δ-triplet and, so, (v_1, w) is an edge in $TD(S_y, x)$, as required. □

To establish the main result of this section (Theorem 2.4.8), we need to be able to distinguish between the sets defined in lines 8 and 9 of algorithm Build-Cycle when given a symbolic 3-dissimilarity δ on X and the restriction $\delta|_Y$ of δ to a subset $Y \subseteq X$ with $|Y| \geq 3$. To this end, we augment for a symbolic 3-dissimilarity κ on X the definition of those sets by writing $S_i(\kappa)$ rather than S_i, $i = y, z$.

Observe first that if δ is level-1 representable and $Y \subseteq X$ such that $|Y| \geq 3$, then the restriction $\delta|_Y$ of δ to Y is clearly level-1 representable. Indeed, a level-1 representation $\mathcal{N}(\delta|_Y)$ of $\delta|_Y$ can be obtained from a level-1 representation $\mathcal{N}(\delta)$ of δ using the following 2-step process. First, remove all leaves in $X - Y$ and their respective incoming arcs from $\mathcal{N}(\delta)$ and then suppress all resulting degree two vertices. Next, apply algorithm Transform to the resulting network. This begs the question of when level-1 representations of symbolic 3-dissimilarities on subsets of X give rise to a level-1 representation of a symbolic 3-dissimilarity on X. To answer this question which is the purpose of Theorem 2.4.8 we require the next result.

Proposition 2.4.7. *Let δ be a symbolic 3-dissimilarity on X. Then the following statements hold.*

(i) If $|X| \geq 6$ and δ does not satisfy Property (Pi), $i \in \{1, 2, \dots, 8\}$, then there exists some $Y \subseteq X$ with $3 \leq |Y| \leq 5$ such that that property is also not satisfied by $\delta|_Y$.

(ii) If $|X| \geq 6$ and δ is not level-1 representable then there exists some $Y \subseteq X$ with $3 \leq |Y| \leq 5$ such that $\delta|_Y$ is also not level-1 representable.

Proof. (i) The proposition is straightforward to show for Properties (P1) and (P2), since they involve three and four elements of X, respectively. Note that to see Property (Pi), $3 \leq i \leq 8$, we may assume without loss of generality that Properties (Pj), $1 \leq j \leq i-1$, are satisfied by δ. For ease of readability, we put $S_y := S_y(\delta)$.

If δ does not satisfy Property (P3) then there exists a connected component C of $\mathcal{C}(\delta)$ and δ-tricycles $\tau, \tau' \in V(C)$ such that $\delta(L(\tau)) \neq \delta(L(\tau'))$. Without loss of generality, we may assume that τ and τ' are adjacent. Then $|L(\tau) \cap L(\tau')| = 2$. Let $x, y, z \in X$ such that $\tau = x||yz$. Then either $\tau' = x'||yz$ or $\tau' = x||yz'$ where $x', z' \in X$. But then Property (P3) is not satisfied either for δ restricted to the 5-set $Z = \{x, y, z, x', z'\}$.

For the remainder, let (H, R') denote the pair returned by algorithm Find-Cycles when given δ and let $x \in H$ and $y, z \in R'$ such that $x||yz$ is a vertex in the connected component C of $\mathcal{C}(\delta)$ corresponding to (H, R'). Suppose δ does not satisfy Property (P4). Assume first that the second part of Property (P4) is not satisfied. Then if there exists an element u contained in $H \cap S_y$ or in $H \cap S_z$ or in $S_z \cap S_y$ then u is also contained in the corresponding intersections involving the sets $S_y(\delta|_Z) \subseteq S_y$ and $S_z(\delta|_Z) \subseteq S_z$ found by Build-Cycle in its lines lines 8 and 9 for δ restricted to $Z = \{x, y, z, u\}$. Thus, the second part of Property (P4) does not hold for $\delta|_Z$.

Now assume that the first part of Property (P4) does not hold for δ, that is, $S'_i \neq A := \{w \in S_i : \delta(w, x) \neq \delta(y, z)\}$. By symmetry, we may assume without loss of generality that $i = y$. Then since $S'_y \subseteq A$ clearly holds there must exists some $w \in A - S'_y$. Put $U = \{x, y, z, w\}$. Then $w \notin S'_y(\delta|_U)$ as $w \notin S'_y$. However we clearly have that $w \in S_y(\delta|_U)$ and $\delta|_U(w, x) \neq \delta|_U(y, z)$. Thus, the first part of Property (P4) is not satisfied with δ replaced by $\delta|_U$.

If δ does not satisfy Property (P5) then since $y \in R := H \cup S_y \cup S_z$ it follows for $u := y$ and v and w as in the statement of Property (P5) that the restriction of δ to $\{x, u, z, v, w\}$ does not satisfy Property (P5) either.

If δ does not satisfy Property (P6) then either (a) there exist elements $u, u' \in H$ such that $TD(S_y, u)$ and $TD(S_y, u')$ are not isomorphic or (b) there exists some $u \in H$ such that $TD(S_y, u)$ has a directed cycle C.

Assume first that Case (a) holds. Then there must exist distinct vertices v and w in S_y such that (v, w) is a directed edge in $TD(S_y, u)$ but not in $TD(S_y, u')$. With $Z = \{v, u, u', w, z\}$ it follows that $S_v(\delta|_Z) = \{v, w\}$. Since the directed edge (v, w) is clearly contained in the TopDown graph $TD(\{v, w\}, u)$ associated to $\delta|_Z$ but not in the TopDown graph $TD(\{v, w\}, u')$ associated to $\delta|_Z$, Property (P6) is not satisfied for $\delta|_Z$.

Thus, Case (b) must hold. In view of Proposition 2.3.7(i), we may assume that the size of C is three. Hence, the subgraph G of $TD(S_y, u)$ induced by $Z = V(C) \cup \{z, u\}$ also contains a cycle of length 3. Since G coincides with the TopDown graph $TD(V(C), u)$ for $\delta|_Z$ and $|Z| = 5$ holds, it follows that $\delta|_Z$ does not satisfy Property (P6).

If δ does not satisfy Property (P7) then there must exist undirected edges $e = \{a, b\}$ and $e' = \{a', b'\}$ in $CL(H, S_y, S_z)$ such that $\delta(a, b) \neq \delta(a', b')$. Then for at least one of e and e', say e, we must have that $\delta(a, b) \neq \delta(y, z)$. Put $Z = \{x, y, z, a, b\}$. Then since $\{y, z\}$ is also an undirected edge in $CL(H, S_y(\delta|_Z), S_z(\delta|_Z))$ it follows that $\delta|_Z$ does not satisfy Property (P7) either.

Finally, suppose that δ does not satisfy Property (P8). Considering both alternatives in the statement of Property (P8) together, there must exist vertices $u \in S_y$ and $v, w \in S_y \cup H$ such that both (u, v) and (u, w) are directed edges of $CL(H, S_y, S_z)$ and $\delta(u, v) \neq \delta(u, w)$. Independent of whether $v, w \in S_y$ or $v, w \in H$ or $v \in S_y$ and $w \in H$, it follows that either $\delta(u, x) \neq \delta(u, v)$ or $\delta(u, x) \neq \delta(u, w)$. Assume without loss of generality that $\delta(u, x) \neq \delta(u, v)$. Note that (u, x) is also a directed edge in $CL(H, S_y, S_z)$.

If $v \in H$, then $\delta|_Z$ does not satisfy Property (P8) for $Z = \{x, y, z, u, v\}$. So assume $v \notin H$. Then $v \in S_y$. Since (u, v) is a directed edge in $CL(H, S_y, S_z)$ it follows that there exists a directed path P from u to v in $TD(S_y, x)$. By Lemma 2.4.6, either (a) P has a single directed edge or (b) there exists some $v_1 \in S_y$ such that both (u, v_1) and (v_1, v) are directed edges of $TD(S_y, x)$.

If Case (a) holds, then $\delta|_Z$ does not satisfy Property (P8) for $Z = \{x, y, z, u, v\}$. So assume that Case (b) holds. Then $\delta|_{Z'}$ does not satisfy Property (P8) for $Z' = \{x, y, z, u, v, v_1\}$. Since the definition of $TD(S_y, x)$ implies that $xv|v_1$ is a δ-triplet, it follows that $\delta(x, v) \neq \delta(x, v_1)$. Hence, either $\delta(v, x) \neq \delta(v, z)$ or $\delta(v_1, x) \neq \delta(v, z)$. By Properties (P3) and (P4) it follows in the first case that $x||vz$ is a δ-tricycle, and that $x||v_1z$ is a δ-tricycle in the second case. Thus, either

v or v_1 can play the role of y in τ. Consequently, δ restricted to $Z = Z' - \{y\}$ does not satisfy Property (P8).

(ii) This is a straightforward consequence of Theorem 2.4.1 and Proposition 2.4.7(i). □

From there, we have:

Theorem 2.4.8. *Let δ be a symbolic 3-dissimilarity on a set X such that $|X| \geq 6$. Then δ is level-1 representable if and only if for all subsets $Y \subseteq X$ of size $|X| - 1$, the restriction $\delta|_Y$ is level-1 representable.*

Proof. Suppose first that δ is level-1 representable. Then, by the observation preceding Proposition 2.4.7, $\delta|_Y$ is level-1 representable, for all subsets $Y \subseteq X$ of size $|X| - 1$.

Conversely, suppose that X is such that for all subsets $Y \subseteq X$ of size $|X| - 1$, the restriction $\delta|_Y$ is level-1 representable but that δ is not level-1 representable. Then, by Proposition 2.4.7 there exists a subset $Y \subseteq X$ with $|Y| \in \{3, 4, 5\}$ such that $\delta|_Y$ is also not level-1 representable. But then δ restricted to any subset Z of X size $|X| - 1$ that contains Y also is not level-1 representable which is impossible. □

2.5 Conclusion

As we have seen, the work presented in this chapter successfully extends some of the results reviewed in Section 1.3.1, to labelled level-1 networks. As the notion of a network representation of a symbolic 3-dissimilarity relies upon the uniqueness of the lowest common ancestor, it is unlikely that it will be possible to adapt the results to any type of network without first modifying the current definition of a network representation. However, level-1 networks are not the only type of network satisfying this uniqueness property. It is also enjoyed, for example, by tree-child and tree-sibling networks (which we presented in Section 1.1.3). To be exhaustive, it might be of interest to study the class of phylogenetic networks for which any subset of leaves has exactly one lowest common ancestor.

This, however, is unlikely to be straightforward, as the algorithms and the result established along the way in this chapter are all based on the notion of a cycle, and especially on the key property that no two cycles intersect, which characterizes level-1 networks. For this reason, an extension of these results to other types of networks would probably require a perspective different from the one adopted here.

Apart from this extension problem, a further interesting feature of the work presented in this chapter is the introduction of a new mathematical object, which we call a symbolic 3-dissimilarity. Such a map can be seen as a hybrid between a symbolic distance (Section 1.3.1) and a 3-dissimilarity, the latter being a particular case of a k-dissimilarities (Section 1.3.2). Given that both symbolic distances and k-dissimilarities have already been studied for their relations with phylogenetic trees, the questions becomes, how this new notion of a symbolic 3-dissimilarity fits in the context of these existing results. This question is the starting point of the next chapter.

Chap. 3

On symbolic 3-way tree-maps and ultrametrics

Adapted from:

> K. T. Huber, V. Moulton and G. E. Scholz. Three-way symbolic tree-maps and ultrametrics. *Journal of Classification*, in press.

My personal contribution to this work has been the establishment of the main results (apart from Theorem 3.2.3), as well as writing the first draft of the paper, which includes the proofs of all the results presented.

This chapter addresses the question of the representability of a symbolic 3-way map by a phylogenetic tree, both in the rooted and the unrooted case. For both cases, we provide a characterization of those maps that admit such a representation, and highlight some links existing between this problem and other reconstruction problem, such as, in the rooted case, the reconstruction of a phylogenetic tree from a set of triplets.

3.1 Introduction

In Chapter 2, we extended a result (Theorem 1.3.1) on symbolic ultrametric and phylogenetic trees to a result relating a certain type of symbolic 3-dissimilarities to level-1 phylogenetic networks (Theorem 2.4.1). Inspired by Theorem 1.3.4 characterizing tree-like k-dissimilarities, and its rooted equivalent, Theorem 1.3.5, we may then ask the question, how do symbolic 3-dissimilarities relate to phylogenetic trees? This is the question addressed in this chapter, which is based on [39], both in the rooted and unrooted case. As in the previous chapter, the notions presented in Section 1.3.1 are central, and some parallels are also made with the results in Sections 1.3.2 and 1.4.1.

Throughout this chapter, we will consider the notion of a 3-*way symbolic map* on a set X, that is, a map $\delta : \binom{X}{3} \to \mathcal{M}$, where $\mathcal{M}$ is a nonempty set. We successively consider the case of an unrooted (Section 3.2.1) and of a rooted (Section 3.2.2) tree, for which, as we shall see, the notions of displaying a 3-way symbolic map is different.

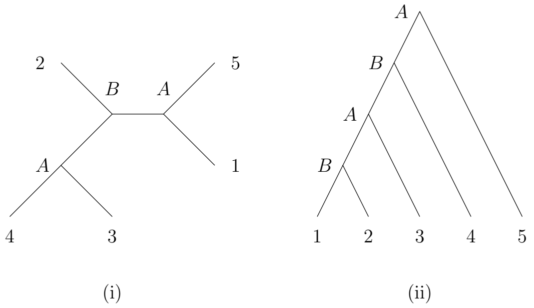


Figure 3.1: Two trees which give rise to (i) a three-way symbolic tree-map and (ii) a three-way symbolic ultrametric.

In the unrooted case, this notion relies on the observation that to each triple of leaves of a tree T there exists a unique vertex contained in all paths between these three leaves. Thus, we can assign to a triple the element labelling this unique vertex. We call maps that arise in this way *3-way symbolic tree-maps*.

In Section 3.2.1 we show that a three-way symbolic tree-map uniquely determines its underlying labelled tree (Proposition 3.2.2), and we also give a 4- and 5-point characterization for such maps (see Theorem 3.2.3). This is analogous for the 4-point condition for tree-metrics presented in Section 1.2.2, and also generalises the conditions Theorem 1.3.4 for determining when a 3-way dissimilarity arises from a tree. To prove this result we introduce a symbolic variant of the Combinatorial Farris transform (see Section 1.1.2), which allows us to apply Theorem 1.3.1 for symbolic ultrametrics. Interestingly, an analogous characterization have been discovered independently in [31], and an alternative one has been proposed, also independently, in [30]. We discuss the differences and similarities between all three approaches at the end of Section 3.2.1.

In Section 3.2.2, we turn our attention to obtaining three-way symbolic maps from rooted trees. One way to define a symbolic map could be to take, for each triple of leaves, the set of symbols consisting of the symbols labelling the least common ancestor of all pairs of leaves in the triple. However, as we shall see in Section 3.2.2, this does not suffice to capture the tree. Even so, if we consider the values assigned to a triple to be *multisets* instead of sets, then we can in fact recover the underlying labelled tree in case $|X| \geq 5$ (Lemma 3.2.7). We call such maps *3-way symbolic ultrametrics*. In Section 3.3.1 we give 3-, 4- and 5-point conditions which ensure that a three-way symbolic map into size 3 multisets of symbols is a three-way symbolic ultrametric. This is somewhat surprising since for three-way dissimilarities, according to Theorem 1.3.5, a 6-point condition is required to ensure that they can be represented by a rooted tree in an analogous way.

In Section 3.3.2, we conclude by considering an alternative approach for deciding whether or not a three-way symbolic map is a symbolic ultrametric. This approach is based on the BUILD algorithm for triplets (see Section 1.4.1). Applying this algorithm to three-ways maps has the advantage that only sets of size three need to be considered to decide when a three-way symbolic map is a tree or symbolic map, as opposed to sets of size up to 5, which could lead to improved computing times in practice.

Throughout this chapter, unless stated otherwise, X denote a finite set of size $n \geq 3$, and M a finite set of symbols of size two or more.

3.2 Two types of maps for two types of trees

We start by considering 3-way symbolic maps that arise from labelled unrooted trees and from labelled rooted trees successively.

3.2.1 3-way symbolic tree-maps for unrooted trees

As mentioned before, for any three leaves x, y, z of a phylogenetic tree T, the three paths between x and y, between x and z and between y and z have exactly one vertex in common. We call this vertex the *median vertex* of x, y and z in T, denoted by $med_T(x, y, z)$. Thus, a labelled unrooted tree $\mathcal{T} = (T, t)$ induces a 3-way symbolic map $\delta_{\mathcal{T}} : \binom{X}{3} \to M$, defined by putting, for all $x, y, z \in X$, $\delta_{\mathcal{T}}(x, y, z) = t(med_T(x, y, z))$. For example, if $\mathcal{T}$ is the tree depicted tree in Figure 3.1(i), we have $\delta_{\mathcal{T}}(1, 3, 5) = A$.

If for a 3-way symbolic map $\delta : \binom{X}{3} \to M$, there exists a labelled unrooted tree $\mathcal{T}$ such that $\delta = \delta_{\mathcal{T}}$, we say that δ is a *3-way symbolic tree-map*, and that $\mathcal{T}$ is a *representation* of δ. We now characterize such maps. To do this, we define a *symbolic Farris transform*, the definition of which is adapted from the Combinatorial Farris transform described in Section 1.1.2 as follows:

Suppose $\mathcal{T} = (T, t)$ is a labelled unrooted tree on X. Pick a leaf $r \in X$, and define a rooted phylogenetic tree T_r on $X - \{r\}$ as follows: direct all edges of T away from r, and remove r and its outgoing edge. This induces a bijection ψ_r

from the set of internal vertices of T to the set of internal vertices of T_r. Hence the map $t_r : V(T_r) \rightarrow M$ which takes any internal vertex v of T_r to M given by $t_r(v) = t(\psi_r^{-1}(v))$ is well-defined, and the pair $\mathcal{T}_r = (T_r, t_r)$ is a labelled rooted tree.

Now, suppose that δ is the 3-way symbolic tree-map induced by $\mathcal{T}$, and that D_r is the symbolic ultrametric on X induced by $\mathcal{T}_r$.

Lemma 3.2.1. *For all $x, y \in X - \{r\}$, we have $D_r(x, y) = \delta(x, y, r)$.*

Proof. It suffices to note that via the symbolic Farris transform, the median vertex of x, y and r in T becomes the last common ancestor of x and y in T_r. Denoting the latter by v, we then have $D_r(x, y) = t_r(v) = t(\psi_r^{-1}(v)) = t(med_T(x, y, r)) = \delta(x, y, r)$. □

Motivated by this observation, for a 3-way symbolic map $\delta : \binom{X}{3} \rightarrow M$ and some $r \in X$, we define

$$\begin{array}{rcl} \delta_r : \binom{X-\{r\}}{2} & \rightarrow & M \\ \{x, y\} & \mapsto & \delta(x, y, r). \end{array}$$

Interestingly, Lemma 3.2.1, together with the Bottom-Up algorithm described in Section 1.3.1 allows us to check if a 3-way symbolic map is a tree-map, and to build its representation if this is the case, as follows. Suppose $\delta : \binom{X}{3} \rightarrow \mathcal{M}$ is a 3-way symbolic map. Pick any $r \in X$. Then, using the Bottom-Up algorithm, we can check whether or not the map δ_r is a symbolic ultrametric. If this is not the case, then by Lemma 3.2.1, δ is not a 3-way symbolic tree-map. Otherwise, if (T, t) is the representation of δ_r returned by Bottom-Up, then we can simply check whether or not this leads to a representation of δ by attaching the leaf r to the root of T. If this is possible then δ is a 3-way symbolic tree-map, otherwise it is not.

Lemma 3.2.1 also allows us to prove a uniqueness result.

Proposition 3.2.2. *Let $\delta : \binom{X}{3} \rightarrow M$ be a 3-way symbolic tree-map. There exists a unique discriminating labelled unrooted tree $\mathcal{T}$ representing δ.*

Proof. Let $r \in X$ and consider the map $\delta_r : \binom{X-\{r\}}{2} \rightarrow M$. By Lemma 3.2.1, δ_r is a symbolic ultrametric, and thus, admits a unique discriminating representation $\mathcal{T}_r$. Moreover, this representation is obtained from a representation of δ, using the symbolic Farris transform. This operation is clearly invertible, and preserves the property of being discriminating. Thus, the labelled unrooted tree $\mathcal{T}$ obtained from $\mathcal{T}_r$ by inverting the symbolic Farris transform is necessarily the only discriminating representation of δ. □

We now state the main result of this section:

Theorem 3.2.3. *Suppose that* $|X| \geq 4$ *and that* $\delta : \binom{X}{3} \to M$ *is a 3-way symbolic map. Then* δ *is a 3-way symbolic tree-map if and only if* δ *satisfies the following two conditions:*

(C1) For all $\{x, y, z, u\} \in \binom{X}{4}$, *either*

$$\delta(x, y, z) = \delta(x, y, u) = \delta(x, z, u) = \delta(y, z, u)$$

or two of these four are equal and so are the remaining two.

(C2) There does not exist $\{x, y, z, u, v\} \in \binom{X}{5}$ *such that*

$$\delta(v, x, y) = \delta(v, y, z) = \delta(v, z, u) \neq \delta(v, z, x) = \delta(v, x, u) = \delta(v, u, y).$$

In order to prove Theorem 3.2.3, we start with a useful lemma.

Lemma 3.2.4. *Suppose that* $|X| \geq 4$ *and let* $\delta : \binom{X}{3} \to M$ *be a 3-way symbolic map satisfying (C1) and (C2). Then for all* $r \in X$, *the map* δ_r *is a symbolic ultrametric.*

Proof. Let $r \in X$. We need to show that δ_r satisfies Properties (U1) and (U2).

To see that δ_r satisfies (U1), consider three elements $x, y, z \in X - \{r\}$. Since δ satisfies (C1) the set $\{\delta(r, x, y), \delta(r, x, z), \delta(r, y, z)\}$ contains at most two distinct elements. As this set is precisely the set $\{\delta_r(x, y), \delta_r(x, z), \delta_r(y, z)\}$, (U1) follows.

To see that (U2) holds, assume for contradiction that there exist four elements $x, y, z, u \in X - \{r\}$ such that $\delta_r(x, y) = \delta_r(y, z) = \delta_r(z, u) \neq \delta_r(z, x) = \delta_r(x, u) = \delta_r(u, y)$. This implies $\delta(r, x, y) = \delta(r, y, z) = \delta(r, z, u) \neq \delta(r, z, x) = \delta(r, x, u) = \delta(r, u, y)$, which is impossible in view of (C2). □

Note that the converse of Lemma 3.2.4 is not true in general. Consider for example the sets $X = \{1, \ldots, n\}$, $n \geq 4$, $M = \{A, B\}$ and the map $\delta : \binom{X}{3} \to M$ defined for $x, y, z \in X$ by $\delta(x, y, z) = A$ if $1 \in \{x, y, z\}$ and $\delta(x, y, z) = B$ otherwise. Clearly, δ does not satisfy (C1), as we have $\delta(1, 2, 3) = \delta(1, 2, 4) = \delta(1, 3, 4) \neq \delta(2, 3, 4)$. However, we have $\delta_1(x, y) = A$ for all $x, y \in X - \{1\}$, which is clearly a symbolic ultrametric. In fact, for any $2 \leq k \leq n$ we have $\delta_k(x, y) = A$ if $1 \in \{x, y\}$ and $\delta_k(x, y) = B$ otherwise and, so, δ_k is also a symbolic ultrametric on $X - \{k\}$.

Armed with Lemma 3.2.4, we can now prove Theorem 3.2.3.

Proof. Assume first that δ is a 3-way symbolic tree-map, and denote by $\mathcal{T} = (T, t)$ its representation. To see that δ satisfies (C1), consider four elements $x, y, z, u \in X$. Two cases may occur. If $med_T(x, y, z) = med_T(x, y, u) = med_T(x, z, u) = med_T(y, z, u)$, it follows immediately that $\delta(x, y, z) = \delta(x, y, u) = \delta(x, z, u) = \delta(y, z, u)$. Otherwise, there exists two pairs, say $\{x, y\}$ and $\{z, u\}$, such that the path between x and y and the path between z and u are disjoint. In this case, we have $med_T(x, y, z) = med_T(x, y, u) \neq med_T(x, z, u) = med_T(y, z, u)$. If $t(med_T(x, y, z)) = t(med_T(x, z, u))$, it follows that $\delta(x, y, z) = \delta(x, y, u) = \delta(x, z, u) = \delta(y, z, u)$. Otherwise, we have $\delta(x, y, z) = \delta(x, y, u) \neq \delta(x, z, u) = \delta(y, z, u)$. Thus, δ satisfies (C1).

To see that δ satisfies (C2), assume for contradiction that there exists $x, y, z, u, v \in X$ such that $\delta(v, x, y) = \delta(v, y, z) = \delta(v, z, u) \neq \delta(v, z, x) = \delta(v, x, u) = \delta(v, u, y)$. We can apply the symbolic Farris transform to $\mathcal{T}$ and v, thus obtaining a labelled rooted tree $\mathcal{T}_v$ which, by Lemma 3.2.1, is a representation of δ_v, implying that δ_v is a symbolic ultrametric. But, by definition, δ_v satisfies $\delta_v(x, y) = \delta_v(y, z) = \delta_v(z, u) \neq \delta_v(z, x) = \delta_v(x, u) = \delta_v(u, y)$, which contradicts (U2).

Conversely, assume that δ satisfies Properties (C1) and (C2), and let $r \in X$. By Lemma 3.2.4, the map δ_r is a symbolic ultrametric. Thus there exists a labelled rooted tree $\mathcal{T}_r = (T_r, t_r)$ on $X - \{r\}$ representing δ_r. Consider the labelled unrooted tree $\mathcal{T} = (T, t)$ on X defined as follows. First, add a new vertex r to T_r and the edge (ρ_{T_r}, r). Then consider all edges in the resulting tree to be undirected. Let $t : V_{int}(T) \to M$ denote the map given by $t(v) = t_r(v)$, for all $v \in V_{int}(T)$. We claim that for all $\{x, y, z\} \in \binom{X}{3}$, we have $\delta(x, y, z) = t(med_T(x, y, z))$, that is, $\mathcal{T}$ is a representation of δ. To prove this it suffices to consider two cases. Suppose $\{x, y, z\} \in \binom{X}{3}$.

Case 1: $\{x, y, z\} \subseteq X - \{r\}$. Without loss of generality, $\delta_r(x, z) = \delta_r(y, z) = t_r(u)$ and $\delta_r(x, y) = t_r(v)$, where u and v are vertices of T_r, and v is below or equal to u in T_r. In this case $med_T(x, y, z)$ is the image of v in T. By (C1) and since $\delta(x, z, r) = \delta(y, z, r)$, we have $\delta(x, y, z) = \delta(x, y, r) = t_r(v) = t(med_T(x, y, z))$. Thus, $\mathcal{T}$ is a representation of δ in this case.

Case 2: $r \in \{x, y, z\}$, say $r = z$. If we denote by v the last common ancestor of x and y in T_r, then $med_T(x, y, z)$ is the image of v in T. Hence $\delta(x, y, r) = \delta_r(x, y) = t_r(v) = t(med_T(x, y, r))$. Thus, $\mathcal{T}$ is a representation of δ in this case, too. □

As for Theorem 1.3.1, an equivalent for Theorem 3.2.3 appears in [31] in the context of game theory. Moreover, an alternative characterization for 3-way symbolic tree-maps can also be found in [30]. The arguments found in [31] bear similarities to the ones presented here, as this work also relies, for a symbolic 3-way map δ on X, on the symbolic 2-way maps D_r, $r \in X$ on $X - \{r\}$ introduced above. This leads to a four-point and a five-point conditions for δ to be tree-like, that are respectively equivalent to conditions (C1) and (C2).

The approach developed in [30], however, is quite different. In this paper, a 3-way symbolic map δ on a set Y of size five is seen as edge-labelled graphs (see Section 1.3.1 for a reminder on such graphs) (H_δ, d_δ) on Y, where $d_\delta(x,y)$ is defined, for $x, y \in X$ distinct, as $\delta(Y - \{x,y\})$. From there an exhaustive study of all such graphs leads to two conditions for a symbolic 3-way map to be tree-like. As it turns out, the first condition is precisely (C1). The second condition, however, is stated as follows:

(C2)' There does not exist $\{x,y,z,u,v\} \in \binom{X}{5}$ such that the image set of $\delta|_{\{x,y,z,u,v\}}$ contains exactly two elements, and δ maps five subsets of $\{x,y,z,u,v\}$ of size three to one of these elements, and the remaining five to the second one.

Interestingly, Properties (C2) and (C2)' are not equivalent. More precisely, we can easily see that (C2)' implies (C2), but the converse is not true in general. However, the equivalence holds under the assumption that Property (C1) is satisfied.

3.2.2 3-way symbolic ultrametrics for rooted trees

In the last section, we considered the problem of deciding when a 3-way symbolic map arises from a labelled unrooted tree. In this section, we start to consider that problem for their rooted counterparts. As we shall see, it suffices to decide this for all subsets of X of size 5. In the context of this, note that if we consider 3 distinct leaves x, y, z of a rooted phylogenetic tree T on X we can naturally identify two internal vertices of the tree given by the set $\{lca_T(x,y), lca_T(x,z), lca_T(y,z)\}$ (in contrast to unrooted phylogenetic trees where we can identify only one, namely the median of the 3 leaves). A natural approach to obtain a 3-way symbolic map δ from a labelled rooted tree $\mathcal{T} = (T,t)$ might therefore be to take $\delta(x,y,z)$ to be the set $\{t(lca_T(x,y)), t(lca_T(x,z)), t(lca_T(y,z))\}$ for $x, y, z \in X$ distinct. However, as can be seen in Figure 3.2 such a map does not necessarily uniquely capture $\mathcal{T}$. For this reason, we shall consider instead maps to multisets.

To formalize this, let $\mathcal{M} = M^3$ denote the set of multisets $\{a,b,c\}$ with $a,b,c \in M$. As it will be useful later on, we shall also sometimes denote an element in $\mathcal{M}$ as a sum. So, for example, for the element $\{a,a,b\} \in \mathcal{M}$ with $a, b \in M$, we sometimes also write $2a+b$. By abuse of terminology, we say that a labelled rooted tree $\mathcal{T} = (T,t)$ on X *represents* a 3-way symbolic map $\delta : \binom{X}{3} \to \mathcal{M}$ (or that δ is *represented* by $\mathcal{T}$) if for all distinct $x, y, z \in X$, we have

$$\delta(x,y,z) = \{t(\mathrm{lca}_T(x,y)), t(\mathrm{lca}_T(x,z)), t(\mathrm{lca}_T(y,z))\}.$$

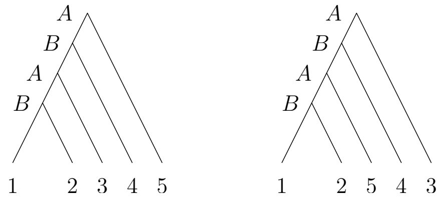


Figure 3.2: Two labelled rooted trees on $X = \{1, 2, 3, 4, 5\}$ with labelling maps t and t' on $M = \{A, B\}$, respectively, for which the sets $\{t(lca_T(x, y)), t(lca_T(x, z)), t(lca_T(y, z))\}$ and $\{t'(lca_T(x, y)), t'(lca_T(x, z)), t'(lca_T(y, z))\}$ coincide, for any three elements $x, y, z \in X$ distinct.

If such a labelled rooted tree $\mathcal{T}$ exists, we say that δ is a *3-way symbolic ultrametric (on X)* and that $\mathcal{T}$ is a *representation* for δ. For example, if $\mathcal{T}$ is the labelled rooted tree depicted in Figure 3.1(ii), we have $\delta_{\mathcal{T}}(1, 3, 5) = \{A, A, A\}$.

Note that we can think of δ as a symbolic analogue of a tree-like symbolic 3-dissimilarity (see Section 1.3.2 and Theorem 1.3.5). Also note that if δ is a 3-way symbolic ultrametric on X then its representation $\mathcal{T}$ canonically induces a symbolic ultrametric D_δ on X in view of Theorem 1.3.1. For the convenience of the reader we picture for $M = \{A, B, C\}$ in Figure 3.3 all seven discriminating labelled rooted trees $\mathcal{T}_i$, $1 \leq i \leq 7$, on $\{1, 2, 3, 4\}$, up to isomorphism. Furthermore, we present in Table 3.1 the values of the map $\hat{\delta}_i : \binom{X}{3} \to \mathcal{M}$ that is represented by $\mathcal{T}_i$.

The following useful observation follows immediately from Property (U1).

Lemma 3.2.5. *Let $\delta : \binom{X}{3} \to \mathcal{M}$ be a 3-way symbolic ultrametric. Then, for any three distinct elements $x, y, z \in X$, the number of distinct elements in the multiset $\delta(x, y, z)$ is at most two.*

Now, for a subset Y of X of size four or more, let $\delta|_Y$ denote the restriction of δ to $\binom{Y}{3} \subseteq \binom{X}{3}$. Clearly, if δ is a 3-way symbolic ultrametric, then $\delta|_Y$ is a 3-way symbolic ultrametric for all subsets $Y \subseteq X$ with $|Y| \geq 4$. Indeed, if $\mathcal{T}$ is a representation of δ, then the subtree $\mathcal{T}_Y$ of $\mathcal{T}$ induced by Y is a representation of $\delta|_Y$. Furthermore, we obtain a discriminating representation of $\delta|_Y$ by collapsing all edges of $\mathcal{T}_Y$ both of whose end vertices have the same label.

Intriguingly, except for $\hat{\delta}_3$, the maps $\hat{\delta}_i$, $i \in \{1, \dots, 7\} - \{3\}$ uniquely capture $\mathcal{T}_i$ in the sense that $\mathcal{T}_i$ is the unique labelled rooted tree that represents $\hat{\delta}_i$. In view of this, we say for a subset $Y \subseteq X$ of size four that $\delta|_Y$ is *of type* $\hat{\delta}_i$, $i \in \{1, \dots, 7\}$ if there exists a bijection between Y and $\{1, 2, 3, 4\}$ and a bijection between the image of $\delta|_Y$ and the image of $\hat{\delta}_i$ such that $\delta|_Y$ and $\hat{\delta}_i$ coincide up to these bijections. Since Table 3.1 is exhaustive, we have:

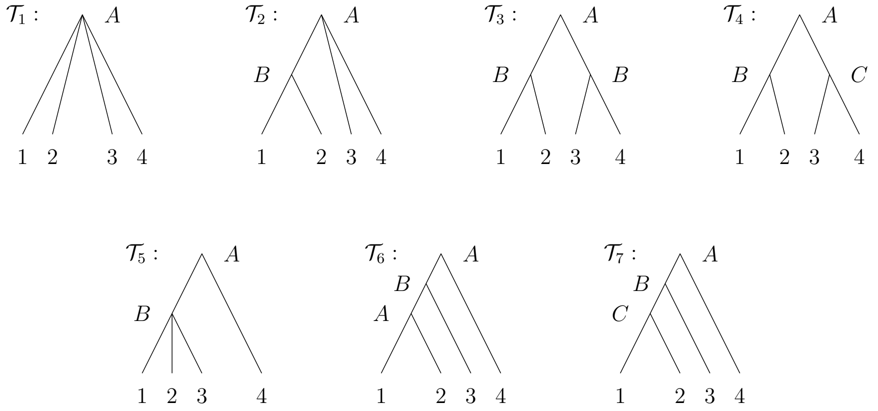


Figure 3.3: All possible discriminating labelled rooted trees $\mathcal{T}_i$, $1 \leq i \leq 7$, on $\{1, 2, 3, 4\}$, up to a relabelling of the leaves.

i	1	2	3	4	5	6	7
$\hat{\delta}_i(1,2,3)$	3A	2A+B	2A+B	2A+B	3B	A+2B	2B+C
$\hat{\delta}_i(1,2,4)$	3A	2A+B	2A+B	2A+B	2A+B	3A	2A+C
$\hat{\delta}_i(1,3,4)$	3A	3A	2A+B	2A+C	2A+B	2A+B	2A+B
$\hat{\delta}_i(2,3,4)$	3A	3A	2A+B	2A+C	2A+B	2A+B	2A+B

Table 3.1: For $1 \leq i \leq 7$ and $M = \{A, B, C\}$, we present the values of the map $\hat{\delta}_i$ represented by the labelled rooted trees $\mathcal{T}_i$ in Figure 3.3. The trees $\mathcal{T}_i$ are given in terms of their index i in the top row.

Proposition 3.2.6. *Suppose that $|X| \geq 4$, that $\delta : \binom{X}{3} \to \mathcal{M}$ is a 3-way symbolic map, and that $Y \subseteq X$ is a subset of size four. Then $\delta|_Y : \binom{Y}{3} \to \mathcal{M}$ is a 3-way symbolic ultrametric on Y if and only if there exists $i \in \{1, \dots, 7\}$ such that $\delta|_Y$ is of type $\hat{\delta}_i$. Moreover, if $i \neq 3$, the representation of $\delta|_Y$ is unique.*

In the last result we have seen that a 3-way symbolic ultrametric on a set of size four may have more than one representation by a labelled tree. However, as we shall now show this does not happen for sets of size five.

Lemma 3.2.7. *If Y is a set of size five and $\delta : \binom{Y}{3} \to \mathcal{M}$ is a 3-way symbolic ultrametric on Y, then δ has a unique discriminating representation.*

Proof. It suffices to show that if δ is a 3-way symbolic ultrametric on Y, then the symbolic ultrametric $D = D_\delta : \binom{Y}{2} \to M$ induced by δ is unique. Since δ is a 3-way symbolic ultrametric on Y, there exists a subset Y_0 of Y such that $\delta|_{Y_0}$ is

$\delta(1,2,3)$	2A+B	$\delta(1,4,5)$	2A+B
$\delta(1,2,4)$	2A+B	$\delta(2,3,4)$	3A
$\delta(1,2,5)$	3B	$\delta(2,3,5)$	2A+B
$\delta(1,3,4)$	3A	$\delta(2,4,5)$	2A+B
$\delta(1,3,5)$	2A+B	$\delta(3,4,5)$	A+2B

Table 3.2: For $M = \{A, B\}$ and $X = \{1, 2, 3, 4, 5\}$ we present a 3-way symbolic map $\delta : \binom{X}{3} \to \mathcal{M}$ which is not a 3-way symbolic ultrametric on X but whose restriction to any subset $Y \subset X$ of size four is a 3-way symbolic ultrametric on Y.

not of type $\hat{\delta}_3$. Thus, the discriminating representation $\mathcal{T}_0$ of $\delta|_{Y_0}$ is unique, and so is the restriction D_{Y_0} of all symbolic ultrametric $D : \binom{Y}{2} \to M$ induced by δ. Then, the value of $D(x_0, x)$, where $x \in Y_0$ and x_0 is the unique element contained in $Y - Y_0$ is uniquely determined by δ and D_{Y_0}. □

Note that, as the example in Table 3.2 shows, it is not true in general that a 3-way symbolic map δ that restricts to a 3-way symbolic ultrametric on all subsets Y of X of size four is a 3-way symbolic ultrametric on X. However, as promised above, we now show that considering sets of size five is enough to ensure this.

Theorem 3.2.8. *Suppose that $|X| \geq 5$ and that $\delta : \binom{X}{3} \to \mathcal{M}$ is a 3-way symbolic map. Then, δ is a 3-way symbolic ultrametric if and only if $\delta|_Y$ is a 3-way symbolic ultrametric for all $Y \subseteq X$ of size five.*

Proof. The fact that a 3-way symbolic ultrametric on X restricts to such an ultrametric on all subsets of X of size five is clear.

Conversely, assume that $\delta|_Y$ is a 3-way symbolic ultrametric for all $Y \subseteq X$ of size five. For such a set Y, we denote by $\mathcal{T}_Y = (T_Y, t_y)$ the unique (by Lemma 3.2.7) discriminating labelled tree that represents $\delta|_Y$ and by D_Y the symbolic ultrametric induced by $\mathcal{T}_Y$.

Clearly, if there exists a map $D : \binom{X}{2} \to M$ such that for all subset $Y \subseteq X$ of size five, then the restriction of D to $\binom{Y}{2}$ coincides with D_Y, then D satisfies $\delta(x, y, z) = \{D(x, y), D(x, z), D(y, z)\}$. Moreover, since D_Y is a symbolic ultrametric on any subset $Y \subseteq X$ of size five, and given that the property of being a symbolic ultrametric is based on a four-point condition, we have that such a map D, if it exists, is also a symbolic ultrametric. Thus, if D is well-defined, then δ is a 3-way symbolic ultrametric.

To show that D is well-defined, assume for contradiction that there exists x and y in X and two distinct subsets Y_1 and Y_2 of X of size five, both containing x and y, such that $D_{Y_1}(x, y) \neq D_{Y_2}(x, y)$. We may assume without loss of generality that $I = Y_1 \cap Y_2$ has size four. Moreover, we claim that x, y, Y_1 and Y_2 can be chosen such that $\delta|_I$ is not of the form $\hat{\delta}_3$, as defined in Table 3.1.

To prove this claim, consider the case where $\delta|_I$ is of type $\hat{\delta}_3$ (otherwise, the claim trivially holds). Assume $Y_1 = \{x, y, z, t, u_1\}$ and $Y_2 = \{x, y, z, t, u_2\}$, which implies $I = \{x, y, z, t\}$. Both the subtree of T_{Y_1} induced by I and the subtree of T_{Y_2} induced by I are of the form T_3 in Figure 3.3, and are not isomorphic. We can assume that one has cherries x, y and t, z and the other has cherries x, z and t, y. Then, we have not only that $D_{Y_1}(x, y) \neq D_{Y_2}(x, y)$, but also that $D_{Y_1}(x, z) \neq D_{Y_2}(x, z)$, $D_{Y_1}(z, t) \neq D_{Y_2}(z, t)$, and $D_{Y_1}(y, t) \neq D_{Y_2}(y, t)$. Moreover, it is easy to check that there exists a subset $Y^* \subset I$ of size three such that neither $\delta|_{Y^* \cup \{u_1\}}$ nor $\delta|_{Y^* \cup \{u_2\}}$ is of type $\hat{\delta}_3$.

Since Y^* is a subset of I of size three and in view of the four inequalities listed above, there exists two elements $x', y' \in Y^*$ such that $D_{Y_1}(x', y') \neq D_{Y_2}(x', y')$. If we denote by Y' the set $Y^* \cup \{u_1\} \cup \{u_2\}$, we have that both $Y' \cap Y_1$ and $Y' \cap Y_2$ have size four, and that at least one of $D_{Y'}(x', y') \neq D_{Y_1}(x', y')$ or $D_{Y'}(x', y') \neq D_{Y_2}(x', y')$ holds. If the first inequality holds, the claim is then satisfied for x', y', Y' and Y_1. Otherwise, it is satisfied for x', y', Y' and Y_1, which completes the proof of the claim.

Now, in light of the claim, the representation $\mathcal{T}_I$ of $\delta|_I$ is unique, and so is the symbolic ultrametric D_I that it induces. Moreover, D_I is precisely the restriction of D_{Y_1} to I, and the restriction of D_{Y_2} to I. In particular, we have $D(x, y) = D_{Y_1}(x, y)$ and $D(x, y) = D_{Y_2}(x, y)$, which contradicts $D_{Y_1}(x, y) \neq D_{Y_2}(x, y)$. □

3.3 Characterizations of 3-way symbolic ultrametrics

To characterize 3-way symbolic ultrametrics on sets of size five or more, two distinct approaches can be considered. We now present both of them successively.

3.3.1 A five-point characterization

We focus here on using the result in the last section to derive conditions for characterizing 3-way symbolic ultrametrics that are analogous to conditions (U1) and (U2) for symbolic ultrametrics.

In the following, we shall consider expressions of the form $\sum_{m \in M} \alpha_m m$ where α_m is a real, for all $m \in M$, which arise when we take linear combinations of multisets in $\mathcal{M}$. We shall say that such an expression $\sum_{m \in M} \alpha_m m$ is *valid for* M if the coefficient for each element in M is contained in $\mathbb{N}$. For example, for $M = \{a, b\}$, if $S_1 = 2a + b$, $S_2 = 2b + a$ and $S_3 = 3a$ are multisets in $\mathcal{M}$, then we have $\frac{1}{3}(S_1 + S_2) = a + b$, which is valid for M, but $S_3 - S_1 = a - b$ and $\frac{1}{2}(S_1 + S_3) = \frac{5}{2}a + \frac{1}{2}b$ which are not valid for M. Note that in this notation, the sum corresponds to the union of multisets, and the difference, if valid, corresponds to the removal of some elements from a multiset.

Now, suppose that $\delta : \binom{X}{3} \to \mathcal{M}$ is a 3-way symbolic map. Let $Y = \{x, y, z, u, v\}$ be a subset of X. Let $\nu_Y(\delta)$ denote the vector

$$(\delta(x, y, z), \delta(x, y, u), \ldots, \delta(z, u, v)).$$

In addition, suppose that $D_Y : \binom{Y}{2} \to M$ is a map such that

$$\delta(a, b, c) = \{D_Y(a, b), D_Y(a, c), D_Y(b, c)\}$$

for all $a, b, c \in Y$, and let $\mu_Y(\delta)$ denote the vector

$$(D_Y(x, y), D_Y(x, z), \ldots, D_Y(u, v)).$$

By definition of D_Y, it is straightforward to check that $A\mu_Y(\delta) = \nu_Y(\delta)$, where

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Note that the matrix A is invertible (see [36]) with inverse

$$A^{-1} = \frac{1}{6}\begin{pmatrix} 2 & 2 & 2 & -1 & -1 & -1 & -1 & -1 & -1 & 2 \\ 2 & -1 & -1 & 2 & 2 & -1 & -1 & -1 & 2 & -1 \\ -1 & 2 & -1 & 2 & -1 & 2 & -1 & 2 & -1 & -1 \\ -1 & -1 & 2 & -1 & 2 & 2 & 2 & -1 & -1 & -1 \\ 2 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & -1 & -1 \\ -1 & 2 & -1 & -1 & 2 & -1 & 2 & -1 & 2 & -1 \\ -1 & -1 & 2 & 2 & -1 & -1 & -1 & 2 & 2 & -1 \\ -1 & -1 & 2 & 2 & -1 & -1 & 2 & -1 & -1 & 2 \\ -1 & 2 & -1 & -1 & 2 & -1 & -1 & 2 & -1 & 2 \\ 2 & -1 & -1 & -1 & -1 & 2 & -1 & -1 & 2 & 2 \end{pmatrix}$$

Consider the product $\mu_Y(\delta) = A^{-1}\nu_Y(\delta)$. Then, as the rows of A^{-1} are indexed by pairs of elements in Y, it is straightforward to check by considering the $\{p, q\}$th row for $p, q \in Y$ distinct, and putting $\{e, f, g\} = Y - \{p, q\}$, that the multiset

$$S^Y_{p,q}(\delta) := \frac{1}{6}(2(\delta(p,q,e)+\delta(p,q,f)+\delta(p,q,g)+\delta(e,f,g))-\sum_{a,b\in Y-\{p,q\}}(\delta(p,a,b)+\delta(q,a,b)))$$

reduces to $S^Y_{p,q}(\delta) = \{D_Y(p,q)\}$. Using the above identity as a definition for $S^Y_{p,q}(\delta)$ where Y is a set of size five $\delta : \binom{X}{3} \to \mathcal{M}$ is a 3-way symbolic map and $p, q \in Y$ are distinct we have the following result.

Proposition 3.3.1. *Suppose that $|X| \geq 5$, that $\delta : \binom{X}{3} \to \mathcal{M}$ is a 3-way symbolic map, and that $Y \subseteq X$ has size five. There exists a map $D^Y : \binom{Y}{2} \to M$ such that $\delta(a,b,c) = \{D^Y(a,b), D^Y(a,c), D^Y(b,c)\}$ for all $a, b, c \in Y$ if and only if for all $p, q \in Y$ distinct, $S^Y_{p,q}(\delta)$ is valid for M. In particular, $S^Y_{p,q}(\delta)$ is a singleton set in this case.*

Proof. For $p, q \in Y$ distinct, put $S_{p,q}(\delta) = S^Y_{p,q}(\delta)$. Suppose first that the map D^Y exists. Then in view of the discussion preceding the proposition, it follows that $S_{p,q}(\delta)$ is valid for M for all $p, q \in Y$ distinct, as $\{D_Y(p,q)\} = S_{p,q}(\delta)$ holds in this case.

To see the converse, assume that $S_{p,q}(\delta)$ is valid for M for all $\{p, q\} \subset Y$. Let $p, q \in Y$ distinct. We claim that $S_{p,q}(\delta)$ is a singleton set. To see this, put $\mathcal{A} = 2(\delta(p,q,e)+\delta(p,q,f)+\delta(p,q,g)+\delta(e,f,g))$ and $\mathcal{B} = \sum_{a,b\in Y-\{p,q\}}(\delta(p,a,b)+\delta(q,a,b))$. Then since $S_{p,q}(\delta)$ is valid for M, every element in $\mathcal{B}$ must also be an element in $\mathcal{A}$. Hence, $S_{p,q}(\delta)$ must contain $\frac{1}{6}(|\mathcal{A} - \mathcal{B}|) = 1$ element as $|\mathcal{A}| = 24$ and $|\mathcal{B}| = 18$. This proves the claim. Let $s_{p,q}(\delta)$ denote the unique element in $S_{p,q}(\delta)$. Then it is straightforward to see that the map $D^Y : \binom{Y}{2} \to M$ defined by putting $D^Y(p,q) = s_{p,q}(\delta)$, for all $p, q \in Y$ distinct, satisfies the stated property. □

We now present conditions for characterizing when a 3-way symbolic map is a 3-way symbolic ultrametric. For $\Sigma \in \mathcal{M}$, we define the elements $m(\Sigma)$ and $n(\Sigma)$ of M as follows:

- If Σ contains a single element $A \in M$ repeated three times, we put $m(\Sigma) = n(\Sigma) = A$.
- If Σ contains two distinct elements, we define $m(\Sigma)$ as the element of Σ appearing twice and $n(\Sigma)$ as the element appearing only once.
- If Σ contains three distinct elements, we put $m(\Sigma) = n(\Sigma) = \emptyset$.

Note that if Σ contains two or fewer distinct elements, then $\Sigma = \{m(\Sigma), m(\Sigma), n(\Sigma)\}$.

Theorem 3.3.2. *Suppose that $|X| \geq 5$ and that $\delta : \binom{X}{3} \to \mathcal{M}$ is a 3-way symbolic map. Then δ is a 3-way symbolic ultrametric if and only if the following hold:*

(D1) For all subsets $Y \subseteq X$ of size five and all $x, y \in Y$ distinct, the multiset $S^Y_{x,y}(\delta)$ is valid for M.

(D2) For all $x, y, z \in X$, $\delta(x, y, z)$ contains at most two distinct elements.

(D3) For all $x, y, z, u \in X$ with $\delta(x, y, z) = \delta(y, z, u) \neq \delta(x, y, u) = \delta(x, z, u)$ holding, we have $m(\delta(x, y, z)) = m(\delta(x, y, u))$.

Proof. Assume first that δ is a 3-way symbolic ultrametric. By Theorem 3.2.8 and Proposition 3.3.1 it follows that Properties (D1) and (D2) must hold. To see that Property (D3) holds too let $x, y, z, u \in X$ be such that $\delta(x, y, z) = \delta(y, z, u) \neq \delta(x, y, u) = \delta(x, z, u)$. Since $\delta|_{\{x,y,z,u\}}$ is a 3-way symbolic ultrametric, Proposition 3.2.6 combined with Table 3.1 implies that $\delta|_{\{x,y,z,u\}}$ is either of type $\hat{\delta}_3$ and $\hat{\delta}_5$. Clearly, $m(\hat{\delta}_i(x, y, z)) = m(\hat{\delta}_i(x, y, u))$ holds for $i = 3, 5$ and, so, Property (D3) follows.

Conversely, assume that δ satisfies Properties (D1) - (D3). Consider a subset $Y \subseteq X$ of size five. By Proposition 3.3.1, there exists a map $D^Y : \binom{Y}{2} \to M$ such that $\delta(x, y, z) = \{D^Y(x, y), D^Y(x, z), D^Y(y, z)\}$ for all $x, y, z \in Y$. We claim that D^Y is a symbolic ultrametric. For this it suffices to show that D^Y satisfies Property (U2) as Property (U1) is a direct consequence of Property (D1). To see that D^Y satisfies Property (U2), assume for contradiction that there exist $x, y, z, u \in Y$ such that $D^Y(x, y) = D^Y(y, z) = D^Y(z, u) \neq D^Y(z, x) = D^Y(x, u) = D^Y(u, y)$. Put $A = D^Y(x, y)$ and $B = D^Y(z, x)$. Then $\delta(x, y, z) = \delta(y, z, u) = 2A + B \neq A + 2B = \delta(x, y, u) = \delta(x, z, u)$. Since, $m(\delta(x, y, z)) = A \neq B = m(\delta(x, y, u))$ also holds this is impossible in view of Property (D3). Thus, D^Y also satisfies Property (U2) and, so, is a symbolic ultrametric, as claimed.

Since D^Y is a symbolic ultrametric there exists a labelled rooted tree $\mathcal{T}$ that represents D^Y. Combined with the definition of D^Y it follows that $\mathcal{T}$ also represents $\delta|_Y$. Thus, $\delta|_Y$ is a 3-way symbolic ultrametric and, so, $\delta|_Y$ is a 3-way symbolic ultrametric for all subsets $Y \subseteq X$ with $|Y| = 5$. By Theorem 3.2.8, it follows that δ is a 3-way symbolic ultrametric. □

Note that Properties (D1) - (D3) are independent of each other. Indeed, that Property (D2) is independent of Properties (D1) and (D3) and that Property (D3) is independent of Properties (D1) and (D2) is a direct consequence of the fact that Properties (U1) and (U2) are independent of each other. To see that Property (D1) is independent of Properties (D2) and (D3), consider the 3-way symbolic map $\delta : \binom{X}{3} \to \mathcal{M}_{\{A,B\}}$ defined, for all $x, y, z \in X$, by putting $\delta(x, y, z) = 2A + B$. The map δ always satisfies (D2) and (D3), but if $|X| \geq 5$, δ does not satisfy (D1).

3.3.2 Triplets as an alternative

In this section we are interested in determining when a 3-way symbolic map δ on X, $|X| \geq 5$, is a a symbolic ultrametric. Clearly, using the conditions given in the results above this can be done by examining every subset X of size 5. However, we now show how to do this using a triplet-based approach, which essentially reduces the problem to considering subsets of X of size 3.

The next proposition highlights some straightforward links between triplets and 3-way symbolic ultrametrics. In the following, we denote the set underlying a multiset $\mathcal{A}$ by $\underline{\mathcal{A}}$.

Proposition 3.3.3. *Let $\mathcal{T} = (T, t)$ be a labelled rooted tree that is a discriminating representation for $\delta_{\mathcal{T}}$. For $x, y, z \in X$ distinct:*

- *If $xy|z$ is a triplet displayed by T, then $t(\mathrm{lca}(x, z)) = t(\mathrm{lca}(y, z)) = m(\delta_{\mathcal{T}}(x, y, z))$ and $t(\mathrm{lca}(x, y)) = n(\delta_{\mathcal{T}}(x, y, z))$*
- *If T does not display any triplet on $\{x, y, z\}$, then $|\underline{\delta_{\mathcal{T}}(x, y, z)}| = 1$.*

To state a consequence of Proposition 3.3.3, we say that a labelled tree $\mathcal{T} = (T, t)$ can be *recovered* from $\delta_{\mathcal{T}}$ if for any other labelled tree $\mathcal{T}' = (T', t')$ for which $\delta_{\mathcal{T}}(x, y) = \delta_{\mathcal{T}'}(x, y)$ holds for all $x, y \in X$ distinct we have that T and T' are isomorphic and $t = t'$.

Corollary 3.3.4. *Let $\mathcal{T} = (T, t)$ be a labelled tree that is discriminating for $\delta_{\mathcal{T}}$. If the set of triplets displayed by T is given then $\mathcal{T}$ can be recovered from $\delta_{\mathcal{T}}$.*

Proof. Put $\delta = \delta_{\mathcal{T}}$, let $t : V_{int}(T) \to M$ and let $\mathcal{R}$ denote the set of triplets displayed by T. In view of Theorem 1.3.1, it suffices to show that the 2-way symbolic map $D_\delta : \binom{X}{2} \to M$ given as follows equals the symbolic ultrametric $D_{\mathcal{T}}$ on X induced by $\mathcal{T}$. Suppose $x, y \in X$ distinct. If there exists some $z \in X - \{x, y\}$ such that no triplet on $\{x, y, z\}$ is contained in $\mathcal{R}$ then we define $D_\delta(x, y)$ to be the element in $\delta_{\mathcal{T}}(x, y, z)$. If there exists some $z \in X - \{x, y\}$ such that $xy|z \in \mathcal{R}$ then we put $D_\delta(x, y) = n(\delta_{\mathcal{T}}(x, y, z))$ and if $xz|y \in \mathcal{R}$ then we put $D_\delta(x, y) = m(\delta_{\mathcal{T}}(x, y, z))$.

In view of Proposition 3.3.3, the map D_δ is clearly well-defined. Since $D_\delta(x, y) = t(lca(x, y)) = D_{\mathcal{T}}(x, y)$ clearly holds for all $x, y \in X$ distinct the corollary follows. □

In light of Corollary 3.3.4, it is of interest to understand when, for a labelled tree $\mathcal{T} = (T, t)$, the set of triplets displayed by T can be recovered from $\delta_{\mathcal{T}}$. The labelled tree $\mathcal{T}_3$ in Figure 3.3, suggests that this is not always possible. In fact, as we shall show, it suffices to exclude a special type of labelled tree which we define next.

A *fixed-cherry tree on* X with $|X| \geq 4$ is a labelled tree $\mathcal{T} = (T, t)$ such that the root ρ_T of T has two children v and w, with $t(v) = t(w) \neq t(\rho_T)$ and such that v is parent of two elements x_1 and x_2 of X and w the parent of all elements in $X - \{x_1, x_2\}$. In that case, we refer to $\{x, y\}$ as *fixed cherry* of $\mathcal{T}$. For example, the labelled tree $\mathcal{T}_3$ in Figure 3.3 is a fixed-cherry tree on $X = \{1, 2, 3, 4\}$. Note that for $x, y, z \in X$ distinct the 3-way symbolic ultrametric $\delta_{\mathcal{T}}$ induced by a fixed-cherry tree $\mathcal{T} = (T, t)$ satisfies $\delta_{\mathcal{T}}(x, y, z) = \{t(w), t(w), t(w)\}$ if neither x_1 nor x_2 belong to the set $\{x, y, z\}$, and $\delta(x, y, z) = \{t(\rho_T), t(\rho_T), t(w)\}$ else. We call such a 3-way symbolic map a *fixed-cherry map* for $\mathcal{T}$. More generally, we call a 3-way symbolic map $\delta : \binom{X}{3} \to \mathcal{M}$ a *fixed cherry map* if there exists some fixed cherry-tree $\mathcal{T}$ such that $\delta = \delta_{\mathcal{T}}$. The following observation is straightforward to check.

Lemma 3.3.5. *Suppose that $|X| \geq 5$ and that δ is a 3-way symbolic map on X. Then δ can be represented by a fixed-cherry tree if and only if δ is a fixed-cherry map.*

Note that a triplet $xy|z$ with $x, y, z \in X$ is displayed by a fixed-cherry tree with fixed cherry $\{x_1, x_2\}$ if and only if either $\{x, y\} = \{x_1, x_2\}$ or $z \in \{x_1, x_2\}$, and $x, y \in X - \{x_1, x_2\}$ hold. In particular, if $|X| > 4$ and δ is a fixed-cherry map for a fixed-cherry tree $\mathcal{T} = (T, t)$, then the elements in the fixed cherry can be easily identified from δ, and therefore also all of the triplets displayed by T.

We now consider how to recover the triplets displayed by the phylogenetic tree underpinning a labelled tree $\mathcal{T}$ in case $\mathcal{T}$ is not a fixed-cherry tree. Given a labelled tree $\mathcal{T} = (T, t)$ and a subset Y of X of size five or more, we denote by $\mathcal{T}_Y = (T_Y, t_Y)$ the unique (see Lemma 3.2.7) discriminating representation $\mathcal{T}_Y$ of $\delta|_Y$. We first start with a useful lemma.

Lemma 3.3.6. *Let $\mathcal{T} = (T, t)$ be a labelled tree that is a discriminating representation for $\delta_{\mathcal{T}}$ and let $Y \subseteq X$ such that $|Y| \geq 5$. If τ is a triplet displayed by T_Y then τ is a triplet displayed by T.*

Proof. It suffices to note that $\mathcal{T}_Y$ is obtained from $\mathcal{T}$ by first taking the subtree T' of T induced by Y, and then collapsing edges of T' both of whose ends have the same label under the restriction t' of t to $V(T')$. Clearly, $\mathcal{T}_Y$ is a discriminating representation of $\delta|_Y$. Note that, by Lemma 3.2.7, such a representation is unique.

It is well-known (see e. g. [55]) that the set $\mathcal{R}$ of triplets displayed by T' is contained in the set of triplets displayed by T. Since the process of collapsing edges of T' removes triplets from $\mathcal{R}$, but does not add any, it follows that a triplet displayed by T_Y is also displayed by T. □

We next present the main result of this section.

Theorem 3.3.7. *Suppose that $|X| \geq 4$ and that $\mathcal{T} = (T, t)$ is a labelled tree that is a discriminating representation for $\delta_{\mathcal{T}}$ but not a fixed-cherry tree. Let $x, y, z \in X$ be pairwise distinct. Then T displays the triplet $xy|z$ if and only if one of the following two properties holds:*

(T1) There exists some $u \in X$ such that $\delta_{\mathcal{T}}(x, u, z) = \delta_{\mathcal{T}}(y, u, z) \neq \delta_{\mathcal{T}}(x, y, u)$ and if $|\underline{\delta_{\mathcal{T}}(x, y, u)}| = 1$ then $\delta_{\mathcal{T}}(x, y, u) \neq \delta_{\mathcal{T}}(x, y, z)$.

(T2) There exists $u \in X$ such that $|\{\delta_{\mathcal{T}}(x, u, z), \delta_{\mathcal{T}}(y, u, z), \delta_{\mathcal{T}}(x, y, u)\}| = 3$ and $m(\delta_{\mathcal{T}}(x, u, z)) = m(\delta_{\mathcal{T}}(y, u, z)) \neq m(\delta_{\mathcal{T}}(x, y, u))$.

Proof. Put $\delta = \delta_{\mathcal{T}}$. Assume first that $x, y, z \in X$ are such that T displays the triplet $xy|z$. Put $v = \mathrm{lca}(x, z)$ and $w = \mathrm{lca}(x, y)$. We proceed using a case-analysis on the structure of T. Since $\mathcal{T}$ is not a fixed-cherry tree we need to consider the following (not necessarily disjoint) cases: (a): w is not a child of v, (b): v is not the root of T or has out-degree three or more, (c): w has a child that is neither x nor y, and (d): there exists a vertex v_0 on the path from v to z with $t(v_0) \neq t(w)$.

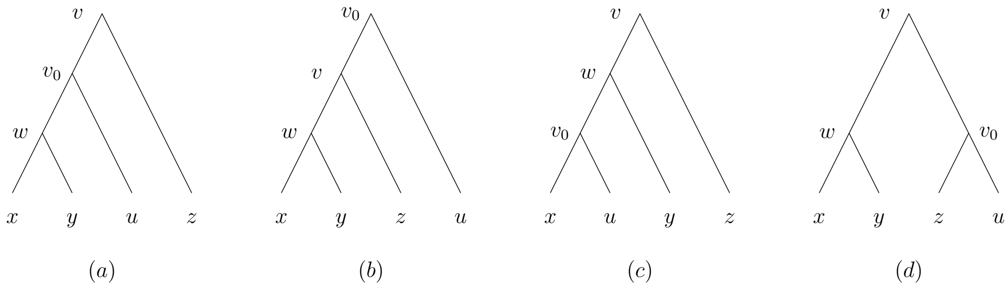


Figure 3.4: Cases (a) to (d) for the case-analysis carried out in the proof of Theorem 3.3.7. See text for details.

Case (a): Consider the parent v_0 of w, and an element u in X that is below v_0 but not below w (see Figure 3.4(a)). Since $\mathcal{T}$ is a discriminating representation for $\delta_{\mathcal{T}}$, we have $t(v_0) \neq t(w)$. Hence, $\delta(x, u, z) = \delta(y, u, z) = \{t(v_0), t(v), t(v)\}$ and $\delta(x, y, u) = \{t(w), t(v_0), t(v_0)\}$. Consequently, $\delta(x, u, z) = \delta(y, u, z) \neq \delta(x, y, u)$. Note that if $t(v) = t(w)$, then $\delta(x, y, z) = \{t(w), t(v), t(v)\}$ and, so, $|\underline{\delta(x, y, z)}| = 1$. But then $\delta(x, y, u) \neq \delta(x, y, z)$ as $|\underline{\delta(x, y, u)}| = 2$. Hence, the second condition in Property (T1) holds, too. So assume that $t(w) \neq t(v)$. But then $|\underline{\delta(x, y, u)}| = 2$ and so the second condition in Property (T1) does not apply.

Case (b): Consider an element of $u \in X$ such that $v_0 := \mathrm{lca}(u, z) = lca(u, x)$ (see Figure 3.4(b)). If w is not a child of v then Property (T1) follows by Case (a). So assume that w is a child of v. Then $t(v) \neq t(w)$ as $\mathcal{T}$ is a discriminating representation for $\delta_{\mathcal{T}}$. Since $\delta(x, u, z) = \delta(y, u, z) = \{t(v), t(v_0), t(v_0)\}$ and $\delta(x, y, u) = \{t(w), t(v_0), t(v_0)\}$ we have $\delta(x, u, z) = \delta(y, u, z) \neq \delta(x, y, u)$. Since the choice of v_0 implies that $|\underline{\delta(x,y,u)}| \neq 1$ the second condition in Property (T1) does not apply. Hence, Property (T1) is also satisfied in this case.

Case (c): Then there is some $u \in X$ below w that is neither x nor y. We may assume without loss of generality that $w = \text{lca}(y,u)$. Put $v_0 = \text{lca}(x,u)$ (see Figure 3.4(c)). Note that $v_0 = w$ may hold. Clearly, $\delta(x,u,z) = \{t(v_0), t(v), t(v)\}$, $\delta(y,u,z) = \{t(w), t(v), t(v)\}$ and $\delta(x,y,u) = \{t(v_0), t(w), t(w)\}$. If $v_0 \neq w$ then $\delta(y,u,z) \neq \delta(x,u,z) \neq \delta(x,u,y)$. Hence, $|\{\delta(y,u,z), \delta(x,u,z), \delta(x,u,y)\}| = 3$. Since $m(\delta(x,u,z)) = t(v) = m(\delta(y,u,z))$ and $m(\delta(x,y,u)) = t(w)$, Property (T2) follows. So assume that $v_0 \neq w$. Then $\delta(y,u,z) = \delta(x,u,z) = \{t(w), t(v), t(v)\}$ and $\delta(y,u,x) = \{t(w), t(w), t(w)\}$. In view of Property (T1) holding in case of Case (a), we may assume without loss of generality that w is a child of v. Since $\mathcal{T}$ is a discriminating representation of $\delta_{\mathcal{T}}$ we have $t(v) \neq t(w)$. Hence, $\delta(y,u,z) = \delta(x,u,z) \neq \delta(x,y,u)$. Since $|\underline{\delta(x,y,z)}| = 1$ and $|\underline{\delta(x,y,z)}| \neq 1$, Property (T1) follows in this case, too.

Case (d): Let $u \in X$ such that $v_0 = \text{lca}(z,u)$ (see Figure 3.4(d)). Then $\delta(x,u,z) = \delta(y,u,z) = \{t(v_0), t(v), t(v)\}$ and $\delta(x,y,u) = \{t(w), t(v), t(v)\}$. If $t(w) = t(v_0)$ we have $\delta(x,u,z) = \delta(y,u,z) = \delta(x,y,u)$. In view of Property (T1) holding if Case (a) applies, we may assume without loss of generality that w is a child of v. Hence, $t(w) \neq t(v)$ because $\mathcal{T}$ is a discriminating representation for δ. But then $|\underline{\delta(x,y,u)}| \neq 1$ and, so, the second condition in Property (T1) does not apply.

Conversely, let $x, y, z \in X$ distinct. Assume first that there exists some $u \in X - \{x,y,z\}$ such that Property (T1) is satisfied for the namesakes of u, x, y, and z. Assume for contradiction that the triplet $xy|z$ is not displayed by T. Consider the restriction δ' of δ to $\{x,y,u,z\}$ and let $\mathcal{T}' = (T', t')$ denote the unique discriminating representation of δ'. Then in view of the first condition in Property (T1), the out-degree of the root $\rho_{T'}$ cannot be four. Hence, one of the triplets $x|yz$ and $y|xz$ must be displayed by T' and T' is either resolved or unresolved. Assume first that T' is resolved. Then a straight forward case analysis concerned with adding u to the triplet $x|yz$ implies that that triplet cannot be displayed by T'. Swapping the roles of x and y in that argument also implies that the triplet $y|xz$ cannot be displayed by T' either. Thus, T' must be unresolved and, so, either $\rho_{T'}$ has out-degree three or one of the children of $\rho_{T'}$ has out-degree three.

If T' displays the triplet $x|yz$ and the out-degree of $\rho_{T'}$ is three then $|\underline{\delta(y,u,z)}| = 1$. Hence, $\delta(x,y,u) = \delta(x,y,z)$ in view of Property (T1) which is impossible. Thus, one of the children of $\rho_{T'}$ has out-degree three. But this is impossible in view of the first condition in Property (T1). Similar arguments imply that the triplet displayed by T' cannot be $y|xz$ either. Thus, T' must display the triplet $xy|z$. Consequently, either $\rho_{T'}$ is the parent of u and z or x, y, and u have the same parent. But the former cannot hold in view of the first condition in Property (T1) and the latter cannot hold in view of the second condition in that property.

Assume next that there exists some $u \in X - \{x, y, z\}$ such that Property (T2) is satisfied for the namesakes of u, x, y, and z. Assume for contradiction that the triplet $xy|z$ is not displayed by T. Consider again the restriction δ' of δ to $\{x, y, u, let \mathcal{T}' = (T', t')$ the unique discriminating representation δ'. In view of Table 3.1, there must be at least two subsets Y and Y' of $\{x, y, z, u\}$ of size three satisfying $\delta(Y) = \delta(Y')$. Since $|\{\delta(x, u, z), \delta(y, u, z), \delta(x, y, u)\}| = 3$, it follows that $\{x, y, z\}$ must be one of these subsets.

If $\delta(x, y, z) = \delta(x, y, u)$, then $D_T(x, u) = m(\delta(x, u, z))$ and $D_T(y, u) = m(\delta(y, u, z))$ must hold. Indeed, since $\delta(x, y, u) = \delta(x, y, z)$, one of the following two cases must hold: (α) $D_T(x, z) = D_T(x, u)$ and $D_T(y, z) = D_T(y, u)$. (β) $D_T(x, z) = D_T(y, u)$ and $D_T(y, z) = D_T(x, u)$. However Case (β) implies $\delta(x, z, u) = \delta(y, z, u)$, which is impossible in view of the assumption that $\{\delta(x, y, u), \delta(y, z, u), \delta(x, z, u)\}$ has size three. Thus, Case (α) must hold. But then $D_T(x, u) = m(\delta(x, u, z))$ and $D_T(y, u) = m(\delta(y, u, z))$, as required.

Since, by assumption, we also have $m(\delta(x, u, z)) = m(\delta(y, u, z))$ we obtain $D_T(x, u) = D_T(y, u)$. Since $D_T(x, u)$ and $D_T(y, u)$ are both elements in the multiset $\delta(x, y, u)$, we therefore have $m(\delta(x, u, z)) = D_T(x, u) = m(\delta(x, y, u))$, which is impossible in view of (T2). Thus, we either have $\delta(x, y, z) = \delta(x, u, z)$ or $\delta(x, y, z) = \delta(y, u, z)$. Note that the roles of x and y are interchangeable here, so we may assume without loss of generality that $\delta(x, y, z) = \delta(y, u, z)$.

Using similar arguments as before, we have $D_T(x, z) = m(\delta(x, u, z))$, $D_T(x, y) = m(\delta(x, u, y))$, and $D_T(y, z) = m(\delta(y, u, z))$in this case. By Property (T2), we therefore have $D_T(x, z) = D_T(y, z) \neq D_T(x, y)$. Thus $xy|z$ is a triplet in T', and by Lemma 3.3.6, it is also a triplet in T. □

As a direct consequence of Lemma 3.3.5 and Theorem 3.3.7 it is possible to decide whether or not a 3-way symbolic map δ on a set X with $|X| \geq 5$ is a 3-way symbolic ultrametric and, if so, construct the labelled tree $\mathcal{T}$ for which $\delta_{\mathcal{T}} = \delta$ holds as follows.

First, check if δ is a fixed-cherry map. If this is the case, then δ is a 3-way symbolic ultrametric and $\mathcal{T}$ can be easily constructed. If not, then compute the set $\mathrm{Tr}(\delta)$ of triplets $xy|z$ of X satisfying (T1) or (T2), and use it as input to the BUILD algorithm. If there is no tree displaying $\mathrm{Tr}(\delta)$, then δ is not a 3-way symbolic ultrametric. Otherwise, using the tree T that is constructed from the BUILD algorithm and the map δ, it is straightforward to decide if there is a labelling map t for T such that (T, t) represents δ. If this is the case, then δ is a 3-way symbolic ultrametric with the computed labelled tree (T, t) as a representation, otherwise it is not.

Note that BUILD may return a tree T from $\mathrm{Tr}(\delta)$ even if the map δ is not a

3-way symbolic ultrametric. For example, let $M = \{A, B\}$ and consider the map $\delta : \binom{X}{3} \to \mathcal{M}$ where $X = \{1, 2, 3, 4, 5\}$, and $\delta(x, y, z) = 3A$ if $\{x, y, z\} = \{3, 4, 5\}$, and $\delta(x, y, z) = 2A + B$ otherwise. Although this map is clearly not representable we have that $\mathrm{Tr}(\delta) = \{34|1, 34|2, 35|1, 35|2, 45|1, 45|2\}$, and it is easy to check that there exists a phylogenetic tree on X whose set of displayed triplets is $Tr(\delta)$.

3.4 Conclusion

The work presented in this chapter successfully introduces links between symbolic 3-way maps and phylogenetic trees in both the rooted and the unrooted case, and provides interesting results extended the existing ones on symbolic ultrametrics and k-dissimilarities. It is then natural to ask: What can be said about symbolic k-way map, for a value of k of four or more? A questions that might be interesting in this context is the following: Since the Symbolic Farris transform introduced in Section 3.2.1 trivially relates 3-way tree maps to symbolic ultrametrics, does a similar relationship holds for k-way tree maps and $(k-1)$-way symbolic ultrametrics, when k id greater than three?

Also, the idea of a minimal number of elements to restrict to in order to check if a map can be represented by a phylogenetic tree is fundamental both in this chapter and in the previous work focusing on k-dissimilarities (see Section 1.3.2). One of the reason for this is that it allows to build the representation of a 3-dissimilarity or of a symbolic 3-way map on X, should such a representation exist, by constructing the representations of the considered map on smaller subsets of X. This is the Divide-and-Conquer approach, already discussed in Chapter 2. As mentioned in the introduction of this chapter, one of the surprising result it contains is that although six points are necessary to check whether a 3-dissimilarity can be represented by a rooted phylogenetic tree (Theorem 1.3.5), only five points are required in the case of symbolic 3-way maps. Does this difference remain for $k \geq 4$, or is the progression of this number with regard to k, linear for k-dissimilarities, faster in case of symbolic maps?

Chap. 4

On circular split systems, 1-nested networks and the Buneman graph

Adapted from:

> P. Gambette, K. T. Huber and G. E. Scholz. Uprooted phylogenetic networks. *Bulletin of Mathematical Biology* (2017) 79(9): 2022-2048.

My personal contribution to this work has been the establishment of the main results (apart from Theorem 4.2.10), as well as writing the first draft of the paper, which includes the proofs of all the results presented.

This chapter addresses the question of the representability of a circular split system by an unrooted phylogenetic network. We successively consider two distinct approaches, based on minimal cuts and split graphs respectively, the latter in terms of a particular split-graph known as the "Buneman graph" of a split system, and then propose a bridge between these two approaches.

4.1 Introduction

In this chapter, which is based on [29], we focus on the more usual notion of a distance. More precisely, we consider here distances arising from split systems. The only prerequisites for reading this chapter are the notions presented in Section 1.2.2 and 1.2.3, which focus on splits and split systems, and on the way such objects can be represented in terms of networks.

As we have seen in Section 1.2.3, two distinct notions of displaying a split system Σ by a network coexist. The first one considers the notion of a minimal

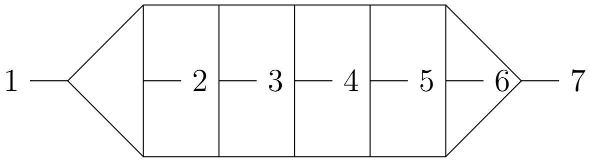


Figure 4.1: A phylogenetic network on $X = \{1, \dots, 7\}$ displaying all splits on X.

cut, and the second one is based on the concept of a split-network. We have also mentioned that these approaches coincide if any two splits in Σ are compatible, or equivalently, if the network representing Σ is a tree. The latter equivalence is guaranteed by Theorem 1.2.3, which also ensures, if the aforementioned condition is satisfied, the uniqueness of the tree representing Σ. In case of networks, however many questions remain unanswered.

Consider first the minimal-cut approach. If the split system Σ is not compatible, it is not true in general that there exists a phylogenetic network N such that $\Sigma = \Sigma(N)$. This is the case, for example, of the split system $\Sigma = \{12|345, 13|245, 14|235, 15|234\}$ on $X = \{1, 2, 3, 4, 5\}$. However, it is always possible to find a network N satisfying $\Sigma \subseteq \Sigma(N)$, as the phylogenetic network on $X = \{1, \dots, 7\}$ depicted in Figure 4.1 displays all splits in $\Sigma(X)$. Thus, the need to characterize splits systems that arise from a phylogenetic, and the need for optimality criteria if this is not the case.

In this chapter, we address these questions in the context of circular split systems, which extend the notion of a compatible split system but still enjoy some attractive properties. As we have seen in Section 1.2.3, the split system induced by a 1-nested network is always circular (Theorem 1.2.7). Moreover, such a split system can be represented by an outerplanar split-network (Theorem 1.2.6). As suggested by Figure 4.2, similarities can be observed between a 1-nested network N and a particular split-network, known as the Buneman graph, representing the split system $\Sigma(N)$. Indeed, the 1-nested network depicted in Figure 4.2(i) can be obtained from the split network depicted in Figure 4.2(ii) by removing all non-bold edges, collapsing two cut-edges, and suppressing resulting degree two vertices. We propose a generalization of this process, as a bridge between both approaches.

The outline of this chapter is as follows. In Section 4.2.1, we introduce relevant basic terminology relative to the minimal-cut approach for displaying a split system, and to the main differences between trees and networks in this regard. In Section 4.2.2, we state a closure rule for split systems which underpins our key tool: the $\mathcal{I}$-intersection closure of a split system. In Section 4.2.3, we characterize maximal circular split systems (Theorems 4.2.10) and present our 1-nested equivalence to Theorem 1.2.3 in the form of Theorem 4.2.12 and its Corollary 4.2.13. We

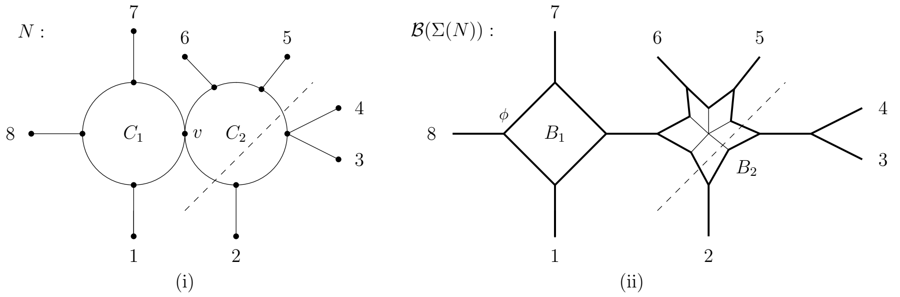


Figure 4.2: (i) A level-1 (unrooted) phylogenetic network N. (ii) A split network on X in the form of the Buneman graph $\mathcal{B}(\Sigma(N))$ on the split system $\Sigma(N)$ induced by N. In both Figures, the dashed line indicates the split $234|15678$.

then move on to the split-network approach, considering a particular such network associated to a split system, the Buneman graph. We use Section 4.3.1 to present a definition and review some relevant properties which will be of interest for the purpose of the following sections. In Section 4.3.2, we introduce and study the novel notion of a Marguerite, which stand as a tool for understanding the structure of the biconnected components of a Buneman graph. Finally, we highlight in Section 4.3.3 a link between the minimal cut and the split network approaches in the form of Theorem 4.3.8, which ensures that the Buneman graph of a split system associated to a 1-nested network N can be used to uniquely recover N (up to isomorphism and a mild condition) in polynomial time.

In this chapter, X denotes a finite set of size $n \geq 3$. Also, unless stated otherwise, split systems on X are assumed to contain all trivial splits on X. The reason for this is that the split system induced by a phylogenetic network must contain all such splits.

4.2 The minimal cut approach

We present here relevant basic definitions concerning split systems and 1-nested networks. We introduce and describe the key concept of a $\mathcal{I}$-intersection, that we then use to unlock some of the main results of this chapter.

4.2.1 Displaying splits

In case of a phylogenetic tree T, there is a trivial bijection between the split system $\Sigma(T)$ and the edge set of T. The situation is different for phylogenetics network in general, as a split S displayed by a phylogenetic network N may be obtained by removal of more than one edge of N (as is the case for the split $234|56781$ in Figure 4.2(i)). For a given split in $\Sigma(N)$, there also may be more than one set of edges of N whose removal induces the split S. Consider for example the leaf 1 in the 1-nested network N depicted in Figure 4.2(i). The deletion of the edge e_1 incident to that leaf induces the split $1|X-\{1\}$. However, the deletion of both edges adjacent to e_1, which is clearly a minimal cut of N, also displays the split $1|X-\{1\}$.

To take into account these differences, we now introduce further terminology, which will be used throughout this chapter. For the following, let N be a 1-nested network.

The *multiplicity* of a split $S \in \Sigma(N)$ is the number of distinct minimal cuts of N that induce S. We call a split S of N of multiplicity two or more an *m-split* of N. Also, we say that a split S is *displayed by a cycle* C of N if there exists a minimal cut of N inducing S that is contained in the edge set of C. If a m-split S of N is displayed by a cycle C, in the sense that it can be obtained by removal of a pair of edges of C, we also sometimes refer to S as a split of C. As an example, the split $1|X-\{1\}$ displayed by the network N depicted in Figure 4.2(i) is, as discussed above, a m-split of N that is also a split of C_1.

Note that in case N is a 1-nested network, a minimal cut E_S of N contains either one or two edges. Also, if $e=\{u,v\}$ is an edge of N such that neither u nor v is contained in a cycle of N then e must be a cut-edge of N and the multiplicity of the split S_e induced by deleting e is one. Moreover, if $e=\{u,v\}$ is a cut-edge of N where u or v is contained in a cycle C of N, say u, then S_e is also induced by deleting the edges of C incident with u. Thus, the multiplicity of a given split in $\Sigma(N)$ can either be one, two, or three.

Furthermore, the split system $\Sigma(N)$ induced by a 1-nested network N on X is the same as the one induced by the resolution of N to a level-1 network by repeatedly applying the following two replacement operations:

(a) A vertex v of a cycle C of N incident with $l \geq 2$ edges $e_1, \dots, e_l$ not contained in C is replaced by an edge one of whose vertices is v and the other is incident with $e_1, \dots, e_l$, and

(b) A cut-vertex v shared by two cycles C_1 and C_2 is replaced by a cut-edge one of whose vertices is contained in C_1 and the other in C_2.

However, the multi-sets of splits induced by both networks are clearly different, as the addition of a new cut-edge by applying (a) or (b) to a 1-nested network induces a new way to display a split that is already present in $\Sigma(N)$. We call a 1-nested network N' a *partial-resolution* of a 1-nested network N if N' can be obtained from N by partially resolving vertices of N. Moreover, we call a partial-resolution N' of N a *maximal partial-resolution* of N if all vertices contained in

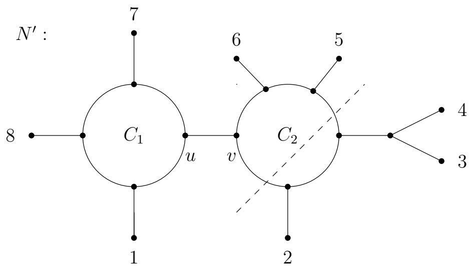


Figure 4.3: A maximal partial resolution N' of the network N depicted in Figure 4.2(i). As in Figure 4.2(i), the dashed line indicates the split 234|15678.

a cycle of N' have degree three. In this case, we also call N' *maximal partially-resolved*. As an example, the network N' depicted in Figure 4.3 is a maximal partial resolution of the network N depicted in Figure 4.2(i). We obtain N' from N by applying operation (a) to the vertex adjacent to the leaves 3 and 4, and operation (b) to the vertex v.

Finally, as can be seen in Figure 4.4, the split system induced by a network N containing a cycle C of length three is the same as the split system of the network N' obtained after deletion of the edges of C and identification of all three vertices in C. Thus, we invoke parsimony to add the requirement that all cycles in a 1-nested network must have length four or more.

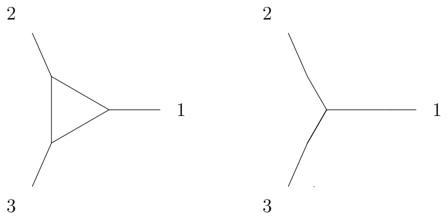


Figure 4.4: Two distinct 1-nested networks on $X = \{1, 2, 3\}$ inducing the same split system.

4.2.2 A characterization in terms of $\mathcal{I}$-intersections

We now introduce and study the $\mathcal{I}$-intersection closure of a split system. As we have seen in Section 1.2.2, the number and the properties of the intersection between two splits S_1 and S_2 depends on whether these two splits are compatible or not. As we shall see, more difference will soon appear between these two distinct situations. To make the distinction clear, if S_1 and S_2 are incompatible, we refer to the intersections of S_1 and S_2 as *incompatible intersection*, or $\mathfrak{I}$-*intersection* for short, and denote it by $\iota(S_1, S_2)$ rather than $int(S_1, S_2)$. See Figure 4.2.2 an illustration.

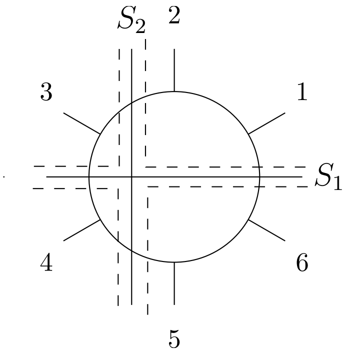


Figure 4.5: For a simple level-1 network on $\{1,\ldots,6\}$, we depict the splits S_1 and S_2 in terms of two straight bold lines and the four splits that make up $\iota(S_1, S_2)$ in terms of four dashed lines.

Figure 4.2.2 suggests that every split in $\iota(S_1, S_2)$ is displayed by the same cycle that displays S_1 and S_2. Establishing that this is indeed the case is the purpose of Proposition 4.2.2. To state it in its full generality we next associate to a split system Σ of X the *intersection closure* $Int(\Sigma)$ of Σ, that is, $Int(\Sigma)$ is a (set-inclusion) minimal split system that contains Σ and is closed by intersection. For example, for $\Sigma = \{12|345, 23|451\}$ on $X = \{1,2,3,4,5\}$, we have $Int(\Sigma) = \Sigma \cup \{1|2345, 2|3451, 3|4512, 13|452, 123|45\}$.

We start our analysis of $Int(\Sigma)$ with remarking that $Int(\Sigma)$ is indeed a closure, that is, $Int(\Sigma)$ trivially satisfies the following three properties

(i) $\Sigma \subseteq Int(\Sigma)$.

(ii) $Int(Int(\Sigma)) = Int(\Sigma)$.

(iii) If Σ' is a split system on X for which $\Sigma \subseteq \Sigma'$ holds then $Int(\Sigma) \subseteq Int(\Sigma')$.

The next lemma implies that the intersection closure of a split system is well-defined.

Lemma 4.2.1. *Suppose Σ is a split system on X and Σ' is a (set-inclusion) minimal superset of Σ that is closed by intersection. Then $\Sigma' = Int(\Sigma)$ must hold.*

Proof. Since Σ' contains Σ and is intersection closed we can obtain Σ' via a (finite) sequence $\Sigma = \Sigma_0 \subsetneq \Sigma_1 \subsetneq \Sigma_2 \subsetneq \ldots \subsetneq \Sigma_k = \Sigma'$, $k \geq 1$, of split systems Σ_i such that, for all $1 \leq i \leq k$, $\Sigma_i := \Sigma_{i-1} \cup \iota(P_i)$ where P_i is a 2-set contained in Σ_{i-1} and $\iota(P_i)$ is not contained in Σ_{i-1}. We show by induction on i that $\Sigma_i \subseteq Int(\Sigma)$ holds.

Clearly, if $i = 0$ then $\Sigma_0 = \Sigma$ is contained in $Int(\Sigma)$. So assume that $\Sigma_i \subseteq Int(\Sigma)$ holds for all $1 \leq i \leq r$, for some $1 \leq r \leq k$, and that Σ_r is obtained from Σ_{r-1} by intersection of two splits $S_1, S_2 \in \Sigma_{r-1}$. Since, by induction hypothesis, $\Sigma_{r-1} \subseteq Int(\Sigma)$ it follows that S_1 and S_2 are contained in $Int(\Sigma)$. Since $Int(\Sigma)$ is intersection-closed, $\iota(S_1, S_2) \subseteq Int(\Sigma)$ follows. Hence, $\Sigma_r = \Sigma_{r-1} \cup \iota(S_1, S_2) \subseteq Int(\Sigma)$, as required. By induction, it now follows that $\Sigma' \subseteq Int(\Sigma)$. Reversing the roles of Σ' and $Int(\Sigma)$ in the previous argument implies that $Int(\Sigma) \subseteq \Sigma'$ holds too which implies $\Sigma' = Int(\Sigma)$. □

We remark in passing that similar arguments as the ones used in the proof of Lemma 4.2.1 also imply that the $\mathcal{I}$-intersection closed (set-inclusion) minimal superset $\mathcal{I}(\Sigma)$ of a split system Σ is also well-defined (and obviously satisfies Properties (i) – (iii)). We will refer to $\mathcal{I}(\Sigma)$ as *$\mathcal{I}$-intersection closure of* Σ.

We next turn our attention to the $\mathcal{I}$-intersection closure of a split systems induced by a 1-nested network.

Proposition 4.2.2. *Suppose N is a 1-nested network on X and S_1 and S_2 are two incompatible splits contained in $\Sigma(N)$. Then $\iota(S_1, S_2) \subseteq \Sigma(N)$.*

Proof. Note first that two splits S and S' induced by a 1-nested network are incompatible if and only if they are displayed by pairs of edges in the same cycle C of N. For $i = 1, 2$, let $\{e_i, e'_i\}$ denote the edge set whose deletion induces the split S_i. Then since S_1 and S_2 are incompatible, we have $\{e_1, e'_1\} \cap \{e_2, e'_2\} = \emptyset$ and none of the connected components of N obtained by deleting e_i and e'_i contains both e_j and e'_j, for all $i, j \in \{1, 2\}$ distinct. Without loss of generality, we may assume that when starting at edge e_1 and moving clockwise through C we first encounter e_2, then e'_1 and, finally e'_2 before returning to e_1. Then it is straightforward to see that a split in $\iota(S_1, S_2)$ is displayed by one of the edge sets $\{e_1, e_2\}$, $\{e_2, e'_1\}$, $\{e'_1, e'_2\}$, and $\{e'_2, e_1\}$. Thus, $\iota(S_1, S_2) \subseteq \Sigma(N)$. □

Combined with the definition of the $\mathcal{I}$-intersection closure, we obtain the following result.

Corollary 4.2.3. *The following statements hold:*

(i) If Σ is a circular split system for some circular ordering of X then $\mathcal{I}(\Sigma)$ is also circular for that ordering.

(ii) If N is a 1-nested network on X then $\Sigma(N)$ is $\mathcal{I}$-intersection closed. Furthermore, N displays a split system Σ on X if and only if N displays $\mathcal{I}(\Sigma)$.

The next observation is almost trivial and is used in the proof of Theorem 4.2.5.

Lemma 4.2.4. *Suppose $x \in X$ and S_1, S_2, and S_3 are three distinct splits of X such that $S_3(x) \subseteq S_1(x)$, S_3 and S_2 are compatible and S_1 and S_2 are incompatible. Then $S_3(x) \subseteq S_2(x)$.*

Proof. Since S_2 and S_3 are compatible either $S_2(x) \subseteq S_3(x)$ or $S_3(x) \subseteq S_2(x)$ or $\overline{S_3(x)} \subseteq S_2(x)$ must hold. If $S_2(x) \subseteq \overline{S_3(x)}$, then $S_2(x) \subseteq S_1(x)$ which is impossible since S_1 and S_2 are incompatible. If $\overline{S_3(x)} \subseteq S_2(x)$ held then $\emptyset \neq \overline{S_1(x)} \cap \overline{S_2(x)} \subseteq \overline{S_3(x)} \cap \overline{S_2(x)} = S_2(x) \cap \overline{S_2(x)} = \emptyset$ follows which is impossible. □

For clarity of presentation, we remark that for the proof of Theorem 4.2.5, we can assume that if a given split S of a 1-nested network N has multiplicity at least two in the multi-set of splits induced by N then S is displayed by a cycle C of N (rather than by a cut-edge of N). Furthermore, we denote the split system of X induced by a cycle C of a 1-nested network N on X by $\Sigma(C)$. Clearly, $\Sigma(C) \subseteq \Sigma(N)$ holds.

Theorem 4.2.5. *Suppose Σ is a split system on X that contains all trivial splits of X. Then the following hold:*

(i) There exists a 1-nested network N on X such that $\Sigma = \Sigma(N)$ if and only if Σ is circular and $\mathfrak{I}$-intersection closed.

(ii) A maximal partially-resolved 1-nested network N is binary if and only if there exists no split of X not contained in $\Sigma(N)$ that is compatible with every split in $\Sigma(N)$.

Proof. (i): Assume first that there exists a 1-nested network N on X such that $\Sigma = \Sigma(N)$. Then arguments similar to the ones used in [27] to establish that the split system induced by a level-1 network is circular imply that $\Sigma(N)$ is circular. Hence, Σ must be circular. That Σ is $\mathfrak{I}$-intersection closed follows by Corollary 4.2.3(ii).

Conversely, assume that Σ is circular and $\mathfrak{I}$-intersection closed. Then there clearly exists a 1-nested network N such that $\Sigma \subseteq \Sigma(N)$. Let N be such that $|\Sigma(N)|$ is minimal among all 1-nested networks on X satisfying that set inclusion[1]. Without loss of generality, we may assume that N is maximal partially-resolved. We show that, in fact, $\Sigma = \Sigma(N)$ holds. Assume for contradiction that there exists a split $S_0 \in \Sigma(N) - \Sigma$. Since $\Sigma(N)$ must contain all trivial splits of X it follows that S_0 cannot be a trivial split of X. In view of the remark preceding Theorem 4.2.5, S_0 is induced by either (a) deleting a cut-edge $e = \{u, v\}$ of N and neither u nor v are contained in a cycle of N or (b) deleting two distinct edges of the same cycle of N.

Assume first that Case (a) holds. Then collapsing e results in a 1-nested network N' on X for which $\Sigma \subseteq \Sigma(N')$ holds. But then $|\Sigma(N')| < |\Sigma(N)|$ which

[1] We refer to Section 4.3.1 for a construction of such a network

is impossible in view of the choice of N. Thus, Case (b) must hold, that is, S_0 is induced by deleting two distinct edges $e = \{u, v\}$ and $e' = \{u', v'\}$ of the same cycle C of N. Let x and y be two elements of X for which there exists a path from u and v, respectively, which does not cross an edge of C. Consider the sets $\Sigma_x := \{S \in \Sigma \cap \Sigma(C) : S(x) \subsetneq S_0(x)\}$, and $\Sigma_y := \{S \in \Sigma \cap \Sigma(C) : S(y) \subsetneq S_0(y)\}$. If Σ_x is non-empty then choose some $S_x \in \Sigma_x$ such that $|S_x(x)|$ is maximal among the splits contained in Σ_x. Similarly, define the split S_y for Σ_y if Σ_y is non-empty. Otherwise let S_x be the m-split of C such that $S_x(x) \subsetneq S_0(x)$. Similarly, let S_y be the m-split of C such that $S_y(y) \subsetneq S_0(y)$ in case Σ_y is empty. Then Corollary 4.2.3(ii) implies that the split

$$S^* = S_x(x) \cup S_y(y) | \overline{S_x(x)} \cap \overline{S_y(y)}$$

is contained in $\Sigma(N)$ (see Figure 4.6(i) for an illustration).

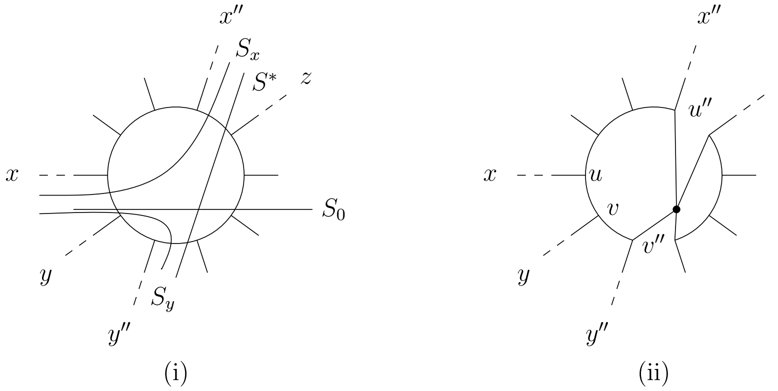


Figure 4.6: (i) An illustration of the reduction process considered in the proof of Case (a) of Theorem 4.2.5. (ii) Again for that theorem, the graph G' obtained from N by adding subdivision vertices r and r'.

We next show that S^* is compatible with every split in Σ. To this end we first claim that every split $S' \in \Sigma$ that is incompatible with S^* must be compatible with at least one of S_x and S_y. To see this, let $S' \in \Sigma$ such that S' and S^* are incompatible. Then S' must be displayed by C. For contradiction, assume that S' is incompatible with both of S_x and S_y. Let $z \in X$ such that $S^*(x) \neq S^*(z)$ and let $u'' \in V(C)$ such that $S_x(x)$ is the interval $[u, u'']$. Choose some element $x'' \in X$ such that there exists a path from x'' to u'' that does not cross an edge contained in C. Similarly, let $v'' \in V(C)$ such that $S_y(y)$ is the interval $[v'', v]$. Choose some element $y'' \in X$ such that there exists a path from y' to v'' that does not cross an edge contained in C. Then since S' is incompatible with S_x and S_y and displayed by C it follows that $S'(x'') = S'(y'') = S'(z)$. Hence, $S^*(z) \subsetneq S'(z)$. But then S^*

and S' are not incompatible which is impossible. Thus S' cannot be incompatible with both of S_x and S_y, as claimed.

To see that S^* is compatible with every split in Σ, we may, in view of the above claim, assume without loss of generality that S' is compatible with S_x. Then Lemma 4.2.4 applied to S', S^*, and S_x implies $S_x(x) \subsetneq S'(x)$. We distinguish between the cases that (α) S_y and S' are compatible and (β) that they are incompatible.

Case (α): Since S_y and S' are compatible, similar arguments as above imply that $S_y(y) \subsetneq S'(y)$. Then the definition of S^* combined with the assumption that S' and S^* are incompatible implies that $S'(x) \neq S'(y)$. But then S' and S_0 must be compatible, and so, $S'(x) \subseteq S_0(x)$ or $S_0(x) \subseteq S'(x)$ must hold. If $S'(x) \subseteq S_0(x)$ held then $S' \in \Sigma_x$ which is impossible in view of the choice of S_x as $S_x(x) \subsetneq S'(x)$. Thus, $S_0(x) \subseteq S'(x)$ must hold. But then $S_y(y) \subsetneq S'(y) \subseteq S_0(y)$ and so $S' \in \Sigma_y$ which is impossible in view of the choice of S_y. Thus, Case (β) must hold.

Case (β): Since S_y and S' are incompatible the split

$$S'' = S'(x) \cap \overline{S_y(y)} | \overline{S'(x)} \cup S_y(y)$$

is contained in Σ because Σ is $\mathfrak{I}$-intersection closed and clearly displayed by C. Note that $x \in \overline{S''(y)}$ and so $S''(x) = \overline{S''(y)}$ must hold. Moreover, since S' and S^* are incompatible, we cannot have $S''(x) = S_x(x)$ as S_x and S^* are compatible. But then S_0 and S'' cannot be compatible. Indeed, if S_0 and S'' were compatible then since $y \in S_0(y) \cap S''(y)$, $x \in \overline{S_0(y)} \cap \overline{S''(y)}$, and, because of $S_x(x) \subsetneq S'(x)$, also $\overline{S_0(y)} \cap S''(y) = S_0(x) \cap S''(y) = S_0(x) \cap (\overline{S'(x)} \cup S_y(y)) \subseteq S_0(x) \cap \overline{S_x(x)} \neq \emptyset$ holds, it follows that $\overline{S''(y)} \subseteq \overline{S_0(y)}$, as required. Hence, $S''(x) = \overline{S''(y)} \subseteq \overline{S_0(y)} = S_0(x)$ and so $S'' \in \Sigma_x$ which is impossible in view of the choice of S_x as $S_x(x) \neq S''(x)$ and $S_0(x) \neq S''(x)$. Thus, S_0 and S'' must be incompatible. But this is also impossible since the interval on C corresponding to $S''(x)$ contains the interval $[x, z]$ which induces the split S_0. Consequently, S_0 and S'' must be compatible. This final contradiction completes that proof that S^* is compatible for every split in Σ.

To conclude, let G be a new graph obtained from N by adding a subdivision vertex r and r', respectively, to each of two edges whose deletion induces the split S^* (see Figure 4.6(ii) for an illustration). Then, the graph G' obtained from G by identifying r and r' is again a 1-nested network on X. By construction, $S_0 \in \hat{\Sigma} := \{S \in \Sigma(N) :\ \text{S is incompatible with } S^*\}$ clearly holds and so $\Sigma(G') = \Sigma(N) - \hat{\Sigma} \subsetneq \Sigma(N)$. Since, by the above, every split in Σ is compatible with S^* it follows that $\Sigma \subseteq \Sigma(G')$. But this impossible in view of the choice of N. Hence, the split S_0 cannot exist and, thus, $\Sigma = \Sigma(N)$.

(ii) Suppose N is a maximal partially-resolved 1-nested network. Assume first that N is binary and, for contradiction, that there exists some split S of X not contained in $\Sigma(N)$ that is compatible with every split in $\Sigma(N)$. Then S' cannot be a trivial split of X. Let N' be the graph obtained from N by deleting from each cycle of N one of its edges and suppressing resulting degree two vertices. Clearly N' is a phylogenetic tree on X. Since every non-leaf vertex of N has degree three every such vertex in N' must also have degree three. Hence, by Theorem 1.2.3, $\Sigma(N')$ is a maximal compatible split system on X. Since S is compatible with every split of $\Sigma(N)$ and $\Sigma(N') \subseteq \Sigma(N)$ it follows that $\Sigma(N') \cup \{S\}$ is also compatible which is impossible in view of the maximality of $\Sigma(N')$.

Conversely, assume that there exists no split of X not contained in $\Sigma(N)$ that is compatible with every split in $\Sigma(N)$. Then if N is not binary it contains a vertex v of degree $k \geq 4$, that, since N is partially resolved, does not belong to a cycle of N. Let $X_1, \dots, X_k$ be the partition of X obtained by deletion of v (suppressing incident edges). Then there exist $i, j \in \{1, \dots, k\}$ distinct, say $i = 1$ and $j = 2$, such that the split $S := X_1 \cup X_2 | \bigcup_{i=3}^{k} X_i$ is compatible with every split in N. Since S does not belong to $\Sigma(N)$ this is impossible. □

Note that Theorem 4.2.5(i) provides a way to decide for a split system Σ if there exists a 1-nested network N such that $\Sigma = \Sigma(N)$ holds. However it does not provide a tool for constructing such a network. The provision of such a tool is the purpose of the next two sections.

4.2.3 The analogue of the Split Equivalence Theorem

As is easy to see, any circular split system on some set X can be represented in terms of a 1-nested network N_Σ on X by first considering a a cycle C of length $|X|$, then assigning the elements of X to the vertices of C according to their induced circular ordering and, finally, attaching to each vertex v of C a pendant edge e and shifting the element of X labelling v to the degree one vertex of e. As in the rooted case (see Chapter 2), we call such a network a *simple* level-1 network.

For the split system Σ on $X = \{1, \dots, 8\}$ comprising of all splits of the form $x|X - \{x\}$ where $x \in X := \{1, \dots, 8\}$ and the splits 81|234567, 78|123456, 781|23456, 234|56781, 34|567812, 345|67812, 2345|6781, 3456|7812 and 56|78123, the network N_Σ is generally not optimal. Put differently, N_Σ displays a total of $\binom{|X|}{2}$ distinct splits of X (including those in Σ) whereas the 1-nested network N depicted in Figure 4.2(i) also displays all splits of Σ but postulates fewer additional splits.

This section is devoted to clarifying the above observation. In particular, we show next that for any circular split system Σ on X it is possible to construct a, in a well-defined sense, optimal 1-nested network on X in $O(n(n + |\Sigma|^2))$ time (Theorem 4.2.12). Central to our proof is Theorem 4.2.10 in which we characterize circular split systems whose $\mathfrak{I}$-intersection closure is (set-inclusion) maximal in terms of their so called incompatibility graphs. As a consequence, we obtain as Corollary 4.2.13 the 1-nested analog of the fundamental "Splits-Equivalence Theorem" (Theorem 1.2.3) for phylogenetic trees.

We start with introducing some more terminology. Suppose Σ is a circular split system on X. Then we say that Σ is *maximal circular* if for all circular split systems Σ' on X that contain Σ, we have $\Sigma = \Sigma'$. As the next result illustrates, maximal circular split systems of X and simple 1-nested networks on X are closely related.

Lemma 4.2.6. *A split system Σ on X is maximal circular if and only if there exists a simple level-1 network N on X such that $\Sigma = \Sigma(N)$.*

Proof. Let Σ be a split system on X. Assume first that Σ is maximal circular. Then there exists a simple level-1 network N on X such that $\Sigma \subseteq \Sigma(N)$. Since $\Sigma(N)$ is clearly a circular split system on X the maximality of Σ implies $\Sigma = \Sigma(N)$.

Conversely, assume that N is a simple level-1 network such that $\Sigma = \Sigma(N)$. Then since $\Sigma(N)$ is a circular split system on X so is Σ. Assume for contradiction that Σ is not maximal circular, that is, there exists a split $S = A|\bar{A} \in \Sigma$ that is not contained in $\Sigma(N)$. Then A and $\bar{A}$ are both intervals in the circular ordering of X induced by $\Sigma(N)$. Hence, S is induced by a minimal cut of N. Consequently, $S \in \Sigma(N)$ which is impossible. □

Note that since a maximal circular split system on X must necessarily contain all splits of X of the form $A|B$ with $|A| = 2$ obtainable as a minimal cuts in the associated simple level-1 network on X, it follows that that ordering of X is unique. The next result suggests that systems of such splits suffice to generate a maximal circular split system. To state it, suppose $\sigma : x_1, ..., x_{n-1}, x_n, x_{n+1} := x_1$ is a circular ordering of X and put $\Sigma_\sigma := \{\{x_i, x_{i+1}\}|X - \{x_i, x_{i+1}\} : 1 \leq i \leq n\}$. Clearly, Σ_σ is a circular split system on X.

In view of Lemma 4.2.6, we say that a circular ordering *displays* a split system Σ if Σ is displayed by the simple level-1 network associated to Σ.

Lemma 4.2.7. *$\mathcal{I}(\Sigma_\sigma)$ is a maximal circular split system on X, for any circular ordering σ of X.*

Proof. Since the result is trivial for $n = 3$, we may assume without loss of generality that $n \geq 4$. Let $\sigma : x_1, ..., x_{n-1}, x_n, x_{n+1} := x_1$ be a circular ordering of X. We proceed by induction on the size $1 \leq l \leq \frac{n}{2}$ of a split S displayed by σ, that is, the size of the smaller set that compose S. Suppose first that $l = 1$. Then there exists some $i \in \{1, \dots, n\}$ such that $S = x_i|X - \{x_i\}$. Clearly, $S_1 = \{x_i, x_{i-1}\}|X - \{x_i, x_{i-1}\}$ and $S_2 = \{x_i, x_{i+1}\}|X - \{x_i, x_{i+1}\}$ are contained in Σ_σ and incompatible. Hence, $S = S_1(x_i) \cap S_2(x_i)|X - (S_1(x_i) \cap S_2(x_i)) \in \iota(S_1, S_2) \subseteq \mathcal{I}(\Sigma_\sigma)$.

Now assume that $l \geq 2$ and that all splits of X displayed by σ of size at most $l-1$ are contained in $\mathcal{I}(\Sigma_\sigma)$. Since S is displayed by σ, there exists some $i \in \{1, \ldots, n\}$ such that $S = [x_i, x_{i+l-1}]|X - [x_i, x_{i+l-1}]$. Without loss of generality we may assume that $i = 1$. Then $S = [x_1, x_l]|X - [x_1, x_l]$. Consider the splits $S_1 = [x_1, x_{l-1}]|X - [x_1, x_{l-1}]$ and $S_2 = \{x_{l-1}, x_l\}|X - \{x_{l-1}, x_l\}$ displayed by σ. By induction, $S_1, S_2 \in \mathcal{I}(\Sigma_\sigma)$ since the size of S_2 is two and that of S_1 is at most $l-1$. Furthermore, S_1 and S_2 are incompatible. Since $S = S_1(x_{l-1}) \cup S_2(x_{l-1})|X - (S_1(x_{l-1}) \cup S_2(x_{l-1})) \in \iota(S_1, S_2) \subseteq \mathcal{I}(\Sigma_\sigma)$, the lemma follows. □

We next employ Lemma 4.2.7 to obtain a sufficient condition on a circular split system Σ for its $\mathcal{I}$-intersection closure to be maximal circular. Central to this is the concept of the *incompatibility graph* $Incomp(\Sigma)$ associated to a split system Σ. The vertex set of that graph is Σ and any two splits of Σ are joined by an edge in $Incomp(\Sigma)$ if they are incompatible. We denote the set of connected components of $Incomp(\Sigma)$ by $\pi_0(\Sigma)$ and, by abuse of terminology, refer to the vertex set of an element in $\pi_0(\Sigma)$ as a *connected component of* $Incomp(\Sigma)$. For example, $Incomp(\Sigma_\sigma)$ is a cycle of length $|\Sigma_\sigma|$ whenever $n \geq 5$. Furthermore, Σ is compatible if and only if $|\Sigma_0| = 1$ holds for all $\Sigma_0 \in \pi_0(\Sigma)$.

We next clarify the relationship between the incompatibility graph and $\mathcal{I}$-intersection closure of a split system.

Lemma 4.2.8. *Suppose Σ is a split system on X. Then for any two distinct connected components $\Sigma_1, \Sigma_2 \in \pi_0(\Sigma)$ and any splits $S_1 \in \mathcal{I}(\Sigma_1)$ and $S_2 \in \mathcal{I}(\Sigma_2)$ we must have that S_1 and S_2 are compatible.*

Proof. Assume for contradiction that there exist two connected components $\Sigma_1, \Sigma_2 \in \pi_0(\Sigma)$ and splits $S_1 \in \mathcal{I}(\Sigma_1)$ and $S_2 \in \mathcal{I}(\Sigma_2)$ such that S_1 and S_2 are incompatible. Then $S_1 \in \Sigma_1$ and $S_2 \in \Sigma_2$ cannot both hold as otherwise $\Sigma_1 = \Sigma_2$. Assume without loss of generality that $S_1 \notin \Sigma_1$. Let $\Sigma^0 := \Sigma_1 \subsetneq \Sigma^1 \subsetneq \ldots \subsetneq \Sigma^k := \mathcal{I}(\Sigma_1)$, $k \geq 1$, be a finite sequence such that, for all $1 \leq i \leq k$, a split in Σ^i either belongs to Σ^{i-1} or is an $\mathcal{I}$-intersection between two splits $S, S' \in \Sigma^{i-1}$ and $\iota(S, S') \not\subseteq \Sigma^{i-1}$. Then, there exists some $i^* > 0$ such that $S_1 \in \Sigma^{i^*} - \Sigma^{i^*-1}$. After possibly renaming S_1, we may assume without loss of generality, that i^* is such that for all $1 \leq i \leq i^* - 1$ there exists no split in Σ^i that is incompatible with S_2. Hence, there must exist two splits S and S' in Σ^{i^*-1} distinct such that $S_1 \in \iota(S, S')$. Since S_2 and S_1 are incompatible, it follows that S_2 is incompatible with one of S and S', which is impossible by the choice of i^*. □

Armed with this result, we next relate for a split system Σ the sets $\pi_0(\mathcal{I}(\Sigma))$ and $\pi_0(\Sigma)$ in Lemma 4.2.9. In particular, we show that $\mathcal{I}(\Sigma)$ can be obtained as the intersection closure of the connected components of $Incomp(\Sigma)$. Also, the set of connected components of $\mathcal{I}(\Sigma)$ can be obtained as the connected components of the intersection closure of the connected components of $Icomp(\Sigma)$.

Lemma 4.2.9. *Suppose Σ is a split system on X. Then the following hold*

(i) $\mathcal{I}(\Sigma) = \bigcup_{\Sigma_0 \in \pi_0(\Sigma)} \mathcal{I}(\Sigma_0)$.

(ii) $\pi_0(\mathcal{I}(\Sigma_0)) \subseteq \pi_0(\mathcal{I}(\Sigma))$, *for all* $\Sigma_0 \in \pi_0(\Sigma)$. *In particular,* $\pi_0(\mathcal{I}(\Sigma)) = \bigcup_{\Sigma_0 \in \pi_0(\Sigma)} \pi_0(\mathcal{I}(\Sigma_0))$.

Proof. (i) Let $\Sigma_0 \in \pi_0(\Sigma)$ and put $\mathcal{A} := \bigcup_{\Sigma' \in \pi_0(\Sigma)} \mathcal{I}(\Sigma')$. Note that since $\Sigma = \bigcup_{\Sigma' \in \pi_0(\Sigma)} \Sigma'$, we trivially have $\Sigma \subseteq \mathcal{A} \subseteq \mathcal{I}(\Sigma)$. To see that $\mathcal{I}(\Sigma) \subseteq \mathcal{A}$ note that Lemma 4.2.8 implies that any two incompatible splits in $\mathcal{A}$ must be contained in the same connected component of $\mathcal{I}(\Sigma)$ and so must be their $\mathcal{I}$-intersection. Hence, $\mathcal{A}$ is $\mathcal{I}$-intersection closed. Since $\Sigma \subseteq \mathcal{A}$ we also have $\mathcal{I}(\Sigma) \subseteq \mathcal{I}(\mathcal{A}) = \mathcal{A}$. Thus $\mathcal{A} = \mathcal{I}(\Sigma)$.

(ii) Suppose $\Sigma_0 \in \pi_0(\Sigma)$ and let $\mathcal{A} \in \pi_0(\mathcal{I}(\Sigma_0))$. To establish that $\mathcal{A} \in \pi_0(\mathcal{I}(\Sigma))$ note that since $\mathcal{A}$ is connected in $Incomp(\mathcal{I}(\Sigma_0))$ it also is connected in $Incomp(\mathcal{I}(\Sigma))$. Hence, it suffices to show that every split in $\mathcal{A}$ is compatible with every split in $\mathcal{I}(\Sigma) - \mathcal{A}$. Suppose $S_1 \in \mathcal{A}$ and $S_2 \in \mathcal{I}(\Sigma) - \mathcal{A} = (\mathcal{I}(\Sigma) - \mathcal{I}(\Sigma_0)) \cup (\mathcal{I}(\Sigma_0) - \mathcal{A})$. If $S_2 \in \mathcal{I}(\Sigma_0) - \mathcal{A}$ then, by definition, S_1 and S_2 are compatible. So assume that $S_2 \in \mathcal{I}(\Sigma) - \mathcal{I}(\Sigma_0)$. Then Lemma 4.2.9(i) implies that S_2 is compatible with every split in $\mathcal{I}(\Sigma_0)$ and, thus, with S_1 as $\mathcal{A} \subseteq \mathcal{I}(\Sigma_0)$. □

To establish the next result which is central to Theorem 4.2.12, we require a further notation. Suppose Σ is a split system on X. Then we denote by Σ^- the split system obtained from Σ by deleting all trivial splits on X.

Theorem 4.2.10. *Let Σ be a circular split system on X. Then $\mathcal{I}(\Sigma)$ is a maximal circular split system on X if and only if the following two conditions hold:*
(i) for all $x, y \in X$ distinct, there exists some $S \in \Sigma^-$ such that $S(x) \neq S(y)$,
(ii) $Incomp(\Sigma^-)$ is connected.
Moreover, if (i) and (ii) hold then there exists an unique, up to isomorphism and partial-resolution, simple 1-nested network N on X such that $\Sigma \subseteq \Sigma(N)$.

Proof. Let $X = \{x_1 \ldots, x_n\}$ and let σ denote the circular ordering on X under which x_i precedes x_{i+1} for all $i \in \{1, \ldots, n-1\}$, and x_n precedes x_1. Note that this circularity allows us to consider indices to be modulo n, meaning that in the following, we may write x_{n+1} for x_1, x_{n+2} for x_2, or x_0 for x_n. We may assume without loss of generality that Σ is circular for that ordering.

Assume first that (i) and (ii) hold. We first show that $\mathcal{I}(\Sigma^-)$ is maximal circular. To this end, it suffices to show that $\Sigma_\sigma \subseteq \mathcal{I}(\Sigma^-)$ since this implies that $\mathcal{I}(\Sigma_\sigma) \subseteq \mathcal{I}(\mathcal{I}(\Sigma^-)) \subseteq \mathcal{I}(\mathcal{I}(\Sigma)) = \mathcal{I}(\Sigma)$. Combined with the fact that, in view of Lemma 4.2.7, $\mathcal{I}(\Sigma_\sigma)$ is maximal circular, it follows that $\mathcal{I}(\Sigma_\sigma) = \mathcal{I}(\Sigma^-) = \mathcal{I}(\Sigma)$. Hence, $\mathcal{I}(\Sigma)$ is maximal circular.

Assume for contradiction that there exists some $i \in \{1, \ldots, n\}$ such that the split $S^* = x_i x_{i+1} | x_{i+2}, \ldots, x_{i-1}$ of Σ_2 is not contained in $\mathcal{I}(\Sigma)$. Then, by assumption, there exist two splits S and S' in Σ such that $S(x_i) \neq S(x_{i-1})$ and $S'(x_{i+1}) \neq S'(x_{i+2})$. Let $P_{SS'}$ denote a shortest path in $Incomp(\Sigma)$ joining S and S'. Without loss of generality, let S and S' be such that the path $P_{SS'}$ is a short as possible. Let $S_0 = S, S_1, ..., S_k = S'$ denote that path. The next lemma is central to the proof.

Lemma 4.2.11. *For all $0 \leq j \leq k$, we have $S_j(x_i) = S_j(x_{i+1})$.*

Proof. First observe that $S_j(x_i) = S_j(x_{i-1})$ and $S_j(x_{i+1}) = S_j(x_{i+2})$ must hold for all $0 < j < k$. Indeed, if there existed some $j \in \{1, \ldots, k-1\}$ such that $S_j(x_i) \neq S_j(x_{i-1})$ then the path $S_j, S_{j+1}, \ldots, S_k$ would be shorter than $P_{SS'}$, in contradiction to the choice of S and S'. Similar arguments also imply that $S_j(x_{i+1}) = S_j(x_{i+2})$ holds for all $j \in \{1, \ldots, k-1\}$.

Assume for contradiction that there exists $0 \leq j \leq k$ such that $S_j(x_i) \neq S_j(x_{i+1})$. Without loss of generality, we may assume that, for all $0 \leq l \leq j-1$, we have $S_l(x_i) = S_l(x_{i+1})$. Then since a trivial split cannot be incompatible with any other split on X we cannot have $j \in \{0, k\}$. Thus, the splits S_{j-1} and S_{j+1} must exist. Note that they cannot be incompatible, since otherwise the path from S to S' obtained by deleting S_j from $P_{SS'}$ is shorter than $P_{SS'}$ which is impossible. So S_{j-1} and S_{j+1} must be compatible. Clearly, $x_i \in S_{j+1}(x_i) \cap S_{j-1}(x_i)$. We next establish that $\overline{S_{j+1}(x_i)} \cap \overline{S_{j-1}(x_i)} = \emptyset$ cannot hold implying that either $S_{j+1}(x_i) \cap \overline{S_{j-1}(x_i)} = \emptyset$ or $\overline{S_{j+1}(x_i)} \cap S_{j-1}(x_i) = \emptyset$.

Indeed, let $q \in \{1, \ldots, n\}$ such that $S_j = x_{i+1} \ldots x_q | x_{q+1} \ldots x_i$. We claim that $x_q \in \overline{S_{j+1}(x_i)} \cap \overline{S_{j+1}(x_i)}$. Assume by contradiction that $x_q \in S_{j-1}(x_i)$ and that $i \leq q$. Then $S_{j-1}(x_i)$ is an interval of X containing $\{x_i, x_q\}$. Hence, either $S_{j-1}(x_i) \supseteq [x_i, x_q] \supseteq S_j(x_{i+1})$ or $S_{j-1}(x_i) \supseteq [x_q, x_i] \supseteq S_j(x_i)$. But both are impossible in view of the fact that S_{j-1} and S_j are incompatible.

Now assume that $S_{j+1}(x_i) \cap \overline{S_{j-1}(x_i)} = \emptyset$, that is, $S_{j+1}(x_i) \subseteq S_{j-1}(x_i)$. We postulate that then $S_{j+1}(x_i) \subseteq S_0(x_i)$ must hold which is impossible since $x_{i-1} \in S_{j+1}(x_i)$ and $S_0(x_i) \neq S_0(x_{i-1})$. Indeed, the choice of S and S' implies that S_{j+1} and S_l must be compatible, for all $0 \leq l \leq j-2$. By Lemma 4.2.4 applied to S_{j-1}, S_{j-2}, and S_{j+1} it follows that $S_{j+1}(x_i) \subseteq S_{j-2}(x_i)$. Repeated application of this argument implies that, for all $0 \leq l \leq j-2$, we have $S_{j+1}(x_i) \subseteq S_l(x_i)$, as required.

Finally, assume that $S_{j-1}(x_i) \cap \overline{S_{j+1}(x_i)} = \emptyset$, that is, $S_{j-1}(x_i) \subseteq S_{j+1}(x_i)$. Then similar arguments as in the previous case imply that $S_{j-1}(x_i) \subseteq S_k(x_i)$. But this is impossible since $x_{i+1}, x_{i+2} \in S_{j-1}(x_i)$ and $S_k(x_i) = S_k(x_{i+1}) \neq S_k(x_{i+2})$. Thus, $S_j(x_i) = S_j(x_{i+1})$ must hold for all $0 \leq j \leq k$. This concludes the proof of Lemma 4.2.11. □

Continuing with the proof of Theorem 4.2.10, we claim that the splits

$$T_j := T_{j-1}(x_i) \cap S_j(x_i) | \overline{T_{j-1}(x_i)} \cup \overline{S_j(x_i)}$$

where $j \in \{1, \ldots, k\}$ and $T_0 := S_0$ are contained in $\mathcal{I}(\Sigma)$. Assume for contradiction that there exists some $j \in \{0, \ldots, k\}$ such that $T_j \notin \mathcal{I}(\Sigma)$. Then $j \neq 0$ because $S \in \mathcal{I}(\Sigma)$, and $j \neq 1$ since $T_1 \in \iota(S, S_1)$ and $S, S_1 \in \Sigma$. Without loss of generality, we may assume that j is such that for all $1 \leq l \leq j-1$, we have $T_l \in \mathcal{I}(\Sigma)$. Then T_{j-1} and S_j cannot be incompatible and so $T_{j-1}(x_i) \subseteq S_j(x_i)$, or $S_j(x_i) \subseteq T_{j-1}(x_i)$, or $\overline{S_j}(x_i) \subseteq T_{j-1}(x_i)$ must hold. But $S_j(x_i) \subseteq T_{j-1}(x_i)$ cannot hold since then $\overline{S_{j-1}(x_i)} \subseteq \overline{T_{j-2}(x_i)} \cup \overline{S_{j-1}(x_i)} = \overline{T_{j-1}(x_i)} \subseteq \overline{S_j(x_i)}$ which is impossible as S_{j-1} and S_j are incompatible. Also, $\overline{S_j}(x_i) \subseteq T_{j-1}(x_i)$ cannot hold since then $\overline{S_j}(x_i) \subseteq T_{j-1}(x_i) = T_{j-2}(x_i) \cap S_{j-1}(x_i) \subseteq S_{j-1}(x_i)$ which is again impossible as S_{j-1} and S_j are incompatible. Thus, $T_{j-1}(x_i) \subseteq S_j(x_i)$ and so $T_j(x_i) = T_{j-1}(x_i)$. Consequently, $T_j = T_{j-1} \in \mathcal{I}(\Sigma)$ which is also impossible and therefore proves the claim. Thus, $T_j \in \mathcal{I}(\Sigma)$, for all $0 \leq j \leq k$. Combined with Lemma 4.2.11 it follows that, for all $0 \leq j \leq k$, we also have $T_j(x_i) = T_j(x_{i+1})$. Consequently, $\{x_i, x_{i+1}\} \subseteq T_k(x_i)$. Combined with the facts that $T_k(x_i)$ is an interval on X and $x_{i-1} \notin S_0(x_i)$, and similarly, $x_{i+2} \notin S_k(x_i)$ it follows that $\{x_i, x_{i+1}\} = T_k(x_i)$. Hence, $S^* = T_k \in \mathcal{I}(\Sigma)$, which is impossible. Thus, $\Sigma_\sigma \subseteq \mathcal{I}(\Sigma^-)$ and so $\mathcal{I}(\Sigma_\sigma) = \mathcal{I}(\Sigma^-)$.

Conversely, assume that $\mathcal{I}(\Sigma)$ is maximal circular. Then $\mathcal{I}(\Sigma)$ clearly satisfies Properties (i) and (ii) that is, for all $x, y \in X$ distinct there exists some $S \in \mathcal{I}(\Sigma)^-$ such that $S(x) \neq S(y)$ and $Incomp(\mathcal{I}(\Sigma)^-)$ is connected. We need to show that Σ also satisfies Properties (i) and (ii). Assume for contradiction that Σ does not satisfy Property (i). Then there exist $x, y \in X$ distinct such that for all splits $S \in \Sigma^-$, we have $S(x) = S(y)$. Let $S \in \mathcal{I}(\Sigma)$ such that $S(x) \neq S(y)$ and let $S_1, S_2, \ldots, S_l = S$ denote a sequence in $\mathcal{I}(\Sigma)$ such that $S_i \in \iota(S_{i-1}, S_{i-2})$, for all $3 \leq i \leq l$. Without loss of generality we may assume that l is such that $S_i(x) = S_i(y)$, for all $3 \leq i \leq l-1$. Then $S_j(x) = S_j(y)$, for all $j \in \{l-1, l-2\}$ and thus $S(x) = S(y)$ which is impossible.

Next, assume for contradiction that Σ does not satisfy Property (ii). Let Σ_1 and Σ_2 denote two disjoint connected components of $Incomp(\Sigma^-)$. For $i = 1, 2$, let $\mathcal{A}_i \in \pi_0(\mathcal{I}(\Sigma_i)^-)$ such that $\Sigma_i \subseteq \mathcal{A}_i$. Then, $2 \leq |\Sigma_i| \leq |\mathcal{A}_i|$, for all $i = 1, 2$. Combined with Lemma 4.2.9(ii), we obtain $\mathcal{A}_1, \mathcal{A}_2 \in \pi_0(\mathcal{I}(\Sigma)^-)$. Since $Incomp(\mathcal{I}(\Sigma)^-)$ is connected, it follows for $i = 1, 2$ that $\mathcal{A}_i \subseteq \mathcal{I}(\Sigma_i)^- \subseteq \mathcal{I}(\Sigma)^- = \mathcal{A}_i$. Thus, $\mathcal{I}(\Sigma_1)^- = \mathcal{I}(\Sigma^-) = \mathcal{I}(\Sigma_2)^-$ and so the incompatibility graphs $Incomp(\mathcal{I}(\Sigma_1)^-)$, $Incomp(\mathcal{I}(\Sigma)^-)$ and $Incomp(\mathcal{I}(\Sigma_2)^-)$ all coincide. Suppose $S \in \Sigma_1$ and $S' \in \Sigma_2$ and let P denote a shortest path in $Incomp(\mathcal{I}(\Sigma)^-)$ joining S and S'. Then there must exist incompatible splits S and S' in P such that $S \in \Sigma_1 \subseteq \mathcal{I}(\Sigma_1)^-$ and $S' \in \mathcal{I}(\Sigma_1)^- = \mathcal{I}(\Sigma_2)^-$ which is impossible in view of Lemma 4.2.8.

The remainder of the theorem follows from the facts that, by Lemma 4.2.7, $\mathcal{I}(\Sigma)$ is maximal circular that, by Lemma 4.2.6, there exists a simple level-1 network N

such that $\mathcal{I}(\Sigma) = \Sigma(N)$, that by Corollary 4.2.3(ii), a 1-nested network displays displays $\mathcal{I}(\Sigma)$ if and only if it displays Σ, and that the split system Σ_σ uniquely determines the underlying circular ordering of X. □

Armed with this characterization, we are now ready to establish Theorem 4.2.12.

Theorem 4.2.12. *Given a circular split system Σ on X, it is possible to build, in time $O(n(n + |\Sigma|^2))$, a 1-nested network N on X such that $\Sigma \subseteq \Sigma(N)$ holds and $|\Sigma(N)|$ is minimal. Furthermore, N is unique up to isomorphism and partial-resolution.*

Proof. Suppose Σ is a circular split system on X. Put $\{V_1, \dots, V_l\} = \pi_0(\Sigma)$. Without loss of generality we may assume that there exists some $j \in \{1, \dots, l\}$ such that $|V_i| = 1$ holds for all $1 \leq i \leq j-1$ and $|V_i| \geq 2$ for all $j \leq i \leq l$. Since $Incomp(\Sigma)$ has $l - j + 1$ connected components with at least two vertices there exist $l - j + 1$ simple 1-nested networks N_i such that $V_i \subseteq \Sigma(C_i)$ holds for the unique cycle C_i of N_i. By Theorem 4.2.10, it follows for all $j \leq i \leq l$ that $\Sigma(C_i) = \mathcal{I}(V_i)$ and that $Q_i \subseteq \mathcal{I}(V_i)$, where Q_i denotes the set of m-splits of C_i.

We claim that the split system Σ' on X given by

$$\Sigma' = \bigcup_{i=1}^{j-1} V_i \cup \bigcup_{i=j}^{l} Q_i \cup \bigcup_{x \in X} \{x|X - x\}$$

is compatible.

Since Σ is circular there exists a 1-nested network N on X such that $\Sigma \subseteq \Sigma(N)$. Without loss of generality, we may assume that N is such that $|\Sigma(N)|$ is minimal among such networks. For clarity of exposition, we may furthermore assume that N is maximal partially-resolved. Then for all $j \leq i \leq l$ there exists a cycle Z_i in N such that $V_i \subseteq \Sigma(Z_i)$. In fact, $\mathcal{I}(V_i) = \Sigma(Z_i)$ must hold for all such i. Combined with the minimality of $\Sigma(N)$, it follows that there exists a one-to-one correspondence between the cycles of N and the set $\mathcal{A} := \{\mathcal{I}(V_i) : j \leq i \leq l\}$ that maps a cycle C of N to the split system $\Sigma_C \in \mathcal{A}$ such that for some $i^* \in \{j, \dots, l\}$ we have $\Sigma_C = \mathcal{I}(V_{i^*})$ and $V_{i^*} \subseteq \Sigma(C)$. Furthermore, for all $1 \leq i \leq j-1$ there exists a cut-edge e_i of N such that the split S_{e_i} induced on X by deleting e_i is the unique element in V_i.

Let $T(N)$ denote the phylogenetic tree on X obtained from N by first shrinking every cycle Z of N to a vertex v_Z and then suppressing all resulting degree two vertices. Since this operation clearly preserves the splits in Q_i, $j \leq i \leq l$, and also does not affect the cut-edges of N (in the sense that a cut edge of $T(N)$ might correspond to a path in N of length at most 3 involving a cut-edge of N and one or two m-splits), it follows that $\Sigma' = \Sigma(T(N))$. Since any split system displayed by a phylogenetic tree is compatible the claim follows.

Since, in addition, Σ' also contains all trivial splits on X, it follows by the "Splits Equivalence Theorem" (see Section 4.1) that there exists a unique (up to isomorphism) phylogenetic tree T on X such $\Sigma(T) = \Sigma'$. Hence, $T(N)$ and T must be isomorphic. But then reversing the aforementioned cycle-shrinking operation that gave rise to $T(N)$ results in a 1-nested network N' on X such that $\Sigma(N) = \Sigma(N')$. Consequently, N' and N are isomorphic and so $\Sigma \subsetneq \Sigma(N')$. Note that similar arguments also imply that N is unique up to partial-resolution and isomorphism.

To see the remainder of the theorem, note first that finding $Incomp(\Sigma)$ can be accomplished in $O(n|\Sigma|^2)$ time. Combined with the facts that X has at most n cycles and any binary unrooted phylogenetic tree on X has $2n-3$ cut-edges it follows that N' can be constructed in $O(n^2 + n|\Sigma|^2)$ time. □

In consequence of Theorems 4.2.5 and 4.2.12, we obtain the 1-nested analogue of the "Splits Equivalence Theorem" (Theorem 1.2.3) for phylogenetic trees.

Corollary 4.2.13. *Suppose Σ is a split system on X that contains all trivial splits of X. Then there exists a 1-nested network N on X such that $\Sigma = \Sigma(N)$ if and only if Σ is circular and $\mathcal{I}$-intersection closed. Moreover, if such a network N exists then it is unique up to isomorphism and partial-resolution and can be constructed in $O(n(n + |\Sigma|^2))$ time.*

4.3 Optimality and the Buneman graph

We investigate here the interplay between the Buneman graph $\mathcal{B}(\Sigma)$ associated to a circular split system Σ and a 1-nested network displaying Σ. More precisely, we first associate to a circular split system Σ a certain subgraph of $\mathcal{B}(\Sigma)$ which we obtain by replacing each block of $\mathcal{B}(\Sigma)$ by a structurally simpler graph which we call a marguerite. As it turns out, marguerites hold the key for constructing optimal 1-nested networks from circular split systems.

4.3.1 The Buneman graph

The *Buneman graph* is a special type of split network (see Section 1.2.3). Among other properties, it is presented in [18] as a generalization of the "Splits Equivalence Theorem" (Theorem 1.2.3) to non compatible split systems and split-networks. We follow [16] for the definitions and terminology.

For Σ a split system on X, the vertices of the Buneman graph $\mathcal{B}(\Sigma)$ are the maps $\phi : \Sigma \to \mathcal{P}(X)$, where $\mathcal{P}(X)$ denote the power set of X, such that:

(B1) $\phi(S) \in S$ for all $S \in \Sigma$.

(B2) $\phi(S) \cap \phi(S') \neq \emptyset$ for any two distinct $S, S' \in \Sigma$.

As is easy to see, the vertex set of $\mathcal{B}(\Sigma)$ is nonempty, since for any $x \in X$, the associated *Kuratowski map* $\phi_x : S \mapsto S(x)$ satisfies both properties. Then, two vertices ϕ and ϕ' of this graph are joined by an edge if they differ in exactly one element, that is, if the set $\phi \triangle \phi' = \{S \in \Sigma : \phi(S) \neq \phi'(S)\}$ has size one.

Figure 4.7(ii) illustrates these definitions for the split system $\Sigma = \{S_1, \ldots, S_5\}$ on $X = \{1, 2, 3, 4, 5\}$ defined in Figure 4.7(i). Each vertex ϕ of that graph is represented as a binary sequence of size five, where for $1 \leq i \leq 5$, the i^{th} element of ϕ is 0 if $\phi(S_i)$ is the first element of S_i, and 1 otherwise. Note that this notation is arbitrary, as neither the split system Σ nor the splits it contains are ordered. As an example, the map 00011 on the top right corner is the map $\phi : \Sigma \to \mathcal{P}(X)$ defined by $\phi(S_1) = 12$; $\phi(S_2) = 1$; $\phi(S_3) = 15$; $\phi(S4) = 1245$; and $\phi(S_5) = 123$. In particular, ϕ is the Kuratowski map ϕ_1. The other Kuratowski maps are $\phi_2 = 01111$, $\phi_3 = 11101$, $\phi_4 = 11110$ and $\phi_5 = 11010$. Finally, note that two sequences are joined by an edge if and only if they differ in exactly one element, which is a direct consequence of this definition and the way an edge of the Buneman graph is defined.

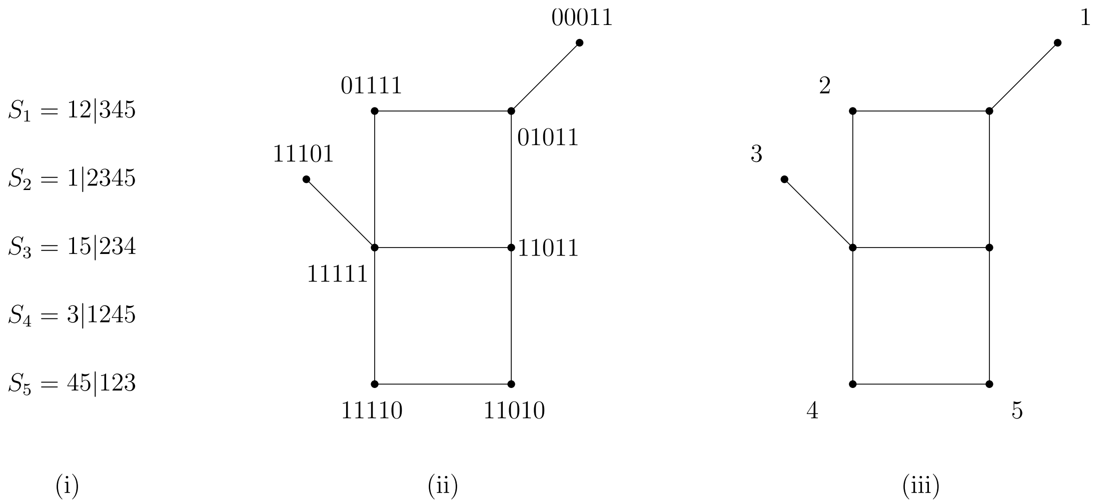


Figure 4.7: (i) A split system $\Sigma = \{S_1, \ldots, S_5\}$ on $X = \{1, \ldots, 5\}$. (ii) The Buneman graph $\mathcal{B}(\Sigma)$. (iii) The split-network representation of $\mathcal{B}(\Sigma)$. (see text for details).

The idea behind the definition is the following: Each edge $\{\phi, \phi'\}$ of $\mathcal{B}(\Sigma)$ is trivially associated to a unique split $S \in \Sigma$, that is, the split S satisfying $\phi(S) \neq \phi'(S)$. For a given split $S \in \Sigma$, we can then consider the set $E_S(\mathcal{B}(\Sigma))$ of edges of $\mathcal{B}(\Sigma)$ associated to S. This set of edges has two properties (see e. g. [16]):

It is nonempty, and the deletion of all edges of $E_S(\mathcal{B}(\Sigma))$ from $\mathcal{B}(\Sigma)$ divides $\mathcal{B}(\Sigma)$ into two connected components, one of which containing all the maps $\phi \in V(\mathcal{B}(\Sigma))$ such that $\phi(S) = A$, the others the maps $\phi \in V(\mathcal{B}(\Sigma))$ such that $\phi(S) = B$. In particular, one connected component contains the Kuratowski maps $\{\phi_x : x \in A\}$, and the other the maps $\{\phi_x : x \in B\}$. Thus, after identifying each Kuratowski map ϕ_x with the corresponding element $x \in X$, we get back to the definition of a split network representing Σ (see Figure 4.7(iii)). In particular, Σ is compatible if and only if $\mathcal{B}(\Sigma)$ is a tree.

It is also possible to define a distance $D_{\mathcal{B}(\Sigma)}$ on the vertex set of $\mathcal{B}(\Sigma)$. For any two vertices ϕ and ϕ', we can define $D_{\mathcal{B}(\Sigma)}$ as the number of splits $S \in \Sigma$ such that $\phi(S) \neq \phi'(S)$, that is, $D_{\mathcal{B}(\Sigma)}(\phi, \phi') = |\phi \triangle \phi'|$. Clearly, $D_{\mathcal{B}(\Sigma)}$ is a metric. By definition, ϕ and ϕ' are joined by an edge in $\mathcal{B}(\Sigma)$ if and only if $D_{\mathcal{B}(\Sigma)}(\phi, \phi') = 1$, and as we already noticed, the split $S \in \Sigma$ associated to an edge $\{\phi, \phi'\}$ is precisely the unique element of $\phi \triangle \phi'$. However, as a split network, $\mathcal{B}(\Sigma)$ enjoys the more general property that for two vertices ϕ and ϕ' and a split $S \in \phi \triangle \phi'$, any shortest path between ϕ and ϕ' crosses exactly one edge associated to S. In particular, there is a trivial bijection between the edges in a given shortest path between ϕ and ϕ' and the set $\phi \triangle \phi'$, and thus, $D_{\mathcal{B}(\Sigma)}(\phi, \phi')$ corresponds to the length of the shortest path between ϕ and ϕ'. This generalizes the distance induced by a split network on X over the set X.

If we consider as the vertex set the maps $\phi : \Sigma \to \mathcal{P}(X)$ satisfying (B1) only, the graph $\mathcal{H}(\Sigma)$ obtained using the same definition for edges is isomorphic to the hypercube of dimension $|\Sigma|$. To see that, we go back to the sequence notation introduced above. The vertex set of $\mathcal{H}(\Sigma)$ is the set of all binary sequences of size $|\Sigma|$, and two such sequences are joined by an edge if they differ in exactly one element. This is precisely the definition of the hypercube of dimension $|\Sigma|$, and this gives a trivial bijection between the splits in Σ and the dimensions of $\mathcal{H}(\Sigma)$.

Since the Buneman graph $\mathcal{B}(\Sigma)$ can be obtained from $\mathcal{H}(\Sigma)$ by removing all vertices not satisfying (B2), it can be seen as a subgraph of this hypercube. More precisely, it has been shown in [18] the $\mathcal{B}(\Sigma)$ is an isometric (and thus, connected) subgraph of $\mathcal{H}(\Sigma)$. As a consequence, we remark that $\mathcal{B}(\Sigma)$ is isomorphic to the hypercube $\mathcal{H}(\Sigma)$ if all splits in Σ are pairwise incompatible, since in that case, all maps $\phi : \Sigma \to \mathcal{P}(X)$ satisfying (B1) also satisfy (B2).

The Buneman graph of a split system Σ has many properties, and share links with different areas of mathematics. We focus here only on the properties playing a role in the following sections.

For a vertex ϕ of $\mathcal{B}(\Sigma)$, we denote by $\min(\phi(\Sigma))$ the set-inclusion minimal elements in $\phi(\Sigma) := \{\phi(S) : S \in \Sigma\}$ and by $\Sigma^{(\phi)}$ the set of pre-images of the elements in $\min(\phi(\Sigma))$ under ϕ. We have:

Proposition 4.3.1 ([19]). *Let Σ be a split system on X and ϕ a vertex of $\mathcal{B}(\Sigma)$.*

The vertices of $\mathcal{B}(\Sigma)$ adjacent to ϕ are precisely the maps $\phi' : \Sigma \to \mathcal{P}(X)$ such that $\phi'(S^*) = \overline{\phi(S^*)}$ for a given split $S^* \in \Sigma^{(\phi)}$, and $\phi'(S) = \phi(S)$, otherwise. In particular, $|\Sigma^{(\phi)}|$ is the degree of ϕ in $\mathcal{B}(\Sigma)$.

Proposition 4.3.1 provides an alternative way to build the Buneman graph $\mathcal{B}(\Sigma)$, without having first to list the maps $\phi : \Sigma \to \mathcal{P}(X)$ satisfying (B1) and (B2). Starting from any Kuratowski map ϕ_x, $x \in X$ as a single vertex, we can find the vertices adjacent to ϕ_x using the set $\Sigma^{(\phi_x)}$. We can repeat this process for each of the vertices created this way, until all vertices ϕ in the graph thus far constructed have degree $|\Sigma^{(\phi)}|$.

The next two properties link the Buneman graph of a split system Σ with the incompatibility graph $Inc(\Sigma)$ defined in Section 4.2.3. For any two distinct compatible splits $S = A|B$ and $S' = A'|B'$ of X there must exist a unique element in $\{A, B, A', B'\}$, say A, such that $A \cap A' \neq \emptyset$ and $A \cap B' \neq \emptyset$ both hold. Denoting that unique element by $\max(S|S')$, and by $\pi_0(\Sigma)$ the set of connected components of $Inc(\Sigma)$, we have:

Proposition 4.3.2 ([19]). *Let Σ be a split system on X. For $\Sigma_1, \Sigma_2 \in \pi_0(\Sigma)$ distinct we have* $\max(S_1|S_2) = \max(S_1|S_2')$, *for all $S_1 \in \Sigma_1$ and all $S_2, S_2' \in \Sigma_2$.*

In consequence, for $\Sigma_1, \Sigma_2 \in \pi_0(\Sigma)$ we can define the set $\max(\Sigma_1|\Sigma_2)$ as $\max(S_1|S_2)$, where the choice of $S_1 \in \Sigma_1$ and $S_2 \in \Sigma_2$ has no relevance. This then allows to state the next proposition:

Proposition 4.3.3 ([19]). *Let Σ be a split system on X. The biconnected components of $\mathcal{B}(\Sigma)$ are in 1-1 correspondence with the connected components of $Inc(\Sigma)$. More precisely, the map $\Theta : \Sigma_0 \in \pi_0(\Sigma) \mapsto \{\phi \in V(\mathcal{B}(\Sigma)) : \phi(S) =$ $\max(S|\Sigma_0)$ for all $S \in \Sigma - \Sigma_0\}$ is a bijection.*

As in Section 1.1.3, we now call a biconnected component of a Buneman graph $\mathcal{B}(\Sigma)$ a *block* of $\mathcal{B}(\Sigma)$. In Figure 4.2(ii), we indicate the blocks of the Buneman graph $\mathcal{B}(\Sigma(N))$ by B_1 and B_2 respectively.

We conclude this short review with introducing a notion that will be of particular interest to us in Section 4.3.3. A subset $Y \subseteq Z$ of a (proper) metric space (Z, D) is called a *gated* subset of Z if there exists for every $z \in Z$ a (necessarily unique) element $y_z \in Y$ such that $D(y, z) = D(y, y_z) + D(y_z, z)$ holds for all $y \in Y$. We refer to y_z as the *gate* for z in Y. We have:

Proposition 4.3.4 ([19]). *Let Σ be a split system on X and $\Sigma_0 \in \pi_0(\Sigma)$. The subgraph $B(\Sigma_0)$ of $\mathcal{B}(\Sigma)$ induced by Σ_0 is gated (with regard to the metric $D_{\mathcal{B}(\Sigma)}$). For every map $\phi \in V(\mathcal{B}(\Sigma))$, the map ϕ_{Σ_0} given by*

$$\phi_{\Sigma_0} : \Sigma(N) \to \mathcal{P}(X) : \quad S \mapsto \begin{cases} \phi(S) & \textit{if } S \in \Sigma_0 \\ \max(S|\Sigma') & \textit{otherwise,} \end{cases}$$

is the gate for ϕ in $B(\Sigma_0)$.

Continuing with the terminology of Proposition 4.3.4, we denote by $Gates(\mathcal{B}(\Sigma))$ the set of all vertices ϕ of $\mathcal{B}(\Sigma)$ for which there exists a block B of $\mathcal{B}(\Sigma)$ such that ϕ is the gate for some $x \in X$ in B.

4.3.2 Marguerites and Blocks

In this section, we first focus on the Buneman graph of a maximal circular split system and then introduce and study the novel concept of a marguerite.

To illustrate these definitions, consider again the Buneman graph depicted in Figure 4.2(ii) and the splits $S = 78|1\dots6$ and $S' = 18|2\dots7$ both of which are displayed by that graph. Then for the marked vertex ϕ, we have $\phi(S) = \{7, 8\}$. The block marked $\mathcal{B}_1$ in that Figure corresponds via Θ to the connected component $\Sigma_0 = \{S, S'\}$ and $\max(S'|\Sigma_0) = X - \{2, 3, 4\}$.

For the following, assume that $k \geq 4$ and that $Y = \{X_1, \dots, X_k\}$ is a partition of X. For clarity of exposition, also assume that $|X_i| = 1$, for all $1 \leq i \leq k$, and that the unique element in X_i is denoted by i. Further, assume that σ is the lexicographical ordering of X where we put $k+1 := 1$. Let Σ_k denote the maximal circular split system displayed by σ bar the trivial splits of X. Since Σ_k contains all 2-splits displayed by σ it follows that $|\pi_0(\Sigma_k)| = 1$. Hence, $\mathcal{B}(\Sigma_k)$ is a block in view of Property 4.3.3. To better understand the structure of $\mathcal{B}(\Sigma_k)$ consider for all $1 \leq i \leq k$ and for all $0 \leq j < k-3$ the map:

$$\phi_i^j : \Sigma_k \to \mathcal{P}(X) : \quad S \mapsto \begin{cases} \overline{S(i)} & \text{if } S(i) \subseteq [i-j, i] \\ S(i) & \text{otherwise.} \end{cases}$$

For example, for $k = 6$ and $k = 8$ the map ϕ_1^2 is indicated by a vertex in Figure 4.8(i) and (ii), respectively.

To establish the next result, we associated to every element $i \in X$ the split system $\Sigma(i)^+ := \{S \in \Sigma_k : S(i+1) = S(i) \neq S(i-1)\}$. Then the partial ordering "$\preceq_i$" defined, for all $S, S' \in \Sigma_k$, by putting $S \preceq_i S'$ if $|S(i)| \leq |S'(i)|$, is clearly a total ordering of $\Sigma(i)^+$ with minimal element $S_i^+ = [i, i+1]|X - [i, i+1]$

Lemma 4.3.5. *For any $k \geq 4$ the following statements hold:*

(i) For all $i \in \{1 \dots, k\}$ and all $0 \leq j < k-3$ the map ϕ_i^j is a vertex of $\mathcal{B}(\Sigma_k)$, $\phi_i^{k-3} = \phi_{i+1}^0$ holds, and $\Delta(\phi_i^j, \phi_i^{j+1}) = \{[i-j-1, i]|X - [i-j-1, i]\}$. In particular, $\{\phi_i^j, \phi_i^{j+1}\}$ is an edge in $\mathcal{B}(\Sigma_k)$.

(ii) For all $i \in \{1 \dots, k\}$ and all $1 \leq j < k-3$, the map

$$\psi_i^j : \Sigma_k \to \mathcal{P}(X) : \quad S \mapsto \begin{cases} \overline{\phi_i^j(S)} & \textit{if } S = S_i^+ \\ \phi_i^j(S) & \textit{otherwise.} \end{cases}$$

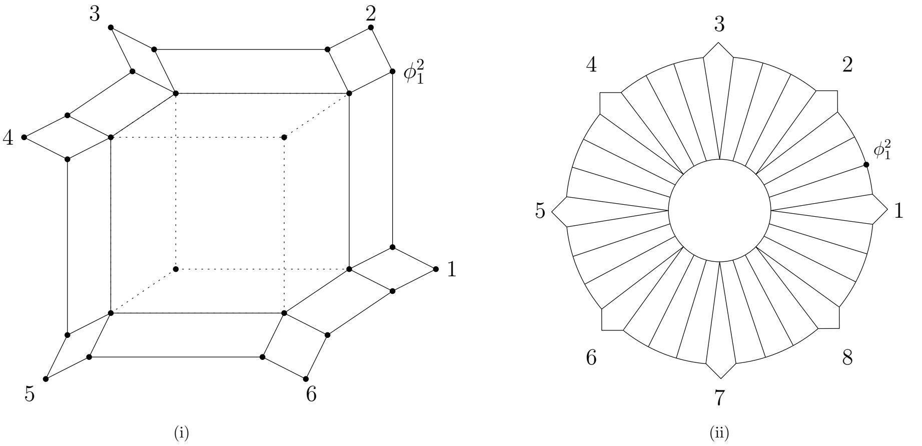


Figure 4.8: For $k = 6$, we depict in (i) the Buneman graph $\mathcal{B}(\Sigma_6)$ in terms of bold and dashed edges and the associated 6-marguerite $M(\Sigma_6)$ in terms of bold edges. In addition, we indicated the vertex ϕ_1^2 of $\mathcal{B}(\Sigma_6)$. We picture the 8-marguerite in (ii) and indicate again the vertex ϕ_1^2.

is a vertex in $\mathcal{B}(\Sigma_k)$ that is adjacent with ϕ_i^j. Moreover $\psi_i^{k-3} = \psi_{i+1}^0$ and $\{\psi_i^j, \psi_i^{j+1}\}$ is an edge in $\mathcal{B}(\Sigma_k)$.

Proof. (i) Suppose $i \in \{1 \dots, k\}$ and $0 \leq j < k-3$. To see that $\phi_i^j \in V(\Sigma_k)$, we distinguish between the cases that (a) $j = 0$, (b) $j = k-3$, and (c) $1 \leq j \leq k-4$. Let $i \in \{1, \dots, k\}$.

Assume first that (a) holds and let $S \in \Sigma_k$. Then $\phi_i^0(S) = S(i)$ must hold since Σ_k does not contain trivial splits. Moreover, $\phi_i^0(S) = \overline{S(i)}$ holds if and only if $S(i) \subseteq \{i\}$ if and only if S is the trivial split $i|X-i$. Thus, ϕ_i^0 is a vertex in $\mathcal{B}(\Sigma_k)$ in this case.

Assume next that (b) holds. We claim that $\phi_i^{k-3} = \phi_{i+1}^0$. Assume again that $S \in \Sigma_k$. Observe that since $i-(k-3) \equiv i+3 \pmod k$ we have $S(i) \subseteq \{i-(k-3), \dots, i\}$ if and only if $\{i+1, i+2\} \subseteq \overline{S(i)}$. We distinguish between the cases that (α) $S(i) = S(i+1)$ and (β) $S(i) \neq S(i+1)$.

Assume first that Case (α) holds, that is, $S(i) = S(i+1)$. Then $\{i+1, i+2\} \not\subseteq \overline{S(i)}$. Combined with the observation made at the beginning of the proof of this case, we obtain $S(i) \not\subseteq \{i-(k-3), \dots, i\}$ and, so, $\phi_i^{k-3}(S) = S(i) = S(i+1) = \phi_{i+1}^0(S)$.

Next, assume that Case (β) holds, that is, $S(i) \neq S(i+1)$. Then $i+1 \in \overline{S(i)}$. Since S cannot be a trivial split it follows that $i+2 \in \overline{S(i)}$ must hold too. Combined again with the observation made at the beginning of the proof of this case, it follows that $S(i) \subseteq \{i-(k-3), \dots, i\}$. Thus, $\phi_i^{k-3}(S) = \overline{S(i)} = S(i+1) = \phi_{i+1}^0(S)$ which completes the proof of the claim. In combination with Case (α), $\phi_i^{k-3} \in \mathcal{B}(\Sigma_k)$ follows.

So assume that (c) holds. Combining (a) with Property 4.3.1 and the fact that $\phi_i^0(S) = \phi_i^1(S)$ for all $S \in \Sigma_k - \{S_i^+\}$ and $\phi_i^0(S_i^+) = \overline{\phi_i^1(S_i^+)}$, it follows that ϕ_i^1 is a vertex of $\mathcal{B}(\Sigma_k)$. Similar arguments imply that if ϕ_i^l is a vertex in $\mathcal{B}(\Sigma_k)$ then so is ϕ_i^{l+1}. This concludes the proof of Case (c).

That $\Delta(\phi_i^j, \phi_i^{j+1}) = \{[i-j-1,i]|X-[i-j-1,i]\}$ holds for all $i \in \{1\dots,k\}$ and $0 \leq j < k-3$ is an immediate consequence of the construction.

(ii) Suppose $i \in \{1\dots,k\}$ and $1 \leq j < k-3$. Then ψ_i^j must be a vertex of $\mathcal{B}(\Sigma_k)$ that is adjacent with ϕ_i^j in view of Property 4.3.1 as $S_i^+ \in \Sigma^{\phi_i^j}$. That $\psi_i^1 = \psi_{i-1}^{k-3}$ holds is implied by the fact that the two splits in which ψ_i^{k-3} and ψ_{i+1}^1 differ from ϕ_{i+1}^0 are incompatible. That $\{\psi_i^j, \psi_i^{j+1}\}$ is an edge in $\mathcal{B}(\Sigma_k)$ follows from the fact that $\{\phi_i^j, \phi_i^{j+1}\}$ is an edge in $\mathcal{B}(\Sigma_k)$. □

Bearing in mind Lemma 4.3.5, we next associate to $\mathcal{B}(\Sigma_k)$ the *k-marguerite* $M(\Sigma_k)$ on X, that is, the subgraph of $\mathcal{B}(\Sigma_k)$ induced by the set of maps ϕ_i^j and ψ_i^l where $1 \leq i \leq k$, $0 \leq j < k-3$ and $1 \leq l < k-3$. We illustrate this definition for $k = 6, 8$ in Figure 4.8. Note that if k or X are of no relevance to the discussion then we shall simply refer to a k-marguerite on X as a *marguerite*.

Clearly, $\mathcal{B}(\Sigma_k)$ and $M(\Sigma_k)$ coincide for $k = 4, 5$. To be able to shed light into the structure of k-marguerites for $k \geq 6$, we require some more terminology. Suppose $k \geq 4$ and $i \in \{0, \dots, k\}$. Then we call a vertex of $M(\Sigma_k)$ of the form ϕ_i^0 an *external vertex*. Moreover, we call for all $0 \leq j < k-3$ an edge of $M(\Sigma_k)$ of the form $\{\phi_i^j, \phi_i^{j+1}\}$ an *external edge*. Note that since $M(\Sigma_k)$ is in particular a subgraph of the $|\Sigma_k|$-dimensional hypercube, any split in Σ_k not of the form $i, i+1|X - \{i, i+1\}$ is displayed in terms of four parallel edges of $M(\Sigma_k)$ exactly two of which are external.

4.3.3 Gates

In this section we establish that any partially-resolved 1-nested network N can be embedded into the Buneman graph associated to $\Sigma(N)$.

To be able to establish that any 1-nested partially-resolved network N can be embedded as a (not necessarily induced) subgraph into the Buneman graph $\mathcal{B}(\Sigma(N))$ associated to $\Sigma(N)$, we require again more terminology. Suppose N is a partially-resolved 1-nested network and v is a non-leaf vertex of N. Then v is either incident with three or more cut-edges of N, or there exists a cycle C_v of N that contains v in its vertex set. In the former case, we choose one of them and denote it by e_v. In addition, we denote by $x_v \in X$ an element such that e_v is not contained in any path in N from x_v to v. In the latter case, we define x_v to be an element in X such that no edge of C_v is contained in any path in N from v to x_v.

Theorem 4.3.6. *Suppose N is a 1-nested partially-resolved network on X. Then the map $\xi : V(N) - X \to Gates(\mathcal{B}(\Sigma(N)))$ defined by mapping every non-leaf vertex $v \in V(N)$ to the map*

$$\xi(v) : \Sigma(N) \to \mathcal{P}(X) : \quad S \mapsto \begin{cases} \max(S|\Sigma^*) & \text{if } S \in \Sigma(N) - \Sigma^* \\ S(x_v) & \text{else} \end{cases}$$

is a bijection between the set of non-leaf vertices of N and the gates of $\mathcal{B}(\Sigma(N))$ where $\Sigma^* = \{S_{e_v}\}$ if v is contained in three or more cut-edges of N and $\Sigma^* = \Sigma(C_v)^-$ else. In particular, ξ induces an embedding of N into $\mathcal{B}(\Sigma(N))$ by mapping each leaf x of N to the leaf ϕ_x of $\mathcal{B}(\Sigma(N))$ and replacing for any two adjacent vertices v and w of a cycle C of N of length k the edge $\{v,w\}$ by the path $\phi_i^0 := \xi(v), \phi_i^1, \dots, \phi_i^{k-3} := \xi(w)$.

Proof. Suppose N is a 1-nested network and put $\Sigma = \Sigma(N)$. To see that ξ is well-defined suppose $v \in V(N) - X$. Then v is either contained in three or more cut-edges of N or v is a vertex of some cycle C of N. In the former case we obtain $\{S_{e_v}\} \in \pi_0(\Sigma(N))$ and in the later we have $C = C_v$ and $\Sigma(C_v)^- \in \pi_0(\Sigma)$. In either case, the definition of the element x_v combined with Property 4.3.4 implies $\xi(v) \in Gates(\mathcal{B}(\Sigma))$.

To see that ξ is injective suppose v and w are two non-leaf vertices of N such that $\xi(v) = \xi(w)$. Assume for contradiction that $v \neq w$. It suffices to distinguish between the cases that (i) v and w are contained in the same cycle, and that (ii) there exists a cut edge e' on any path from v to w.

To see that (i) cannot hold, suppose that v and w are vertices on a cycle C of N. Then, $S(x_v) = \max(S|\Sigma(C)^-) = S(x_w)$ must hold for the m-split S obtained by deleting the two edges of C adjacent to v which is impossible. Thus (ii) must hold. Hence, there must exist a cut-edge e' on the path from v to w. Then $\xi(v)(S_{e'}) \neq \xi(w)(S_{e'})$ follows which is again impossible. Thus, ξ must be injective.

To see that ξ is surjective suppose $g \in Gates(\mathcal{B}(\Sigma))$. Then there exists some $x_g \in X$ and some block B of $\mathcal{B}(\Sigma)$ such that g is the gate for x_g in B. Let $\Sigma_B \in \pi_0(\Sigma(N))$ denote the connected component that, in view of Property 4.3.3, is in one-to-one correspondence with B. If there exists a cycle C of N such that $\Sigma(C)^- = \Sigma_B$ then let v_g be a vertex of N such that no edge on any path from v_g to x_g crosses an edge of C. Then, by construction, $\xi(v_g) = g$. Similar arguments show that $\xi(v_g) = g$ must hold if Σ_B contains precisely one split and thus corresponds to a cut-edge of N. Hence, ξ is also surjective and thus bijective.

The remainder of the theorem is straightforward. □

Theorem 4.3.6 implies that by carrying out the two steps (a) and (b) stated in Corollary 4.3.7 any 1-nested partially-resolved network N induces a 1-nested network $N(\Sigma(N))$ such that the split system $\Sigma(N(\Sigma(N)))$ induced by $N(\Sigma(N))$ is the split system $\Sigma(N)$ induced by N.

Corollary 4.3.7. *Let Σ be a split system on X for which there exists a 1-nested network N such that $\Sigma = \Sigma(N)$. Then $N(\Sigma)$ can be obtained from $\mathcal{B}(\Sigma)$ by carrying out the following steps:*

(a) For all $x \in X$, replace each leaf ϕ_x of $\mathcal{B}(\Sigma)$ by x.

(b) For all blocks B of $\mathcal{B}(\Sigma)$ that contain a k-marguerite M for some $k \geq 4$, first add the edges $\{\phi_i^0, \phi_{i+1}^0\}$ for all $i \in \{1, \dots, k\}$ where $k+1 := 1$ and then delete all edges and all vertices of B not of the form ϕ_i^0 for some $1 \leq i \leq k$.

We next show that even if the circular split system under consideration does not satisfy the assumptions of Corollary 4.3.7, steps (a) and (b) still give rise to a, in a well-defined sense, optimal 1-nested network.

Theorem 4.3.8. *Let Σ be a circular split system on X that contains all trivial splits on X. Then $N := N(\Sigma)$ is a 1-nested network such that:*

(i) $\Sigma \subseteq \Sigma(N)$,

(ii) $|\Sigma(N)|$ is minimal among the 1-nested network satisfying (i),

(iii) A vertex v of a cycle C of N is partially resolved if and only if the splits displayed by the edges of C incident with v belong to Σ.

Moreover N is unique up to isomorphism and partial-resolution.

Proof. (i) & (ii): Suppose for contradiction that there exists a 1-nested network N' such that $\Sigma \subseteq \Sigma(N')$ and $|\Sigma(N')| < |\Sigma(N(\Sigma))|$. Without loss of generality, we may assume that N' is such that $|\Sigma(N')|$ is as small as possible. Moreover, we may assume without loss of generality that N' and $N(\Sigma)$ are both maximal partially-resolved. To obtain the required contradiction, we employ Corollary 4.2.13 to establish that N' and $N(\Sigma)$ are isomorphic.

Since $\Sigma \subseteq \mathcal{I}(\Sigma)$ it is clear that $\mathcal{I}(\Sigma)$ contains all trivial splits of X. Furthermore, since Σ is circular, Corollary 4.2.3(i) implies that $\mathcal{I}(\Sigma)$ is circular. Since $\mathcal{I}(\Sigma)$ is clearly $\mathcal{I}$-intersection closed and, by Property (Bi), $\mathcal{I}(\Sigma)$ is the split system displayed by $G(\mathcal{I}(\Sigma))$ it follows that $\mathcal{I}(\Sigma)$ comprises all splits displayed by $N(\mathcal{I}(\Sigma))$. Hence, by Corollary 4.2.13, up to isomorphism and partial-resolution, $N(\mathcal{I}(\Sigma))$ is the unique 1-nested network for which the displayed split system is $\mathcal{I}(\Sigma)$.

We claim that $\mathcal{I}(\Sigma) = \Sigma(N')$ holds too. By Corollary 4.2.3(iii), we have $\mathcal{I}(\Sigma) \subseteq \Sigma(N')$. To see the converse set inclusion assume that $S \in \mathcal{I}(\Sigma)$. Then S is either induced by (a) a cut-edge of N' or (b) S is not an m-split and there exists a cycle C of N' that displays S. In case of (a) holding, $S \in \Sigma$ follows by the minimality of $|\Sigma(N')|$. So assume that (b) holds. Then there must exist some connected component $\Sigma_C \in \pi_0(\Sigma)$ that displays S. Hence, by Property 4.3.3, there exists some block B_C of $\mathcal{B}(\Sigma)$ such that the split system displayed by B_C is Σ_C. Hence, Σ_C is also displayed by $N(\Sigma)$. Since, as observed above, $\Sigma(N(\Sigma)) = \mathcal{I}(\Sigma)$ we also have $\Sigma(N') \subseteq \mathcal{I}(\Sigma)$ the claim follows.

(iii) Suppose C is a cycle of N and v is a vertex of C. Assume first that v is partially-resolved. Then there exists a cut-edge e of N that is incident with v. Note that the split S_e displayed by e is also displayed by the two edges of C incident with v. In view of Corollary 4.3.7, the cut-edges of N are in 1-1 correspondence with the cut-edges of $\mathcal{B}(\Sigma)$, so we obtain $S_e \in \Sigma$.

To see the converse assume that e_1 and e_2 are the two edges of C incident with v such that the split S displayed by $\{e_1, e_2\}$ is contained in Σ. Then S is compatible with all splits in $\Sigma - \{S\}$. By Property 4.3.3, it follows that there exists a cut-edge e in $\mathcal{B}(\Sigma)$ such that $S_e = S$. Combined with Corollary 4.3.7, it follows that v is partially resolved. □

4.4 Conclusion

The questions raised by this work are similar to the ones discussed at the end of Chapter 2. Indeed, here as well as in Chapter 2, we have extended existing results on phylogenetic trees to 1-nested networks, which can be seen as a first step towards a generalization to any type of phylogenetic network.

Split systems have proven to be very powerful for phylogenetic tree reconstruction. In the context of this, it turns out that one of the difficulties to obtain such a generalization lies in the fact that if the requirement for a split system Σ to be circular is dropped, then the structure of a network representing Σ becomes much more complex.

Uniqueness of the network representation of a split system also seems to become a major problem outside of the space of circular split systems. If we consider for example the split system Σ of all splits of the set $\{1, \dots, 7\}$, we have seen that the network N depicted in Figure 4.1 is a representation of Σ. However, any network obtained from N by a permutation of its leaves is also a representation of Σ. Since a network similar in essence to the network in Figure 4.1 can easily be constructed for any set X, and since this permutation property is independent of the size of X, this implies that for all set X, there exists at least $|X|!/2$ distinct phylogenetic networks representing the split system $\Sigma(X)$. Moreover, all these networks are isomorphic as graphs (as only leaf labels are modified), meaning that no criteria based on topological complexity can be used to differentiate between them.

Finally, we have seen that if Σ is circular, the minimal superset of Σ that can be represented by a phylogenetic network is unique, and is precisely the $\mathcal{I}$-intersection closure $\mathcal{I}(\Sigma)$ of Σ. This uniqueness is generally also lost when we do not require Σ to be circular.

In view of these observations, the results presented in this chapter suggest that circularity for split systems is a very special and powerful property. Thus, a generalization of these results to more general split systems does not seem to be straightforward, and probably requires the development of alternative tools and techniques.

Chap. 5

On introgression and multiple rooted networks

Adapted from:

> G. E. Scholz, A.-A. Popescu, M. I. Taylor, V. Moulton and K. T. Huber. OSF-Builder: A new tool for reconstructing and representing phylogenetic histories involving introgression. *submitted*,

except for Section 5.2.1, which provides an overview of the framework in which this work takes place. My personal contribution to this work has been to assess the performance of the algorithm OSF-Builder *applied to real biological datasets, and to conduct the simulation study. I also wrote the first draft for the parts of the paper related to both of these tasks, as well as the first draft for the description of the inner-working of the algorithm and the worked example. As a support to the paper comes a Python implementation of the algorithm* OSF-Builder*. The development of this algorithm, as well as the implementation of its first version, is due to Andrei-Alin Popescu. To this first version, I added a certain number of improvements, among which are the possibility to run the software on a species forest of more than two trees, a graphical output (of the type of Figure 5.1) to help "read" the results, and the possibility for the user to try (and compare the output from) different orderings of the trees in the species forest.*

This chapter proposes a new method to reconcile a set of species trees and an allele tree, using the novel concept of an "overlaid species forest". We introduce the algorithm OSF-Builder, aimed at finding such a reconciliation, and study its properties and performances.

5.1 Introduction

In this chapter, adapted from [54], we introduce and study a new type of event-based reconciliation problem. Roughly speaking, reconciliation problems can be seen as the research for an "embedding" of a rooted phylogenetic tree T_0 into a further one T_1, whose set of leaves are different but related in some way (see Section 5.2.1 for a formal definition). Such an embedding has to take into account the relationships between both sets of leaves, and to be optimal, it has to minimize an overall cost induced by a set of pre-given events.

As we shall see in Section 5.2.1, the solution of a reconciliation problem generally takes the form of a map from the vertex set of T_0 to the set of arcs and vertices of T_1. Motivated by some recent works on gene introgression (see e. g. [2] or [62]), we extend this approach to the problem of a reconciliation of an *allele tree* G into a set of lineage trees F, which we shall refer to as a *species forest* (see Section 5.2.2 for all the terminology). The choice to map the vertices of G "inside" a set F of rooted phylogenetic trees, rather than inside a single tree, results in the existence of arcs (u, v) of G such that u and v are mapped to distinct trees of F.

In the context of gene introgression, these arcs, which we shall refer to as *contact arcs*, can be thought of as postulated introgression events. To identify them, we present a new model for introgression which captures the idea that an allele can spontaneously arise within a lineage and that introgression is rare. To formalize this we allow the root of the allele tree G to be mapped to any vertex of a lineage tree (i. e. not necessarily its root) of F and invoke parsimony to minimize the number of contact arcs necessary to reconcile G with the trees in F. Adapting the Fitch-Hartigan algorithm ([25], see Section 5.3.1 for a description of that algorithm) enables us to then find an optimal embedding in G into the forest F, which we call an *Overlaid Species Forest* or *OSF* for short.

To represent an OSF, we essentially add new arcs, corresponding to the contact arcs postulated by the OSF, between the trees in F, thus mimicking the construction behind the idea of a tree-based network (see Section 1.1.3). The graph obtained this way differs from a phylogenetic network as defined in Section 1.1.2 in the fact that it may have more than one vertex of in-degree zero. In Figure 5.1 we present an example of an OSF that we computed for a set F consisting of 7 lineage trees $L_1, \ldots, L_7$ given in [46] (see Section 5.4.2 for the details). Here the lineage trees are depicted in different colors and the contact arcs are given as red dashed arrows. Note that the OSF has three roots, respectively the roots of the lineage trees L_1, L_2 and L_5.

We begin this chapter by defining and reviewing event-based reconciliation problems involving two rooted phylogenetic trees (Section 5.2.1). We then formalize the allele tree-species forest reconciliation problem (Section 5.2.2) and define all the concept and notation related to that problem. As a central tool for its reso-

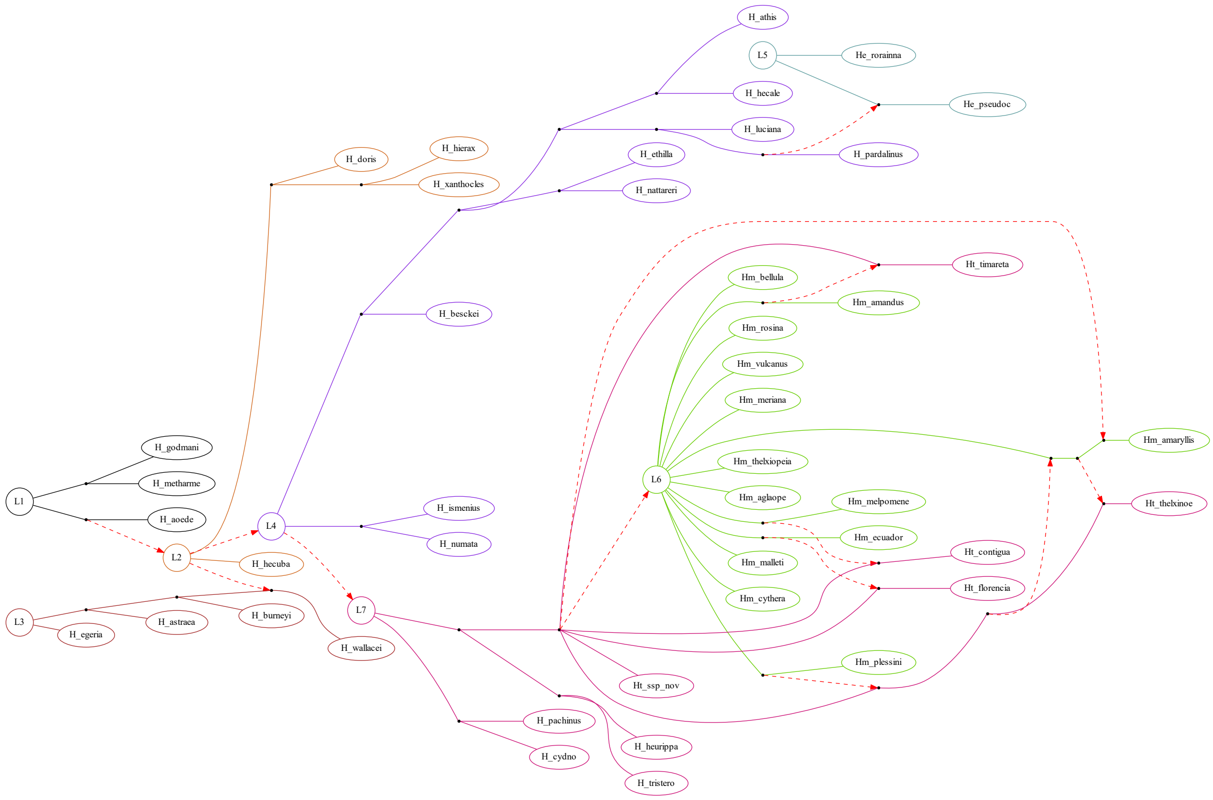


Figure 5.1: An OSF depicting introgression events of the dennis allele in [46] of the optix gene from the *Heliconius melpomene* lineage (L_7) into the *Heliconius timareta* lineage (L_6) and from lineage L_4 into the *Heliconius elevatus* lineage (L_5). To help readability, arcs are directed from left to right.

lution, we introduce in Section 5.3.1 the Fitch-Hartigan algorithm and present some of its key features. We conclude the first part by presenting our model for introgression and the OSF-Builder algorithm (Section 5.3.2).

Sections 5.4.1 and 5.4.2 present the output of OSF-Builder on two real biological dataset, *Scaevola* and *Heliconius* respectively, the latter being the one depicted in Figure 5.1. In addition to this, we use the *Scaevola* dataset as a worked example to describe in details the inner-working of the algorithm. Also, we use the *Heliconius* dataset to investigate the effect of the initial ordering of the trees in the species forest on the postulated contact arcs. Then, we conduct a simulation study to investigate the performances of OSF-Builder in the presence of noise. We present its setup and what is meant by “noise” in Section 5.5.1, and its results in case of noise in the allele tree and in the species forest respectively in Sections 5.5.2 and 5.5.3. For some theoretical results on the algorithm OSF-Builder and on the graph-representation of an OSF, some of which mentioned in this chapter, we refer the reader to Appendix A.

5.2 Preliminaries

5.2.1 Tree reconciliation

It is sometimes of interest to "compare" phylogenetic trees that are not defined on the same set of taxa. This may be the case, for example, if their respective set of leaves share a particular relationship, such that a host-parasite relationship (as in [5] for example), or an gene-species relationship (as in [15] and [51]). As such pairs can be expected to evolve together, the idea of reconciliation studies is to try to explain the evolution of one in terms of the other.

Given two rooted phylogenetic trees T_0 and T_1, on two sets X_0 and X_1 respectively, and a map $\phi : X_0 \to X_1$ relating the elements in X_0 with the elements of X_1, we call a map $\overline{\phi} : V(T_0) \to V(T_1) \cup E(T_1)$ that satisfies the following properties an *embedding* of T_0 into T_1:

(E1) For all $x \in X_0$, we have $\overline{\phi}(x) = \phi(x)$.

(E2) For all vertices u, v of T_0 such that $\overline{\phi}(u) \neq \overline{\phi}(v)$, if u is an ancestor of v, then there exists no directed path in T_1 from $\overline{\phi}(v)$ to $\overline{\phi}(u)$.

Condition (E1) ensures that $\overline{\phi}$ is an extension of ϕ, and condition (E2) guarantees that the ancestor relationship are, if not necessarily preserved, not broken. The underlying idea is to "represent" T_0 inside T_1, as suggested by Figure 5.2.

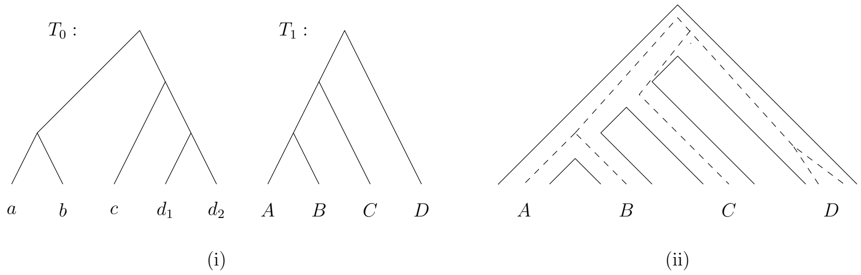


Figure 5.2: (i) Two phylogenetic trees T_0 and T_1 on $\{a, b, c, d_1, d_2\}$ and $\{A, B, C, D\}$ respectively. (ii) An embedding of T_0 (dashed) in T_1 (whose edges are represented as tubes).

Reconciliation problems aim at finding an embedding satisfying a certain number of conditions (see [63] for a detailed although non-exhaustive review). Most of the time, these conditions take the form of *event-based parsimony*. To better understand this concept, we will separate both parts of it.

The first part, *event-based*, means that we can interpret the behavior of $\overline{\phi}$ on specific parts of the tree T_0 as evolutionary events. Although the definition of these events may change from one author to another, the underlying idea is to compare, for an internal vertex v of T_0 with children[1] v_1 and v_2, the relative position of $\overline{\phi}(v_1)$ and $\overline{\phi}(v_2)$ with $\overline{\phi}(v)$ in T_1. For example, in [5], all internal vertices of T_0 are mapped to arcs of T_1. If v is such a vertex and $a = \overline{\phi}(v)$, we denote by w the head of a, by w_1 and w_2 the children of w if w is not a leaf of T_1, and by v_1 and v_2 the children of v (see Figure 5.3). Then, the authors of [5] consider four events, called cospeciation, duplication, loss and switch, respectively defined (up to a permutation of v_1 and v_2) as follows:

- If $\overline{\phi}(v_1) = (w, w_1)$ and $\overline{\phi}(v_2) = (w, w_2)$, the event is a *cospeciation* (Figure 5.3(i)).
- If $\overline{\phi}(v_1) = \overline{\phi}(v_2) = a$, the event is a *duplication* (Figure 5.3(ii)).
- If $\overline{\phi}(v_1)$ is an arc of T_1 that is neither (w, w_1) nor (w, w_2) but lies on a directed path from w to a leaf of T_1, the event is a *loss* (Figure 5.3(iii)).
- If $\overline{\phi}(v_1)$ is an arc of T_1 that can not be reached from w via a directed path, the event is a *switch* (Figure 5.3(iv)).

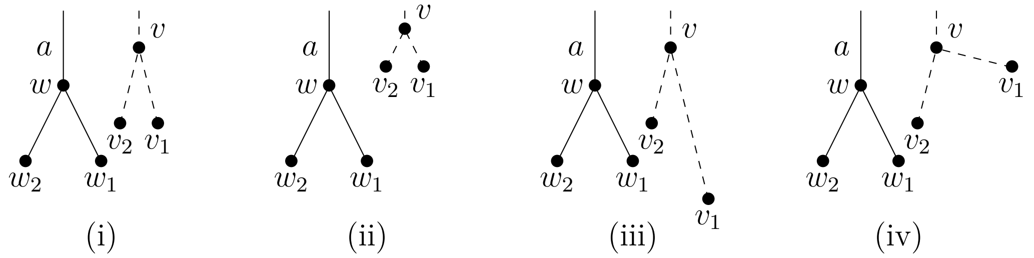


Figure 5.3: For a vertex v of a tree T_0 (dashed), four possible mappings of the children v_1 and v_2 of v to arcs of T_1 with regard to the image a of v in T_1. Each of these mappings corresponds to a particular co-evolutionary event: (i) Cospeciation. (ii) Duplication. (iii) Loss. (iv) Switch. See text for details.

[1] We assume here that both trees are binary.

Based on the knowledge indicating e. g. how likely an event is, a *cost* $c \in \mathbb{R}$ can be assigned to each of them. Then, the idea of a *parsimony* framework means that the goal is to find an embedding that minimizes the overall cost.

5.2.2 Model of introgression

We begin by introducing our model of introgression, which also involves introducing some terminology. The starting point of our model is the observation that in many introgression studies, lineage information is available for the species under consideration, and that these lineages are sometimes uniquely identifiable by the alleles of a gene not involved in the identification of those lineages. In case of introgression, species of a lineage carry both lineage specific and lineage foreign alleles of that gene making such species potential indicators of introgression. A useful way to represent introgression therefore is to add in branches that connect one lineage tree with another so that the information provided by both the lineage trees and the allele tree is displayed in a single structure. To obtain our model, we make the following additional biologically motivated assumptions.

(A1) Introgression is relatively rare.

(A2) An allele can only originate in one tree of the species forest.

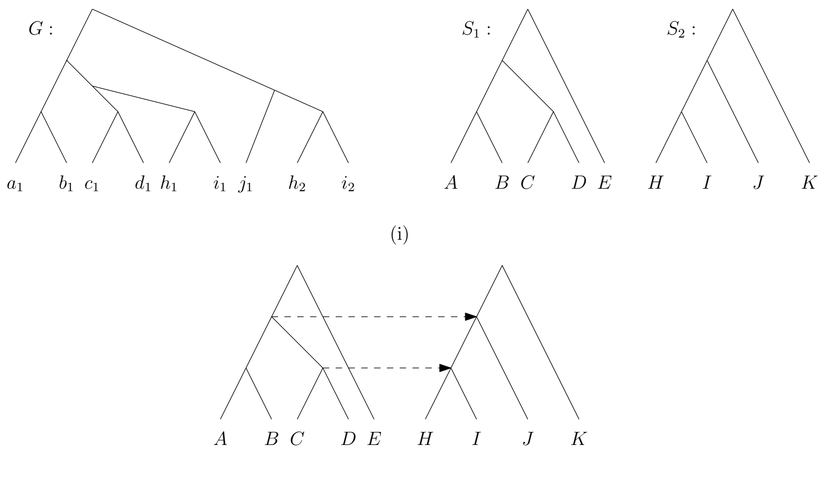
(A3) If an allele has introgressed from one lineage into a different lineage then it cannot introgress back into that lineage unless the start of the first introgression event precedes the end of the second one. (i. e. we do not allow directed cycles)

(A4) The allele composition of a species x is the sum of the allele that identifies the lineage x belongs to and all the alleles that x obtained via introgression events. (In particular, we do not allow losses).

(A5) The only other permissible evolutionary events are speciation and whole genome duplication.

Assumption (A1) motivates the use of a parsimony framework for modeling introgression, Assumption (A2) is motivated by the observation that lineages sometimes carry a specific allele and Assumption (A3) reflects time consistency. Assumption (A4) captures the idea that the aimed for model of introgression displays both the lineage trees and the allele tree. To formalize our problem subject to the assumptions above we require some terminology and notation. We distinguish in this chapter two types of phylogenetic trees, depending on the nature of the elements labelling their leaves. Thus, we call a phylogenetic tree whose leaves are species a *species tree*, and a phylogenetic tree whose leaves are alleles an *allele tree*.

We also call a set F of one or more species trees on pairwise distinct leaves sets a *species forest*, in which case we refer to the trees in F as lineage trees. Note that $V(F) = \bigcup_{T\in F} V(T)$ and that $L(F) = \bigcup_{T\in F} L(T)$. For F a species forest and G a gene tree, we refer to a map $\phi : L(G) \to L(F)$ as an *allele-species map*. This can be understood as the assignment to a given allele $a \in L(G)$ of the species $A \in L(G)$ that carries this allele. To facilitate readability, we always denote the species that contains an allele by the capitalization of its first letter, without the index if the allele has one (i. e. A for a or for a_1, Ch for ch). Finally, we refer to the triple (F, G, ϕ) as an *AS-forest* (for *Allele-Species forest*). Such an AS-forest is depicted in Figure 5.4(i).



(i)

(ii)

Figure 5.4: (i) An allele tree G and two species trees S_1 and S_2. (ii) An OSF for the trees in (i).

To formalize the aforementioned idea of adding additional arcs to the trees in a species forest F based on the information provided by an allele tree G, we use a map ψ from the set of vertices of G into the set of vertices of F. The purpose of that map is to capture, for every arc (u, v) of G, whether both u and v are contained in the same tree of F or not. To make this more precise, suppose F is a species forest, G is an allele tree, ϕ is an allele-species map and ψ is a map from $V(G)$ to $V(F)$. Inspired by the works on reconciliation problems described in Section 5.2.1, we call ψ an *overlaid species forest* or *OSF*, for short, for F and G and ϕ if ψ satisfies the following three properties:

(F1) If $u \in L(G)$ is an allele of species $U \in L(F)$, then $\psi(u) = U$.

(F2) If u and v are two vertices of G such that u is an ancestor of v in G, and

$\psi(u)$ and $\psi(v)$ belong to the same tree T of F, then $\psi(u)$ is an ancestor of $\psi(v)$ in T.

(F3) If v is a vertex of G such that $\psi(v)$ is a vertex of a given tree T of F, then there exists an offspring taxon g of v such that $\phi(g)$ is a species of T.

Note that in case a species forest F contains a single tree, then the problem of finding an OSF for F and G boils down to finding a reconciliation map for a species tree and a gene tree under parsimony where the only two permissible evolutionary events are speciation and duplication (see Section 5.2.1). Indeed, if the forest F contains a single tree, Property (F1) reduces to Property (E1), Property (F2) implies Property (E2), and Property (F3) trivially holds. Also, note that for any species forest F on X, any allele tree G and any allele-species map ϕ, there always exists an OSF for F, G and ϕ. Indeed, a map $\psi : V(G) \to V(F)$ satisfying $\psi(g) = \phi(g)$ for all leaves g of G, and such that $\psi(v)$ is the root of a given tree T in F for all internal vertices v of G satisfies Properties (F1) and (F2), and satisfies Property (F3) provided that T is chosen in such a way that the image set of ϕ contains at least one leaf of T.

A representation of an OSF ψ is a rooted directed acyclic graph H obtained from F by adding, for all edges (u, v) of G such that $\psi(u)$ and $\psi(v)$ belong to two distinct trees of F, the arc $(\psi(u), \psi(v))$. We call these new arcs *contact arcs* for this OSF. By abuse of terminology, we also refer to the network H as an *overlaid species forest (OSF)* for the AS-forest (F, G, ϕ). For the AS-forest depicted in Figure 5.4(i), we represent such a network in Figure 5.4(ii), in which the contact arcs are represented as dashed arrows.

5.3 Building an optimal OSF

As is easy to see, an OSF can potentially postulate a large number of contact arcs and therefore need not be optimal in the sense of Assumption (A1). To tackle the problem of finding an optimal OSF (in the sense of that assumption), we adapt the well-known Fitch-Hartigan algorithm [25].

5.3.1 The Fitch-Hartigan Algorithm

The Fitch-Hartigan algorithm has been introduced in [25]. Its initial purpose is to reconstruct an optimal, in a sense to be define, explanation of the observable diversity of a character shared by a set of taxa.

Given a rooted phylogenetic tree T on X with root ρ, and a map $\chi : X \to C$ from X to a nonempty set C as in Figure 5.5(i), the key point is to find an extension $\overline{\chi} : V(T) \to C$ of χ minimizing the number of arcs (u, v) of T such that

$\overline{\chi}(u) \neq \overline{\chi}(v)$. Such arcs are called *jumps*, and we denote this minimal number by $m(T, \chi)$. Figure 5.5(ii) and (iii) show two distinct such extensions, for the initial situation depicted in Figure 5.5(i).

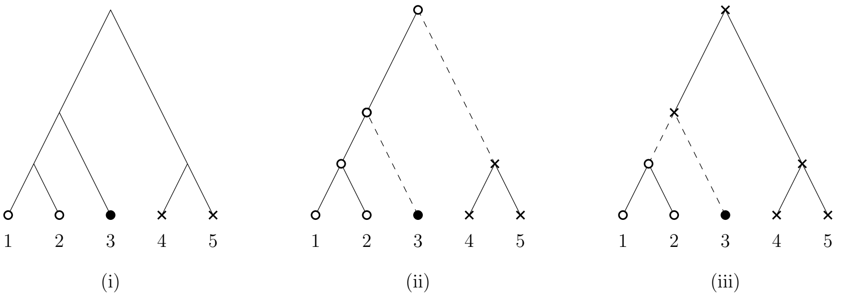


Figure 5.5: (i) A phylogenetic tree T on $X = \{1, 2, 3, 4, 5\}$ and a coloring of its leaves as a map $\chi : X \to C = \{\circ, \bullet, \times\}$. (ii) and (iii) Two distinct optimal extensions of χ to the vertex set of T. The jumps for these extensions are represented as dashed arcs. (see text for details).

The Fitch-Hartigan algorithm proceeds in two steps. The *Bottom-up* step recursively defines two maps: $\sigma : V(T) \to C$, which can be seen as a "first selection" of elements of C that can possibly be associated to a vertex $v \in V(T)$ and $l : V(T) \to \mathbb{N}$ recording the minimal number of jumps in the subtree rooted at v. We initialize these maps on X by putting, for all $x \in X$, $\sigma(x) = \{\chi(x)\}$ and $l(x) = 0$. Indeed, for $x \in X$, $\chi(x)$ is the only possible choice for $\overline{\chi}(x)$, and the subtree rooted at x does not contain any jump. Now consider a vertex v of T and assume that, for all children $v_1, \dots, v_k$, $k \geq 2$, of v the set $\sigma(v_i)$ and the value $l(v_i)$ are known. Then we put :

$$f(v) = \max_{c \in C} |\{i \in \{1, \dots, k\} : c \in \sigma(v_i)\}|$$

and define $\sigma(v)$ as the set of characters $c \in C$ realizing this maximum. In other word, $\sigma(v)$ is the set of character that are the most frequent among the children of v. We also put $l(v) = \sum_{i=1}^{k} l(v_i) + k - f(v)$. This step is depicted in Figure 5.6. Note that if v has only two children v_1 and v_2, we have $\sigma(v) = \sigma(v_1) \cap \sigma(v_2)$ if this intersection is nonempty, and $\sigma(v) = \sigma(v_1) \cup \sigma(v_2)$ otherwise. In the first case, we have $l(v) = l(v_1) + l(v_2)$, and in the latter, $l(v) = l(v_1) + l(v_2) + 1$. These observations make the recursive definitions of σ and l easier to handle in case the tree T we consider is binary.

Once all vertices of T have been processed, and the maps σ and l completely defined, the minimal numbers of jumps $m(T, \chi)$ of jumps an extension of χ realizes is given by $l(\rho)$. The *Top-Down* step of the Fitch-Hartigan algorithm then

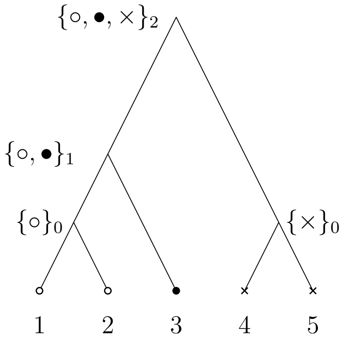


Figure 5.6: The Bottom-up step of the Fitch-Hartigan algorithm applied to the input (T, χ) depicted in Figure 5.5(i). Each internal vertex v is labelled by the set $\sigma(v)$, and the value of $l(v)$ is shown as index of this set (see text for details).

allows to find an extension $\overline{\chi}$ realizing this minimum. We start by choosing an element of $\sigma(\rho)$ which we define as $\overline{\chi}(v)$. Then, for all $v \in V(T)$ whith ancestor u for which $\overline{\chi}(u)$ has been defined, we put $\overline{\chi}(v) = \overline{\chi}(u)$ if $\overline{\chi}(u) \in \sigma(v)$. Otherwise, we can choose any element of $\sigma(v)$ to be $\overline{\chi}(v)$. The map depicted in Figure 5.5(ii) has been obtained this way from Figure 5.6 after assigning the symbol $\circ \in C$ to the root of T as an initial choice.

The Fitch-Hartigan algorithm is attractive due to its fast runtime. Indeed, processing all vertices of T can be done in $O(|X|)$ time, and for each of these vertices, all the characters in C have to be tested in the the Bottom-up step. This gives a runtime of $O(|X|.|C|)$ for the algorithm.

Clearly, as choices have to be made, the resulting map $\overline{\chi}$ is in general non unique. Moreover, there may exist extensions $\overline{\chi}$ of χ realizing this minimal number of jumps $m(T, \chi)$, but can not be recovered using the Fitch-Hartigan algorithm. This is the case of the extension $\overline{\chi}$ depicted in Figure 5.5(iii). A method for counting the number of optimal solutions for a given input has been developed in [4]. As this method presents little interest to this chapter and is quite technical, we describe it in Section A.3 for the sake of completeness.

5.3.2 The algorithm OSF-Builder

We next provide an outline of OSF-Builder and refer to Section 5.4.1 for a worked example. OSF-Builder takes as input a species forest F, an allele tree G and an allele-species map $\phi : L(G) \to L(F)$. We can assume without loss of generality that F, G and ϕ are such that for all trees T in F, there exists an allele $g \in L(G)$ such that $\phi(g) \in L(T)$. Put $\mathcal{F} = (F, G, \phi)$. Then OSF-Builder finds an overlaid species forest $\psi^*_{\mathcal{F}} : V(G) \to V(F)$ postulating a minimal number of contact arcs. For the following, we denote that minimal number by $t(F, G, \phi)$, or $t(\mathcal{F})$.

OSF-Builder works by first assigning to each allele v in $L(G)$ the tree P_v in F that contains the species $\phi(v)$ in its leaf set (Step 0). Referring to this assignment as a map $P : L(G) \rightarrow F$ from the leaf set of G into the trees of F defined by putting $P(v) := P_v$, the algorithm then applies the Fitch-Hartigan algorithm to G and P to find an extension $\overline{P} : V(G) \rightarrow V(F)$ of P to a map from the vertex set of G to the trees in F (Step 1). Note that by virtue of the Fitch-Hartigan algorithm, the number of arcs (u, v) of G for which the tree $\overline{P}(u)$ of F associated to u is distinct from the tree $\overline{P}(v)$ of F associated to v is as small as possible.

In the final step, OSF-Builder considers each interior vertex v of G in turn to obtain the vertex $\psi^*_{\mathcal{F}}(v)$ in the tree $\overline{P}(v)$ of F that v is assigned to via $\psi^*_{\mathcal{F}}$. More precisely, OSF-Builder first associates to every vertex v of G the subset $U_v \subseteq L(F)$ of all species in F that contain an offspring allele of v. Subsequent to this, it considers the subset U'_v of species in U_v that are also contained in the leaf set of the tree $\overline{P}(v)$ of F (i. e. $U'_v = U_v \cap L(\overline{P}(v))$). Note that, by virtue of the Fitch-Hartigan algorithm, the set U'_v is nonempty. Finally, the last common ancestor of the species in U'_v with regards to the tree $P(v)$ is taken by OSF-Builder to be $\psi^*_{\mathcal{F}}(v)$.

The construction of the map P is done using the Fitch-Hartigan algorithm, and as we have seen, this is done in $O(|L(G).|F|)$ time. The last part of OSF-Builder, which derives the map $\psi^*_{\mathcal{F}}$ from the map P, requires to consider for all internal vertices v the leaves that lie below v, so this second part runs in $O(|L(G)|^2)$ time. Recalling that we required for all tree T in F to have a leaf l belonging the the image of ϕ, we have $|F| \leq |L(G)|$, and thus, $O(|L(G).|F|) \subseteq O(|L(G)|^2)$. Then, the overall runtime of OSF-Builder is $O(|L(G)|^2)$.

Since we might have to break ties in the top-down step of the Fitch-Hartigan algorithm, an OSF resulting from the above approach need not be unique. To overcome this problem, we may assume that the trees in the species forest F are ordered in some way. In the case that we have to break a tie, we then do this in favor of the first one in that ordering, thus ensuring that for a given ordering κ of F, the output $\psi^{\kappa}_{\mathcal{F}}$ of OSF-Builder is unique. From the AS-forest depicted in Figure 5.4(i), we obtain the OSF depicted in Figure 5.4(ii) for the ordering $\kappa_1 = (S_1, S_2)$, and the OSF depicted in Figure 5.7(i) for the ordering $\kappa_2 = (S_2, S_1)$. The OSF depicted in Figure 5.7(ii) is also an optimal one, but cannot be obtained using OSF-Builder. The existence of such OSFs is due to the fact, mentioned before, that the Fitch-Hartigan algorithm may not recover all extensions $\overline{\chi}$ of a leaf-coloring map χ minimizing the number of jumps.

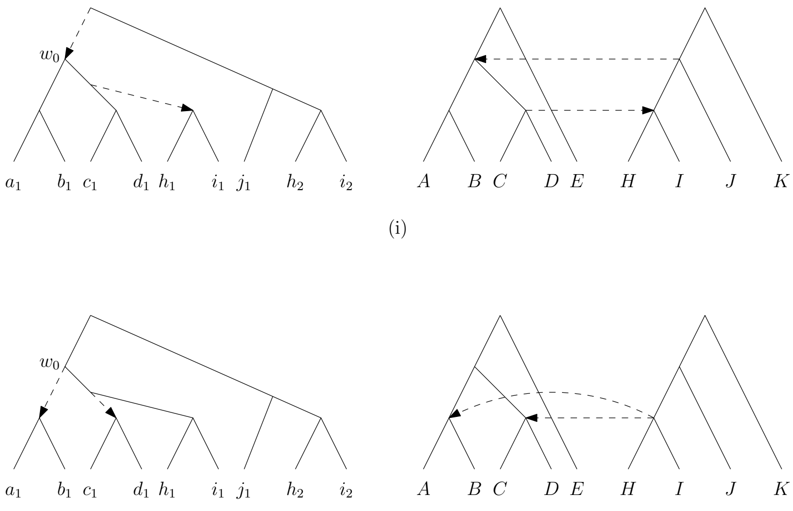

w_0
a_1 b_1 c_1 d_1 h_1 i_1 j_1 h_2 i_2
A B C D E H I J K
(i)
w_0
a_1 b_1 c_1 d_1 h_1 i_1 j_1 h_2 i_2
A B C D E H I J K
(ii)

Figure 5.7: (i) - (ii) Graphical representations of two distinct OSFs for the AS-forest $\mathcal{F} = (F, G, \phi)$ pictured in Figure 5.4(i). For both of them, we represent both the arcs in G that give rise to its contact arcs as well as the contact arcs themselves in terms of dashed arrows.

5.4 Real biological dataset

We now illustrate the applicability of OSF-Builder on two real biological data sets. The first is a small data set that concerns homoploid hybridization in *Scaevola (goodentaceae)*, which we use to explain how OSF-Builder works. The second is a larger data set and concerns wing pattern evolution in *Heliconous* butterflies.

5.4.1 The *Scaevola (goodentaceae)* dataset

Mounting evidence suggests that homoploid speciation has played an important role in plant evolution [37]. To help better understand this, the authors of [37] studied a clade of seven *Scaevola (goodeniaceae)* species found on the Hawaiian islands. For these species which include the hybrid species *Scaevola procera* and *Scaevola kilaueae* they constructed phylogenetic trees based on their alleles for the ITS region (among other regions). They attributed the discordance between trees to the hybrid origin of *S.procera* and *S.kilaueae* and summarized the resulting evolutionary scenario for all seven species in terms of a phylogenetic network. From this network we first derived a species tree and then used that tree to obtain a pair of lineage trees based on the clades in [37]. We depict that pair in Figure 5.8(i).

Using that pair of lineage trees as species forest and the phylogenetic tree for

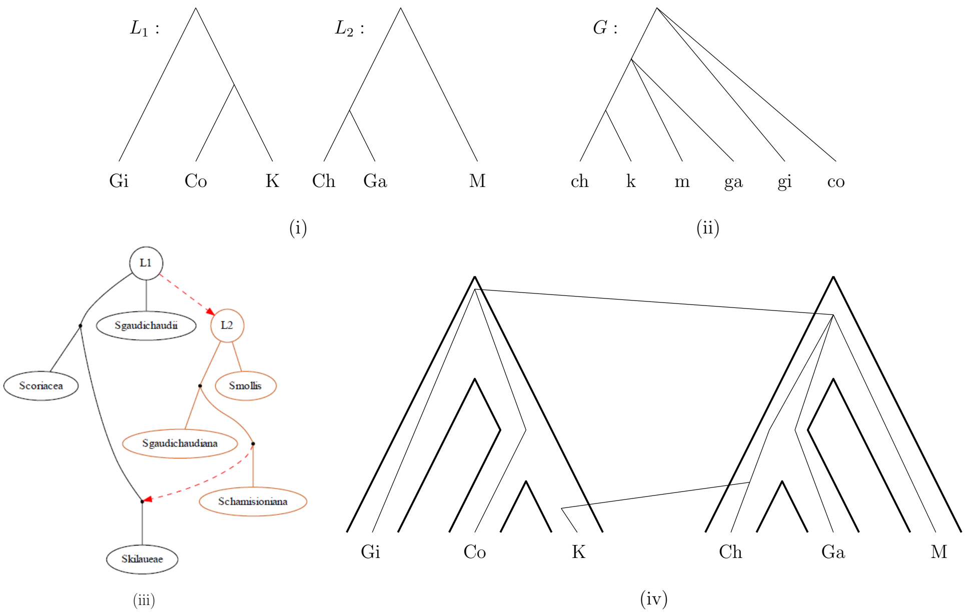


Figure 5.8: (i): The two lineage trees L_1 and L_2 obtained from the species network in [37]. (ii): The allele tree G obtained from the ITS region considered in [37]. (iii): The OSF found by OSF-BUILDER for the lineage tree pair and the allele tree in Figure 5.8(i) and (ii). (iv): A depiction of the found OSF in terms of the species tree (heavy edges) and the allele tree for the ITS gene where Gi is *Scaevola gaudichaudii*, Co is *Scaevola coriacea*, K is *S.kilaueae*, Ch is *Scaevola chamisioniana*, Ga is *Scaevola gaudichaudiana*, M is *Scaevola mollis*.

the ITS region of those six species from [37] as allele tree (see Figure 5.8(ii)), we computed an OSF which we depict in Figure 5.8(iii). In Figure 5.8(iv) we present an alternative representation of that OSF in terms of a tubelike structure in heavy edges (species tree) and a tree in lighter edges (allele tree). The bottom arc joining both lineage trees represents the introgression event between *S.coriacea* and *S.chamisioniana* that was postulated in [37] (giving rise to *S.kilaueae*). The second contact arc (i. e. the arc in the horizontal gray "tube" joining both lineage trees) is due fact that, as indicated in the figure, the two lineage trees are on sister clades of the species tree and therefore, that arc may or may not suggest introgression.

We now explain how OSF-BUILDER generates the OSF in Figure 5.8(iii) using Figure 5.9. Let L_1 and L_2 denote the two lineage trees depicted in Figure 5.8(i) that make up the species forest F, let G denote the allele tree in Figure 5.8(ii) and let ϕ denote the allele species map for G and F. Then OSF-BUILDER first applies the bottom-up step of the Fitch-Hartigan algorithm to assign to each vertex v of

G a set $\sigma(v)$ of trees of F. We illustrate this assignment in Figure 5.9(i), where we write i rather than L_i for $i = 1, 2$ to improve readability of the figure. For example, vertex v_1 of G has children k and ch which are both leaves mapped to tree L_1 and tree L_2, respectively. Thus, $\sigma(v_1) = \{L_1, L_2\}$. In the case of vertex u_1 of G, all three of its children v satisfy $L_2 \in \sigma(v)$, and only one satisfies $L_1 \in \sigma(v)$. Thus, $\sigma(u_1) = \{L_2\}$.

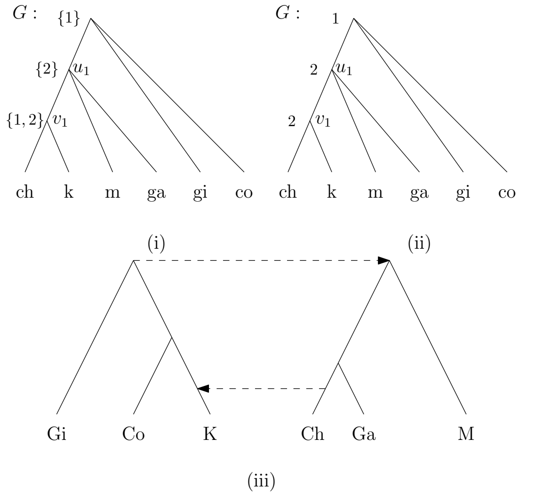


Figure 5.9: (i) and (ii): an illustration of the bottom up (i) and top-down phase (ii) of the Fitch-Hartigan part of the OSF-Builder algorithm applied to the lineage tree pair and the allele tree from Figure 5.8. (iii): The obtained OSF for that input.

Once all vertices of the allele tree have been assigned a set of trees in F this way, the top-down step of the Fitch-Hartigan algorithm associates the tree L_1 to the root ρ_G of G as that tree is the sole element in $\sigma(\rho_G)$. Next, it associates to each vertex v of G the tree T associated to its parent if $T \in \sigma(v)$ (case of v_1), and the unique tree in $\sigma(v)$ otherwise (case of u_1) -see Figure 5.9(ii).

It is clear from Figure 5.9(ii) that the generated OSF will have two contact arcs since it contains two branches for which the labels of the end nodes differ. These are the arc from the root of L_1 to u_1 and the arc from v_1 to k. Next, for each non-leaf vertex v of G the found label (e. g. lineage tree L_2 for u_1) is used to identify the vertex in F that is the last common ancestor of the species in that tree that carry an offspring taxa of v. For example, the offspring taxa of u_1 that are assigned to L_2 via ϕ are m, ga and ch. The species containing them are M, Ga and Ch respectively, and their last common ancestor in L_2 is the root of that tree. We depict in Figure 5.9(iii) the OSF found for G, F and ϕ. The found contact arcs are given in terms of dashed arrows. We remark in passing that we push out a leaf in a generated OSF in case that leaf is involved in a contact arc, thus preventing the creation of labelled vertices that are not leaves.

5.4.2 The *Heliconius* butterfly dataset

In [13] it was suggested that introgression played a role in the evolution of wing pattern in *Heliconius* butterfly species. To investigate this further, the authors of [62] studied the evolutionary relationships between 71 *Heliconius* butterfly species based on the dennis and ray allele, both of which are known to be implicated in wing pattern production. In particular, they found that the dennis allele introgressed from *H.melpomene* into *H.timareta* and from an ancestor of *Heliconius luciana*, *Heliconius pardalinus* and *H.elevatus* into *H.melpomene* and that the ray allele also introgressed from *H.melpomene* into *H.timareta*, and, in addition, also into *H.elevatus* (see [62]).

All four of these events were identified from a qualitative point of view using OSF-Builder (see Figure 5.1). Furthermore Figure 5.1 suggests that introgression between these lineages did not only occur once but multiple times including backcrossing and that the dennis allele might have introgressed into *H.melpomene* via *H.timareta*. The former is particularly interesting given that the authors of [62] aised the question if multiple introgression events might have occurred between these lineages. To better understand how the order used by OSF-Builder to resolve ties influences the construction of an OSF, we ran it with all possible (7!=5040) orderings of the forest. We depict our findings in Table 5.1, where we present the number of contact arcs supported by an OSF for that dataset.

	1	2	3	4	5	6	7
1	⊙	252	252	0	0	0	252
2	1008	⊙	2520	630	0	0	630
3	1008	2520	⊙	630	0	0	630
4	1008	630	630	⊙	2520	2520	2520
5	0	0	0	0	⊙	2520	0
6	0	0	0	0	2520	⊙	5040
7	1008	630	630	2520	0	5040	⊙

Table 5.1: For $1 \leq i \neq j \leq 7$, we present in line i column j the number of times we observe a contact arc from lineage L_i to lineage L_j, over all 5040 orderings of seven lineage trees $L_1, \ldots, L_7$.

Interestingly, out of the $42 = 7 \times 6$ possible contact arcs between the seven lineages we only observe 26 different ones and out of those only 10 occur more than half the time. Out of those 10, the contact arc from lineage *H.melpomene* (L_6) to *H.timareta* (L_7) and from (L_7) to (L_6) were recovered under every ordering suggesting that there is strong signal in the data concerning introgression between these two lineages. The remaining contact arcs of high frequencies involve lineages that are sister cherries in the species tree from [13] (see Figure 5.10 for a simplified version of that tree) and appear highly symmetric. This could be due to the fact that OSF-Builder breaks ties based on lineage age and that such information is not available from that species tree (i. e. out of the 5040 orderings, L_4 precedes lineage L_5 a total of $5040/2 = 2520$ times and so does lineage L_6 for lineage L_7).

From a parsimony point of view it also suggests that it might be of interest to investigate if introgression has indeed occurred between the ancestor of lineages L_4 to L_5 and the ancestor of lineages L_6 to L_7. The high frequency of the contact arcs involving lineages L_2 and L_3 could be an artifact of the OSF construction, since those lineages are "neighbors" in the species tree in [13] (see also Figure 5.10).

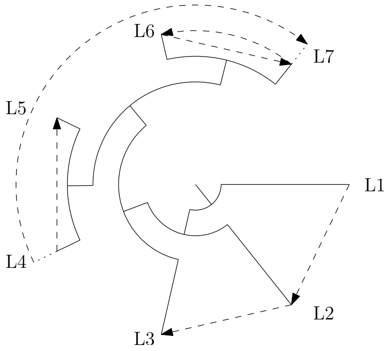


Figure 5.10: A simplified version of the *Heliconious* butterfly species tree from [13]. Contacts aecs postulated by OSF-Builder are represented as dashed arrows.

5.5 Simulation study

We now use a simulation study to assess the performance of OSF-Builder in the presence of noise.

5.5.1 Method

To study the effect of noisy input data we simulated two scenarios as follows. Using the software *SimCoal* [21] with default settings, we generated a phylogenetic tree T on 100 leaves. From that tree T, we derived a species forest F, by removing a random set of non-trivial arcs (and suppressing resulting in-degree and out-degree one vertices). Next, we added contact arcs between the trees in F to obtain an OSF, which we call OSF_1, from which we derive allele tree G and a map $\phi : L(G) \to L(F)$. Finally, we randomly simulate noise in G and in the trees within F, whilst fixing the other, to produce a new allele tree G' and species forest F', respectively. We repeated this several times, and we then computed an OSF for the pair so generated and compared it with the perfect scenario.

To simulate noise in an allele tree G, motivated by Theorem A.2.2, we rely on the notion of a SPR-operation (for *Subtree Prune and Regraft*), which we now describe.

Suppose T is a binary phylogenetic tree on X and T_0 is a proper subtree of T, that is, the root of T_0 is not the root ρ_T of T. Let $v \in V(T)$ denote the root of T_0. Then we refer to the following two step process as an *SPR-operation on v and T* (see e.g. [55]). In the first step, we delete T_0 and the incoming arc of its root. If v is not a child of ρ_T then we also suppress the resulting degree two vertex. If v is a child of ρ_T then we declare the other child w of ρ_T the root of the tree obtained from T by collapsing the outgoing arc of ρ_T. Let T' denote the resulting phylogenetic tree on $X - L(G)$. In the second step we either (i) subdivide an arc in T' by a new vertex w and add T_0 to T' via the new arc (w, v) or (ii) add a new incoming arc a to the root of T' and graft T_0 onto T' via the new arc $(tail(a), v)$. Note that in case the knowledge of v is of no relevance then we refer to an SPR-operation on T and v as just an *SPR-operation on T*.

We considered three separate cases to obtain a new allele tree G' from G. In the first case, we randomly applied one SPR-operation to G to obtain G', in the second case we applied three such operations to G and in the third case we applied five such operations to G. The original species forest and the respective allele trees G' we then used as input to OSF-Builder. Denoting the output of OSF-Builder for F and G' by OSF_2, we then repeated this process 100 times for each of the three chosen number of operations resulting in a total of 300 AS-forests. For each case we then measured the difference between OSFs OSF_1 and OSF_2 in terms of the difference of number of contact arcs postulated for G and G', respectively.

Using a similar approach, we simulated noise in the species forest F to obtain a new species forest F'. More precisely, we deleted a randomly chosen subtree from a randomly chosen tree T in F and then added that subtree to a tree in F other than T, repeating this process one, three and five times respectively. By abuse of terminology and due to the similarities of this operation with an SPR-operation on a tree, we shall refer to this process as an *SPR-operation* on F. Each of the resulting species forests F', we then combined with the allele tree G and used as input to OSF-Builder. Denoting the output of OSF-Builder for G' and F generated by OSF-Builder again by OSF_2, we then measured, again over 100 runs per considered case, the difference between OSFs OSF_1 and OSF_2 in terms

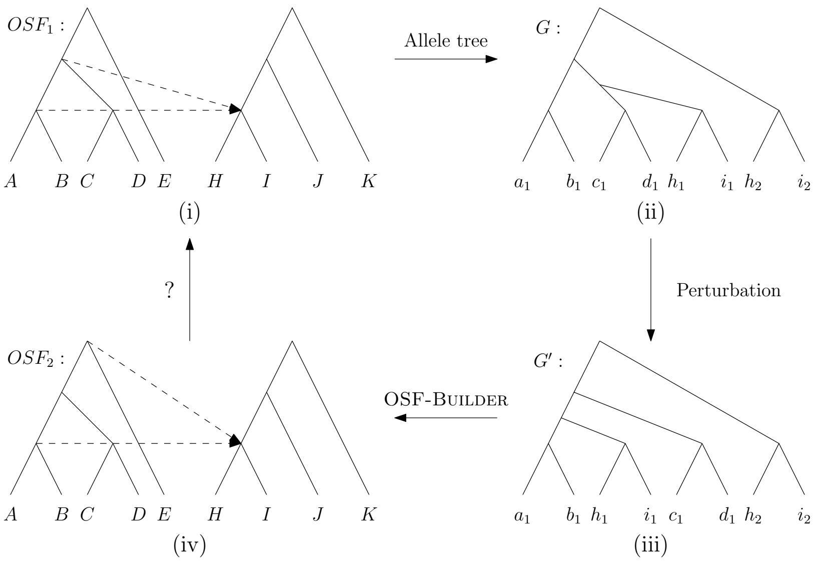


Figure 5.11: A schematic outline of our simulation study in the case of noise in the input allele tree. (i) An OSF OSF_1 (ii): An allele tree G obtained from OSF_1. (iii): The allele tree G' obtained from G by application of one SPR-operation to G. (iv) The OSF OSF_2 generated by OSF-Builder when given G' and F as input.

of the difference of number of contact arcs postulated for F and F', respectively.

5.5.2 Noise in the allele tree

For the case where the species forest F is fixed and the allele tree G is varied, we depict in Figure 5.12 the distribution of the differences $t(F, G', \phi) - t(F, G, \phi)$. Here, G' is an allele tree obtained from G by applying to G one, three, or five SPR-operations. As expected, the larger the number of operations is the more the number of contact arcs differs between G and G'. Having said this, in the majority of the cases, and irrespective of the number of SPR-operations, this difference is in terms of at most one contact arc. A potential reason for this might be that, mimicking the data sets analyzed in [46], the number of taxa for the trees in F is very diverse ranging from a tree in F on 74 taxa to a tree on just two taxa. Put differently, trees with a large number of taxa have a higher chance of being affected by an SPR-operation on G than ones with a small number of leaves. Consequently, an SPR-operation on G can result in the cutting and regrafting of a subtree G' of G onto G such that the affected parts of G and G' are mapped into the same tree in F under their respective OSFs. Since contact arcs can only be added by the OSF-Builder algorithm between trees of F and not within a tree of F, the difference in topology between G and G' does not therefore contribute to the number of postulated contact arcs for G'.

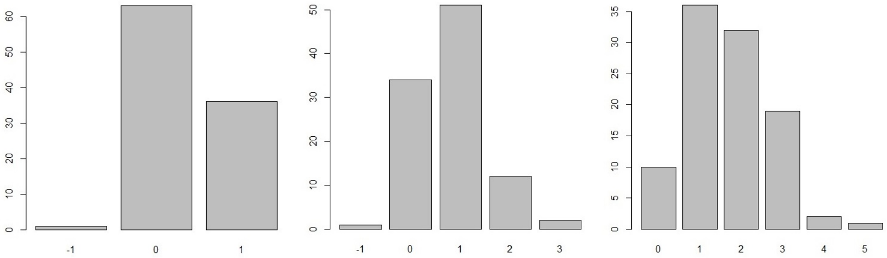


Figure 5.12: For F, G, G' and ϕ as described in the text, we depict the distribution of $t(F, G', \phi) - t(F, G, \phi)$ over 100 runs. The x-axis is labelled by $t(F, G', \phi) - t(F, G, \phi)$ and the y-axis gives the percentage for how often a difference is observed. Note that the value of -1 on the x-axis means that G' postulates one contact arc less than G.

The fact that the two trees G and G' underpinning Figure 5.12 have a similar number of contact arcs does not necessarily imply that they have contact arcs in common. This is because the optimality criterion employed by the OSF-Builder algorithm to find an OSF for G is dependent on the topology of G and that topology can be substantially affected by an SPR-operation.

In Figure 5.13, we depict the number of contact arcs that are common to G and G'. Note that the bars are not to be understood in a cumulative manner but as absolute values, i. e. the OSFs containing k of the contact arcs common to G and G' are not included in the count of the OSFs that contain $k+1$ of the contact arcs common to G and G'.

In line with our observations for Figure 5.12, the number of times that a contact arc is shared by G and G' is highest when the number of operations applied to G to obtain G' is small. In fact, if we applied only one such operation to G, then in 60% of the cases all eight of the contact arcs postulated by G are recovered by OSF-Builder when given G' as the gene tree. Furthermore, in a reassuring 92% of the cases at least seven of the eight contact arcs postulated for G were recovered when given G'.

5.5.3 Noise in the species forest

We now turn our attention to the case where the allele tree G is fixed and the noise affects the species forest F. We depict in Figure 5.14 the distribution of the

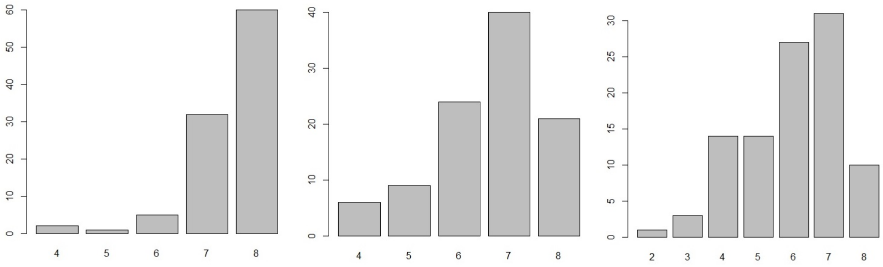


Figure 5.13: For F, G and G' as described in the text, we depict the distribution of the number of original contact arcs recovered, for 100 runs (see text for details). The x-axis is labelled by number of potential contact arcs and the y-axis by the percentage of times that such a number was observed. For ease of readability, contact arc numbers which were not observed are omitted.

differences $t(F', G, \phi) - t(F, G, \phi)$ where F' is a forest obtained from F by applying one, three, or five SPR-operations to F, respectively.

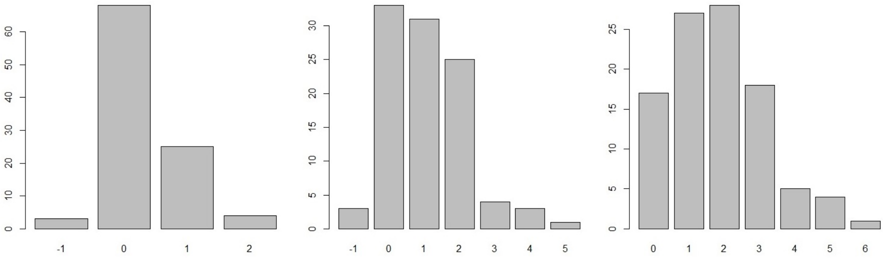


Figure 5.14: For F, F' and G as described in the text, we depict the distribution of $t(F', G, \phi) - t(F, G, \phi)$ over 100 runs (see text for details). The labelling of the axis is as in Figure 5.12.

The distribution highlighted by Figure 5.14 is similar to the one observed in Figure 5.12, in the sense that the larger the number of operations applied to F is, the more F and F' differ in their number of contact arcs. Taken together, Figs. 5.12 and 5.14 suggest that, in general, the difference $t(F', G, \phi) - t(F, G, \phi)$ is higher than $t(F, G', \phi) - t(F, G, \phi)$ when where F' and G' are obtained from F and G respectively via the same number of SPR-operations. In other words, it seems that noise in the species forest has a greater effect on OSF-Builders ability to recover an OSF than noise in an allele tree. A potential reason for this might be that the applied SPR-operations require the OSF-Builder algorithm to break up an "introgressed" subtree of G (i. e. the root of that subtree is the tip of an contact arc) in F' whereas this subtree is not broken up in F, thus increasing the number of contact arcs.

This observation is consistent with our theoretical observations, summarized in Theorems A.2.3 and A.2.2, that the range of values for $t(F', G, \phi) - t(F, G, \phi)$ is larger than the range of values for $t(F, G', \phi) - t(F, G, \phi)$.

5.6 Conclusion

In this chapter, we have presented a new approach that allows us to generate networks for exploring data were introgression is suspected to have occurred. We have investigated it both from a theoretical and a practical point of view involving two biological datasets and simulation studies. Our results are encouraging and indicate that OSF-Builder could be a useful new tool for studying this phenomenon. Moreover, the approach is fast (e. g. for the *Heliconious* butterfly dataset it took 3.8s to run on a HP laptop running Windows 10), which should be useful in light of the ever increasing amounts of data.

As we have seen, OSF-Builder depends on the ordering of the trees in the input forest which it employs to break ties. The default ordering in its current version is based on the age of the lineage trees in the forest. Bearing in mind that lineages correspond to subtrees in the species tree T from which they were derived, we break ties between lineages in favor of a lineage for which that root is closest to the root of T. In case there is still a tie (which can e. g. happen if two lineages are part of a cherry in T) then we randomly break it. In view of the fast runtime of OSF-Builder, the user can run the program several times each time changing that order (as in the *Heliconious* butterfly example in Section 5.4.2) to get an idea of which contact arcs are more common than others. Even so, it might be of interest to develop alternative ways to choose orderings for the lineages trees to allow OSF-Builder to break ties. For example, it could be worth investigating weighting schemes for the lineage trees based on either confidence values for the trees or the size of their leaf sets (or a combination of both). In a similar vein, it could also be possible to investigate how an introgression scenario is affected by the choice of lineage trees. For example a lineage tree could be replaced by sub-lineages corresponding to subtrees of the species tree whose roots are further away from the root of that tree.

The OSF-Builder method has some limitations. For example, the simulations suggest that it can be difficult to detect deep events. This is probably to be expected since deeper signals tend to get confounded by more recent ones. Moreover, contact arcs found by OSF builder can be due to the fact that the allele tree shares branches with the species tree from which the lineages were derived (as in the *Scaevola* data set in Section 5.4.1), and such shared branches are more likely to occur closer to the root of the allele tree. It might therefore be interesting to develop ways to distinguish between contact arcs that are shared and represent true introgression events from those that do not. A starting point for this might be to compare the set of contact arcs of an OSF with the arc set of the species tree from which the lineage trees underlying the OSF were derived. Another limitation of our approach is the use of parsimony. Hence, it might be worth exploring if our optimization criterion could be replaced with a more sophisticated one which takes into account costs of introgression events. For example, different costs could be assigned to introgression events based on their source trees and/or their recipient trees. Developing a probabilistic model might also be useful in this regards. In addition, it could be of interest to try extending our underlying model to include other types of events (such as losses or horizontal gene transfer, which in that case would correspond to a contact arc within a single lineage) as those described in Section 5.2.1.

Finally, from a more theoretical point of view, it might be of interest to develop ways that would allow one to compare OSFs and to better understand their combinatorial properties. Appendix A presents some first results in that direction.

Conclusion and future work

A common theme for all scientific research is the fact that new results lead to new questions, and answering them raises even more new and interesting questions. The new results contained in this thesis contribute to in this ever-expanding pattern.

As an illustration of that fact, we have successfully expanded, in Chapters 2, 3 and 4, some of the results reviewed in Chapter 1 to more general frameworks. In Chapter 2 we generalized some results that are known for phylogenetic trees to 1-nested networks, a type of phylogenetic networks that both enjoy attractive combinatorial properties and are relevant to evolutionary biology. More precisely, we addressed the question of the representability of symbolic maps, that are maps with range a set of symbols as opposed to real-valued maps, by such networks, when phylogenetic trees turn out to be inappropriate. Such type of maps arise, for example, in the context of reconciliation problems using orthology relations.

Chapter 3 remains within the realm of phylogenetic trees. However, it expands the notion of a symbolic distance, that is, a map returning a non-numeric value for a set of two elements,to the notion of a symbolic 3-way map, which takes as input a set of three elements instead of two. In particular, we addressed the question of the representability of such a map by a phylogenetic tree. For this, we distinguished between two cases, unrooted trees and rooted trees, and we successfully settled this question for both cases.

Chapter 4 follows the same pattern as Chapter 2, in that it generalizes results on phylogenetic trees to 1-nested networks. More precisely, we addressed the question of the representability of split systems, that is, sets of bipartitions of a set. Called splits, these structure can arise, for example, from morphological data for a taxa set. In addition to characterizing those split systems that can be represented by 1-nested networks, we have also proposed a “bridge” between two distinct ways of representing a split system.

Thus, by going deeper into questions that have been answered for simple cases, we successfully developed in these three chapters some alternative that may be used in situations where these simple cases turn out to be inapplicable or unsatisfactory. By doing so, new questions have arisen, along the line of “what could the next step be?”, “what if these new approaches are still inapplicable?”, or “what can we learn from these results?”. Some of these questions are proposed in the conclusion of each chapter, together with some leads towards their resolution, and an anticipation of the difficulties possibly lying ahead.

In Chapter 5, we have proposed a new method for reconciling a set of phylogenetic trees with a further one, inspired by existing reconciliation methods. In particular, we were interested in devising a method that allows us to infer introgressions events, which can be seen from a graphical point of view as “links” joining a phylogenetic tree with a further one. That method is accompanied with an implementation, an assessment of its performance on both synthetic and real data, and some theoretical results (which, in order to ease the reading of Chapter 5, we present in Appendix A). As mentioned at the end of Chapter 5, improving that method and understanding its properties, by expanding the aforementioned theoretical results, are two challenges among others. In particular, we proposed at the end of this chapter some directions into which some improvements of the method could be undertaken.

Along the way, this work has also lead to the introduction of a new type of phylogenetic network, which we called multiple rooted networks, admitting more than a single root. The space of multiple rooted networks stands in itself as a new field of study for mathematicians and computer scientists. The definition of such a network as a phylogenetic network for which the requirement of having only one root is dropped is, in many aspects, similar to the definition, given in the introduction, of a phylogenetic network as a phylogenetic tree for which the property of not containing a cycle is dropped. Put differently, a (single-rooted) phylogenetic network is a particular case of a multiple rooted phylogenetic network in the same respect that a phylogenetic tree is a particular case of a phylogenetic network. As we have seen throughout this thesis, of which it is one of the key features, many theoretical results on phylogenetic trees have been expanded to phylogenetic networks since their introduction. Given that multiple-rooted networks turned out to be useful for representing introgression scenarios, the question becomes, how to extend theoretical results on single-rooted phylogenetic networks to multiple rooted ones.

Appendix A

Some properties of OSF-Builder

We present here some key theoretical results for the algorithm OSF-Builder, introduced in Chapter 5 (Section 5.3.2). Most of these results appear in the Appendix of [54], that is, the paper on which Chapter 5 is based. In particular, we follow all definitions and terminology introduced in Chapter 5. All of the results (apart from the work described in Section A.3 which, as mentioned in the introduction of that section, comes from [4]) are the product of my own work.

Each of the Section is dedicated to an attractive property enjoyed by the algorithm OSF-Builder. First, we show in Section A.1 that the OSF we construct for a given set of lineage trees F and allelle tree G is optimal in the number of contact arcs that are added to F (Theorem A.1.2), and we bound in Section A.2 the number of contact arcs postulated by OSF-Builder when the allele tree (Theorem A.2.2) and the species forest (Theorem A.2.3) are slightly modified. We present in Section A.3 a method to count the number of possible outputs for the Fitch algorithm, described in [4], which we then slightly modify in order to get the number of distinct OSFs that may be obtained from a given input. In Section A.4, mimicking the definition of a tree-based network given in [26] (see Section 1.1.3), we introduce the class of forest-based network, and we provide a characterization of forest-based networks that can be obtained as an output of OSF-Builder (Theorem A.4.2).

A.1 Optimality

Suppose that X is a finite non-empty set and that T is a phylogenetic tree on X. Suppose that $u, v \in V(T)$. Then we put $u \preceq_T v$ if u is an ancestor of v in T. In case there is no ambiguity as to the tree we are referring to or that tree is of no relevance to the discussion, we also write $u \preceq v$ rather than $u \preceq_T v$.

For the following, suppose that $\mathcal{F}$ is an AS-forest with underlying species forest

F, allele tree G, and allele-species map ϕ. Suppose that $v \in V(G)$. Then we associate to v the set

$$\Lambda(v) = \{\phi(g) : g \in C_G(v)\}$$

of all species in F that carry an offspring allele of v, and the set

$$F(v) = \{S \in F | L(S) \cap \Lambda(v) \neq \emptyset\}$$

of all trees in F which contain a leaf (i. e. a species) that carries an offspring allele of v. Note that both $\Lambda(v) \neq \emptyset$ and $F(v) \neq \emptyset$ hold.

Our first result (Lemma A.1.1) ensures that for any ordering κ of the trees in F, the map $\psi^{\kappa}_{\mathcal{F}} : V(G) \to V(F)$ defined in Section 5.3.2 for an AS-forest $\mathcal{F}' = (F', G', \phi')$ is indeed an OSF for $\mathcal{F}$. We denote the powerset of a set Y by $\mathcal{P}(Y)$ and refer to the map $\sigma : V(G) \to \mathcal{P}(F)$ underpinning $\psi^*_{\mathcal{F}}$ as *label-set map for* $\psi^*_{\mathcal{F}}$.

Lemma A.1.1. *For any AS-forest $\mathcal{F}$ the map $\psi^*_{\mathcal{F}}$ is an OSF.*

Proof. Suppose $\mathcal{F} = (F, G, \phi)$ and put $\psi = \psi_{\mathcal{F}}$. We need to show that ψ satisfies Properties (F1) – (F3). To see Property (F1), suppose $x \in L(G)$. Choose P_x to be the unique tree in F that contains $\phi(x)$ in its leaf set. Then $\Lambda(x) = \{\phi(x)\}$. Then $\psi(x) = lca_{P_x}(L(P_x) \cap \Lambda(x)) = lca_{P_x}(\{\phi(x)\}) = \phi(x)$. Consequently, $\psi|_{L(G)} = \phi$.

To see Property (F2), suppose $u, v \in V(G)$ such that $P_u = P_v$ and $u \preceq_G v$. Then $L_G(v) \subseteq L_G(u)$ and, so, $\Lambda(v) \subseteq \Lambda(u)$. Hence, $\psi(u) = lca_{P_u}(L(P_u) \cap \Lambda(u)) \preceq lca_{P_v}(L(P_v) \cap \Lambda(v)) = \psi(v)$. Finally, Property (F3) is an immediate consequence of the definition of ψ and the set $\Lambda(v)$ where $v \in V(G)$. □

The definition of $\psi^*_{\mathcal{F}}$ combined with Lemma A.1.1 implies the following optimality result for $\psi^*_{\mathcal{F}}$.

Theorem A.1.2. *For any ordered AS-forest $\mathcal{F}$, the map $\psi^*_{\mathcal{F}}$ is an optimal OSF for $\mathcal{F}$.*

We also remark that the inner workings of the Fitch-Hartigan algorithm imply that the OSF returned by OSF-Builder when given F, G, and ϕ always also minimizes the number of contact arcs on any subtree G' of G. Put differently, for any ordering κ of the trees in F, the OSF $\psi^{\kappa}_{\mathcal{F}}$ restricted to the vertex set of G' is an OSF for $\mathcal{F}' = (F, G', \phi)$. However, it should be noted that that OSF need not necessarily be of the form $\psi^*_{\mathcal{F}'}$.

This property is not true in general, as can be seen on Figure 5.7. If we look at the subtree G' of G rooted at w_0, we can see that the number of contacts arcs contained in G' is one for the first OSF, and two in the second one. This is consistent with our previous observation that the second OSF is not of the form

$\psi^*_{\mathcal{F}}$, whereas the fist one is. Note that due to the fact that OSF-Builder uses a slightly modified version of the Fitch-Hartigan algorithm, the converse does not hold in general. This means that there may exist minimal OSFs for an AS-forest $\mathcal{F} = (F, G, \phi)$ that are not on the form $\psi^*_{\mathcal{F}}$ but still satisfy the property that the number of contact arcs in any subtree G' of G is minimal.

A.2 Stability

To perform our theoretical noise analysis, we require further terminology. Suppose T is a phylogenetic tree on X. Then T is called a *caterpillar tree* if every interior vertex is adjacent to either one or two leaves. For the following, assume that $\mathcal{F} = (F, G, \phi)$ is an ordered, binary AS-forest, and that G is binary. Also, we associate to $\psi^*_{\mathcal{F}}$ and a directed path P from ρ_G to a further non-leaf vertex v of G a new AS-forest called a *P-induced AS-forest* $\mathcal{F}_P = (F_P, G_P, \phi_P)$ which we define as follows. Let $v_1, v_2 \in V(G)$ denote the two children of v. Then G_P is the caterpillar tree obtained from G by collapsing (i) for every vertex $w \in V(P) - \{v\}$ the subtree of G rooted at the child of w not crossed by P into a new leaf and (ii) the subtrees rooted at v_1 and v_2 into a new leaf. Let F_P denote the forest obtained from F by collapsing all subtrees T of trees in F with $\rho_T \in U := \{u \in V(F) :$ there exists $w \in L(G_P)$ such that $u = \psi^*_{\mathcal{F}}(w)\}$ into a new leaf. Note that $L_P := L(F) - \bigcup_{u \in U} C(u) \cup U$. Also, note that $\mathcal{F}_P$ might contain trees with one or two leaves. Abusing terminology, we also refer to such trees as phylogenetic trees for the remainder of this section. Let $\phi_P : L(G_P) \to L_P$ denote the map that assigns to every leaf $y \in L(G_P)$ a leaf $w \in L(F_P)$ if $w = \psi^*_{\mathcal{F}}(y)$ holds and $\phi(y)$ otherwise. For example, if P is the path from the root of G to w_0 in the OSF depicted in Figure 5.7(i), we represent the P-induced AS-forest in Figure A.1.

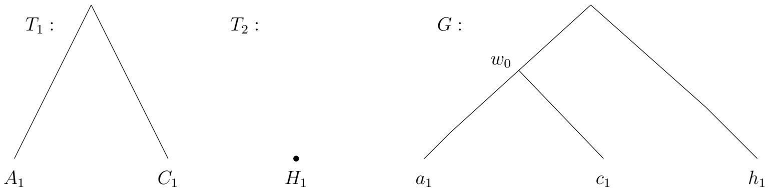


Figure A.1: The P-induced AS-forest for ψ the OSF depicted in Figure 5.4 *(i)* and P the path from the root of G to w_0.

The SPR-operation introduced in Section 5.5.1 naturally lends itself as a tool for comparing two phylogenetic trees T and T' on X by computing the minimal number of SPR-operations that have to be carried out to transform T into T'. That number is commonly referred to as the *SPR-distance* between T and T', denoted by $d_{SPR}(T,T')$. Note that the property that T' can be obtained from T via a sequence $k \geq 2$ random SPR-operations, which we used in the simulation study described in Section 5.5.1, does not mean that $d_{SPR}(T,T') = k$, but only that $d_{SPR}(T,T') \leq k$. Indeed, a shorter sequence of SPR-operations allowing to obtain T' from T may exist.

Lemma A.2.1. *Suppose $\mathcal{F} = (F, G, \phi)$ is an ordered, binary AS-forest and G_0 is a subtree of G rooted at a vertex $v_1 \in V(G)$ that is neither ρ_G nor a child of ρ_G. Suppose further that G' is a phylogenetic tree on $X - L(G_0)$ obtained from G by first deleting G_0 and the incoming arc of its root and then suppressing the resulting degree two vertex. Then both $\mathcal{F}_0 = (F, G_0, \phi_0 := \phi|_{L(G_0)})$ and $\mathcal{F}' = (F, G', \phi' := \phi|_{L(G')})$ are ordered binary AS-forests and*

$$t(\mathcal{F}_0) \leq t(\mathcal{F}) - t(\mathcal{F}') \leq t(\mathcal{F}_0) + 1, \tag{A.1}$$

Proof. As $\mathcal{F}$ is an ordered, binary AS-forest it is immediately clear that $\mathcal{F}_0$ and $\mathcal{F}'$ are also ordered, binary AS-forests.

To see that Inequality (A.1) holds, let $v \in V(G)$ denote the parent of v_1. Let κ denote the ordering of F. Then since $v \neq \rho_G$ there must exist a vertex $u \in V(G)$ that is the parent of v. Let $v_2 \in V(G)$ denote the other child of v and let u_1 denote the other child of u. Put $\psi := \psi_{\mathcal{F}}^{\kappa}$ and let $\sigma : V(G) \to \mathcal{P}(F)$ denote the label-set map underlying ψ. Also, put $\psi' = \psi_{\mathcal{F}'}^{\kappa}$. Let $\sigma' : V(G') \to \mathcal{P}(F)$ denote the label-set map underlying ψ'. Let P denote the directed path in G from ρ_G to v. Clearly, $\sigma'(w) = \sigma(w)$ if $w \notin V(P)$ and $\psi_0 := \psi_{\mathcal{F}_0}^{*} = \psi|_{V(G_0)}$. Then, it suffices to show that

$$0 \leq t(\mathcal{F}_P) - t(\mathcal{F}'_P) \leq 1, \tag{A.2}$$

where $\mathcal{F}_P$ and $\mathcal{F}'_P$ denote the P-induced ordered, binary AS-forests for $\mathcal{F}$ and $\mathcal{F}'$, respectively, defined above.

To establish Inequality (A.2), we perform induction on the number $k \geq 2$ of arcs on P. If $k = 2$ then G_P is the caterpillar tree on $\{u_1, v_1, v_2\}$ and G'_P is the phylogenetic tree on $\{u_1, v_2\}$. Clearly, $t_{\mathcal{F}'_P} \leq 1$. If $t_{\mathcal{F}'_P} = 1$ then $t_{\mathcal{F}_P} \leq 2$ and if $t_{\mathcal{F}'_P} = 0$ then $t_{\mathcal{F}_P} \leq 1$. Hence, Inequality (A.2) holds.

Suppose for the remainder that $3 \leq k$ and assume that Inequality (A.2) holds whenever P has $2 \leq k' \leq k-1$ arcs. Let $w \in V(G_P)$ denote the non-leaf child of $\rho_P := \rho_{G_P}$ and let $w' \in V(G_P)$ denote the leaf of G_P adjacent with w. Also, let $x \in V(P)$ denote the leaf of G_P adjacent with its root ρ_{G_P}. Then $w, x \in G'_P$ and $\sigma(x) = \sigma'(x)$. Since G_P is binary and a caterpillar tree, we have for all $z \in V(P)$ that $\sigma'(\rho_P) \subseteq \sigma(u)$. Combined with the fact that $\sigma'(\rho_P) \subseteq \sigma'(w)$ and that $\sigma(\rho_P) \neq \emptyset$, it follows that there exists a tree $T \in \sigma(w') = \sigma'(w')$ such that $T \in \sigma(\rho_{G_P}) \cap \sigma'(\rho_{G_P})$. Consequently, (ρ_P, w) is a contact arc for ψ if and only if it $(\rho_{G'_P}, w)$ is a contact arc for ψ'. Combined with the induction hypothesis applied to the P-induced ordered, binary AS-forests obtained from $\mathcal{F}$ and $\mathcal{F}'$ by removing x and ρ_P and the arcs (ρ_P, x) and (ρ, w) in case of G_P and $\rho_{G'_P}$ and the arcs $(\rho_{G'_P}, x)$ and $(\rho_{G'_P}, w)$ in case of G'_P, Inequality (A.2) follows. This concludes the induction step and thus the proof of the lemma. □

Theorem A.2.2. *Suppose $\mathcal{F} = (F, G, \phi)$ and $\mathcal{F}' = (F, G', \phi)$ are two ordered binary AS-forests. Then, $0 \leq |t(\mathcal{F}) - t(\mathcal{F}')| \leq d_{SPR}(G, G')$.*

Proof. Clearly, the stated lower bound holds. To see the stated upper bounds, it suffices to show that if $d_{SPR}(G, G') = 1$ then $t(\mathcal{F}) - t(\mathcal{F}') \leq 1$. Suppose $d_{SPR}(G, G') = 1$. Let G_0 denote the subtree of G to whose root ρ_G an SPR-operation on G is applied. Clearly, the tree G_1 obtained from G' in the first step of the SPR operation on G is a phylogenetic tree on $X - L(G_0)$. Put $\mathcal{F}_0 = (F, G_0, \phi|_{L(G_0)})$ and $\mathcal{F}_1 = (F, G_1, \phi|_{L(G_1)})$. By Lemma A.2.1, it follows that

$$t(\mathcal{F}_0) \leq t(\mathcal{F}) - t(\mathcal{F}_1) \leq t(\mathcal{F}_0) + 1$$

and

$$t(\mathcal{F}_0) \leq t(\mathcal{F}') - t(\mathcal{F}_1) \leq t(\mathcal{F}_0) + 1.$$

Taken in combination, $|t(\mathcal{F}) - t(\mathcal{F}')| \leq 1$ follows. □

As it turns out, both bounds stated in Theorem A.2.2 are sharp. Continuing with the notation from Theorem A.2.2, an example for the stated lower bound is provided by the subtree G_0 containing all contact arcs for $\psi^*_{\mathcal{F}}$ and $\psi^*_{\mathcal{F}'}$. An example for the stated upper bound is furnished by the AS-forests $\mathcal{F}$ and $\mathcal{F}'$ whose unique representations are depicted in Figure 5.4(i) and Figure A.2, respectively as $|t(\mathcal{F}) - t(\mathcal{F}')| = 1 = d_{SPR}(G, G')$. It is worth noting however that Theorem A.2.2 does not imply that $\psi^*_{\mathcal{F}}$ and $\psi^*_{\mathcal{F}'}$ must have contact arcs in common.

We continue this section by turning our attention to the question of measuring the difference between representations of OSFs in terms of the difference between their underlying species forests. To this end, note first that an SPR-operation affecting a single tree T in the underlying species forest of an AS-forest $\mathcal{F}$ does not influence the minimum number of contact arcs of an OSF for $\mathcal{F}$. Thus, we turn our attention to species forests F where two distinct trees of F are affected by an SPR-operation. We again require further terminology.

Suppose $\mathcal{F} = (F, G, \phi)$ is an AS-forest $\mathcal{F}$ and T_0 is a subtree of some tree in F. Then we put $\gamma_{\mathcal{F}}(T_0) := \{g \in L(G) : \phi(g) \in L(T_0)\}$. For example, for the AS-forest $\mathcal{F}$ depicted in Figure 5.4(i) and the subtree T_0 of T_1 rooted at the parent of C, we have $\gamma_{\mathcal{F}}(T_0) = \{c_1, d_1\}$.

Theorem A.2.3. *Suppose $\mathcal{F} = (F, G, \phi)$ is an ordered, binary AS-forest and $T_1 \in F$. Assume that T_2 is the phylogenetic tree obtained from T_1 by performing an SPR operation on T_1 and the root ρ_{T_0} of a subtree T_0 of T_1 where $\rho_{T_0} \neq \rho_{T_1}$ Denoting the resulting ordered binary AS-forest by $\mathcal{F}'$ we have $0 \leq |t(\mathcal{F}') - t(\mathcal{F})| \leq |\gamma_{\mathcal{F}}(T_0)|$.*

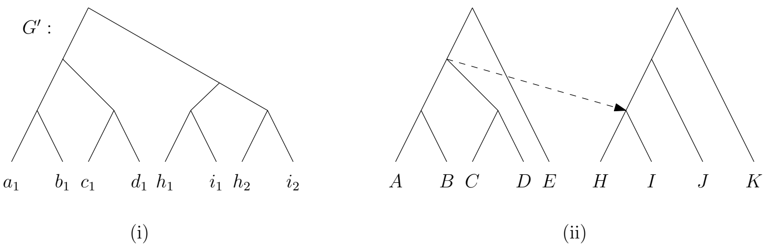


Figure A.2: (i) A phylogenetic tree G' obtained from the phylogenetic tree G in Figure 5.4 by applying one SPR operation to G and the parent of c_1 and d_1. (ii) A representation for $\psi^*_{\mathcal{F}}$ where $\{T_1, T_2\}$, G', ϕ are the ordered species forest, allele tree, and allele-species map underlying $\mathcal{F}$, respectively. For $\mathcal{F}'$ the ordered binary AS-forest depicted in Figure 5.4(i), we have $t(\mathcal{F}') - t(\mathcal{F}) = 1$.

Proof. Clearly, the stated lower bound holds. To see the stated upper bound, suppose $\mathcal{F} = (F, G, \phi)$, $\mathcal{F}' = (F', G, \phi)$ and T_0 are as in the statement of the proposition. Note that $L(F) = L(F')$. We perform induction on $n := |\gamma_{\mathcal{F}}(T_0)| \geq 0$. If $n = 0$, then the proposition clearly holds as no contact arc of an OSF for $\mathcal{F}$ shares a vertex with T_0.

So suppose $|\gamma_{\mathcal{F}}(T_0)| = n + 1$ and assume that the proposition holds for all $\overline{\mathcal{F}}$, $\overline{\mathcal{F}'}$ and $\overline{T_0}$ for which $|\gamma_{\overline{\mathcal{F}}}(\overline{T_0})| \leq n$ where $\overline{\mathcal{F}}$, $\overline{\mathcal{F}'}$ and $\overline{T_0}$ are as their canonical namesakes in the statement of the proposition. Let $g \in \gamma_{\mathcal{F}}(T_0)$ and let G' denote the phylogenetic tree obtained from G by deleting g and its incoming arc (suppressing the resulting degree two vertex). Put $\phi' := \phi|_{L(G')}$, $\mathcal{F}_1 = (F, G', \phi')$, and $\mathcal{F}_2 = (F', G', \phi)$. Then

$$\begin{aligned} t(\mathcal{F}) - t(\mathcal{F}') = \quad & t(\mathcal{F}) - t(\mathcal{F}_1) \\ & + t(\mathcal{F}_1) - t(\mathcal{F}_2) \\ & + t(\mathcal{F}_2) - t(\mathcal{F}'). \end{aligned}$$

Since $|\gamma_{\mathcal{F}'}(T_0)| = n$ clearly holds, it follows by induction hypothesis that $-n \leq t(\mathcal{F}_1) - t(\mathcal{F}_2) \leq n$. Moreover, Lemma A.2.1 implies that $0 \leq t(\mathcal{F}) - t(\mathcal{F}_1) \leq 1$ and $-1 \leq t(\mathcal{F}_2) - t(\mathcal{F}') \leq 0$. In summary, we obtain $-n - 1 \leq t(\mathcal{F}) - t(\mathcal{F}') \leq n + 1$. Hence, $|t(\mathcal{F}) - t(\mathcal{F}')| \leq n - 1$ which concludes the proof of the induction step and, thus, the proof of the proposition. □

As in the case of Theorem A.2.2, the bounds stated in Theorem A.2.3 are sharp. For the lower bound this follows from the base case of the induction underlying the proof of Theorem A.2.3. For the upper bound an example is furnished by the AS-forest depicted in Figure A.3.

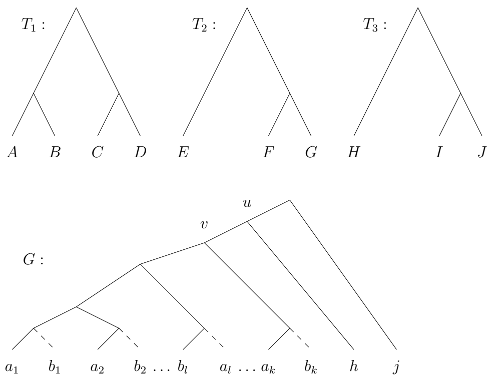


Figure A.3: An ordered, binary AS-Forest $\mathcal{F} = (\{T_1, T_2, T_3\}, G, \phi)$ such that $\psi^*_{\mathcal{F}}$ has precisely one contact arc i. e. the arc (u, v). Application of an SPR operation to T_1 and leaf B results in an ordered, binary AS-forest $\mathcal{F}'$ such that the contact arcs of $\psi^*_{\mathcal{F}'}$ are (u, v) plus the k dashed arcs of G.

A.3 Number of optimal solutions

In [4], the Fitch-Hartigan algorithm is used as a tool towards an explanation of the difference in size of the shells of some species of turtles. The authors propose a method to count the number of optimal extensions $\overline{\chi}$ of χ for a given initial condition (T, χ), which they call the *Enumeration Recursion Formula.* The idea is to associate to each vertex v of T two maps $S_v : C \to \mathbb{N}$ and $T_v : C \to \mathbb{N}$. For $c \in C$, $S_v(c)$ represents the minimal number of jumps in the subtree rooted at v that can be obtained by labelling v with $c \in \mathcal{C}$, and $T_v(c)$ represents the number of possible labellings of the vertices below v that produce this minimal number of jumps. For ρ the root of T, the number of possible optimal extensions $\overline{\chi}$ of χ is then given by:

$$\sum_{\substack{p \in \operatorname{argmin}\{S_\rho(c)\} \\ c \in \mathcal{C}}} T_\rho(p).$$

The maps S_v and T_v are recursively defined as follows. If v is a leaf of T, we put for $c \in C$:

$$S_v(c) = \begin{cases} 0 & \text{if } \chi(v) = c \\ \infty & \text{otherwise.} \end{cases}$$

and

$$T_v(c) = \begin{cases} 1 & \text{if } \chi(v) = c \\ 0 & \text{otherwise.} \end{cases}$$

If v is a non leaf vertex with children $\{v_1, \ldots, v_k\}$, $k \geq 2$, we put for $c \in C$:

$$S_v(c) = \sum_{i=1}^{k} \min_{p \in C} \{S_{v_i}(p) + \mathbb{I}(p \neq c)\}$$

and

$$T_v(c) = \prod_{i=1}^{k} \sum_{p \in \kappa_c^i} T_{v_i}(p)$$

where

$$\kappa_c^i = \operatorname*{argmin}_{p \in C} \{S_{v_i}(p) + \mathbb{I}(p \neq c)\}.$$

This method can be used to count the number of optimal OSFs for a given AS-forest $\mathcal{F} = \{F, G, \phi\}$. However, it takes into account a situation that, translated to the OSF framework, is forbidden by one of the properties defining an OSF. More precisely, consider the labelled tree in Figure 5.5(iii), where the symbols $\times$, $\bullet$ and $\circ$ are thought of as distinct trees in a species forest. That tree contains an internal vertex mapped to the tree $\times$, whereas none of the leaves below that vertex is mapped to $\times$. This is in contradiction with Property (F3), thus that mapping, although optimal in the number of jumps, does not correspond to an OSF.

To deal with this problem, it suffices to slightly modify the definition of the maps T_v. For v an internal vertex of T and $c \in C$, we define $T_v(c)$ using the formula above only if there exists a leaf $x \in C(v)$ such that $\phi(x) = c$. Otherwise, we put $T_v(c) = 0$. Roughly speaking, this means that the formula does not record labellings such that a vertex v is labelled with an element $c \in C$ whereas no leaf below v is labelled with c.

Note that in the context of OSFs, using this enumeration method on an AS-forest $\mathcal{F} = \{F, G, \phi\}$ only allows to count the different possible affectations of vertices of G to trees of F. However, once for all vertex v of G, the tree $T \in F$ to which v is mapped via an OSF is known, there may be different vertices in T to which v may be mapped. Thus, this method does not count the number of possible OSFs in the strict sense, but only the number of distinct associations vertices of G-trees of F.

A.4 Representing OSFs in terms of graphs

The nature of an OSF ψ for an AS-forest $\mathcal{F}$ naturally lends itself to a representation in terms of a graph obtained via adding contact arcs for ψ to the arc set of the species forest underpinning $\mathcal{F}$. To make this more precise, we require more terminology. Suppose N is a positive integer. Then we call a graph N a *m-rooted* *network (on X)* if $L(N) = X$ and N has exactly m *roots*, that is, vertices of in-degree 0. If the number m of roots is of no relevance, we simply refer to N as a *multiple rooted netork*. Clearly, a phylogenetic tree is a special case of a 1-rooted network.

Suppose $\mathcal{F} = (F, G, \phi)$ is an AS-forest and ψ is an OSF for $\mathcal{F}$. Clearly, we can refer to such a map ψ as an OSF without having to precise the AS-forest it is associated to, as all informations about the AS-forest is contained in ψ. Consider the multiple rooted network N obtained graph from F by carrying the following two steps:

(i) For all contact arcs (u, v) of G, add to F the arc $(\psi(u), \psi(v))$.

(ii) For all elements x of $L(F)$ labelling a non-leaf vertex of N, rename x as u_x and add to N the arc (u_x, x).

The first step aims at representing contact arcs for ψ in N. By abuse of terminology, we also refer to these arcs as *contact arc* of N. The second step ensures that the leaf sets of F and N are the same, which may not be the case after step (i) if some contact arcs of N are of the form (u, x) or (x, u), where u is a vertex of F and x a leaf of F.

Although each root of N is the root of a given tree in F, the number m of roots of N is not necessarily equal to the size of the forest $|F|$. Indeed, some contact arcs of N may be of the form (u, r), where u is a vertex of F and r the root of a tree in F. In that case, r is not a root of N, as it has an incoming arc. We then have $\{r \in V(F) : r \text{ is a root in } F\} \subseteq \{r \in V(N) : r \text{ is a root in } N\}$.

Note that the network N obtained by applying (i) and (ii) successively is not binary. It can be made binary by applying the following steps for all contact arcs (u, v) of N : First, we subdivide the incoming arcs of u and v that is not a contact arc (if such an arc exists) by introduction of the vertices u_0 and v_0 respectively. Then, we replace in N the arc (u, v) by the arc (u_0, v_0). We call the network N' obtained this way a *binary representation* of the *OSF* (F, G, ψ). The important information provided by a contact arc (u, v) of N, that are, the sets $C(u)$ and $C(v)$ are clearly not affected by this transformation. However, such a representation may not be unique.

Clearly, any OSF ψ has a unique representation. However, it should be noted that a multiple rooted network can be a representation for more than one OSF ψ even if the AS-forest underpinning $\mathcal{F}$ is the same in both pairs. An example for this is furnished in terms of the AS-forest $\mathcal{F} = (F, G, \phi)$ depicted in Figure A.4. With putting $\rho_i = \rho_{T_i}$ for all $i = 1, 2$ the maps $\psi, \psi' : V(G) \to V(F)$ which both assign to every vertex in $u \in V(G) - \{v, w\}$ the indicated label in $X \cup \{\rho_1, \rho_2, \rho_3\}$ and differ in v and w in that $\psi(v) = \rho_2$, $\psi(w) = \rho_1$, $\psi'(v) = \rho_1$, and $\psi(w) = \rho_2$

are clearly minimum OSFs for $\mathcal{F}$. However ψ and ψ' have the same representation as the contact arcs for both ψ and ψ' are (ρ_3, ρ_1), (ρ_1, ρ_2), (ρ_3, ρ_2) and (ρ_2, ρ_1).

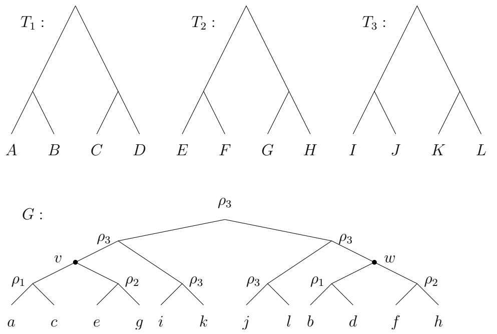


Figure A.4: An AS-forest $\mathcal{F}$ consisting of the depicted species forest $F = \{T_1, T_2, T_3\}$ on $\{A, B, \ldots, L\}$, allele tree G on $= \{a, b, \ldots, l\}$ and allele-species map ϕ. The representation of ψ and ψ' are the same where the OSFs $\psi, \psi' : V(G) \to V(F)$ are as indicated. See text for details.

Although some multiple rooted networks are representations of OSFs it is clearly too much to hope for that every such network enjoys that property. We next turn our attention to characterizing multiple rooted network for which this is the case. We start with introducing further terminology.

Suppose N is a multiple rooted network with leaf set X. Then we denote by $E_t = E_{t,N}$ a non-empty set of arcs of N whose deletion results in a forest $F_t = F_{E_t}$ of N such that (after suppressing any vertices with indegree and outdegree one that might result) (i) every tree $T \in F_t$ is a phylogenetic tree with $L(T) \subseteq X$ and (ii) $E_t = \{(u, v) \in E(N)$: there exist $T, T' \in F_t$ distinct such that $u \in V(T)$ and $v \in V(T')\}$. Clearly, for any allele tree G for which there exists such a set $E_t \subseteq A(N)$ and an allele-species map $\phi : L(G) \to L(F_t)$ the triple (F_t, G, ϕ) is an AS-forest.

As is easy to check, a representation N of an OSF ψ with contact arcs E_t contains a vertex $r_0 \in V(N)$ such that the following property is satisfied:

(*) every arc $a = (u, v) \in E_t$ can be reached from r_0 via a directed path P_a and there exists no arc $(u', v') \in E_t - a$ on P_a such that v and u' belong to the same tree $T \in F$ and $C_T(v) \cap C_T(u') = \emptyset$.

Indeed, the vertex $r_0 = \psi(\rho_G)$, where $_{r}ho_G$ is the root of the allele tree G, satisfies (*). In view of this, we say for a multiple rooted network N that some non-empty set $E_t \subseteq E(N)$ is *valid* if there exists some $r_0 \in V(N)$ such that Property (*) is satisfied for E_t.

The next result lies at the heart of the proof of Theorem A.4.2 which provides structural insight into a representation of an OSF. To help establish it, we require again more terminology. Suppose G_1 and G_2 are two phylogenetic trees with $V(G_1) \cap V(G_2) = \emptyset$. Then we denote by $G_1 \wedge G_2$ the phylogenetic tree on $L(G_1) \cup L(G_2)$ obtained from G_1 and G_2 by adding a new vertex ρ to $V(G_1) \cup V(G_2)$ and two new arcs (ρ, ρ_{G_1}) and (ρ, ρ_{G_2}) to $E(G_1) \cup E(G_2)$.

Proposition A.4.1. *Suppose N is a multiple rooted network on X and $E_t = E_{N,t} \neq \emptyset$ is valid for N. Then there exist an AS-forest $\mathcal{F}_t$ and an OSF ψ_t for $\mathcal{F}_t$ such that N is a representation for ψ_t. In particular, there exists an order κ on $\mathcal{F}_t$ such that $\psi_t = \psi^{\kappa}_{\mathcal{F}_t}$ holds.*

Proof. It suffices to show that there exists an ordered AS-forest $\mathcal{F}_t$ such that N is a representation for $\psi^{*}_{\mathcal{F}_t}$. We use induction on $n := |E_t| \geq 1$. Put $F = F_t = F_{E_t}$. Let $r_0 \in V(M)$ such that Property (*) is satisfied for E_t.

Suppose $n = 1$. Then there exist trees $T, T' \in F$ distinct and a unique arc $a \in E_t$ such that $u := tail(a) \in V(T)$ and $v := head(a) \in V(T')$. Let G_u and G_v denote two phylogenetic trees such that G_u and T_u are isomorphic and G_v and T'_v are isomorphic. Let $\phi_u : L(G_u) \to L(T_u)$ and $\phi_v : L(G_v) \to L(T_v)$ denote the maps induced by the underlying bijections. Clearly, $G_0 := G_u \wedge G_v$ is a phylogenetic tree on $X' := L(G_u) \cup L(G_v)$. Consider the map $\phi : L(G_0) \to L(F)$ given by $\phi(x) = \phi_u(x)$ if $x \in V(G_u)$ and $\phi(x) = \phi_v(x)$ if $x \in V(G_v)$. Let κ denote some ordering of F such that, when starting at the minimal element of κ, we encounter T' before T'. Clearly, $\mathcal{F} := (F, G_0, \phi)$ is an ordered AS-forest. Also, it is easy to check that $\psi^{\kappa}_{\mathcal{F}}$ maps all vertices of G_u to vertices of T, all vertices of G_v to vertices of T', and the root of G_0 to u. Furthermore, the only contact arc for $\psi^{\kappa}_{\mathcal{F}}$ is a. That N is a representation for $\psi^{\kappa}_{\mathcal{F}}$ is straightforward to see.

Now, suppose $n > 1$ and assume that the stated characterization holds true for all multiple rooted networks N' where all non-empty subsets $E'_t = E_{t,N'} \subseteq E(N)$ of size $n-1$ that are valid for N'. Let $a \in E_t$ denote an arc of N such that no arc in $E_t - a$ can be reached from $u := tail(a)$ via a directed path. Out of the connected components of N obtained by removing a, let N' denote the one which contains u in its vertex set. Note that N' is clearly a k-network with $k \in \{m, m-1\}$ and that $E_t - a \subsetneq E(N')$. Putting $E'_t = E_{t,N'} := E_t - a$ and $F'_t := F_{E'_t}$ it follows that $|F'_t| = |F_t| - 1$ if a is a cut-arc of N and that $F'_t = F_t$ if it is not.

Since N enjoys Property (*) for r_0 and $r_0 \not\in V(T')$ it follows that E'_t is valid for N'. Since $|E'_t| = n - 1$ the induction hypothesis implies that there must exist an ordering κ' of the trees in F'_t, a phylogenetic tree G' and an allele-species map $\phi' : L(G') \to L(F'_t)$ such that N' is a representation for $\psi^{*}_{\mathcal{F}'}$ where $\mathcal{F}' = (F'_t, G', \phi')$. Put $\rho' = \rho_{G'}$ and let $\sigma' := \sigma_{\kappa'} : V(G') \to (F'_t)$ denote the label-set map for $\psi' := \psi^{*}_{\mathcal{F}'}$.

To obtain $\mathcal{F}_t$, we first construct the allele tree G_t underlying $\mathcal{F}_t$. Let $T, T' \in F_t$ such that $u \in V(T)$ and $v := head(a) \in V(T')$. Put $F_0 = \{T, T'\}$. As in the base case of the induction, let G_u and G_v denote two phylogenetic trees such that T_u and G_u are isomorphic and T'_v and G_v are isomorphic. Again, put $G_0 := G_u \wedge G_v$ and $\rho_0 = \rho_{G_0}$. Denote by $\phi_0 : L(G_0) \to L(F_0)$ the map induced by the bijections $\chi_u : V(T_u) \to V(G_u)$ and $\chi_v : V(T'_v) \to V(G_v)$ underlying the isomorphisms between G_u and T_u and between G_v and T'_v, respectively. Let P denote a directed path from $\psi'(\rho')$ to u in N'. We distinguish between the cases that (α) P contains an arc $a' = (u', v') \in E'_t$, and (β) that P does not contain such an arc.

Case (α): Let P' denote the directed path from ρ' to $\psi'^{-1}(u)$ in G' that corresponds to P. Let $a_t = (u_t, v_t)$ denote the arc in G' for which $\psi'(u_t) = u'$ and $\psi'(v_t) = v'$ holds. Let G_t denote the tree obtained from G' and G_0 by first subdividing a_t by adding a new vertex w_0 and then adding the arc (w_0, ρ_0). Put $\rho_t = \rho_{G_t}$. Note that G_t is clearly a phylogenetic tree on $L(G') \cup L(G_0)$.

We next define an ordering κ for F_t by putting $\kappa = \kappa'$ if $F_t = F'_t$. If $F_t \neq F'_t$ then we define κ as the ordering obtained from κ' by adding T' as new maximal element to κ'. We first show that κ is well-defined. For this, it suffices to consider the case that $F_t \neq F'_t$. So assume that $F_t \neq F'_t$. Then $T' \notin F_t$. Employing a straightforward minimality argument, we have $\sigma(z) = \sigma'(z)$, for all $z \in V(G') - V(P')$.

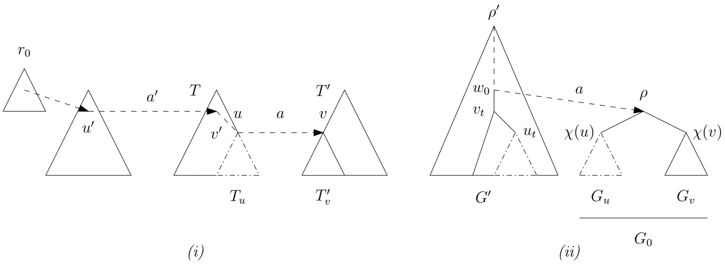


Figure A.5: (i) The situation in N in case (α). (ii) The construction of G_t (see text for details).

Since $\rho_u := \rho_{G_u}$ and $\rho_v := \rho_{G_v}$ are the only two children of ρ' and G_u and G_v are isomorphic with subtrees of T and T', respectively, it follows that $\sigma(\rho_0) = \{T, T'\}$. Since $\psi'(v_t) = v' \in V(T)$, we also have $T \in \sigma(v_t)$. Thus, w_0 cannot be the tail of a contact arc of $\psi_t := \psi^*_{\mathcal{F}_t}$. Since v_t and ρ_0 are the only two children of w_0 we either have $\sigma(w_0) = \{T\}$ or $\sigma(w_0) = \{T, T'\}$. In the first case, $P^\kappa_v = P^\kappa_{v_t} = P^\kappa_\rho = T$ follows. The second case implies $\{T, T'\} \subseteq \sigma(v_t) = \sigma'(v_t)$. Since $P^\kappa_{v_t} = T'$ and (u_t, v_t) is a contact arc for ψ', it follows that $P^\kappa_{u_t} \notin \sigma(v_t)$. Consequently, κ is well-defined, as required. Note that similar arguments also imply in the case that $F_t = F'_t$ that, when starting at the minimal element of κ, we first encounter T and then T'.

We claim that $\psi_t|_{V(G_t)} = \psi'$. Suppose $z \in V(G_t)$. If $z \in V(G')$ but not an ancestor of w_0 then $\sigma(z) = \sigma'(z)$. By the definition of κ, it follows that $\psi_t(z) = \psi'(z)$. If $z \in V(P') - w_0$ and an ancestor of w_0 then the definition of a label-set map implies $\sigma(w) \subseteq \sigma(v_t)$. Since a is a contact arc of ψ_t and $T' \notin \{T\} = \sigma(v_t)$, we obtain $\psi_t(z) = \psi'(z)$ by the definition of κ. This completes the proof of the claim.

We next analyze $\psi_t(z)$ where $z \in V(G_0) \cup \{w_0\}$. Note first that employing arguments similar to the ones used in the former of the two previous claims, we also obtain $P^{\kappa}_{w_0} = T$. In turn, this implies that $\psi_t(w_0) = v'$. Moreover the choice of a implies $\sigma(\chi_u(u)) \cap \sigma(\chi_v(v)) = \emptyset$. Hence $\sigma(\rho_0) = \sigma(\chi_u(u)) \cup \sigma(\chi_v(v))$. Combined with the definition of κ, it follows that $\psi_t(\chi_u(u)) = \psi_t(\rho_0) = u$. Furthermore, since for all $y \in V(T_u)$, we have $\sigma(y) = \sigma'(\chi_u(y))$ it follows that $\psi_t(\chi_u(y)) = \psi'(\chi(y))$, for all such y. Finally since $\sigma(\chi_v(y)) = \{T'\}$ for all $y \in V(T_v)$ it follows that $\psi_t(\chi_v(y)) = y$.

Combined with the previous claim, it follows that the set of contact arcs for ψ_t equals the set of contact arcs for ψ' augmented by $\{a\}$. Since N' is a representation for ψ' it follows that N is a representation for $(\psi_t$. This concludes the proof of this Case (α).

Case (β): Assume that P does not contain an arc in E_t. Then $r_0 \in V(T)$ as $u \in V(T)$. In that case, we put $G_t = G_0 \wedge G'$. Then the map $\phi_t : L(G_t) \to L(F_t)$ defined as in Case (α) is clearly an allele-species map, and $\mathcal{F}_t = (F_t, G_t, \phi_t)$ is an ordered AS-forest with ordering $\kappa := \kappa'$. Again, we put $\psi_t := \psi^*_{\mathcal{F}_t}$.

To see that N is a representation for ψ_t, we first claim that $\psi_t|_{V(G')} = \psi'$. To see the claim note that since G' is the subtree of G_t rooted at ρ', we have for all vertices $w \in V(G')$ that $\sigma(w) = \sigma'(w)$. Therefore, it suffices to show that $\psi_t(\rho') = \psi'(\rho')$.

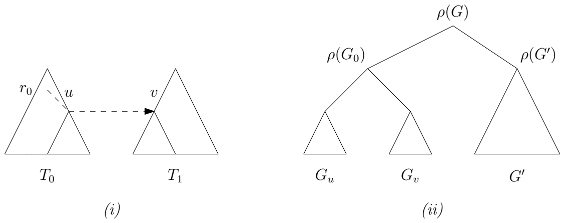


Figure A.6: (i): The situation in N in case (β). (ii): The construction of G_t (see text for details).

To this end, note first that, as in the previous case, $\sigma(\rho_0) = \{T, T'\}$ must hold by construction. Also, note that since $r_0 \in V(T)$ and, as was observed above, r_0 and $\psi'(\rho')$ are vertices of the same tree of F_t, it follows that $\psi'(\rho') \in V(T)$. Consequently, $T \in \sigma(\rho') \cap \sigma(\rho_0)$ and, so, $T \in \sigma(\rho_t)$. Since σ is a label-set map for

$\mathcal{F}_t$ and ρ' and ρ_0 are the only children of ρ, it follows that either $\sigma(\rho_t) = \{T\}$ or $\sigma(\rho_t) = \{T, T'\}$. In the first case, $P^{\kappa}_{\rho_t} = P^{\kappa}_{\rho'} = P^{\kappa}_{\rho_0} = T \in F'_t \subseteq F_t$ follows which implies the claim. In the second case, we have that T is encountered before T' when starting at the minimal element f κ as $\psi'(\rho') \in V(T)$ and $\sigma(\rho_t) \subseteq \sigma(\rho') = \sigma'(\rho')$. Hence, $\{T, T'\} = \sigma(\rho_t) =\subsetneq \sigma(\rho') = \sigma'(\rho')$. Consequently, $P^{\kappa}_{\rho_t} = P^{\kappa}_{\rho'} = P^{\kappa}_{\rho_0} = T_0 \in F'_t \subseteq F_t$ must hold again which implies the claim in this case too. Similar arguments as in Case (α), imply that N is a representation for ψ_t. □

Armed with Proposition A.4.1, we next establish one of our main results.

Theorem A.4.2. *Suppose N is a m-rooted network on X and $E_t \neq \emptyset$. Then E_t is valid for N if and only if there exists an AS-forest $\mathcal{F} = (F_t, G, \phi)$ and an OSF $\psi^*_{\mathcal{F}}$ for $\mathcal{F}$ such that N is a representation of $\psi^*_{\mathcal{F}}$.*

Proof. Assume first that there exists an AS-forest (F_t, G, ϕ) and an OSF ψ for $\mathcal{F}$ such that N is a representation for ψ. Then the set of contact arcs of N is E_t. Also, every contact arc of N can clearly be reached from the vertex $r_0 := \psi(\rho_G)$ via a directed path.

It remains to show that r_0 satisfies the second part of Property (*). Assume that $a \in E_t$ and that $T, T' \in F_t$ distinct such that $u := \mathit{tail}(a) \in V(T)$ and $v := \mathit{head}(a) \in V(T')$. Then, by the definition of an OSF, there exists a directed path P from r_0 to u in N. Assume for contradiction that there exists an arc $a' \in E_t - a$ on P such that with $u' := \mathit{tail}(a')$ and $v' := \mathit{head}(a')$ we have $S(u') = S(v) = T'$ and $\mathcal{C}_{T'}(u') \cap \mathcal{C}_{T'}(v) = \emptyset$. Then there is a directed path in N from u' to v. Putting $u'' = {\psi^*_{\mathcal{F}}}^{-1}(u')$ and $v'' = {\psi^*_{\mathcal{F}}}^{-1}(v)$, it follows that there exists a directed path from u'' to v'' in G. Hence, $\mathcal{C}_G(v'') \subseteq \mathcal{C}_G(u'')$ and, so, $\Lambda(u'') \subseteq \Lambda(v'')$. Since $S(u') = S(v) = T'$ it follows that $\mathcal{C}_{T'}(v) \cap \mathcal{C}_{T'}(u')$ which is impossible.

Conversely, assume that E_t is valid for N. Then Proposition A.4.1 implies that there exists an ordered AS-forest $\mathcal{F} = (F_t, G, \phi)$ such that N is a representation of $\psi^*_{\mathcal{F}}$. □

Appendix B

List of algorithms

Various methods and algorithms are described throughout this thesis. The following provides an exhaustive list of these algorithm, together with some key information and a brief description.

B.1 Reviewed algorithms

These algorithms are presented in this thesis either because they are used in one or more of the original research works, or for the sake of completeness in the literature review. We indicate the pages in which a description can be found.

- UPGMA ([56]): Builds an ultrametric tree $\mathcal{T}$ from a distance d such that the distance induced by $\mathcal{T}$ is d if and only if d is an ultrametric (page 17).

- Neighbor-Joining ([53]): Builds a weighted unrooted tree $\mathcal{T}$ from a distance d such that the distance induced by $\mathcal{T}$ is d if and only if d is tree-like (page 17).

- Meacham's Tree Popping ([50]): Builds an unrooted phylogenetic tree representing a split system Σ is Σ is compatible, or returns the statement that Σ is not compatible (page 20).

- Neighbor-Net ([9]): Builds an outerplanar split-network N from a distance d, such that the distance induced by N is d if and only if d is totally-decomposable and the underlying split system $\mathcal{S}_d$ is circular (page 23).

- Bottom-Up ([35]): Builds a labelled rooted tree $\mathcal{T}$ representing a symbolic distance δ if δ is a symbolic ultrametric, or returns the statement that δ is not a symbolic ultrametric (page 27).

- BUILD ([1]): Builds a rooted tree T displaying a set of triplets C if such a tree exists, or returns the statement that there exist no tree displaying all triplets in C (page 31). Used in Chapters 2 and 3.

- Fitch-Hartigan ([25]): Labels the internal vertices of a rooted tree T with labelled leves, in order to minimize the number of arcs (u, v) of T such that u and v have different labels (page 122).

B.2 New algorithms

These algorithm are the outcome of original research works that I have been carrying on during my PhD. The exact credits can be found in the Introduction of the relevant chapters.

- Find-Cycles ([41], Chapter 2): Sub-algorithm of Network-Popping.

- Build-Cycle ([41], Chapter 2): Sub-algorithm of Network-Popping.

- Vertex-Growing ([41], Chapter 2): Sub-algorithm of Network-Popping.

- Network-Popping ([41], Chapter 2): Builds a labelled level-1 network $\mathcal{N}$ representing a symbolic 3-dissimilarity δ if such a network exists, or returns the statement that δ is not level-1 representable.

- Transform ([41], Chapter 2): Transforms a labelled level-1 network $\mathcal{N}$ into a semi-discriminating, partially resolved and weakly labelled network $\mathcal{N}'$ such that $\mathcal{N}$ and $\mathcal{N}'$ represent the same symbolic 3-dissimilarity.

- OSF-Builder ([54], Chapter 5): Builds an optimal OSF $\psi : V(G) \to V(F)$ from an AS-forest $\mathcal{F} = (F, G, \phi)$.

Index

References

[1] A. V. Aho, Y. Sagiv, T. G. Szimansky, and J. D. Ullman. Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. *SIAM Journal of Computing*, 10(3):405–421, 1981. 31, 32, 154

[2] M. A. Alexandrou, C. Oliveira, M. Maillard, R. A. R. McGill, J. Newton, S. Creer, and M. I. Taylor. Competition and phylogeny determine community structure in Müllerian co-mimics. *Nature*, 469:84–88, 2011. 116

[3] H.-J. Bandelt and A. W. M. Dress. A Canonical Decomposition Theory for Metrics on a Finite Set. *Advances in Mathematics*, 92:47–105, 1992. 19, 20, 21, 23, 24

[4] P. Bastide, M. Mariadassou, and S. Robin. Detection of adaptive shifts on phylogenies using shifted stochastic processes on a tree. *Journal of the Royal Statistical Society*, 79(4):1067–1093, 2017. 124, 139, 145

[5] C. Baudet, B. Donati, B. Sinaimeri, P. Crescenzi, C. Gautier, C. Matias, and M.-F. Sagot. Cophylogeny reconstruction via an approximate Bayesian computation. *Syst. Biol.*, 55:1–30, 2015. 118, 119

[6] S. Böcker and A. W. M. Dress. Recovering symbolically dated, rooted trees from symbolic ultrametrics. *Advances in Mathematics*, 138:105–125, 1998. 26, 27

[7] M. Bordewich and C. Semple. A universal tree-based network with the minimum number of reticulations. *arXiv:1707.08274*, 2017. 15

[8] U. Brandes and S. Cornelsen. Phylogenetic graph models beyond trees. *Discrete Applied Mathematics*, 157(10):2361–2369, 2010. 24

[9] D. Bryant and V. Moulton. Neighbor-Net: An agglomerative method for the construction of phylogenetic networks. *Molecular Biology and Evolution*, 21(2):255–265, 2004. 23, 153

[10] P. Buneman. The recovery of trees from measures of dissimilarity. *Mathematics in Archeological and Historical Sciences, Edimburgh University Press*, pages 387–395, 1971. 3, 17, 20

[11] G. Cardona, M. Llabrés, F. Rosselló, and G. Valiente. A distance metric for a class of tree-sibling phylogenetic networks. *Bioinformatics*, 24(13):1481–1488, 2008. 14

[12] G. Cardona, F. Rosselló, and G. Valiente. Comparison of tree-child phylogenetic networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6:552–569, 2007. 14

[13] K. K. Dasmahapatra, J. R. Walters, A. D. Briscoe, J. W. Davey, A. Whibley, N. J. Nadeau, A. V. Zimin, D. S. T. Hughes, L. C. Ferguson, S. H. Martin, C. Salazar, J. J. Lewis, S. Adler, S.-J. Ahn, D. A. Baker, S. W. Baxter, N. L. Chamberlain, R. Chauhan, B. A. Counterman, T. Dalmay, L. E. Gilbert, K. Gordon, D. G. Heckel, H. M. Hines, K. J. Hoff, P. W. H. Holland, E. Jacquin-Joly, F. M. Jiggins, R. T. Jones, D. D. Kapan, P. Kersey, G. Lamas, D. Lawson, D. Mapleson, L. S. Maroja, A. Martin, S. Moxon, W. J. Palmer, R. Papa, A. Papanicolaou, Y. Pauchet, D. A. Ray, N. Rosser, S. L. Salzberg, M. A. Supple, A. Surridge, A. Tenger-Trolander, H. Vogel, P. A. Wilkinson, D. Wilson, J. A. Yorke, F. Yuan, A. L. Balmuth, C. Eland, K. Gharbi, M. Thomson, R. A. Gibbs, Y. Han, J. C. Jayaseelan, C. Kovar, T. Mathew, D. M. Muzny, F. Ongeri, L.-L. Pu, J. Qu, R. L. Thornton, K. C. Worley, Y.-Q. Wu, M. Linares, M. L. Blaxter, R. H. ffrench Constant, M. Joron, M. R. Kronforst, S. P. Mullen, R. D. Reed, S. E. Scherer, S. Richards, J. Mallet, and W. O. M. . C. D. Jiggins. Butterfly genome reveals promiscuous exchange of mimicry adaptations among species. *Nature*, 487(7405):94–8, 2012. 129, 130

[14] M. Deza and M. Laurent. *Geometry of Cuts and Metrics*. Springer, Berlin, 1997. 19

[15] J.-P. Doyon, S. Hamel, and C. Chauve. An efficient method for exploring the space of gene tree/species tree reconciliations in a probabilistic framework. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9(1):26–39, 2012. 118

[16] A. Dress, K. T. Huber, J. Koolen, V. Moulton, and A. Spillner. *Basic Phylogenetic Combinatorics*. Cambridge University Press, 2012. 22, 41, 104, 105

[17] A. W. Dress, K. T. Huber, and V. Moulton. Some uses of the farris transform in mathematics and phylogenetics. a review. *Annals of Combinatorics*, 11:1–37, 2007. 12

[18] A. W. M. Dress, M. Hendy, K. T. Huber, and V. Moulton. On the number of vertices and edges of the Buneman Graph. *Ann. Comb.*, 1:339–352, 1997. 104, 106

[19] A. W. M. Dress, K. T. Huber, J. Koolen, and V. Moulton. Blocks and cut vertices of the Buneman graph. *Siam J. Discrete Mathematics*, 25(4):1902–1919, 2011. 106, 107

[20] A. W. M. Dress and D. H. Huson. Constructing Splits Graphs. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(3):109–115, 2004. 21, 23

[21] L. Excoffier, J. Novembre, and S. Schneider. Simcoal: A general coalescent program for the simulation of molecular data in interconnected populations with arbitrary demography. *Hered.*, 91(6):506–509, 2000. 130

[22] J. S. Farris, A. G. Kluge, and M. J. Eckardt. A numerical approach to phylogenetic systematics. *Systematic Zoology*, 19:172189, 1970. 12

[23] J. Felsenstein. An alternating least squares approach to inferring phylogenies from pairwise distances. *Syst. Biol.*, 46(1):101–111, 1997. 16

[24] J. Felsenstein. *Inferring Phylogenies.* Sinauer Associates, 2003. 29

[25] W. M. Fitch. Toward defining the course of evolution: minimum change for a specific tree topology. *Syst. Zool.*, 20:406–416, 1971. 116, 122, 154

[26] A. R. Francis and M. Steel. Which phylogenetic networks are merely trees with additionals arcs? *Syst. Biol.*, 64(5):768–777, 2006. 14, 139

[27] P. Gambette, V. Berry, and C. Paul. Quartets and unrooted phylogenetic networks. *J Bioinform Comput Biol*, 10(4), 2012. 12, 13, 24, 94

[28] P. Gambette and K. T. Huber. On encodings of phylogenetic networks of bounded level. *Journal of Mathematical Biology*, 61(1):157–180, 2012. 33

[29] P. Gambette, K. T. Huber, and G. E. Scholz. Uprooted phylogenetic networks. *Bulletin of Mathematical Biology*, 79(9):2022–2048, 2017. 4, 24, 87

[30] S. Grünewald, Y. Long, and Y. Wu. Reconstructing unrooted phylogenetic trees from symbolic ternary metrics. *arXiv:1702.00190*, 2017. 68, 72, 73

[31] V. Gurvich. Some properties and applications of complete edge-chromatic graphs and hypergraphs. *Soviet Math. Dokl.*, 30(3):803–807, 1984. 27, 68, 72

[32] V. Gurvich. Decomposing complete edge-chromatic graphs and hypergraphs. *Discrete Applied Math.*, 157:3069–3085, 2009. 27

[33] D. Gusfield. *ReCombinatorics: the algorithms of ancestral recombination and explicit phylogenetic networks.* MIT Press, 2014. 8, 13

[34] M. Hayamizu and K. Fukumizu. On the existence of infinitely many universal tree-based networks. *Journal of theoretical biology*, 396:204–206, 2016. 15

[35] M. Hellmuth, M. Hernandez-Rosales, K. T. Huber, V. Moulton, P. F. Stadler, and N. Wieseke. Orthology relations, symbolic ultrametrics and cographs. *J. Math. Biol.*, 66(1-2):399–420, 2013. 27, 37, 52, 153

[36] S. Herrmann, K. T. Huber, V. Moulton, and A. Spillner. Recognising treelike k-dissimilarities. *Journal of Classification*, 29(3):321–340, 2012. 29, 30, 78

[37] D. Howarth and D. Baum. Genealogical evidence of homoploid hybrid speciation in an adaptive radiation of scaevola (goodeniaceae) in the hawaiian islands. *Evolution*, 59(5):948–961, 2005. 126, 127

[38] K. T. Huber and V. Moulton. Encoding and constructing 1-nested phylogenetic networks with trinets. *Algorithmica*, 66(3):714–738, 2013. 33, 34

[39] K. T. Huber, V. Moulton, and G. E. Scholz. 3-way symbolic tree maps and 3-way symbolic ultrametrics. *Journal of Classification*, 2017. 4, 30, 67

[40] K. T. Huber, V. Moulton, L. van Iersel, and T. Wu. How much information is needed to infer reticulate evolutionary history ? *Syst. Biol.*, 64(1):102–111, 2015. 34

[41] K. T. Huber and G. E. Scholz. Beyond representing orthology relations by trees. *Algorithmica*, 80(1):73–103, 2018. 4, 29, 35, 154

[42] D. Huson and R. Rupp. Summarizing multiple gene trees using cluster networks. *WABI 2008: Algorithms in Bioinformatics*, pages 296–305, 2008. 10

[43] D. Huson, R. Rupp, and C. Scornavacca. *Phylogenetic Networks.* Cambridge University Press, 2010. 5, 8, 31

[44] J. Jansson, N. B. Nguyen, and W.-K. Sung. Algorithm for combining rooted triplets into a galled tree. *SIAM Journal of Computing*, 35(5):1098–1121, 2006. 31, 33

[45] J. Jansson and W.-K. Sung. Nonbinary tree-based phylogenetic networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2016. 15

[46] K. M. Kozak, N. Wahlberg, A. F. E. Neild, K. K. Dasmahapatra, J. Mallet, and C. D. Jiggins. Multilocus species trees show the recent adaptive radiation of the mimetic heliconius butterflies. *Syst. Biol.*, 64(3):505–524, 2015. 116, 117, 132

[47] D. M. Kristensen, Y. I. Wolf, A. R. Mushegian, and E. V. Koonin. Computational methods for gene orthology inference. *Brief. Bioinf.*, 12(5):379–91, 2011. 25

[48] M. Lafond and N. El-Mabrouk. Orthology relation and gene tree correction: complexity results. *WABI 2015, Algorithms in Bioinformatics*, 9289:966–79, 2015. 36

[49] T. Marcussen, S. R. Sandve, L. Heier, M. Spannagl, M. Pfeifer, The International Wheat Genome Sequencing Consortium, K. S. Jakobsen, B. B. H. Wulffe, B. Steuernagel, K. F. X. MAyer, and O.-A. Olsen. Theoretical and computational considerations of the compatibility of qualitative taxonimic characters. *Science*, 1250092, 2014. 1

[50] C. A. Meacham. Theoretical and computational considerations of the compatibility of qualitative taxonimic characters. *NATO ASI Series Vol. G1*, pages 304–314, 1983. 20, 153

[51] L. Nakhleh. Computational approaches to species phylogeny inference and gene tree reconciliation. *Trends in Ecology & Evolution*, 28(12):719–728, 2013. 25, 118

[52] L. Pachter and D. Speyer. Reconstructing trees from subtree weights. *Applied Mathematics Letters*, 17(6):615–621, 2004. 29

[53] N. Saitou and M. Nei. The Neighbour-Joining method : a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4:406–425, 1987. 17, 153

[54] G. E. Scholz, A.-A. Popescu, M. Taylor, V. Moulton, and K. T. Huber. OSF-Builder: A new tool for constructing and representing phylogenetic histories involving introgression. *submitted*, 2017. 4, 116, 139, 154

[55] C. Semple and M. Steel. *Phylogenetics*. Oxford University Press, 2003. 17, 33, 52, 82, 131

[56] R. R. Sokal and C. D. Michener. A statistical method for evaluating systematic relationships. *University of Kansas Scientific Bulletin*, 28:1409–1438, 1958. 17, 153

[57] R. Tatusov, E. V. Koonin, and D. J. Lipman. A genomic perspective on protein families. *Science*, 278:631–637, 1997. 25

[58] L. J. J. van Iersel, J. Keijsper, S. M. Kelk, L. Stougie, F. Hagen, and T. Boekhout. Constructing level-2 phylogenetic networks from triplets. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(4):667–681, 2009. 31, 33

[59] L. J. J. van Iersel and S. M. Kelk. Constructing the simplest possible phylogenetic network from triplets. *Algorithmica*, 60(2):207–235, 2011. 31

[60] L. J. J. van Iersel, S. M. Kelk, and M. Mnich. Uniqueness, intractability and exact algorithm : reflections on level-k phylogenetic networks. *Journal of Bioinformatics and Computational Biology*, 7(4):597–623, 2009. 33

[61] L. J. J. van Iersel and V. Moulton. Trinets encode tree-child and level-2 phylogenetic networks. *Journal of Mathematical Biology*, 68(7), 2013. 34

[62] R. W. R. Wallbank, S. W. Baxter, C. Pardo-Diaz, J. J. Hanly, M. S. H., J. Mallet, K. K. Dasmahapatra, C. Salazar, M. Joron, N. Nadeau, W. O. McMillan, and C. D. Jiggins. Evolutionary novelty in a butterfly wing pattern through enhancer shuffling. *PLoS Biol.*, 14(1):e1002353, 2016. 116, 129

[63] N. Wieseke, M. Bernt, and M. Middendorf. Unifying parsimonious tree reconciliation. *Lecture Notes in Computer Science*, 8126, 2013. 119