Efficiency of Algorithms in Phylogenetics

NIHAN TOKAC

Efficiency of Algorithms in Phylogenetics

Nihan Tokaç

Department of Engineering and Computing Sciences
University of Durham

This dissertation is submitted for the degree of
Doctor of Philosophy

St. Aidan's College September 2016

ABSTRACT

Phylogenetics is the study of evolutionary relationships between species. Phylogenetic trees have long been the standard object used in evolutionary biology to illustrate how a given set of species are related. There are some groups (including certain plant and fish species) for which the ancestral history contains reticulation events, caused by processes that include hybridization, lateral gene transfer, and recombination. For such groups of species, it is appropriate to represent their ancestral history by phylogenetic networks: rooted acyclic digraphs, where arcs represent lines of genetic inheritance and vertices of in-degree at least two represent reticulation events. This thesis is concerned with the efficiency, accuracy, and tractability of mathematical models for phylogenetic network methods.

Three important and related measures for summarizing the dissimilarity in phylogenetic trees are the minimum number of hybridization events required to fit two phylogenetic trees onto a single phylogenetic network (the *hybridization number*), the (rooted) subtree prune and regraft distance (the *rSPR distance*) and the tree bisection and reconnection distance (the *TBR distance*) between two phylogenetic trees. The respective problems of computing these measures are known to be NP-hard, but also fixed-parameter tractable in their respective natural parameters. This means that, while they are hard to compute in general, for cases in which a parameter (here the hybridization number and rSPR/TBR distance, respectively) is small, the problem can be solved efficiently even for large input trees. Here, we present new analyses showing that the use of the "cluster reduction" rule – already defined for the hybridization number and the rSPR distance and introduced here for the TBR distance – can transform any $O(f(p) \cdot n)$-time algorithm for any of these problems into an $O(f(k) \cdot n)$-time one, where n is the number of leaves of the phylogenetic trees, p is the natural parameter and k is a much stronger (that is, smaller) parameter: the minimum *level* of a phylogenetic network displaying both trees. These results appear in [9].

Traditional "distance based methods" reconstruct a phylogenetic tree from a matrix of pairwise distances between taxa. A phylogenetic network is a generalization of a phylogenetic tree that can describe evolutionary events such as reticulation and hybridization that are not tree-like. Although evolution has been known to be more accurately modelled by a network than a tree for some time, only recently have efforts been made to directly reconstruct a phylogenetic network from sequence data, as opposed to reconstructing several trees first

and then trying to combine them into a single coherent network. In this work, we present a generalisation of the UPGMA algorithm for ultrametric tree reconstruction which can accurately reconstruct ultrametric tree-child networks from the set of distinct distances between each pair of taxa. This result will also appear in [15]. Moreover, we analyse the safety radius of the NETWORKUPGMA algorithm and show that it has safety radius 1/2. This means that if we can obtain accurate estimates of the set of distances between each pair of taxa in an ultrametric tree-child network, then NETWORKUPGMA correctly reconstructs the true network.

STATEMENT OF COPYRIGHT

ACKNOWLEDGEMENTS

I could never have written this without a great amount of help from many different people. I want to thank my supervisor Magnus Bordewich for guiding me, for helping me, for correcting my mistakes, for supplying ideas, and for the enjoyable time I had while working with him. I want to thank Charles Semple for the work we did together and inviting me Canterbury University. I want to thank Celine Scornavacca and Matthias Weller for the work we did together. I also want to thank Turkish Ministry of Education for funding my research.

I would also like to thank Professor Katharina Huber and Matthew Johnson for serving on my thesis committee, for taking time to review and evaluate my dissertation, and for their helpful comments on my dissertation, my presentation and communication skills, as well as on my research and career.

I thank my friends who are always with me since the beginning of the postgraduate studies, and I know that they are going to be lifelong friends. First of all, Hatice Buber is the first person from the time I learned the scholarship to today. Hulya Bicer and Fatma Betul Durak are my first room mates. Saliha Oner and Samet Caliskan listened and shared all problems related to PhD and life. Haticenur and Bunyamin Keskin, and Aysenur Aydin, always felt us the warmness of Turkish community. Selime Soyucok, personal problems we shared, politics we discussed, fashion we talked about, ..., such a perfect coffee times. Murat Aydemir, how much time we shared together. And finally, Cigdem Sazak who is the last one I spent less time but I feel like I know her since I was born.

I thank my mother, my father and my brother for always being with me, believing in me, and supporting me.

Finally, I thank my husband Mustafa Tokac, for his love and support, although words do not suffice to thank him for understanding and tolerating my stress.

I would like to dedicate this thesis to my loving family & Mustafam ...

DECLARATION

I hereby declare that the work contained in this thesis is my own and has not been submitted for examination for any other degree at any University. The work of collaborators is acknowledged as appropriate.

On the fixed parameter tractability of agreement-based phylogenetic distances presented in Chapter 3 were made by Magnus Bordewich, Céline Scornavacca, Mathias Weller and by the author [9].

An algorithm for reconstructing ultrametric tree-child networks from intertaxa distances presented in Chapter 4 were made by Magnus Bordewich and by the author [15].

Nihan Tokaç
September 2016

CONTENTS

LIST OF FIGURES

LIST OF TABLES

List of Abbreviations and Symbols

Abbreviations

$\mathrm{LCA}()$	Lowest common ancestor
rSPR Distance	Rooted subtree prune and regraft distance
TBR Distance	Tree bisection and reconnection distance
AF	Agreement forest
FPT	Fixed parameter tractability
MAF	Maximum agreement forest
uMAF	Unrooted maximum agreement forest

Symbols

$\mathcal{D}$	Set distance matrix
$\mathcal{D}(x, y)$	Set of distances between x and y
$\equiv$	Equivalance Relation
$\mathcal{N}$	A phylogenetic network
$d_{x,y}$	Shortest distance between x and y
ρ	The labelled root of a tree
n	The number of leaves in a tree
$\mathcal{T}$	A tree
$\mathcal{F}$	A forest
$h(\mathcal{T}, \mathcal{T}')$	The hybridization number of $\mathcal{T}$ and $\mathcal{T}'$
$d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}')$	Rooted subtree prune and regraft distance between $\mathcal{T}$ and $\mathcal{T}'$
$d_{\mathrm{TBR}}(\mathcal{T}, \mathcal{T}')$	Tree bisection and reconnection distance between $\mathcal{T}$ and $\mathcal{T}'$

1

INTRODUCTION

> *Molecular phylogeneticists will have failed to find the "true tree," not because their methods are inadequate or because they have chosen the wrong genes, but because the history of life cannot properly be represented as a tree.*
>
> Ford Doolittle, *Phylogenetic classification and the universal tree*, 1999

1.1 WHY HYBRIDIZATION NETWORKS

Since Darwin's first sketch of an evolutionary tree (Figure 1.1), biologists have used leaf labelled trees, namely 'phylogenetic trees', to represent evolutionary relationships. Typically a set of extant species are represented as leaves, common ancestors are represented as internal nodes above a set of species, and the universal common ancestor of species is represented as the root of the tree.

Although phylogenetic trees provide a useful representation of evolutionary relationships in biology, evolution cannot always be adequately described by the classical tree model.

Two online sources are useful to understand phylogenetic networks. First of all, the website named "Who is Who in Phylogenetic Networks" [27]. The website is not only shows the researchers and their publications, it also shows the software to analyse phylogenetic networks. Another source is the blog named "The Genealogical World of Phylogenetic Networks" [46]. The aim of the blog is posting news, announcements, results, and opinions in the field of biology, anthropology, computational science, and networks in phylogenetic analysis.

Molecular genetic processes such as hybridization, where species inherit genes from multiple parent species, lateral gene transfer, where organisms obtain genetic material from

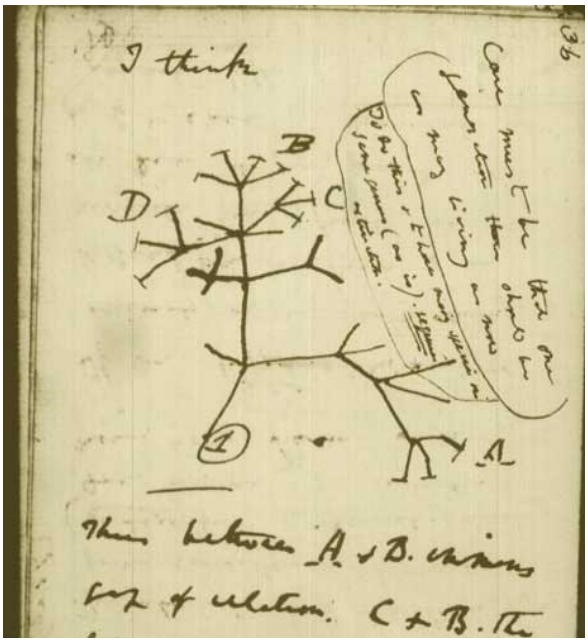


Figure 1.1 The first diagram by Charles Darwin of an evolutionary tree (First notebook on transmutation of species, 1837).

other organisms without actually being their offspring, or other such events, while still maintaining a flow of time was hard to answer. However, the advent of high-throughput sequencing technology addresses these questions.

With the increasing recognition of the role of reticulation events (such as hybridization or lateral gene transfer) in evolution, has come the need for developing mathematical models and new tools capable of better representing these phenomena. For such group of species, it is appropriate to represent their ancestral history by *phylogenetic networks* which are a generalization of trees that allow vertices with multiple parents.

Figure 1.2 illustrates the phylogenetic network *"Table de L'Ordre des Chiens"* was produced by Georges-Louis Leclerc, comte de Buffon (1707-1788). The figure illustrates the different kind of dogs and mixture of their races. While dark lines in the network represent the underlying tree of parent to offspring, light grey lines represent the hybridization events.

This thesis is about the use of networks in phylogenetic analysis, as a replacement for (or an adjunct to) the usual use of trees. Note that a tree is a network without hybridization nodes. Also, it develops new mathematical techniques and tools to study reticulate evolution in biology.

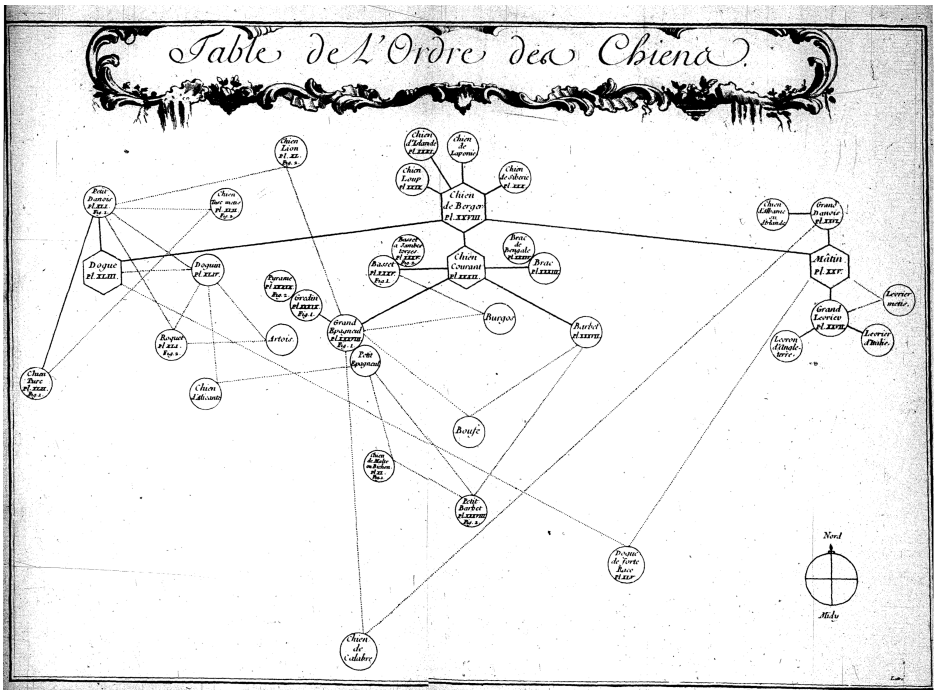


Figure 1.2 Genealogical network of races of dogs (*"Table de L'Ordre des Chiens"*) produced by Georges-Louis Leclerc, comte de Buffon (1707-1788). Reprinted from Histoire Naturelle by Buffon: the web edition. Retrieved December 15, 2015, from http://www.buffon.cnrs.fr/

1.2 THESIS OUTLINE

The thesis is modular in nature, and it is intended that each of its chapters could be read stand alone. The rest of this chapter gives a brief overview of the content. Readers unfamiliar with phylogenetic terminology might benefit from skipping ahead to Chapter 2 which provides some useful graph-theoretic background, focusing on some concepts from the mathematical foundations of phylogenetics. Specifically, Section 2.2 gives a brief introduction to phylogenetic trees, and Section 2.4 explains phylogenetic networks.

In Chapter 3 which is joint work with Celine Scornavacca and Mathias Weller, we discuss *fixed parameter tractability* algorithms which have a running time that is exponential in some parameter that is specific to the problem but independent of the input size. See [48] for an introduction to fixed-parameter tractability. To show that the problem is fixed parameter tractable a minimum number of reticulation events can be computed in time $O(f(k) + p(n))$, where n is the number of species, k is the minimum number of reticulation events, f is some computable function, and p is a fixed polynomial in n.

For HYBRIDIZATION NUMBER problem, minimum number of reticulation events, ℓ, is the number of hybridization events of hybridization network $\mathcal{N}$ which displays two

trees. Fixed parameter algorithm for HYBRIDIZATION NUMBER is given by Bordewich and Semple [11] that runs in $O((28 \cdot \ell)^{\ell} + n^3)$ time. Whidden and Zeh [63] gave an algorithm runs in $O(3^{\ell} \cdot n \cdot \log n)$ time. Whidden et al. [61] reduce the running time of algorithm to $O(2.42^{\ell} \cdot n \cdot \log n)$.

For RSPR DISTANCE problem, minimum number of reticulation events, ℓ, is the $rSPR$ distance between two trees. Fixed parameter algorithm for the RSPR DISTANCE is given by Bordewich et al. [10] that runs in $O(4^{\ell} \cdot \ell^4 + n^3)$ time where ℓ is the distance between two trees. Whidden and Zeh [63] gave an algorithm runs in $O(3^{\ell} \cdot n)$ time. Van Iersel et al. [57] extends the algorithm of Whidden and Zeh [63] to non necessarily binary phylogenies and requires $O(4^{\ell} \cdot p(n))$ time, where is a polynomial in n. Recently, Whidden et al. [61] reduce the running time of algorithm $O(2.42^{\ell} \cdot n)$.

For TBR DISTANCE problem, minimum number of reticulation events, ℓ, is the TBR distance between two trees. Fixed parameter algorithm for TBR DISTANCE is given by Hallett et al. [35] that runs in $O(4^{\ell} \cdot \ell^5 + p(n))$ time. Whidden et al. [63] reduce the running time of algorithm $O(4^{\ell} \cdot n)$.

We present new analysis in [9] showing that the use of the "cluster reduction" rule, and give a fixed parameter algorithm for HYBRIDIZATION NUMBER, RSPR DISTANCE, and TBR DISTANCE problems where the parameter k is the distance between clusters instead of the distance between trees which is a number of operation to obtain a tree from the another tree. Table 1.1 summarises all FPT results regarding HYBRIDIZATION NUMBER, RSPR DISTANCE and TBR DISTANCE and the comparison with our new results.

In Chapter 4, we focus on distance based network reconstruction methods which determine the evolutionary relationships between a set of species typically from information contained in biomolecular sequence data. Reconstructing phylogenetic trees is one of the problems in evolutionary biology. There are two approaches under this reconstruction methods: computationally intensive methods and faster methods. Computationally intensive methods find a tree that best displays the full DNA data, such as maximum likelihood or maximum parsimony. Faster methods reduce the data to a matrix of pairwise genetic distances between taxa, and find a tree is closest to realizing these distances as inter-taxa path lengths, such as Neighbor Joining [51], UPGMA [54], BioNJ [28], least squares [25], weighted least squares [44], which take as input a distance matrix between species, belong this strategy. Recently, Bordewich and Semple [13] apply distance based methods to networks.

Our phylogenetic network construction [15] contains two main parts. Sokal [54] introduced UPGMA algorithm for ultrametric tree reconstruction from distance information. At first part, we focus principally on generalisation of the UPGMA algorithm, which is called

	Fixed Parameter Tractability	
HYBRIDIZATION NUMBER	Previous:	$O((28 \cdot \ell)^{\ell} + n^3)$ time [11] $O(3^{\ell} \cdot n \cdot \log n)$ time [63] $O(2.42^{\ell}.n.\log n)$ time [61] $O((18 \cdot \ell)^{\ell} + n^3)$ time [41] $O(3.18^{\ell} \cdot n)$ time [62]
	New:	$O(3.18^{k} \cdot n)$ time [9]
RSPR DISTANCE	Previous:	$O(4^{\ell} \cdot \ell^4 + n^3)$ time [10] $O(4^{\ell} \cdot \ell^4 + n^3)$ time [7] $O(3^{\ell} \cdot n)$ time [63] $O(2.42^{\ell} \cdot n)$ time [61] $O(2.344^{\ell} \cdot n)$ time [20]
	New:	$O(2.344^{k} \cdot n)$ time [9]
TBR DISTANCE	Previous:	$O(\ell^{3\ell} + p(n))$ time [1] $O(4^{\ell} \cdot \ell^5 + p(n))$ time [34] $O(4^{\ell} \cdot n)$ time [63]
	New:	$O(3^{k} \cdot n)$ time [9]

Table 1.1 Previous and new results on HYBRIDIZATION NUMBER, RSPR DISTANCE, TBR DISTANCE where n is the input size, ℓ is the distance between two trees, and k is the distance between clusters.

NETWORKUPGMA, to reconstruct ultrametric tree-child networks from the set of distinct distances.

The fundamental result behind the distance based reconstruction methods is a network $\mathcal{N}$ can be reconstructed from its metric $\mathcal{D}_{\mathcal{N}}$ [30]. However, in practice $\mathcal{D}_{\mathcal{N}}$ is unknown, the distance matrix $\mathcal{D}$ has noise due to systematic errors such as incorrect assumptions in the tree-construction method. A minimal requirement for any distance based method is consistency: for any network $\mathcal{N}$, and for distance matrices $\mathcal{D}$ "close enough" to $\mathcal{D}_{\mathcal{N}}$, the algorithm should output a network with the same topology as $\mathcal{N}$. The second part of Chapter 4 deals with the question of when any distance algorithm for phylogeny reconstruction can be guaranteed to output the correct phylogeny as a function of the divergence between $\mathcal{D}$ and $\mathcal{D}_{\mathcal{N}}$. Atteson [3] introduced 'safety radius' to measure the maximum error in set of estimated distances. It has been shown that we can reconstruct ultrametric tree-child network when we obtain accurate estimates of the set of distances between each pair of taxa. The second part of Chapter 4 deals with the safety radius of NETWORKUPGMA algorithm.

Finally, in Chapter 5, I present concluding remarks and discuss some work that might be pursued based of the results in this thesis.

2

PRELIMINARIES

In this chapter, we review concepts and definitions relevant to this dissertation. The first section covers basic graph-based definitions. Section 2.2 introduces phylogenetic trees and discusses tree arrangement operations, and Section 2.3 explains phylogenetic trees with tree based metrics and tree reconstruction methods. Then we introduce phylogenetic networks in Section 2.4. See [53] for more basic definitions. Additional definitions and notations are introduced in the chapters as necessary.

2.1 BASIC CONCEPTS AND DEFINITIONS

As far as possible we have tried to keep the notation consistent between chapters to follow Semple and Steel [53]. Wherever encountered, X is a set of taxa. A taxon can be a species, in the case of interspecies data, or an individual in an intra-species data-set (taxa is the plural set of taxon). The number of taxa in the set X is denoted by n.

A graph G is an ordered pair (V, E) consisting of a non-empty set V of vertices and a multiset E of *edges* each of which is an element of $\{\{x, y\} : x, y \in V\}$. If $e = (u, v)$ is an edge of a graph G, then u an v are adjacent, and e is said to be incident with u and v.

A path in a graph G is a sequence of distinct vertices $v_1, v_2, ..., v_k$ where $k \geq 1$ such that, for all $i \in \{1, 2, ..., k-1\}$, v_i and v_{i+1} are adjacent. If, in addition, v_1 and v_k are adjacent, then the subgraph of G whose vertex set is $\{v_1, v_2, ..., v_k\}$ is a *cycle*. A graph is said to be *connected* if there is a path between each pair of vertices; otherwise, G is *disconnected*. For example, in Figure 2.1 (a), there are 9 vertices and 9 edges. The ordered set of vertices (a, b, c) is a cycle. There is an edge $u = (a, b)$ where a and b are adjacent and end vertices of u.

A *directed graph (digraph)* D is an ordered pair (V, E) consisting of a non-empty set V of vertices and a set $E \subseteq V \times V$ of arcs. If $e = (u, v)$ is an arc, then u is the tail and v is the head of e. The arc e is said to be directed from u to v. A *directed path* of a digraph $D = (V, E)$ is an ordered set of vertices $(v_1, v_2, ..., v_k)$ where for each pair of vertices v_i, v_{i+1} there is an edge $e \in E$ with $e = (v_i, v_{i+1})$. If $v_0 = v_k$ then p is a *directed cycle* of D. Let v

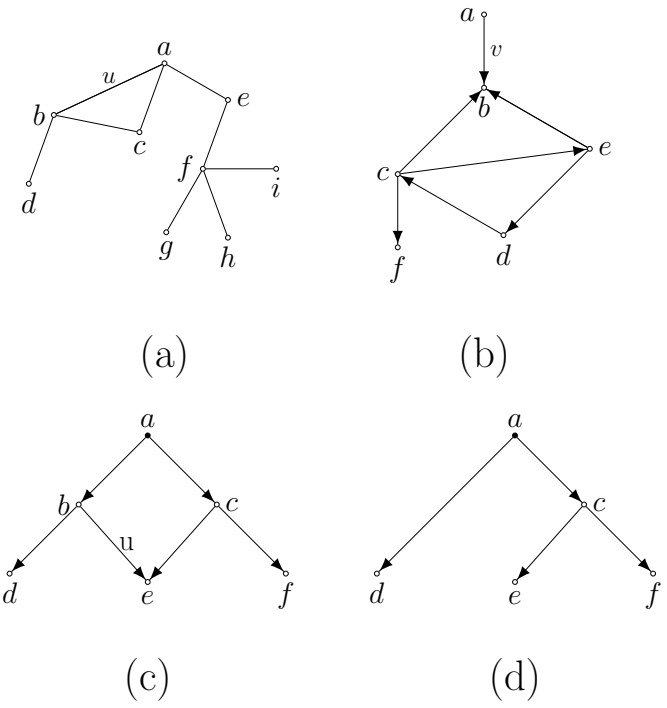


Figure 2.1 Examples of basic concepts with graphs. (a) A graph. (b) A digraph. (c) A tree $\mathcal{T}$. (d) A subtree T/u.

be a vertex of a graph D. The *in-degree* (respectively out-degree) of a vertex v of D, denoted $d^-(v)$ (respectively, $d^+(v)$), is the number of arcs of D whose head (respectively, tail) is v. The *degree* of a vertex is the sum of its in-degree and out-degree. Figure 2.1 (b) is a digraph with 5 vertices and 7 edges. The ordered set of vertices (c, e, d) is a directed cycle. For the edge $v = (a, b)$, a is called tail of v, and b is called head of v. The in-degree of vertex b is $d^-(b) = 3$, and the out-degree of vertex b is $d^+(b) = 0$.

A *tree* $T = (V, E)$ is a connected digraph with no cycles. A vertex of degree zero is said to be isolated, a vertex of in-degree one and out-degree zero is called a *leaf* or an *external vertex*, and a vertex of out-degree bigger than zero is called an *internal vertex* or *tree vertex*. A tree is binary if every interior vertex has degree 3. A *rooted tree* is a tree that has exactly one distinguished vertex called the *root* which has in-degree 0 and, out-degree 2. The rooted tree shown in Figure 2.1 (c) where a is the root.

A connected subgraph of $\mathcal{T}$ is a *subtree* of $\mathcal{T}$. Obtaining a subtree is an important operation on a tree $T = (V, E)$. Let v be a tree vertex of $\mathcal{T}$, and let e be an edge of $\mathcal{T}$ incident with v. The tree T/e or T_v is said to be obtained from $\mathcal{T}$ by *suppressing* v. The subtree in Figure 2.1 (d), T/u or T_b is obtained from the tree $\mathcal{T}$ in Figure 2.1 (c) by suppressing vertex b.

2.2 PHYLOGENETIC TREES

An X-tree is an ordered pair (T, ϕ), where $\mathcal{T}$ is a tree with vertex set V and $\phi : X \to V$ is a map with the property that, for each $v \in V$ of degree at most two, $v \in \phi(X)$.

A phylogenetic X-tree (or a phylogenetic tree on X) $\mathcal{T}$ is an X-tree $(T; \phi)$ with the property that ϕ is a bijection from X into the set of leaves of $\mathcal{T}$. If, in addition, every interior vertex of $\mathcal{T}$ has degree three, then $\mathcal{T}$ is called a *binary phylogenetic tree*. The set X is called the label set of $\mathcal{T}$ and is denoted by $\mathcal{L}(\mathcal{T})$. We can view X as the set of leaves of $\mathcal{T}$ and consequently, denote the leaves of $\mathcal{T}$ by the elements of X, see Figure 2.2 (a) where $\mathcal{L}(T) = \{human, chimp, mouse, camel, frog, goat, sheep, cow\}$. A phylogenetic tree with single internal vertex that is adjacent to all the leaves, is called a *star tree*, see Figure 2.2 (b) which is a star tree on six leaves.

A vertex u is called *ancestor* of a vertex v, and a vertex v is called *descendant* of a vertex u in $\mathcal{T}$ if $\mathcal{T}$ contains a directed path from u to v. Consistent with this terminology, throughout the thesis, we will draw a rooted tree (or network) with the root at the top of the figure and oriented so as to respect the ancestor-descendant relationship.

For a vertex set $U \neq \emptyset$ in a network $\mathcal{N}$, we define the *common ancestor set* of U in $\mathcal{N}$ as the set of all vertices v that are ancestors of all $u \in U$ in $\mathcal{N}$ and the *lowest common ancestor set* of U in $\mathcal{N}$ is the result of removing all common ancestors that are ancestors of other common ancestors from this set. We write $\mathrm{LCA}_{\mathcal{N}}(U)$ to denote the set of lowest common ancestors of U in $\mathcal{N}$.

A phylogenetic tree can be a rooted digraph or an unrooted graph. Tree $\mathcal{T}$ is rooted if there is a distinguished vertex, ρ. In a rooted phylogenetic tree, the root corresponds to the common ancestor of all species or genes at its leaves. A rooted phylogenetic tree, therefore, shows not only the relative relationships of species but also the direction of evolution, from its root towards its leaves. An unrooted phylogenetic tree, on the other hand, only shows the relationship among species. Figure 2.2 (a) and (b) show examples of an unrooted phylogenetic tree, while Figure 2.2 (c) and (d) are examples of a rooted phylogenetic tree.

A *rooted binary phylogenetic X-tree* $\mathcal{T}$ is a rooted phylogenetic tree where the root has degree two, and every interior vertex has degree three. Since, in a rooted binary phylogenetic X-tree, all arcs are directed away from ρ, we can actually consider $\mathcal{T}$ as undirected [38] as shown in Figure 2.2 (c) and (d). For all $n \geq 1$, a *rooted caterpillar tree* on n leaves is any rooted binary phylogenetic tree for which the induced subtree on the internal vertices is a path graph with the root at one end of the path, see Figure 2.2 (d).

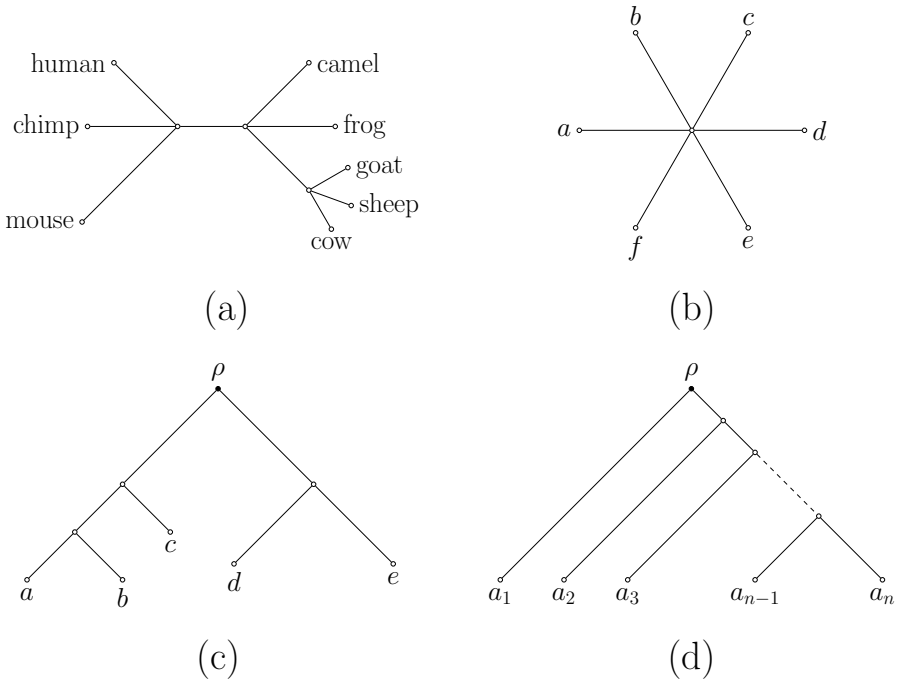


Figure 2.2 (a) An unrooted phylogenetic tree. (b) A star tree. (c) A rooted binary phylogenetic tree where $X = a, b, c, d, e$. (d) A rooted caterpillar tree.

2.2.1 Tree Rearrangement Operations

Numerous methods exist for reconstructing a tree from given data (such as aligned DNA sequences). Different data sets and different methods lead to different trees being reconstructed for the same set of species. Thus is is important to determine how "close" two reconstructed trees are. An easy way to say two trees are "close together" is if one can be obtained from the other by a small number of tree rearrangement operations.

There are three types of tree rearrangement operations on binary phylogenetic trees to understand how close two trees are. These operations are useful in numerous ways for reconstruction and comparison of phylogenetic trees [1]. The main focus for this thesis is the rooted tree bisection and reconnection operation and subtree prune and regraft operation. Results in this section play an important role in Chapter 3.

In the following descriptions of tree rearrangement operations, let $\mathcal{T}$ be a binary phylogenetic X-tree and $e = (u, v)$ is an edge of $\mathcal{T}$. We start to describe from the most general one.

Definition 2.2.1 (Tree Bisection and Reconnection Operation (TBR))**.** *Let $\mathcal{T}$ be an unrooted binary phylogenetic X-tree and $e = \{u, v\}$ be an edge of $\mathcal{T}$. Let $\mathcal{T}'$ be the unrooted binary*

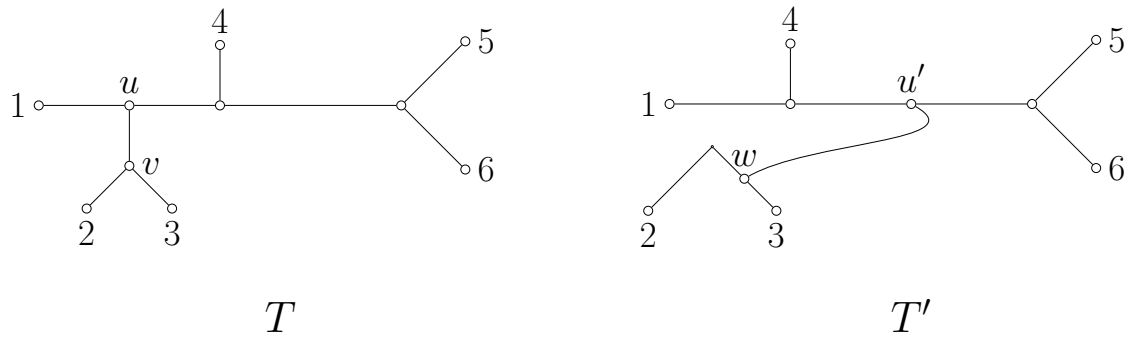


Figure 2.3 A schematic representation of the TBR operation.

phylogenetic X-tree obtained from $\mathcal{T}$ by deleting e and reconnecting the subtrees T_u and T_v by

(i) subdividing an edge of T_u with a new vertex u',

(ii) subdividing an edge of T_v with a new vertex w,

(iii) adding the edge $\{u', w\}$, and

(iv) suppressing any vertices of degree two.

See Figure 2.3 for schematic representation of an TBR.

Definition 2.2.2 (Rooted Subtree Prune and Regraft (rSPR))**.** *Let $\mathcal{T}$ be a binary rooted phylogenetic X-tree, and let $e = (u, v)$ be an arc of $\mathcal{T}$. Let $\mathcal{T}'$ be the rooted binary phylogenetic X-tree obtained from $\mathcal{T}$ by deleting e and then reconnecting v to the component T_u by:*

(i) creating a new vertex u' which subdivides an arc in T_u,

(ii) adding the arc (u', v), and

(iii) contracting the degree-two vertex u.

In this case, we say that $\mathcal{T}'$ is obtained from $\mathcal{T}$ by one r*ooted subtree prune and regraft (rSPR) operation. Note that the SPR operation can also be considered as a TBR, but not conversely. See Figure 2.4 for schematic representation of an SPR.*

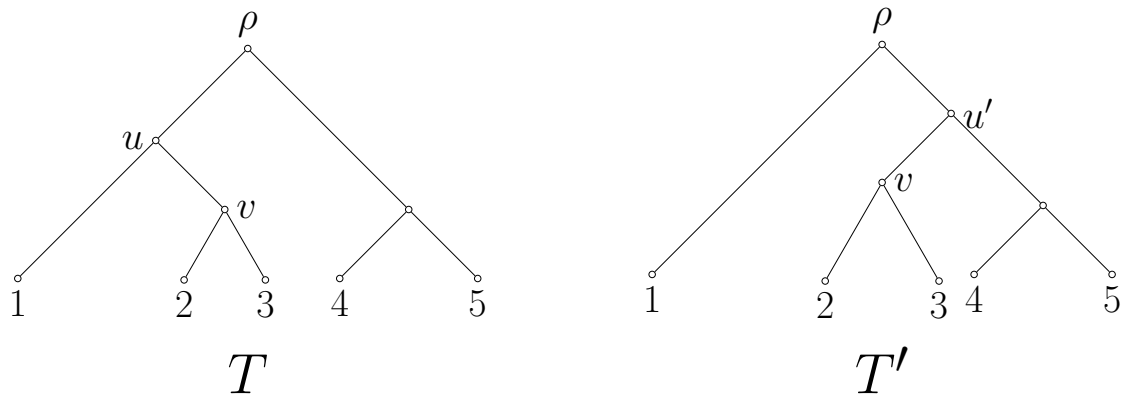


Figure 2.4 A schematic representation of the rSPR operation where T and T' are binary rooted trees.

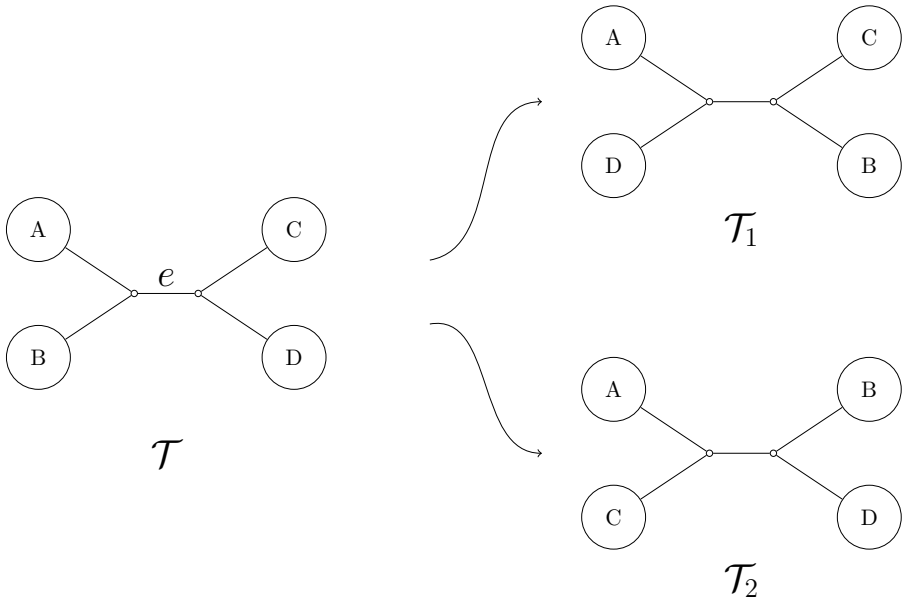


Figure 2.5 Binary unrooted trees $\mathcal{T}_1$ and $\mathcal{T}_2$ result from two possible NNI's about edge e in $\mathcal{T}$. Figure is adapted from [1].

Definition 2.2.3 (Nearest Neighbour Interchange (NNI))**.** *Let $\mathcal{T}$ be a binary phylogenetic X-tree. Let $e = (u, v)$ be an internal arc of $\mathcal{T}$, which has four subtrees attached to it. A nearest neighbour interchange (NNI) occurs when one subtree on one side of an internal edge is swapped with a subtree on the other side of the edge, as illustrated in Figure 2.5. NNI operation can also be applied for rooted trees (for details see [22]).*

The TBR and NNI distance between any two binary phylogenetic trees or rSPR distance between any two rooted binary phylogenetic trees $\mathcal{T}, \mathcal{T}'$ is the minimum number of TBR,

rSPR, NNI operations respectively, (of the given type) required to convert $\mathcal{T}$ into $\mathcal{T}'$. Chapter 3 focuses on the complexity of the TBR DISTANCE and RSPR DISTANCE problems, specifically their fixed parameter tractability.

The distance problem is *fixed parameter tractable* if the distance between two binary phylogenetic trees, each with n leaves and whose distance is at most k can be solved by an algorithm which runs in polynomial time (in n) and for which the degree of this polynomial is independent of k.

TBR DISTANCE problem is fixed parameter tractable where the number of TBR operations is bounded by a parameter k. Proof of this problem starts with reducing the size of the input phylogenetic trees $\mathcal{T}, \mathcal{T}'$. These reduction rules preserve the TBR distance between original trees. After applying these rules recursively, the resulting tree have size $k' \leq 4c(k-1)$, for constant c, and note that k' is independent of the leaf set size n (for details see Allen and Steel 2001 [1]). The theorem 2.2.4 shows that after applying the reduction rules the problem is parameterized by k, and the time for reduction is polynomial time over number of leaves.

Theorem 2.2.4. *[1] The parameterized* TBR DISTANCE *problem is fixed parameter tractable in $O(k^{3k} + p(n))$ time where $p(n)$ is a polynomial time required to apply the reduction rules.*

[10] showed that computing the RSPR DISTANCE between two rooted binary phylogenetic X-trees is fixed parameter tractable when parameterized by the rSPR distance between two trees (d_{rSPR}).

Theorem 2.2.5. *[10] The decision problem* RSPR DISTANCE*, parameterized by d_{rSPR}, is fixed parameter tractable.*

2.3 DISTANCE BASED METHODS

In this section, we describe some methods for constructing an edge-weighted X-tree from a distance matrix $\mathcal{D}$ on X. Distance based methods in phylogeny are tasked with recovering the tree $\mathcal{T}$ from an empirical distance between pairs of taxa in a labelled set X. The distance-based reconstruction problem is then that of recovering the underlying graph and edge lengths from $\mathcal{D}$.

In a weighted tree each edge has a specified weight (or length). In phylogenetics, a *distance function* d is a function from $V \times V$, where V is the set of vertices, to the set of non-negative real numbers. A distance function $d : X \times X \to \mathbf{R}^{\geq 0}$ should satisfy the following conditions, where $x, y \in X$ set of leaves:

1. *reflexive* if $d_{x,y} = 0$ if and only if $x = y$,
2. *symmetry* if $d(x, y) = d(y, x)$ for any $x, y \in V$,
3. *triangular inequality* if $d_{x,z} \leq d_{x,y} + d_{y,z}$ for any $x, y, z \in V$.

Let T be a phylogenetic tree on X. A *distance matrix* D is an $n \times n$ matrix where each entry $d_{x,y}$ gives the weight of the (unique) path between leaves $x, y \in X$. The distance matrix in Figure 2.6 (b) corresponds to the tree in Figure 2.6 (a). Each pair of taxa $x, y \in X$ is connected by a unique path in the tree. If x and y are leaves, then the distance between x and y written $d_{x,y}$, is the sum of the edge lengths (weights) on the unique path joining x and y. Let $\mathcal{D}$ be a distance matrix on X. The main goal is to construct a phylogenetic tree T for which the tree distances $T_{\mathcal{D}}$ provide a good approximation of the distances in $\mathcal{D}$.

Given a distance matrix $\mathcal{D}$ on X. $\mathcal{D}$ is *tree-like* if there exits some phylogenetic tree T such that $\mathcal{D} = \mathcal{D}_T$ where D_T is the true distances. Tree-like distances are also called *additive* since they can be obtained by adding the lenghts of edges along paths in a suitable tree. This global property of a distance function can be determined using the following lemma:

Lemma 2.3.1 (Four-point condition)**.** *A distance matrix $\mathcal{D}$ on X satisfies the* four point condition *if for every $w, x, y, z \in X$ the equation*

$$d_{w,x} + d_{y,z} \leq \max\{d_{w,y} + d_{x,z}, d_{w,z} + d_{x,y}\}$$

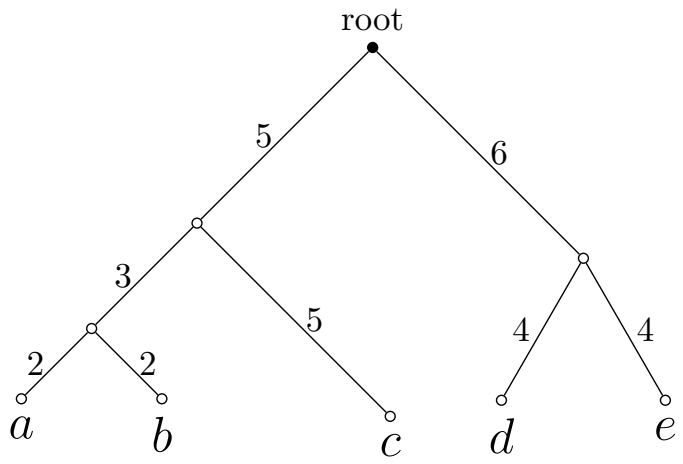
holds (that is, the two larger of the three possible sums are equal).

The following results is a fundamental result in phylogenetics:

Theorem 2.3.2 ([16])**.** *Let $\mathcal{D}$ be a distance matrix on X. Then $\mathcal{D}$ is additive, if and only if $\mathcal{D}$ satisfies the four point condition.*

A weighted, rooted tree is *ultrametric*, if every directed path from the root to a leaf has the same length, or clock-like when the weights on the edges correspond to units of time (see Figure 2.6 (a)). For any rooted phylogenetic tree $\mathcal{T}$, an evolutionary path between two leaves x and y is a simple path which goes up (i.e. moving in a child-to-parent direction) from x to a common ancestor u of x and y, and then down (i.e. moving in a parent-to-child direction) from u to y.

The most popular methods for computing a phylogenetic tree from distances are the UPGMA cited more than 3000 times, and NJ methods cited more than 40000 times. These methods start with identifying a cherry. A *cherry* of $\mathcal{T}$ is a pair of leaves of $\mathcal{T}$ with a common neighbor.



(a)

	a	b	c	d	e
a	0	4	10	20	20
b		0	10	20	20
c			0	20	20
d				0	8
e					0

(b)

Figure 2.6 (a) Ultrametric weighted rooted binary phylogenetic X-tree. (b) A distance matrix on $X = \{a, b, c, d, e\}$.

2.3.1 UPGMA (UNWEIGHTED PAIR GROUP METHOD WITH ARITHMETIC MEAN)

UPGMA is a hierarchical clustering method to reconstruct an ultrametric phylogenetic tree developed by Sokal and Michener [54]. Its main advantages are being a simple algorithm for tree construction and very fast. The main disadvantage is assuming a constant rate of evolution (molecular clock hypothesis): the clustering procedure works only if the data is ultrametric. Table 2.1 gives the full definition of UPGMA algorithm.

UPGMA

Input: Given a set of taxa X and a pair wise distance matrix D

Output:

1. Iteration: Nearest two clusters are combined into a higher-level cluster.

2. The distance between any two clusters A and B is taken to be the average of all distances between pairs of objects x in A and y in B, that is, the mean distance between each elements of each clusters:

$$D(A, B) = \frac{1}{|A|.|B|} \sum_{x \in A} \sum_{y \in B} D(x, y)$$

3. Clustering Procedure:

(a) Assume that initially each species is a cluster on its own.

(b) Join closest 2 clusters A, B and recalculate distance of the joint pair by taking the average $D(A, B)$.

(c) Repeat this process until all species are connected in a single cluster.

Table 2.1 UPGMA algorithm

The UPGMA algorithm is illustrated in Figure 2.7 for a set of taxa $X = \{a, b, c, d, e\}$. In the first iteration, the algorithm identify the smallest element in the distance matrix D, here, the distance between a and b is the shortest. Then, a, b form a cherry in the tree to be constructed. A subtree to be constructed is drawn with the parent halfway between the two, $D_{a,b}/2$. Since $D_{a,b} = 2$, each branch length is 1. In Cycle 2, replace the taxas a and b with ab and recalculate the distance matrix where ab is the new leaf in X. Choose the shortest distance in matrix; d, e. And draw the subtree similarly in Cycle 1. In Cycle 3, replace the taxas d and e with de and recalculate the distance matrix. Similarly, identify the smallest element in D; ab, c. Drawn a subtree to be constructed with ab and c. Then the ab subtree is attached to the ab branch at parent of a and b which has equal length to the a and b. At last cycle, replace the taxas ab and c with abc and recalculate the distance matrix, and repeat the process.

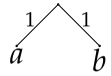
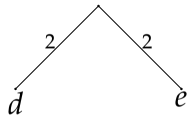
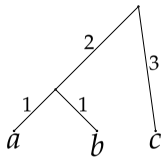
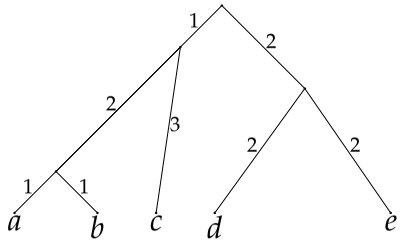
Step	Cycle 1	Cycle 2	Cycle 3	Cycle 4
Distance Matrix	$\begin{array}{c\|cccc} & a & b & c & d & e \\ \hline a & 0 & \boxed{2} & 6 & 8 & 8 \\ b & & 0 & 6 & 8 & 8 \\ c & & & 0 & 8 & 8 \\ d & & & & 0 & 4 \\ e & & & & & 0 \end{array}$	$\begin{array}{c\|cccc} & ab & c & d & e \\ \hline ab & 0 & 6 & 8 & 8 \\ c & & 0 & 8 & 8 \\ d & & & 0 & \boxed{4} \\ e & & & & 0 \end{array}$	$\begin{array}{c\|ccc} & ab & c & de \\ \hline ab & 0 & \boxed{6} & 8 \\ c & & 0 & 8 \\ de & & & 0 \end{array}$	$\begin{array}{c\|cc} & abc & de \\ \hline abc & 0 & \boxed{8} \\ de & & 0 \end{array}$
Identify Smallest D	$D_{a,b} = 2$	$D_{d,e} = 4$	$D_{ab,c} = 6$	$D_{abc,de} = 8$
Taxa Joined the tree to be constructed				

Figure 2.7 Example of UPGMA based tree construction where notation for a cluster $C = \{a, b, ..., d\}$ is $ab...d$.

2.3.2 NEIGHBOR JOINING (NJ)

NJ is a bottom-up clustering method for reconstructing an edge-weighted binary phylogenetic tree from an arbitrary distance matrix developed by Saitou and Nei [51]. The main advantages of the algorithm are that is work fast even for large dataset, and contrary to UPGMA algorithm, data does not need to be ultrametric. Table 2.2 gives the full definition of neighbor joining algorithm.

NJ algorithm is illustrated in Figure 2.8 for a set of taxa $X = \{a, b, c, d, e\}$. In first cycle, at first step, calculates the S values for each taxa using the Equation 2.1. At second step, calculate the Q matrix by Equation 2.2. At third step, by Theorem 2.3.3, smallest element in Q matrix, where c and d, form a cherry. At fourth step, creating a parent vertex u_1 of c and d, and calculate the distances from c to u and d to u by Equation 2.4. At fifth step, start with a star tree, then join c and d according to third step. Second cycle starts with reduced distance matrix which is constructed using Equation 2.5. The algorithm works similarly until there are three taxa left.

Neighbor Joining (NJ)

Input: Given a set of taxa X and a pair wise distance matrix D

Output:

1. Compute S values for every taxa in D where $i \in X$ and $k \in X$:

$$S_i = \sum_{k \neq i} D(i,k) \tag{2.1}$$

2. Calculate Q matrix for D:

$$Q_{i,j} = D(i,j) - \frac{1}{|X|-2}(S_i + S_j) \tag{2.2}$$

3. Then select a pair $i, j \in X$ that minimize $Q_{i,j}$ as motivated by the following theorem;

Theorem 2.3.3. *[51] Let D be the tree metric corresponding to the tree $\mathcal{T}$. The pair a, b that minimizes $Q_{i,j}$ is a cherry in the tree.*

4. If there are more than three taxa, replace the cherry i and j with a new vertex u such that:

$$S_{i,u} = D_{i,j}/2 + (S_i - S_j)/2 \tag{2.3}$$

5. Construct a new distance matrix of all other taxa to u where $x \in X - \{i, j\}$:

$$D_{x,u} = (D_{i,x} + D_{j,x} - D_{i,j})/2 \tag{2.4}$$

6. If there are more than three taxa, replace the cherry i and j with a leaf u, i.e. calculate the distance from the leaves $x \in X - \{i, j\}$ to the parent of i and j, and construct a new distance matrix where

$$D(x,u) = \frac{1}{2}(D(i,a) + D(i,b)) \tag{2.5}$$

This is called the reduction step.

7. Repeat until there are three taxa.

Table 2.2 The Neighbor-Joining algorithm

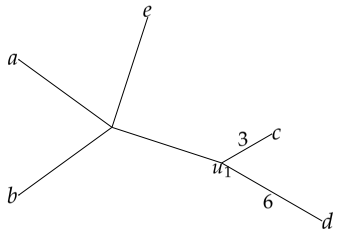
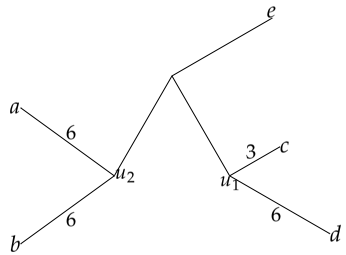
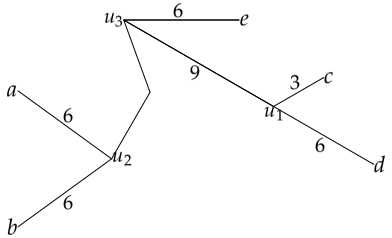
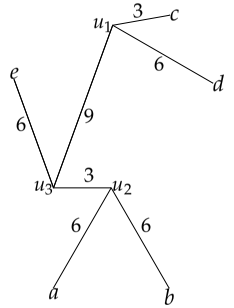
Step	Cycle 1	Cycle 2	Cycle 3	Cycle 4
Distance Matrix	$\begin{array}{c\|ccccc} & a & b & c & d & e \\ \hline a & 0 & 6 & 18 & 21 & 12 \\ b & & 0 & 18 & 21 & 12 \\ c & & & 0 & 9 & 18 \\ d & & & & 0 & 21 \\ e & & & & & 0 \end{array}$	$\begin{array}{c\|cccc} & a & b & u_1 & e \\ \hline a & 0 & 6 & 15 & 12 \\ b & & 0 & 15 & 12 \\ u_1 & & & 0 & 15 \\ e & & & & 0 \end{array}$	$\begin{array}{c\|ccc} & u_2 & u_1 & e \\ \hline u_2 & 0 & 12 & 9 \\ u_1 & & 0 & 15 \\ e & & & 0 \end{array}$	$\begin{array}{c\|cc} & u_2 & u_3 \\ \hline u_2 & 0 & 3 \\ u_3 & & 0 \end{array}$
Step 1: S values	$S_a = (6+18+21+12)/3 = 19$ $S_b = (6+18+21+12)/3 = 19$ $S_c = (18+18+9+18)/3 = 21$ $S_d = (21+21+9+21)/3 = 24$ $S_e = (12+12+18+21)/3 = 21$	$S_a = (6+15+12)/2 = 16.5$ $S_b = (6+15+12)/2 = 16.5$ $S_{u_1} = (15+15+15)/2 = 22.5$ $S_e = (12+12+15)/2 = 19.5$	$S_{u_2} = (12+9)/1 = 21$ $S_{u_1} = (12+15)/1 = 27$ $S_e = (9+15)/1 = 24$	Because $\|X\| - 2 = 0$, we cannot do this calculation.
Step 2: Q matrix	$\begin{array}{c\|ccccc} & a & b & c & d & e \\ \hline a & 0 & -32 & -22 & -22 & -28 \\ b & & 0 & -22 & -22 & -28 \\ c & & & 0 & -36 & -24 \\ d & & & & 0 & -24 \\ d & & & & & 0 \end{array}$	$\begin{array}{c\|cccc} & a & b & u_1 & e \\ \hline a & 0 & -27 & -24 & -24 \\ b & & 0 & -24 & -24 \\ u_1 & & & 0 & -27 \\ e & & & & 0 \end{array}$	$\begin{array}{c\|ccc} & u_2 & u_1 & e \\ \hline u_2 & 0 & -36 & -36 \\ u_1 & & 0 & -36 \\ e & & & 0 \end{array}$	
Step 3: Smallest $Q_{i,j}$	$Q_{c,d} = -36$	$Q_{a,b} = -27$ $Q_{u_1,e} = -27$ Choose one of these (a, b here).	$Q_{u_2,u_1} = -36$ $Q_{u_2,e} = -36$ $Q_{u_1,e} = -36$ Choose one of these (u_1, e here).	
Step 4: Creating node U that joins smallest pair i and j	u_1 joins c and d $S_{c,u_1} = D_{c,d}/2 + (S_c - S_d)/2 = 3$ $S_{d,u_1} = D_{c,d}/2 + (S_d - S_c)/2 = 6$	u_2 joins a and b $S_{a,u_2} = D_{a,b}/2 + (S_a - S_b)/2 = 3$ $S_{b,u_2} = D_{a,b}/2 + (S_b - S_a)/2 = 3$	u_3 joins u_1 and e $S_{u_1,u_3} = D_{u_1,e}/2 + (S_{u_1} - S_e)/2 = 9$ $S_{e,u_3} = D_{u_1,e}/2 + (S_e - S_{u_1})/2 = 6$	For last pair, connect u_2 and u_3 with branch length$= 3$
Step 5: Joining i and j and make all other taxa in form of a star.				

Figure 2.8 Example of NJ tree construction

Gascuel and Steel [31] gives a review to provide an answer for "What does neighbor-joining do?". A deeper and more computational exploration of these connections is offered in [23].

Among other popular distance-based methods include ADDTREE [52], BioNJ [28], Unweighted Neighbor-Joining (UNJ) [29], and Minimum Evolution [50].

2.4 PHYLOGENETIC NETWORKS

A phylogenetic network is a generalisation of a phylogenetic tree, which can be used to describe the evolutionary history of a set of species that is non-tree like because of reticulation events, such as hybridization, recombination, and horizontal gene transfer. Since the rest of the thesis focuses on networks, rest of this chapter concerns networks.

A digraph is *acyclic* (a *DAG*) if it has no directed cycles. An acyclic digraph is *rooted* if there exists a distinguished vertex ρ, called the *root*, such that ρ has in degree 0, $d^-(\rho) = 0$, and there exists a directed path from ρ to every vertex of D. Let us observe that, except for ρ, no other vertex has in-degree zero.

A rooted binary phylogenetic network $\mathcal{N} = (V, E)$ on a finite non-empty set X is a rooted, connected, directed acyclic graph with the following properties:

1. exactly one vertex (the *root*) has in-degree 0 and all other vertices have in-degree 1 or 2,

2. any vertex with in-degree 2 (called a *reticulation vertex*) has out-degree 1 and all other vertices have out-degree 0 or 2, and

3. each vertex with out-degree 0 (a leaf) is labelled with a distinct element of X.

4. each vertex with out-degree 2 and in-degree 1 is a *tree vertex*.

An *unrooted binary phylogenetic network* $\mathcal{N}$ on a set X is a graph G containing only vertices of degree three or one. An *unrooted binary phylogenetic* X*-tree* (or *unrooted binary phylogenetic tree*) on X is an unrooted binary phylogenetic network on X that is connected and acyclic (a tree).
We denote the set of leaf labels associated to a rooted binary phylogenetic network $\mathcal{N}$ by $\mathcal{L}(\mathcal{N})$ (note that $X = \mathcal{L}(\mathcal{N})$).

A vertex $v \in V$ is a *child* of $u \in V$ if an edge $(u, v) \in E$; we also say that u is a *parent* of v. An *immediate reticulation* v is a reticulation vertex if there are vertices u, w such that $(u, v), (w, v)$ and (u, w) are all arcs.

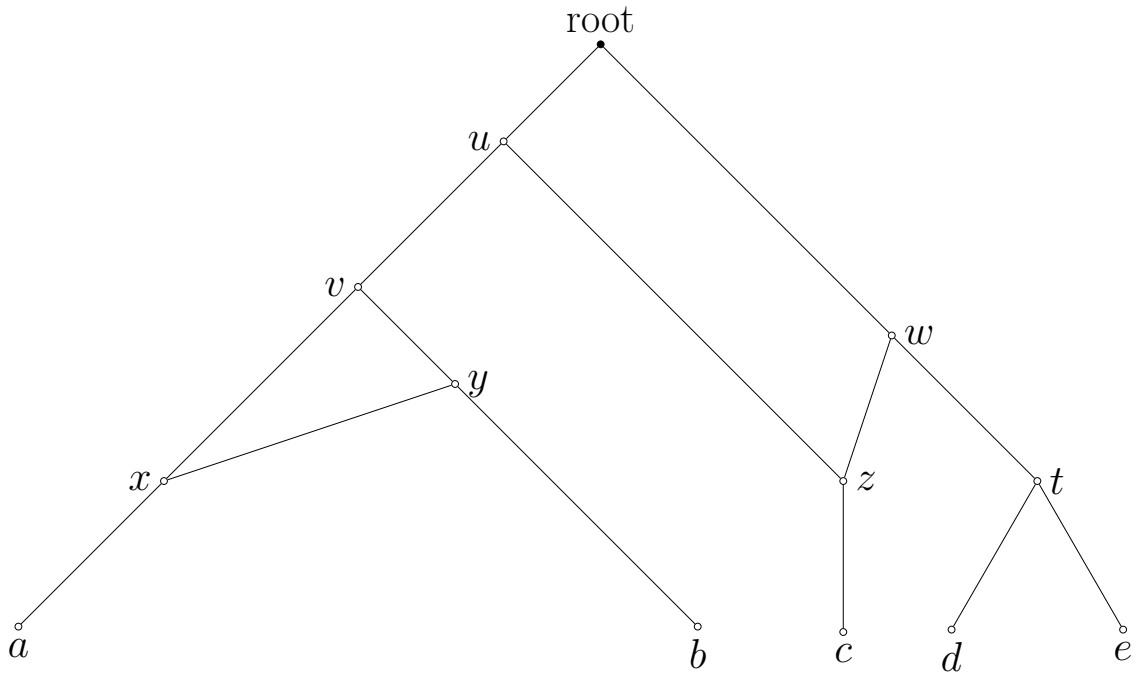


Figure 2.9 A rooted binary phylogenetic network.

For vertices u, v such that there is a directed path from u to v in $\mathcal{N}$, we say the path is a *tree path* if every vertex on the path, except possibly u, is a tree vertex or a leaf. If (u, v) is an arc and v is a tree-vertex of a leaf, we say (u, v) is a *tree edge*.

In Figure 2.9, the vertex $v \in V$ is a child of u, i.e. u is a parent of v. The vertex x is an immediate reticulation since its parents v and y are parent and child. z is a reticulation vertex with parents u and w. There is a tree path from root to e, since w, t are tree vertices. The arc (w, t) is a tree edge, since both w and t are tree vertices.

Note that, a *rooted binary phylogenetic X-tree $\mathcal{T}$* is a binary rooted phylogenetic network on X with no vertex with in-degree 2. Often drawn with without arrows as arcs are assumed to point down to the page.

The number of arcs we need to remove from a rooted phylogenetic network $\mathcal{N}$ on X to obtain a rooted binary phylogenetic tree on X is denoted by $h(\mathcal{N})$ and referred to as the *hybridization number* of $\mathcal{N}$, see Chapter 3 for details. Note that, when focusing on rooted binary phylogenetic networks, $h(\mathcal{N})$ coincides with the number of reticulation vertices in $\mathcal{N}$. A *cut vertex (cut arc)* is a vertex (an arc) whose removal disconnects the graph. A *biconnected component* is a maximal connected subgraph that does not contain a cut vertex. The maximum $h(B)$ in any biconnected component B of $\mathcal{N}$ is called the *level* of the phylogenetic network $\mathcal{N}$. For all vertices v of $\mathcal{N}$, let $c(v)$ denote the subset of X consisting

of the elements x for which there is a directed path in $\mathcal{N}$ from v to $\phi(x)$. We call $c(v)$ the *cluster* corresponding to v. A subset C of X is a *cluster* of $\mathcal{N}$ if there is some vertex v of $\mathcal{N}$ such that $C = c(v)$ and C is non-trivial if $C \neq X$ and $|C| > 1$.

Let $\mathcal{T}$ be a rooted binary phylogenetic X-tree with root ρ. We define the size of the tree $\mathcal{T}$ to be $|\mathcal{T}| := |X|$ and abbreviate $n := |X|$. Let P be a subset of leaves of $\mathcal{T}$. We denote the minimal rooted subtree of $\mathcal{T}$ that connects the leaves of P by $\mathcal{T}(P)$. The root of $\mathcal{T}(P)$ is the unique degree-two vertex of $\mathcal{T}(P)$ that is closest to the root of $\mathcal{T}$ in $\mathcal{T}$. Furthermore, the *restriction* of $\mathcal{T}$ to P (denoted $T|P$) is the rooted binary phylogenetic tree that is obtained from $\mathcal{T}(P)$ by suppressing all non-root vertices of degree two. For a non-trivial cluster C corresponding to a vertex v of $\mathcal{T}$, we define the *contraction* of $\mathcal{T}$ with respect to C (denoted by $T{\downarrow_C}$) as the result of contracting the subgraph rooted at v in $\mathcal{T}$ onto v, removing all labels of C from X, and giving v a new label (we use the label a_C unless otherwise specified). *Cutting an arc* (u, v) of $\mathcal{T}$ means deleting the arc (u, v) from $\mathcal{T}$, producing disconnected subtrees $\mathcal{T}_u$ and $\mathcal{T}_v$, containing u and v, respectively, and then suppressing u if it has degree two in $\mathcal{T}_u$.

Figure 2.10 shows a rooted phylogenetic network $\mathcal{N}$ with one hybridization vertex u. In $\mathcal{N}$, cutting the arc (w, u), and suppressing the degree two vertices w and u produces a rooted binary phylogenetic subtree T_v; similarly, cutting the arc (v, u), and suppressing the degree two vertices v and u produces a rooted binary phylogenetic subtree T_w.

A rooted phylogenetic network is a *level-k phylogenetic network* if each biconnected component contains at most $k \geq 0$ hybridization vertices. A level-0 phylogenetic network is a phylogenetic tree, and a level-1 network is commonly called a galled tree. General level-k networks were first introduced by Choy, Jansson, Sadakane and Sung [21], and are discussed in [55]. In Figure 2.11, $\mathcal{N}_1$ is an example of a level-1 network with u the hybridization vertex. $\mathcal{N}_2$ is an example of a level-2 network with u and v the hybridization vertices in one biconnected component, which is the most, as w is separated by the cut edges either side of the root.

A tree-child network is a phylogenetic network such that every internal vertex $v \in V$ has at least one child that is a tree vertex. The network $\mathcal{N}_1$ on the left in Figure 2.12 is not a tree-child network since there is a vertex x that has no tree vertex children. The network $\mathcal{N}_2$ on the right in Figure 2.12 is a tree-child network since each internal vertex has at least one child tree vertex. Note that in a tree-child network every vertex has a tree path to a leaf. Tree-child networks are discussed in [17]. We say an ultrametric network is *ultrametric tree-child network* if every non-leaf has a child which is either a tree vertex or a leaf.

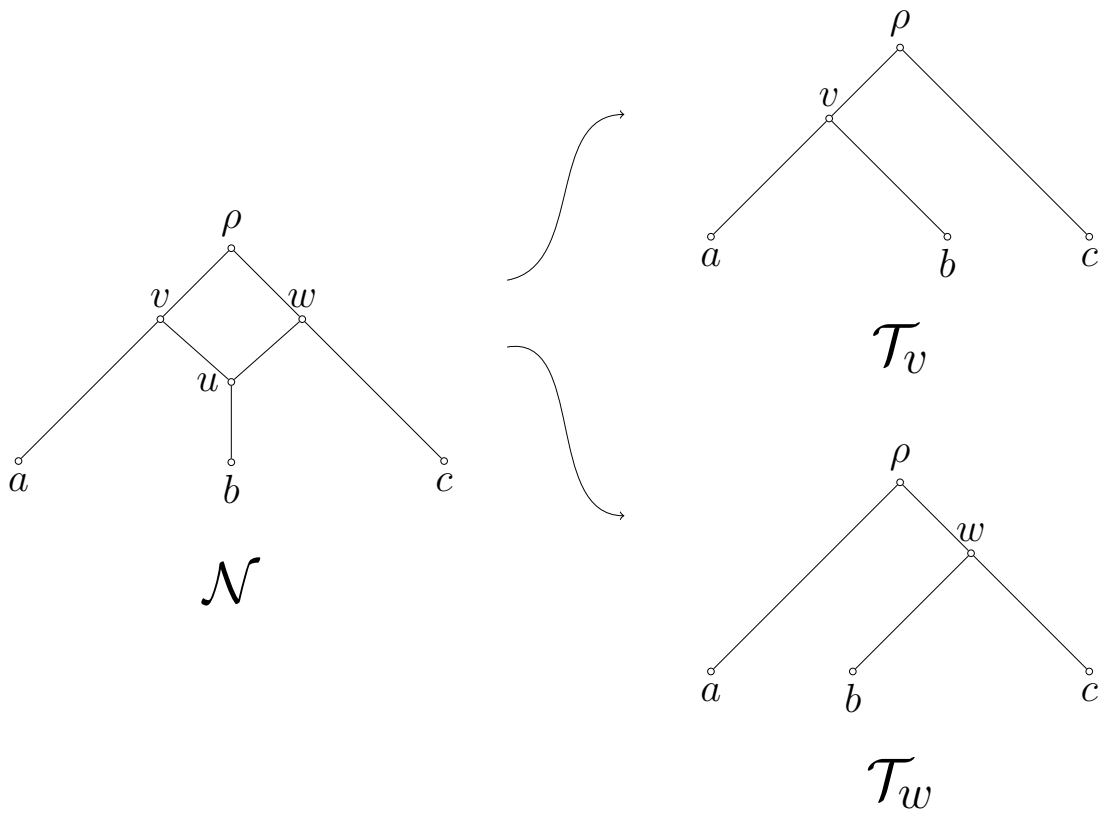


Figure 2.10 A rooted phylogenetic network $\mathcal{N}$ with three leaves and one hybridization vertex. T_v and T_w are rooted binary phylogenetic trees obtained from a phylogenetic network $\mathcal{N}$.

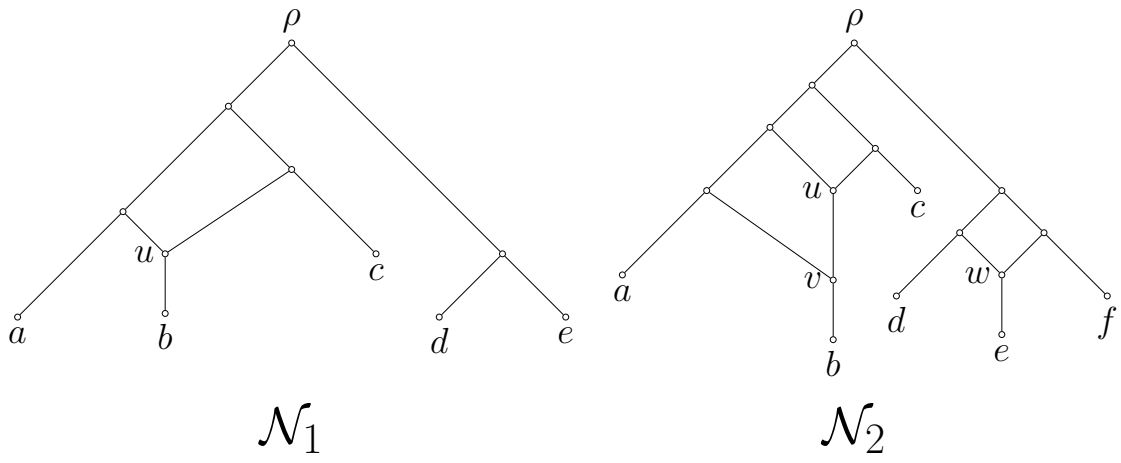


Figure 2.11 $\mathcal{N}_1$ is a level-1 network and $\mathcal{N}_2$ is a level-2 network.

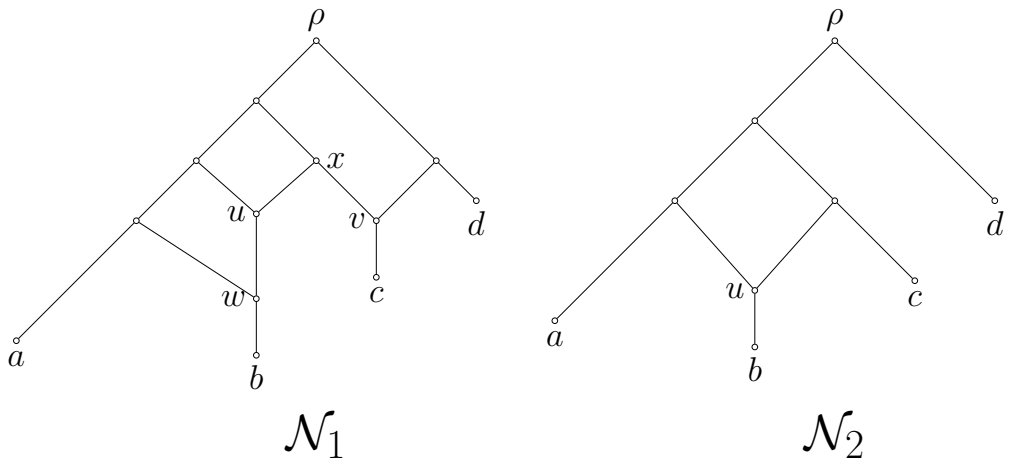


Figure 2.12 A network $\mathcal{N}_1$ that is not tree-child, and a tree-child network $\mathcal{N}_2$.

A network $\mathcal{N}$ is *weighted* if there is a non-zero weighting (or length) associated with each arc, which is strictly non-zero for all tree arcs (those arcs whose head is a tree vertex or leaf). For arc $e = (u, v)$ the weight is denoted by l_e or $l(u, v)$. The *weight of a path* is the sum of the weights of arcs it contains.

An *ultrametric network* is a weighted phylogenetic network such that every directed path from the root to any leaf has the same weight [2, 18]. This implies that for any vertices u, v such that there is a directed path from u to v in $\mathcal{N}$, every path from u to v has the same weight, which we denote $d_{u,v}$. It is observed that even if $\mathcal{N}$ is ultrametric, there can be more than one evolutionary path between x and y, and moreover, these paths may have different lengths. In Figure 2.13, all of the leaves in X have same distance from the root, and there are two different up-down paths between b and c; one path include vertices $\{b, u', v, v', c\}$ which has length 6, and another path include vertices $\{b, u', u, \rho, w, v, v', c\}$ which has length 10.

Given a phylogenetic network $\mathcal{N}$ on X, we define the *set-distance matrix $\mathcal{D}$ of inter-taxa distances* as follows. For any two elements $x, y \in X$, an up-down path from x to y is an underlying path $x, v_1, v_2, ..., v_{k-1}, y$ in $\mathcal{N}$ such that, for some $i \leq k - 1$, $\mathcal{N}$ contains the arcs

$$(v_i, v_{i-1}), (v_{i-1}, v_{i-2}), ..., (v_1, x)$$

and

$$(v_i, v_{i+1}), (v_{i+1}, v_{i+2}), ..., (v_{k-1}, y).$$

The weight of an up-down path is the sum of the weights of the two directed paths it contains. For example, in Figure 2.13, b, u', v, v', x, d is an up-down path in $\mathcal{N}$ from b to d.

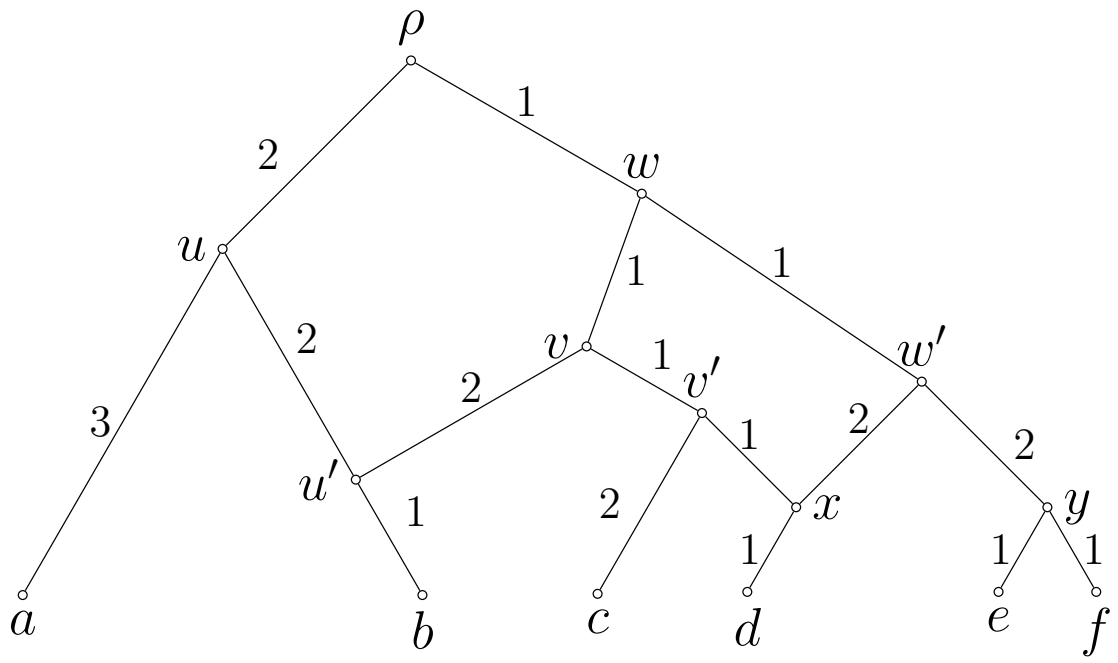


Figure 2.13 An ultrametric network.

The vertex v_i is called the peak of the up-down path. In any rooted network $\mathcal{N}$, a *least common ancestor* of two vertices x and y is a vertex v such that there is an up-down path from x to y with v the peak of the path. By this definition there might be multiple least common ancestors for x and y illustrated in Figure 2.13. However for each, the paths v to x and v to y are arc-disjoint, so there could be some genetic inheritance from the root of the network to x and y that has a common path as far as v and then diverges.

Now let $\mathcal{P}_{x,y}$ be the set of distinct up-down paths from x to y in $\mathcal{N}$. The set of distances between x and y, denoted $\mathcal{D}_{x,y}$, is the set of path weights in $\mathcal{P}_{x,y}$; similarly, the multiset of distances between x and y is the multiset of path weights in $\mathcal{P}_{x,y}$. The distance $d_{x,y}$ denotes the minimum weight in $\mathcal{D}_{x,y}$. The *set-distance matrix* $\mathcal{D}$ of $\mathcal{N}$ is the $|X|$ by $|X|$ matrix whose (x, y) entry is $\mathcal{D}_{x,y}$. If $\mathcal{D}$ is the set-distance matrix of $\mathcal{N}$, we say $\mathcal{N}$ *displays* $\mathcal{D}$.

The ultrametric network in Figure 2.13 displays the multiset distance matrix $\mathcal{D}_1$ and set distance matrix $\mathcal{D}_2$ below. It is easily checked that the multiset of distances between a and d is $\{10, 10\}$ and the set of distance between a and b is $\{10\}$.

$$\mathcal{D}_1 = \begin{bmatrix} & a & b & c & d & e & f \\ a & \{0\} & \{6,10\} & \{10\} & \{10,10\} & \{10\} & \{10\} \\ b & & \{0\} & \{6,10\} & \{6,10\} & \{10\} & \{10\} \\ c & & & \{0\} & \{4,8\} & \{8\} & \{8\} \\ d & & & & \{0\} & \{6,8\} & \{6,8\} \\ e & & & & & \{0\} & \{2\} \\ f & & & & & & \{0\} \end{bmatrix}$$

$$\mathcal{D}_2 = \begin{bmatrix} & a & b & c & d & e & f \\ a & \{0\} & \{6,10\} & \{10\} & \{10\} & \{10\} & \{10\} \\ b & & \{0\} & \{6,10\} & \{6,10\} & \{10\} & \{10\} \\ c & & & \{0\} & \{4,8\} & \{8\} & \{8\} \\ d & & & & \{0\} & \{6,8\} & \{6,8\} \\ e & & & & & \{0\} & \{2\} \\ f & & & & & & \{0\} \end{bmatrix}$$

Note that the set-distance matrix is really a 2-dimensional array of sets of distances, not a matrix in the mathematical sense. However we use the terminology to emphasise that set-distance matrices are an extension of the distance matrices widely used in phylogenetics.

In this thesis, our focus is the task of reconstructing phylogenetic networks, rather than phylogenetic trees, from information about inter-taxa distances. In a recent paper, Bordewich and Semple [13] showed that determining the topological structure of binary tree-child phylogenetic network given multiset distance matrix is possible in polynomial time in the size of the input.

Theorem 2.4.1. *[13] Let $\mathcal{D}$ be a multiset-matrix of distances between elements of a set X. If there is a binary tree-child network $\mathcal{N}$ on X displaying $\mathcal{D}$, with no arc joining the two children of the root then, up to isomorphism, $\mathcal{N}$ is the unique binary phylogenetic network on X realising $\mathcal{D}$, in which case $\mathcal{N}$ can be found in time quadratic in $|\mathcal{D}|$.*

Related work on reconstructing phylogenetic networks from inter-taxa distances has been done by Willson [64, 65]. In Willson [65], sufficient conditions are given for when a network without weights itself may be inferred from the average inter-taxa genetic distances, and these conditions are shown to be satisfied whenever the distances arise from a normal network with a single reticulation cycle. Hence Willson deals with a more complex and general case and so achieves more restricted results (handling a single reticulation, rather than all tree-child networks).

3

ON THE FIXED PARAMETER TRACTABILITY OF AGREEMENT-BASED PHYLOGENETIC DISTANCES

In this chapter we consider three important problems in phylogenetics: HYBRIDIZATION NUMBER, RSPR DISTANCE and TBR DISTANCE. The underlying question is to determine how much reticulation is required to explain the evolution of a given set of taxa: given a collection of rooted phylogenetic trees on a set of taxa that correctly represent the tree-like evolution of different parts of their genomes, what is the smallest number of reticulation vertices needed to display the trees within a single phylogenetic network (the HYBRIDIZATION NUMBER problem)?

This question, along with the closely related problems of determining the minimum number of subtree prune and regraft, respectively tree bisection and reconnection, operations required to transform one rooted phylogenetic tree into another (the RSPR DISTANCE and TBR DISTANCE problem, respectively) has been considered in a number of papers [1, 4, 5, 11, 35, 37, 43, 47]. Key theoretical developments have shown that each of these three problems is NP-hard even in the restricted case that the input consists of two binary phylogenetic trees [10, 12, 37]. Moreover, HYBRIDIZATION NUMBER, RSPR DISTANCE and TBR DISTANCE problems are all fixed-parameter tractable where the parameter is hybridization number, rSPR distance and TBR distance, respectively [1, 10, 11].

In essence, this means that there are efficient algorithms for computing the hybridization number and the rSPR/TBR distance on two trees of large size, as long as there have not been too many reticulations in the evolutionary history of the considered taxa.

Two rules are used to reduce the size of an instance without changing the rSPR/TBR distance, i.e. to kernalise the problem [1]. The *Rule 1* is *subtree reduction* and the *Rule 2* is *chain reduction*. Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be a pair of weighted phylogenetic trees on X. Then *Rule 1* is replacing any pendant subtree that occurs identically in both trees by a single leaf with a

new label. Figure 3.1 illustrates Rule 1. Here B is a pendant subtree which occurs in both trees and is replaced with a new single leaf with a label w. *Rule 2* is redefined in [10] to preserve the rSPR distance as replacing any chain of pendant subtrees that occurs identically and with the same orientation relative to the root in both trees by three new leaves with new labels correctly oriented to preserve the direction of the chain. For both rules, the position of attachment of each pendant subtree must be the same in the two trees. Figure 3.1 illustrates Rule 2, where $A_1, A_2, A_3, \ldots, A_n$ is a chain of pendant subtrees that occurs in both trees and is replaced with three new single leaves with labels x, y, z.

Lemma 3.0.2 ([10]). *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two rooted phylogenetic X-trees. Let $\mathcal{T}_1'$ and $\mathcal{T}_2'$ be two rooted phylogenetic X-trees, i.e. X is set of leaves, obtained from $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively, by applying either Rule 1 or Rule 2. Then $d_{rSPR}(\mathcal{T}_1, \mathcal{T}_2) = d_{rSPR}(\mathcal{T}_1', \mathcal{T}_2')$.*

Lemma 3.0.2 says that the tree reduction Rules 1 and 2 preserve rSPR distance, if these rules repeatedly applied until the label set of the resulting rooted binary phylogenetic trees has size linear in the rSPR distance between them.

Lemma 3.0.3 ([10]). *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two rooted phylogenetic X-trees. Let $\mathcal{T}_1'$ and $\mathcal{T}_2'$ be two rooted phylogenetic X-trees obtained from $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively, by applying Rule 1 or Rule 2 repeatedly until no further reduction is possible. Then $|X'| \leq 28 \cdot d_{rSPR}(\mathcal{T}_1, \mathcal{T}_2)$, i.e. X' is the new set of leaves after the reduction rules are applied.*

Another approach for reducing the size of the instance (by reducing common clusters) is cluster reduction [5]. Definition of rooted cluster reduction is following [6].

Definition 3.0.4 (rooted cluster reduction). *Let $\mathcal{T}$ and $\mathcal{T}'$ be rooted binary phylogenetic X-trees and let C be a non-trivial cluster common to both $\mathcal{T}$ and $\mathcal{T}'$. A* cluster reduction *is the operation of splitting $(\mathcal{T}, \mathcal{T}')$ into the two pairs of smaller trees $(\mathcal{T}_C, \mathcal{T}_C'), (\mathcal{T}_\rho, \mathcal{T}_\rho') := (\mathcal{T}|C, \mathcal{T}'|C), (\mathcal{T}{\downarrow_C}, \mathcal{T}'{\downarrow_C})$. Note that $(\mathcal{T}_C, \mathcal{T}_C')$ is a pair of phylogenetic C-trees, and $(\mathcal{T}_\rho, \mathcal{T}_\rho')$ is a pair of phylogenetic $((X \setminus C) \cup \{a_C\})$-trees that contain the original roots of $\mathcal{T}$ and $\mathcal{T}'$ respectively. See Figure 3.2 for an example.*

We now define a cluster sequence, which is essentially the result of applying several cluster reductions to a pair of trees. Let $\mathcal{T}$ and $\mathcal{T}'$ be rooted binary phylogenetic X-trees. Set $\hat{\mathcal{T}}_0 = \mathcal{T}$ and $\hat{\mathcal{T}}_0' = \mathcal{T}'$. For a cluster sequence consisting of t reductions, for $i = 1, \ldots, t$ let A_i be a non-trivial cluster common to both $\hat{\mathcal{T}}_{i-1}$ and $\hat{\mathcal{T}}_{i-1}'$, and define $\mathcal{T}_i := \hat{\mathcal{T}}_{i-1}|A_i$ and $\mathcal{T}_i' := \hat{\mathcal{T}}'|A_i$, and also $\hat{\mathcal{T}}_i := \hat{\mathcal{T}}_{i-1}{\downarrow_{A_i}}$ and $\hat{\mathcal{T}}_i' := \hat{\mathcal{T}}_{i-1}'{\downarrow_{A_i}}$, where the newly created leaf in $\hat{\mathcal{T}}_i$ and $\hat{\mathcal{T}}_i'$ is labelled by a_i. Finally, we denote $(\hat{\mathcal{T}}_t, \hat{\mathcal{T}}_t')$ as $(\mathcal{T}_\rho, \mathcal{T}_\rho')$, to emphasize that these

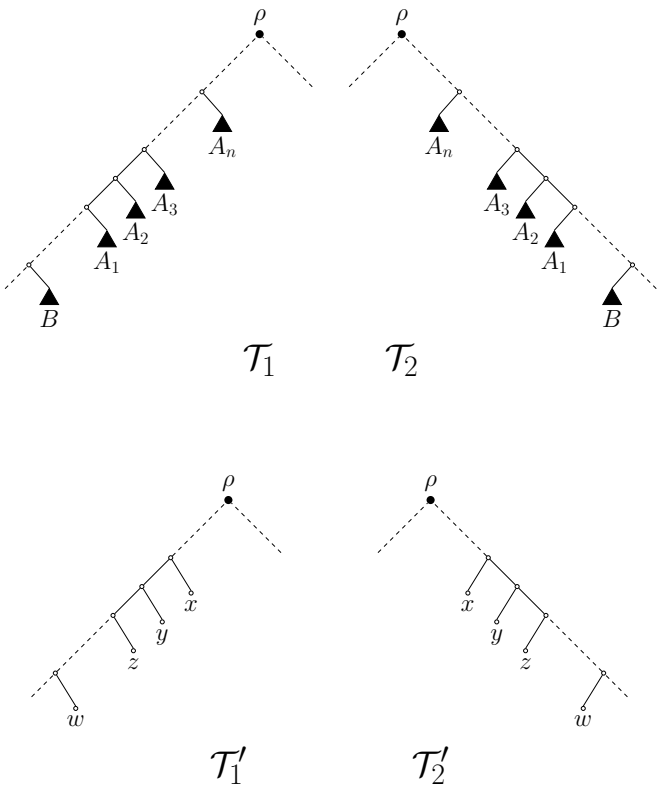


Figure 3.1 Two rooted binary phylogenetic trees $\mathcal{T}_1$ and $\mathcal{T}_2$ reduced under Rule 1 and Rule 2, where $\mathcal{T}_1'$ and $\mathcal{T}_2'$ are the resulting trees.

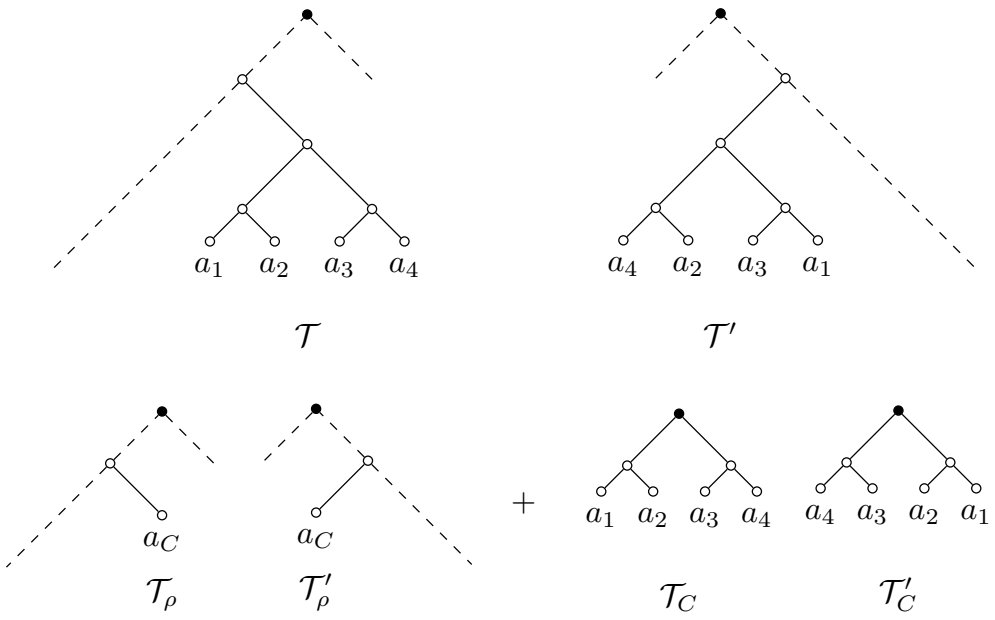


Figure 3.2 An example of the rooted cluster reduction. Black vertices are the respective roots.

two trees contain the original roots of $\mathcal{T}$ and $\mathcal{T}'$. The result is a sequence of pairs of trees $(\mathcal{T}_1, \mathcal{T}'_1), \ldots, (\mathcal{T}_t, \mathcal{T}'_t), (\mathcal{T}_\rho, \mathcal{T}'_\rho)$ which we call a *cluster sequence*. Note that the leaf set of $\mathcal{T}_i$ and $\mathcal{T}'_i$ is A_i and the leaf set of $\mathcal{T}_\rho$ and $\mathcal{T}'_\rho$ is $(X \cup \bigcup_i \{a_i\}) \setminus \bigcup_i A_i$.

We say a cluster sequence is a *full* cluster reduction of $\mathcal{T}$ and $\mathcal{T}'$ if at each step the cluster A_i is a minimal non-trivial common cluster and the trees $\mathcal{T}_\rho$ and $\mathcal{T}'_\rho$ contain no further non-trivial common clusters. Observe that the full cluster reduction is unique, up to the ordering of pairs, since any non-trivial common cluster of $\mathcal{T}$ and $\mathcal{T}'$ will at some point become minimal (once all common subclusters have been reduced), and it will then itself be reduced. In addition, no pair $(\mathcal{T}_i, \mathcal{T}'_i)$ in a full cluster reduction contains a non-trivial common cluster.

The subtree reduction operation does not seem to help the algorithms much in practice. On the other hand, the *cluster reduction*, which did not crop up in the theoretical analyses, greatly speeds up the algorithms in practice. The cluster reduction for HYBRIDIZATION NUMBER has been included in algorithms since the first parameterized algorithms appeared [6], and recent work has shown the applicability of an equivalent cluster reduction for RSPR DISTANCE [42].

Proofs in Section 3.2 and 3.3 for rooted binary phylogenetic networks/trees. These proofs can be easily adapted to the unrooted framework by disregarding the root and considering the graph as undirected. In order to define the required parameter, the *TBR distance* of two unrooted binary phylogenetic X-trees, we need to define a unrooted cluster reduction.

In the unrooted framework, we will use the word *edge* instead of *arc*. Note that each edge e of any phylogenetic X-tree uniquely partitions X into nonempty sets C and $\overline{C} := X \setminus C$ such that all paths between a leaf labelled with an element of C and a leaf labelled with an element of $\overline{C}$ contain e. A set C for which such an edge exists in $\mathcal{T}$ is called a *cluster* of $\mathcal{T}$. A cluster is called *trivial* if $|C| = 1$ or $|\overline{C}| = 1$. Given an unrooted binary phylogenetic X-tree $\mathcal{T}$ and a nontrivial cluster C of $\mathcal{T}$, let $\mathcal{T}|C$ denote the minimal subtree of $\mathcal{T}$ containing each leaf whose label is in C (analogous to the rooted case) and denote by $\mathcal{T}\downarrow_C$ the unrooted phylogenetic tree where $\mathcal{T}|C$ has been replaced by a leaf labelled by a_C.

Definition 3.0.5. *A rooted agreement forest for $\mathcal{T}$ and $\mathcal{T}'$ is a leaf-labelled forest $\mathcal{F}$ that can be obtained from $\mathcal{T}$ and $\mathcal{T}'$, respectively, by a series of edge deletions, deletions of unlabeled leaves, and suppressions of degree-two vertices. Figure 3.3 gives an example of agreement forests $\mathcal{F}_1$ and $\mathcal{F}_2$ for two rooted binary phylogenetic trees $\mathcal{T}$ and $\mathcal{T}'$.*

Definition 3.0.6. *An unrooted agreement forest (uAF) for two unrooted phylogenetic X-trees is the unrooted version of a rooted agreement forest. A uAF of minimal cardinality is called an unrooted* maximum-*agreement forest (uMAF). $\mathcal{F}$ is said to* isolate *some $x \in X$ if $\mathcal{F}$*

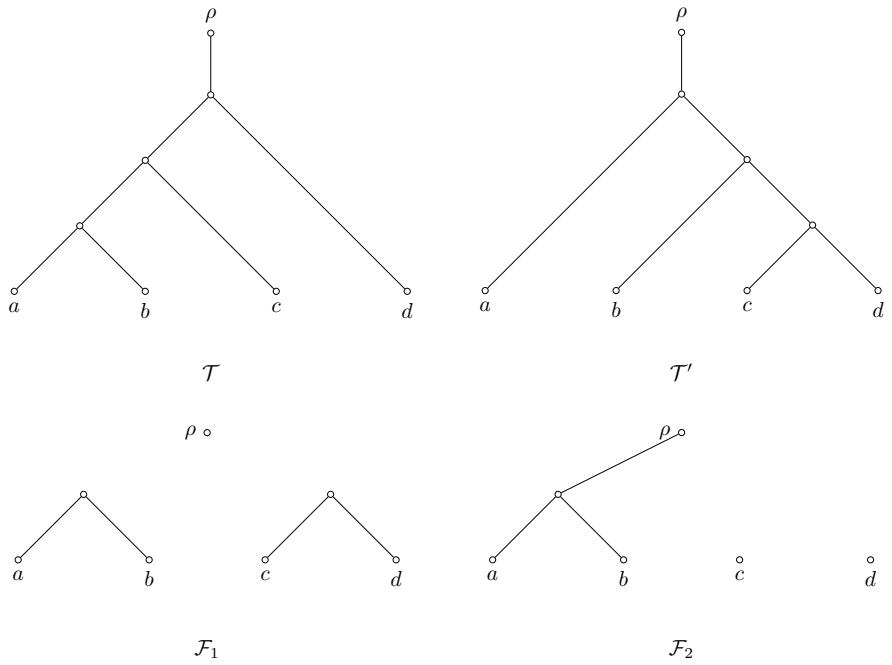


Figure 3.3 Agreement forests $\mathcal{F}_1$ and $\mathcal{F}_2$ for two rooted binary phylogenetic trees $\mathcal{T}$ and $\mathcal{T}'$.

contains a singleton tree consisting of the leaf labelled x (denoted by $\{x\} \in \mathcal{F}$). Finally, we denote the number of trees in $\mathcal{F}$ by $|\mathcal{F}|$.

In the following, we describe a cluster reduction for unrooted binary phylogenetic trees, slightly different from the rooted case.

Definition 3.0.7 (unrooted cluster reduction)**.** *Let $\mathcal{T}$ and $\mathcal{T}'$ be unrooted binary phylogenetic trees and let C be a non-trivial cluster common to both $\mathcal{T}$ and $\mathcal{T}'$ (note that $\overline{C}$ is also a common cluster of $\mathcal{T}$ and $\mathcal{T}'$). A* cluster reduction *is the operation of splitting $(\mathcal{T}, \mathcal{T}')$ into the two pairs of smaller trees $(\mathcal{T}_C, \mathcal{T}'_C), (\mathcal{T}_{\overline{C}}, \mathcal{T}'_{\overline{C}}) := (\mathcal{T}{\downarrow_C}, \mathcal{T}'{\downarrow_C}), (\mathcal{T}{\downarrow_{\overline{C}}}, \mathcal{T}'{\downarrow_{\overline{C}}})$. See Figure 3.4 for an example.*

Analogously to the rooted case, we call the result $(\mathcal{T}_1, \mathcal{T}'_1), \ldots, (\mathcal{T}_t, \mathcal{T}'_t)$ of repeatedly applying the cluster reduction to two unrooted binary phylogenetic trees $\mathcal{T}$ and $\mathcal{T}'$ a *cluster sequence* for $\mathcal{T}$ and $\mathcal{T}'$ and such a sequence is called *full* if each cluster reduction leading to the sequence reduces a *minimal* non-trivial common cluster and the trees $\mathcal{T}_t$ and $\mathcal{T}'_t$ contain no further non-trivial common clusters. Again, the full cluster reduction is unique, up to the ordering of pairs and no pair $(\mathcal{T}_i, \mathcal{T}'_i)$ in the full cluster reduction contains a non-trivial common cluster.

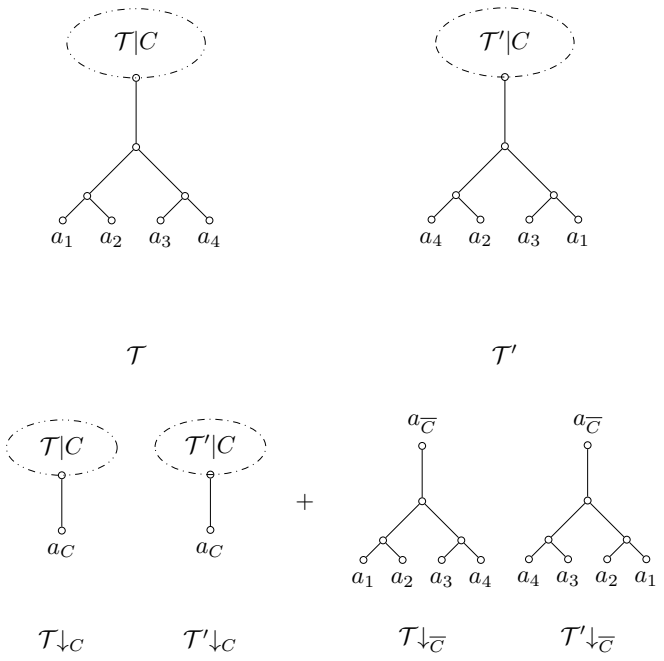


Figure 3.4 An example of an unrooted cluster reduction. The common cluster is $C = \{a_1, a_2, a_3, a_4\}$.

Here, we give a theoretical justification of why the cluster reduction for HYBRIDIZATION NUMBER is so useful in practice by showing that the divide-and-conquer approach that follows from it implies fixed-parameter tractability where the parameter is not the *total number of reticulations* in the optimal network displaying the two input trees, but instead the *maximum number of reticulations seen in any biconnected component* of such a network. This concept has been studied before as the *level* of the network (see for example [39, 58]). In essence, this means that for large input trees, even when there have been many reticulations, as long as not too many of the reticulations are entangled with each other, the problem may still be solved efficiently. This is what is expected to happen for real biological data, in part because reticulation events such as hybridization events are less likely to happen between genetically-distant species.

We show something stronger: the use of the cluster reduction can transform any $O(f(p) \cdot n)$-time algorithm for any of the considered problems into an $O(f(k) \cdot n)$-time algorithm, where n is the number of leaves of the phylogenetic trees. Also, p is a natural parameter, such as the rSRR distance between two trees, and k is the minimum level of a phylogenetic network displaying both trees, which is a much stronger (that is, smaller) parameter than p.

The fact that the cluster reduction implies fixed-parameter tractability in the level for HYBRIDIZATION NUMBER was already implicitly present in [56, 40]. Still, we think that it is worth proving explicitly and formally, and extending the reasoning to RSPR DISTANCE and TBR DISTANCE, thus giving hard evidence for the importance of implementing the cluster reduction in any software.

In the next section, we present the definition of problems HYBRIDIZATION NUMBER, RSPR DISTANCE and TBR DISTANCE, prove their fixed parameter tractability with respect to level in Section 3.2, 3.3 and 3.4, respectively.

3.1 DEFINITION OF PROBLEMS

Basic definitions of tree rearrangement operations (SPR, TBR and NNI) and level of network are given in Chapter 2. In this section, we give the formal definition of HYBRIDIZATION NUMBER, RSPR DISTANCE and TBR DISTANCE problems.

3.1.1 THE HYBRIDIZATION NUMBER PROBLEM

Let $\mathcal{T}$ be a rooted binary phylogenetic X-tree and let $\mathcal{N} = (D, \phi)$ be a rooted phylogenetic network on X. We say that $\mathcal{N}$ *displays* $\mathcal{T}$ means $\mathcal{T}$ can be obtained from $\mathcal{N}$ by first deleting a subset of the arcs of D and then deleting isolated vertices and suppressing the non-root degree-two vertices. For two rooted binary phylogenetic X-trees, $\mathcal{T}$ and $\mathcal{T}'$, we define the *hybridization number* of $\mathcal{T}$ and $\mathcal{T}'$ as

$$h(\mathcal{T}, \mathcal{T}') := \min\{h(\mathcal{N}) \mid \mathcal{N} \text{ displays } \mathcal{T} \text{ and } \mathcal{T}'\}.$$

We also define the *hybridization level* of $\mathcal{T}$ and $\mathcal{T}'$ as the minimum k such that there is a level-k rooted phylogenetic network, i.e. a rooted phylogenetic network with level k, that displays $\mathcal{T}$ and $\mathcal{T}'$. The decision problem, HYBRIDIZATION NUMBER, is formally stated as follows.

Problem: HYBRIDIZATION NUMBER

Input: Two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$, and $l \in \mathbb{N}$.

Question: Is $h(\mathcal{T}, \mathcal{T}') \leq l$?

3.1.2 THE RSPR PROBLEM

Let $\mathcal{T}$ be a rooted binary phylogenetic X-tree. For the upcoming definition of a rooted subtree prune and regraft operation, we regard the root of $\mathcal{T}$ as a vertex labelled by a dummy

taxon l_ρ at the end of a pendant arc adjoined to the original root (for details see [10]. This is done to be able to *regraft* above the original root). Recall Definition 2.2.2 in Chapter 2 for rooted binary phylogenetic tree $\mathcal{T}$, now let $e = (u, v)$ be an arc of $\mathcal{T}$ not incident with the vertex labelled l_ρ, and the same SPR operation applies to a rooted binary phylogenetic tree $\mathcal{T}$. We say that $\mathcal{T}'$ is obtained from $\mathcal{T}$ by one *r*ooted subtree prune and regraft (rSPR) operation. We define the *r*SPR distance between two rooted binary phylogenetic X-trees $\mathcal{T}_1$ and $\mathcal{T}_2$ to be the minimum number of rSPR operations that are required to transform $\mathcal{T}_1$ into $\mathcal{T}_2$. We denote this distance by $d_{\mathrm{rSPR}}(\mathcal{T}_1, \mathcal{T}_2)$. The associated decision problem is the following.

Problem: rSPR DISTANCE

Input: Two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$ and $l \in \mathbb{N}$.

Question: Is $d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}') \leq l$?

3.1.3 THE TBR PROBLEM

The definition of TBR operation is already defined in Chapter 2 as a Definition 2.2.1.

For two unrooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$, the *TBR distance* is the maximum TBR distance between a pair of trees in a unrooted full cluster reduction of $\mathcal{T}$ and $\mathcal{T}'$, i.e. the maximum of $d_{\mathrm{TBR}}(\mathcal{T}_i, \mathcal{T}_i')$ over $i \in \{1, \ldots, t, \rho\}$. Note that the unrooted hybridization level is always smaller or equal to the TBR distance, since the unrooted hybridization number equals the TBR distance (see Theorem 3.4.2 in Section 3.4). The decision problem TBR DISTANCE is formally stated as follows.

Problem: TBR DISTANCE

Input: Two unrooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$ and $l \in \mathbb{N}$.

Question: Is $d_{\mathrm{TBR}}(\mathcal{T}, \mathcal{T}') \leq l$?

3.2 FIXED PARAMETER TRACTABILITY OF HYBRIDIZATION NUMBER

In this section we prove the fixed parameter tractability of hybridization number when parameterized by the hybridization level which is given in Subsection 3.2. It was already known that HYBRIDIZATION NUMBER is fixed-parameter tractable when parameterized by the hybridization number [11] but our result is stronger as the hybridization level can be small, even 1, for pairs of trees for which the hybridization number is arbitrarily large. On the other hand, it is clear that the hybridization level never exceeds the hybridization number.

First, the following lemma shows how the cluster reduction can be used as part of a divide-and-conquer approach for computing the hybridization number.

Lemma 3.2.1 ([5]). *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees. Suppose that $C \subset X$ is a cluster of both $\mathcal{T}$ and $\mathcal{T}'$, where $(\mathcal{T}_C, \mathcal{T}'_C)$and $(\mathcal{T}_\rho, \mathcal{T}'_\rho)$ are the results of performing a cluster reduction of C on $(\mathcal{T}, \mathcal{T}')$. Then,*

$$h(\mathcal{T}, \mathcal{T}') = h(\mathcal{T}_C, \mathcal{T}'_C) + h(\mathcal{T}_\rho, \mathcal{T}'_\rho).$$

A straightforward consequence of Lemma 3.2.1 is that if $(\mathcal{T}_1, \mathcal{T}'_1), \cdots, (\mathcal{T}_t, \mathcal{T}'_t), (\mathcal{T}_\rho, \mathcal{T}'_\rho)$ is a cluster sequence of $\mathcal{T}$ and $\mathcal{T}'$, then

$$h(\mathcal{T}, \mathcal{T}') = h(\mathcal{T}_1, \mathcal{T}'_1) + \cdots + h(\mathcal{T}_t, \mathcal{T}'_t) + h(\mathcal{T}_\rho, \mathcal{T}'_\rho).$$

Next, we show that the hybridization level of two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$ is equal to the maximum hybridization number between a pair of trees in a full cluster reduction of $\mathcal{T}$ and $\mathcal{T}'$. Recall that, for a rooted phylogenetic network $\mathcal{N}$, its level is the maximum number of reticulation vertices in any biconnected component of $\mathcal{N}$.

Lemma 3.2.2. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees and let*

$$(\mathcal{T}_1, \mathcal{T}'_1), \ldots, (\mathcal{T}_t, \mathcal{T}'_t), (\mathcal{T}_\rho, \mathcal{T}'_\rho)$$

be a full cluster reduction of $\mathcal{T}$ and $\mathcal{T}'$. Then, the hybridization level of $\mathcal{T}$ and $\mathcal{T}'$ equals

$$\max_{i \in \{1,\ldots,t,\rho\}} h(\mathcal{T}_i, \mathcal{T}'_i).$$

Proof: For each $i \in \{1, \ldots, t\}$, let $\mathcal{N}_i$ be a rooted phylogenetic network displaying $\mathcal{T}_i$ and $\mathcal{T}'_i$ with hybridization number $h(\mathcal{T}_i, \mathcal{T}'_i)$ and let A_i and a_i denote the set of leaves of $\mathcal{T}_i$ and the new leaf created to represent the cluster A_i in the i^{th} cluster reduction, respectively. We may now rebuild a rooted phylogenetic network $\mathcal{N}$ displaying $\mathcal{T}$ and $\mathcal{T}'$ from the smaller rooted phylogenetic networks $\mathcal{N}_i$ as follows. We start with $\mathcal{N} = \mathcal{N}_\rho$. While $\mathcal{N}$ contains a leaf v labelled a_i for some i, we replace v by a pendant copy of $\mathcal{N}_i$ in $\mathcal{N}$. Since each arc incident with such a leaf is a cut arc of the resulting rooted phylogenetic network $\mathcal{N}$, each biconnected component of $\mathcal{N}$ is a subnetwork of $\mathcal{N}_i$ for some $i \in \{1, \ldots, t, \rho\}$.

Thus, $\mathcal{N}$ displays $\mathcal{T}$ and $\mathcal{T}'$ and the level of $\mathcal{N}$ is at most the maximum of $h(\mathcal{T}_i, \mathcal{T}'_i)$ over $i \in \{1, \ldots, t, \rho\}$, hence the hybridization level of $\mathcal{T}$ and $\mathcal{T}'$ is at most the maximum of $h(\mathcal{T}_i, \mathcal{T}'_i)$ over $i \in \{1, \ldots, t, \rho\}$.

Conversely, let $\mathcal{N}$ be any rooted phylogenetic network displaying $\mathcal{T}$ and $\mathcal{T}'$ and let k denote its level. Let the vertex set of $\mathcal{N}$ be V and the root be ρ. We will construct a cluster sequence for $\mathcal{T}$ and $\mathcal{T}'$. Each cut arc (u, v) of $\mathcal{N}$ gives rise to a cluster $c(v)$ which is a common cluster to $\mathcal{T}$ and $\mathcal{T}'$. A cut arc (u, v) of $\mathcal{N}$ is trivial if v is a leaf of $\mathcal{N}$, and it is a minimal non-trivial cut arc if there is no other non-trivial cut arc (w, x) of $\mathcal{N}$ such that there is a directed path from v to w in $\mathcal{N}$. We obtain a cluster sequence for $\mathcal{T}$ and $\mathcal{T}'$ by iteratively:

- selecting v in V at the head of a minimal non-trivial cut arc of $\mathcal{N}$, which gives rise to $c(v)$, a minimal non-trivial common cluster of $\mathcal{T}$ and $\mathcal{T}'$;
- performing the cluster reduction of $\mathcal{T}$ and $\mathcal{T}'$ by $c(v)$ replacing the cluster with a new vertex c_v, and
- replacing the subnetwork below the cut edge with a single pendant leaf c_v in $\mathcal{N}$

Note that the deleted subnetwork is either a subtree (in fact, due to minimality, cherry or a biconnected component of $\mathcal{N}$ with pendant leaves, since otherwise, we could choose a smaller common cluster. Since the level of the network is k, this subnetwork of $\mathcal{N}$ is a phylogenetic network on $c(v)$ containing at most k hybridization vertices and displaying $\mathcal{T}|c(v)$ and $\mathcal{T}'|c(v)$. Hence the cluster pair in the cluster reduction has hybridization number at most k. We repeat this process until $\mathcal{N}$ has no further cut arcs, obtaining a cluster sequence $(\mathcal{T}_1, \mathcal{T}'_1), ..., (\mathcal{T}_t, \mathcal{T}'_t), (\mathcal{T}_\rho, \mathcal{T}'_\rho)$ for $\mathcal{T}$ and $\mathcal{T}'$. Every cluster pair $(\mathcal{T}_i, \mathcal{T}'_i)$ from the cluster sequence has hybridization number at most k. It remains to consider the final pair $(\mathcal{T}_\rho, \mathcal{T}'_\rho)$. Since in the end $\mathcal{N}$ had no (non-trivial) cut arcs, either $\mathcal{N}$ was reduced to a cherry or $\mathcal{N}$ was a biconnected component with pendant leaves, and again we deduce that $h(\mathcal{T}_\rho, \mathcal{T}'_\rho) \leq k$. Thus if $\mathcal{T}$ and $\mathcal{T}'$ can be displayed on a level-k phylogenetic network, then there is a cluster sequence for $\mathcal{T}$ and $\mathcal{T}'$ such that the maximum hybridization number between a pair of trees in the cluster reduction is at most k.

It remains to show that the maximum hybridization number between a pair of trees in the *full* cluster reduction is therefore also at most k. We will make use of the fact that if a cluster reduction is not a reduction by a minimal non-trivial common cluster, then it can be broken down into a series of cluster reductions each of which is by a minimal non-trivial common cluster. To see this consider a cluster reduction of $\mathcal{T}$ and $\mathcal{T}'$ by a common cluster A and suppose it is not a minimal non-trivial common cluster. Then, there is a subset $A_1 \subset A$ such that A_1 is a minimal non-trivial common cluster. We first reduce by A_1, obtaining $(\mathcal{T}_{A_1}, \mathcal{T}'_{A_1}), (\mathcal{T}_\rho, \mathcal{T}'_\rho)$, where there is a leaf a_1 in $\mathcal{T}_\rho$ and $\mathcal{T}'_\rho$ replacing the cluster A_1. We may then reduce by the common cluster $A \cup \{a_1\} \setminus A_1$ of $\mathcal{T}_\rho$ and $\mathcal{T}'_\rho$. This has broken the cluster

reduction by A into a minimal cluster reduction by A_1 and a cluster reduction by a proper subset of A. By repeating this process until the remaining reduction is itself by a minimal non-trivial common cluster, we iteratively break down the cluster reduction by A into a sequence of cluster reductions, each of which is by a minimal non-trivial common cluster.

So we first form a full cluster reduction from $(\mathcal{T}_1, \mathcal{T}_1'), ..., (\mathcal{T}_t, \mathcal{T}_t'), (\mathcal{T}_\rho, \mathcal{T}_\rho')$ by following the same sequence of cluster reductions used to create the cluster sequence, but at each step where we would reduce $\mathcal{T}$ and $\mathcal{T}'$ by a common cluster A, we instead reduce by a sequence of minimal non-trivial common clusters, as described above, whose union contains all the elements of A. Finally, once we have finished breaking down the cluster reductions in the original cluster sequence, we continue to perform cluster reductions on $\mathcal{T}_\rho$ and $\mathcal{T}_\rho'$ by any remaining minimal common clusters until none remain. The result is a full cluster reduction $(\hat{\mathcal{T}}_1, \hat{\mathcal{T}}_1'), ..., (\hat{\mathcal{T}}_s, \hat{\mathcal{T}}_s'), (\hat{\mathcal{T}}_\rho, \hat{\mathcal{T}}_\rho')$ such that each pair $(\mathcal{T}_i, \mathcal{T}_i')$ of the original cluster sequence corresponds to a subsequence $(\hat{\mathcal{T}}_j, \hat{\mathcal{T}}_j'), \ldots, (\hat{\mathcal{T}}_q, \hat{\mathcal{T}}_q')$ of the full cluster reduction, in the sense that $(\hat{\mathcal{T}}_j, \hat{\mathcal{T}}_j'), \ldots, (\hat{\mathcal{T}}_q, \hat{\mathcal{T}}_q')$ is itself a cluster reduction of $(\mathcal{T}_i, \mathcal{T}_i')$. Then, by Lemma 3.2.1,

$$h(\mathcal{T}_i, \mathcal{T}_i') = \sum_{j \leq l \leq q} h(\hat{\mathcal{T}}_l, \hat{\mathcal{T}}_l') \geq \max_{j \leq l \leq q} h(\hat{\mathcal{T}}_l, \hat{\mathcal{T}}_l'),$$

implying

$$k \geq \max_{i \in \{1,\ldots,t,\rho\}} h(\mathcal{T}_i, \mathcal{T}_i') \geq \max_{j \in \{1,\ldots,s,\rho\}} h(\hat{\mathcal{T}}_j, \hat{\mathcal{T}}_j'),$$

and, since this holds for every phylogenetic network $\mathcal{N}$ displaying $\mathcal{T}$ and $\mathcal{T}'$, whatever the level of $\mathcal{N}$, the lemma follows. ■

From Lemmas 3.2.1 and 3.2.2 it follows that there is a network displaying $\mathcal{T}$ and $\mathcal{T}'$ minimizing the hybridization level that also minimizes the hybridization number.

Lemma 3.2.3. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees. A full cluster reduction of $\mathcal{T}$ and $\mathcal{T}'$ can be computed in time $O(n)$, where n is the size of the leaf set of $\mathcal{T}$.*

Proof: We start by applying the algorithm in [36] to $\mathcal{T}$, which preprocesses $\mathcal{T}$ in time $O(n)$ and creates a data structure that returns the least common ancestor (LCA) of any two specific vertices of $\mathcal{T}$ in $O(1)$ time. Then, we compute, for each vertex x of $\mathcal{T}$, the number $l(x)$ of leaves below it in $O(n)$ total time. We do the same for $\mathcal{T}'$. Finally, for each vertex x of $\mathcal{T}$, we store the vertex x' of $\mathcal{T}'$ with $x' := \mathrm{LCA}_{\mathcal{T}'}(c(x))$ as $m(x)$. Since, assuming the children of x are y and z, we have $m(x) = \mathrm{LCA}_{\mathcal{T}'}(m(y), m(z))$, this can be done in $O(n)$ time via a post-order traversal of $\mathcal{T}'$ using the precomputed data structure. Then, a cluster reduction of $\mathcal{T}$ and $\mathcal{T}'$ can be found as follows:

1: $i \leftarrow 1$
2: **for** x in a post-order traversal of $\mathcal{T}$ **do**
3: **if** $l(x) \geq 2$, $l(x) = l(m(x))$ **and** x is not the root of $\mathcal{T}$ **then**
4: $A_i \leftarrow c(x)$
5: $(\mathcal{T}_i, \mathcal{T}_i') \leftarrow (\mathcal{T}_{A_i}, \mathcal{T}_{A_i}')$
6: reduce A_i to a single leaf a_i in both $\mathcal{T}$ and $\mathcal{T}'$
7: **end if**
8: **end for**
9: $(\mathcal{T}_\rho, \mathcal{T}_\rho') \leftarrow (\mathcal{T}, \mathcal{T}')$

The overall worst-case running time of this algorithm is $O(n)$; indeed, although there are $O(n)$ iterations of the outer loop, each one involving reducing a cluster A_i of size $O(n)$ in line 6, the sum of the sizes of the clusters is at most $O(n)$, and so the amortized running-time of this line is $O(1)$. ■

We are now in a position to present and prove our first theorem and Corollary 3.2.5.

Theorem 3.2.4. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees.* HYBRIDIZATION NUMBER *is fixed-parameter tractable with respect to the hybridization level of $\mathcal{T}$ and $\mathcal{T}'$.*

Proof: Let the two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$ and the integer l be an instance of HYBRIDIZATION NUMBER. Let $|X| = n$, and let k be the hybridization level of $\mathcal{T}$ and $\mathcal{T}'$. We may first compute a full cluster reduction $(\mathcal{T}_1, \mathcal{T}_1'), ..., (\mathcal{T}_t, \mathcal{T}_t'), (\mathcal{T}_\rho, \mathcal{T}_\rho')$ of $\mathcal{T}$ and $\mathcal{T}'$ in time $O(n)$ by Lemma 3.2.3. We then apply the algorithm of [62] to each pair $(\mathcal{T}_i, \mathcal{T}_i')$ to obtain $h(\mathcal{T}_i, \mathcal{T}_i')$ in time $O(3.18^{h(\mathcal{T}_i, \mathcal{T}_i')} \cdot |\mathcal{T}_i|)$. By Lemma 3.2.2, $h(\mathcal{T}_i, \mathcal{T}_i') \leq k$, and clearly $\sum_i |\mathcal{T}_i| = O(n)$, hence we may compute $h(\mathcal{T}, \mathcal{T}') = h(\mathcal{T}_1, \mathcal{T}_1') + ... + h(\mathcal{T}_t, \mathcal{T}_t') + h(\mathcal{T}_\rho, \mathcal{T}_\rho')$ in time $O(3.18^k \cdot n)$. By a comparison of $h(\mathcal{T}, \mathcal{T}')$ and l we may answer the decision problem in the same time bound, and hence HYBRIDIZATION NUMBER is fixed parameter tractable when parameterized by the hybridization level of $\mathcal{T}$ and $\mathcal{T}'$. ■

Plugging in current results for HYBRIDIZATION NUMBER [62], Theorem 3.2.4 implies the following.

Corollary 3.2.5. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees.* HYBRIDIZATION NUMBER *can be solved in time $O(3.18^k \cdot n)$, where n is the size of the leaf set of $\mathcal{T}$ and k is the hybridization level of $\mathcal{T}$ and $\mathcal{T}'$.*

Figure 3.5 illustrates how cluster reduction works showing HYBRIDIZATION NUMBER is fixed parameter tractable with respect to the hybridization level of $\mathcal{T}$ and $\mathcal{T}'$. $\mathcal{T}$ and $\mathcal{T}'$ are two rooted binary phylogenetic X-trees where $X = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, and the hybridization level of $\mathcal{T}$ and $\mathcal{T}'$ is $k = 2$. The cluster $C_1 = \{1, 2, 3, 4\}$ is a minimal common cluster in both trees. After first cluster reduction applied, the resulting pair of subtrees are $(\mathcal{T}_\rho, T'_\rho)$ and (T_{C_1}, T'_{C_1}). Then second cluster reduction applies for the cluster $C_2 = \{5, 6, 7, 8, 9\}$, and we obtained the full cluster reduction $(\mathcal{T}_\rho, T'_\rho), (T_{C_1}, T'_{C_1}), (T_{C_2}, T'_{C_2})$ of $\mathcal{T}$ and $\mathcal{T}'$. The next step is computing the hybridization number of each pair of trees; $h(\mathcal{T}_\rho, T'_\rho) = 0, h(T_{C_1}, T'_{C_1}) = 2, h(T_{C_2}, T'_{C_2}) = 2$. Note that, the level of the network is 2 and $h(\mathcal{T}_i, T'_i) \leq k = 2$. Hence we compute $h(\mathcal{T}, \mathcal{T}') = h(\mathcal{T}_\rho, T'_\rho) + h(T_{C_1}, T'_{C_1}) + h(T_{C_2}, T'_{C_2}) = 0 + 2 + 2 = 4$.

3.3 FIXED PARAMETER TRACTABILITY OF rSPR DISTANCE

In this section, we show that for two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$, the *rSPR level* is the maximum rSPR distance between a pair of trees in a full cluster reduction of $\mathcal{T}$ and $\mathcal{T}'$, i.e. the maximum of $d_{\mathrm{rSPR}}(\mathcal{T}_i, \mathcal{T}'_i)$ over $i \in \{1, \ldots, t, \rho\}$.

Recall that, for solving instances of rSPR DISTANCE with two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$, we add to each of them a vertex labelled by a dummy taxon l_ρ at the end of a pendant edge adjoined to the original root. Given such an "augmented" tree $\mathcal{T}$ and a label x, let $\mathcal{T}|_{l_\rho \to x}$ denote the result of removing the vertex labelled l_ρ and replacing the label x by l_ρ. In the following, we make use of the concept of *rooted agreement forests*: Given two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$, a leaf-labelled forest $\mathcal{F}$ is called a *rooted agreement forest* of $\mathcal{T}$ and $\mathcal{T}'$ if $\mathcal{F}$ can be obtained from $\mathcal{T}$ and $\mathcal{T}'$, respectively, by a series of edge cuts as defined in Section 2.4. We say that a rooted agreement forest is *root-isolating* if it contains the singleton tree that consists of the leaf labelled l_ρ. A rooted agreement forest for a cluster sequence $(\mathcal{T}_1, \mathcal{T}'_1), ..., (\mathcal{T}_t, \mathcal{T}'_t), (\mathcal{T}_\rho, \mathcal{T}'_\rho)$ of two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$, is a leaf-labelled forest $\mathcal{F}$ on $X \cup \{a_1, \ldots, a_t\}$ which can be obtained from the forests $\{\mathcal{T}_1, ..., \mathcal{T}_t, \mathcal{T}_\rho\}$ and $\{\mathcal{T}'_1, ..., \mathcal{T}'_t, \mathcal{T}'_\rho\}$ by a series of arc cuts.

For the proof of Theorem 3.3.3, we need to define the concept of *cluster hierarchy*: the cluster hierarchy for a full cluster sequence $(\mathcal{T}_1, \mathcal{T}'_1), ..., (\mathcal{T}_t, \mathcal{T}'_t), (\mathcal{T}_\rho, \mathcal{T}'_\rho)$ of two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$ is defined as the directed tree with a vertex for each component $(\mathcal{T}_i, \mathcal{T}'_i)$ of the cluster sequence, and a directed edge from vertex $(\mathcal{T}_i, \mathcal{T}'_i)$ to vertex $(\mathcal{T}_j, \mathcal{T}'_j)$ if a leaf labelled by a_j is present in $\mathcal{T}'_i$. Then, by starting with $(\mathcal{T}_\rho, \mathcal{T}'_\rho)$ as the root of the tree, and using a breadth-first search, since $t < n$ we have the following:

Observation 3.3.1. *The* cluster hierarchy *for a full cluster sequence can be computed in time $O(n)$, where n is the size of the leaf set of $\mathcal{T}$.*

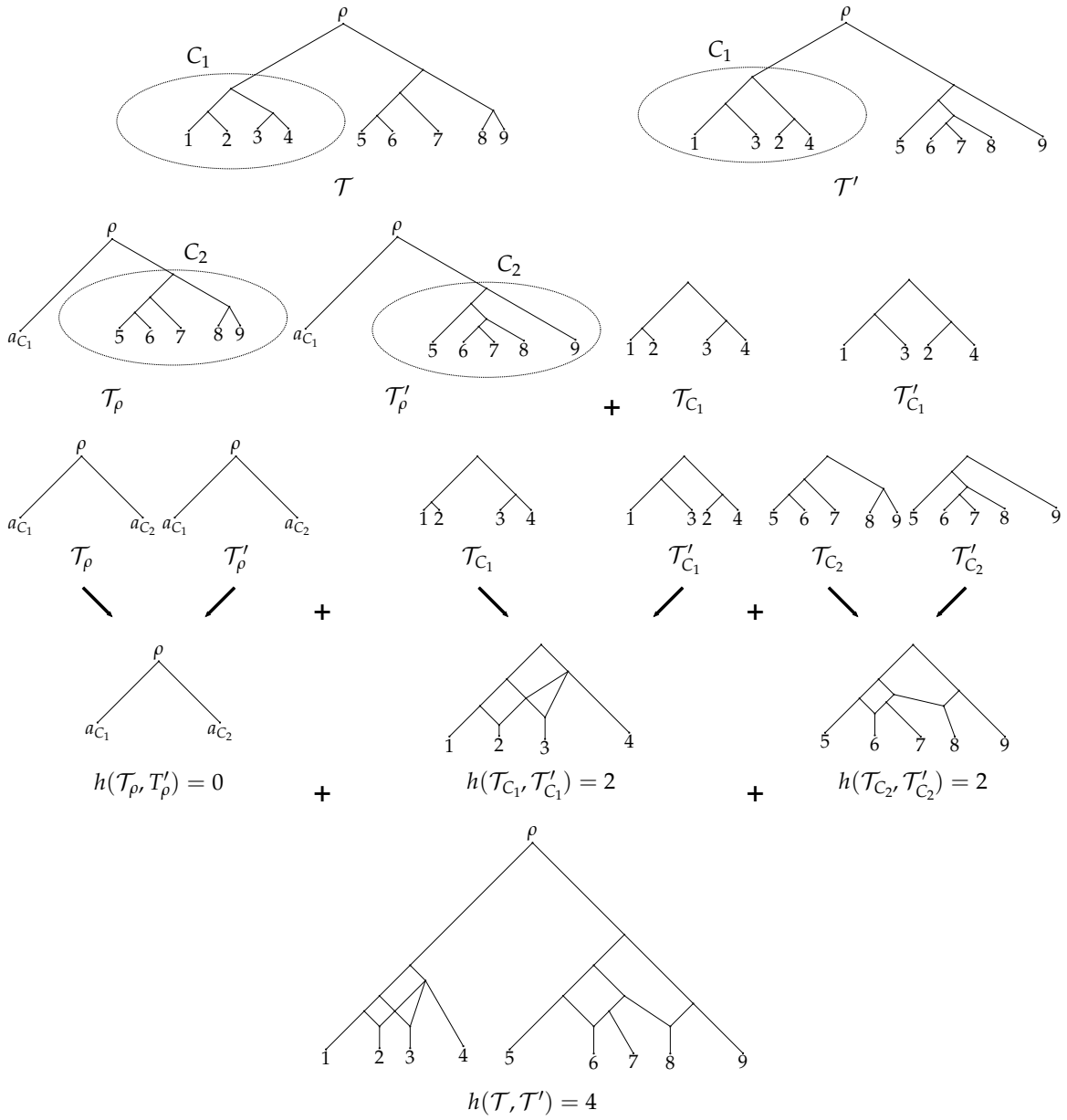


Figure 3.5 An example for fixed parameter tractability of HYBRIDIZATION NUMBER

For the proof of Theorem 3.3.3, we will also make use of the *Minimum-Weight Forest Algorithm* of Linz and Semple [42], which establishes the correctness of the use of a cluster reduction in a divide-and-conquer approach for computing the rSPR distance. In particular, they offer the following theorem and algorithm.

Theorem 3.3.1 (Theorem 2.2 of [42])**.** *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees. Let $(\mathcal{T}_1, \mathcal{T}'_1), ..., (\mathcal{T}_t, \mathcal{T}'_t), (\mathcal{T}_\rho, \mathcal{T}'_\rho)$ be a cluster sequence for $\mathcal{T}$ and $\mathcal{T}'$. Let $\mathcal{G}$ be a rooted agreement forest for this sequence of minimum weight $w(\mathcal{G})$. Then $d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}') = w(\mathcal{G}) - 1$.*

Algorithm MINIMUM-WEIGHT FOREST [42]
Input: A cluster sequence $(\mathcal{T}_1, \mathcal{T}'_1), ..., (\mathcal{T}_t, \mathcal{T}'_t), (\mathcal{T}_\rho, \mathcal{T}'_\rho)$ of two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$, along with its cluster hierarchy.
Output: The minimum weight of a rooted agreement forest for this sequence.

Without needing to give a precise definition of a minimum-weight rooted agreement forest for a cluster sequence (for details see [42]), it suffices to note that if we start with two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$, first compute a full cluster reduction and its cluster hierarchy, and then apply the *Minimum-Weight Forest* algorithm, our output is one more than the rSPR distance between $\mathcal{T}$ and $\mathcal{T}'$. It remains to bound the running time of this approach. To do so, we need the following lemma:

Lemma 3.3.2. *Let $\mathcal{T}$ and $\mathcal{T}'$ be rooted binary phylogenetic X-trees and let $x \in X$. Then, there is a root-isolating rooted maximum-agreement forest $\mathcal{F}$ for $\mathcal{T}$ and $\mathcal{T}'$ if and only if $d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}') = d_{\mathrm{rSPR}}(\mathcal{T}|_{l_\rho \to x}, \mathcal{T}'|_{l_\rho \to x}) + 1$.*

Proof: Let $\mathcal{T}_* := \mathcal{T}|_{l_\rho \to x}$ and $\mathcal{T}'_* := \mathcal{T}'|_{l_\rho \to x}$.

"$\Rightarrow$": Let $\mathcal{F}$ be a root-isolating rooted maximum-agreement forest for $\mathcal{T}$ and $\mathcal{T}'$ and let $\mathcal{T}_\rho$ be the tree in $\mathcal{F}$ that consists of the singleton labelled l_ρ. Then, $d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}') = |\mathcal{F}|$. Let $\mathcal{F}'$ be the result of removing $\mathcal{T}_\rho$ from $\mathcal{F}$ and relabelling the leaf labelled x by l_ρ. Clearly, $\mathcal{F}'$ is a rooted agreement forest for $\mathcal{T}_*$ and $\mathcal{T}'_*$ and, thus, $d_{\mathrm{rSPR}}(\mathcal{T}_*, \mathcal{T}'_*) \leq |\mathcal{F}'| = |\mathcal{F}| - 1$.

To show that $\mathcal{F}'$ maximizes agreement, assume towards a contradiction that there is a rooted agreement forest $\mathcal{F}^*$ for $\mathcal{T}_*$ and $\mathcal{T}'_*$ with $|\mathcal{F}^*| < |\mathcal{F}'|$. Then, relabelling the leaf labelled l_ρ by x in $\mathcal{F}^*$ and adding $\mathcal{T}_\rho$ to $\mathcal{F}^*$ yields a rooted agreement forest for $\mathcal{T}$ and $\mathcal{T}'$ with $|\mathcal{F}^*| + 1 < |\mathcal{F}|$ components, contradicting optimality of $\mathcal{F}$.

"$\Leftarrow$": Let $d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}') = d_{\mathrm{rSPR}}(\mathcal{T}_*, \mathcal{T}'_*) + 1$. We construct a root-isolating rooted maximum-agreement forest $\mathcal{F}$ for $\mathcal{T}$ and $\mathcal{T}'$. To this end, let $\mathcal{F}_*$ be a rooted maximum-agreement forest for $\mathcal{T}_*$ and $\mathcal{T}'_*$ and let $\mathcal{F}$ be the result of relabelling the leaf labelled l_ρ by x

in $\mathcal{F}_*$ and adding a singleton tree whose only vertex is labelled l_ρ. Then, $|\mathcal{F}| = |\mathcal{F}_*| + 1 = d_{\mathrm{rSPR}}(\mathcal{T}_*, \mathcal{T}'_*) + 1 = d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}')$. Thus, $\mathcal{F}$ is a root-isolating rooted maximum-agreement forest for $\mathcal{T}$ and $\mathcal{T}'$. ■

We have now all the building blocks to prove the main results of this section. Our second theorem is an analogue of Theorem 3.2.4 for RSPR DISTANCE instead of HYBRIDIZATION NUMBER.

Theorem 3.3.3. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees.* RSPR DISTANCE *is fixed-parameter tractable with respect to the rSPR level of $\mathcal{T}$ and $\mathcal{T}'$.*

Proof: Let the two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$ and the integer l be an instance of RSPR DISTANCE. Let $|X| = n$, and let k be the rSPR level of $\mathcal{T}$ and $\mathcal{T}'$. We may first compute a full cluster reduction $(\mathcal{T}_1, \mathcal{T}'_1), ..., (\mathcal{T}_t, \mathcal{T}'_t), (\mathcal{T}_\rho, \mathcal{T}'_\rho)$ of $\mathcal{T}$ and $\mathcal{T}'$ and its cluster hierarchy in time $O(n)$ by Lemma 3.2.3 and Observation 3.3.1. We then apply the algorithm of [42] to obtain $d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}')$. The time-consuming step in this algorithm is finding a maximum-agreement forest for each pair $\mathcal{T}_i, \mathcal{T}'_i$ (if possible a root-isolating one). These may be found, using Lemma 3.3.2, in time $O(2.344^{d_{\mathrm{rSPR}}(\mathcal{T}_i, \mathcal{T}'_i)} \cdot |\mathcal{T}_i|)$ by the approach of [20]. By definition, $d_{\mathrm{rSPR}}(\mathcal{T}_i, \mathcal{T}'_i) \leq k$ and, clearly, $|\mathcal{T}_i| \in O(n)$. Hence, the whole algorithm runs in time $O(2.344^k \cdot n)$. By a comparison of $d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}')$ and l we may answer the decision problem in the same time bound, and hence RSPR DISTANCE is fixed parameter tractable when parameterized by the rSPR level of $\mathcal{T}$ and $\mathcal{T}'$. ■

Figure 3.6 illustrates how cluster reduction works by showing that RSPR DISTANCE is fixed parameter tractable with respect to the rSPR level of $\mathcal{T}$ and $\mathcal{T}'$. $\mathcal{T}$ and $\mathcal{T}'$ are two rooted binary phylogenetic X-trees where $X = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Similarly to Figure 3.5, the cluster reduction is applied to $\mathcal{T}$ and $\mathcal{T}'$, and we obtained the full cluster reduction $(\mathcal{T}_\rho, T'_\rho), (T_{C_1}, T'_{C_1}), (T_{C_2}, T'_{C_2})$. Then, $\mathcal{F}(\mathcal{T}_\rho, T'_\rho) = 1$ and $d_{\mathrm{rSPR}}(\mathcal{T}_\rho, T'_\rho) = 0$, $\mathcal{F}(T_{C_1}, T'_{C_1}) = 3$ and $d_{\mathrm{rSPR}}(T_{C_1}, T'_{C_1}) = 2$, and $\mathcal{F}(T_{C_2}, T'_{C_2}) = 3$ and $d_{\mathrm{rSPR}}(T_{C_2}, T'_{C_2}) = 2$. Thus, rSPR level of $\mathcal{T}$ and $\mathcal{T}'$ is 2. Moreover, $d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}') = d_{\mathrm{rSPR}}(\mathcal{T}_\rho, T'_\rho) + d_{\mathrm{rSPR}}(T_{C_1}, T'_{C_1}) + d_{\mathrm{rSPR}}(T_{C_2}, T'_{C_2}) = 0 + 2 + 2 = 4$.

Analogous to the hybridization number, the rSPR level of a pair of trees is at most the rSPR distance between the trees, and may be much smaller, even 1 for trees that have arbitrarily large rSPR distance. Plugging in current results for RSPR DISTANCE [20], Theorem 3.3.3 implies the following.

Corollary 3.3.4. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees.* RSPR DISTANCE *can be solved in time $O(2.344^k \cdot |X|)$, where k is the rSPR level of $\mathcal{T}$ and $\mathcal{T}'$.*

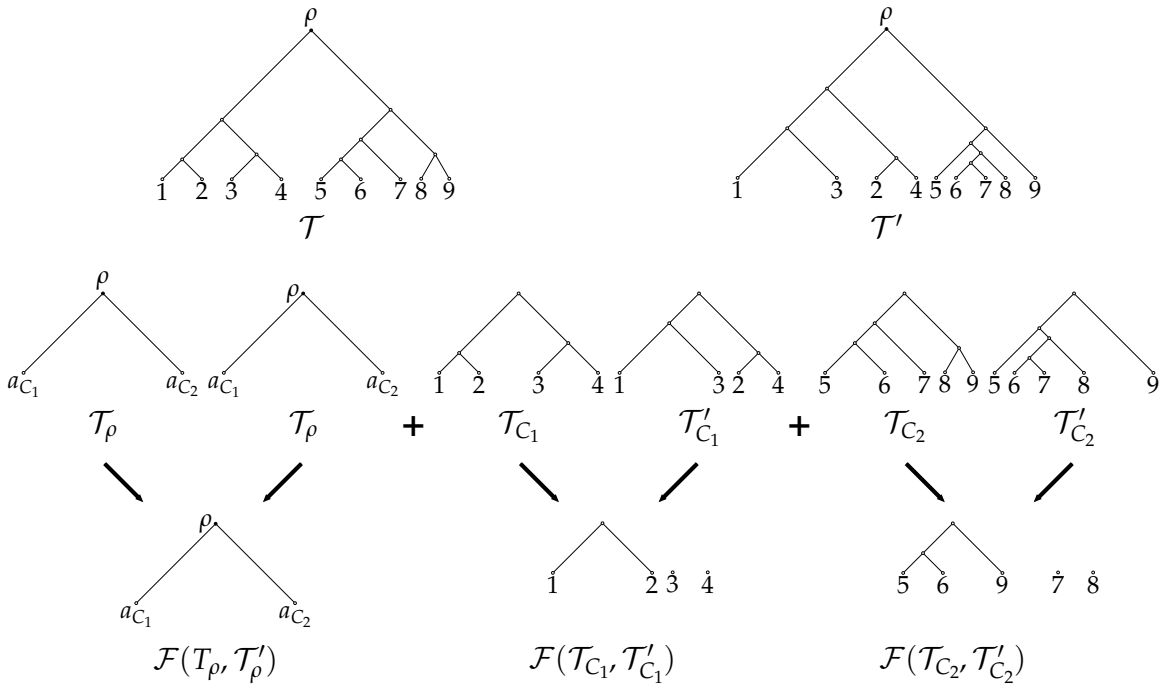


Figure 3.6 An example for fixed parameter tractability of RSPR DISTANCE

Note that the hybridization number of two trees is always bigger than their rSPR distance [4], and so Lemma 3.2.2 and Corollary 3.3.4 imply the following:

Corollary 3.3.5. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees.* RSPR DISTANCE *can be solved in time $O(2.344^k \cdot |X|)$, where k is the hybridization level of $\mathcal{T}$ and $\mathcal{T}'$.*

Note also that the authors of [62] claim to have an algorithm to solve RSPR DISTANCE in $O(2^{d_{\mathrm{rSPR}}(\mathcal{T},\mathcal{T}')} \cdot n)$ [60]. If this is true, the running time in Corollaries 3.3.4 and 3.3.5 will reduce to $O(2^k \cdot n)$.

3.4 FIXED PARAMETER TRACTABILITY OF TBR DISTANCE

In this section, we consider unrooted binary phylogenetic X-trees. An unrooted cluster sequence can be computed as described in Lemma 3.2.3 by previously rooting the two trees on the same leaf.

The following results are fundamental for proving that TBR DISTANCE is fixed parameter tractable in the hybridization level.

Theorem 3.4.1 ([1])**.** *Let $\mathcal{T}$ and $\mathcal{T}'$ be two unrooted binary phylogenetic X-trees. Let $\mathcal{F}$ be a uMAF for $\mathcal{T}$ and $\mathcal{T}'$. Then $d_{\mathrm{TBR}}(\mathcal{T}, \mathcal{T}') = |\mathcal{F}| - 1$.*

Theorem 3.4.2 ([59]). *Let $\mathcal{T}$ and $\mathcal{T}'$ be unrooted binary phylogenetic X-trees. Then $h(\mathcal{T}, \mathcal{T}') = d_{\mathrm{TBR}}(\mathcal{T}, \mathcal{T}')$.*

Note that the concepts of *hybridization number* and *level* refer to the *undirected* versions. The following observation is straightforward.

Observation 3.4.1. *A forest $\mathcal{F} = \{F_1, \ldots, F_k\}$ is a uAF of $\mathcal{T}$ and $\mathcal{T}'$ if and only if*

1. *each tree of $\mathcal{F}$ is displayed by both $\mathcal{T}$ and $\mathcal{T}'$,*
2. *all labels of $\mathcal{T}$ and $\mathcal{T}'$ occur in $\mathcal{F}$, and*
3. *the subtrees $\mathcal{T}(\mathcal{L}(F_1)), \ldots, \mathcal{T}(\mathcal{L}(F_k))$ and $\mathcal{T}'(\mathcal{L}(F_1)), \ldots, \mathcal{T}'(\mathcal{L}(F_k))$ are all vertex disjoint.*

The following two lemmas constitute a portation of Lemma 3.3.2 and Lemma 3.2.1 to unrooted binary phylogenetic trees.

Lemma 3.4.3. *Let $\mathcal{T}$ and $\mathcal{T}'$ be unrooted binary phylogenetic X-trees and let $x \in X$. If there is a uMAF $\mathcal{F}$ for $\mathcal{T}$ and $\mathcal{T}'$ that isolates x, then*

$$d_{\mathrm{TBR}}(\mathcal{T}, \mathcal{T}') = d_{\mathrm{TBR}}(\mathcal{T}|(X - x), \mathcal{T}'|(X - x)) + 1$$

and, otherwise,

$$d_{\mathrm{TBR}}(\mathcal{T}, \mathcal{T}') = d_{\mathrm{TBR}}(\mathcal{T}|(X - x), \mathcal{T}'|(X - x)).$$

Proof: Let $\mathcal{F}'$ be a uMAF for $\mathcal{T}|(X - x)$ and $\mathcal{T}'|(X - x)$.

First, suppose that there is a uMAF $\mathcal{F}$ for $\mathcal{T}$ and $\mathcal{T}'$ that isolates x. Then, $\mathcal{F}$ can be turned into a uAF for $\mathcal{T}|(X - x)$ and $\mathcal{T}'|(X - x)$ by deleting the singleton tree containing x and $\mathcal{F}'$ can be turned into a uAF for $\mathcal{T}$ and $\mathcal{T}'$ by adding a singleton tree containing a vertex labelled x. Thus, $|\mathcal{F}| = |\mathcal{F}'| + 1$.

Next, suppose that there is no uMAF for $\mathcal{T}$ and $\mathcal{T}'$ that isolates x and let $\mathcal{F}$ be a uMAF for $\mathcal{T}$ and $\mathcal{T}'$. Since adding a singleton tree containing a vertex labelled x to $\mathcal{F}'$ yields a uAF for $\mathcal{T}$ and $\mathcal{T}'$ that isolates x, we have $|\mathcal{F}| < |\mathcal{F}'| + 1$. However, since removing x from the tree of $\mathcal{F}$ that contains x yields a uAF for $\mathcal{T}|(X - x)$ and $\mathcal{T}'|(X - x)$, we also have $|\mathcal{F}| \geq |\mathcal{F}'|$. Thus, $|\mathcal{F}| = |\mathcal{F}'|$. The lemma follows by Theorem 3.4.1. ■

Lemma 3.4.4. *Let $\mathcal{T}$ and $\mathcal{T}'$ be unrooted binary phylogenetic X-trees and let C be a nontrivial cluster of $\mathcal{T}$ and $\mathcal{T}'$. If there is a uMAF for $\mathcal{T}{\downarrow_C}$ and $\mathcal{T}'{\downarrow_C}$ that isolates the leaf labelled a_C, then*

$$d_{\mathrm{TBR}}(\mathcal{T}, \mathcal{T}') = d_{\mathrm{TBR}}(\mathcal{T}{\downarrow_C}, \mathcal{T}'{\downarrow_C}) + d_{\mathrm{TBR}}(\mathcal{T}|C, \mathcal{T}'|C),$$

and, otherwise,

$$d_{\mathrm{TBR}}(\mathcal{T},\mathcal{T}') = d_{\mathrm{TBR}}(\mathcal{T}{\downarrow_C},\mathcal{T}'{\downarrow_C}) + d_{\mathrm{TBR}}(\mathcal{T}{\downarrow_{\overline{C}}},\mathcal{T}'{\downarrow_{\overline{C}}}).$$

Proof: First off, suppose that there is a uMAF for $\mathcal{T}{\downarrow_C}$ and $\mathcal{T}'{\downarrow_C}$ that isolates the leaf labelled a_C.

"$\leq$": Let $\mathcal{F}_C$ be a uMAF for $\mathcal{T}|C$ and $\mathcal{T}'|C$. Let $\mathcal{F}_{\overline{C}}$ be analogous for $\overline{C}$.

Let $\mathcal{F}' := \mathcal{F}_{\overline{C}} \uplus \mathcal{F}_C$. Then, all trees of $\mathcal{F}'$ are displayed by $\mathcal{T}$ and $\mathcal{T}'$ and by Observation 3.4.1, $\mathcal{F}'$ is a uAF for $\mathcal{T}$ and $\mathcal{T}'$.

Thus,

$$\begin{aligned} d_{\mathrm{TBR}}(\mathcal{T},\mathcal{T}') &\leq |\mathcal{F}'| - 1 \\ &= |\mathcal{F}_{\overline{C}}| + |\mathcal{F}_C| - 1 \\ &\overset{\text{Theorem 3.4.1}}{=} d_{\mathrm{TBR}}(\mathcal{T}|\overline{C},\mathcal{T}'|\overline{C}) + d_{\mathrm{TBR}}(\mathcal{T}|C,\mathcal{T}'|C) + 1 \\ &\overset{\text{Lemma 3.4.3}}{=} d_{\mathrm{TBR}}(\mathcal{T}{\downarrow_C},\mathcal{T}'{\downarrow_C}) + d_{\mathrm{TBR}}(\mathcal{T}|C,\mathcal{T}'|C) \end{aligned}$$

"$\geq$": Let $\mathcal{F}$ be a uMAF for $\mathcal{T}$ and $\mathcal{T}'$. Let $\mathcal{F}(C)$ denote the set containing exactly the trees of $\mathcal{F}$ that contain only leaves labelled by elements of C. Let $\mathcal{F}(\overline{C})$ be defined analogously for $\overline{C}$.

Case 1: $\mathcal{F} = \mathcal{F}(C) \uplus \mathcal{F}(\overline{C})$. Then, $|\operatorname{uMAF}(\mathcal{T}|C,\mathcal{T}'|C)| = |\mathcal{F}(C)|$ since, otherwise, exchanging $\mathcal{F}(C)$ for a uMAF of $\mathcal{T}|C$ and $\mathcal{T}'|C$ in $\mathcal{F}$ yields a uAF that is smaller than $\mathcal{F}$, contradicting optimality of $\mathcal{F}$. Likewise, $|\operatorname{uMAF}(\mathcal{T}|\overline{C},\mathcal{T}'|\overline{C})| = |\mathcal{F}(\overline{C})|$. Then,

$$\begin{aligned} d_{\mathrm{TBR}}(\mathcal{T},\mathcal{T}') &= |\mathcal{F}| - 1 = |\mathcal{F}(C)| + |\mathcal{F}(\overline{C})| - 1 \\ &\overset{\text{Theorem 3.4.1}}{=} d_{\mathrm{TBR}}(\mathcal{T}|C,\mathcal{T}'|C) + d_{\mathrm{TBR}}(\mathcal{T}|\overline{C},\mathcal{T}'|\overline{C}) + 1 \\ &\overset{\text{Lemma 3.4.3}}{=} d_{\mathrm{TBR}}(\mathcal{T}{\downarrow_C},\mathcal{T}'{\downarrow_C}) + d_{\mathrm{TBR}}(\mathcal{T}|\overline{C},\mathcal{T}'|\overline{C}) \end{aligned}$$

Case 2: There is a tree H in $\mathcal{F}$ containing a leaf labelled $x \in C$ and a leaf labelled $y \in \overline{C}$ (note that only one of such "mixed" trees can be present in $\mathcal{F}$; indeed, since C is a cluster of both trees, the existence of two such trees will contradict Condition 3 of Observation 3.4.1). Then, $\mathcal{F} = \mathcal{F}(C) \uplus \mathcal{F}(\overline{C}) \uplus \{H\}$.

Let $H{\downarrow_C}$ denote the result of contracting all edges of H that are on a path between two leaves with labels of C in H and labelling the vertex on which they are all contracted with C. Let $H{\downarrow_{\overline{C}}}$ be analogous for $\overline{C}$.

Then, all labels of C and the special label $a_{\overline{C}}$ occur in $\mathcal{F}_1 := \mathcal{F}(C) \uplus \{H{\downarrow_{\overline{C}}}\}$ and all its trees are displayed by $\mathcal{T}{\downarrow_{\overline{C}}}$ and $\mathcal{T}'{\downarrow_{\overline{C}}}$. Thus, by Observation 3.4.1, $\mathcal{F}_1$ is a uAF for $\mathcal{T}{\downarrow_{\overline{C}}}$ and $\mathcal{T}'{\downarrow_{\overline{C}}}$. Likewise, $\mathcal{F}(\overline{C}) \uplus \{H{\downarrow_C}\}$ is a uAF for $\mathcal{T}{\downarrow_C}$ and $\mathcal{T}'{\downarrow_C}$. Thus,

$$\begin{aligned} d_{\mathrm{TBR}}(\mathcal{T},\mathcal{T}') = |\mathcal{F}| - 1 &= |\mathcal{F}(C) \uplus \mathcal{F}(\overline{C}) \uplus \{H\}| - 1 \\ &= |\mathcal{F}(C) \uplus \{H{\downarrow_{\overline{C}}}\}| + |\mathcal{F}(\overline{C}) \uplus \{H{\downarrow_C}\}| - 2 \\ &\geq d_{\mathrm{TBR}}(\mathcal{T}{\downarrow_C},\mathcal{T}'{\downarrow_C}) + d_{\mathrm{TBR}}(\mathcal{T}{\downarrow_{\overline{C}}},\mathcal{T}'{\downarrow_{\overline{C}}}) \\ &\geq d_{\mathrm{TBR}}(\mathcal{T}{\downarrow_C},\mathcal{T}'{\downarrow_C}) + d_{\mathrm{TBR}}(\mathcal{T}|C,\mathcal{T}'|C) \end{aligned}$$

Next, suppose that there is no uMAF for $\mathcal{T}{\downarrow_C}$ and $\mathcal{T}'{\downarrow_C}$ that isolates the leaf labelled a_C.

"$\leq$": First, note that if there is a uMAF for $\mathcal{T}{\downarrow_{\overline{C}}}$ and $\mathcal{T}'{\downarrow_{\overline{C}}}$ that isolates the leaf labelled $a_{\overline{C}}$, the first part of our proof implies that

$$\begin{aligned} d_{\mathrm{TBR}}(\mathcal{T},\mathcal{T}') &= d_{\mathrm{TBR}}(\mathcal{T}|\overline{C},\mathcal{T}'|\overline{C}) + d_{\mathrm{TBR}}(\mathcal{T}{\downarrow_{\overline{C}}},\mathcal{T}'{\downarrow_{\overline{C}}}) \\ &\leq d_{\mathrm{TBR}}(\mathcal{T}{\downarrow_C},\mathcal{T}'{\downarrow_C}) + d_{\mathrm{TBR}}(\mathcal{T}{\downarrow_{\overline{C}}},\mathcal{T}'{\downarrow_{\overline{C}}}). \end{aligned}$$

Now, let us consider the case where there is no uMAF for $\mathcal{T}{\downarrow_C}$ and $\mathcal{T}'{\downarrow_C}$ (respectively $\mathcal{T}{\downarrow_{\overline{C}}}$ and $\mathcal{T}'{\downarrow_{\overline{C}}}$) that isolates the leaf labelled a_C (respectively labelled $a_{\overline{C}}$). Let $\mathcal{F}_C$ be a uMAF for $\mathcal{T}{\downarrow_{\overline{C}}}$ and $\mathcal{T}'{\downarrow_{\overline{C}}}$ and let $H_{\overline{C}}$ denote the tree of $\mathcal{F}_C$ containing the label $a_{\overline{C}}$.

Let $\mathcal{F}_{\overline{C}}$ and H_C be analogous for C. Let H be the result of joining H_C and $H_{\overline{C}}$ by identifying the leaves labelled $a_{\overline{C}}$ and a_C, respectively and suppressing this degree-two vertex.

Let $\mathcal{F}' := (\mathcal{F}_{\overline{C}} \setminus \{H_C\}) \uplus (\mathcal{F}_C \setminus \{H_{\overline{C}}\}) \uplus \{H\}$. Then, H is displayed by $\mathcal{T}$ and $\mathcal{T}'$ and, thus, all trees of $\mathcal{F}'$ are displayed by $\mathcal{T}$ and $\mathcal{T}'$. Moreover, it is easy to see that $\mathcal{T}(H)$ is vertex disjoint with the other trees in the forest, and the same holds for $\mathcal{T}'(H)$. Then, by Observation 3.4.1, $\mathcal{F}'$ is a uAF for $\mathcal{T}$ and $\mathcal{T}'$.

Thus,

$$\begin{aligned} d_{\mathrm{TBR}}(\mathcal{T},\mathcal{T}') &\leq |\mathcal{F}'| - 1 \\ &= |\mathcal{F}_{\overline{C}} \setminus \{H_C\}| + |\mathcal{F}_C \setminus \{H_{\overline{C}}\}| + |\{H\}| - 1 \\ &= |\mathcal{F}_{\overline{C}}| + |\mathcal{F}_C| - 2 \\ &= d_{\mathrm{TBR}}(\mathcal{T}|\overline{C},\mathcal{T}'|\overline{C}) + d_{\mathrm{TBR}}(\mathcal{T}|C,\mathcal{T}'|C) \\ &\overset{\text{Lemma 3.4.3}}{=} d_{\mathrm{TBR}}(\mathcal{T}{\downarrow_C},\mathcal{T}'{\downarrow_C}) + d_{\mathrm{TBR}}(\mathcal{T}{\downarrow_C},\mathcal{T}'{\downarrow_C}) \end{aligned}$$

"$\geq$": Let $\mathcal{F}$ be a uMAF for $\mathcal{T}$ and $\mathcal{T}'$. Let $\mathcal{F}(C)$ denote the set containing exactly the trees of $\mathcal{F}$ that contain only leaves labelled by elements of C. Let $\mathcal{F}(\overline{C})$ be defined analogously for $\overline{C}$.

Case 1: $\mathcal{F} = \mathcal{F}(C) \uplus \mathcal{F}(\overline{C})$. Then, $|\operatorname{uMAF}(\mathcal{T}|C,\mathcal{T}'|C)| = |\mathcal{F}(C)|$ since, otherwise, exchanging $\mathcal{F}(C)$ for a uMAF of $\mathcal{T}|C$ and $\mathcal{T}'|C$ in $\mathcal{F}$ yields a uAF that is smaller than $\mathcal{F}$, contradicting optimality of $\mathcal{F}$. Likewise, $|\operatorname{uMAF}(\mathcal{T}|\overline{C},\mathcal{T}'|\overline{C})| = |\mathcal{F}(\overline{C})|$.

Let $\mathcal{F}'(\overline{C})$ be a uMAF for $\mathcal{T}{\downarrow_C}$ and $\mathcal{T}'{\downarrow_C}$ and note that, by Lemma 3.4.3 $|\mathcal{F}'(\overline{C})| = |\mathcal{F}(\overline{C})|$. Further, let $\mathcal{F}'(C)$ be a uMAF for $\mathcal{T}{\downarrow_{\overline{C}}}$ and $\mathcal{T}'{\downarrow_{\overline{C}}}$ and note that $|\mathcal{F}'(C)| \leq |\mathcal{F}(C)| + 1$. Then,

$$\begin{aligned} d_{\mathrm{TBR}}(\mathcal{T},\mathcal{T}') = |\mathcal{F}| - 1 &= |\mathcal{F}(C)| + |\mathcal{F}(\overline{C})| - 1 \\ &\geq |\mathcal{F}'(C)| + |\mathcal{F}'(\overline{C})| - 2 \\ &= d_{\mathrm{TBR}}(\mathcal{T}{\downarrow_C},\mathcal{T}'{\downarrow_C}) + d_{\mathrm{TBR}}(\mathcal{T}{\downarrow_{\overline{C}}},\mathcal{T}'{\downarrow_{\overline{C}}}) \end{aligned}$$

Case 2: There is a tree H in $\mathcal{F}$ containing a leaf labelled $x \in C$ and a leaf labelled $y \in \overline{C}$. This is completely analogous to Case 2 above. ■

It is worth mentioning that, in the two cases of Lemma 3.4.4, the TBR distances differ by exactly one, that is, $d_{\mathrm{TBR}}(\mathcal{T}|C,\mathcal{T}'|C) \leq d_{\mathrm{TBR}}(\mathcal{T}{\downarrow_C},\mathcal{T}'{\downarrow_C}) \leq d_{\mathrm{TBR}}(\mathcal{T}|C,\mathcal{T}'|C) + 1$, Lemma 3.4.4 implies that, if there is a uMAF for $\mathcal{T}{\downarrow_C}$ and $\mathcal{T}'{\downarrow_C}$ that isolates the leaf labelled a_C and a uMAF for $\mathcal{T}{\downarrow_{\overline{C}}}$ and $\mathcal{T}'{\downarrow_{\overline{C}}}$ that isolates the leaf labelled $a_{\overline{C}}$, then, when gluing the forests of the subtrees back together to form a uMAF $\mathcal{F}$ for $\mathcal{T}$ and $\mathcal{T}'$, we have a tree that does not contain any labelled leaf. Thus, an optimal uMAF has size $|\mathcal{F}| - 1$. This means that, to minimize the size of a forest for $\mathcal{T}$ and $\mathcal{T}'$, we need to favour the forests isolating the dummy taxa. Then, we have the following:

Corollary 3.4.5. *Let $\mathcal{T}$ and $\mathcal{T}'$ be unrooted binary phylogenetic X-trees. Let*

$$(\mathcal{T}_1, \mathcal{T}'_1), \ldots, (\mathcal{T}_t, \mathcal{T}'_t)$$

be a cluster sequence of $\mathcal{T}$ and $\mathcal{T}'$. Let $\mathcal{F}$ be a maximum-agreement forest of $\mathcal{T}$ and $\mathcal{T}'$. For $i \in \{1, \ldots, t\}$, let $\mathcal{F}_i$ be a maximum-agreement forest for $\mathcal{T}_i$ and $\mathcal{T}'_i$ such that $r := |\{C : \{a_C\}, \{a_{\overline{C}}\} \in \uplus_i \mathcal{F}_i\}|$ is maximal. Then, $d_{\mathrm{TBR}}(\mathcal{T}, \mathcal{T}') = (\sum_i |\mathcal{F}_i|) - t - r$.

Corollary 3.4.5 is a drop-in replacement for Theorem 3.3.1 and lets us use the entire cluster-sequence-based machinery of [42] for unrooted phylogenetic trees. Thus, a slight modification of the MINIMUM-WEIGHT FOREST algorithm of [42] (solving the TBR DISTANCE instead of the RSPR DISTANCE and using the unrooted cluster reduction instead of the rooted one) leads right to the following theorem:

Theorem 3.4.6. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two unrooted binary phylogenetic X-trees and let*

$$(\mathcal{T}_1, \mathcal{T}'_1), \ldots, (\mathcal{T}_t, \mathcal{T}'_t)$$

be a full cluster reduction of $\mathcal{T}$ and $\mathcal{T}'$. Then, the hybridization level of $\mathcal{T}$ and $\mathcal{T}'$ equals

$$\max_{i \in \{1,\ldots,t\}} d_{\mathrm{TBR}}(\mathcal{T}_i, \mathcal{T}'_i).$$

Proof: First, from Lemma 3.4.2, we have that

$$\max_{i \in \{1,\ldots,t\}} d_{\mathrm{TBR}}(\mathcal{T}_i, \mathcal{T}'_i) = \max_{i \in \{1,\ldots,t\}} h(\mathcal{T}_i, \mathcal{T}'_i).$$

The fact that $\max_{i \in \{1,\ldots,t\}} h(\mathcal{T}_i, \mathcal{T}'_i)$ equals the hybridization level of $\mathcal{T}$ and $\mathcal{T}'$ can be proven similarly to Lemma 3.2.2, and we do not repeat the proof here. ■

Thanks to Theorem 3.4.6, Theorem 3.4.7 and Corollary 3.4.8 can be proven similarly to Theorem 3.3.3 and Corollary 3.3.4, since TBR DISTANCE can be solved in $O(3^k \cdot n)$, where k is the TBR distance of $\mathcal{T}$ and $\mathcal{T}'$ [19].

Note that the notions of *displaying*, *hybridization number* and *hybridization level* of two unrooted trees are defined as in the rooted framework. Our third theorem is an analogue of Theorem 3.2.4 for TBR DISTANCE instead of HYBRIDIZATION NUMBER.

Theorem 3.4.7. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two unrooted binary phylogenetic X-trees.* TBR DISTANCE *is fixed-parameter tractable with respect to the hybridization level of $\mathcal{T}$ and $\mathcal{T}'$.*

Plugging in current results for TBR DISTANCE [19], Theorem 3.4.7 implies the following.

Corollary 3.4.8. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two unrooted binary phylogenetic X-trees.* TBR DISTANCE *can be solved in time $O(3^k \cdot |X|)$, where k is the hybridization level of $\mathcal{T}$ and $\mathcal{T}'$.*

Figure 3.7 illustrates how cluster reduction works by showing that TBR DISTANCE is fixed parameter tractable with respect to the TBR level of $\mathcal{T}$ and $\mathcal{T}'$. $\mathcal{T}$ and $\mathcal{T}'$ are two unrooted binary phylogenetic X-trees where $X = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

The cluster $C_1 = \{7, 8, 9\}$ is a minimal common cluster in both trees. After first cluster reduction applied, resulting pair of subtrees are $(\mathcal{T}{\downarrow_{C_1}}, \mathcal{T}'{\downarrow_{C_1}})$ and $(\mathcal{T}{\downarrow_{\overline{C}_1}}, \mathcal{T}'{\downarrow_{\overline{C}_1}})$. Then second cluster reduction applies for the cluster $C_2 = \{4, 5\}$, resulting pair of subtrees are $(\mathcal{T}{\downarrow_{C_2}}, \mathcal{T}'{\downarrow_{C_2}})$, $(\mathcal{T}{\downarrow_{\overline{C}_1}}, \mathcal{T}'{\downarrow_{\overline{C}_1}})$ and $(\mathcal{T}{\downarrow_{\overline{C}_2}}, \mathcal{T}'{\downarrow_{\overline{C}_2}})$. Then third cluster reduction applies for the cluster $C_2 = \{1, 2\}$, and we obtained the full cluster reduction $(\mathcal{T}{\downarrow_{C_3}}, \mathcal{T}'{\downarrow_{C_3}})$, $(\mathcal{T}{\downarrow_{\overline{C}_1}}, \mathcal{T}'{\downarrow_{\overline{C}_1}})$, $(\mathcal{T}{\downarrow_{\overline{C}_2}}, \mathcal{T}'{\downarrow_{\overline{C}_2}})$ and $(\mathcal{T}{\downarrow_{\overline{C}_3}}, \mathcal{T}'{\downarrow_{\overline{C}_3}})$. Then uMAF calculated for each pair of trees: $\mathcal{F}(\mathcal{T}{\downarrow_{C_3}}, \mathcal{T}'{\downarrow_{C_3}}) = 3$, $\mathcal{F}(\mathcal{T}{\downarrow_{\overline{C}_1}}, \mathcal{T}'{\downarrow_{\overline{C}_1}}) = 2$, $\mathcal{F}(\mathcal{T}{\downarrow_{\overline{C}_2}}, \mathcal{T}'{\downarrow_{\overline{C}_2}}) = 1$ and $\mathcal{F}(\mathcal{T}{\downarrow_{\overline{C}_3}}, \mathcal{T}'{\downarrow_{\overline{C}_3}}) = 1$. By Corollary 3.4.5,

$$d_{\mathrm{TBR}}(\mathcal{T}, \mathcal{T}') = (\sum_i |\mathcal{F}_i|) - t - r = 7 - 4 - 1 = 2.$$

3.5 CONCLUSION

We have described a strictly stronger parameter to calculate the minimum number of hybridization number, rSPR distance and TBR distance that explains two binary phylogenetic trees. Beside previous approaches, we use a divide and conquer approach to break the problem into smaller pieces. For the HYBRIDIZATION NUMBER problem, each small piece corresponds to a biconnected component of the hybridization network of the input trees. Thus, the maximum hybridization number among the pieces equals the level of network. For the RSPR DISTANCE problem, we define rSPR level as a maximum rSPR distance between each pair of pieces. Finally, we show that calculating HYBRIDIZATION NUMBER, RSPR DISTANCE, TBR DISTANCE problems are fixed parameter tractable, where the parameter is hybridization level, rSPR-level and TBR-level respectively. For two rooted phylogenetic X trees, the time taken for HYBRIDIZATION NUMBER problem is $O(3.18^k \cdot n)$ where n is the size of leaf set and k is the hybridization level. Similarly, the time taken for RSPR DISTANCE problem is $O(2.344^k \cdot |X|)$ where k is the rSPR level. Lastly, the time taken for TBR DISTANCE problem is $O(3^k \cdot |X|)$ where k is the hybridization level.

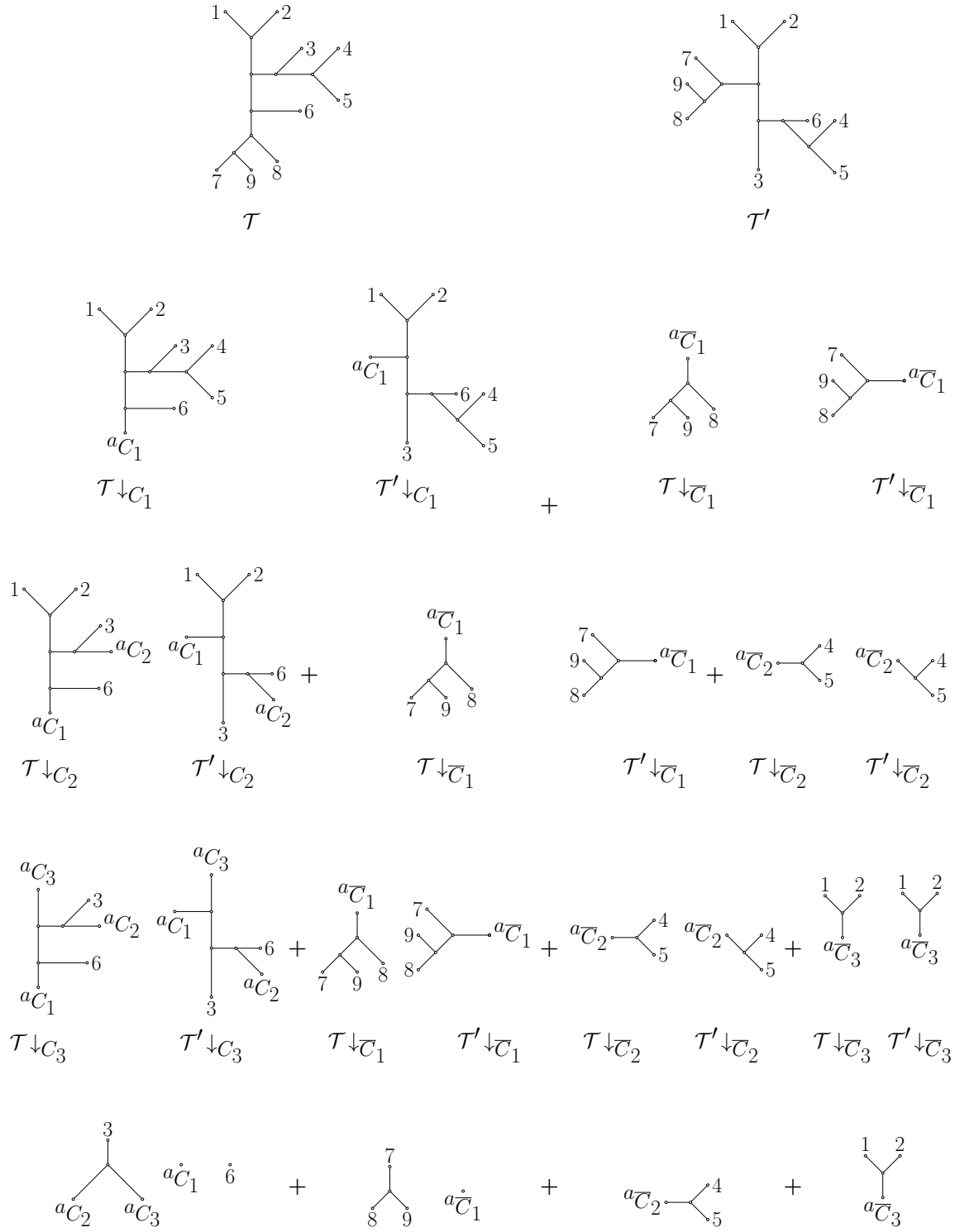


Figure 3.7 An example for fixed parameter tractability of TBR DISTANCE

4

NETWORKUPGMA ALGORITHM AND ITS SAFETY RADIUS

The evolutionary history of organisms is generally represented by a phylogenetic tree. One popular and fast method for reconstructing phylogenetic tree from DNA or protein sequence data is to first compute a matrix of pairwise distances between the taxa, and then infer the phylogenetic tree from this distance matrix. Such approaches are called *distance-based* methods, and they are very widely used due to their simplicity and speed. The two most well known and long standing approaches are UPGMA [54] and Neighbor Joining [51]. In recent years several variants of these and new approaches have been suggested, including Least Squares [26], BioNJ [28] and Balanced Minimum Evolution [31]. In [3] and [31] gives the accuracy and properties of distance based methods, specifically Neighbor Joining Method. In [13], distance based methods are applied to phylogenetic networks. Moret [45] gives the experimental results of distance based phylogenetic reconstruction methods.

In this chapter, first, we introduce the algorithm NETWORKUPGMA which shows that a weighted ultrametric tree-child network is reconstructed from the true (*i.e.* perfectly accurate) set-distance matrix by this algorithm in polynomial time. Second, we determine the safety radius of the NETWORKUPGMA algorithm.

4.1 NETWORKUPGMA ALGORITHM

In this section, we consider the task of reconstructing phylogenetic networks from distance data. The reconstruction of restricted classes of phylogenetic network from inter-taxa distances have been studied in a number of recent papers. A key feature of this problem is that in a network there is no longer a unique distance between a pair of taxa (as there is in a tree), so one must work with shortest distances, average distances or sets or subsets of distances. Chan et al. [18] take a matrix of inter-taxa distances and reconstruct an ultrametric galled network (more commonly called a galled tree or a level-1 network, see Chapter 2) such that there is a path between each pair of taxa having the weight given in the matrix, if

such a network exists. Willson [64] studied the problem of determining the network given the average distance between taxa, where each reticulation vertex assigns a probability to its two incoming arcs. He manages the reconstruction of phylogenetic networks which have a single reticulation cycle from such distances in polynomial time [65]. In a recent paper [13], Bordewich and Semple showed that (unweighted) tree-child phylogenetic networks may be reconstructed from the multi-set of path lengths between taxa and that temporal, tree-child, phylogenetic networks may be reconstructed from the set of path lengths between taxa, each in polynomial time in the size of the input.

In this section, which builds on and extends the approach of [14], we present a polynomial-time algorithm (which we have called NETWORKUPGMA) that reconstructs an ultrametric tree-child network from the set of distances between each pair of taxa. Our algorithm offers an improvement over previous works in two ways. First ultrametric tree-child networks are a much wider class of networks than networks with only a single reticulation or ultrametric galled networks, which are a subclass of ultrametric tree-child networks. In particular note that: the total number of reticulations in a tree-child network on n taxa can be as large as $n-1$ [17], whereas a galled network has at most $n/2$ retculations; and the interrelation of reticulations may be more complex, as each 2-connected component of our networks may contain many reticulations (again linear in the number of taxa), whereas in a galled network there can only be one reticulation in each 2-connected component. Second, the algorithm takes the *set of distances* between each pair of taxa as input, where Bordewich and Semple [14] required the *multiset of path lengths* (for unweighted tree-child networks). This is an important distinction: the distance matrices come from estimating evolutionary distance based upon sequence data of some type. Real phylogenies are weighted: edge weights correspond to some measure of genetic difference. Furthermore, while it is quite conceivable that by sampling different genes or regions of the genome one might build up an accurate picture of the set of different evolutionary path weights between a given pair of taxa, it seems hard to imagine how one might manage to measure the number of distinct evolutionary paths of a given observed weight. Thus the set of distances seems a much more reasonable input for an algorithm in practice.

4.1.1 DEFINITIONS AND STATEMENT OF RESULTS

In this subsection we give further definitions which we shall require in order to present our algorithm and proof. Definition of 'ultrametric tree-child networks' and 'distance matrices' is given in Chapter 2.

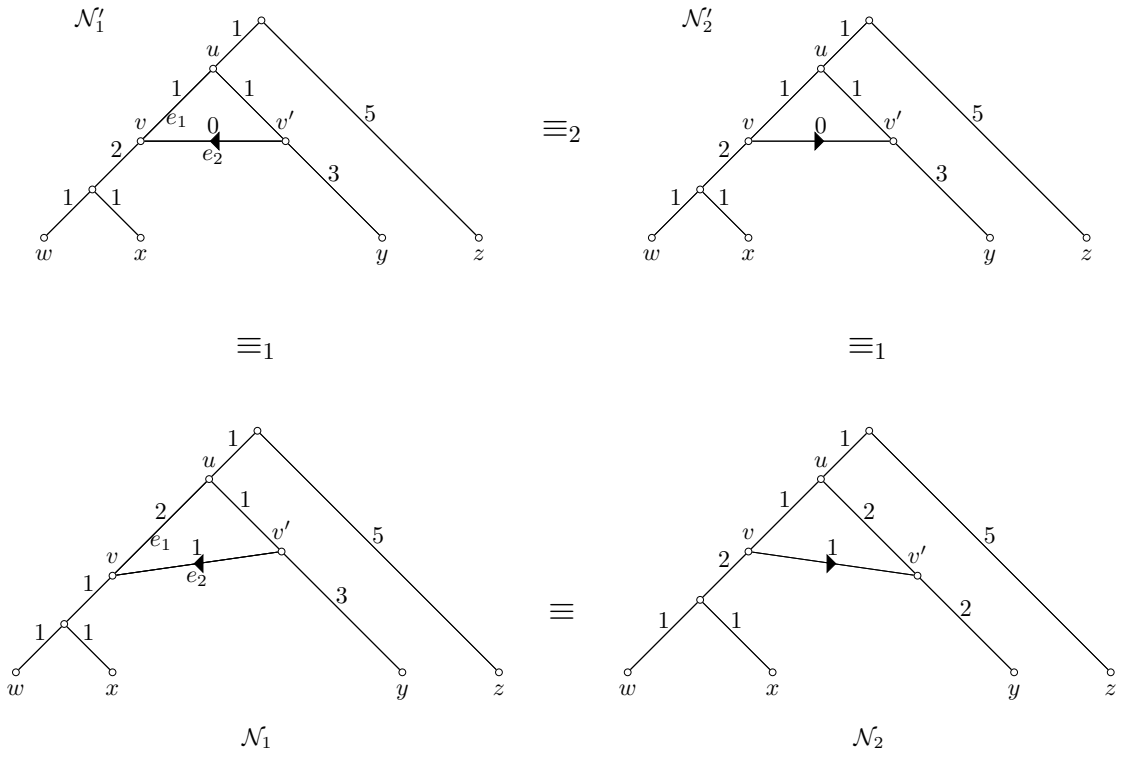


Figure 4.1 Four ultrametric tree-child networks each containing an immediate reticulation. Networks $\mathcal{N}_1$ and $\mathcal{N}_1'$ (and $\mathcal{N}_2$ and $\mathcal{N}_2'$) are equivalent up to weights at reticulations. Networks $\mathcal{N}_1'$ and $\mathcal{N}_2'$ are equivalent up to direction of immediate reticulations. Thus networks $\mathcal{N}_1$ and $\mathcal{N}_2$ are equivalent under $\equiv$.

4.1.1.1 EQUIVALENT NETWORKS

It will turn out that the set-distance matrix is not sufficient to determine a unique ultrametric tree-child network that displays it. However it is nearly sufficient. We now define an equivalence relation ($\equiv$) on ultrametric tree-child networks which captures precisely when two such networks display the same set-distance matrix.

Two ultrametric tree-child networks $\mathcal{N}_1, \mathcal{N}_1'$ are said to be *equivalent up to weights at reticulations* (denoted $\equiv_1$) if the underlying unweighted networks are isomorphic and: at each reticulation v with incoming arcs e_1 and e_2 and outgoing arc e_3, the weight of the path e_1, e_3 is the same in $\mathcal{N}_1$ and $\mathcal{N}_1'$, and also the weight of the path e_2, e_3 is the same in $\mathcal{N}_1$ and $\mathcal{N}_1'$. Thus if arcs e_1, e_2, e_3 have weights l_1, l_2, l_3 respectively, any network $\mathcal{N}'$ formed by changing the weights of arcs e_1, e_2, e_3 to $l_1 - \epsilon, l_2 - \epsilon, l_3 + \epsilon$ respectively for some $\epsilon \in (-l_3, \min\{l_1, l_2\})$ is equivalent to $\mathcal{N}_1$ up to weights at reticulations. We define a class representative for each equivalence class as the network in which one of the incoming edge

weights is zero at every reticulation. E.g. for the network $\mathcal{N}_1$, the class representative would have arcs e_1, e_2, e_3 with weights $l_1 - \epsilon, l_2 - \epsilon, l_3 + \epsilon$ where $\epsilon = \min\{l_1, l_2\}$. In Figure 4.1 networks $\mathcal{N}_1 \equiv_1 \mathcal{N}_1'$ and $\mathcal{N}_2 \equiv_1 \mathcal{N}_2'$. Moreover $\mathcal{N}_1'$ and $\mathcal{N}_2'$ are class representatives.

We next define a second equivalence relation, denoted $\equiv_2$, on the class representatives. A reticulation vertex whose two parents are also parent and child is said to be an *immediate reticulation*, i.e. v is an immediate reticulation if it is a reticulation node with parents u and w such that w is also a child of u. In the case that the network is a class representative, the arc (w, v) has weight 0. An immediate reticulation occurs when a parent species immediately recombines with its own offspring. In each network shown in Figure 4.1 the reticulation is an immediate reticulation. We say two class representative phylogenetic networks $\mathcal{N}_1', \mathcal{N}_2'$ are *equivalent up to direction of immediate reticulations* if for some set of immediate reticulations R in $\mathcal{N}_1'$ such that v in R has parents u, w where (u, w) is an arc, then the network $\mathcal{N}_2'$ is formed by removing the arc (w, v) and inserting the arcs (v, w) (so that w is now an immediate reticulation with parents u, v), where the new arc has weight 0. Note that $\mathcal{N}_1'$ and $\mathcal{N}_2'$ display the same set-distance matrix. In Figure 4.1, the network $\mathcal{N}_1' \equiv_2 \mathcal{N}_2'$.

Finally we define the equivalence relation $\equiv$ on phylogenetic networks, where $\mathcal{N}_1 \equiv \mathcal{N}_2$ if the class representatives (under $\equiv_1$) for $\mathcal{N}_1$ and $\mathcal{N}_2$ are equivalent under $\equiv_2$. For example in Figure 4.1, $\mathcal{N}_1 \equiv \mathcal{N}_2$ since $\mathcal{N}_1 \equiv_1 \mathcal{N}_1' \equiv_2 \mathcal{N}_2' \equiv_1 \mathcal{N}_2$.

4.1.1.2 CHERRY REDUCTIONS

Let $\mathcal{N}$ be an ultrametric tree-child network on X. If there is a pair of leaves $\{x, y\}$ is a cherry, then note that the distances from this parent to x and y are the same. Figure 4.2 (a) depicts a cherry $\{x, y\}$. *Reducing a cherry* $\{x, y\}$ is the operation replacing the cherry with a single new node while keeping the ultrametric property, see Figure 4.2(d). Note that the number of leaves in the resulting network is reduced by one, but the number of reticulations is unchanged.

A two-element subset $\{x, y\}$ of X is a *reticulated cherry* in $\mathcal{N}$ if there is an up-down path consisting of three arcs, say $(x, u), (u, v), (v, y)$, between x and y where u is a tree vertex, and v is a reticulation vertex. Necessarily, the arc joining u and v is directed from a tree vertex to the reticulation vertex. This arc is referred to as the reticulation arc of the reticulated cherry. The leaf adjacent to the tree vertex is called the tree leaf of the reticulated cherry, and the leaf adjacent to the reticulation is the reticulation leaf of the reticulated cherry. Figure 4.2 (b) depicts a reticulated cherry $\{x, y\}$ where v is the reticulation vertex, x is the tree leaf and y is the reticulation leaf. Note that the distance between u and x is equal to the distance between u and y because of the ultrametric property. *Reducing a reticulated*

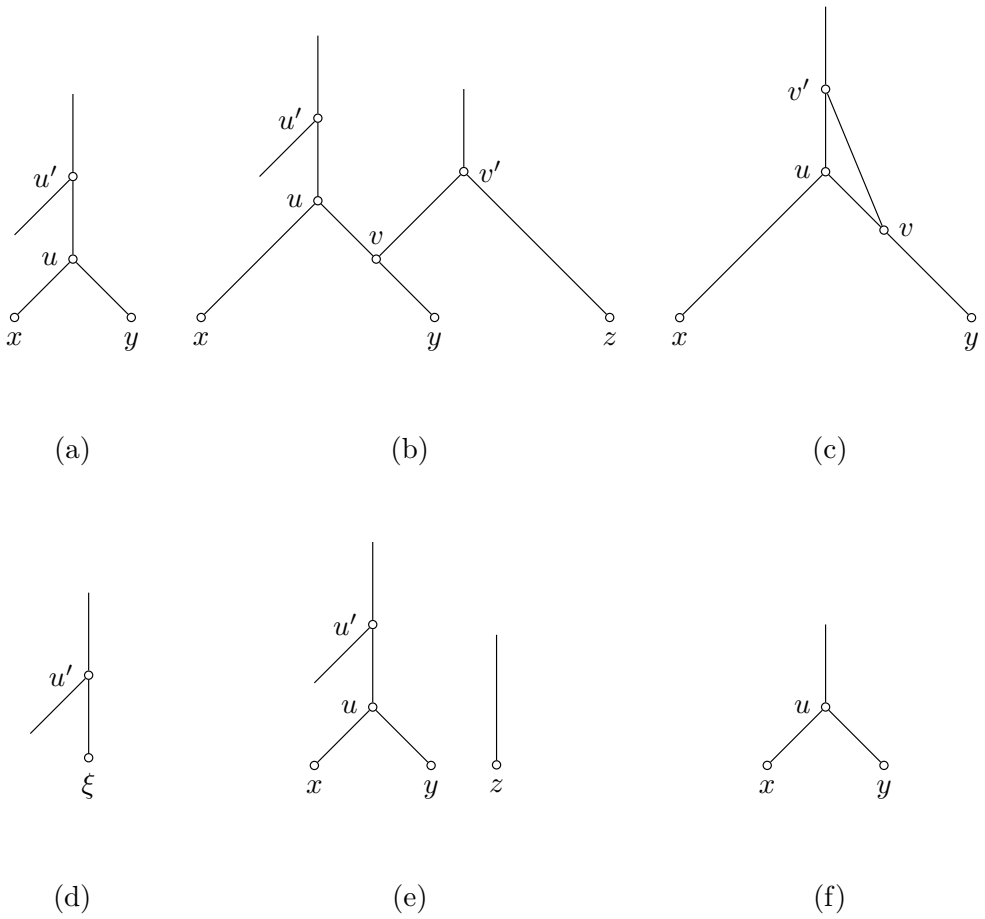


Figure 4.2 (a) cherry; (b) reticulated cherry; (c) reticulated cherry with immediate reticulation; (d) reduced cherry; (e) reduced reticulated cherry; (f) reduced reticulated cherry with immediate reticulation.

cherry $\{x, y\}$ is the operation of deleting the incoming arc to the reticulation vertex that is not part of the reticulated cherry (i.e. the incoming arc that is not (u, v)) and suppressing the degree-two vertices resulting from the deletion, see Fig 4.2(e). Observe that, by reducing a reticulated cherry, the number of reticulations in the resulting network is reduced by one, but the number of leaves and, in particular, the leaf set, is unchanged. An immediate reticulation is a special case of a reticulation, and the reduction of an immediate reticulation is the same as for a normal reticulation. Figure 4.2(c) shows an immediate reticulation, and Figure 4.2(f) shows the result of reducing this immediate reticulation.

Note that the above definition is different from the reticulated cherry reduction used in [14] since they delete the arc (u, v).

4.1.1.3 MAIN RESULT

The main result of this section is that given a set-distance matrix $\mathcal{D}$, if there is an ultrametric tree-child network $\mathcal{N}$ that displays $\mathcal{D}$ then, through a process of identifying cherries and reticulated cherries and reducing them, we can (essentially) determine $\mathcal{N}$ in polynomial time. Before expressing our main theorem, the following theorem is helpful.

It was already known that a tree-child network with every edge weight 1 can be reconstructed from the *multiset matrix* [13] (up to the direction of an immediate reticulation at the root).

Theorem 4.1.1. *[13] Let $\mathcal{D}$ be a multiset-matrix of distances between elements of a set X. If there is a binary tree-child network $\mathcal{N}$ on X displaying $\mathcal{D}$, with no arc joining the two children of the root then, up to isomorphism, $\mathcal{N}$ is the unique binary phylogenetic network on X displaying $\mathcal{D}$, in which case $\mathcal{N}$ can be found in time quadratic in $|\mathcal{D}|$.*

Our result generalises this to arbitrary non-negative edge weights and reconstructing from the set-distance matrix; however, this comes at the cost of restricting attention to only *ultrametric* tree-child networks.

Theorem 4.1.2. *Given a set-distance matrix $\mathcal{D}$ on X, if there is an ultrametric tree-child network $\mathcal{N}$ that displays $\mathcal{D}$, then $\mathcal{N}$ is the unique such network (up to $\equiv$) and may be found in polynomial time.*

The remainder of the section is organised as follows. In Subsection 4.1.2, we describe the algorithm NETWORKUPGMA that is central to the subsection. In Subsection 4.1.3 we show that the algorithm is correct, and in Subsection 4.1.4 we show that the algorithm's running time is polynomial in the number of taxa $|X|$.

4.1.2 THE ALGORITHM NETWORKUPGMA

In this subsection, we present the algorithm NETWORKUPGMA for reconstructing an ultrametric tree-child network from a set-distance matrix of inter-taxa distances.

For a set X and a set-distance matrix $\mathcal{D}$ of distances on X, the algorithm NETWORKUPGMA applied to input X and $\mathcal{D}$ works by recursively finding a pair of elements $x, y \in X$ that form a cherry or a reticulated cherry. After finding the pair x, y, the algorithm reduces $\{x, y\}$, updates X and $\mathcal{D}$, and repeats. Eventually, NETWORKUPGMA either reduces X to a singleton or determines that there is no pair of leaves yielding a cherry or reticulated cherry. If the former holds, then the algorithm works backwards and reconstructs an ultrametric tree-child network on X and checks that this displays $\mathcal{D}$. If this succeeds, the constructed network is the unique (up to equivalence under $\equiv$) ultrametric tree-child network on X displaying $\mathcal{D}$. If the latter holds or the reconstruction fails to display $\mathcal{D}$, then there is no ultrametric tree-child network on X displaying $\mathcal{D}$. The algorithm relies heavily on being able to recognise a cherry or reticulated cherry just from the distance information $\mathcal{D}$.

Now we are in a position to present NETWORKUPGMA formally. The main body of the NETWORKUPGMA algorithm looks for a pair $\{x, y\}$ which form a cherry or reticulated cherry. If such a pair is found, the algorithm forms a set of elements X' and a set-distance matrix $\mathcal{D}'$ resulting from reducing this cherry or reticulated cherry. It then makes a recursive call to NETWORKUPGMA$(X', \mathcal{D}')$. If this yields a suitable network $\mathcal{N}'$ displaying $\mathcal{D}'$ then a subroutine REVERSEREDUCTION is called, which reconstructs $\mathcal{N}$ by reversing the cherry reduction on $\mathcal{N}'$. Finally we need to check that the resulting network does display $\mathcal{D}$ before returning the network $\mathcal{N}$ (see Figure 4.6 in Section 4.1.3 for an example illustrating why).

The pseudocode of NETWORKUPGMA is given in Algorithm 1 (see also Figure 4.3), and the pseudocode of the subroutine REVERSEREDUCTION is given in Algorithm 2.

Algorithm 1 NETWORKUPGMA

Input: A set-distance matrix $\mathcal{D}$ on a finite set X

Output: An ultrametric tree-child network $\mathcal{N}$ displaying $\mathcal{D}$, or **Network not found** if no such network exists

if $|X| = 1$ **then**
  return $\mathcal{N}$: a single vertex labelled with the element of X
else if $|X| = 2$ **and** $|\mathcal{D}_{x,y}| = 1$ **then**
  return $\mathcal{N}$: a cherry on two leaves with both arcs of weight $d_{x,y}/2$
else if $|X| = 2$ **and** $|\mathcal{D}_{x,y}| = 2$ **then**
  let $\{x, y\} = X$ and $\{d_1, d_2\} = \mathcal{D}_{x,y}$ such that $d_1 < d_2$
  return $\mathcal{N}$ on two leaves $\{x, y\}$ as given in Figure 4.3
else if $|X| = 2$ **and** $|\mathcal{D}_{x,y}| > 2$ **then**
  return "Network not found"
  if there is a pair $x, y \in X$ such that $\{x, y\}$ forms a cherry **then**
    $X' = (X - \{x, y\}) \cup \{\xi\}$, where $\xi \notin X$
    ▷ Create the set-distance matrix $\mathcal{D}'$ on X' as follows:
    $\mathcal{D}'_{v,w} = \mathcal{D}_{v,w}$ if $v, w \in X - \{x, y\}$
    $\mathcal{D}'_{\xi,v} = \mathcal{D}'_{v,\xi} = \mathcal{D}_{x,v}$ if $v \in X - \{x, y\}$.
  else if there is a pair $x, y \in X$ such that $\{x, y\}$ forms a reticulated cherry with an immediate reticulation **then**
    $X' = X$
    ▷ Create the set-distance matrix $\mathcal{D}'$ on X' as follows:
    $\mathcal{D}'_{x,y} = \{d_{x,y}\}$
    $\mathcal{D}'_{v,w} = \mathcal{D}'_{v,w}$ for all pairs $\{v, w\} \neq \{x, y\}$.
  else if there is a pair $x, y \in X$ such that $\{x, y\}$ forms a reticulated cherry with y the reticulation leaf **then**
    $X' = X$
    ▷ Create the set-distance matrix $\mathcal{D}'$ on X' as follows:
    for all $v \in X - \{x, y\}$ **do**
      let $\{d_1, d_2, ..., d_k\} = \mathcal{D}_{x,v}$ and $\{d'_1, d'_2, ..., d'_l\} = \mathcal{D}_{y,v} - \mathcal{D}_{x,v}$
      $\mathcal{D}'_{v,w} = \mathcal{D}_{v,w}$ if $v, w \in X - \{y\}$
      $\mathcal{D}'_{y,v} = \mathcal{D}'_{v,y} = \mathcal{D}_{x,v}$ if $v \in X - \{y\}$
      $\mathcal{D}'_{x,y} = \mathcal{D}'_{y,x} = d_{x,y}$.
    end for
  else
    return "Network not found".
  end if
end if

Algorithm 1 NETWORKUPGMA (continued)

if NETWORKUPGMA$(X', \mathcal{D}')$ == "Network not found" **then**
    return "Network not found"
else
    let $\mathcal{N}' =$ NETWORKUPGMA$(X', \mathcal{D}')$
    let $\mathcal{N} =$ REVERSEREDUCTION$(\mathcal{D}, \mathcal{N}', X', \mathcal{D}', x, y)$.
    if $\mathcal{N}$ displays $\mathcal{D}$ **then**
        return $\mathcal{N}$
    else
        return "Network not found"
    end if
end if

Algorithm 2 REVERSEREDUCTION

Input: A set-distance matrix $\mathcal{D}$ on a finite set X, and a set-distance matrix $\mathcal{D}'$ on a finite set X', a phylogenetic network $\mathcal{N}'$, a pair of leaves x, y
Output: Phylogenetic network $\mathcal{N}$, or **Network not found**
if $X' = (X - \{x, y\}) \cup \{\xi\}$ **and** $l_{(\xi', \xi)} > d_{x,y}/2$ where the parent of ξ is ξ' **then**
▷ Reversing a cherry reduction
    form $\mathcal{N}$ from $\mathcal{N}'$ by appending leaves x, y as a children of ξ
    set $l_{\mathcal{N}}(\xi, x) = l_{\mathcal{N}}(\xi, y) = d_{x,y}/2$
    set $l_{\mathcal{N}}(\xi', \xi) = l_{\mathcal{N}'}(\xi', \xi) - d_{x,y}/2$
    for all other edges e set $l_{\mathcal{N}}(e) = l_{\mathcal{N}'}(e)$
    return $\mathcal{N}$
else if $|\mathcal{D}_{x,y}| = 2$ **and** $\mathcal{D}_{x,z} = \mathcal{D}_{y,z}$ for all $z \in X - \{x, y\}$ **then**
▷ Reversing an immediate reticulated cherry reduction
    form $\mathcal{N}$ from $\mathcal{N}'$ as follows:
    let the common parent of x, y in $\mathcal{N}'$ be u, and its parent be u'
    subdivide the arc (u', u) with a new vertex v'
    subdivide the arc (u, y) with a new vertex v
    add an arc (v', v)
    let $d^*_{x,y} = \mathcal{D}_{x,y} - d_{x,y}$
    $l_{\mathcal{N}}(u, v) = 0$
    $l_{\mathcal{N}}(u, x) = l_{\mathcal{N}}(v, y) = d_{x,y}/2$
    $l_{\mathcal{N}}(v', v) = l_{\mathcal{N}}(v', u) = (d^*_{x,y} - d_{x,y})/2$
    $l_{\mathcal{N}}(u', v') = l_{\mathcal{N}'}(u', u) - l_{\mathcal{N}}(v', u)$
    for all other edges e set $l_{\mathcal{N}}(e) = l_{\mathcal{N}'}(e)$
    return $\mathcal{N}$

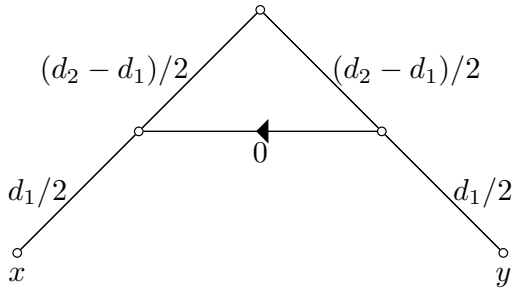


Figure 4.3 The unique (up to $\equiv$) ultrametric tree-child network on two leaves x, y, such that there are two distinct distances d_1, d_2 between the leaves. See Algorithm 1, Line 7.

Algorithm 2 REVERSEREDUCTION (continued)

else
▷ Reversing a (not immediate) reticulated cherry reduction
  let $Z = \{z \in X' : \mathcal{D}'_{x,z} = \mathcal{D}'_{y,z} = \mathcal{D}_{x,z}$ **and** $|\mathcal{D}_{y,z}| = |\mathcal{D}'_{y,z}| + 1\}$
  for $z \in Z$ **do**
    let $d^*_{y,z}$ be the unique value in $\mathcal{D}_{y,z} - \mathcal{D}'_{y,z}$
    if in some $\mathcal{N}'' \equiv \mathcal{N}'$ there is an arc (a, b) such that: b is a tree vertex, the path from b to z is a tree-path and $d_{b,z} < d^*_{y,z}/2 < d_{a,z}$ **then**
      form $\mathcal{N}$ from $\mathcal{N}''$ as follows:
      subdivide the incoming arc to y in $\mathcal{N}''$ with a new vertex v
      subdivide the arc (a, b) with new vertex v'
      add an arc (v', v)
      $l_{\mathcal{N}}(v', b) = d^*_{y,z}/2 - d_{b,z}$
      $l(a, v') = l_{\mathcal{N}''}(a, b) - l_{\mathcal{N}}(v', b)$
      if $d_{x,y} < d^*_{y,z}$ **then**
        $l(u, v) = 0, l(v, y) = d_{x,y}/2$
        $l(v', v) = (d^*_{y,z} - d_{x,y})/2$
      else if $d_{x,y} \geq d^*_{y,z}$ **then**
        $l(v', v) = 0, l(v, y) = d^*_{y,z}/2$
        $l(u, v) = (d_{x,y} - d^*_{y,z})/2$
      end if
      return $\mathcal{N}$
    end if
  end for
end if

4.1.3 PROOF THAT NETWORKUPGMA IS CORRECT

The following lemmas establish that the various steps in the algorithms work and can be accomplished in polynomial time. The first lemma, from [13], shows that every tree-child network contains either a cherry or reticulated cherry. After that, we present the proof of Lemma 4.1.4, which shows that we can recognise a cherry, immediate reticulation or reticulated cherry in an ultrametric tree-child network. Then we present lemmas showing that we can modify the set-distance matrix appropriately to effect an appropriate reduction in each case, and that we can also reverse the reduction once we have a network displaying the reduced set-distance matrix.

Lemma 4.1.3. *[13] Let $\mathcal{N}$ be a tree-child network on X. If $|X| \geq 2$, then $\mathcal{N}$ contains either a cherry or a reticulated cherry.*

The above lemma establishes that every tree-child network contains either a cherry or reticulated cherry; Lemma 4.1.4 stated that we can identify a pair of leaves involved in a cherry or reticulated cherry (with or without immediate reticulation), and moreover which of the cases it is. We now present the lemma to recognise a cherry or reticulated cherry from the distance information.

Lemma 4.1.4. *Let $\mathcal{N}$ be an ultrametric tree-child network on X, and let $\mathcal{D}$ be the set-distance matrix of inter-taxa distances of $\mathcal{N}$. A pair of leaves x, y form a cherry or reticulated cherry if and only if there is a leaf z such that*

$$d_{x,y} < d_{x,z} : \forall z \in X - \{x, y\}.$$

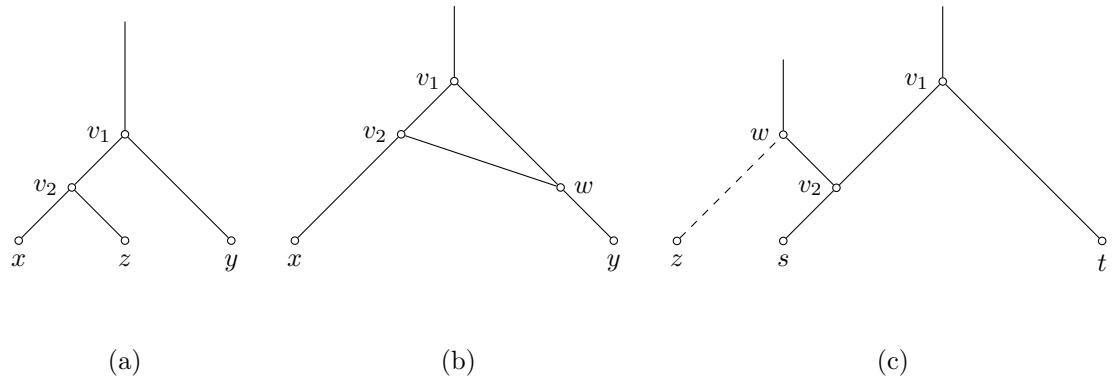
Moreover such a pair x, y:

(i) forms a cherry if and only if $|\mathcal{D}_{x,y}| = 1$.

(ii) forms a reticulated cherry in which the reticulation vertex is an immediate reticulation if and only if $|\mathcal{D}_{x,y}| = 2$ and $\mathcal{D}_{x,z} = \mathcal{D}_{y,z} : \forall z \notin \{x, y\}$.

(iii) forms a reticulated cherry of $\mathcal{N}$ without immediate reticulation, with y the reticulation leaf, if and only if $\mathcal{D}_{x,z} \subseteq \mathcal{D}_{y,z}$ for all $z \notin \{x, y\}$, Furthermore, there exists a leaf z such that $|\mathcal{D}_{x,z}| = |\mathcal{D}_{y,z}| - 1$.

Furthermore we can recognise which of these cases occurs in polynomial time (in the size of X).



(a) (b) (c)

Figure 4.4 The situation when: (a) there is a descendant leaf z of v_1 such that $z \notin \{x, y\}$; (b) there is a tree vertex v_2 that is a descendant of v_1; (c) there is no tree vertex that is a descendant of v_1 but there is a reticulation vertex v_2 that is a child of v_1, also note that $\{s, t\} = \{x, y\}$.

Proof: If $\{x, y\}$ do form a cherry, immediate reticulation or reticulated cherry, then it is easy to verify that the claimed conditions do hold. We therefore concentrate on proving that if the stated conditions hold, then $\{x, y\}$ must indeed be a cherry/immediate reticulation/reticulated cherry.

Let vertex v_1 be a least common ancestor of leaves x and y such that v_1 is at minimal distance from the root. Suppose there is a descendant leaf z of v_1 such that $z \notin \{x, y\}$ as shown in Figure 4.4 (a). Then there is an up-down path from x to z that has peak either v_1 or a descendant of v_1. Thus by the ultrametric property, $d_{x,z} \leq 2d_{x,v_1} = d_{x,y}$. This contradicts the condition $d_{x,y} < d_{x,z} : \forall z \notin \{x, y\}$, thus we may conclude that if the condition holds, there are no descendant leaves of v_1 except x, y.

Suppose there is a tree vertex v_2 that is a descendant of v_1, and without loss of generality take v_2 to be a tree vertex at maximal distance from v_1. Since in a tree-child network every tree vertex has a tree-path to a leaf, the two children of v_2 must either be leaves or have tree-paths to leaves. They cannot have tree-paths to the same leaf, since it would require a reticulation where the paths meet. Since there are no descendant leaves of v_1 except x and y, it must be that one child of v_2 has a tree-path to x and the other a tree-path to y and there are no other descendant leaves of v_2. If v_2 is on the path from v_1 to x, then directed path from v_2 to y must join the path from v_1 to y at a reticulation w as shown in Figure 4.4 (b). By the tree-child property, the child of w must be a tree vertex or leaf, and by v_2's maximality of distance from v_1, it must be a leaf, thus y. Again by the tree-child property, the other child of v_2 is a tree vertex or leaf, and by maximality of distance from v_1, it must be a leaf, therefore

x. There can be no other tree vertex that is a descendant of v_1, as it could not have a tree-path to y since w is the parent of y; thus w is an immediate reticulation with parents v_1 and v_2. Since v_2 is a descendant of v_1, the paths with peaks v_1 and v_2 have different weights and so $|\mathcal{D}_{x,y}| = 2$. If v_2 is on the path from v_1 to y, then it gives the equivalent network obtained by reversing the direction of the arc (v_2, w).

Suppose now that there is no tree vertex that is a descendant of v_1 but that there is a reticulation vertex v_2 that is a child of v_1. Observe that by the tree-child property applied to v_1, there can be reticulations only down the path to x or the path to y, not both, and by the tree-child property applied v_2, there can be no other reticulations that are descendants of v_1. Thus x and y must form a reticulated cherry. Let s denote the element of $\{x, y\}$ such that v_2 is on the path from v_1 to s. Let w be the other parent of v_2 (i.e. $w \neq v_1$), and let z be a leaf that is reached by a tree-path from w as shown in Figure 4.4 (c). Observe that w cannot be a parent of v_1 because then w would be a least common ancestor of x and y at shorter distance from the root than v_1. Then every path from s to z is either the (unique) path P that starts s, v_2, w, or is a path via v_1. Note also that any path via v_1 must also pass through w and therefore (by the ultrametric property) is longer than the path P. Thus $|\mathcal{D}(s, z)| = |\mathcal{D}(t, z)| - 1$, where t is the element in $\{x, y\} - \{s\}$.

Thus if $d_{x,y} < d_{x,z} : \forall z \notin \{x, y\}$, then either there are no tree vertices or reticulations below v_1, in which case i follows, or there is a tree vertex below v_1, in which case ii follows, or there are no tree vertices below v_1, but there is a reticulation vertex, in which case iii follows. For each pair $x, y \in X$ we can check if $d_{x,y} < d_{x,z} : \forall z \notin \{x, y\}$ in polynomial time. Determining which of the 3 subsequent cases holds is then a matter of comparing sets of polynomial size, which can also be done in polynomial time. ■

The next lemmas establish the effect of reducing a cherry or a reticulated cherry on the set-distance matrices. Recall that two networks are equivalent under $\equiv$ if one can be obtained from the other by adjusting weights at reticulations and flipping the direction of immediate reticulations (see Section 4.1.1.1). We show that reducing a cherry or reticulated cherry has a deterministic effect on the set-distance matrix/ Moreover if, up to equivalence under $\equiv$, there is a unique ultrametric tree-child network that displays the reduced set-distance matrix, then there is, up to equivalence under $\equiv$, a unique ultrametric tree-child network that displays the original set-distance matrix.

Lemma 4.1.5. *If $\mathcal{N}_1 \equiv \mathcal{N}_2$ and $\mathcal{N}_1$ is an ultrametric tree-child network, then $\mathcal{N}_2$ is an ultrametric tree-child network. Also, $\mathcal{N}_1$ and $\mathcal{N}_2$ display the same set-distance matrix.*

Lemma 4.1.6. *Let $\mathcal{N}$ be an ultrametric tree-child network on $|X| > 2$. Let $\mathcal{D}$ be the set-distance matrix of inter-taxa distances of $\mathcal{N}$. Let $\{x, y\}$ be a cherry of $\mathcal{N}$ with common parent v, so that $d_{x,y} = 2 \times d_{v,x}$. Let $X' = (X - \{x, y\}) \cup \{\xi\}$ and $\mathcal{D}'$ be the set-distance matrix of inter-taxa distances on X' given by $\mathcal{D}'_{z,z'} = \mathcal{D}_{z,z'}$ if $z, z' \in X - \{x, y\}$, and $\mathcal{D}'_{z,\xi} = \mathcal{D}_{z,x}$ if $z \in X - \{x, y\}$. Then the following hold:*

(i) $\mathcal{D}'$ is displayed by the ultrametric tree-child network $\mathcal{N}'$ on X' obtained from $\mathcal{N}$ by reducing the cherry $\{x, y\}$, where the new leaf is labelled ξ.

(ii) Moreover, if $\mathcal{N}'$ is the unique ultrametric tree-child network on X' displaying $\mathcal{D}'$ up to equivalence under $\equiv$, then $\mathcal{N}$ is the unique ultrametric tree-child network on X displaying $\mathcal{D}$ up to equivalence under $\equiv$.

Proof: Let w be the parent of v. We reduce the cherry by deleting leaf y and its incident edge, and suppressing the degree two vertex v and relabelling the leaf x as ξ. Set the weight of the edge (w, ξ) to be $d_{w,x}$ to obtain $\mathcal{N}'$. Thus, for all $z, z' \in X - \{y\}$ the set of path distances between z and z' is unchanged by the reduction. Hence $\mathcal{D}'$ is displayed by the network $\mathcal{N}'$ on X'.

For (ii), suppose $\mathcal{N}'$ is the unique (up to $\equiv$) ultrametric tree-child network displaying $\mathcal{D}'$, and let $\mathcal{N}_1$ be an ultrametric tree-child network on X displaying $\mathcal{D}$. By Lemma 4.1.4, $\{x, y\}$ is a cherry in $\mathcal{N}_1$. Furthermore, by (i), the network $\mathcal{N}'_1$ on X obtained from $\mathcal{N}_1$ by reducing the cherry $\{x, y\}$ also displays $\mathcal{D}'$. Therefore, by the assumption in the statement of part ii, $\mathcal{N}'_1 \equiv \mathcal{N}'$. Since the pair x, y are not involved in any reticulations, it follows that $\mathcal{N}_1 \equiv \mathcal{N}$. ■

Lemma 4.1.7. *Let $\mathcal{N}$ be an ultrametric tree-child network, and let $\mathcal{D}$ be the set-distance matrix of inter-taxa distances of $\mathcal{N}$. Let $\{x, y\}$ be a reticulated cherry in which the reticulation vertex is an immediate reticulation, with v the reticulation, u the parent and sibling of v, and v' the parent of u and v (See Figure 4.2(c)). Let $\mathcal{D}'$ be the set-distance matrix of inter-taxa distances on X given by*

$$\mathcal{D}'_{x,y} = \{d_{x,y}\}$$

and

$$\mathcal{D}'_{z,z'} = \mathcal{D}_{z,z'}$$

for $\{z, z'\} \in X - \{x, y\}$. Then the following hold:

(i) $\mathcal{D}'$ is displayed by the ultrametric tree-child network on $\mathcal{N}'$ on X obtained from $\mathcal{N}$ by reducing the reticulated cherry $\{x, y\}$.

(ii) If $\mathcal{N}'$ is the unique ultrametric tree-child network displaying $\mathcal{D}'$, up to equivalence under $\equiv$, then, $\mathcal{N}$ is the unique ultrametric tree-child network on X displaying $\mathcal{D}$, up to equivalence under $\equiv$.

Proof: We reduce the reticulated cherry by removing arc (v', v). This leaves only a single up-down path between x and y, and by the ultrametric property it is the shorter of the original paths, having weight $d_{x,y}$. For all other paths between pairs $z, z' \in X$, either the path did not use arc (v', v), in which case it is unchanged, or it had the same weight as an equivalent path traversing arcs $(v', u), (u, v)$, which still exists after the reduction. Hence $\mathcal{D}'$ is displayed by the network $\mathcal{N}'$ on X'.

For (ii), suppose $\mathcal{N}'$ is the unique (up to $\equiv$) ultrametric tree-child network displaying $\mathcal{D}'$, and let $\mathcal{N}_1$ be an ultrametric tree-child network on X displaying $\mathcal{D}$. By Lemma 4.1.4, $\{x, y\}$ is a reticulated cherry with immediate reticulation in $\mathcal{N}_1$. Furthermore, by (i), the network $\mathcal{N}'_1$ on X obtained from $\mathcal{N}_1$ by reducing the reticulated cherry $\{x, y\}$ also displays $\mathcal{D}'$. Therefore, by the assumption in the statement of (ii), $\mathcal{N}'_1 \equiv \mathcal{N}'$. Since each of these networks was formed by the removal of a single arc subdividing the incoming arcs to y and to its parent, it follows that $\mathcal{N}_1 \equiv \mathcal{N}$. ■

Lemma 4.1.8. *Let $\mathcal{N}$ be an ultrametric tree-child network, and let $\mathcal{D}$ be the set-distance matrix of inter-taxa distances of $\mathcal{N}$. Let $\{x, y\}$ be a reticulated cherry of $\mathcal{N}$ with y the reticulation leaf, and not part of an immediate reticulation. Let $\mathcal{D}'$ be the set-distance matrix of inter-taxa distances on X given by $\mathcal{D}'_{x,y} = \{d_{x,y}\}$, $\mathcal{D}'_{y,z} = \mathcal{D}_{x,z}$ for $z \in X - \{x, y\}$ and $\mathcal{D}'_{z,z'} = \mathcal{D}_{z,z'}$ for $z, z' \in X - \{y\}$. Then the following hold:*

(i) $\mathcal{D}'$ is displayed by the ultrametric tree-child network on $\mathcal{N}'$ on X obtained from $\mathcal{N}$ by reducing the reticulated cherry.

(ii) If $\mathcal{N}'$ is the unique ultrametric tree-child network displaying $\mathcal{D}'$, up to equivalence under $\equiv$, then $\mathcal{N}$ is the unique ultrametric tree-child network on X displaying $\mathcal{D}$, up to equivalence under $\equiv$.

Proof: Let u be the parent of x and v be the parent of y in $\mathcal{N}$, as shown in Figure 4.5. Since u is a tree vertex, it has a unique parent u'. Since v is a reticulation vertex, it has a parent v' additional to u, and v' has a tree-path to a leaf z. The reduction of the reticulated cherry involves removing the arc (v', v) and suppressing the resulting degree 2 vertices v and v'. Intuitively, we delete v and v', and their incident arcs, and introduce arcs (u, y) and (b, a), where b is the parent of v', and a is the other child of v'.

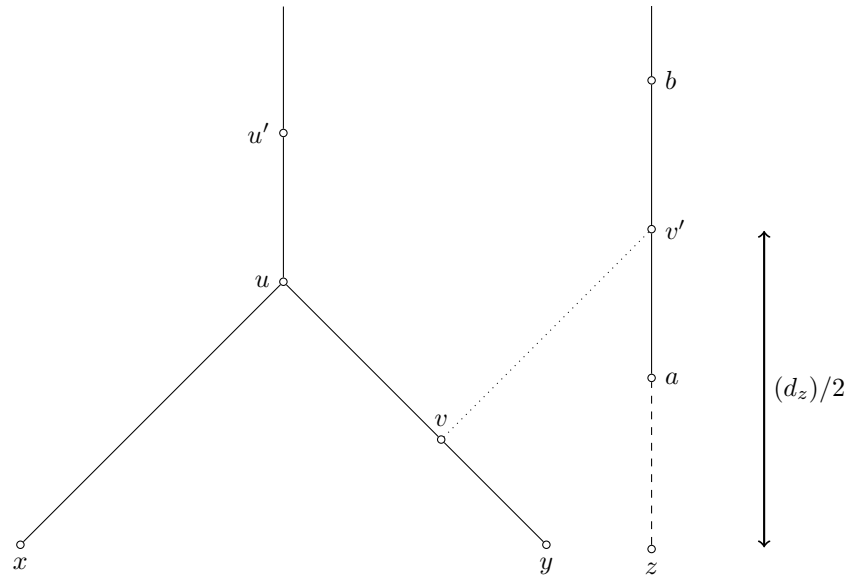


Figure 4.5 A reticulated cherry $\{x, y\}$. Reducing the reticulated cherry involves deleting the arc (v', v), shown with dotted lines, and suppressing the degree-2 vertices v', v. Since there is a tree-path in $\mathcal{N}$ from v' to a leaf z, there is an additional distance d_z in $\mathcal{D}_{y,z}$ that is not in $\mathcal{D}_{x,z}$.

For (i), consider first the up-down paths from x to y in $\mathcal{N}$. The up-down paths present in $\mathcal{N}$ but not $\mathcal{N}'$ between x and y are precisely those that use the arc (v', v). The remaining up-down path between x and y is unique and preserves the shortest weight $d_{x,y}$. (Since all up-down paths between x and y that use the arc (v', v) pass through the ancestor of u, they must be longer than $d_{x,y}$.)

Now consider the up-down paths between y and $z \neq x$ in $\mathcal{N}$. Every up-down path between y and z does exactly one of the following: either passes through arc (v', v), in which case the path is not in $\mathcal{N}'$, or it passes through arc (u', u) in which case the weight of the path is equal to some $d \in d_{x,z}$. Moreover, for any path from x to z, there is an equal weight path from y to z because $d_{u,x} = d_{u,y}$. This concludes the proof of part (i).

For the proof of (ii), let $\mathcal{N}_1$ be some ultrametric tree-child network that displays $\mathcal{D}$. Then by Lemma 4.1.4, $\{x, y\}$ form a reticulated cherry in $\mathcal{N}_1$ where there is no immediate reticulation and y is the reticulation leaf. Thus if we reduce this reticulated cherry we will obtain a network $\mathcal{N}_1'$ which displays $\mathcal{D}'$ and is therefore equivalent to $\mathcal{N}'$. Observe that $\mathcal{N}_1'$ differs from $\mathcal{N}_1$ by the removal of a single arc (v_1', v_1) whose head v_1 was a parent of y. We now show that there is only one possible position in $\mathcal{N}_1'$ for the tail v_1' of this arc to have been in order for the matrix $\mathcal{D}$ to be displayed by $\mathcal{N}'$.

In $\mathcal{N}$, $\{x, y\}$ form a reticulated cherry with y the reticulation leaf and (v', v) the arc deleted in forming $\mathcal{N}'$. There is some $z \in X - \{x, y\}$ that is a tree-path descendant of v' in $\mathcal{N}$. Thus there is a unique additional distance between y and z in $\mathcal{D}$ (compared to $\mathcal{D}'$), i.e. $\mathcal{D}_{x,z} \subset \mathcal{D}_{y,z}$ and $|\mathcal{D}_{x,z}| = |\mathcal{D}_{y,z}| - 1$. Let this additional distance be d_z. In $\mathcal{N}_1'$ it must be that v_1' was also at height exactly $d_z/2$ in order for $\mathcal{N}_1'$ to display $\mathcal{D}'$.

If there were a unique arc in $\mathcal{N}_1'$ such that a point on it was at height $d_z/2$ above z, then we would be done. In $\mathcal{N}'$, this arc is unique, since the path from v' to z is a tree-path. However $\mathcal{N}_1'$ is only equivalent to $\mathcal{N}'$ under $\equiv$. Therefore *a priori* the arc may not be unique in $\mathcal{N}_1'$ for one of two reasons. Firstly, in $\mathcal{N}$ vertex v' was the child of a reticulation b and, under the equivalence up to weights at reticulations, in $\mathcal{N}_1'$ the outgoing edge from b was reduced (in weight) and the incoming edges increased, so that the point $d_z/2$ above z is now above b. However both parents of b must have tree-paths to some leaves z', z'' respectively. In $\mathcal{N}_1$, if we had subdivided either incoming edge to b in placing v_1', then either z' or z'' does not have a path to y via b, and misses out on a path weight that is present in $\mathcal{N}$, and therefore $\mathcal{D}$. This contradicts $\mathcal{N}_1$ displaying $\mathcal{D}$, so it cannot happen.

Secondly, it might be that in $\mathcal{N}'$, the arc (b, a) is the arc of an immediate reticulation (in $\mathcal{N}'$) not incoming to the reticulation, but this immediate reticulation was 'flipped' in $\mathcal{N}_1'$ under the equivalence up to immediate reticulations. In this case there is a vertex c such that b and a are both parents of c in $\mathcal{N}'$, but c and b are parents of a in $\mathcal{N}_1'$. Also there is a tree-path from a to z, with c not on this path. In $\mathcal{N}_1'$, vertex v_1' would have to be placed at height $d_z/2$ above z: either on one of the two arcs (b, a) or (c, a) that are incoming to the immediate reticulation, which would contradict the tree-child property as both children of v_1' would be reticulations, or the arc (b, c). However then a new path between y and z with peak b would exist in $\mathcal{N}_1$, for which there is no path of equal weight between y and z in $\mathcal{N}$, contradicting that $\mathcal{N}, \mathcal{N}_1$ both display $\mathcal{D}$. Thus neither of these cases can occur, and since $\mathcal{N}'$ and $\mathcal{N}_1'$ are equivalent, then $\mathcal{N}$ and $\mathcal{N}_1$ are also. ■

Lemma 4.1.9. *Let $\mathcal{N}$ be an ultrametric tree-child network displaying set-distance matrix $\mathcal{D}$, such that leaves x, y form a cherry or reticulated cherry in $\mathcal{N}$. Let $\mathcal{D}'$ and X' be as formed by lines 10-25 of* NETWORKUPGMA*, corresponding to reducing the cherry or reticulated cherry $\{x, y\}$, and let $\mathcal{N}'$ be an ultrametric tree-child network displaying $\mathcal{D}'$. Then Algorithm* REVERSEREDUCTION *applied to $\mathcal{D}, X, \mathcal{D}', X', \mathcal{N}', \{x, y\}$ returns a network equivalent to $\mathcal{N}$ under $\equiv$.*

Proof: First suppose that x, y form a cherry in $\mathcal{N}$. Then by Lemma 4.1.4 and Lemma 4.1.6, $\mathcal{N}'$ is equivalent to a network obtained by reducing the cherry $\{x, y\}$ in $\mathcal{N}$. Hence $|X'| =$

$|X| - 1$, and so lines 2-6 are executed. By construction the arc (ξ', ξ) in $\mathcal{N}'$ has weight greater than $d_{x,y}/2$, and lines 2-5 of REVERSEREDUCTION correctly reconstruct a network $\mathcal{N}_1 \equiv \mathcal{N}$ by Lemma 4.1.6.

Secondly suppose that x, y form a reticulated cherry with immediate reticulation in $\mathcal{N}$. Then by Lemma 4.1.4 and Lemma 4.1.7, $\mathcal{N}'$ is equivalent to a network obtained by reducing the reticulated cherry $\{x, y\}$ in $\mathcal{N}$. Thus $X' = X$ and $\{x, y\}$ form a cherry in $\mathcal{N}'$. Therefore, by Lemma 4.1.4, lines 9-19 of REVERSEREDUCTION are executed, and the resulting network displays $\mathcal{D}$. So by Lemma 4.1.7 the reconstructed $\mathcal{N}_1 \equiv \mathcal{N}$.

Thirdly suppose that x, y form a reticulated cherry without immediate reticulation in $\mathcal{N}$. Then by Lemma 4.1.4 and Lemma 4.1.8, $\mathcal{N}'$ is equivalent to a network obtained by reducing the reticulated cherry $\{x, y\}$ in $\mathcal{N}$. Thus $X' = X$ and $\{x, y\}$ form a cherry in $\mathcal{N}'$. Therefore, by Lemma 4.1.4, lines 21-39 of REVERSEREDUCTION are executed. Since $\mathcal{N}'$ is equivalent to the network $\mathcal{N}''$ obtained by reducing the reticulated cherry $\{x, y\}$ in $\mathcal{N}$, then the set Z at line 21 of REVERSEREDUCTION (that is the set of leaves z which have a single extra distance in $\mathcal{D}_{y,z}$ that is not present in $\mathcal{D}'_{y,z}$) is non-empty, and moreover for at least one $z \in Z$ the arc (a, b) exists (since in $\mathcal{N}$ there is a tree-path to a leaf z from the vertex v' at the tail of the deleted arc). The question remains of whether we can detect the arc given $\mathcal{N}'$ instead of $\mathcal{N}''$. There are two reasons for possible failure. First, the arc (a, b) satisfies $d_{b,z} < d^*_{y,z}/2 < d_{a,z}$ in $\mathcal{N}''$ but not in $\mathcal{N}'$, which may occur if a is a reticulation and under $\equiv_1$ the weights of the arcs into and out of a have been adjusted. However we can easily determine the class representative for $\mathcal{N}'$ under $\equiv_1$, which will satisfy this condition if $\mathcal{N}''$ does, and use that in place of $\mathcal{N}'$. Second, the vertex b may be a reticulation in $\mathcal{N}'$ but not in $\mathcal{N}''$ if it is an immediate reticulation that has been created by reversing an arc in $\mathcal{N}''$ under $\equiv_2$. However we can again easily identify when this occurs, and reverse the incoming arc to b that is not (a, b) in the case that the only reticulation on the path b to z is an immediate reticulation at b. Once such a z is found, there can be only one place to insert an arc to reverse the reduction, by Lemma 4.1.8, and so we correctly reconstruct a network $\mathcal{N}_1 \equiv \mathcal{N}$. ■

Theorem 4.1.10. *Algorithm* NETWORKUPGMA *is correct. Moreover, if Algorithm* NETWORKUPGMA *returns a network $\mathcal{N}$ on input $\mathcal{D}$ then $\mathcal{N}$ is, up to $\equiv$, the unique ultrametric tree-child network displaying $\mathcal{D}$.*

Proof: The proof is by induction on $|X|$. It is straightforward to verify that if $|X| \leq 2$ then NETWORKUPGMA takes the correct action. Assume now that $|X| > 2$ and the algorithm is correct on inputs with fewer than $|X|$ leaves. By Lemma 4.1.4, we can determine if one of the three cases in lines 10, 14, 18 applies, and by Lemmas 4.1.6, 4.1.7, and 4.1.8, assuming

there is a network that displays $\mathcal{D}$, then the correct $\mathcal{D}', X'$ are created corresponding to a network after the appropriate reduction. By Lemma 4.1.3, if there is a tree-child network displaying $\mathcal{D}$, then it contains a cherry or reticulated cherry, so if none is found then we are correct to return "Network not found" in line 27 of NETWORKUPGMA algorithm.

Again by Lemmas 4.1.6, 4.1.7, and 4.1.8, assuming there is an ultrametric tree-child network that displays $\mathcal{D}$, then the recursive call in line 30 would return a valid network, so we are correct to return "Network not found" if the recursive call does not return a network. Finally, in the case that a network $\mathcal{N}'$ is returned, we call REVERSEREDUCTION. If there is a network that displays $\mathcal{D}$, then by Lemma 4.1.9, we reconstruct a valid network displaying $\mathcal{D}$ from $\mathcal{N}'$, and hence return a correct answer. If there is not an ultrametric tree-child network that displays $\mathcal{D}$, then the check in line 38 fails and we correctly return "Network not found". Hence in all cases NETWORKUPGMA is correct.

Finally observe that when NETWORKUPGMA returns a network, it is built up from a network on one or two leaves by successively reversing reductions. Since there is a unique possible network for each case when $|X| = 1$ or $|X| = 2$, and by Lemmas 4.1.6, 4.1.7, and 4.1.8, each reduction reversal results in a unique network (up to $\equiv$), it must be that $\mathcal{N}$ is also unique up to $\equiv$. ■

Note that the final check that the network displays $\mathcal{D}$ is required. Figure 4.6 gives an example of where given input corresponding to a non-ultrametric phylogenetic tree the algorithm would correctly identify and reduce a cherry, reconstruct a network $\mathcal{N}'$ displaying $\mathcal{D}'$, and then reverse the reduction, but there is no valid ultrametric network that displays $\mathcal{D}$. This example also serves to illustrate the extent to which NetworkUPGMA generalises UPGMA: given a set-distance matrix with each set of size 1 that corresponds to an ultrametic tree, then both UPGMA and NetworkUPGMA will return the same correct tree, by Theorem 4.1.10. However, given data that does not a set-distance matrix with each set of size 1 that does not correspond to an ultrametric tree, our algorithm will halt with "Network not found" whereas UPGMA will output a phylogenetic tree that may or may not be close to displaying the distances in the data, but will not display them exactly.

The algorithm NETWORKUPGMA applied to input $X = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8\}$, and a set distance matrix $\mathcal{D}$ on X in Figure 4.7. The algorithm works recursively finding the pair $\{x, y\}$ form a cherry or a reticulated cherry. At first step, by Lemma 4.1.4, conducting three reductions below in one step to obtain $\mathcal{D}^{iii}$ from $\mathcal{D}$:

1. $\{a_1, a_2\}$ form a reticulated cherry without immediate reticulation since $\mathcal{D}_{a_1,a_i} \subseteq \mathcal{D}_{a_2,a_i} : \forall i \in \{3, 4, 5, 6, 7, 8\}$, also there exist a leaf a_3 such that $|\mathcal{D}_{a_1,a_3}| = |\mathcal{D}_{a_2,a_3}| - 1$,

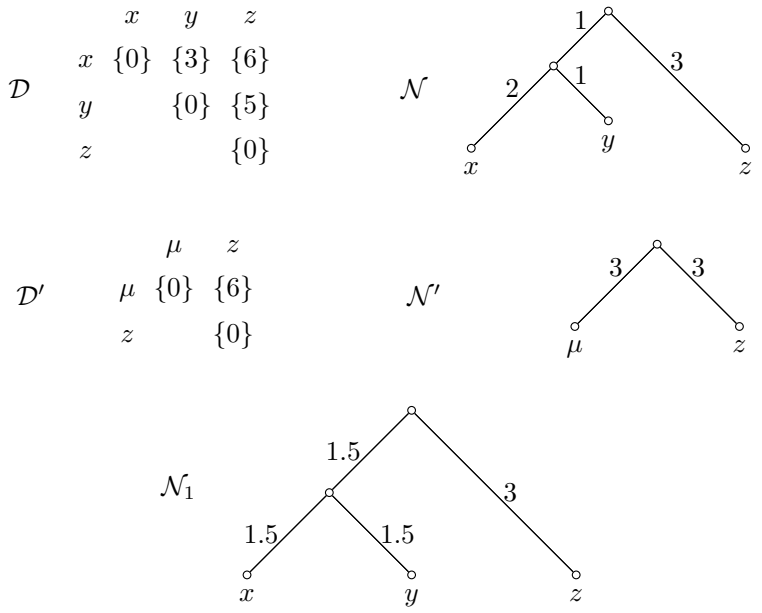


Figure 4.6 An example of an input $\mathcal{D}$ corresponding to a non-ultrametric phylogenetic tree $\mathcal{N}$, the reduced set-distance matrix $\mathcal{D}'$ and corresponding tree $\mathcal{N}'$, and the ultrametric tree $\mathcal{N}_1$ created in the algorithm by reversing the reduction, which is then rejected in the test at line 38 of NETWORKUPGMA.

2. $\{a_4, a_5\}$ form a cherry since $|\mathcal{D}_{a_4,a_5}| = 1$, and

3. $\{a_7, a_8\}$ form a reticulated cherry with immediate reticulation since $|\mathcal{D}_{a_7,a_8}| = 2$ and $\mathcal{D}_{a_7,a_i} = \mathcal{D}_{a_8,a_i} : \forall i \in \{1, 2, 3, 4, 5, 6\}$.

Then the algorithm reduces these pairs to update the matrix as $\mathcal{D}'$ as follows:

1. $\mathcal{D}^i_{a_1,a_2} = \{d_{a_1,a_2}\}$, $\mathcal{D}^i_{a_2,a_i} = \mathcal{D}_{a_1,a_i} : \forall i \in \{3, 4, 5, 6, 7, 8\}$ by Lemma 4.1.8,

2. The new leaf is labelled by b_1 and $\mathcal{D}^{ii}_{a_i,b_1} = \mathcal{D}_{a_i,a_4} : \forall i \in \{1, 2, 3, 6, 7, 8\}$ by Lemma 4.1.6,

3. $\mathcal{D}^{iii}_{a_7,a_8} = \{d_{a_7,a_8}\}$ by Lemma 4.1.7.

The algorithm works similarly until $|X| = 2$ and to obtain the matrix $\mathcal{D}^{viii}$. Then, the algorithm REVERSEREDUCTION applied to input $X' = \{b_5, b_6\}$, and a set distance matrix $\mathcal{D}^{viii}$ on X is used to construct the network $\mathcal{N}^{viii}$, and then reverse each reduction until $\mathcal{N}$ display and $\mathcal{D}$ is obtained.

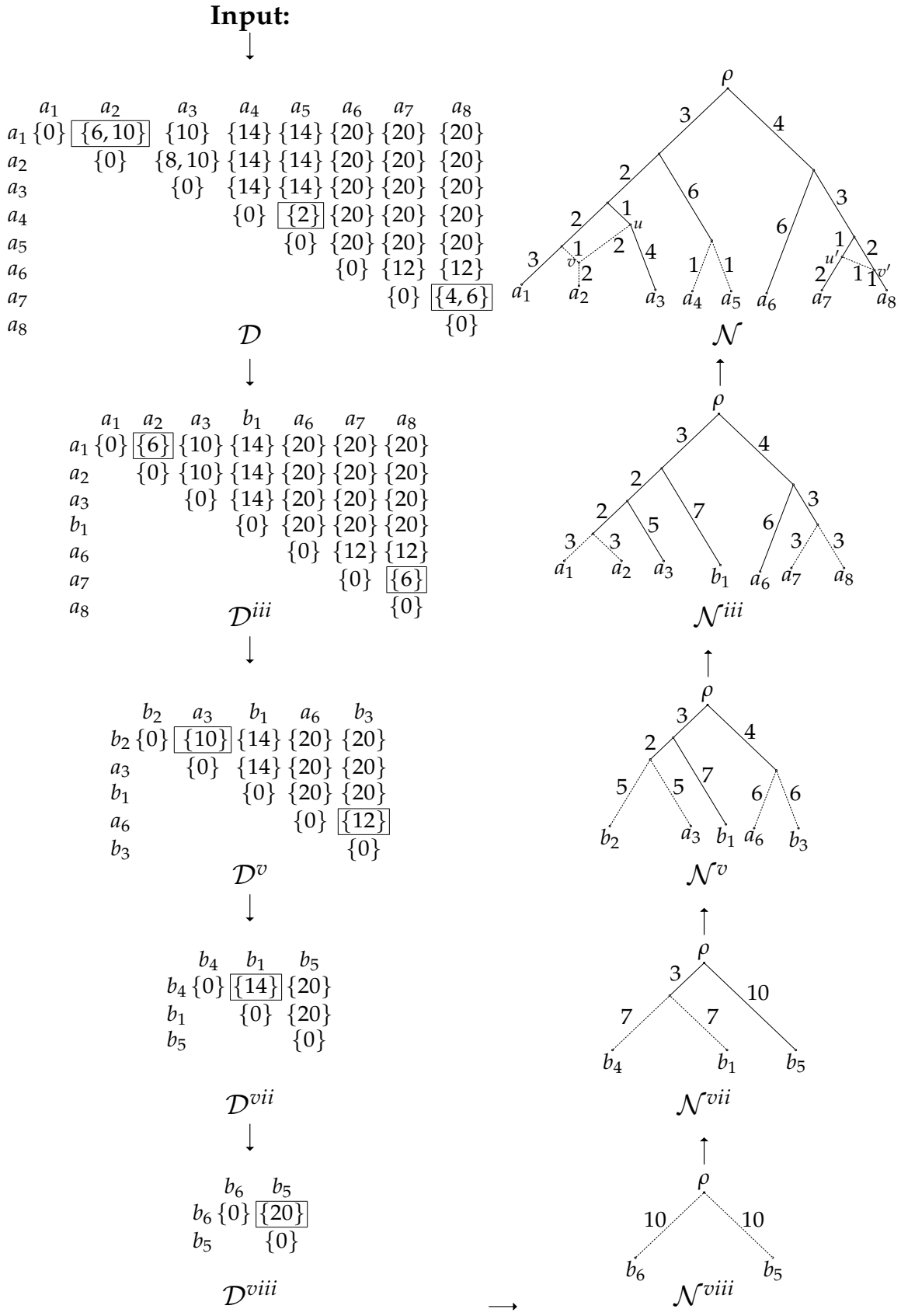


Figure 4.7 Example for NETWORKUPGMA Algorithm. Reductions defined with box of distances.

4.1.4 RUNNING TIME OF NETWORKUPGMA ALGORITHM

In this subsection, we analyse the running time of NETWORKUPGMA. First we consider the size of the input. Typically in phylogenetic algorithms the running time is given in terms of $|X|$, the number of taxa under consideration. Here, the actual input is a set X and a $|X|$ by $|X|$ set-distance matrix $\mathcal{D}$ of inter-taxa distances on X. For all $x, y \in X$, we will assume that each entry $\mathcal{D}_{x,y}$ is presented as a sorted list of distances. The size of each set $\mathcal{D}_{x,y}$ is linear in $|X|$, as the ultrametric condition means that the weight of any up-down path between x and y is twice the distance from x to the peak of the up-down path, and there are less than $3|X|$ internal vertices in $\mathcal{N}$ that could be the peak since $\mathcal{N}$ is tree-child (see [17], Proposition 1). Thus the input (essentially $\mathcal{D}$) has size $O(|X|^3)$, at least when the input is displayed by some tree-child network; excessively large inputs could be rejected out of hand before running the algorithm if need be.

Theorem 4.1.11. *Given a set X and a set-distance matrix $\mathcal{D}$, the algorithm* NETWORKUPGMA *runs in time $O(|X|^4)$.*

Proof: First we consider algorithm REVERSEREDUCTION. Given $\mathcal{N}'$ and x, y, we can easily determine which of the three cases applies (reversing a cherry reduction, immediate reticulation reduction, or reticulated cherry reduction) in time in $|X|$, using Lemma 4.1.4. The most (time) complex of the three is reversing a reticulated cherry. In this case, we can determine the set Z in $O(|X|^2)$ steps, and for each candidate $z \in Z$ we look for an arc at height $d^*_{y,z}/2$ such that there is a tree-path b to z. Technically we need to first make $\mathcal{N}'$ a class representative, but in fact our algorithms only reconstruct class representatives. Also we need to check that b is a tree vertex or immediate reticulation, in the latter case changing the direction or the incoming arc not (a, b). Since $\mathcal{N}'$ is tree-child, it has a linear (in $|X|$) number of arcs each of which we can check in linear time, and so for each z we can check all arcs in $O(|X|^2)$ steps. Overall we identify the correct edge to subdivide and construct $\mathcal{N}$ in $O(|X|^3)$ steps.

Finally we consider algorithm NETWORKUPGMA. Lines 1-7 deal with constant sized X and can be accomplished in constant time. Determining which case to undertake in the next **if** statement (lines 10, 14, 26) can be done in time $O(|X|^3)$, by applying Lemma 4.1.4. Also creating X' and $\mathcal{D}'$ take at most $O(|X|^3)$ steps. The call to REVERSEREDUCTION is also at most $O(|X|^3)$, and finally checking whether $\mathcal{D}$ is displayed by $\mathcal{N}$ is $O(|X|^3)$, since we need only check for each internal vertex of $\mathcal{N}$ whether it is an ancestor of each leaf, and its height, in order to determine the set-distance matrix displayed by $\mathcal{N}$. Thus the work done in NETWORKUPGMA outside of the recursive call, takes at most $O(|X|^3)$ steps. In each

recursive call $\mathcal{D}$ has strictly smaller size, thus the whole algorithm takes at most $O(|X|^6)$ steps. However we can do better than this. Any reduction of a reticulated cherry results in x, y being a cherry in $\mathcal{N}'$, so in fact every other reduction (at least) on an input that is displayed by a tree-child network reduces the size of $|X|$ by one. Thus there are only $2|X|$ recursions, so the entire algorithm completes in $O(|X|^4)$ steps. An additional check that we do not make two reticulated cherry reductions in a row, else return "Network not found", would be needed in the algorithm to obtain this running time for all inputs. ■

Combining Theorems 4.1.10 and 4.1.11 gives Theorem 4.1.2.

The first question related to NETWORKUPGMA algorithm is determining what accuracy guarantees can be given for the new algorithm, and testing its performance on real or simulated data. The next section answers this question.

4.2 SAFETY RADIUS OF NETWORKUPGMA ALGORITHM

In the second part of this chapter, we consider the task of reconstructing phylogenetic *networks* from estimated distance data. The robustness (ability to cope with noise in the data) of distance based reconstruction methods has been considered in a number of papers. Atteson [3] introduced *safety radius*, a measure of the maximum absolute error in any distance estimate for a given algorithm to be guaranteed to return the correct result. Formal definition of safety radius is given in Subsection 4.2.1.1. Gascuel, Pardi and Truszkowski [30] give a recent survey which presents many results in this field. More recently Gascuel and Steel [32] have developed a stochastic safety radius, and show Neighbor Joining has stochastic safety radius close to optimal. In this section we determine the safety radius of the NETWORKUPGMA algorithm.

Theorem 4.2.1. *The algorithm* NETWORKUPGMA *has safety radius* $1/2$.
I.e. Let $\mathcal{N}$ be an ultrametric tree-child network on X with weight function l on the edges and minimum edge weight $l^{\mathcal{N}}_{min}$, and let $\mathcal{D}^{\mathcal{N}}$ be the induced set-distance matrix on X. Let $\mathcal{D}$ be an estimated set-matrix of inter-taxa distances of X with maximum error at most $\epsilon < l^{\mathcal{N}}_{min}/2$. Then NETWORKUPGMA *applied to input $\mathcal{D}$ will return a network $\hat{\mathcal{N}}$ such that as unweighted networks $\hat{\mathcal{N}} \equiv \mathcal{N}$ and also $|\mathcal{D}^{\hat{\mathcal{N}}} - \mathcal{D}^{\mathcal{N}}| < \epsilon$. Moreover,* NETWORKUPGMA *runs in time polynomial in $|X|$.*

The remainder of this section is organised as follows. In Subsection 4.2.1, we give the notation and definitions. Subsection 4.1.2 gives the formal definition of NETWORKUPGMA algorithm, and finally Subsection 4.1.3 gives the lemmas and the proof of the Theorem 4.2.1.

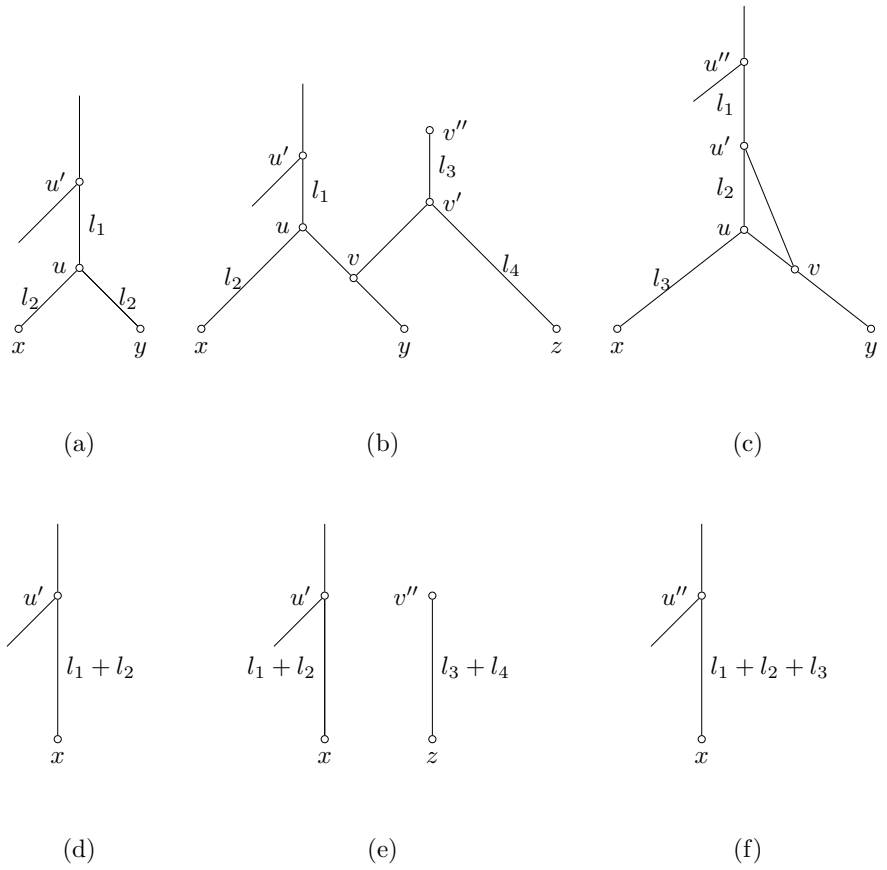


Figure 4.8 (a) cherry, (b) reticulated cherry, (c) reticulated cherry with immediate reticulation. Reduction of (d) cherry, (e) reticulated cherry, (f) reticulated cherry with immediate reticulation.

4.2.1 NOTATION AND DEFINITIONS

In this subsection we give further definitions which we shall require in order to present our algorithm and proof.

Note, a slight modification in immediate reticulation is that the arc (w, v) has weight 0, and is called the *immediate reticulation arc*.

Reducing a cherry or reticulated cherry $\{x, y\}$ is the operation of: if $\{x, y\}$ is a cherry deleting either x or y and the incident edge, or if $\{x, y\}$ is a reticulated cherry deleting the reticulation leaf, the reticulation vertex and their incident edges, and finally suppressing any degree two vertices while keeping the ultrametric property as shown in Figure 4.8(d), (e) and (f), respectively.

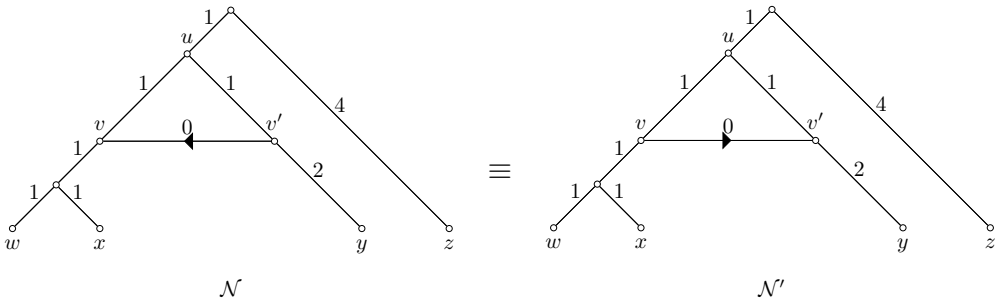


Figure 4.9 $\mathcal{N}$ and $\mathcal{N}'$ are equivalent networks up to the direction of immediate reticulation.

In Subsection 4.1.1.1 an equivalence relation $\equiv$ on the set of ultrametric tree-child phylogenetic networks is introduced. It is a consequence of the results in Section 4.1 that any two weighted ultrametric tree-child networks are equivalent if and only if they display the same set-distance matrix. In this section, since we are dealing with noisy data, we redefine equivalence to only take into account the possible topological differences, and deal with possible differences in weight function by the assumption that one of the edges coming into each reticulation has weight zero. Hence we redefine the relation $\equiv$ as follows. Two (unweighted) phylogenetic networks $\mathcal{N}, \mathcal{N}'$ are said to be equivalent, or equivalent up to the direction of immediate reticulations, denoted $\mathcal{N} \equiv \mathcal{N}'$, if in $\mathcal{N}$ there is a set of immediate reticulation arcs such that $\mathcal{N}'$ can be obtained from $\mathcal{N}$ by reversing the direction of each of these arcs. For example see Figure 4.9.

4.2.1.1 SAFETY RADIUS

The notion of safety radius, introduced by Atteson [3], relates the noise that a distance based reconstruction method for phylogenetic trees can cope with to the weight of the lightest edge in the target phylogenetic tree ($l_{\min}$). Let $\mathcal{D}^T = [d^T_{x,y}]$ be the distance matrix of a phylogenetic tree T, and let $\mathcal{D} = [d_{x,y}]$ be an observed distance matrix, which is a noisy version of $\mathcal{D}^T$. An algorithm has safety radius α if it returns the correct phylogeny whenever the maximum error in any distance estimate is at most $l_{\min}$, i.e. $\mathcal{D}$ and $\mathcal{D}^T$ satisfy

$$\left\|\mathcal{D}^T - \mathcal{D}\right\|_\infty = \max_{x,y} \left|d^T_{x,y} - d_{x,y}\right| < \alpha l_{\min}.$$

An important remark is there are distances $\mathcal{D}$ which lie at $1/2l_{\min}$ which implies that:

1. robustness must be measured relative to the length of the shortest branch in T, as no maximum value for the difference between $\mathcal{D}$ and $\mathcal{D}^T$ can guarantee a correct tree reconstruction if nothing is assumed regarding $l_{\min}$ [49];

2. no method reconstructing a unique tree can have a safety radius greater than $1/2$ [3].

Atteson [3] also proved that a number of agglomerative algorithms, including NJ, have optimal safety radius $1/2$. A related result was recently shown by Bordewich et al. [8] who proved that another heuristic aimed at minimizing BME (based on subtree pruning and regrafting) has at least radius $1/3$. Gascuel, Pardi and Truszkowski [30] review safety radius, and give the safety radius of the following well-known distance based algorithms: Neighbor Joining [3], Fast Neighbor Joining [24] and Balanced Minimum Evolution [8].

Here we extend the concept of safety radius to phylogenetic network reconstruction. For two set-distance matrices $\mathcal{D}$ and $\mathcal{D}^{\mathcal{N}}$ such that for all $x, y \in X$ $|\mathcal{D}_{x,y}| = |\mathcal{D}^{\mathcal{N}}_{x,y}|$ we define

$$\left\|\mathcal{D} - \mathcal{D}^{\mathcal{N}}\right\|_{\infty} = \max_{x,y \in X} \max_{1 \leq i \leq k} |d_i - d_i^{\mathcal{N}}|,$$

where $\mathcal{D}_{x,y} = \{d_1, d_2, \ldots, d_k\}$ and $\mathcal{D}^{\mathcal{N}}_{x,y} = \{d_1^{\mathcal{N}}, d_2^{\mathcal{N}}, \ldots, d_k^{\mathcal{N}}\}$, and each set is listed in increasing order. Also, for a weighted phylogenetic network $\mathcal{N}$ we define $l^{\mathcal{N}}_{\min}$ to be the weight of the lightest (shortest) tree-edge.

A distance based phylogenetic network reconstruction method has safety radius α if for any phylogenetic network $\mathcal{N}$ and any estimated set-distance matrix $\mathcal{D}$ such that $\left\|\mathcal{D} - \mathcal{D}^{\mathcal{N}}\right\|_{\infty} < \alpha l_{\min}$, the method correctly reconstructs a network $\mathcal{N}'$ such that the underlying unweighted networks of $\mathcal{N}'$ and $\mathcal{N}$ are equivalent up to direction of immediate reticulations.

4.2.2 FORMAL DEFINITION OF NETWORKUPGMA ALGORITHM

In this subsection, we recall the essential operation of the NETWORKUPGMA algorithm is given in Algorithm 1, but has two minor changes for convenience:

(i) We use a slightly different condition to check for a cherry or reticulated cherry that allows for noise in the data, see Lemma 4.2.4.

(ii) When reducing a reticulated cherry we remove the reticulation leaf entirely. In Section 4.1 one incoming reticulation edge was removed leaving the reticulation leaf as a cherry, which could then be removed in a subsequent step. So we have effectively concatenated two steps into one.

The main body of the NETWORKUPGMA algorithm looks for a pair $\{x, y\}$ which form a cherry or reticulated cherry. If such a pair is found, the algorithm forms a set of elements $X' = X - \{y\}$ and a set-distance matrix $\mathcal{D}'$ resulting from reducing this cherry or reticulated

cherry. It then makes a recursive call to NETWORKUPGMA$(X', \mathcal{D}')$. If this yields a suitable network $\mathcal{N}'$ displaying $\mathcal{D}'$, it then reconstructs $\mathcal{N}$ by reinstating y. Finally we need to check that the resulting network does display $\mathcal{D}$ before returning the network $\mathcal{N}$.

Algorithm 3 NETWORKUPGMA

Input: A distance matrix $\mathcal{D}$ on a finite set X

Output: An ultrametric tree-child network $\mathcal{N}$ displaying $\mathcal{D}$, or **Network not found** if no such network exists

1: **if** $|X| \leq 2$ **then**
2: **return** $\mathcal{N}$ or "Network not found" as appropriate
3: **else if** there is a pair $x, y \in X$ such that $\{x, y\}$ forms a cherry or a reticulated cherry **then**
4: form the set-distance matrix $\mathcal{D}'$ by reducing the pair $\{x, y\}$
5: **if** NETWORKUPGMA($\mathcal{D}'$)=$\mathcal{N}'$ **then**
6: form $\mathcal{N}$ from $\mathcal{N}'$ by reinstating y
7: **return** $\mathcal{N}$
8: **end if**
9: **end if**
10: **return** "Network not found"

For a set X and an estimated set distance matrix $\mathcal{D}$ of distances on X, Lemma 4.2.4 shows how we can finding a pair of elements $x, y \in X$ that must form a cherry or reticulated cherry in any ultrametric tree-child network with set-distance matrix close to $\mathcal{D}$. If such a pair is found, Lemma 4.2.5 shows how the algorithm can form a set of elements $X' = X - \{y\}$ and a set distance matrix $\mathcal{D}'$ resulting from reducing this cherry or reticulated cherry. If the recursive call to NETWORKUPGMA$(X', \mathcal{D}')$ yields a suitable network $\mathcal{N}'$ close to displaying $\mathcal{D}'$, then Lemma 4.2.5 shows how line 6 of the algorithm can reconstruct $\mathcal{N}$ by reversing the cherry or reticulated cherry reduction on $\mathcal{N}'$.

4.2.3 LEMMAS AND PROOF OF THEOREM

The following lemmas establish that the various steps in the algorithm work and can be accomplished in polynomial time. First we define the $\mathcal{S}$-score on $X \times X$, given a set-distance matrix $\mathcal{D}$ on X, and the following lemmas show we can recognise a cherry or reticulated cherry in the network using the $\mathcal{S}$-score and set-distance matrix. Then we present the lemma that demonstrates how to modify the set-distance matrix when reducing a cherry or reticulated cherry pair, and finally how to reinstate the leaf y.

Let $\mathcal{D}$ be an estimated set distance matrix. We define the $\mathcal{S}$-score as follows. For any $s, t \in X$ let $d_{s,t}$ denote the smallest element of $\mathcal{D}_{s,t}$, i.e. the lightest (estimated) up-down

path weight between s and t. Then:

$$\mathcal{S}_{\mathcal{D}}(x,y) = \min_{z \in X-\{x,y\}} (d_{x,z} - d_{x,y}).$$

Note that the $\mathcal{S}$-score is not symmetric, i.e. $\mathcal{S}_{\mathcal{D}}(x,y)$ is not necessarily equal to $\mathcal{S}_{\mathcal{D}}(y,x)$.

Lemma 4.2.2. *Let $\mathcal{N}$ be an ultrametric phylogenetic network on X, and let $\mathcal{D}^{\mathcal{N}}$ be the true set distance matrix of inter-taxa distances of $\mathcal{N}$. Let u be the peak of a shortest path from x to y. The following properties are satisfied:*

(i) If $\{x,y\}$ form a cherry or reticulated cherry then $\mathcal{S}_{\mathcal{D}^{\mathcal{N}}}(x,y) \geq 2l_{\min}$.

(ii) If there is a leaf $z \in X - \{x,y\}$ that is a descendant of u, then $\mathcal{S}_{\mathcal{D}^{\mathcal{N}}}(x,y) \leq 0$.

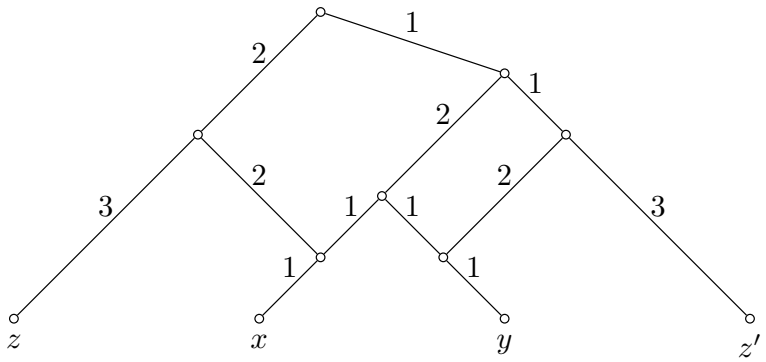
(iii) If $\mathcal{N}$ is ultrametric tree-child phylogenetic network and $\mathcal{S}_{\mathcal{D}^{\mathcal{N}}}(x,y) > 0$, then $\{x,y\}$ form a cherry or reticulated cherry.

Proof: For the proof of (i), let u' be the peak of the shortest up-down path from x to leaf $z \in X - \{x,y\}$, and let $l^{\mathcal{N}}_{min}$ be the weight of the minimum edge in the network. Observe that if $\{x,y\}$ form a cherry or reticulated cherry with y reticulation leaf, then there is no descendant of u other than x and y. Hence u' is an ancestor of u. Then the weight of the (shortest) path from u' to u is at least $l^{\mathcal{N}}_{min}$. Since the $\mathcal{N}$ is ultrametric network, $d_{x,z} - d_{x,y} \geq 2l^{\mathcal{N}}_{min}$. From the definition of $\mathcal{S}$-score, $\mathcal{S}_{\mathcal{D}^{\mathcal{N}}}(x,y) \geq 2l^{\mathcal{N}}_{min}$.

For part (ii), since leaf z is a descendant of u, then there is a tree path to z from a tree vertex v on the path either from u to x or u to y. Suppose v is on the path from u to y, then $d_{x,z} \leq d_{x,y}$. Thus, $\mathcal{S}_{\mathcal{D}^{\mathcal{N}}}(x,y) \leq d_{x,z} - d_{x,y} \leq 0$. Alternatively suppose that v is on the path from u to x. Now $d_{x,z} < d_{x,y}$ and so $\mathcal{S}_{\mathcal{D}^{\mathcal{N}}}(x,y) \leq d_{x,z} - d_{x,y} < 0$. Hence in either case, (ii) holds.

For part (iii), observe that since $\mathcal{S}_{\mathcal{D}^{\mathcal{N}}}(x,y) > 0$, then by ii, there can be no z that is a descendant of u. Thus, since the network is tree-child, there is no tree vertex on the path u to x or u to y, and moreover there can then be at most one reticulation on this pair of paths. Hence $\{x,y\}$ form a cherry or reticulated cherry. Alternatively note that by our assumption on $\mathcal{S}_{\mathcal{D}^{\mathcal{N}}}(x,y)$, for all $z \in X - \{x,y\}$ we have $d_{x,z} > d_{x,y}$. By Lemma 4.1.4, in an ultrametric tree-child network this is sufficient to show that $\{x,y\}$ form a cherry or reticulated cherry. ■

Note that, if $\{x,y\}$ form a cherry, then $\mathcal{S}_{\mathcal{D}^{\mathcal{N}}}(x,y) = \mathcal{S}_{\mathcal{D}^{\mathcal{N}}}(y,x)$, and if $\{x,y\}$ form a reticulated cherry with y the reticulation leaf, then $\mathcal{S}_{\mathcal{D}^{\mathcal{N}}}(x,y) \geq \mathcal{S}_{\mathcal{D}^{\mathcal{N}}}(y,x)$. Also note that we cannot drop the requirement of a tree-child network from part iii above, since the pair



$\mathcal{N}$

Figure 4.10 $\mathcal{N}$ is ultrametric network on X, but not a tree child network.

of leaves x, y in Figure 4.10 satisfy $\mathcal{S}_{\mathcal{D}^{\mathcal{N}}}(x, y) > 0$ but do not form a cherry or reticulated cherry.

Lemma 4.2.3. *Let $\mathcal{N}$ be an ultrametric tree-child network on $|X| > 2$ with minimum edge weight $l^{\mathcal{N}}_{min}$. Let $\mathcal{D}^{\mathcal{N}}$ be the true set distance matrix, and let $\mathcal{D}$ be an estimated distance matrix such that the maximum error is ϵ. If $\epsilon < l^{\mathcal{N}}_{min}/2$, then the pair $\{x, y\}$ that maximize $\mathcal{S}_{\mathcal{D}}\{x, y\}$ form a cherry or reticulated cherry.*

Proof: Suppose that $\epsilon < l^{\mathcal{N}}_{min}/2$, that $\{x, y\}$ maximize $\mathcal{S}_{\mathcal{D}}\{x, y\}$ over all $x, y \in X$ but assume for contradiction that $\{x, y\}$ is not a cherry or reticulated cherry. Let u be the peak of a shortest path from x to y. Since $\{x, y\}$ is not a cherry or reticulated cherry and $\mathcal{N}$ is tree-child, there must be some leaf $z \in X - \{x, y\}$ that is a descendant of u. Hence $d^{\mathcal{N}}_{x,z} \leq d^{\mathcal{N}}_{x,y}$. Hence

$$\mathcal{S}_{\mathcal{D}}\{x, y\} \leq (d_{x,z} - d_{x,y}) \leq (d^{\mathcal{N}}_{x,z} + \epsilon) - (d^{\mathcal{N}}_{x,y} - \epsilon) \leq 2\epsilon.$$

Since $\{x, y\}$ is not a cherry or a reticulated cherry and $\mathcal{N}$ is tree-child, there is some other pair of taxa $\{x', y'\}$ which do form a cherry or reticulated cherry. Hence

$$\mathcal{S}_{\mathcal{D}}\{x', y'\} = \min_z(d_{x',z} - d_{x',y'}) \geq \min_z((d^{\mathcal{N}}_{x',z} - \epsilon) - (d^{\mathcal{N}}_{x',y'} + \epsilon)) = \mathcal{S}_{\mathcal{D}^{\mathcal{N}}}\{x', y'\} - 2\epsilon.$$

And so by Lemma 4.2.2,

$$\mathcal{S}_{\mathcal{D}}\{x', y'\} \geq 2l^{\mathcal{N}}_{min} - 2\epsilon.$$

Since $\epsilon < l_{\min}/2$

$$\mathcal{S}_{\mathcal{D}}\{x', y'\} \geq 2l^{\mathcal{N}}_{min} - 2\epsilon > 4\epsilon - 2\epsilon = 2\epsilon \geq \mathcal{S}_{\mathcal{D}}\{x, y\}.$$

This contradicts the assumption about $\{x, y\}$, hence they must form a cherry or reticulated cherry. This completes the proof of the lemma. ■

The Lemma 4.1.4 establishes that we can identify a pair of leaves form a cherry or reticulated cherry (with or without immediate reticulation) from the true set-distance matrix.

The next lemma establish the effect of safety radius when identifying a cherry or a reticulated cherry on the set-distance matrices.

Lemma 4.2.4. *Let $\mathcal{N}$ be an ultrametric tree-child network on X, and let $\mathcal{D}$ be the estimated set distance matrix of inter-taxa distances of $\mathcal{N}$ such that the maximum error is $\epsilon < l^{\mathcal{N}}_{min}/2$. Let x, y be the pair of leaves that maximise $\mathcal{S}_{\mathcal{D}}(x, y)$. Then x, y:*

(i) form a cherry if and only if $|\mathcal{D}_{x,y}| = 1$.

(ii) form a reticulated cherry in which the reticulation vertex is an immediate reticulation if and only if $|\mathcal{D}_{x,y}| = 2$ and $|D_{x,z}| = |D_{y,z}|$ for all $z \in X - \{x, y\}$.

(iii) form a reticulated cherry of $\mathcal{N}$ without immediate reticulation, with y the reticulation leaf, if and only if there exists a leaf z such that $|D_{x,z}| < |D_{y,z}|$.

Proof: Since $\{x, y\}$ is the pair of leaves that maximise $\mathcal{S}_{\mathcal{D}}(x, y)$, by Lemma 4.2.3 $\{x, y\}$ form a cherry or reticulated cherry. Let $\mathcal{D}^{\mathcal{N}}$ be the true set matrix of inter-taxa distances.

For i, by Lemma 4.1.4 (i), the pair of leaves $\{x, y\}$ form a cherry if and only if $|\mathcal{D}^{\mathcal{N}}_{x,y}| = 1$. By the definition of estimated set distance matrix $|\mathcal{D}^{\mathcal{N}}_{x,y}| = |\mathcal{D}_{x,y}|$, so $\{x, y\}$ form a cherry if and only if $|\mathcal{D}_{x,y}| = 1$.

By Lemma 4.1.4 (ii), the pair of leaves $\{x, y\}$ form an immediate reticulation if and only if $|\mathcal{D}^{\mathcal{N}}_{x,y}| = 2$ and

$$\mathcal{D}^{\mathcal{N}}_{x,z} = \mathcal{D}^{\mathcal{N}}_{y,z} : \forall z \in X - \{x, y\}.$$

Thus if $\{x, y\}$ form an immediate reticulation then $|\mathcal{D}_{x,y}| = 2$ and $|\mathcal{D}_{x,z}| = |\mathcal{D}_{y,z}|$.

Again by Lemma 4.1.4 (iii), the pair of leaves $\{x, y\}$ form a reticulated cherry without immediate reticulation, with y the reticulation leaf, if and only if there exist a leaf $z \in X - \{x, y\}$ such that $|\mathcal{D}^{\mathcal{N}}_{x,z}| = |\mathcal{D}^{\mathcal{N}}_{y,z}| - 1$. Thus if $\{x, y\}$ form a reticulated cherry without immediate reticulation, then $|\mathcal{D}_{x,z}| < |\mathcal{D}_{y,z}|$. For the reverse directions, observe that $\{x, y\}$ form a cherry or reticulated cherry and, by (i), if $|\mathcal{D}_{x,y}| \geq 2$ then they must form a reticulated

cherry. Hence it is either with an immediate reticulation (and so $|\mathcal{D}_{x,z}| = |\mathcal{D}_{y,z}|$) or without (and so $|\mathcal{D}_{x,z}| < |\mathcal{D}_{y,z}|$). ■

Lemma 4.2.5. *Let $\mathcal{N}$ be an ultrametric tree-child network on $|X| > 2$. Let $\mathcal{D}$ be an estimated set distance matrix of inter-taxa distances within $\epsilon < l^{\mathcal{N}}_{min}/2$ of $\mathcal{D}_{\mathcal{N}}$. Let $\{x, y\}$ be a cherry or reticulated cherry (with y the reticulation leaf). Let $X' = X - \{y\}$ and $\mathcal{D}'$ be the set-matrix of inter-taxa distances on X' given by $\mathcal{D}'_{z,z'} = \mathcal{D}_{z,z'}$ for all $z, z' \in X'$. Let $\mathcal{N}'$ be the ultrametric tree-child network on X' is obtained from $\mathcal{N}$ by reducing the cherry or reticulated cherry $\{x, y\}$. Then:*

(i) The set-distance matrix $\mathcal{D}'$ is within ϵ of $\mathcal{D}^{\mathcal{N}'}$ and $\epsilon < l^{\mathcal{N}'}_{min}/2$.

(ii) Given $\hat{\mathcal{N}}'$ on X' such that as unweighted networks $\hat{\mathcal{N}}' \equiv \mathcal{N}'$ and also $|\mathcal{D}^{\hat{\mathcal{N}}'} - \mathcal{D}^{\mathcal{N}'}| < \epsilon$, then we can determine a network $\hat{\mathcal{N}}$ such that as unweighted networks $\hat{\mathcal{N}} \equiv \mathcal{N}$ and also $|\mathcal{D}^{\hat{\mathcal{N}}} - \mathcal{D}^{\mathcal{N}}| < \epsilon$ and moreover this can be computed in time $O(|X|^3)$.

(iii) If for all networks $\hat{\mathcal{N}}'$ on X' such that $|\mathcal{D}^{\hat{\mathcal{N}}'} - \mathcal{D}^{\mathcal{N}'}| < \epsilon$, we have that as unweighted networks $\hat{\mathcal{N}}' \equiv \mathcal{N}'$, then for any network $\hat{\mathcal{N}}$ on X such that $|\mathcal{D}^{\hat{\mathcal{N}}} - \mathcal{D}^{\mathcal{N}}| < \epsilon$ it also holds that as unweighted networks $\hat{\mathcal{N}} \equiv \mathcal{N}$.

Proof:

For a cherry: let u be the peak of the shortest path from x to y, and let w be the parent of u. Reducing a cherry means deleting the leaf y and its incident edge, and suppressing degree two vertex u.

For a reticulated cherry: let v be the reticulation vertex on the edge from u to y. Since v is a reticulation vertex, let u' be the parent of v additional to u. Reducing the reticulated cherry means, removing the arcs $(u, v), (u', v)$ and deleting leaf y and its incident edge, and suppressing degree two vertices u and u'.

The set of path distances between z and z' are unchanged by the cherry reduction for all $z, z' \in X'$, thus $\mathcal{D}'$ is within ϵ of $\mathcal{D}^{\mathcal{N}'}$. The act of reducing the cherry or reticulated cherry only removes edges and merges edges (when a degree 2 vertex is supressed). Thus $l^{\mathcal{N}'}_{min}$, the minimum edge in $\mathcal{N}'$, is at least $l^{\mathcal{N}}_{min}$ in $\mathcal{N}$, and so $\epsilon < l^{\mathcal{N}'}_{min}/2$. This completes the proof of part i.

For part ii, we first consider the case $\{x, y\}$ form a cherry in $\mathcal{N}$ (which we can observe from $\mathcal{D}$ by Lemma 4.2.4). Let w be the parent of leaf x in $\hat{\mathcal{N}}'$. Then we form $\hat{\mathcal{N}}$ from $\hat{\mathcal{N}}'$ by subdividing the arc (w, x) with a new vertex u, and appending a leaf y as a new child of u. All that remains is to set the weight of arcs (u, x) and (u, y) to $d_{x,y}/2$, and set the weight of the arc (w, u) to the weight of the original arc (w, x) less $d_{x,y}/2$. This can be done as long

as the arc (w, x) had weight at least $d_{x,y}/2$ initially, which we now argue must be the case. Note that if w is a tree vertex, then there is a path from w to some leaf $z \in X - \{x\}$ in $\hat{\mathcal{N}}'$; if w is a reticulation vertex, then let a and b parents of w, and there is a tree path from a to some leaf and from b to some leaf, then pick z to be the leaf of these two which minimizes the distance $d_{y,z}$. By construction of $\mathcal{N}'$, the distance from x to its parent in $\mathcal{N}'$ is at least $d^{\mathcal{N}}_{x,y}/2 + l^{\mathcal{N}}_{min} > (d_{x,y} - \epsilon)/2 + 2\epsilon = d_{x,y}/2 + 3\epsilon/2$, since by assumption $|\mathcal{D} - \mathcal{D}^{\mathcal{N}}| < \epsilon$. Now since $|\mathcal{D}^{\hat{\mathcal{N}}'} - \mathcal{D}^{\mathcal{N}'}| < \epsilon$, the weight of (w, x) (in $\hat{\mathcal{N}}'$) is at least the corresponding weight in $\mathcal{N}'$ less $\epsilon/2$, since it corresponds to half the inter-taxa distance between x and z, thus is at least $d_{x,y}/2 + \epsilon$. So we can construct $\hat{\mathcal{N}}$ as claimed. Since $\hat{\mathcal{N}}' \equiv \mathcal{N}'$, and we have reinstated y as a cherry with x, as it is in $\mathcal{N}$ itself, which does not impact on any immediate reticulations, $\hat{\mathcal{N}} \equiv \mathcal{N}$. For all $z \in X - \{x, y\}$ we have $\mathcal{D}^{\hat{\mathcal{N}}}_{y,z} = \mathcal{D}^{\hat{\mathcal{N}}}_{x,z} = \mathcal{D}^{\hat{\mathcal{N}}'}_{x,z}$ and also $\mathcal{D}^{\mathcal{N}'}_{x,z} = \mathcal{D}^{\mathcal{N}}_{x,z} = \mathcal{D}^{\mathcal{N}}_{y,z}$. Hence $|\mathcal{D}^{\hat{\mathcal{N}}} - \mathcal{D}^{\mathcal{N}}| = |\mathcal{D}^{\hat{\mathcal{N}}'} - \mathcal{D}^{\mathcal{N}'}| < \epsilon$, by assumption.

Secondly suppose that $\{x, y\}$ form a reticulated cherry with immediate reticulation in $\mathcal{N}$ where y is a reticulation leaf (which we can observe from $\mathcal{D}$ by Lemma 4.2.4). Let w be the parent of x in $\hat{\mathcal{N}}'$. To form $\hat{\mathcal{N}}$ from $\hat{\mathcal{N}}'$ first construct the cherry $\{x, y\}$ as above. Let u be the parent of x and y in $\hat{\mathcal{N}}$. Then to form $\{x, y\}$ as an immediate reticulation, first subdivide the arc (w, u) with a new vertex v and subdivide the arc (u, y) with a new vertex v', then add an arc (v, v'). By Lemma 4.2.4, if $\{x, y\}$ form a reticulated cherry with immediate reticulation, then $|\mathcal{D}_{x,y}| = 2$. Let $d_{x,y}$ be the shortest distance in $\mathcal{D}_{x,y}$, and $\hat{d}_{x,y}$ be the other distance in $\mathcal{D}_{x,y}$. Since u is descendant of v, the peak of the shortest path between x and y is u. Thus set the weight of arcs (u, x) and (u, y) to $d_{x,y}/2$ and the weight of arcs (v, x) and (v, y) to $\hat{d}_{x,y}/2$, so the weight of the arc (v, u) is $(\hat{d}_{x,y} - d_{x,y})/2$. By the equivalence relation, assume that the weight of the arc (u, v') is 0, then the weight of the (v, u) equal to the weight of the arc (v, v'). As for the cherry case, it works if weight of the arc (w, x) is less than $\hat{d}_{x,y}/2$, and similar arguments show that it is.

Thirdly suppose that $\{x, y\}$ form a reticulated cherry without immediate reticulation in $\mathcal{N}$ where y is a reticulation leaf (which we can observe from $\mathcal{D}$ by Lemma 4.2.4). Let w be the parent of x in $\hat{\mathcal{N}}'$. To form $\hat{\mathcal{N}}$ from $\hat{\mathcal{N}}'$ first construct the cherry $\{x, y\}$ by subdividing the arc (w, x), as above, and let u be the new vertex with height set to $d_{x,y}/2$, then appending y as a child of u. Next subdivide the arc (u, y) with a new vertex v, and subdivide an arc (a, b) (to be specified shortly) with a new vertex u', finally inserting the arc (u', v) to obtain $\hat{\mathcal{N}}$. As for the cases done the weight of the edge (w, x) is large enough. We now describe how to determine the arc (a, b).

Recall that $\{x, y\}$ form a reticulated cherry with y reticulation leaf in $\mathcal{N}$, and let v be the reticulation vertex, u the parent of v and x, and u' the other parent u of v. Then there is a tree

path from u' to a leaf $z \in X - \{x, y\}$ in $\mathcal{N}$. Every up-down path from y to z either starts with arcs $(y, v), (v, u)$, and therefore is identical in weight to an up-down path x to z starting $(x, v), (v, u)$, or is the unique up-down path from y to z starting with arcs $(y, v), (v, u')$, and then completing with the tree-path from u' to z; let this up-down path have weight $d'_{y,z}$. Thus $|\mathcal{D}_{y,z}| = |\mathcal{D}^{\mathcal{N}}_{y,z}| = |\mathcal{D}^{\mathcal{N}}_{x,z}| + 1 = |\mathcal{D}_{x,z}| + 1$. Also, every up-down path from x to z has peak that is either an ancestor or descendant of u' since the path from u' to z is a tree path, and hence gives rise to a distance in $\mathcal{D}^{\mathcal{N}}_{x,z}$ that is different from $d'_{y,z}$ by at least $2l^{\mathcal{N}}_{min}$. So for each true distance d in $\mathcal{D}^{\mathcal{N}}_{y,z} \setminus \{d'_{y,z}\}$, there are corresponding observed distances $s \in \mathcal{D}_{y,z}, t \in \mathcal{D}_{x,z}$ such that $|s - t| < 2\epsilon$, whereas for the observed distance $s' \in \mathcal{D}_{y,z}$ corresponding to $d'_{y,z}$ we have $\min\{|s' - t| : t \in \mathcal{D}_{x,z}\} > 2l^{\mathcal{N}}_{min} - 2\epsilon > 2\epsilon$.

Now let a be the parent of u', and b the child of u' that is not v. Then the weight of the path from b to z in $\mathcal{N}'$ is less than $d_{y,z}/2 - l^{\mathcal{N}}_{min}$ and so the weight of the path in $\hat{\mathcal{N}}'$ is at most $d_{y,z}/2 - l^{\mathcal{N}}_{min} + \epsilon < d_{y,z}/2$. Similarly, the weight of the path from a to z is at least $d_{y,z}/2 + l^{\mathcal{N}}_{min} - \epsilon > d_{y,z}/2$. Thus, in summary, there is a leaf z in $\hat{\mathcal{N}}'$ with a unique extra distance in $\mathcal{D}_{y,z}$ compared to $\mathcal{D}_{x,z}$, such that there is a unique arc (a, b) spanning the height $d_{y,z}/2$ on a tree path above z. Thus if we know z, finding the correct arc (a, b) is trivial. Moreover we can simply try each possible leaf (as z), since if we pick the wrong leaf and attempt to insert u' on the incorrect arc, then the distances from x and y to the correct z will be out by at least $2l^{\mathcal{N}}_{min} - \epsilon > 0$.

In terms of running time, reinstating y adjacent to x is a trivial operation in the case that $\{x, y\}$ is a cherry, and in the case that $\{x, y\}$ is a reticulated cherry, all the work is in determining the arc (a, b). However since we need only consider $|X|$ possible leaves as z, each one requiring a simple walk up tree-edges of the network to determine the corresponding candidate arc (a, b), and a check that $O(|X|^2)$ distances (from x to all other leaves) match, we are accomplished in $O(|X|^3)$ time overall.

For part (iii), if there is a unique (up to equivalence under $\equiv$) network that is close to displaying the reduced distance matrix, then there is also a unique (up to equivalence under $\equiv$) network that is close to displaying the original distance matrix. Note that, since a pair of leaves $\{x, y\}$ must form correct cherry by Lemma 4.2.4, the location of the arc (a, b) is unique, $\hat{\mathcal{N}}'$ is also unique (up to $\equiv$). ■

4.2.3.1 PROOF OF THEOREM 4.2.1

The proof is by induction on $|X|$. It is straightforward to verify that if $|X| \leq 2$ then NETWORKUPGMA takes the correct action. Assume now that $|X| > 2$ and the algorithm is correct on inputs with fewer than $|X|$ leaves. By Lemma 4.2.4, we can determine if there

is a cherry of reticulated cherry in line 3 applies, and by Lemma 4.2.5 assuming there is a network that displays $\mathcal{D}$, then the correct $\mathcal{D}', X'$ are created corresponding to a network after the appropriate reduction.

Again by Lemma 4.8, assuming there is an ultrametric tree-child network that displays $\mathcal{D}$, then the recursive call in line 5 would return a valid network, so we are correct to return "Network not found" if the recursive call does not return a network. Finally, in the case that a network $\mathcal{N}'$ is returned, if there is a network that displays $\mathcal{D}$, then by Lemma 4.2.5, we reconstruct a valid network displaying $\mathcal{D}$ from $\mathcal{N}'$, and hence return a correct answer. If there is not an ultrametric tree-child network that displays $\mathcal{D}$, then the check in line 10 fails and we correctly return "Network not found". Hence in all cases NETWORKUPGMA is correct.

Finally observe that when NETWORKUPGMA returns a network, it is built up from a network on one or two leaves by successively reversing reductions. Since there is a unique possible network for each case when $|X| = 1$ or $|X| = 2$, and by Lemma 4.2.5, each reduction reversal results in a unique network (up to $\equiv$), it must be that $\mathcal{N}$ is also unique up to $\equiv$.

Now we consider the running time of NETWORKUPGMA algorithm. Recall that the estimates set-distance matrix $\mathcal{D}$ has $|X|^2$ entries, each of which is a set of size at most $|X|$ 4.1. Lines 1-2 deal with constant sized $|X|$ and can be accomplished in constant time. By Lemma 4.2.4, line 3, which determines if there is a cherry or a reticulated cherry, can be accomplished in $O(|X|^3)$ time, since each of the $|X|^2$ and $\mathcal{S}$ values is a minimum over $|X|$ elements. Once x, y have been identified, line 4 to form the set-distance matrix $\mathcal{D}'$ is a simple manipulation of $\mathcal{D}$. Line 5-7 deal with reinstating the cherry or reticulated cherry, which can be done in $O(|X|^3)$ by applying Lemma 4.2.5. Since we reduce the number of leaves being considered in each recursive call, there are at most $|X|$ recursions and the whole algorithm runs in time $O(|X|^4)$. ■

4.3 CONCLUSION

We presented NETWORKUPGMA algorithm and proved its safety radius is $1/2$. Thus, NETWORKUPGMA reconstructs the ultrametric tree child network from not only true set of distances, also the estimated set of distances.

5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

Most research in phylogenetics is related to trees. In this thesis, we consider networks, which which have been gaining wide interest in the past few years, as evidence by the recent publication of the books dedications to the topic [33, 38]

In Chapter 3, we have shown better bounds for the running time of algorithms computing the hybridization number and the rSPR/TBR distance between two phylogenetic trees using cluster reductions. We have thus given an explanation for the curious divergence between theoretical results, which relied only on chain reductions, and observed running time.

In Chapter 4, first, we have presented a generalisation NETWORKUPGMA of the widely used ultrametric tree reconstruction algorithm UPGMA to ultrametric tree-child networks. This expands the class of weighted networks that can be directly reconstructed from inter-taxa distance information to include much more complex networks than galled trees or single reticulation networks.

In the second part of Chapter 4, we have shown that the safety radius of NETWORKUPGMA algorithm is $1/2$. The importance of proving safety radius is that we can guarantee that the whole network is correctly reconstructed if the estimated set-distance matrix is sufficiently accurate when compared to the minimum edge length in the network.

5.2 FUTURE WORK

Here, we discuss some possible directions that research might follow in the future.

5.2.1 CHAPTER 3

A deeper biological question that warrants further research is:

Question 5.2.1. *Why does real biological data partition so effectively under the cluster reduction? In other words, why are observed networks of low hybridization level, i.e. only closely related species can hybridise?*

Since many fixed parameter tractable algorithms work poorly in practice in spite of pleasing theoretical bounds, the question is:

Question 5.2.2. *What is the effect of these new results on real or simulated date?*

5.2.2 CHAPTER 4

The algorithm NETWORKUPGMA reconstructs a network from a set distance matrix $\mathcal{D}$, if there is an ultrametric tree-child network that displays the set-distance matrix $\mathcal{D}$. The question is:

Question 5.2.3. *Suppose that a set-distance matrix D on X is not displayed by any ultrametric tree-child network, then is it possible to determine the largest subset $Y \subset X$ such that there is an ultrametric tree-child network on Y where the set of distances between any $y, z \in Y$ is given by $D_{y,z}$?*

This maximisation problem is clearly harder than the simple decision problem of determining whether a given subset of X may be displayed on an ultrametric tree-child network, which could be answered by the algorithm in Chapter 4.

Another question about the accuracy guarantees of NETWORKUPGMA algorithm is determining a tree child network from the set matrix of shortest inter-taxa distance which is given with the theorem is below:

Question 5.2.4. *What is the class of binary phylogenetic networks that, up to equivalence, are correctly reconstructed when* NETWORKUPGMA *applied to a set X and their true set matrix of shortest inter-taxa distances?*

In Section 4.2, we determined the accuracy guarantees of NETWORKUPGMA algorithm. Further work could consider firstly a stochastic form of safety radius:

Question 5.2.5. *How robust is the algorithm to random noise, rather than absolute deviations from the true set-distance matrix, and secondly robustness to not just changes in the values in the set-distance matrix, but also different size of sets of distances between pairs of taxa.*

Another open problem that arises from our work is the following. It is a consequence of our results that any two ultrametric tree-child networks display the same set distance matrix if and only if they are equivalent up to the direction of immediate reticulations (under our assumption that each reticulation has an incoming arc of weight zero). Further, it follows from our work and [13] that any two ultrametric tree-child networks display the same multiset-distance matrix if and only if they are isomorphic except for the reversal of an immediate reticulation arc between the children of the root. The question is:

Question 5.2.6. *We have effectively categorised when two ultrametric tree child networks display the same set distance matrix. Is there a simple categorisation for when two tree-child networks display the same set-distance matrix? Is there a simple categorisation for when two rooted binary phylogenetic networks display the same set-distance matrix?*

Our algorithm suffers the same drawbacks as the original UPGMA algorithm does for trees: it relies on the assumption that the target network is ultrametric. It is clear that this assumption is not always valid, and this is part of the reason that Neighbor Joining (NJ) has proved to be an even more popular and robust method for reconstructing phylogenetic trees than UPGMA.

Another question is reconstructing a hybridization network with Neighbor joining algorithm from the multiset matrix of inter-taxa distances. The Neighbor joining method is widely used to construct phylogenetic trees because of its speed and consistency. The following question extends the results of the paper [13] and Chapter 4 to the class of binary tree-child phylogenetic network on X.

Question 5.2.7. *What is the class of binary phylogenetic networks that, up to equivalence, are correctly reconstructed when Neighbor Joining applied to their multiset matrix (or set-matrix) of inter-taxa distances?*

BIBLIOGRAPHY

[1] Allen, B. L. and Steel, M. (2001). Subtree transfer operations and their induced metrics on evolutionary trees. *Annals of combinatorics*, 5(1):1–15.

[2] Apostolico, A., Comin, M., Dress, A., and Parida, L. (2013). Ultrametric networks: a new tool for phylogenetic analysis. *Algorithms for molecular biology*, 8(1):1.

[3] Atteson, K. (1999). The performance of neighbor-joining methods of phylogenetic reconstruction. *Algorithmica*, 25(2-3):251–278.

[4] Baroni, M., Grünewald, S., Moulton, V., and Semple, C. (2005). Bounding the number of hybridisation events for a consistent evolutionary history. *Journal of mathematical biology*, 51(2):171–182.

[5] Baroni, M., Semple, C., and Steel, M. (2006). Hybrids in real time. *Systematic biology*, 55(1):46–56.

[6] Bordewich, M., Linz, S., St John, K., and Semple, C. (2007). A reduction algorithm for computing the hybridization number of two trees. *Evolutionary bioinformatics online*, 3:86–98.

[7] Bordewich, M., McCartin, C., and Semple, C. (2008). A 3-approximation algorithm for the subtree distance between phylogenies. *Journal of discrete algorithms*, 6(3):458–471.

[8] Bordewich, M. and Mihaescu, R. (2010). Accuracy guarantees for phylogeny reconstruction algorithms based on balanced minimum evolution. *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM*, 10(3):576–83.

[9] Bordewich, M., Scornavacca, C., Tokac, N., and Weller, M. (2016). On the fixed parameter tractability of agreement-based phylogenetic distances. *Journal of mathematical biology*, pages 1–19.

[10] Bordewich, M. and Semple, C. (2005). On the computational complexity of the rooted subtree prune and regraft distance. *Annals of combinatorics*, 8(4):409–423.

[11] Bordewich, M. and Semple, C. (2007a). Computing the hybridization number of two phylogenetic trees is fixed-parameter tractable. *IEEE/ACM transactions on computational biology and bioinformatics (TCBB)*, 4(3):458–466.

[12] Bordewich, M. and Semple, C. (2007b). Computing the minimum number of hybridization events for a consistent evolutionary history. *Discrete applied mathematics*, 155(8):914–928.

[13] Bordewich, M. and Semple, C. (2015). Determining phylogenetic networks from inter-taxa distances. *Journal of mathematical biology*, pages 1–21.

[14] Bordewich, M. and Tokac, N. (2014). On the fixed parameter tractability of hybridization number and rooted subtree prune and regraft distance. pages 1–6.

[15] Bordewich, M. and Tokac, N. (2016). An algorithm for reconstructing ultrametric tree-child networks from inter-taxa distances. *Discrete applied mathematics*.

[16] Buneman, O. P. (1971). The recovery of trees from measures of dissimilarity. *Mathematics in the archaeological and historical sciences*.

[17] Cardona, G., Rossello, F., and Valiente, G. (2009). Comparison of tree-child phylogenetic networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 6(4):552–569.

[18] Chan, H.-L., Jansson, J., Lam, T.-W., and Yiu, S.-M. (2006). Reconstructing an ultrametric galled phylogenetic network from a distance matrix. *Journal of bioinformatics and computational biology*, 4(04):807–832.

[19] Chen, J., Fan, J.-H., and Sze, S.-H. (2013). Parameterized and approximation algorithms for the maf problem in multifurcating trees. In *Graph-Theoretic Concepts in Computer Science*, pages 152–164. Springer.

[20] Chen, Z.-Z., Fan, Y., and Wang, L. (2015). Faster exact computation of rspr distance. *Journal of combinatorial optimization*, 29(3):605–635.

[21] Choy, C., Jansson, J., Sadakane, K., and Sung, W.-K. (2005). Computing the maximum agreement of phylogenetic networks. *Theoretical computer science*, 335(1):93–107.

[22] DasGupta, B., He, X., Jiang, T., Li, M., Tromp, J., Wang, L., and Zhang, L. (2013). Computing distances between evolutionary trees. In *Handbook of Combinatorial Optimization*, pages 747–781. Springer.

[23] Eickmeyer, K., Huggins, P., Pachter, L., and Yoshida, R. (2008). On the optimality of the neighbor-joining algorithm. *Algorithms for molecular biology*, 3(1):1.

[24] Elias, I. and Lagergren, J. (2005). Fast neighbor joining. In *Automata, Languages and Programming*, pages 1263–1274. Springer.

[25] Felsenstein, J. (1997). An alternating least squares approach to inferring phylogenies from pairwise distances. *Systematic biology*, 46(1):101–111.

[26] Fitch, W. M. and Margoliash, E. (1967). Construction of phylogenetic trees. *Science*, 155(3760):279–284.

[27] Gambette,P. (2016). Who is Who in Phylogenetic Networks. http://phylnet.univ-mlv.fr/. Online; accessed 22 September 2016.

[28] Gascuel, O. (1997a). Bionj: an improved version of the nj algorithm based on a simple model of sequence data. *Molecular biology and evolution*, 14(7):685–695.

[29] Gascuel, O. (1997b). Concerning the nj algorithm and its unweighted version, unj. *Mathematical hierarchies and biology*, 37:149–171.

[30] Gascuel, O., Pardi, F., and Truszkowski, J. (2015). Distance-based phylogeny reconstruction: safety and edge radius. *Encyclopedia of algorithms*, pages 1–6.

[31] Gascuel, O. and Steel, M. (2006). Neighbor-joining revealed. *Molecular biology and evolution*, 23(11):1997–2000.

[32] Gascuel, O. and Steel, M. (2016). A stochastic safety radius for distance-based tree reconstruction. *Algorithmica*, 74(4):1386–1403.

[33] Gusfield, D. (2014). *ReCombinatorics: the algorithmics of ancestral recombination graphs and explicit phylogenetic networks*. The MIT Press.

[34] Hallett, M. and McCartin, C. (2007). A faster fpt algorithm for the maximum agreement forest problem. *Theory of computing systems*, 41(3):539–550.

[35] Hallett, M. T. and Lagergren, J. (2001). Efficient algorithms for lateral gene transfer problems. In *Proceedings of the fifth annual international conference on Computational biology, ACM*, pages 149–156.

[36] Harel, D. and Tarjan, R. E. (1984). Fast algorithms for finding nearest common ancestors. *SIAM journal on computing*, 13(2):338–355.

[37] Hein, J., Jiang, T., Wang, L., and Zhang, K. (1996). On the complexity of comparing evolutionary trees. *Discrete applied mathematics*, 71:153–169.

[38] Huson, D. H., Rupp, R., and Scornavacca, C. (2010). *Phylogenetic networks: concepts, algorithms and applications*. Cambridge University Press.

[39] Jansson, J. and Sung, W. K. (2006). Inferring a level-1 phylogenetic network from a dense set of rooted triplets. *Theoretical computer science*, 363:60–68.

[40] Kelk, S., Scornavacca, C., and Van Iersel, L. (2012a). On the elusiveness of clusters. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 9(2):517–534.

[41] Kelk, S., Van Iersel, L., Lekic, N., Linz, S., Scornavacca, C., and Stougie, L. (2012b). Cycle killer... qu'est-ce que c'est? on the comparative approximability of hybridization number and directed feedback vertex set. *SIAM journal on discrete mathematics*, 26(4):1635–1656.

[42] Linz, S. and Semple, C. (2009). A cluster reduction for computing the subtree distance between phylogenies. *Annals of combinatorics*, 15(3):465–484.

[43] Maddison, W. P. (1997). Gene trees in species trees. *Systematic biology*, 46(3):523–536.

[44] Makarenkov, V. and Leclerc, B. (1999). An algorithm for the fitting of a tree metric according to a weighted least-squares criterion. *Journal of classification*, 16(1):3–26.

[45] Moret, B. M. and Warnow, T. (2002). Reconstructing optimal phylogenetic trees: A challenge in experimental algorithmics. In *Experimental Algorithmics*, pages 163–180. Springer.

[46] Morrison, D., Van Iersel, L., Kelk,S. and List, M. (2016). The Genealogical World of Phylogenetic Networks. http://phylonetworks.blogspot.co.uk/. Online; accessed 22 September 2016.

[47] Nakhleh, L., Warnow, T., and Linder, C. R. (2004). Reconstructing reticulate evolution in species: theory and practice. In *Proceedings of the eighth annual international conference on Resaerch in computational molecular biology*, pages 337–346. ACM.

[48] Niedermeier, R. (2002). Invitation to fixed-parameter algorithms. *Habilitationschrift, University of Tübingen*.

[49] Pardi, F., Guillemot, S., and Gascuel, O. (2010). Robustness of phylogenetic inference based on minimum evolution. *Bulletin of mathematical biology*, 72(7):1820–1839.

[50] Rzhetsky, A. and Nei, M. (1993). Theoretical foundation of the minimum-evolution method of phylogenetic inference. *Molecular biology and evolution*, 10(5):1073–1095.

[51] Saitou, N. and Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular biology and evolution*, 4(4):406–425.

[52] Sattath, S. and Tversky, A. (1977). Additive similarity trees. *Psychometrika*, 42(3):319–345.

[53] Semple, C. and Steel, M. (2003). *Phylogenetics*. Oxford University Press.

[54] Sokal, R. R. (1958). A statistical method for evaluating systematic relationships. *Univ Kans Sci Bull*, 38:1409–1438.

[55] Van Iersel, L., Keijsper, J., Kelk, S., Stougie, L., Hagen, F., and Boekhout, T. (2009a). Constructing level-2 phylogenetic networks from triplets. *IEEE/ACM transactions on computational biology and bioinformatics (TCBB)*, 6(4):667–681.

[56] Van Iersel, L. and Kelk, S. (2011). When two trees go to war. *Journal of Theoretical Biology*, 269(1):245–255.

[57] Van Iersel, L., Kelk, S., Lekic, N., and Stougie, L. (2014). Approximation algorithms for nonbinary agreement forests. *SIAM Journal on Discrete Mathematics*, 28(1):49–66.

[58] Van Iersel, L., Kelk, S., and Mnich, M. (2009b). Uniqueness, intractability and exact algorithms: reflections on level-k phylogenetic networks. *Journal of bioinformatics and computational biology*, 7(4):597–623.

[59] Van Iersel, L., Kelk, S., Stamoulis, G., Stougie, L., and Boes, O. (2016). On unrooted and root-uncertain variants of several well-known phylogenetic network problems. *arXiv preprint arXiv:1609.00544*.

[60] Whidden, C., Beiko, R., and Zeh, N. (In preparation). Computing the SPR distance of binary rooted trees in $O(2^k n)$ time.

[61] Whidden, C., Beiko, R. G., and Zeh, N. (2010). Fast fpt algorithms for computing rooted agreement forests: Theory and experiments. In *International Symposium on Experimental Algorithms*, pages 141–153. Springer.

[62] Whidden, C., Beiko, R. G., and Zeh, N. (2013). Fixed-parameter algorithms for maximum agreement forests. *SIAM journal on computing*, 42(4):1431–1466.

[63] Whidden, C. and Zeh, N. (2009). A unifying view on approximation and fpt of agreement forests. In *International Workshop on Algorithms in Bioinformatics*, pages 390–402. Springer.

[64] Willson, S. J. (2012). Tree-average distances on certain phylogenetic networks have their weights uniquely determined. *Algorithms for molecular biology : AMB*, 7(1):13.

[65] Willson, S. J. (2013). Reconstruction of certain phylogenetic networks from their tree-average distances. *Bulletin of mathematical biology*, 75(10):1840–78.