Barking up the wrong tree: Some obstacles to phylogenetic reconstruction

Michael Hendriksen

A thesis presented to Western Sydney University
in fulfillment of the requirements
for the degree of
Doctor of Philosophy

May 26, 2020

Abstract

Phylogenetics is the study of evolutionary relationships between entities, usually biological in nature. The primary aim of such study is to elucidate the structure of these evolutionary histories. Unfortunately, such study can run into a variety of obstacles, both practical and theoretical. In this thesis we explore theoretical obstacles to phylogenetic reconstruction, by examining several scenarios in which distinguishing between similar structures can become quite difficult.

In Chapter 2, we consider when metrics on trees and metrics on networks can become indistinguishable, and present several novel results in this area, showing that it is possible for any tree metric to be represented on a non-trivial network, and provide early results on the possible structures of these networks.

In Chapter 3, we consider tree-based networks — a phenomenon in which networks have a strong tree-like signal. We present the first findings on these networks in the context of unrooted non-binary networks. We characterise the circumstances under which such networks can become 'saturated' by these signals, and provide some graph theoretical results in this area as well.

In Chapter 4 we consider the scenario in which two trees can appear similar due to their hierarchical structure. We present a new metric to quantify this similarity, and use simulations to show several promising properties of the metric and the relative accuracy of a function that gives an upper bound to the metric.

Chapter 3 is largely based on the article "Tree-based unrooted nonbinary phylogenetic networks", written by myself and published in *Mathematical Biosciences*, Volume 302, August 2018, Pages 131-138. Chapters 2 and 4 are based on two papers currently submitted to journals, authored by myself and my supervisor, Professor Andrew Francis. The remainder of this thesis is, unless otherwise indicated, my own work.

Acknowledgements

I must first thank my supervisor Professor Andrew Francis, who has consistently gone above and beyond the duties of a supervisor. For example, I vividly recall a meeting in which he was bottle-feeding a kitten I had found that morning while I talked him through my latest results. He has also been extremely supportive and understanding of the various challenges I have faced throughout my candidature, and I can confidently say this thesis would not be half as good without him.

I would secondly like to thank my partner Rosie Flory, for her consistent interest in my work and her willingness to put up with my endless ramblings about trees, her sympathy when a particular research path was found to be a dead end, and her celebration of the successes. The fact that my sanity remains somewhat intact now is almost entirely her doing.

I would also like to thank my family for nurturing my love of mathematics and for believing in me. Without them this thesis wouldn't even exist.

Many thanks to Chad Clark, Tanzila Chowdhury, Stuart Serdoz, Shona Yu and the team at CRM (now CRMDS) for many interesting conversations and insights. Thanks to Mike Steel, Alexei Drummond, Michael Charleston, Simone Linz, Mareike Fischer and her team and countless others for their help, support and worthwhile chats at various conferences.

Thank you to my cats, Zeta and Momo, and my bunnies, Milo, Macca and the late Hermes, for performing many important and adorable functions, like laying directly on top of my keyboard when I've been doing too much work.

Finally, thank you for reading this!

Statement of Authentication

The work presented in this thesis is, to the best of my knowledge and belief, original except as acknowledged in the text. I hereby declare that I have not submitted this material, either in full or in part, for a degree at this or any other institution.

. . . . . .

Michael Arent Hendriksen

Notation

X: A set of taxa
T: A phylogenetic tree
N: A phylogenetic network
$RP(X)$: The set of rooted phylogenetic trees on X
$BRP(X)$: The set of binary rooted phylogenetic trees on X
$MRP(X)$: The set of multi-hierarchies on X
$\mathcal{M}$: A multihierarchy
w: A weight function on a rooted binary phylogenetic network
d: The distance on a network induced by a weight function
T^w: A weighted tree
N^w A weighted network
$e = \{v_1, v_2\}$: an undirected edge of a network between vertices v_1 and v_2
$a = (v_1, v_2)$: a directed edge of a network between vertices v_1 and v_2
V_R: The set of reticulation vertices (of some network N)
T_N: The underlying tree of a network
$\mathcal{T}_N$: The set of weighted support trees of N
$H(T)$: The hierarchy of a tree T
$P(T)$: The proper clusters of a tree T
$f(T)$: The rank of a tree T
S: The star tree
δ: A hierarchy-preserving map
$\leq_{HP}$: The poset relation induced by hierarchy-preserving maps
$\mathcal{H}(X)$: The Hasse diagram of $RP(X)$ under $\leq_{HP}$
d_{HP}: The distance on $\mathcal{H}(X)$
Δ_{HP}: The diameter of $RP(X)$ under d_{HP}
e_{HP}: A function that serves as an upper bound for d_{HP}

Contents

Chapter 1

Introduction

1.1 Phylogenetic Trees

The primary issue in phylogenetics is that for the vast majority of our evolutionary history on Planet Earth there were no phylogeneticists. As a result, we are forced to piece together the evolutionary history of organisms through clues found in the modern day, a process termed phylogenetic reconstruction.

Phylogenetic trees have been used to represent the relationships among a set of taxa labelling the leaves since at least 1755. Especially in the case of trees drawn with a root, the arcs of such rooted trees represent an evolutionary process proceeding over time away from the root and towards the leaves, and vertices in the tree represent divergence, or speciation, events.

Likewise, phylogenetic *networks* have come to prominence recently as a way to represent evolutionary processes in which branches of the tree interact with each other. Two key examples of such interactions are *hybridization*, in which genetic contributions from different lineages combine to give rise to a new lineage, and *horizontal gene transfer*, in which genetic material from one lineage is acquired by a second [26].

There are a variety of possible obstructions to phylogenetic reconstruction, and in this document we will consider those caused by similarity between different kinds of phylogenetic networks, for a number of different interpretations of the word 'similarity'. These will broadly fall into two categories - topological similarity and metric similarity.

We shall initially recall important details concerning phylogenetic trees (in both the rooted and unrooted contexts), before considering phylogenetic networks as a generalisation thereof.

1.1.1 Rooted Phylogenetic Networks

Definition 1.1.1. A *rooted phylogenetic network* N on a set of taxa X, is a rooted acyclic digraph (V, E) with vertices that fall into the following categories:

1. the *root vertex*, a unique vertex of in-degree 0;
2. the *tree vertices*, those of in-degree 1 and non-zero out-degree;
3. the *reticulation vertices*, those of out-degree 1; and
4. the *leaves* those vertices of in-degree 1 and out-degree 0, which are bijectively labelled by the set X; and

no non-root vertices of degree-2. The vertices other than the root and the leaves are called *internal* vertices. If the root has out-degree 2 and all non-leaf and non-root vertices of N have degree 3, N is referred to as *binary*. If there are no reticulation vertices in N, N is referred to as a *rooted phylogenetic tree*, commonly denoted T. The set of rooted phylogenetic trees on X is denoted $RP(X)$, and the set of binary rooted phylogenetic trees on X is denoted $BRP(X)$.

We write $V(N)$ for the set of vertices of N, and $E(N)$ for the set of arcs. In a rooted network, arcs are ordered pairs of vertices: we will denote the (directed) arc from u to v by (u, v), for $u, v \in V(N)$. Throughout this thesis, unless otherwise stated, we do not permit parallel edges (multiple copies of the same edge) or loops (an edge from a vertex to itself).

If (u, v) is a directed arc, then we say that u is the *parent* of v, and v is the *child* of u. If v and w are both children of the same vertex u, we say that v and w are *siblings* of each other.

By *suppressing* a vertex v of degree 2, we mean deleting the vertex v and the edges incident to it, (u_1, v) and (u_2, v), and adding a new edge (u_1, u_2). The reverse operation — deleting an edge (u_1, u_2) and replacing it with a new vertex v and a pair of edges (u_1, v) and (u_2, v) — is called *subdividing* the edge (u_1, u_2).

Let L be a set of leaves, and define *lowest stable ancestor* of L, denoted $LSA(L)$ as the lowest vertex that lies on every path from the root to a vertex in L. If N is a network, define the *restriction of N to L*, denoted $N|_L$ to be the network consisting of every vertex and edge on a directed path from $LSA(L)$ to a leaf in L.

Rooted phylogenetic trees are often described by their hierarchies (see for example [45]).

Definition 1.1.2. A *hierarchy* H on a set X is a collection of subsets of X with the following properties:

1. H contains both X and all singleton sets $\{x\}$ for $x \in X$.
2. If $H_1, H_2 \in H$, then $H_1 \cap H_2 = \varnothing$, $H_1 \subseteq H_2$ or $H_2 \subseteq H_1$.

Definition 1.1.3. Let $T \in RP(X)$ be a tree and v be a vertex of T. Then the *cluster* of T associated with v is the subset of X consisting of the descendants of v in T. If a cluster C is not X or a singleton, C is referred to as a *proper cluster*, and the set of proper clusters of T is denoted $P(T)$.

A collection of subsets of X is a hierarchy if and only if it is the set of clusters of some rooted phylogenetic tree T taken over all vertices of T (see [45] for instance). For this reason we refer to the set of clusters of T as the *hierarchy* of T, denoted $H(T)$.

A *horizontal gene transfer* (HGT) network is a binary phylogenetic network with some additional structure imposed.

Definition 1.1.4. A *HGT network* N on a set X is a rooted binary phylogenetic network with the following properties:

1. the arc set E of N is the disjoint union of two subsets, the set of 'reticulation arcs' E_R and the set of 'tree arcs' A_T; moreover each reticulation arc ends at a reticulation vertex, and each reticulation vertex has exactly one incoming reticulation arc;
2. every interior vertex has at least one outgoing tree arc; and
3. there is a time function $t : V \to \mathbb{R}$ so that (a) if (u, v) is a tree arc then $t(u) < t(v)$ and (b) if (u, v) is a reticulation arc, then $t(u) = t(v)$.

1.2 Unrooted Phylogenetic Networks

Definition 1.2.1 (Unrooted)**.** Let X be a finite, non-empty set. An *unrooted phylogenetic network* N *on* X is a connected graph (V, E) with $X \subseteq V$, no loops (edges of the form (v, v)) and no degree-2 vertices, such that the set of degree-1 vertices (referred to as *leaves*) is bijectively labelled by X. A network with $|V| = 1$ and $|E| = 0$ is called *trivial*. If all vertices of a phylogenetic network have degree 1 or 3, then it is termed *binary*. If N is acyclic, N is termed an *unrooted phylogenetic tree on* X.

Each rooted phylogenetic tree T can be made into an unrooted phylogenetic tree by suppressing the root vertex and converting all directed edges into undirected edges, although the unrooted tree formed in this way is not unique to T. Suppression and subdivision of edges are defined analogously to the operations in rooted phylogenetic networks.

1.3 When should we be worried about barking up the wrong tree(-like structure)?

As previously alluded to, there are several circumstances in which phylogenetic structures can be thought of as being very similar to either a particular tree, or a set of trees.

Recently, it has been shown that HGT networks can carry a metric that obeys the 'four-point' condition, which means that the same metric may also be realised

on a weighted tree [18]. This result was shown on a particular four-leaf network, and implies that showing that a given evolutionary history can be realised on a tree does not preclude the possibility that the history actually has a network structure, and there can be networks that are indistinguishable from a given tree from a metric perspective. This presents our first obstacle to phylogenetic reconstruction, as we now know that obeying the four-point condition is not sufficient to show that an evolutionary history took the form of a tree.

In Chapter 2 we examine just how bad this problem is. In particular we extend the result of Francis et al. [18] to show that no binary tree with at least 5 leaves is safe from this phenomenon, a troubling realisation. However, we also provide some initial results regarding the structure of a network that is indistinguishable from a given tree, so that even if no tree is safe from being displayed by a structurally different network, there are bounds on *how* structurally different they may be.

Furthermore a network may be *topologically* similar to a given tree. Indeed, it is of great interest whether networks can be considered tree-like, or whether they do not resemble trees at all. This has lead to the introduction of the concept of *tree-based* networks, which roughly speaking, are trees with additional arcs placed between certain edges in the tree. In this way, if it is tree-based, the evolutionary history can be considered to have a strong resemblance to a tree. This presents our second obstacle, as the existence of tree-based networks shows that networks can be quite similar in structure to trees.

In Chapter 3 we consider, in the context of unrooted, non-binary networks, several new questions. We define what it means to be tree-based in this new context, characterise networks that can be thought of as 'saturated' with tree signals, and provide some results on identification of tree-based networks. These results help to characterise when a network-shaped evolutionary history may present topological similarity to trees, or when networks carry strong tree-like signals.

We do not even need to look outside tree space to find cases where strong tree signals can cause confusion. A given tree may be mistaken for another just due to similarity in their tree structures. Metrics on tree space allow us to quantify similarity, and what 'similarity' means can be encoded into our metrics. Metrics on tree space are used in many areas of biology, where the utility of a given metric is judged on a number of merits, including speed of calculation, ease of traversal in tree space and ease of generation of a 'neighbourhood' of trees that are similar to a given tree. One way in which two trees may resemble each other is by having similar hierarchical structures, which can occur with trees that have arisen under related processes, such as gene trees in the presence of incomplete lineage sorting. In fact, two metrics based on hierarchical similarity of trees have recently been developed.

In Chapter 4 we develop a third one with several potential benefits in terms of utility - it can be easily estimated, it includes a 'local operation' which allows for easy computation of neighbourhoods and the space is highly interconnected, allowing for ease of tree space traversal. We can therefore characterise which trees are similar to each other, and therefore know which trees to be wary of.

We have thus studied three mathematical notions related to the similarity of

phylogenetic trees and networks and argued that these may present obstacles to phylogenetic reconstruction. Each presents a case study of how phylogenetic structures can be similar, how to characterise those structures that are similar, and in the third case, how to measure such an error. We hope this allows phylogeneticists to avoid these obstacles, or at least be aware of their existence.

Chapter 2

Metric Similarity

2.1 Introduction

Phylogenetic trees have been used to represent the relationships among a set of taxa labelling the leaves since the days of Darwin [13], and even before that. Especially in the case of trees drawn with a root, the arcs of such rooted trees represent an evolutionary process proceeding over time away from the root and towards the leaves, and vertices in the tree represent divergence, or speciation, events. Likewise, phylogenetic *networks* have come to prominence recently as a way to represent evolutionary processes in which branches of the tree interact with each other. Two key examples of such interactions are *hybridization*, in which genetic contributions from different lineages combine to give rise to a new lineage, and *horizontal gene transfer*, in which genetic material from one lineage is acquired by a second [26].

In particular, horizontal gene transfer is highly relevant for studies of evolutionary history — it is thought to be the primary driver of early cellular evolution [46], and still is relevant to ongoing evolution, with over half of total genes in the genomes of human-associated microbiota involved in horizontal gene transfer [29]. We will therefore focus in particular on HGT networks throughout this chapter.

While phylogenetic trees and networks can be constructed in many ways, current approaches often involve a metric on the set of taxa. That is, a matrix giving the pairwise distances between each pair of leaves of the tree or network. While such distances are natural to define on a tree, there are different ways one may define the distance between leaves in a network; we will give more details of the approach we take to this, below.

In this chapter we are concerned with metrics on a set of taxa that are able to be placed on a tree — "tree metrics" — but that can also be placed on a network. It was recently observed that some tree metrics have this property, and the resulting networks that have a single "reticulation" were characterized [18]. This chapter extends this by investigating networks with *more* than one reticulation that can nevertheless carry tree metrics. We call such networks "tree-metrizable". Such tree-metrizable networks present a serious problem for phylogenetic reconstruction, as data that appears to come from a tree cannot be guaranteed to have done so.

Fortunately, there is an explicit characterization of when a metric can be placed on a tree. The famous "four-point condition" (Theorem 2.2.1), due to Buneman [5], says that if a metric d on a set X satisfies the condition then there exists a unique weighted phylogenetic tree with leaf-set X whose induced metric is d. This, incidentally, provides a characterisation of weighted phylogenetic trees on X: a pair of trees are isomorphic as weighted graphs if and only if their induced metrics are identical.

Surprisingly, it has recently been shown [18] that it is possible for both hybridization networks and HGT networks (defined in Section 2.2.2) to produce metrics that satisfy the four-point condition. That is, they may carry tree metrics. The implication is that a metric being a tree metric cannot rule out the evolutionary history of the taxa X being explained by a network. In fact, *any* tree metric can be displayed by a network ([18, Theorem 2]), although it may be a very simple one.

A natural question then, is what phylogenetic networks might possibly carry tree metrics? We call such networks *tree-metrizable networks*. This question, for (binary) hybridization networks, was answered in [18]: the answer was "not many". There are tight restrictions on where hybridizations can occur for the network to carry a tree metric. The case of HGT networks, however, was left open. Conditions on networks with a single HGT arc were established, but an example of a HGT network with two HGT arcs was given that carries a tree metric, and what's more, the tree metric corresponded to a tree that was not a base-tree of the network (in the sense of [19])!

This chapter seeks to explore this phenomenon. That is, we are interested in the situation in which the inferred metric from a weighted rooted binary HGT network might satisfy the four-point condition, and so be indistinguishable from a tree. The key approach in this chapter is to graft structures on to the leaf of a tree or network, while maintaining the existence of a metric on the leaves. With such tools, complicated tree-metrizable networks can be built up from a base tree or network.

The chapter begins with background definitions and results on metrics on trees and HGT networks (Section 2.2). We then begin our exploration of tree-metrizable HGT networks in Section 2.3, by first extending the four-point condition to the HGT network context, and then deriving some natural extensions to the results of [18]. These results effectively show that tree-metrizable networks can be constructed with any number of HGT arcs at all, by adding certain HGT arcs (Lemma 2.3.4).

The final two sections show how complicated tree-metrizable networks can be constructed by grafting trees onto small tree-metrizable networks (Section 2.4), and then the reverse (Section 2.6). The chapter concludes with a discussion of the results and some further questions, in Section 2.7.

The results in this chapter have been published [24].

2.2 Background

2.2.1 Trees and tree metrics

Unless otherwise stated, all trees in this chapter are rooted binary phylogenetic X-trees.

We will say two trees T_1 and T_2 are isomorphic if there is a one-to-one correspondence between their vertices $\phi : V(T_1) \to V(T_2)$ that also maps their edges: $E(T_2) = \{(\phi(u), \phi(v)) \mid (u, v) \in E(T_1)\}$ and preserves leaf labels. Note that if isomorphic trees are rooted, their roots must map to each other.

In particular, there are exactly 3 isomorphism classes of unrooted trees on a set of taxa X for $|X| = 4$ (referred to as a *quartet*). If $X = \{a, b, c, d\}$, and we denote the tree in which the unique paths from a to b and c to d do not intersect by $ab|cd$, then these classes correspond to $ab|cd$, $ac|bd$ and $ad|bc$.

A *weight function* on a rooted binary phylogenetic X-tree $T = (V, E)$ is a map $w : E \to \mathbb{R}^{>0}$ that assigns strictly positive weights to the arcs of a tree. We denote a tree T with associated weight function w by T^w. This allows us to define the distance $d(x, y)$ between two leaves x and y in X to be the sum of the weights on the arcs in the unique path between x and y. This distance is referred to as the *tree distance* between x and y, and any set of pairwise distances between elements of X that can be represented on a tree in this way is referred to as a *tree metric*. If there are two trees $T_1^{w_1}$ and $T_2^{w_2}$ such that T_1 and T_2 are isomorphic as unrooted trees (but not necessarily $w_1 = w_2$), we denote this as $T_1 \cong T_2$. If $w_1 = w_2$ as well, we will refer to $T_1^{w_1}$ and $T_2^{w_2}$ as *isomorphic as weighted trees*, denoted $T_1^{w_1} \cong_w T_2^{w_2}$, or $T_1 \cong_w T_2$ when the corresponding weight functions are clear from context.

A fundamental characterisation of tree metrics is the 'four point condition'.

Theorem 2.2.1 (Four point condition [5])**.** *A distance function d on a set X is a tree metric on X if and only if for any $x_1, x_2, x_3, x_4 \in X$, two of the three sums*

$$d(x_1, x_2) + d(x_3, x_4); \quad d(x_1, x_3) + d(x_2, x_4); \quad d(x_1, x_4) + d(x_2, x_3)$$

are equal, and are greater than the third sum.

2.2.2 HGT networks

A *horizontal gene transfer* (HGT) network is a generalisation of a binary phylogenetic tree that allows the modelling of certain reticulation events. Recall the following definition.

Definition 2.2.2. A *HGT network* N on a set X is a rooted acyclic digraph (V, E) with the following properties:

1. the *root* vertex has in-degree 0 and out-degree 2;
2. X labels the set of vertices with out-degree 0 and in-degree 1 (the *leaves*);

3. all remaining vertices are *interior vertices* and have either in-degree 1 and out-degree 2 (a *tree vertex*), or in-degree 2 and out-degree 1 (a *reticulation vertex*);
4. the arc set E of N is the disjoint union of two subsets, the set of 'reticulation arcs' E_R and the set of 'tree arcs' E_T; moreover each reticulation arc ends at a reticulation vertex, and each reticulation vertex has exactly one incoming reticulation arc;
5. every interior vertex has at least one outgoing tree arc; and
6. there is a time function $t : V \to \mathbb{R}$ so that (a) if (u, v) is a tree arc then $t(u) < t(v)$ and (b) if (u, v) is a reticulation arc, then $t(u) = t(v)$.

Informally, one can think of a HGT network as a binary phylogenetic X-tree for which certain arcs are subdivided and a horizontal arc is placed between the subdivisions. Throughout this chapter, unless otherwise stated, all networks are HGT networks.

Given a HGT network N on X, suppose that for each reticulation vertex we delete exactly one of the incoming arcs, and if a child arc of the root vertex is deleted, make the remaining child of the root the new root and delete the original root and its outgoing arc. If we then delete any unlabelled leaves formed by this process, the resulting graph is a rooted tree on X. If we then repeat this as many times as necessary to eliminate all unlabelled leaves, and then suppress all of the degree 2 vertices aside from the root, the resulting graph is a rooted binary phylogenetic X-tree, T. We say that T is displayed by N, and $\mathcal{T}_N$ denotes the set of trees displayed by N. See Figure 2.1 for an example of this process.

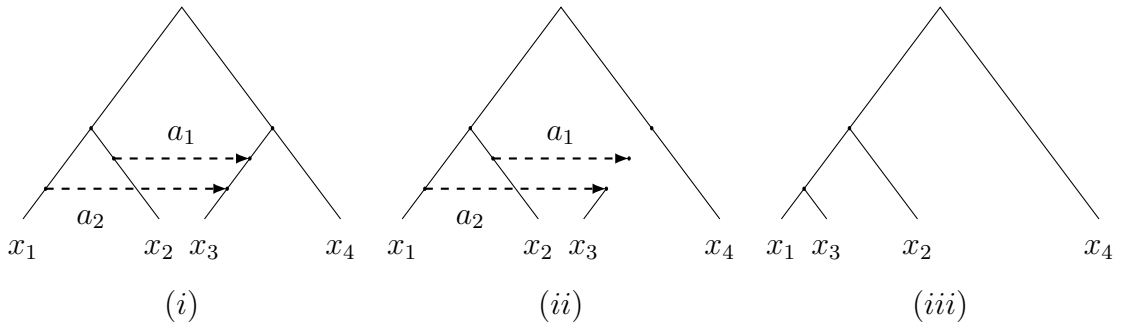


Figure 2.1: (i) a HGT network N with HGT arcs a_1, a_2. Denote the other parent arcs of the reticulation vertices by a_1', a_2' respectively; (ii) The resulting graph after deleting a_1', a_2'; (iii) The resulting display tree after deletion of unlabelled leaves and suppression of degree 2 nodes.

HGT networks have the particularly useful property of having a 'canonical' display tree, obtained by deleting all of the reticulation arcs. This tree is referred to as the *underlying tree* of N, and is denoted T_N. Note that the underlying tree of a HGT network is a *base* tree in the sense of [19], but is not necessarily the only base tree of the network.

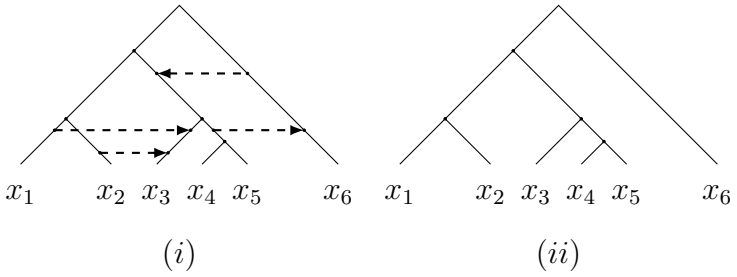


Figure 2.2: (i) a HGT network N, with reticulation arcs shown dashed; (ii) the underlying tree T_N of N.

2.2.3 HGT network distances

Following [18], we define distances on a HGT network N by treating it as a weighted union of the set of X-trees obtained by making choices at each reticulation. For each vertex v in the set V_R of reticulation vertices of N, let $R(v)$ denote the two arcs that end at v. We write N^w for a HGT network N with w a weight function on the tree arcs, $w : E_T \to \mathbb{R}^{>0}$ and let β be a strictly positive probability distribution on the set F_N of functions $f : V_R \to E$ for which $f(v) \in R(v)$. Each function f describes a weighted tree T_f with induced weight function w_f, by specifying a parent for each reticulation vertex. This function f, and its tree T_f, can then be given an associated probability β_f, which we construct as follows.

For each reticulation vertex v with incoming arcs $R(v) = \{a, a'\}$, we associate a function α that gives a number between 0 and 1 to each such reticulation arc, $\alpha : R(v) \to (0, 1)$, that satisfies $\alpha(a) + \alpha(a') = 1$. We refer to $\alpha(a)$ and $\alpha(a')$ as the *reticulation probabilities* of a and a' respectively. Then let

$$\beta_f = \prod_{v \in V_R} \alpha(f(v)). \tag{2.1}$$

That is, β_f is the product of the weights on the arcs chosen by f for each reticulation vertex.

A distance function

$$d = d_{(N^w,\beta)} : X \times X \to \mathbb{R}^{\geq 0}$$

on X can then be defined for N^w by setting

$$d(x, y) = \sum_{f \in F_N} \beta_f d_{\left(T_f^{w_f}\right)}(x, y), \tag{2.2}$$

where w_f is the weight function induced by N^w on T_f. If there are no reticulation vertices in N^w, d is the tree metric d_{T^w}. As noted in [18, §2.3], since $d_{(N^w,\beta)}$ is a convex combination of metrics on X, d is also a metric.

We denote the set of weighted trees obtained in this way from N^w by $\mathcal{T}_N^w$.

The probability distribution on functions f naturally corresponds to a probability distribution on the associated trees $T_f^w \in \mathcal{T}_N^w$, by setting $\beta(T_f^w) = \beta(f)$. We will drop the reference to w and f where this is not explicitly needed, writing $\beta(T)$.

Example 2.2.3. Consider the weighted HGT network N^w with horizontal reticulation arcs a_1, a_2, a_3 shown in Figure 2.3. We intend to calculate the associated probability β_f of a particular weighted display tree. We have omitted the weights from the diagram for ease of interpretation, but note that we are calculating β_f for a particular *weighted* display tree.

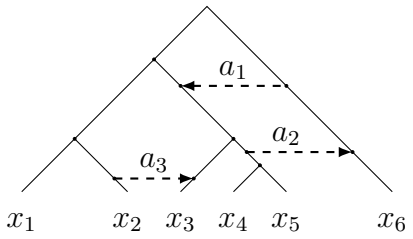


Figure 2.3: A weighted network N^w (with weights omitted) on 6 leaves.

Denote the second arc ending at the same vertex as a_1, a_2 and a_3 respectively by a_1', a_2' and a_3'. Then by making the selection a_1, a_2', a_3 (and thus deleting a_1', a_2 and a_3'), we obtain the following display tree T^w.

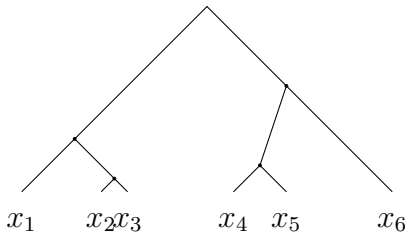


Figure 2.4: The weighted display tree T^w (with weights omitted) obtained from N^w in Figure 2.3 by deleting a_1', a_2 and a_3'.

Then if the reticulation probabilities are $\alpha(a_1) = 0.6, \alpha(a_2) = 0.2, \alpha(a_3) = 0.1$, then the probability assigned to T^w is

$$\begin{aligned}\beta(T) &= \alpha(a_1)\alpha(a_2')\alpha(a_3)\\ &= \alpha(a_1)(1-\alpha(a_2))\alpha(a_3)\\ &= 0.6 \times 0.8 \times 0.1\\ &= 0.048.\end{aligned}$$

Somewhat surprisingly, it has recently been shown that HGT networks under this weighted average distance model can obey the four-point condition [18]. That is, the distances represented by some HGT networks can also be represented on a unique tree. We call such networks "tree-metrizable":

Definition 2.2.4. Let N be a HGT network on X. If there exist arc weights and reticulation probabilities that can be placed on N so that d_N is a tree metric that can be placed on some unweighted tree T, we say that N is *tree-metrizable*, or specifically *T-metrizable*.

We now provide some of the background results on tree-metrizable networks, with wording changed to use the language of tree-metrizability. Throughout this chapter, we say that two arcs are adjacent if they are both child arcs of the root vertex, or they are adjacent in the unrooted tree obtained by suppressing the root vertex.

Lemma 2.2.5 ([18], Lemma 4)**.** *For any unweighted HGT network N, if each reticulation arc is between adjacent tree arcs of T_N, then N is T-metrizable if and only if $T \cong T_N$.*

It easily follows as a side note that any weighted network on 3 leaves is T-metrizable for T any 3-leaf tree, as all arcs in a 3-leaf tree are adjacent.

In light of this result, we want to focus on networks that can potentially represent tree metrics that are *not* the underlying tree.

Definition 2.2.6. Let N be a HGT network with reticulation arc A. If A is between two non-adjacent arcs of the underlying tree, we say that A is a *non-trivial* reticulation arc. If N contains at least one non-trivial reticulation arc, then N is said to be a *non-trivial* HGT network.

Of course, this distinction would not be very useful in our context if there were no networks that were tree-metrizable on a tree that was not the underlying tree. The proof of the following theorem involves constructing such a network on four leaves with two reticulations.

Theorem 2.2.7 ([18], Theorem 5(b))**.** *There exist 2-reticulated HGT networks N that are T_N-metrizable and (for other parameter settings) T-metrizable for $T \ncong T_N$, even when the mixing distribution treats the two reticulations independently.*

The following simple (yet surprisingly powerful) result will be our primary tool for showing a network is not tree-metrizable throughout this chapter.

Lemma 2.2.8 ([18], Lemma 6)**.** *Let N be a HGT network with display trees $\mathcal{T}_N = \{T_1, \ldots, T_k\}$. Suppose that there is a quartet $q \subseteq X$ in the unrooted sense, for which $\left|\{T_i|_q\}_{i=1,\ldots,k}\right| = 2$. Then N is not tree-metrizable.*

An immediate consequence of this lemma is that if a network contains a single non-trivial reticulation arc, it is certainly not tree-metrizable, as it will have exactly two non-isomorphic display trees (in both the rooted and unrooted senses). One can then find a quartet upon which the two trees do not agree and then apply the lemma.

2.3 Tree-metrizability: first results

The following lemma will make our calculations involving the four-point condition easier by phrasing the four-point condition in terms of the lengths of the internal arcs of a quartet, instead of the tree distances between leaves.

Definition 2.3.1. For T a rooted tree, let T^U be the unrooted tree obtained by suppressing the root vertex. That is, if r is the root vertex, we delete the vertex r and edges (r,u) and (r,v), then add (u,v). All edges are then interpreted as undirected.

Lemma 2.3.2. *Let N be a four-leaf HGT network with exactly three display trees, $\mathcal{T}_N = \{T_r, T_s, T_t\}$, and let $\{T_1^{w_1}, \ldots, T_k^{w_k}\}$ be the weighted trees obtained from N by choices of reticulations. Let α_j be the probability assigned to T_j, and p_j be the length of the internal arc of T_j^U.*

Then N is tree-metrizable on T_r if and only if

$$\sum_{T_j \cong T_r} p_j\alpha_j > \sum_{T_j \cong T_s} p_j\alpha_j = \sum_{T_j \cong T_t} p_j\alpha_j.$$

Proof. Label the four leaves $\{x_1, x_2, x_3, x_4\}$. Let

$$d_i = \sum_{T_j \cong T_i} \alpha_j d_{(T_j^{w_j})}$$

for $i = r, s, t$, and let $d = d_1 + d_2 + d_3$. Let $a_i = d_{(T_i^{w_i})}(x_1, x_2) + d_{(T_i^{w_i})}(x_3, x_4)$, $b_i = d_{(T_i^{w_i})}(x_1, x_3) + d_{(T_i^{w_i})}(x_2, x_4)$ and $c_i = d_{(T_i^{w_i})}(x_1, x_4) + d_{(T_i^{w_i})}(x_2, x_3)$. Define

$$\begin{aligned} A_i = d_i(x_1, x_2) + d_i(x_3, x_4) &= \sum_{T_j \cong T_i} a_j\alpha_j \\ B_i = d_i(x_1, x_3) + d_i(x_2, x_4) &= \sum_{T_j \cong T_i} b_j\alpha_j \\ C_i = d_i(x_1, x_4) + d_i(x_2, x_3) &= \sum_{T_j \cong T_i} c_j\alpha_j. \end{aligned}$$

Without loss of generality, suppose T_r, T_s and T_t are the quartets $x_1x_2|x_3x_4$, $x_1x_3|x_2x_4$ and $x_1x_4|x_2x_3$ respectively. It follows that $A_r < B_r = C_r, B_s < A_s = C_s$, and $C_t < A_t = B_t$.

We shall now find a probability distribution so that d is a tree metric on $x_1x_2|x_3x_4$. Let $S_1 = d(x_1, x_2) + d(x_3, x_4), S_2 = d(x_1, x_3) + d(x_2, x_4)$, and $S_3 = d(x_1, x_4) + d(x_2, x_3)$. Then d is a tree metric iff $S_1 < S_2 = S_3$. Considering $S_2 = S_3$ first,

$$\begin{aligned} & d(x_1, x_3) + d(x_2, x_4) = d(x_1, x_4) + d(x_2, x_3) \\ \implies \quad & B_r + B_s + B_t = C_r + C_s + C_t \\ \implies \quad & B_t - C_t = C_s - B_s \\ \implies \quad & \sum_{T_j \cong T_t} b_j\alpha_j - \sum_{T_j \cong T_t} c_j\alpha_j = \sum_{T_j \cong T_s} c_j\alpha_j - \sum_{T_j \cong T_s} b_j\alpha_j \\ \implies \quad & \sum_{T_j \cong T_t} (b_j - c_j)\alpha_j = \sum_{T_j \cong T_s} (c_j - b_j)\alpha_j, \end{aligned}$$

which, noting that $b_j - c_j = 2p_j$ for $T_j \cong T_t$ and $c_j - b_j = 2p_j$ for $T_j \cong T_s$ implies that

$$\sum_{T_j \cong T_t} p_j \alpha_j = \sum_{T_j \cong T_s} p_j \alpha_j,$$

which is the desired equality for this lemma.

Now considering the requirement that $S_1 < S_2$, we have

$$\begin{aligned}
& d(x_1, x_2) + d(x_3, x_4) < d(x_1, x_3) + d(x_2, x_4) \\
\implies \quad & A_r + A_s + A_t < B_r + B_s + B_t \\
\implies \quad & A_s - B_s < B_r - A_r \\
\implies \quad & \sum_{T_j \cong T_s} a_j \alpha_j - \sum_{T_j \cong T_s} b_j \alpha_j < \sum_{T_j \cong T_r} b_j \alpha_j - \sum_{T_j \cong T_r} a_j \alpha_j \\
\implies \quad & \sum_{T_j \cong T_s} (a_j - b_j) \alpha_j < \sum_{T_j \cong T_r} (b_j - a_j) \alpha_j
\end{aligned}$$

which, noting that $a_j - b_j = 2p_j$ for $T_j \cong T_s$ and $b_j - a_j = 2p_j$ for $T_j \cong T_r$ implies that

$$\sum_{T_j \cong T_s} p_j \alpha_j < \sum_{T_j \cong T_r} p_j \alpha_j,$$

which is the inequality required for this lemma. The remaining cases (e.g. when $T_r = x_1x_3|x_2x_4, T_s = x_1x_2|x_3x_4$ and $T_t = x_1x_4|x_2x_3$, etc.) are proved similarly. □

Example 2.3.3. Lemma 2.3.2 somewhat surprisingly reveals that tree-metrizability is dependant only on the internal arcs of the display trees. For example, let N be the HGT network shown in Figure 2.5. Then, in order for N to be T_1-metrizable, for example, by Lemma 2.3.2 we require that

$$(1 - \alpha_1)(1 - \alpha_2)(a_1 + a_4) \geq \alpha_1(1 - \alpha_2)a_3 = \alpha_2 a_1.$$

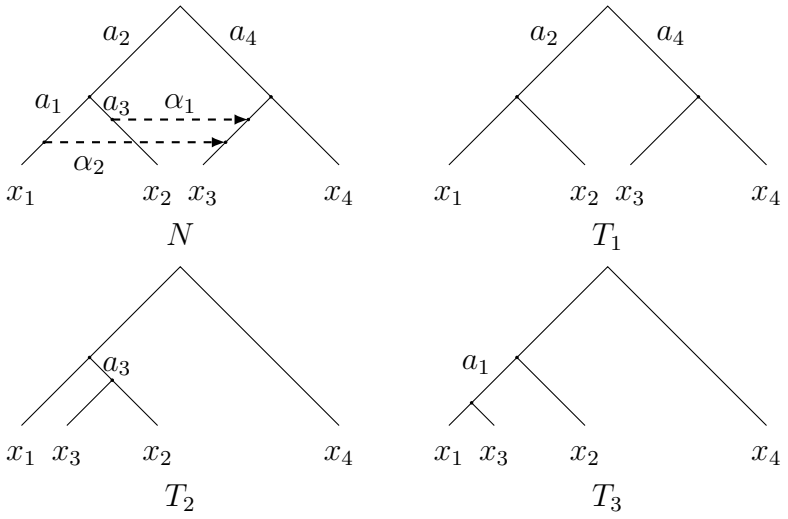


Figure 2.5: A HGT network N with its three display trees, T_1, T_2 and T_3.

The next lemma shows that certain substructures in a weighted HGT network can be interchanged without changing the metric on the leaves.

Recall that Lemma 2.2.5 says that if all of the HGT arcs in a HGT network are between adjacent tree arcs, then the network is tree-metrizable. In Lemma 2.3.4, we generalise this to show how a HGT arc can be placed between any adjacent pair of tree arcs in a network, and retain tree distances (and hence tree-metrizability, if our network is tree-metrizable). In particular, this implies that tree-like distances are preserved with the addition of an arc between adjacent arcs, even if the network contains other arcs between non-adjacent arcs.

Lemma 2.3.4. *Let N^w be a HGT network with reticulation probabilities β, and a HGT arc a between a pair of siblings such that there is no other vertex between the ends of a and their common parent vertex. Let $\widehat{N}$ be the HGT network obtained by deleting a from N and suppressing the vertices at each end. Then there exist arc weights $\widehat{w}$ and reticulation probabilities $\widehat{\beta}$ on N_1 so that*

$$d_{(\widehat{N}^{\widehat{w}},\widehat{\beta})} = d_{(N^w,\beta)}.$$

Proof. Suppose the weighted display trees of N^w are $\mathcal{T}_N^w = \{T_1^{w_1}, \ldots, T_{2^r}^{w_{2^r}}\}$ with respective probabilities β_i and the weighted display trees of $\widehat{N}$ are $\mathcal{T}_{\widehat{N}}^{\widehat{w}} = \{\widehat{T}_1^{\widehat{w}_1}, \ldots, \widehat{T}_{2^{r-1}}^{\widehat{w}_{2^{r-1}}}\}$ with respective probabilities $\widehat{\beta}_i$.

Consider $\mathcal{T}_{\widehat{N}}^{\widehat{w}}$. We can associate each weighted display tree $\widehat{T}_i^{\widehat{w}_i}$ of $\widehat{N}^{\widehat{w}}$ in a natural way to a pair of weighted display trees of N^w, by considering those trees made with the same selection of arcs as N plus either keeping or deleting h. Rearrange the indexing of $\mathcal{T}_N^w$ if necessary so that $\widehat{T}_i^{\widehat{w}_i}$ is associated with $T_{2i-1}^{w_{2i-1}}, T_{2i}^{w_{2i}}$. If we can ensure that

$$\beta_{2i-1} d_{(T_{2i-1}^{w_{2i-1}})} + \beta_{2i} d_{(T_{2i}^{w_{2i}})} = \widehat{\beta}_i d_{(\widehat{T}_i^{\widehat{w}_i})}$$

then the lemma is proven.

To this end, set all arc weights and reticulation probabilities to be identical in N^w and $\widehat{N}^{\widehat{w}}$ except for those depicted in Figure 2.6. Label v_1, v_2, v_3 as in Figure 2.6 such that there are no vertices between them that are not shown in the diagram. Then we label the arc weights and reticulations between v_1, v_2 and v_3 in N^w and $\widehat{N}^{\widehat{w}}$ as in Figure 2.6.

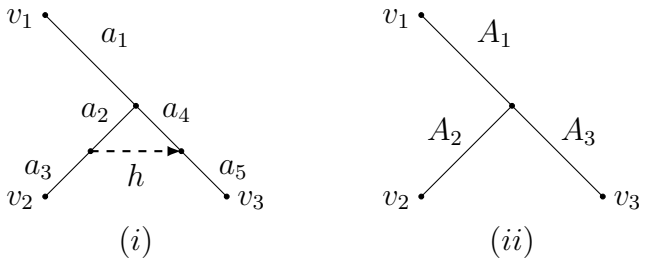


Figure 2.6: (i) The section between vertices v_1, v_2 and v_3 in N^w; (ii) The corresponding section of $\widehat{N}^{\widehat{w}}$.

We now consider the resulting weighted display trees. Of course, v_2 and v_3 may not appear in T_{2i-1} and T_{2i}, but this will occur if and only if those same arcs/vertices do not appear in $\widehat{T}_i$. The corresponding arc weights of the weighted display trees obtained by keeping/deleting h (in the case that v_2 and v_3 are in the display tree) are shown in Figure 2.7.

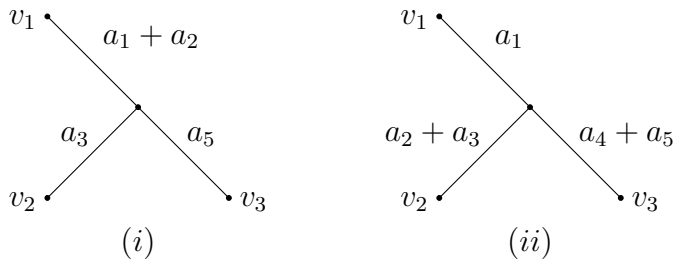


Figure 2.7: (i) The relevant section of a display tree of N that contains vertices v_1, v_2, v_3 when keeping α; (ii) The corresponding section with deletion of α.

If we can set values for A_1, A_2, A_3 so that

$$d_{(\widehat{T}_i^{\widehat{w}_i}, \widehat{\beta}_i)}(s,t) = d_{(T_{2i-1}^{w_{2i-1}}, \beta_{2i-1})}(s,t) + d_{(T_{2i}^{w_{2i}}, \beta_{2i})}(s,t)$$

for $(s,t) = (v_1, v_2), (v_1, v_3)$ and (v_2, v_3), then we achieve the required result. In particular, this would follow if these equalities hold:

$$\begin{aligned}
A_1 + A_2 &= \alpha(a_1 + a_2 + a_3) + (1-\alpha)(a_1 + a_2 + a_3) && \text{(distance } v_1 \text{ to } v_2\text{)} \\
&= a_1 + a_2 + a_3 \\
A_1 + A_3 &= a_1 + a_5 + \alpha a_2 + (1-\alpha)a_4 && \text{(distance } v_1 \text{ to } v_3\text{)} \\
A_2 + A_3 &= a_3 + a_5 + (1-\alpha)(a_2 + a_4). && \text{(distance } v_2 \text{ to } v_3\text{)}
\end{aligned}$$

Setting $A_1 = a_1 + \alpha a_2$, $A_2 = a_3 + (1-\alpha)a_2$ and $A_3 = a_5 + (1-\alpha)a_4$ satisfies these criteria, and all of these values are positive. This completes the result. □

2.4 Leaf-Grafting

Theorem 2.2.7 provides an example of a HGT network on four leaves that has two non-trivial reticulation arcs but still is tree-metrizable. In the previous section, we addressed the question of whether tree-metrizable HGT networks with more reticulation arcs exist; in this section we address the analogous question for the number of leaves. In particular, we will show how "leaf-grafting" trees onto the leaves of a tree-metrizable HGT network can create a non-trivial tree-metrizable network on any base tree at all.

We begin by defining the notion of *leaf-grafting*.

Definition 2.4.1. Let N and N' be two HGT networks on X and Y respectively, and x a leaf of N. If we identify the root of N' with the leaf x of N, the resulting

network $N\#_x N'$ on $X \cup Y - \{x\}$ is termed a *leaf-graft of N' onto N at leaf x*. In particular, N is referred to as the *stock*, N' is referred to as the *scion* and x as the *grafting vertex*.

To begin with, we address the case where the stock is a HGT network and the scion is a tree, that is, grafting a tree onto a network. We will consider the reverse problem later. Recall the following relevant definition.

Definition 2.4.2 ([14], Section 3)**.** Let N be a HGT network and $L \subseteq V(N)$. Then the *lowest stable ancestor* of L, denoted $LSA(L)$ is the lowest vertex that lies on every path from the root to a vertex in L.

We note that the lowest stable ancestor must always exist, as at the very least the root will fit the criterion.

Theorem 2.4.3 (Replacement Theorem)**.** *Let N be a tree-metrizable network, T be a tree, and ℓ a leaf of N. Then $N\#_\ell T$ is a tree-metrizable network if and only if N is a tree-metrizable network.*

Proof. Suppose N is tree-metrizable. Consider a quartet of leaves $q = \{x_1, x_2, x_3, x_4\}$ of $N\#_\ell T$. We will show that the distances between these leaves satisfy the inequality in the four point condition.

Let the leaves of T be denoted by Y and the leaves of N denoted by X, so that the leaf set of $N\#_\ell T$ is $(X\backslash\{\ell\}) \cup Y$. We consider cases according to how many of the leaves of q are in Y.

Case (4): All four leaves of q are in Y. In this case the distances between the leaves are determined by their distances in T, and so satisfy the four point condition.

Case (3): Three leaves (say x_1, x_2, x_3) are in Y while the fourth, x_4, is in X. Let $x = LSA\{x_1, x_2, x_3\}$, and suppose without loss of generality that $x_1x_2|x_3$ forms a rooted triple with root ρ. Then in the network $N\#_\ell T$, the distances between x_1, x_2, x_3 and ρ are all determined by the tree T. All distances between x_1, x_2 or x_3 and x_4 go through ρ, so that for instance $d(x_1, x_4) = d(x_1, \rho) + d(\rho, x_4)$. It is immediate that the inequality holds in this case (namely $d(x_1, x_2) + d(x_3, x_4) \leq d(x_1, x_3) + d(x_2, x_4) = d(x_1, x_4) + d(x_2, x_3)$).

Case (2): Suppose x_1, x_2 are leaves of Y and x_3, x_4 are leaves of X. Then

$$d(x_1, x_3) = d_T(x_1, \ell) + d_N(\ell, x_3), \tag{2.3}$$

and likewise for other cross-pairs $d(x_1, x_4), d(x_2, x_3)$ and $d(x_2, x_4)$. It is easy to check that $d(x_1, x_3)+d(x_2, x_4) = d(x_1, x_4)+d(x_2, x_3)$. The inequality $d(x_1, x_2)+d(x_3, x_4) < d(x_1, x_3) + d(x_2, x_4)$ also follows using the observation that $d_T(x_1, x_2) < d_T(x_1, \ell) + d_T(x_2, \ell)$ (and similarly for $d(x_3, x_4)$), by the triangle inequality.

Case (1): Suppose that x_1 is in Y and $x_2.x_3, x_4$ are in X. Then the pairwise distances between x_2, x_3 and x_4 are determined by their distances in N, while the distance $d(x_1, x_2) = d_T(x_1, \ell) + d_N(x_2, \ell)$ and similarly for the distances from x_1 to x_3 and x_4. Then as ℓ was a leaf of a tree-metrizable network, it follows that the pairwise distances obey the four-point condition.

Case (0): If all leaves in q are in X, then their pairwise distances are determined by their distances in N, and so satisfy the four-point condition as N is tree-metrizable.

It follows that $N\#_\ell T$ is tree-metrizable.

Now suppose that N is not tree-metrizable. It follows that there exists a quartet of leaves $q' = \{x_1, x_2, x_3, x_4\}$ of N that does not obey the four-point condition. If q' does not contain ℓ, then the same quartet in $N\#_\ell T$ does not obey the four-point condition, and therefore $N\#_\ell T$ is not tree-metrizable.

If q' does contain ℓ, suppose $q' = \{x_1, x_2, x_3, \ell\}$. Then we can select some leaf k of T, and observe that $d(x_1, k) = d_N(x_1, \ell) + d(\ell, k)$, with similar forms for x_2 and x_3. We now consider the distances arising from the quartet $\{x_1, x_2, x_3, k\}$:

$$d(x_1, x_2) + d(x_3, k); \quad d(x_1, k) + d(x_2, x_3); \quad d(x_1, x_3) + d(x_2, k).$$

We can see that

$$\begin{aligned} d(x_1, x_2) + d(x_3, k) &= d_N(x_1, x_2) + d_N(x_3, \ell) + d(\ell, k), \\ d(x_1, k) + d(x_2, x_3) &= d_N(x_1, \ell) + d_N(x_2, x_3) + d(\ell, k), \\ d(x_1, x_3) + d(x_2, k) &= d_N(x_1, x_3) + d_N(x_2, \ell) + d(\ell, l). \end{aligned}$$

In particular, these are just the corresponding distances of the four-point condition applied to $q' = \{x_1, x_2, x_3, \ell\}$, each with the same distance $d(\ell, k)$ added. It follows that if q' does not obey the four-point condition, neither does $\{x_1, x_2, x_3, k\}$. Therefore $N\#_\ell T$ is not tree-metrizable. □

Corollary 2.4.4. *There exist non-trivial tree-metrizable networks with 2 HGT arcs on n leaves for $n \geq 4$. Furthermore, there exist networks N on n leaves that are T-metrizable for some tree T that is not the underlying tree T_N.*

Proof. Theorem 2.2.7 proves this for $n = 4$.

If $n > 4$, simply take the network N with 2 HGT arcs described in Theorem 2.2.7 and leaf-graft some tree T with $n-3$ leaves onto any leaf of N. Theorem 2.2.7 also provides an example where a quartet represents a tree that is not its underlying tree. Using this example as our network N provides an example for the second part of this result. □

2.5 Caterpillar Networks

Leaf-grafting provides a neat method for constructing tree-metrizable networks on an arbitrary number of leaves with interesting properties. We will now define a class of tree-metrizable HGT networks on n leaves with $n-2$ non-trivial reticulations, referred to as *caterpillar networks*. In combination with leaf-grafting, this result shows that any tree T of height h can be represented on a tree-metrizable HGT network with $h-1$ reticulation arcs.

We will require the following standard definition.

Definition 2.5.1. Let T be a tree, and suppose the leaves x_1 and x_2 are both children of the same vertex v. Then we say that x_1 and x_2 form a *cherry*, and denote it $\widehat{x_1x_2}$.

We can now define caterpillar networks.

Definition 2.5.2. Let C be a HGT network with a caterpillar underlying tree T on $n > 3$ leaves. Let C be depicted with each internal tree vertex the left child of its parent vertex, and label the leaves $x_1, \ldots, x_n$ from left to right, so that $\widehat{x_1x_2}$ is the unique cherry in T. For $1 \leq i < n-1$, let each leaf x_i have a reticulation arc extending to leaf x_n, so that the arcs are attached to x_n in numerical order from bottom to top. Then C is referred to as a *caterpillar network*.

For a caterpillar network C on n leaves, let T_i $(1 \leq i \leq n-2)$ denote the unique display tree containing the cherry $\widehat{x_ix_n}$, and let T_{n-1} be the underlying tree of C. This uniquely defines T_i because each display tree of N can only contain at most one of the reticulation arcs added to T, because they all end on the same tree edge between the root and leaf x_n. Thus C contains exactly $n-1$ display trees (so that $\mathcal{T}_C = \{T_1, \ldots, T_{n-1}\}$), with T_i $(1 \leq i \leq i-2)$ the tree displayed by C choosing the reticulation arc from leaf x_i, and so containing the cherry $\widehat{x_ix_n}$, and T_{n-1} the tree that chooses no reticulation arcs (the underlying tree $T_{n-1} = T_N$).

For each leaf x_i $(1 \leq i \leq n-2)$, let the distance from the parent tree vertex to the start of the reticulation arc be ℓ_i. Label each internal arc from left to right by $m_2, \ldots, m_{n-2}$, so that m_i is to the right of x_i.

We further note here that if C is a caterpillar network, then for any two weighted display trees $T_1^{w_1}, T_2^{w_2}$, if T_1 and T_2 are isomorphic as unweighted trees, they are isomorphic as weighted trees as well, because any weighted display tree of N is uniquely determined by the lowest HGT arc that is not deleted in its formation. In the following lemma, due to this fact we will denote the sum of probabilities assigned across all of the isomorphic weighted copies of T_i by $\beta_{\sum}(T_i)$, noting that this will be a function mapping $\mathcal{T}$ to $(0, 1)$.

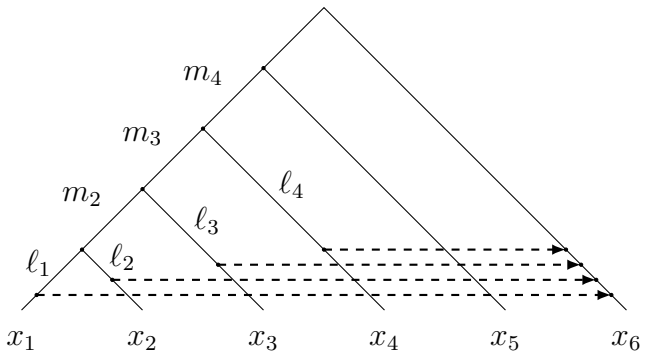


Figure 2.8: A caterpillar network on six leaves, labelled as required, except for reticulation probabilities.

In order to prove that these caterpillar networks are tree-metrizable, we will have to make use of two technical lemmas. The first is Lemma 2.3.2, which we recall is

used to reduce the problem of determining whether distances on a network obey the four-point condition to a condition on the sums of the length of the internal arcs of each quartet shape in its display trees.

The second is the next lemma, Lemma 2.5.3, in which we calculate the sums of the lengths of the internal arcs of each quartet shape in the display trees of a caterpillar network. To this end, we will denote the sum of the length of the internal arcs of each quartet shape $x_ax_b|x_cx_d$ in the display trees of the caterpillar network C by $int(C, x_ax_b|x_cx_d)$. That is, if we denote the set of weighted display trees of C that display $x_ax_b|x_cx_d$ by $\mathcal{T}|_{\{x_ax_b|x_cx_d\}}$, and the length of the internal arc of $T_i|_{\{x_a,x_b,x_c,x_d\}}$ by p_j,

$$int(C, x_ax_b|x_cx_d) = \sum_{T_i^w \in \mathcal{T}^w|_{\{x_ax_b|x_cx_n\}}} \beta_{\sum}(T_i)p_i.$$

We are now ready to state and prove the lemma.

Lemma 2.5.3. *Let C be a caterpillar network on $X = \{x_1, \ldots, x_n\}$, with edge-lengths ℓ_i and m_i as shown in Figure 2.8. Let $\beta_{\sum} : \mathcal{T} \to (0, 1)$ be the function that maps each tree $T_i \in \mathcal{T}_N$ to the total probability assigned to T_i (see Equation (2.1)). For some quartet $q = \{x_a, x_b, x_c, x_n\}$ where $a < b < c < n$, denote the length of the internal arc of $T_i|_{\{x_a,x_b,x_c,x_d\}}$ by p_i.*

$$\begin{aligned}
int(C, x_ax_b|x_cx_n) &= \beta_{\sum}(T_c)\ell_c + \sum_{s=b+1}^{c} \left(\sum_{t=s}^{n-1} \beta_{\sum}(T_t)m_{t-1} \right) \\
int(C, x_ax_c|x_bx_n) &= \beta_{\sum}(T_b)\ell_b \\
int(C, x_ax_n|x_bx_c) &= \beta_{\sum}(T_a)\ell_a + \sum_{s=a+1}^{b} \left(\sum_{t=1}^{s} \beta_{\sum}(T_t)m_t \right).
\end{aligned}$$

Proof. We note that T_i will display the quartet $x_ax_b|x_cx_n$ if $i > b$, $x_ax_c|x_bx_n$ if $i = b$, and $x_ax_n|x_bx_c$ if $i < b$, recalling that for a caterpillar network C, T_i is a specific tree in $\mathcal{T}_C$, and contains the cherry $\widehat{x_ix_n}$.

We now consider the internal arcs of each T_j.

If $j < b$, T_j will display $x_ax_n|x_bx_c$ and the internal arc will extend from the lowest stable ancestor of x_a and x_n, as depicted in Figure 2.9 (as a dotted line), to the parent vertex of x_b, so will have a length of $\ell_a + \sum_{k=2}^{b-1} m_k$ in the case that $j = a$, and $\sum_{k=max(a,j)}^{b-1} m_k$ otherwise.

If $j = b$, then T_j will display $x_bx_n|x_ax_c$, and will have an internal arc length of ℓ_b (the distance from the lowest stable ancestor of x_b and x_n to its parent vertex).

If $j > b$, then T_j will display $x_ax_b|x_cx_n$ and will have an internal arc extending from the parent vertex of x_b to the parent vertex of whichever of is further left out of x_c and x_n, so will have length of $\ell_c + \sum_{k=b}^{c-1} m_k$ in the case that $j = c$ and $\sum_{k=b}^{min(c-1,j-1)} m_k$ otherwise.

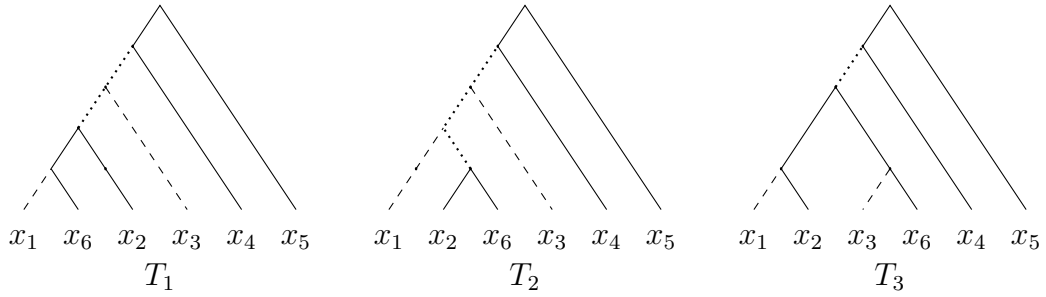


Figure 2.9: The display trees T_1, T_2, T_3 of the six-leaf caterpillar network in Figure 2.8. Here we have taken $q = \{2, 4, 5, 6\}$, indicated $T_i|_q$ with filled lines, the internal arc with dotted lines and edges not included in $T_i|_q$ with dashed lines. Note in particular that all three display $x_2x_6|x_4x_5$. In T_1, $j = 1 < a = 2$, in T_2, $j = 2 = a$, and in T_3, $j = 3 > a = 2$.

If we denote the sum of the contributions from each T_j that displays $x_ax_b|x_cx_n$ by $d_{x_ax_b|x_cx_n}$, it follows that $d_{x_ax_b|x_cx_n}$ will be the sum of the contributions from T_1 up to T_{b-1}, $d_{x_ax_c|x_bx_n}$ will be the contribution from T_b, and $d_{x_ax_n|x_bx_c}$ will be the sum of the contributions from T_{b+1} to T_{n-1}. Hence

$$\begin{aligned} int(C, x_ax_b|x_cx_n) &= \beta_{\sum}(T_c)\ell_c + \sum_{s=b+1}^{c}\left(\sum_{t=s}^{n-1}\beta_{\sum}(T_t)m_{t-1}\right) \\ int(C, x_ax_c|x_bx_n) &= \beta_{\sum}(T_b)\ell_b \\ int(C, x_ax_n|x_bx_c) &= \beta_{\sum}(T_a)\ell_a + \sum_{s=a+1}^{b}\left(\sum_{t=1}^{s}\beta_{\sum}(T_t)m_t\right). \end{aligned}$$

as required. □

Now that we have proven the technical lemma, the main theorem follows.

Theorem 2.5.4. *Let C be a caterpillar network. Then C is tree-metrizable on every tree it displays.*

Proof. For each leaf x_i ($i < n-1$), let the distance from the parent tree vertex to the start of the reticulation arc be ℓ_i, and label each internal arc from left to right by $a_2, \ldots, a_{n-2}$, so that m_i is to the right of x_i, as in Figure 2.8. Let $\beta_{\sum} : \mathcal{T} \to (0, 1)$ be the function that maps each tree T_i to the total probability assigned to T_i. As all weighted display trees of C that are isomorphic as unweighted trees are also isomorphic as weighted trees, this function can be defined unambiguously.

Recall that there are $n-1$ isomorphism classes of display trees of C, $\mathcal{T}_C = \{T_1, \ldots, T_{n-1}\}$, where T_i is the unique tree for which x_i has x_n as its closest neighbour for each $i \in \{1, \ldots, n-2\}$, plus the underlying tree T_{n-1}. We will show that C is T_i-metrizable for each i.

Consider a quartet $q = \{x_a, x_b, x_c, x_d\}$, supposing without loss of generality that $a < b < c < d$. We will now find the internal arc weights of the weighted display trees, with a view to invoking Lemma 2.3.2.

We first suppose that $n \neq a, b, c, d$. Any weighted display tree restricted to q is isomorphic to $T_C^{w(T_C)}|_q$, where $w(T_C)$ is the induced weighting on the underlying tree, and furthermore $T_i|_q$ is isomorphic as an unweighted tree to $T_C|_q$, because x_n is not involved in q and so none of the reticulation arcs are involved in either $T_C|_q$ or $T_i|_q$. Therefore the four-point condition will be obeyed for all such q, and $N|_q$ is tree-metrizable on $T_i|_q = T_C|_q$.

Define the functions

$$\gamma^+(j) = \frac{\beta_{\sum}(T_{j-1})\ell_{j-1} - \sum_{k=j+1}^{n-1} \beta_{\sum}(T_k)m_{k-1}}{\beta_{\sum}(T_j)}$$
$$\gamma^-(j) = \frac{\beta_{\sum}(T_{j+1})\ell_{j+1} - \sum_{k=1}^{j-1} \beta_{\sum}(T_k)m_k}{\beta_{\sum}(T_j)},$$

noting that the function γ^+ is undefined for $j = n-2$, and similarly γ^- is undefined for $j = 1$. Note that as $n > 3$ for a caterpillar network every j has at least one output between the two functions.

Fix all a_j to be some arbitrary non-zero lengths. Let $\ell_1, \ldots, \ell_{i-1}, \ell_{i+1}, \ldots, \ell_{n-2}$ be positive solutions to the following system of linear equations.

$$\ell_j = \begin{cases} \ell_{n-3}, & \text{if } j = n-2 \\ \gamma^+(j), & \text{if } i+1 < j \leq n-3, \\ \gamma^-(j), & \text{if } 2 \leq j < i-1, \\ \ell_2, & \text{if } j = 1, \end{cases} \tag{2.4}$$

Note that either ℓ_{i+1} or ℓ_{i-1} will not exist if $i = n-2$ or $i = 1$ respectively, but if both exist, we further require that

$$\beta_{\sum}(T_{i+1})\ell_{i+1} + \sum_{k=i+1}^{n-1} \beta_{\sum}(T_k)m_k = \beta_{\sum}(T_{i-1})\ell_{i-1} + \sum_{k=1}^{i-1} \beta_{\sum}(T_k)m_k. \tag{2.5}$$

It is a simple exercise in linear algebra that there exist strictly positive values of ℓ_j for all $j \neq i$ that satisfy these equations, as follows.

First, observe that due to equation (5), one of ℓ_{i+1} and ℓ_{i-1} can be written as a linear expression in terms of the other with positive coefficient and non-negative constant. Without loss of generality, suppose that $\ell_{i+1} = a_{i-1}\ell_{i-1} + b_{i-1}$ for some positive a_{i-1} and non-negative b_{i-1}. It follows from equations (4) that this means that for each ℓ_j, ℓ_{i+1} can be written as a linear expression in terms of ℓ_j with positive coefficient and constants, say $\ell_{i+1} = a_j\ell_j + b_j$. It follows from this and the fact that we have $n-3$ linear equations in $n-2$ variables that we can let ℓ_{i+1} be a free variable, and by setting it to be larger than any constant b_j we force all ℓ_j to be positive.

Finally, set ℓ_i to be any value larger than $\max\{\gamma^+(i), \gamma^-(i)\}$, where if either $\gamma^+(i)$ or $\gamma^-(i)$ are undefined we just require ℓ_i to be larger than the existing expression. We

claim that this causes the quartet x_a, x_b, x_c, x_n to satisfy the conditions of Lemma 2.3.2.

In particular, we claim that $C|_q$ displays $x_a x_b | x_c x_n$ if $i > b$, $x_a x_c | x_b x_n$ if $i = b$, and $x_a x_n | x_b x_c$ if $i < b$. To show this, we must show that the appropriate sum of internal arcs is larger than the other two, which must be equal as per Lemma 2.3.2.

First suppose $i > b$. Then to satisfy Lemma 2.3.2 we must show

$$\sum_{T_j \cong x_a x_b | x_c x_n} p_j \beta_{\sum}(T_j) > \sum_{T_j \cong x_a x_c | x_b x_n} p_j \beta_{\sum}(T_j) = \sum_{T_j \cong x_a x_n | x_b x_c} p_j \beta_{\sum}(T_j).$$

Therefore we require

$$d_{x_a x_b | x_c x_n} > d_{x_a x_c | x_b x_n} = d_{x_a x_n | x_b x_c}.$$

We first check the equality condition. From Lemma 2.5.3 we know

$$\begin{aligned}
int(C, x_a x_c | x_b x_n) &= \beta_{\sum}(T_b)\ell_b \\
&= \beta_{\sum}(T_{b-1})\ell_{b-1} + \sum_{k=1}^{b} \beta_{\sum}(T_k) m_k \\
&= \beta_{\sum}(T_{b-2})\ell_{b-2} + \sum_{k=1}^{b-1} \beta_{\sum}(T_k) m_k + \sum_{k=1}^{b} \beta_{\sum}(T_k) m_k \\
&= \dots \\
&= \beta_{\sum}(T_a)\ell_a + \sum_{s=a+1}^{b} \left(\sum_{t=1}^{s} \beta_{\sum}(T_t) m_t \right) \\
&= int(C, x_a x_n | x_b x_c)
\end{aligned}$$

We then check the inequality condition, which must be checked in two parts - for $i \geq c$ or $i < c$. First observe that where both exist, $\beta_{\sum}(T_j)\gamma^+(j) > \beta_{\sum}(T_{j+1})\ell_{j+1}$, and similarly that $\beta_{\sum}(T_j)\gamma^-(j) > \beta_{\sum}(T_{j-1})\ell_{j-1}$. Now, if $i \geq c$, from Lemma 2.5.3 we know

$$\begin{aligned}
int(C, x_a x_b | x_c x_n) &= \beta_{\sum}(T_c)\ell_c + \sum_{s=b+1}^{c} \left(\sum_{t=s}^{n-1} \beta_{\sum}(T_t) m_{t-1} \right) \\
&> \beta_{\sum}(T_c)\ell_c \\
&> \beta_{\sum}(T_b)\ell_b \\
&= int(C, x_a x_c | x_b x_n).
\end{aligned}$$

Otherwise, if $i < c$

$$\begin{aligned}
int(C, x_a x_b | x_c x_n) &= \beta_{\sum}(T_c)\ell_c + \sum_{s=b+1}^{c}\left(\sum_{t=s}^{n-1}\beta_{\sum}(T_t)m_{t-1}\right) \\
&= \beta_{\sum}(T_{c-1})\ell_{c-1} + \sum_{s=b+1}^{c-1}\left(\sum_{t=s}^{n-1}\beta_{\sum}(T_t)m_{t-1}\right) \\
&= \dots \\
&= \beta_{\sum}(T_{i+1})\ell_{i+1} + \sum_{s=b+1}^{i+1}\left(\sum_{t=s}^{n-1}\beta_{\sum}(T_t)m_{t-1}\right) \\
&= \beta_{\sum}(T_{i-1})\ell_{i-1} + \sum_{k=1}^{i-1}\beta_{\sum}(T_k)m_k \\
&> \beta_{\sum}(T_{i-1})\ell_{i-1} \\
&> \beta_{\sum}(T_b)\ell_b \\
&= int(C, x_a x_c | x_b x_n).
\end{aligned}$$

This proves the case where $i > b$. The cases where $i = b$ and $i < b$ are proved similarly. □

Corollary 2.5.5. *Let T be a tree of height $h > 2$. Then there exists a T-metrizable HGT network with underlying tree T and $h - 1$ non-trivial reticulations.*

Proof. Let T^{cat} be the caterpillar tree on $h+1$ leaves, with leaves $Y = \{y_1, \dots, y_{h+1}\}$. As T is of height h, there exists a graph embedding δ from T^{cat} into T, for instance by mapping the 'backbone' of the caterpillar to a path of length h in T.

Let T_i be the subtree of T induced by $\delta(y_i)$ for each $i \in \{1, \dots, h+1\}$. Now, by Theorem 2.5.4, there exists a HGT network N on Y with caterpillar underlying tree that is T^{cat}-metrizable (specifically the network with base tree T^{cat}). Note further that N has $h-1$ non-trivial arcs. By repeated application of Theorem 2.4.3, we can graft each of $T_1, \dots, T_{h+1}$ to each of $y_1, \dots, y_{h+1}$ respectively in N, and the resulting network will be T-metrizable. □

With Corollary 2.5.5 we have now shown that for every tree T of height at least 3 there exists a non-trivial T-metrizable HGT network. Additionally, with Theorem 2.5.4 we have found an infinite class of T-metrizable HGT networks where T is not the underlying tree. Together, these results extend the surprising result of Theorem 2.2.7 in two different directions. Finally, Theorem 2.4.3 shows us that we can form a tree-metrizable HGT network very easily by grafting trees onto leaves of a network. However, we have not yet considered the opposite case - grafting networks onto the leaves of a tree. We will address this in the next section.

2.6 Leaf-Grafts with Network Scions

The question of whether we can form a tree-metrizable HGT network by leaf-grafting a network onto a tree is more complicated. For instance, consider the HGT network N_1 shown in Figure 2.10. It is formed by leaf-grafting N, a slight modification of the network from [18], onto a 2-leaf binary tree. The network N is tree-metrizable by a combination of Theorem 2.2.7 and Lemma 2.3.4. However, despite the fact that N_1 is formed by grafting a tree-metrizable HGT network into a tree, N_1 is not itself tree-metrizable - restricting its display trees to $\{x_1, x_3, x_4, x_5\}$ only gives $x_1x_5|x_3x_4$ and $x_1x_4|x_3x_5$, which by Lemma 2.2.8 implies that N_1 is not tree-metrizable.

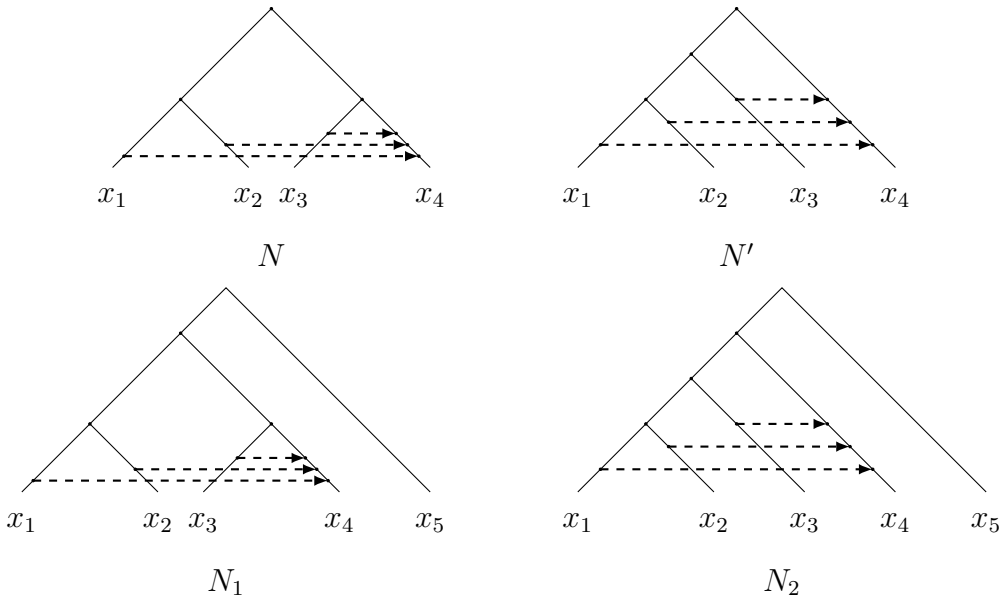


Figure 2.10: Two examples of a networks formed by leaf-grafting a tree-metrizable network onto a tree. The resulting network N_1 is not tree-metrizable, but the network N_2 is tree-metrizable.

However, if we take the HGT network N and just move the root to leaf x_4 to form N', and *then* graft it, we obtain the network in Figure 2.10 (ii), which is tree-metrizable by the observation that it is an example from Theorem 2.5.4 with a relocation of the root for which the pairwise distances for any weighting do not change (for details, see Theorem 2.6.7).

The remainder of the chapter has two aims. Firstly, we classify all possible level-2 HGT networks N for which the leaf-graft of N onto a tree T will be tree-metrizable. Subsequently, we will define a class of HGT networks for which the leaf-graft of a HGT network onto a tree T will always produce a tree-metrizable HGT network.

Definition 2.6.1. Let N be a HGT network, with underlying tree T_N. Suppose that some pair of edges a_1, a_2 of T_N are subdivided and a HGT arc placed between them in either direction. Then we say a_1, a_2 are *HGT-connected* and denote it $a_1 - a_2$.

As we will often be considering the incoming edges of the leaves, call such an edge a *leaf arc*, and if the leaf is denoted x_i, denote the leaf arc of x_i by e_i.

The following theorem greatly reduces the possibilities for tree-metrizable quartets and will serve as a useful tool for the following theorems.

Theorem 2.6.2. *Let N be a HGT network on $q = \{x_1, x_2, x_3, x_4\}$ with an underlying tree of the form $x_1x_2|x_3x_4$. Then N displays three isomorphism classes of trees if and only if there exists one leaf in q that is HGT-connected to both of the other non-adjacent leaves (e.g. $x_1 - x_3$ and $x_1 - x_4$).*

Proof. Let e_i denote the leaf arc of x_i in *the underlying tree*. First suppose there exists one leaf x_i in q so that e_i is HGT-connected to both of the other non-adjacent leaf arcs. Without loss of generality suppose N has two reticulation arcs a, b that connect the leaf e_1 to e_3 and e_1 to e_4 respectively. Observe that the display tree obtained by deleting all HGT arcs except for a has the cherry $\widehat{x_1x_3}$, the display tree obtained by deleting all HGT arcs except b has the cherry $\widehat{x_1x_4}$. Thus these display trees are non-isomorphic as unrooted trees, and neither is isomorphic to the underlying tree (which is always displayed as we can just delete all HGT arcs). Hence N displays three isomorphism classes.

Now suppose that N does not have one leaf in q that is HGT-connected to both of the other non-adjacent leaves. We shall consider each possible scenario.

Firstly, suppose there are at least two arcs that connect $\{e_1, e_2\}$ to $\{e_3, e_4\}$, but that no leaf arc is HGT-connected to both of the leaf arcs in the other set. That is, that $e_1 - e_3$ and $e_2 - e_4$ or $e_1 - e_4$ and $e_2 - e_3$. These cases are identical up to relabelling, so it suffices to consider $e_1 - e_3$, $e_2 - e_4$. It is then easy to observe that in this case we obtain only two display trees namely $x_1x_2|x_3x_4$ and $x_1x_3|x_2x_4$).

Now suppose that there is exactly one leaf arc in $\{e_1, e_2\}$ HGT-connected to a leaf arc in $\{e_3, e_4\}$. Without loss of generality, assume it is $e_1 - e_3$. Observe that N therefore cannot display $e_1e_4|e_2e_3$, as this requires $e_2 - e_3$ or $e_1 - e_4$, and so N does not display all three isomorphism classes.

Finally, suppose that no leaf arc in $\{e_1, e_2\}$ is HGT-connected to a leaf arc in $\{e_3, e_4\}$. Then all reticulation arcs are between adjacent arcs, so by Lemma 2.2.5, N only displays the underlying tree. □

In order to examine our graftings of networks into trees, we will need to isolate them from the rest of the tree. In order to do this we shall use the concept of a biconnected component.

Definition 2.6.3. A *biconnected component* of a HGT network N is a maximal HGT-connected subgraph B of N for which the removal of any arc of B is a HGT-connected graph.

Observe that if $V(B)$ is the collection of vertices contained in a biconnected component, then $LSA(V(B))$ is always contained in B. However, a biconnected component is never going to be a binary network: it cannot contain any leaves as the leaf vertices are degree-1, and so the removal of a leaf arc will result in a disconnected graph. We therefore must find the smallest sub-network of our network that contains B, including all necessary information. For this purpose we will use the concept of induced networks [28, p. 143], but with a small modification.

Definition 2.6.4. Let B be a biconnected component of a HGT network N on X. Let $\{v_1, \ldots, v_k\}$ be the set of vertices in B that have smaller outdegree in B than in N. For each v_i add a vertex w_i and the edge (v_i, w_i). Finally, label all the resulting leaves by a unique descendent of v_i, so that all leaves have a distinct label. The resulting phylogenetic network is referred to as a *minimal support network of B in N*, denoted $N(B)$, and is unique up to rearrangement of the leaf labels.

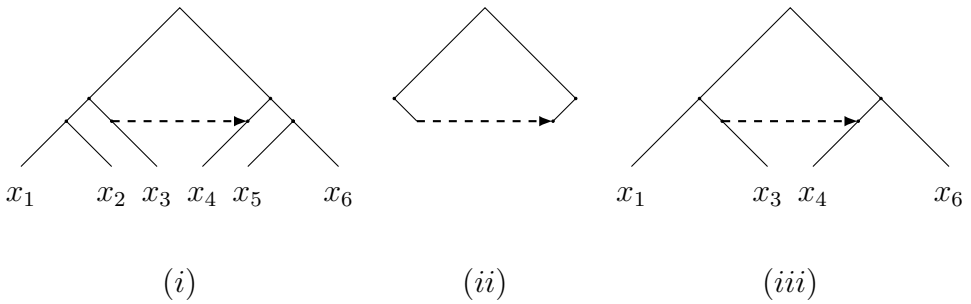


Figure 2.11: (i) a HGT network N with a single non-trivial biconnected component, B; (ii) The non-trivial biconnected component B of N; (iii) A minimal support network of B.

We note that minimal support networks are perhaps most easily understood as the rooted equivalent of B_N from [16]. We also note here that if a minimal support network of a biconnected component in a HGT network N is not tree-metrizable, it easily follows that N is not tree-metrizable.

Theorem 2.6.5. *Let N be a level-2 tree-metrizable HGT network, containing a biconnected component B. Then any minimal support network $N(B)$ in N has a caterpillar underlying tree unless either*

1. *$N(B)$ contains the root of N, or*
2. *all reticulation arcs in $N(B)$ are between adjacent arcs of $T_{N(B)}$.*

Proof. Suppose, seeking a contradiction, that the underlying tree $T_{N(B)}$ of $N(B)$ contains two cherries, since a tree is a caterpillar tree if and only if it has exactly one cherry. Additionally suppose $N(B)$ does not contain the root. First observe that if $N(B)$ contains exactly one reticulation arc, and it is not between adjacent arcs of $T_{N(B)}$, then $N(B)$ is not tree-metrizable [18, Theorem 5] and it follows that N is not tree-metrizable. Hence we can assume that there are two reticulation arcs in $N(B)$, and at least one of them is not between adjacent arcs of the underlying tree.

Let $q = x_1x_2|x_3x_4$ be a quartet in $N(B)$ such that q has two cherries of minimal height, that is, there is no leaf ℓ that separates x_1 from x_2, or x_3 from x_4. Denote the leaf edges of x_i *in the underlying tree of* $N(B)$, $T_{N(B)}$, by e_i. Then, by construction, both cherries of N_q contain the source or target of a reticulation arc, since $N(B)$ is a minimal support network and so the tree vertex parent of the underlying tree of

$N(B)$ must have at least one reticulation descendant. Without loss of generality, suppose e_1 and e_3 are the sources or targets of reticulation arcs.

We first assume that at least one of the HGT arcs starting or ending on e_1 and e_3 has the other end on an arc outside of $N|_q$. We will show that this means N is not tree-metrizable, by running through the cases.

Suppose without loss of generality, that e_1 is HGT-connected to an arc A outside of $N|_q$, and that A has some descendant ℓ.

1. *Either e_2 or e_4 are HGT-connected to e_3:* Suppose first that e_2 or e_4 is HGT-connected to e_3. If $e_2 - e_3$, $N(B)$ is not tree-metrizable by considering $N(B)|_{\{x_1,x_2,x_3,\ell\}}$ with Theorem 2.6.2. If $e_3 - e_4$, then as no other arcs connect to e_3 or e_4 and x_3 forms a cherry with x_4, this is a case of an arc connecting siblings immediately below a tree vertex, which can be omitted by Lemma 2.3.4. This leaves a single arc between non-adjacent arcs, which by Lemma 2.2.8 implies N is not tree-metrizable. We can therefore assume that e_3 is not HGT-connected to e_2 or e_4.

2. *Neither e_2 nor e_4 are HGT-connected to e_3 and the endpoints of the arcs HGT-connected to e_1 and e_3 are connected in N by a tree-path:* First, suppose the arc HGT-connected to e_1 is a tree-path descendant of the arc HGT-connected to e_3, or vice versa. Then we can select ℓ to be a descendant of both and consider the network induced by $\{x_1, x_2, x_3, \ell\}$. It is clear that there exist display trees with $x_1x_2|x_3\ell$ and $x_1x_3|x_2\ell$, but there are none with $x_1\ell|x_2x_3$, since x_1 and x_2 either form a cherry or x_2 and ℓ form a cherry. Thus N cannot be tree-metrizable unless e_1 and e_3 are HGT-connected to arcs that are not descended from one another, by Lemma 2.2.8.

3. *Neither e_2 nor e_4 are HGT-connected to e_3 and the endpoints of the arcs HGT-connected to e_1 and e_3 are not connected in N by a tree-path:* Let the arcs HGT-connected to e_1 and e_3 be denoted A and B respectively. As A and B are not descendants of one another, there exists ℓ and k that are each only descended from A and B respectively. If the underlying tree $T_{N(B)}$ of $N(B)$ restricted to $\{x_1, x_3, k, \ell\} = x_1x_3|k\ell$ or $x_1\ell|x_3k$, then $N(B)$ is not tree-metrizable by Lemma 2.6.2. However, if $T_{N(B)}$ restricted to $\{x_1, x_3, k, \ell\}$ is $x_1k|x_3\ell$, then we can consider the reticulation a between x_3 and ℓ. If a ends on x_3, we consider $q_1 = \{x_1, x_2, k, \ell\}$, and if a ends on ℓ, we consider $q_2 = \{x_1, x_2, x_3, k\}$. In both cases, the set of $N(B)$'s display trees restricted to q_i has exactly two non-isomorphic elements and so by Lemma 2.2.8, $N(B)$ — and thus N — is not tree-metrizable.

We can therefore assume all reticulation arcs in the biconnected component are between arcs of N_q, that is, that $N_B = N_q$.

If there are no arcs between the two cherries then we have the trivial case of all reticulation arcs in $N(B)$ being between adjacent arcs. Hence assume $e_1 - e_3$. By Theorem 2.6.2, N_q needs either $e_2 - e_3$ or $e_1 - e_4$ in order to display all 3

isomorphism classes and thus be tree-metrizable. By symmetry, it suffices to consider the $e_1 - e_3, e_2 - e_3$ case.

Now, as N_q does not contain the root, there exists a leaf ℓ attached to the central arc of N_q (omitting reticulation arcs), as in Figure 2.12.

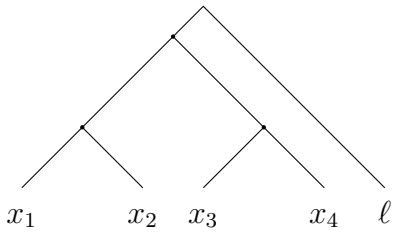


Figure 2.12: A leaf ℓ attached to the edge between two cherries.

One can quickly observe that the display trees obtained by this configuration display $x_1\ell|x_2x_4$ and $x_1x_2|x_4\ell$ but not $x_1x_4|x_2\ell$, so N is not tree-metrizable by Lemma 2.2.8.

As all possibilities are exhausted, the theorem follows. □

We will now define a class of networks $\mathcal{C}$ for which all leaf-grafts of $N \in \mathcal{C}$ onto a tree T are tree-metrizable.

Definition 2.6.6. Let N be a HGT network obtained by taking a caterpillar network C and adding a reticulation arc from leaf edge e_{n-1} to leaf edge e_n, ending above all of the others. Then N is termed an *enhanced caterpillar network.*

Theorem 2.6.7. *Let N be an enhanced caterpillar network. Let T be a tree. Then any network N' formed by leaf-grafting N onto T is tree-metrizable.*

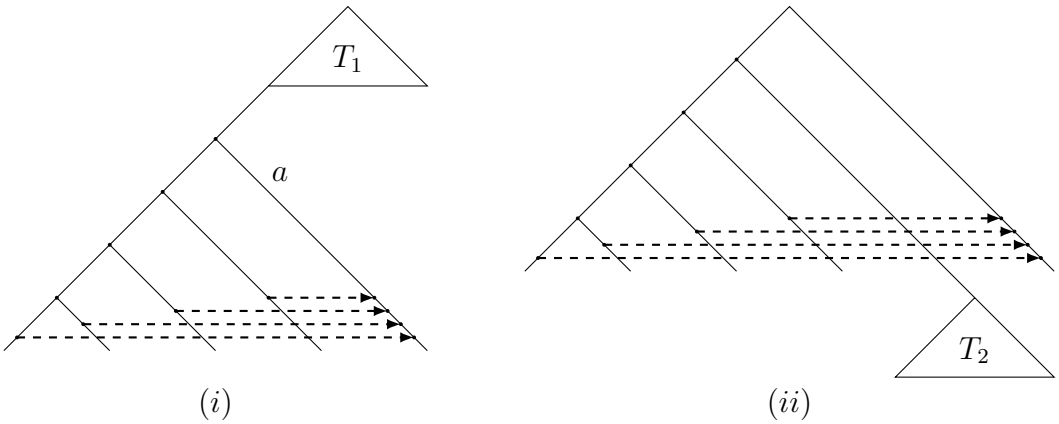


Figure 2.13: (i) A caterpillar network grafted to a tree to form the network N' in Theorem 2.6.7; (ii) The network N' from (i) modified by relocating the root to the arc a. In both diagrams a triangle indicates a tree structure.

Proof. Let N'' be the network obtained by relocating the root of N to the arc connecting the root of the enhanced caterpillar network to its reticulation descendant, marked a in Figure 2.13(i). Then N'' is of the form depicted in Figure 2.13(ii), and all pairwise distances are retained. This is because for any display tree $\mathcal{T}_i'$ of N', the corresponding display tree $\mathcal{T}_i$ of N is identical when considered as an unrooted tree. It follows that pairwise distances are retained for any pair of leaves x_i, x_j in N'.

With the new root location in N'' we have a tree-metrizable network — a caterpillar network — with a tree leaf-grafted onto it, so we can invoke Theorem 2.4.3 to observe that N'' is tree-metrizable. Since pairwise distances are retained in the transformation from N' to N'', it follows that N' is tree-metrizable too. □

Example 2.6.8. We now return to considering the examples from Figure 2.10. We can now see that the relevant difference between N_1 and N_2 is that when we perform the corresponding root relocations, N_1 does not become a tree-metrizable network, whereas N_2 does.

2.7 Discussion and Further Questions

In this chapter we have explored the observation from [18] that it is possible for a HGT network to carry a tree metric. This observation means that a tree metric may be consistent with a HGT network (or even many HGT networks), in addition to the unique tree specified by the four point condition (Theorem 2.2.1).

Specifically, we have asked which HGT networks could possibly carry a tree metric, and addressed this in two main directions. Firstly, we have shown in Section 2.4 that one can "grow" a tree-metrizable HGT network, using leaf-grafting, so that one may construct a tree-metrizable HGT network of any height or number of leaves by grafting a tree onto the leaf of an existing tree-metrizable HGT network. Secondly, we have shown that any tree metric at all of height h can be represented on a HGT network with $h-1$ non-trivial reticulations (Corollary 2.5.5). Furthermore, we have considered the grafting of HGT networks onto trees to obtain tree-metrizable networks, and described a class of HGT networks for which this is always possible (Section 2.6).

There are several possible avenues of research opened by these results. For instance, we showed that there exists tree-metrizable HGT networks on n leaves with $n-1$ non-trivial reticulation arcs. A reasonable question is whether even more complexity can be concealed: do there exist tree-metrizable HGT networks with more non-trivial arcs?

A second question for consideration is whether we can characterise exactly which tree-metrizable HGT networks can graft onto a tree? We have discovered one such class of structures (Theorem 2.6.7), but are there more? Such a classification would, hopefully, give us insight into what possible structures involving horizontal gene transfer can occur in recent history and still appear tree-like.

We finish the chapter with two conjectures regarding the mathematical structures behind tree-metrizability. Proof in either direction of these conjectures would

allow deeper insight into tree-metrizability and may lead to answering our previous questions.

Conjecture 2.7.1. *Let N be a level-k HGT network with a biconnected component B. Then the minimal support network $N(B)$ of B has at most $k-1$ cherries unless $N(B)$ strictly embeds in the top of N or all reticulation arcs in $N(B)$ are between adjacent arcs of $T_{N(B)}$.*

Conjecture 2.7.2. *Let $N(B)$ be a tree-metrizable minimal support network of a biconnected component B. Suppose $N(B)$ has n leaves. Then $N(B)$ has at least $n-2$ reticulation arcs.*

Chapter 3

Topological Similarity

3.1 Introduction

In the previous chapter we considered obstacles for phylogenetic reconstruction arising from metric similarity. Another potential issue for phylogenetic reconstruction involves topological similarity. Certain networks can present a strong tree-like signal due to their similarity to one or more trees. In these cases, it is possible to interpret a network-like evolutionary history as tree-like. Such networks are also of interest due to being, in some sense, fundamentally tree-like.

Specifically, while historically it has been standard to model evolutionary history using trees, recently, it has become common to represent histories by networks instead, due to recombination events and uncertainty. There is some discussion within the biological community about whether all such networks are still fundamentally 'tree-like' with some reticulation, or whether the histories they represent are not tree-like at all [8, 10, 11, 33]. For instance, prokaryotes and certain groups of eukaryotes undergo reticulate evolution in the form of horizontal gene transfer, in which genes are transferred from one species to another [2, 9]. Such evolutionary histories may be represented by a tree with some cross-connecting edges. In other scenarios, though, it may be impossible to consider a given history as meaningfully tree-like in any way. In such a scenario it would be better to model the history as a network without any reference to a tree-like structure.

The question of whether a network can be meaningfully said to be tree-like has recently lead to the introduction of the concept of *tree-based* networks by Francis and Steel [19], which, roughly speaking, are phylogenetic trees with additional edges placed between edges of the tree. In particular, the concept was introduced and applied to binary, rooted, phylogenetic networks. Under this definition, a reasonable claim can be made for a network to be quite similar to a tree.

More recently, the concept was extended to nonbinary rooted networks [30], and subsequently to binary unrooted networks [16] . Considering tree-basedness in the unrooted setting allows us to study the structure of a history in cases where there is uncertainty about the placement of the root. The nonbinary setting allows for histories in which there is uncertainty about order of speciation, as well as those in

which rapid speciation occurs - so-called soft and hard polytomies respectively.

In particular, in the nonbinary setting, it is possible to place an edge between an edge and a non-leaf vertex of the base tree (producing a vertex with at least degree 4), which was not possible in the binary case. The concepts of *strictly tree-based* networks and *tree-based networks* were therefore distinguished in the nonbinary setting, where in the former case edge-to-vertex edges are not permitted [30].

A possible issue with considering whether a given evolutionary history is tree-like, is that under the definition of a tree-based network it is possible for multiple non-isomorphic trees to be a base tree for a given network. In this circumstance, while a network may have a reasonable claim to be tree-like, a claim that it is like a *particular* tree is much harder to make.

The issue is particularly magnified by the possibility for every single tree embedded in a network to be a base tree. Semple showed that for binary rooted networks, the class of tree-based networks for which every embedded tree is a base tree (later termed *fully tree-based* networks by [16]) coincides with the familiar class of tree-child networks [42] introduced by Cardona et al [6]. Later, it was shown that in the binary unrooted setting, a network is fully tree-based if and only if it is a level-1 network [16].

This chapter extends the study of tree-based networks to the *non-binary, unrooted* setting, therefore including more complex structures as can be seen in Figure 3.1 (with leaf labels omitted). The networks in Figure 3.1, (see [37]), represent the uncertain regions of the evolutionary history of *Danthonioideae*, a Southern Hemisphere subfamily of grasses. These were generated by 70% bootstrap support consensus trees using Splitstree version 4.8 [27]. As Splitstree produces unrooted networks that are often nonbinary, it is expected that the results in this chapter will be of particular interest in analyses of Splitstree outputs.

The results in this chapter have been published by the author [22].

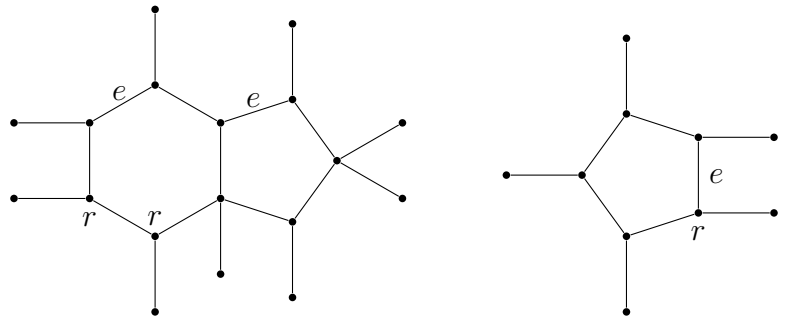


Figure 3.1: The non-trivial simple networks contained in Figure 3b) of a study by Pirie et al [37]. Leaf labels are omitted, edges labelled e denote possible edges to be deleted to find a strict base tree (defined later), and vertices labelled r are recombination sites proposed by the authors. Note that both networks are unrooted and the left one is nonbinary

Of particular interest to us is identification of those networks that can be considered tree-like (and to what degree), and how to identify them. To this end, in

Section 3.2, we define strictly tree-based and tree-based networks in the unrooted nonbinary setting, and introduce a third analogue, the *loosely tree-based* network. These are then characterised in terms of spanning trees of the network, generalising the characterisation by Francis et al [16]. In Section 3.3 we then characterise the nonbinary unrooted analogues of fully tree-based networks and provide some constructions for these networks of arbitrary level where this is possible. In Section 3.4 we end with some results on colourability of tree-based networks, which can assist in identifying networks that are not tree-based or not strictly tree-based.

3.2 Nonbinary Unrooted Tree-Based Networks

In this chapter we examine phylogenetic networks that are unrooted and are allowed to be nonbinary, so our first task is to make sensible definitions of tree-basedness for this new context.

Tree-based networks were initially introduced by Francis and Steel [19], in which they were constructed by subdividing some number of edges of a binary, rooted, phylogenetic tree, then adding additional edges between pairs of attachment points so that only one new edge is added to each attachment point.

In the nonbinary case, we have more freedom. In this case, we can additionally add edges between attachment points and the original vertices of the tree, or even between two vertices in the original tree, as we no longer need to worry about the resulting network being binary. Jetten and van Iersel consider this idea in the rooted, nonbinary setting [30] . If the new edges are exclusively between attachment points with no more than one edge at each attachment point, they termed the network *strictly tree-based*. If edges between attachment points and vertices of the original tree are allowed, they termed the network *tree-based*. We will formally define these in the unrooted nonbinary case shortly. In this chapter, we will additionally consider the possibility in which we allow edges between two vertices in the original tree and more than one additional edge incident to an attachment point.

Definition 3.2.1. Let N be a network. Then a *spanning tree* of N is a subgraph of N that contains every vertex of N and is a tree.

Francis et al. define a tree-based network in the binary, unrooted setting to be a network N on X which contains a spanning tree on the same leaf-set X [16]. If we consider a binary network N with some spanning tree T and some edge $e \in N - T$, then e can only be incident to degree-2 vertices of T. If e were incident to some degree-1 vertex of T then N and T would not have the same leaf set, and if e were incident to some vertex of degree 3 or more in T, then N would be strictly nonbinary, contradicting the fact that N is binary. Additionally, no pair of edges $e_1, e_2 \in N-T$ can be incident to the same vertex v of T, as this would force v to have degree 4 or more. Observe that, with the exception of the case where e is incident to a degree-1 vertex, the limitation was due to N being binary.

Therefore, if N is a nonbinary, unrooted network on X, then given some spanning tree T of N on X, some edge $e \in N - T$ may be between a pair of degree-2 vertices

of T, between a degree-2 vertex and a degree-3 or more vertex, or between a pair of degree-3 or more vertices. From the point of view of constructing N from a base tree T, this respectively coincides with adding an edge between attachment points, between an attachment point and an original vertex of the tree, or between two original vertices of the tree. We will refer to networks that satisfy this nonbinary analogue of the spanning tree definition introduced by Francis and Steel [16] by the term *loosely tree-based* networks.

As spanning trees are well-known and well-understood, we formally define loosely tree-based networks, as well as the nonbinary unrooted analogues of tree-based and strictly tree-based networks, in terms of spanning trees. We will then show that these definitions are equivalent to the attachment point definitions.

Definition 3.2.2. Let N be a network on leaf-set X. Then N is referred to as

1. *loosely tree-based* on X if there exists a spanning tree in N whose leaf-set is equal to X,

2. *tree-based* on X if there exists a spanning tree T in N whose leaf-set is equal to X such that T contains all edges between two vertices of degree 4 or more, and all degree-2 vertices of T have degree 3 in N, and

3. *strictly tree-based* on X if there exists a spanning tree T in N whose leaf-set is equal to X and T contains every edge incident to the vertices of degree at least 4.

In each case the spanning tree T is referred to as a *support tree* of N, and the tree obtained from T by suppressing degree-2 vertices is referred to as the *base tree* of N. If the spanning tree T meets the requirements for N to be loosely or strictly tree-based, then T may be referred to as a *loose* or *strict base tree* if it is not clear from context.

It follows that the class of strictly tree-based networks is contained in the class of tree-based networks, which are in turn contained in the class of loosely tree-based networks. The distinction between the three types of tree-basedness is shown by the networks in Figure 3.2. Note that in this figure, and all figures throughout this chapter, leaf labels are omitted.

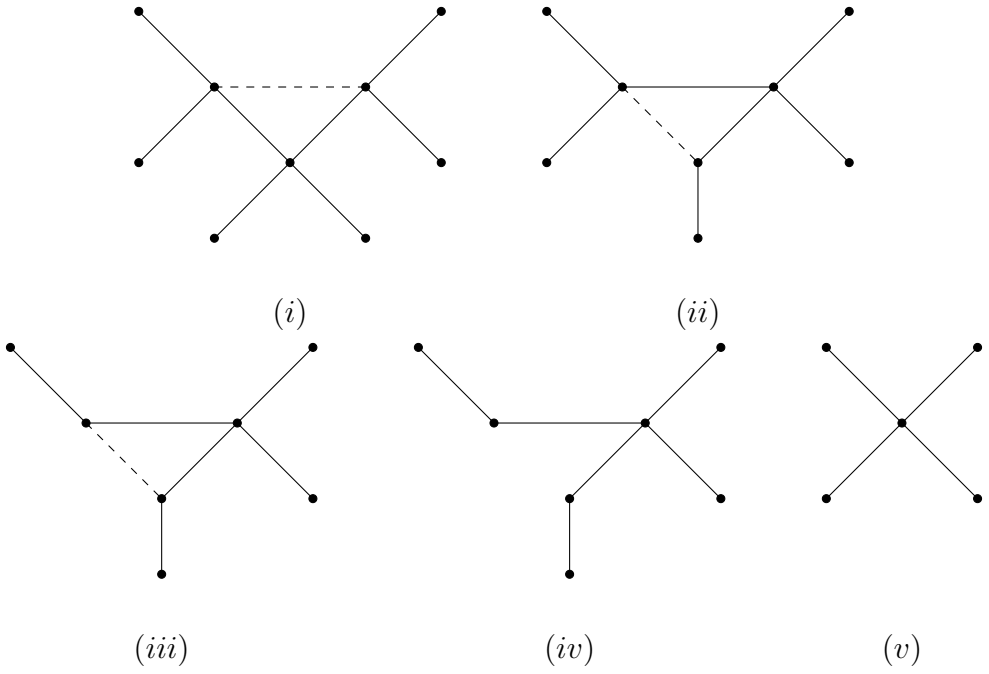


Figure 3.2: (i) A loosely tree-based network that is not tree-based; (ii) A network that is tree-based but not strictly tree-based; (iii) A strictly tree-based network; (iv) A support tree of the network in (iii); (v) A base tree of the network in (iii). The edges indicated by dashed lines in (i)-(iii) are possible edges that were added to a base tree to construct the network (and in (i) and (ii) are not the only possible edges).

We note that these definitions provide immediate access to some insights that may be tougher to see using a construction-based definition. For example, we can see that if a network contains a cycle consisting of vertices of degree 4 it cannot be tree-based, and if it contains a cycle that does not have two adjacent degree-3 vertices it cannot be strictly tree-based.

We will now show in Theorems 3.2.3 and 3.2.4 that our spanning tree definitions are equivalent to the familiar tree-based definitions from previous work on this topic.

Theorem 3.2.3. *Let N be a network on a leaf-set X. Then the following are equivalent:*

1. *N is tree-based.*
2. *N can be obtained by taking a tree T, subdividing edges of T to form attachment points and adding edges either between attachment points or between an attachment point and an original vertex of T (so that each attachment point now has degree 3).*

Proof. Suppose N was obtained by taking a tree T and performing the procedure outlined in the Theorem statement. Let T^+ be the tree T with the required attachment points added. Then T^+ is necessarily a spanning tree of N on X, as no step

in the procedure adds vertices or adds an edge to a leaf. As we add precisely one edge to each degree-2 vertex (that is, attachment point) of T^+, all vertices of T^+ of degree 2 have degree 3 in N. Furthermore, no edges are added between a pair of vertices of degree 3 or more in N, so T^+ contains all edges of N that lie between two vertices of degree 4 or more. Hence N is tree-based.

Now suppose N is tree-based, so there exists a spanning tree T of N with leaf-set X such that all degree-2 vertices in T were degree-3 vertices in N, and T contains all edges between two vertices of degree 4 or more. Denote the tree obtained by suppressing any degree-2 vertices of T by T^-. Then we can subdivide edges of T^- to make T, and add edges between the attachment points and either other attachment points or original vertices to make N (keeping each attachment point to degree 3). As all edges deleted from N to make T are incident to at least one degree-3 vertex, this means all required additional edges are incident to an attachment point. As all degree-2 vertices in T are degree-3 vertices in N it is not possible for two edges in $N - T$ to be incident to the same attachment point. It follows that T^- is the tree required by the Theorem. □

We now consider strictly tree-based networks.

Theorem 3.2.4. *Let N be a network on leaf-set X. Then the following are equivalent:*

1. *N is strictly tree-based.*
2. *N can be obtained by taking a tree T, subdividing edges of T to form attachment points, and adding edges between those attachment points (so that each attachment point has degree 3).*

Proof. Suppose N is strictly tree-based, so contains a spanning tree T that includes all edges incident to vertices of degree 4 or more. Then all edges in $N - T$ are between vertices of degree 3 in N, as all edges incident to vertices of degree 1 or degree 4 are in T, and there are no vertices of degree 2.

Let e be an edge in $N - T$. There cannot be two edges in $N - T$ incident to the same degree-3 vertex, as otherwise either T contains a leaf that is not a leaf of N or T does not span every vertex. Hence the deletion of e from N causes there to be two degree-2 vertices, which may be suppressed to obtain an edge. It follows that N is obtained by taking a tree, subdividing edges to form attachment points and adding edges between those attachments points (so that each attachment point has degree 3).

Now suppose N is obtained by taking some tree T, then subdividing edges of T and connecting those vertices obtained by subdivision. Let the subdivided subtree of N corresponding to T be denoted T^+. Then clearly T^+ is a spanning tree, as it contains every vertex in N. We then observe that every vertex of degree 4 or more is contained within the spanning tree, since the additional edges are only incident to vertices of degree 3. □

We now provide an example of a strictly nonbinary network that is not tree-based in Figure 3.3. In this case, tree-basedness is equivalent to the existence of a path from one leaf to the other that passes through every vertex - a Hamiltonian path. However, the graph in Figure 3.3 is a bipartite graph (as shown by the colouring of the vertices), and so any Hamiltonian path must alternate between the two sets of vertices. As there is one more black vertex than white, any path must therefore start and end on a black vertex, which is impossible as we must start and end on leaves. This example also demonstrates the distinction between containing a spanning tree *on the same leaf set* and merely having a spanning tree, as this example contains several spanning trees.

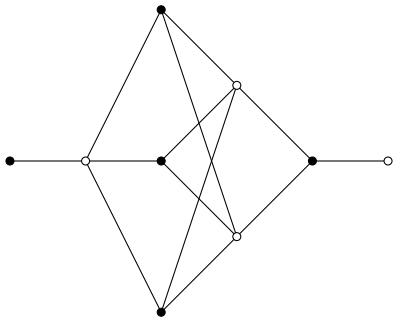


Figure 3.3: Example of an unrooted strictly nonbinary network that is not loosely tree-based. Colouring of the vertices demonstrates that the network is a bipartite graph.

Francis et al. prove a decomposition theorem for unrooted tree-based binary networks [16]. A similar one exists for nonbinary tree-based networks.

Recall the following standard definitions.

Definition 3.2.5. Let $N = (V, E)$ be a network on leaf set X. Let $N - e$ be the network $(V, E \backslash \{e\})$. A *cut-edge* is an edge such that $N - e$ is a disconnected graph. A *pendant* edge is a cut-edge where one of the connected components of $N - e$ is a single degree-0 vertex. We refer to N as *simple* if all cut-edges of N are pendant. A *blob* is a maximal connected subgraph of N with no cut-edges that is not a vertex.

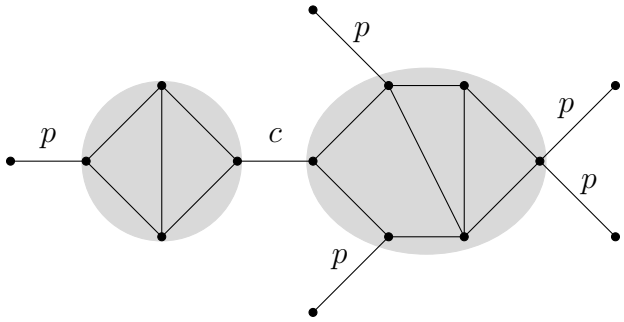


Figure 3.4: An example of an unrooted strictly nonbinary network with two blobs indicated in grey. Edges labelled p are pendant edges, and the edge labelled c is a cut-edge that is not pendant.

Given a network N and a blob B in N, we define a simple network B_N by taking the union of B and all cut-edges in N incident with B, where the leaf-set of B_N is just the set of end vertices of these cut-edges that are not already a vertex in B.

Definition 3.2.6. A phylogenetic network N is referred to as a *level-k* network if at most k edges have to be removed from each blob of N in order to obtain a tree.

Proposition 3.2.7. *Suppose N is a network. Then N is loosely tree-based, tree-based or strictly tree-based if and only if B_N is (respectively) loosely tree-based, tree-based or strictly tree-based for every blob B in N.*

Proof. This proof is extremely similar to Proposition 1 by [16]. We first note that if N is loosely tree-based, tree-based or strictly tree-based, then all cut-edges of N are contained in any spanning tree of N. It follows that any spanning tree T on N induces a spanning tree on each blob of N, and the spanning tree for each blob will inherit the loosely tree-based, tree-based or strictly tree-based properties from T.

Now suppose N is loosely tree-based, tree-based or strictly tree-based for every blob B in N. Then by taking a support tree for each blob in N we can clearly construct a loose support tree, support tree or strict support tree for N, and thus N is loosely tree-based, tree-based or strictly tree-based. □

Again, as in the paper by Francis et al. [16] we can immediately classify networks on a single leaf.

Remark 3.2.8. Suppose N is a network on $\{x\}$. Then N is tree-based if and only if N is precisely a degree-0 vertex labelled by x.

3.3 Fully Tree-Based Networks

A *fully tree-based* network N on leaf-set X is a network where every embedded tree with leaf-set X is a support tree, with the original concept introduced by Semple [42] and the terminology by Francis et al [16]. Of course, in the nonbinary setting we must be clear about what sort of base tree we obtain from the support tree - strict, normal or loose.

Definition 3.3.1. Let N be a network on leaf-set X. Then N is *loosefully, fully,* or *strictfully* tree-based if every embedded tree with leaf-set X is a base tree in the (respectively) loose, usual or strict sense.

In the binary case, a network is fully tree-based if and only if it is a level-1 network [16]. Correspondingly, we will show that in the nonbinary unrooted case, a simple network N is *strictfully* tree-based if and only if it is a binary level-1 network or a star tree. In general this means that a network N is strictfully tree-based if and only if for all blobs B of N, B_N is a binary level-1 network.

For example, let T be a strictly nonbinary tree such that there is a trio of vertices v_1, v_2, v_3 of degree 1 or 3 with edges $e_1 = (v_1, v_2)$ and $e_2 = (v_2, v_3)$. Then by adding

an attachment points each to e_1 and e_2 and connecting them with an edge to form a network N, we obtain a biconnected component B so that B_N is a binary level 1 network (even though N is not binary). It follows that N is strictfully tree-based as the only blob in N is binary and level-1. We will now find characterisations of simple strictfully, fully and loosefully tree-based networks, from which similar general results can be drawn.

We make the following fairly trivial but extremely useful remark.

Remark 3.3.2. Let N be a network on leaf-set X, possibly with degree-2 vertices. Suppose S is a connected subgraph of N with leaf-set X. Then S contains an embedded tree with leaf-set X, contained within the network obtained by the union of the shortest paths in S between leaves. Note that there may be multiple shortest paths between leaves, so we are locating an embedded tree *within* the resulting network.

We will now prove our statement regarding strictfully tree-based networks, which is the direct analogue of the binary result on fully tree-based networks in [16].

Theorem 3.3.3. *Let N be a simple network. Then N is strictfully tree-based if and only if N is a level-1 binary network or a star tree.*

Proof. Star trees are obviously strictfully tree-based, and any tree that is not a star tree is not simple. We therefore assume that N is a strictfully tree-based level-k network for $k > 0$. Suppose N is strictly nonbinary, so there exists some non-leaf vertex v of degree 4 or more. Then v has at least two incident non-pendant edges as N is simple. Label one of them e, and observe that $N - e$ is a connected graph with leaf-set X. By Remark 3.3.2, $N - e$ contains a subtree T on leaf-set X. As N is strictfully tree-based, T must be a spanning tree. But T is then a spanning tree that does not include all edges incident to vertices of degree 4 or more, so N is not strictfully tree-based.

Therefore all level > 0 simple strictfully tree-based networks are binary networks. In the binary case, the definition of strictfully tree-based coincides with the definition of fully tree-based, and Theorem 5 in [16] states that binary, phylogenetic networks are fully tree-based if and only if they are level-1. The theorem follows. □

Definition 3.3.4. Let N be a network and v a vertex of N. We say that v is a *cut-vertex* if deletion of v and all edges incident to v from N forms a disconnected graph. The connected components of this disconnected graph are referred to as the *cut-components* of v in N.

Note that all non-leaf vertices incident with a pendant edge are cut-vertices, but not all cut-vertices have incident pendant edges. For an example, see Figure 3.5 (ii), where the central black vertex is a cut-vertex but has no incident pendant edges.

Theorem 3.3.5. *Let N be a simple network on X. Then N is loosefully tree-based if and only if every non-leaf vertex is a cut-vertex.*

Proof. The level-0 case is trivial, so we assume N is level-k for $k \geq 1$.

Suppose all non-leaf vertices of N are cut-vertices. We claim that this implies that every cut-component contains a leaf. Seeking a contradiction, suppose a given cut-component C of some vertex contains no leaf. It is a classical result that every finite graph must contain at least two non-cut-vertices [40, Section 8.4.2], so C must contain a non-leaf non-cut-vertex, which is a contradiction.

Now for any given vertex v, each of its cut components must contain at least one leaf. Suppose a and b are leaves from two separate cut-components of v. Then any path from a to b must include v, and so any embedded tree on X must include v. It follows that any subtree of N on X must also be a spanning tree of N on X, since all vertices of N are cut-vertices. Hence N is loosefully tree-based.

Now suppose that N is loosefully tree-based. Seeking a contradiction, let v be a non-leaf vertex that is not a cut-vertex. Then $N - v$ is a connected network on leaf-set X and hence there exists a subtree of $N - v$ on leaf-set X. This implies that N contains a subtree T on leaf-set X that does not contain v, and hence T is not a spanning tree. It follows that N is not loosefully tree-based, a contradiction. Therefore all vertices of N are cut-vertices. □

We note that this makes it fairly trivial to find loosefully tree-based networks of level-k for any $k \geq 0$. It suffices to find a loosely tree-based level-k network N, then add leaves to each vertex in N that does not already have an incident pendant edge.

Finally, we classify fully tree-based networks.

Theorem 3.3.6. *Let N be a simple network on taxa X. Then N is fully tree-based if and only if*

1. *every non-leaf vertex in N either is degree 3 with an incident pendant edge or is a cut-vertex with at least 3 cut-components, and*
2. *every vertex of degree 4 or more in N is only adjacent to vertices of degree 3 or 1.*

Proof. Suppose the vertices obey the conditions outlined in the statement. Let T be an embedded tree in N with leaf set X. We again can see that T must be a spanning tree, as all non-leaf vertices of N are cut-vertices. Suppose v is a non-leaf vertex in N. We note that v must be adjacent in T to at least one vertex from each of its cut components, or otherwise T does not span N. Hence if v has at least 3 cut-components it must have degree at least 3 in T. Otherwise v must have degree 3 by the assumptions of the theorem. It follows that all vertices of degree 2 in T were degree 3 in N.

Finally, as all vertices of degree 4 or more are only adjacent to vertices of degree 3 or 1, all spanning trees contain all edges of N between two adjacent vertices of degree 4 or more (as there are none). It follows that N is fully tree-based.

Now suppose N is fully tree-based. We will show that both property 1 and 2 from the statement of the theorem must hold, by seeking a contradiction. First

suppose 2 does not hold, that is, let N contain a pair of adjacent vertices of degree 4 or more and denote their shared edge e. We see that $N - e$ must be a connected subgraph (or e would be a cut-edge), so by Remark 3.3.2 $N - e$ contains a subtree on leaf-set X. Thus N contains a subtree T on leaf-set X that does not include e, and thus N cannot be fully tree-based, a contradiction. We therefore assume all vertices of degree 4 or more in N are only adjacent to vertices of degree 3 or 1.

We now note that being fully tree-based is a stronger condition than being loose-fully tree-based, so by Theorem 3.3.5 every vertex of N must be a cut-vertex.

Suppose that v is a cut-vertex with precisely 2 cut-components. We claim that v must have an incident pendant edge, thus meeting the requirements in the statement of the theorem. Let the set of vertices adjacent to v in one cut-component be $A_1 = \{a_1, ..., a_s\}$ and let the vertices adjacent to v in the other be $A_2 = \{b_1, ..., b_t\}$. We can assume that either A_1 or A_2 contains more than one vertex, as v cannot have degree 2. Therefore suppose $|A_1| > 1$. As A_1 is in a single cut component, there must exist a path between a_1 and each of $a_2, ..., a_s$ that does not contain v. Similarly, if $|A_2| > 1$, there must exist a path between b_1 and each of $b_2, ..., b_t$ that does not contain v.

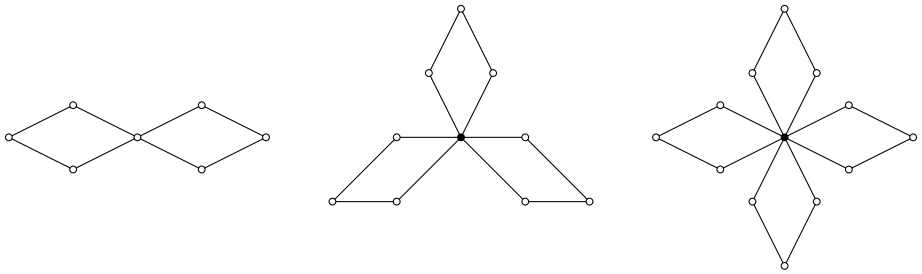
Denote the edge between v and a_i by e_i, and the edge between v and b_i by f_i. Additionally, let $C = \{e_2, ..., e_s, f_2, ..., f_t\}$, interpreted as just $\{e_2, ..., e_s\}$ if $|A_2| = 1$. Then $N - C$ is a connected subgraph with leaf-set X, so $N - C$ contains a subtree on leaf-set X. This means N contains a subtree T on leaf-set X that does not contain $e_2, ..., e_s, f_2, ..., f_t$, so v is degree 2 in T, which is in fact a spanning tree as N is fully tree-based.

We note that if $|A_i| > 1$ for both $i = 1, 2$, or if $|A_i| > 2$ for either $i = 1$ or $i = 2$, v has degree 4 or more. In either case we obtain a spanning tree for which v has degree 4 or more in N but 2 in the spanning tree, so N is not fully tree-based by definition. Hence $|A_1| = 2$ and $|A_2| = 1$, so if v is a cut-vertex with 2 cut-components, v must be degree 3 and have an incident pendant edge.

Hence N meets the conditions outlined in the statement. □

It is still rather easy to construct a level-k fully tree-based network for any $k \geq 0$. Figure 3.2 (iii) provides an example of a fully tree-based level-1 network (that is also strictly tree-based, but not strictfully tree-based).

Consider Figure 3.5, which illustrates examples for $k = 2, 3, 4$. Every white vertex in the figure indicates the vertex has an omitted pendant edge, and every black vertex does not. Then we can see that every cut-vertex without an incident pendant edge has at least 3 cut-components (in particular note that the central vertex in (i) has an omitted pendant edge), every other vertex is of degree 3 and no vertex of degree 4 or more is adjacent to another one. Constructing level-k fully tree-based graphs is a simple matter of adding additional 'diamond formations' around the central cut-vertices.



(i) (ii) (iii)

Figure 3.5: Level-2, 3, and 4 fully tree-based networks. In these diagrams every white vertex indicates the vertex has an omitted pendant edge, and every black vertex does not.

It is also worth noting that, for example, loosefully tree-based networks can also be strictly tree-based without being strictfully tree-based, as Example 3.3.7 illustrates. This is especially pertinent in light of the fact that Figure 3.2 (iii) is a strict, fully tree-based network but not a strictfully tree-based network. We further note that Figure 3.2 (ii) is an example of a loosefully tree-based network that is tree-based but not strictly tree-based.

Example 3.3.7. Figure 3.6 shows an example of a level-2 strictly tree-based network N that is loosefully tree-based. To see this, observe that every vertex has at least one incident pendant edge, so N is loosefully tree-based by Theorem 3.3.5. A strict support tree may be obtained by deleting edges A and B. We note that this example is not fully tree-based, as there exist spanning trees that can be obtained by deleting any two non-pendant edges incident to a single degree-5 vertex.

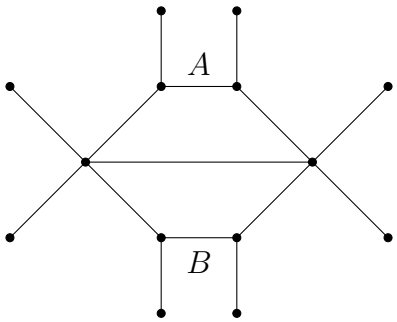


Figure 3.6: A level-2 strictly tree-based network that is loosefully tree-based.

3.4 Tree-Based Networks and Colourability

Let $\chi(G)$ denote the chromatic number of a graph G, that is, the minimum number of colours required to colour the vertices of G so that for each edge (v, w), the vertices v and w are coloured differently. In the binary unrooted case, all phylogenetic networks are easily shown to be 3-colourable as a consequence of Brooks' Theorem,

which states that all graphs with maximum degree d are d-colourable unless they are complete or an odd cycle [4]. However, as we are now examining graphs with no bound on the degree, we require more delicate reasoning to prove results on colourability for tree-based networks.

Theorem 3.4.1. *Let N be a network. If N is tree-based, then N is 4-colourable. However, there exist loosely tree-based networks with chromatic number at least k for any $k > 0$.*

Proof. Suppose N is tree-based. Then N is obtained by taking a tree T, adding attachment points to form T^+, then adding edges between pairs of attachments points or between an attachment point and a vertex that was in T. Now, recall that a tree has chromatic number 2. Colour the vertices of T^+ according to any valid 2-colouring, and suppose we have 4 colours available. Then insert edges between the attachment points to obtain N, noting that it may not result in a valid colouring. We then consider the colouring for the attachment points. Let the (only) new edge incident to some attachment point v_1 be e. There are two possibilities - either e was added between v_1 and a vertex of the base tree, or e was added between v_1 and another attachment point.

If e was attached between v_1 and a vertex of the base tree, then v_1 is adjacent to 3 other vertices, with up to two different colours. As there are four options for colours, there exists one that we can colour v_1 and any conflict with v_1 disappears.

Suppose e was attached between v_1 and another attachment point v_2. As v_1 is adjacent to at most two different colours, we can pick another colour and any conflict between v_1 and another vertex disappears. Similarly, v_2 is only adjacent to three vertices, so we can do the same thing

We repeat this process for each attachment point, noting that our new vertices may be adjacent to (at most) three colours now due to our previous recolouring. However, we are permitting four colours so may continue to colour our vertices approprimately. As this is true for every attachment point, we see that N is 4-colourable.

We now construct a loosely tree-based network with chromatic number at least k for some $k \geq 0$. The $k = 1, 2$ cases are trivially true, as we can just take a single vertex and a tree respectively. Otherwise, take any tree T and insert k attachment points. We can then insert an edge between each attachment point and each other attachment point to form a k-clique. It follows that the resulting loosely tree-based network N has $\chi(N) \geq k$. □

We can improve this bound for strictly tree-based networks, but require a technical Lemma first.

Lemma 3.4.2. *Let N be a simple loosely tree-based network that is not a tree, and let $P = \{v_1, ..., v_k\}$ be the set of non-leaf vertices with a pendant edge. Then there is no base tree that contains every non-pendant edge incident to a vertex in P. Moreover, if N is strictly tree-based there exists a degree-3 vertex v with one incident pendant edge and one incident edge that is not in the base tree.*

Proof. Suppose N is a simple loosely tree-based network that is not a tree. Observe that every non-leaf vertex in N must have at least 2 incident non-pendant edges, as N is simple.

Let $N - S$ be a network obtained by deleting some set S of leaves (and their incident edges) so that there is exactly one pendant edge incident to each vertex in P and thus precisely k leaves. Observe that $N-S$ is loosely tree-based with support tree $T - S$ if and only if N is loosely tree-based with support tree T.

Let T_S be the tree obtained by suppressing degree-2 vertices in $T - S$ - that is, a base tree. If T_S contains every non-pendant edge incident to the vertices in P, every vertex in P has degree at least 3 in T, as it must have at least 2 non-pendant incident edges and 1 incident pendant edge. If we consider just those edges in T incident to vertices in P, there are k pendant edges. There are $2k$ non-pendant edges, some of which may be double-counted. As each one can be counted no more than twice, there are at least $\frac{2k}{2} = k$ non-pendant edges incident to the vertices in P. Summing the pendant and non-pendant edges, we have $2k$ total edges incident to vertices in P. However, T_S is a phylogenetic tree on k leaves, and thus can contain at most $2k - 3$ edges, which is a contradiction. It follows that T_S is not a base tree of $N - S$, so T is not a support tree of N. Hence T cannot contain all non-pendant edges incident to all vertices in P.

Now, suppose N is strictly tree-based with strict base tree T. Then at least one vertex in P has smaller degree in T than it does in N, by the first half of the lemma. However, we know that T must contain all edges incident to vertices of degree 4 or more (by Definition 3.2.2), which means that P contains a degree-3 vertex, with degree 2 in the base tree. The lemma follows. □

We now demonstrate that strictly tree-based networks are 3-colourable. This proof contains an induction, and Lemma 3.4.2 is critical for the inductive step.

Theorem 3.4.3. *If N is a strictly tree-based network, then N is 3-colourable. Moreover there exist strictly tree-based networks of chromatic number 3.*

Proof. We first find a strictly tree-based network of chromatic number 3. Take the star tree with 3 leaves, then place attachment points on two of the edges and add an edge between them. The resulting network has chromatic number 3, as it contains a 3-clique and it is trivial to find a 3-colouring.

Suppose N is a strictly tree-based network. We observe that if a graph G is 3-colourable, the graph G' obtained from G by subdividing an edge $e = (v, w)$ to form $e_1 = (v, x), e_2 = (x, w)$ is also 3-colourable. This can be seen by applying the 3-colouring of G to the corresponding vertices of G', and then observing that v and w are two different colours. We can then set x to be the third colour.

Secondly, observe that it suffices to consider simple networks, as N will be 3-colourable if and only if every blob B of N is 3-colourable.

We now proceed by induction on the level of N. Suppose N is level-1. By definition, this means that we can take a base tree T of N, then subdivide two edges of T to form T^+, then add an edge between the two attachment points to form N.

Thus, suppose we have formed T^+. As T^+ is a tree, it is 2-colourable, so set a valid 2-colouring for T^+. We then add the edge between attachment points a and b. If a and b are different colours, T still has a valid 2-colouring. Otherwise, if a and b are the same colour, we can just set one of them to the third colour to obtain a valid 3-colouring.

Now suppose that N is a simple level-k network for $k \geq 2$ and that all level-$(k-1)$ strictly tree-based networks are 3-colourable. Then select some degree-3 vertex v together with a strict base tree T so that v has an incident pendant edge p and edge $e = (v, w)$ in N that is not in T, which we can do by Lemma 3.4.2. We then consider the network N^- obtained by deleting e and suppressing v and w (which are necessarily degree-3 vertices as N is strictly tree-based).

We can see that N^- is a level-$(k-1)$ network and is thus 3-colourable by the inductive assumption. If we subdivide the appropriate edges needed to obtain v and w again, then the graph is still 3-colourable by the observation early in this theorem. It follows that once we add e back in, the only conflict is potentially between the colouring of v and w. In particular, v is adjacent to three vertices, which may be three different colours. However, one vertex adjacent to v is a leaf, so we can set the leaf-vertex to be the same colour as w without generating any more conflicts. Now v is only adjacent to up to two different colours, so we can set v to be the third colour and N is 3-colourable. It follows that all level-k strictly tree-based networks are 3-colourable, so by induction, all strictly tree-based networks are 3-colourable. □

Using this, if we know some network has a subgraph H such that its chromatic number $\chi(H) > 3$ we can immediately say that it is not strictly tree-based. Furthermore, if $\chi(H) > 4$, it is not even tree-based.

3.5 Discussion and Further Questions

In the initial part of the chapter we extended the current forms of tree-basedness to the unrooted nonbinary setting and defined a new form of tree-basedness, inspired by the spanning tree definition given by Francis et al. [16]. These were then characterised in terms of spanning trees with particular properties. In the second section of the chapter, we extended the concept of fully tree-based networks to the unrooted nonbinary setting, characterising each of three possible interpretations. In the final section we proved some results on colourability of unrooted nonbinary networks.

The NeighborNet algorithm implemented in Splitstree is often used in papers where it is difficult to resolve speciation events, and produces an unrooted network. Many of these networks are also nonbinary, making them perfect candidates to consider tree-basedness. The method often produces networks containing a number of 4-cycles 'glued' together in grids, with each cycle referred to as a *box*.

Remark 3.5.1. Let N be a network containing a 3×3 grid of boxes. Then N is not tree-based.

This observation follows from the fact that the interior box is formed by a cycle

of 4 vertices of degree at least 4. As the base tree of a tree-based network must contain every edge between pairs of vertices of degree 4 or more, it must contain a cycle, which is a contradiction. Networks produced by the NeighborNet algorithm that contain 3×3 grids may be found in various studies [12, 31, 41].

Figure 3b) in [37] represents the evolutionary history of *Danthonioideae* as an unrooted, nonbinary network which contains two blobs, which we shall denote B and C. The simple networks B_N and C_N produced by B and C are depicted in Figure 3.1. Observe that these networks are both strictly tree-based, as we can delete the edges labelled e. However, the authors of the paper proposed that the vertices marked r are the recombination sites in the history of *Danthonioideae*, which produces base trees that are not strict. Further note that the left diagram is not loosefully tree-based, but the right one is strictfully tree-based. It follows that the whole network is strictly tree-based, but not loosefully tree-based.

As unrooted nonbinary tree-based networks have not yet been heavily studied, there are a number of interesting avenues for further research. For instance, given the wide variety of characterisations for tree-based networks in the binary and nonbinary rooted settings [17, 19, 30, 47, 38], the following natural question arises:

Question 3.5.2. Are there analogous characterisations for tree-based networks for the unrooted nonbinary case, especially computationally efficient ones?

Certainly some of these results cannot apply directly, as many rely on the antichain-to-leaf property or modifications thereof, which only makes sense in a rooted setting.

Additionally, several results shown by Francis et al. [16] may be worth considering in the new setting. A *proper* network is a network N for which every cut-edge splits the leaves of N. The nonbinary setting also allows for the possibility of cut-vertices that do not have pendant edges, so a suitable extension of the definition of proper would include the requirement that all cut-vertices with k cut-components partition the leaf set into k non-empty subsets. There exist proper level-1 networks (in this sense) that are not tree-based and thus not strictly tree-based. However, the author has yet to find an example of a proper loosely tree-based network (in this sense) of level less than 5 - one of level-5 is depicted in Figure 3.3. In the binary unrooted setting there are no networks that are not tree-based of level less than 5 [16].

Question 3.5.3. Do there exist networks of level less than 5 that are not loosely tree-based?

We note that subsequently to the publishing of the paper this chapter is based on, this question was answered in the affirmative by Fischer et al. [15] in an arXiV paper. At the date of writing this paper has not been formally published.

We showed in Theorem 3.4.1 that if N is a tree-based network, $\chi(N) \leq 4$, and in the subsequent Theorem that strictly tree-based networks are 3-colourable. So far the author has yet to find an example of a tree-based network with chromatic number 4.

Question 3.5.4. Are tree-based networks 3-colourable?

Finally, we note that as determining whether an unrooted *binary* network is tree-based is NP-complete, the problem is also NP-complete for determining whether an unrooted nonbinary network has a base tree of any sort, as the definitions coincide in the binary case. It is also not difficult to produce strictly nonbinary networks for which finding a base tree in the loose sense is equivalent to finding a Hamiltonian path, so the strictly nonbinary case is also NP-complete.

Chapter 4

Hierarchical Similarity

4.1 Introduction

While in previous chapters we have considered circumstances in which phylogenetic networks may be mistaken for trees (whether through metric similarity or a strong tree-like topological signal), we have not yet considered when a tree may be mistaken for another tree. We will consider this in a topological sense, as in particular trees with very similar structure may be difficult to distinguish experimentally. In such circumstances it is often useful to introduce a metric, allowing us to determine exactly which trees are similar and dissimilar according to some sense of similarity - and in this chapter we will primarily consider hierarchical similarity. This will allow comparisons of trees that have arisen under related processes, such as gene trees in the presence of incomplete lineage sorting.

Additionally, phylogenetic trees arise frequently in attempts to describe relations among species, and it is often necessary to be able to compare trees that represent different possible relations among the same set of taxa. For instance, in a study of the bamboo genus *Phyllostachys*, assigning a distance between phylogenetic trees was important for assessment of the consistency among the tree topologies inferred from different sampling of alleles [48].

Metrics are also used in a number of other areas in phylogenetics to measure dissimilarity between phylogenetic trees, such as the exploration of tree space, computation of consensus methods, and assessments of phylogenetic reconstruction. Although the earliest metric on rooted phylogenetic trees was discovered in 1981 — the Robinson-Foulds metric [39] — since the 1990's there has been a relative explosion of metrics on rooted trees, including split nodal [6], transposition [1], matching cluster [3], and a parsimony-based metric [35], as well as rNNI and rSPR distances (apparently first studied on rooted trees in [34] and [21] respectively).

A major downside of several easily computable metrics - including the most commonly used, Robinson-Foulds distance - is that the majority of distances between a random pair of trees are comparatively very large. That is, most trees are as far away from each other as possible, leading to a right skew in the distribution of distances between pairs of trees in tree space [44]. This is undesirable, as it translates

to a limited ability to meaningfully distinguish between trees.

Despite this, the Robinson-Foulds metric is well-represented in studies where a metric is required to distinguish between two or more groups of trees, with [7, 43, 48] all using the Robinson-Foulds distance. This is likely due to both the ease of calculating the Robinson-Foulds distance, as well as the fact that it outperforms many other metrics on real data under two criteria proposed by Kuhner and Yamato [32].

Additionally, metrics based on local operations such as Subtree Prune and Regraft (SPR) and Nearest Neighbour Interchange (NNI) — often used due to the ease of calculating the neighbourhood of a given tree — have the potentially undesirable property that trees in the neighbourhood of one another can have very different hierarchies.

In response, some new metrics have been introduced based on cluster similarity, and have been shown to have fewer of the aforementioned downsides of other metrics [3, 44]. In the present chapter, we introduce an alternative metric based on cluster similarity, with several potential benefits. The metric is based on a graded partial order, and we show that the grading can be used to estimate tree distances. It also relies on a natural local operation to move around in tree space, allowing for easy computation of the neighbourhood of a given tree — a particularly useful property in MCMC exploration of tree space. Finally, the trees have correspondingly much larger neighbourhoods than other local operation metrics, also useful for MCMC exploration [20]. Given the widespread use of MCMC to infer phylogenies, for instance in [36], these aspects are especially important to consider.

While calculating distances within the metric is non-trivial (we have not yet found a sub-exponential algorithm to do so), we provide an upper bound approximation that matches the true distance in the majority of cases in some experimental simulations. Furthermore, these simulations suggest that the upper bound for the metric does not have a skew (unlike the Robinson-Foulds distance), so it is hoped that this metric will also not be skewed.

As with previous cluster-similarity metrics, trees that are a short distance apart have similar hierarchies. Indeed, for any pair of trees of distance 1 apart, the symmetric difference of their hierarchies contains at most three clusters. The metric is based upon the concept of a hierarchy-preserving map, which, as the name suggests, relates trees that have similar hierarchies. The partial order and the hierarchy-preserving maps may also be of independent interest.

Specifically, we anticipate that this new metric will outperform Robinson-Foulds metrics in discrimination between sets of trees, especially on real data as computational experiments have shown the present metric to remain successful at discrimination in the specific case of bifurcating trees. Additionally, it should increase accuracy of phylogenetic reconstruction using Markov Chain Monte Carlo methods. Finally, as the upper bound approximation is easy to calculate and relatively accurate, it will ameliorate computation speed concerns as well.

In Section 4.2 we introduce the notion of a hierarchy-preserving map between trees, and show that there is a unique maximal hierarchy-preserving map between

any pair of trees for which a hierarchy-preserving map exists. We then show that hierarchy-preserving maps induce a partial order on the set of rooted phylogenetic trees, and make some initial observations about the partial order, including that it generalises refinement. In Section 4.3 we introduce a metric based on the Hasse diagram of the partial order induced by hierarchy-preserving maps. In Section 4.4 we introduce an algorithm for calculating an upper bound on the metric, and present initial results on its properties. Finally, in Section 4.5 we present some computational findings from a program used to calculate the upper bound on the metric, with the program available at [23].

The results in this chapter have been published by the author [25].

4.2 Hierarchy-preserving maps

In this section we introduce *hierarchy-preserving* maps on the set of trees $RP(X)$. These are used to define a partial order on $RP(X)$.

Recall the following standard definitions in phylogenetics.

Definition 4.2.1. A *hierarchy* H on a set X is a collection of subsets of X with the following properties:

1. H contains both X and all singleton sets $\{x\}$ for $x \in X$.
2. If $H_1, H_2 \in H$, then $H_1 \cap H_2 = \varnothing$, $H_1 \subseteq H_2$ or $H_2 \subseteq H_1$.

Definition 4.2.2. Let $T \in RP(X)$ be a tree and v be a vertex of T. Then the *cluster* of T associated with v is the subset of X consisting of the descendants of v in T. If a cluster C is not X or a singleton, C is referred to as a *proper cluster*, and the set of proper clusters of T is denoted $P(T)$.

A collection of subsets of X is a hierarchy if and only if it is the set of clusters of some rooted phylogenetic tree T taken over all vertices of T (see [45] for instance). For this reason we refer to the set of clusters of T as the *hierarchy* of T, denoted $H(T)$.

Definition 4.2.3. Let $T, T' \in RP(X)$ with hierarchies $H(T)$ and $H(T')$. Then $\delta : H(T) \to H(T')$ is a *hierarchy-preserving* map if δ is the identity on singletons and the following properties hold for all $A, B \in H(T)$:

1. **Enveloping:** $A \subseteq \delta(A)$, and
2. **Subset-Preserving:** $A \subset B$ implies $\delta(A) \subset \delta(B)$.

There are several interesting properties that follow almost immediately from the definitions. It is easy to check, for instance, that the composition of two hierarchy-preserving maps is also a hierarchy-preserving map. Furthermore, a hierarchy-preserving map will always map X to X.

If $\delta : H(T) \to H(T')$ is a hierarchy-preserving map and there exists no hierarchy preserving map $\varphi : H(T) \to H(T')$ with $\varphi \neq \delta$ so that $\delta(A) \subseteq \varphi(A)$ for all $A \in H(T)$, then δ is termed *maximal* (with respect to T and T').

Example 4.2.4. Let $T, T' \in RP(X)$ where $X = \{a, b, c, d, e, f\}$ as depicted in Figure 4.1. Then $P(T) = \{ab, cd, abcd\}$ and $P(T') = \{abcd, abcde\}$. Then there exists a hierarchy-preserving map φ from $H(T)$ to $H(T')$ that is the identity on singletons and X, maps ab and cd to $abcd$ and maps $abcd$ to $abcde$. One can easily confirm the necessary properties hold, and that this is the unique hierarchy-preserving map from T to T'.

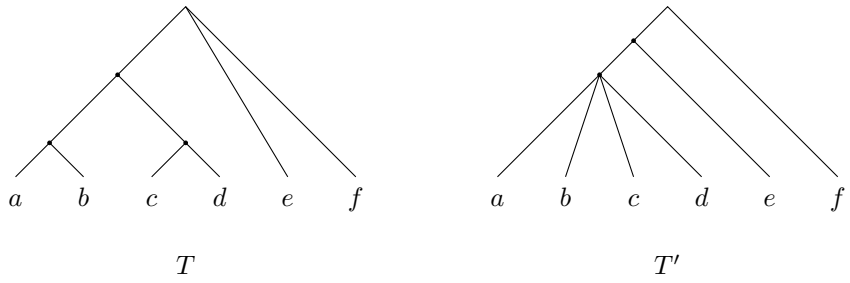


Figure 4.1: A pair of trees T and T' with a hierarchy-preserving map from $H(T)$ to $H(T')$ that maps ab and cd to $abcd$, and maps $abcd$ to $abcde$.

Theorem 4.2.5. *For $T, T' \in RP(X)$, if there exists a hierarchy-preserving map from T to T' then there is a unique maximal hierarchy-preserving map from T to T'.*

Proof. Suppose that $\delta_1 : H(T) \to H(T')$ and $\delta_2 : H(T) \to H(T')$ are distinct maximal hierarchy preserving maps. As they are distinct, there must be a cluster B of $H(T)$ such that δ_1 and δ_2 disagree. In particular, since at the very least $\delta_1(X) = \delta_2(X) = X$, there must be some non-singleton cluster B so that δ_1 and δ_2 disagree, but δ_1 and δ_2 agree on all clusters containing B. Denote the inclusion-minimal cluster containing B in $H(T)$ by C. Now, as δ_1, δ_2 are enveloping, both $\delta_1(B)$ and $\delta_2(B)$ contain B. Therefore either $\delta_1(B) \subset \delta_2(B)$ or vice versa. Assume without loss of generality that $\delta_1(B) \subset \delta_2(B)$. Define $\delta_1' : H(T) \to H(T')$ as follows:

$$\delta_1'(M) = \begin{cases} \delta_1(M), & \text{if } M \neq B \\ \delta_2(B), & \text{if } M = B. \end{cases}$$

We will show that this is a hierarchy-preserving map, which contradicts the maximality of δ_1. It follows that there is a unique maximal hierarchy-preserving map.

We can immediately see that δ_1' is certainly enveloping, as for $M \neq B$ we can use the fact that δ_1 is enveloping, and for $M = B$ we can use that δ_2 is enveloping.

We will now prove that δ_1' is subset-preserving. First suppose $M \subset B$. Then $\delta_1'(M) = \delta_1(M) \subset \delta_1(B) \subseteq \delta_2(B) = \delta_1'(B)$, by definition of δ_1' and the fact that δ_1 is

subset-preserving. Now, suppose that $B \subset M$. As B is inclusion-maximal in C, this means that $M \supseteq C$, and we know that $\delta_1'(B) = \delta_2(B) \subset \delta_2(C) = \delta_1(C) \subseteq \delta_1(M)$ by definition of δ_1' and the fact that δ_1 is subset-preserving again. Hence δ_1' is subset-preserving.

Thus we have found a hierarchy preserving map $\delta_1' : H(T) \to H(T')$ with $\delta_1' \neq \delta_1$ for which $\delta_1(A) \subseteq \delta_1'(A)$ for all $A \in H(T)$, contradicting the maximality of δ_1. It follows that there is a unique maximal hierarchy preserving map from T to T'. □

We now use the hierarchy-preserving maps just introduced, to define a partial order $\leq_{HP}$ on $RP(X)$. We say $T \leq_{HP} T'$ if there is a hierarchy-preserving map from $H(T)$ to $H(T')$. We will make use of the notion of a "maximal vertical subhierarchy", as defined below.

Definition 4.2.6. Let $T \in RP(X)$. Let C_1 be a cluster in $H(T)$, and suppose that $C_1, \ldots, C_k$ are distinct clusters in $H(T)$ with the property that $C_1 \subset \cdots \subset C_k = X$ and there are no other clusters D for which $C_i \subset D \subset C_{i+1}$. Then we say $\{C_1, \ldots, C_k\}$ is a *maximal vertical subhierarchy* of C_1 in $H(T)$.

Theorem 4.2.7. *The set $RP(X)$ forms a poset under the relation $\leq_{HP}$.*

Proof. The observation that the identity map from the hierarchy of a tree to itself is a hierarchy-preserving map gives reflexivity, and the transitivity of hierarchy-preserving maps is also easy to check. It remains to show antisymmetry.

Suppose $T \leq_{HP} T'$ and $T' \leq_{HP} T$. Then there exist hierarchy-preserving maps $\varphi_1 : H(T) \to H(T')$ and $\varphi_2 : H(T') \to H(T)$. We claim that both must be the identity mapping.

Suppose, seeking a contradiction, that φ_1 is not an identity mapping. Then there must be some cluster $C_1 \in H(T)$ so that $\varphi_1(C_1) = D_1 \neq C_1$, and as $\varphi_1(X) = X$, we can choose C_1 such that all clusters containing C_1 are mapped to themselves under φ_1 - that is, φ_1 acts as the identity on all elements of the maximal vertical subhierarchy $C_1, \ldots, C_k$ of C_1 except C_1 itself. In particular this implies that $C_2, \ldots, C_k$ are all clusters of T' as well.

We first show that φ_2 is the identity on $C_2, \ldots, C_k$. Let C_i be the inclusion-maximal element in this maximal vertical subhierarchy for which $\varphi_2(C_i) \neq C_i$. As φ_2 is subset-preserving, $\varphi_2(C_i)$ must be some inclusion-maximal subcluster of C_{i+1}, and as φ_2 is enveloping $C_i \subseteq \varphi_2(C_i)$. But $C_1, \ldots, C_k$ is a maximal vertical subhierarchy of T and so this means $\varphi_2(C_i) = C_i$, a contradiction. Therefore φ_2 is the identity on $C_2, \ldots, C_k$.

We now finally consider $\varphi_2(D_1)$. As $\varphi_1(C_1) = D_1$ and φ_1 is enveloping, $C_1 \subset D_1 \subset \varphi_2(D_1)$. Therefore $\varphi_2(D_1)$ must be an element of the maximal vertical subhierarchy of C_1, which by subset-preservation and the fact that φ_2 is the identity on $C_2, \ldots, C_k$ forces $\varphi_2(D_1) = C_1$. But then we get that $C_1 \subseteq D_1 \subseteq \varphi_2(D_1) = C_1$ and hence $C_1 = D_1$, contradicting the assumption that $\varphi_1(C_1) = D_1 \neq C_1$. It follows that φ_1 is the identity mapping and so $T = T'$, giving antisymmetry, and completing the proof. □

For several results in the remainder of this section, we will show given two trees $T \leq_{HP} T'$, how to construct a tree T'', so that $T \leq_{HP} T'' \leq_{HP} T'$. The tree we construct will be a "binding" of T.

Definition 4.2.8. Let $T \in RP(X)$, and let $A_1, \ldots, A_m \in H(T)$ (with $m \geq 2$) be distinct inclusion-maximal subclusters of a cluster $D \in H(T)$ such that $\bigcup_{i=1}^{m} A_i \neq D$. Take $H(T)$, delete all A_i for which $|A_i| > 1$ from $H(T)$, and add $\bigcup_{i=1}^{m} A_i$, forming a new set of clusters,

$$\mathcal{H} := (H(T) \setminus \{A_i \colon |A_i| > 1\}) \cup \left\{ \bigcup_{i=1}^{m} A_i \right\}.$$

Then $\mathcal{H}$ is a hierarchy (see Lemma 4.2.10), and the corresponding tree is termed a *binding* of T at $\bigcup_{i=1}^{m} A_i$, and denoted $T^D_{\bigcup_{i=1}^{m} A_i}$. If a tree T' can be obtained from T by binding, then T is termed an *unbinding* of T'.

Example 4.2.9. Let $X = \{a, b, c, d, e, f, g, h\}$ and let $T \in RP(X)$ be such that $P(T) = \{ab, abc, de, abcdefg\}$. Let $A = abcde$, $B = abcdef$ and $D = abcdefg$. Then the binding of T at A, denoted T^D_A, is the tree on X corresponding to the hierarchy with proper clusters $ab, abcde, abcdefg$. The binding of T at B, denoted T^D_B, is the tree on X corresponding to the hierarchy with proper clusters $ab, abcdef, abcdefg$; specifically, note that we do not delete f as it is a singleton and the result would no longer be a hierarchy. These three trees can be seen in Figure 4.2.

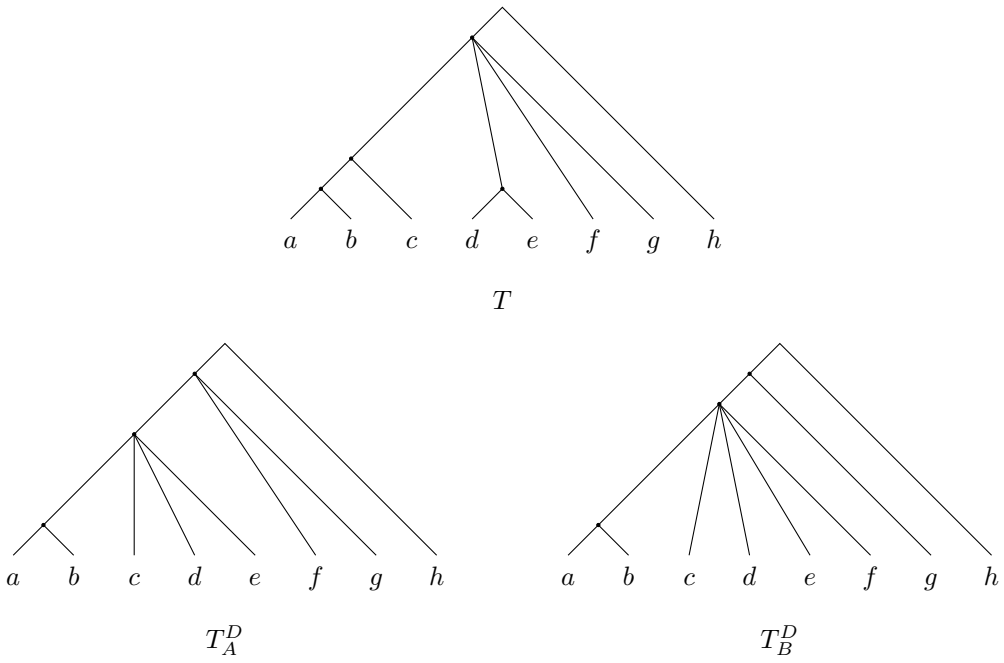


Figure 4.2: Two potential bindings of the tree T, as described in Example 4.2.9, with $A = abcde$, $B = abcdef$, and $D = abcdefg$.

Lemma 4.2.10. *Let $T \in RP(X)$, and suppose A, B are distinct inclusion-maximal subclusters of some cluster $D \in H(T)$, where $D \neq A \cup B$. Then the binding of T at $A \cup B$ is a hierarchy. Moreover, if $T^D_{A \cup B}$ is the corresponding tree, then $T <_{HP} T^D_{A \cup B}$.*

Proof. In a minor abuse of notation, let $H(T^D_{A\cup B})$ be the set of clusters corresponding to the binding of T at $A \cup B$. To confirm that $H(T^D_{A\cup B})$ is a hierarchy, it suffices to check that any $M \in H(T^D_{A\cup B})$ for which $M \cap (A \cup B) \neq \varnothing$ is either contained in or contains $A \cup B$.

If $M \cap (A \cup B)$ is non-empty, then $M \cap A$ or $M \cap B$ is non-empty. Hence, since M is a cluster in $H(T)$, and as A, B are inclusion-maximal in D, it follows that M either contains D (and so contains $A \cup B$), or is a subset of A or B (and thus is contained in $A \cup B$). Thus $H(T^D_{A\cup B})$ is a hierarchy.

The second statement of the lemma follows for two reasons. Firstly, as $A \cup B$ is certainly not a cluster in T we know that $T \neq T^D_{A\cup B}$. Secondly, because the map from $H(T)$ to $H(T^D_{A\cup B})$ that is the identity on all clusters except for A and B, which are mapped to $A \cup B$, is clearly hierarchy-preserving. □

Theorem 4.2.11. *Suppose $T \leq_{HP} T'$ and $\delta : H(T) \to H(T')$ is a maximal hierarchy preserving map. If A, B, C are three inclusion-maximal subclusters of some cluster D in $H(T)$, and $\delta(A) = \delta(B)$ contains $A \cup B \cup C$, then $T <_{HP} T^D_{A\cup B} < T'$.*

Proof. By Lemma 4.2.10 we know that the set of clusters $H(T^D_{A\cup B})$ is a hierarchy, and that $T <_{HP} T^D_{A\cup B}$. We can also see that $T^D_{A\cup B} \neq T'$ - if they were equal, the maximal hierarchy-preserving map δ from T to $T^D_{A\cup B} = T'$ must be the identity on all clusters of $H(T)$ except A and B, and map both A and B to $A \cup B$. But then $\delta(A)$ could not contain $A \cup B \cup C$, contradicting the assumptions of the theorem and showing $T^D_{A\cup B} \neq T'$.

It therefore suffices to show that there is a hierarchy-preserving map $\delta' : H(T^D_{A\cup B}) \to H(T')$. Noting that all clusters in $H(T^D_{A\cup B})$ other than $A \cup B$ are also clusters in $H(T)$, for any cluster $M \in H(T^D_{A\cup B})$, define

$$\delta'(M) = \begin{cases} \delta(M), & \text{if } M \neq A \cup B \\ \delta(A), & \text{if } M = A \cup B. \end{cases}$$

We claim that δ' is a hierarchy-preserving map from $H(T^D_{A\cup B})$ to $H(T')$ as required.

Certainly δ' is enveloping as δ is enveloping (so $M \subseteq \delta'(M)$ for all $M \neq A \cup B$), and $\delta(A \cup B) = \delta(A) \supseteq (A \cup B \cup C) \supset (A \cup B)$.

We now check subset preservation. For Y and Z clusters in $H(T^D_{A\cup B})$, we need to check $Y \subset Z$ implies $\delta'(Y) \subset \delta'(Z)$. If neither are equal to $A \cup B$, then this follows immediately from the definition of δ' and the properties of δ. It remains to check the two cases: (i) $Y = A \cup B \subset Z$, and (ii) $Y \subset A \cup B = Z$.

In the first case, $A \cup B \subset Z$ implies $D \subseteq Z$, because A and B are inclusion-maximal subclusters of D. Then $\delta'(A \cup B) = \delta(A)$ by definition of δ', and $\delta(A) \subset \delta(D) \subseteq \delta(Z)$ because δ is subset-preserving and B, D, Z are all clusters in $H(T)$. Finally noting that $\delta'(Z) = \delta(Z)$ completes this case.

In the second case, $Y \subset A \cup B$ implies $Y \subset A$ or $Y \subset B$ because Y, A, B are all part of a single hierarchy, $H(T)$. Assuming without loss of generality that $Y \subset A$, we have: $\delta'(Y) = \delta(Y) \subset \delta(A)$ by subset-preservation of δ; and $\delta(A) = \delta'(A \cup B)$ by definition of δ'. Therefore $\delta'(Y) \subset \delta'(A \cup B) = \delta'(Z)$, as required. □

We finish this section with a result describing the maximal elements under the partial order $\leq_{HP}$. Note that the $\leq_{HP}$-minimal element is the star tree.

Proposition 4.2.12. *The set of $\leq_{HP}$-maximal elements of $RP(X)$ is precisely $BRP(X)$, the set of binary trees.*

Proof. First, if a tree is non-binary, then its hierarchy has a cluster with at least three inclusion-maximal subclusters. Therefore, by Theorem 4.2.11, we can bind two of them to create a tree that is strictly greater in the partial order. So non-binary trees are not $\leq_{HP}$-maximal.

Second, if two trees T and T' are binary and there is a hierarchy-preserving map between them, they must be equal, as follows.

Let $\varphi : H(T) \to H(T')$ be a hierarchy-preserving map. Observe that φ maps X to X (by definition of a hierarchy-preserving map), and let Y be a non-singleton cluster of T such that for every cluster Z in the maximal vertical subhierarchy of Y, $\varphi(Z) = Z$. As T and T' are binary, Y has two inclusion-maximal subclusters in each of $H(T)$ and $H(T')$. Let C_1 and C_2 be the inclusion-maximal clusters of Y in $H(T)$, and D_1 and D_2 be the inclusion-maximal clusters of Y in $H(T')$. As φ is subset-preserving, C_1 and C_2 must each be mapped to some subcluster of D_1 and D_2. As φ is enveloping, this implies that each of C_1 and C_2 are subsets of D_1 or D_2. Additionally, $C_1 \cup C_2 = Y = D_1 \cup D_2$, which forces $C_1 = D_1$ and $C_2 = D_2$ or $C_1 = D_2$ and $C_2 = D_1$. It follows that φ is the identity on all elements of $H(T)$, so $T = T'$. □

We will often consider the partial order restricted to the set of trees below every element of a set of trees P.

If $P = \{T_1, \ldots, T_k\}$ is a set of trees, then the set of trees T for which there exists a hierarchy-preserving map $\delta_i : H(T) \to H(T_i)$ for each i is denoted by $HP(P)$. In other words,

$$HP(P) := \{T \in RP(X) \mid T \leq_{HP} T_i, \text{for all } T_i \in P\}.$$

Recall the following standard definition in phylogenetics

Definition 4.2.13. Let T, T' be rooted phylogenetic trees on the same set X. Then if every cluster of T is a cluster of T', T' is referred to as a *refinement* of T, denoted $T \preceq T'$.

In particular, observe that if T is the star tree S or T' is a refinement of T, then a hierarchy-preserving map from T to T' will always exist, namely the identity map on clusters in T. Therefore $HP(P)$ is always non-empty, as it will certainly contain S. We further note that if P consists of the single tree T, then $HP(P)$ can immediately be seen to be a bounded lattice, with least element S and greatest element T, as every element of $HP(P)$ has a hierarchy-preserving map into T by definition. It follows that if $P = (T, \ldots, T_k)$, then $HP(P)$ forms the poset obtained by taking the intersection of the bounded lattices corresponding to each tree in P.

In fact, as T' being a refinement of T implies there is a hierarchy-preserving map from T to T', the partial order $\leq_{HP}$ actually *refines* refinement. By this we mean that if $T \preceq T'$, then $T \leq_{HP} T'$, or equivalently, that edges in $RP(X)$ under the refinement partial order correspond to paths in $RP(X)$ under $\leq_{HP}$ that consist either entirely of up-moves or entirely of down-moves.

Proposition 4.2.14. *Let $T \preceq T'$ in $RP(X)$. Then $T \leq_{HP} T'$ in $RP(X)$.*

The converse of this proposition is not true, that is, the existence of a hierarchy-preserving map from T to T' does not imply that T' is a refinement of T. One can see this, for example, from either binding in Figure 4.2.

4.3 An induced metric on the set of rooted phylogenetic trees

The hierarchy-preserving maps, and associated partial order on the set of phylogenetic trees, allow us to define a new metric on the set of rooted phylogenetic trees. In this section we set out the metric, and prove some of its key properties, including information about the neighbourhood of a tree and the diameter of the space.

Let $\mathcal{H}(X)$ denote the Hasse diagram of $RP(X)$ under $\leq_{HP}$. That is, $\mathcal{H}(X)$ is the symmetric directed graph $(RP(X), E)$ where $(T_1, T_2) \in E$ if and only if for either $i = 1, j = 2$ or $i = 2, j = 1$, we have $T_i \leq_{HP} T_j$ and for any tree T_3 such that $T_i \leq_{HP} T_3 \leq_{HP} T_j$, either $T_3 = T_i$ or $T_3 = T_j$ (that is, T_j covers T_i under the $\leq_{HP}$ relation). We then define the distance $d_{HP}(T, T')$ to be the shortest distance from T to T' in $\mathcal{H}(X)$. We know that $\mathcal{H}(X)$ is connected as every tree has a path to the star tree, so d_{HP} is certainly a metric.

The following theorem shows that if two trees are distance one apart in $\mathcal{H}(X)$, then one is a binding of the other - in particular the binding of a pair of clusters in the hierarchy.

Theorem 4.3.1. *Let T, T' be trees. Then $d_{HP}(T, T') = 1$ iff $T' = T^V_{A \cup B}$, for some pair of distinct clusters A, B that are inclusion-maximal in V in $H(T)$, or the reverse.*

Proof. Suppose first that $d_{HP}(T, T') = 1$ and without loss of generality that $T \leq_{HP} T'$. Then T' covers T under $\leq_{HP}$, that is, for any tree T'' such that $T \leq_{HP} T'' \leq_{HP} T'$, either $T'' = T$ or $T'' = T'$. Let $\delta : H(T) \to H(T')$ be the maximal hierarchy-preserving map between them, as defined in Definition 4.2.3.

Now, let C be a cluster common to T and T' such that the maximal vertical subhierarchy of C is common to both trees, and contains X, but that the inclusion-maximal subclusters of C are different in T and T'. Such a cluster always exists since $C = X$ is possible. Denote the distinct inclusion-maximal subclusters of C in $H(T)$ by $A_1, \ldots, A_j$, and the distinct inclusion-maximal subclusters of C in $H(T')$ by $B_1, \ldots, B_k$.

The hierarchy-preserving map $\delta : H(T) \to H(T')$ acts as the identity on each element of the maximal vertical subhierarchy of C, for the following reasons. If δ is the identity on any cluster D, and if D' is a subcluster of D in both trees, then D' must map to a subcluster of D (by subset-preservation), that also contains D' (enveloping). This forces D' in T to map to D' in T'. Since δ acts as the identity on X, this forces it to act as the identity on the whole maximal vertical subhierarchy.

Considering the subclusters of C in T and T', this means that $\delta(A_h) = B_i$ for some unique B_i, and thus that $A_h \subseteq B_i$. Furthermore, each B_i must be the union of some subcollection of the A_h's.

Suppose there is some B_i that is the union of more than two A_h's. Then by Lemma 4.2.10 there exists a binding of two of those A_h's that produces a tree that also maps into T', contradicting the fact that $d_{HP}(T, T') = 1$. Hence each B_i is the union of at most two A_h's.

As $T \neq T'$, there must exist at least one such cluster, so suppose $B_j = A_k \cup A_\ell$. Now, suppose that there is any other cluster $A \in H(T)$ such that $\delta(A) \neq A$, or any cluster $B \in H(T')$ that is not the image of some cluster in $H(T)$. Then the binding $T_{A_k \cup A_\ell}$ is certainly different from both T and T', but we can see that $T <_{HP} T_{A_k \cup A_\ell} <_{HP} T'$, which is a contradiction as $d_{HP}(T, T') = 1$. It follows that the only difference between the hierarchies of T and T' is that T contains A_k and A_ℓ while T' contains B_j, and the result follows.

We now suppose, without loss of generality, that $T' = T^V_{A \cup B}$, for some pair of clusters A, B that are inclusion-maximal in V in $H(T)$. Then certainly $T \leq_{HP} T'$, so in order to show $d(T, T') = 1$ it only remains to show that T' covers T - that is, that if there is a tree T'' so that $T \leq_{HP} T'' \leq_{HP} T'$, then $T'' = T$ or $T'' = T'$.

Let T'' be a tree so that $T \leq_{HP} T'' \leq_{HP} T'$, and let $\varphi_1 : H(T) \to H(T'')$ and $\varphi_2 : H(T'') \to H(T')$ be hierarchy-preserving maps. By Theorem 4.2.5, there is a unique maximal hierarchy-preserving map $\varphi_{\max} : H(T) \to H(T')$, and this must certainly be the map that is the identity on all clusters of T except for A and B, which are mapped to $A \cup B$ in T'. The composition of two hierarchy-preserving maps is also a hierarchy-preserving map, so $\varphi_2 \circ \varphi_1$ is a hierarchy-preserving map too, and due to $\varphi_{\max}$ being maximal we have that $\varphi_2 \circ \varphi_1(A) \subseteq \varphi_{\max}(A)$ for all clusters A in T. Therefore $\varphi_2 \circ \varphi_1$ is the identity on all clusters of T except for A and B, and $\varphi_2 \circ \varphi_1(A) = \varphi_2 \circ \varphi_1(B) \subseteq A \cup B$. Furthermore, this implies that

$$H(T) \cap H(T') \cap H(T'') = H(T) \backslash \{A, B\} = H(T') \backslash \{A \cup B\}$$

and both φ_1 and φ_2 are the identity on this intersection.

There are two possibilities - either $\varphi_1(A) \cap \varphi_1(B) = \varnothing$ or not.

1. $\varphi_1(A) \cap \varphi_1(B) = \varnothing$*:* As φ_1 is enveloping, $B \subseteq \varphi_1(B)$ and therefore $\varphi_1(A) \cap B = \varnothing$. But $A \subseteq \varphi_1(A) \subseteq \varphi_2 \circ \varphi_1(B) \subseteq A \cup B$, so $\varphi_1(A) = A$. Similarly, $\varphi_1(B) = B$.

   Let M be some cluster of $H(T'')$. If $\varphi_2(M) \neq A \cup B$, then $\varphi_2(M) = C$ for some C in $H(T') \backslash \{A \cup B\} = H(T) \cap H(T') \cap H(T'')$. But φ_2 is the identity on all elements of this intersection, so $C \in H(T)$. On the other hand, if

$\varphi_2(M) = A \cup B$, as φ_2 is enveloping $M \cap A$ or $M \cap B$ is non-empty. Then as M and A are both clusters in the same hierarchy $H(T'')$, so M contains or is contained in A or B. But if M strictly contains or is strictly contained in A or B, then M could not map to $A \cup B$ as $\varphi_2(A) = \varphi_2(B) = A \cup B$ and this would contradict subset-preservation. This leads us to conclude that $M = A$ or $M = B$, which are again in $H(T)$. Therefore every cluster in $H(T'')$ is in $H(T)$, so as $T \leq_{HP} T''$ this gives us $T'' = T$.

2. $\varphi_1(A) \cap \varphi_1(B) \neq \varnothing$: Without loss of generality we can assume $\varphi_1(A) \supseteq \varphi_1(B)$, then as φ_1 is enveloping $A \cup B \subseteq \varphi(A)$. Furthermore, as $\varphi_2 \circ \varphi_1(A) \subseteq A \cup B$ this forces $\varphi_1(A) = A \cup B$. As $H(T'')$ contains every cluster of $H(T')$ and $T'' \leq_{HP} T'$ it follows that $T'' = T'$.

As $T'' = T$ or $T'' = T'$, it follows T' covers T under $\leq_{HP}$ and hence that $d(T, T') = 1$. □

For the rest of this section we will focus on movement around the Hasse diagram of trees, $\mathcal{H}(X)$.

Definition 4.3.2. Let T, T' be trees in $RP(X)$, and $e = (T, T') \in E(\mathcal{H}(X))$. Then e is referred to as an *up-move* if $T \leq_{HP} T'$ and a *down-move* if $T' \leq_{HP} T$.

Note that by Theorem 4.3.1, an up-move takes one from a tree to a binding of two clusters of that tree (that are inclusion-maximal in some third cluster), and a down move does the reverse. See Figures 4.3 and 4.4 for some examples.

Let us now clearly elucidate what a down-move actually *does*. One can consider the up-move to be the deletion of some distinct pair of clusters $A, B \in H(T)$ that are inclusion-maximal in a third cluster C, with $A \cup B \subsetneq C$ (unless A or B are singletons in which case only non-singletons are deleted) and then the addition of $A \cup B$.

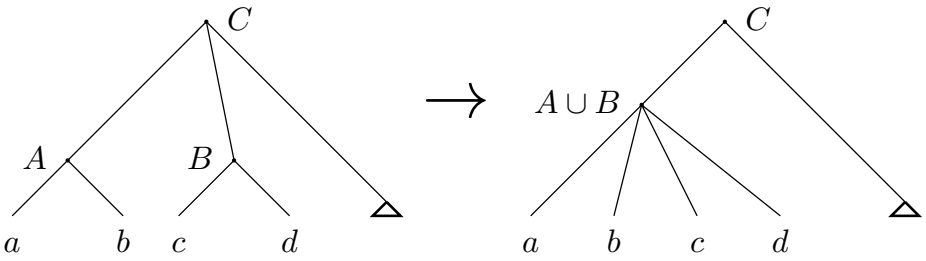
A down-move is therefore the reverse of this. To be precise, we select some cluster $Z \in H(T)$ with distinct inclusion-maximal clusters $Y_1, \ldots, Y_k$. We then partition these inclusion-maximal clusters into two, to form (after relabelling) $\bigcup_{i=1}^{j} Y_i$ and $\bigcup_{i=j+1}^{k} Y_i$, under the restriction that each union can only contain one element if that element is a singleton. Then, we add the clusters from $\{\bigcup_{i=1}^{j} Y_i, \bigcup_{i=j+1}^{k} Y_i\}$ that are not singletons, and delete Z.

For a tree T, recall that $P(T)$ is the set of proper clusters of the hierarchy corresponding to T, and let

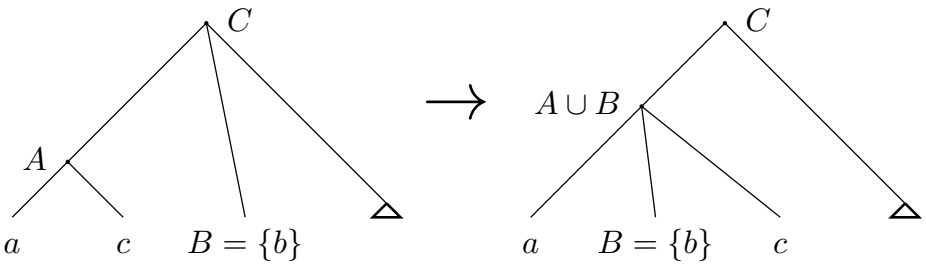
$$f(T) = \left(\sum_{A \in P(T)} |A| \right) - |P(T)| = \sum_{A \in P(T)} (|A| - 1),$$

noting that this number will always be non-negative, and will only be zero if T is the star tree, in which case $P(T) = \emptyset$.

We call $f(T)$ the *rank* of T. The rank provides an easy shortcut to calculating the distance between certain trees, if one is above the other in $\mathcal{H}(X)$:



(a) Up-move without singletons



(b) Up-move with a singleton

Figure 4.3: Examples of up-moves. The up-moves in (A) show one example without singleton clusters, and in (B) one in which one of the clusters is a singleton (it is also possible for both to be singletons). In all cases, a bold triangle indicates a non-singleton cluster.

Theorem 4.3.3. *If $T, T' \in RP(X)$, with $T \leq_{HP} T'$, then*

$$d_{HP}(T, T') = f(T') - f(T).$$

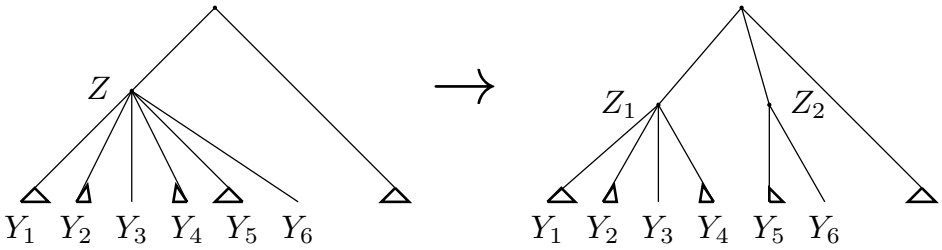
Proof. Recall that an up-move corresponds to taking the union of two distinct clusters A, B that are inclusion-maximal in some cluster C and deleting A if $|A| > 1$ and deleting B if $|B| > 1$.

Let $T, T' \in RP(X)$ and $\delta : H(T) \to H(T')$ a maximal hierarchy-preserving map between them. For $A \in H(T')$, let $\delta^{-1}(A)$ denote the set of clusters that map to A, and let $c_A := |\delta^{-1}(A)|$. We can see that for each cluster $A \in H(T')$ for which $c_A > 1$, we can bind the clusters in $\delta^{-1}(A)$ to form $\bigcup_{B \in \delta^{-1}(A)} B$, which will take $c_A - 1$ moves.

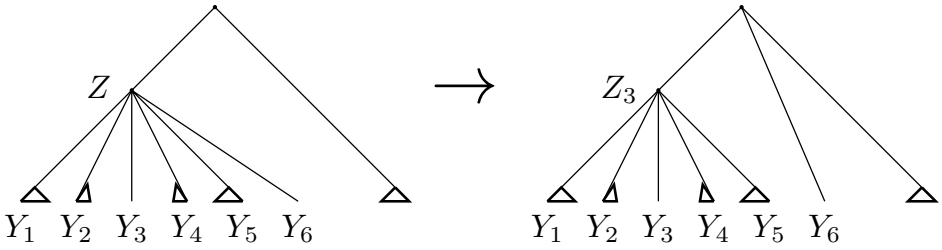
As δ is maximal, all elements of $\delta^{-1}(A)$ are inclusion-maximal in some cluster C. We will then need to bind each singleton element of $A \setminus \bigcup_{B \in \delta^{-1}(A)} B$ with B, which will take $|A| - \left|\bigcup_{B \in \delta^{-1}(A)} B\right|$ moves (which will again always form a tree due to maximality of δ)

It follows that it takes

$$\left(c_A + |A| - \left| \bigcup_{B \in \delta^{-1}(A)} B \right| - 1 \right)$$



(a) Down-move with unions of multiple clusters



(b) Down-move with a union and a singleton

Figure 4.4: Examples of down-moves. The down-moves in (A) show one example in which each union contains more than one cluster, and in (B) one in which one union is just a single cluster, in which case it must be a singleton (here Y_6). In all cases, a bold triangle indicates a non-singleton cluster.

moves to obtain A using this method.

If $c_A = 0$, then we can form a subcluster of size 2 of A, then add the remaining elements of A one at a time, which will require $|A| - 1$ moves. Observe that in this case $c_A = 0$ and $|\bigcup_{B \in \delta^{-1}(A)} B| = 0$. so

$$\left(c_A + |A| - |\bigcup_{B \in \delta^{-1}(A)} B| - 1 \right) = |A| - 1.$$

It follows that using this method (starting with inclusion-maximal proper clusters of $H(T)$ and working our way down, so that we will always have a valid tree), it will take

$$\sum_{A \in P(T')} \left(c_A + |A| - |\bigcup_{B \in \delta^{-1}(A)} B| - 1 \right)$$

$$= -|P(T')| + \sum_{A \in P(T')} \left(c_A + |A| - |\bigcup_{B \in \delta^{-1}(A)} B| \right)$$

$$= |P(T)| - |P(T')| + \sum_{A \in P(T')} \left(|A| - |\bigcup_{B \in \delta^{-1}(A)} B| \right)$$

$$= \left(\sum_{A \in P(T')} |A| \right) - \left(\sum_{A \in P(T)} |A| \right) + |P(T)| - |P(T')|$$

$$= f(T') - f(T).$$

Therefore, $d_{HP}(T, T') \leq f(T') - f(T)$.

We now observe that, by Theorem 4.3.1, each binding can only increase or decrease the rank by 1. Hence there is a lower bound on $d_{HP}(T, T')$ of the difference between their ranks, so $d_{HP}(T, T') = f(T') - f(T)$. □

Corollary 4.3.4. *If $T, T' \in RP(X)$, then*

$$|f(T) - f(T')| \leq d_{HP}(T, T') \leq f(T) + f(T').$$

Proof. That $|f(T) - f(T')| \leq d_{HP}(T, T')$ follows immediately from Theorem 4.3.3. To see that $d_{HP}(T, T') \leq f(T) + f(T')$, observe that one can always get from T to T' by taking a path of down-moves to the star tree S, then a path of up-moves to T'. Hence for any $T, T' \in RP(X)$, by Theorem 4.3.3 we have $d_{HP}(T, T') \leq f(T) + f(T') - 2f(S) = f(T) + f(T')$. □

We now derive some results on the diameter and neighbourhood of $RP(X)$ under d_{HP}.

Theorem 4.3.5. *If $|X| = n$ and $T \in RP(X)$, then $0 \leq f(T) \leq \frac{(n-1)(n-2)}{2}$, with bounds tight and every integer value achieved by some tree in $RP(X)$. Equivalently, if $|X| = n$, $\mathcal{H}(X)$ is a graded poset with rank function f and maximum rank $\frac{(n-1)(n-2)}{2}$.*

Proof. By Theorem 4.3.3 if $T <_{HP} T'$, then $f(T) < f(T')$. Also, by Theorem 4.3.1 the function f is compatible with the covering relation, so $\mathcal{H}(X)$ is a graded poset with rank function f.

Minimal f is achieved by the star tree S (as down-moves decrease f), which has $f(S) = 0$.

Elements with maximal f must be binary trees, because they are maximal in the poset and up-moves increase f. For all binary trees, $|H(T)| = 2n - 1$. We claim that caterpillar trees have maximal f, and we know for any caterpillar tree C, $f(C) = \frac{(n-1)(n-2)}{2}$. To see that caterpillar trees have maximal f, suppose you have some cluster C of size k that does not have a subcluster of size $k-1$. Observe that the 'contribution' to f of a cluster is strictly bounded above by the contribution of the cluster that contains it. There has to be at most two inclusion-maximal subclusters or we could make a binding, so call them B_1, B_2. Then the sum of the sizes of subclusters of B_1 has to be be at most $|B_1| - 1 + |B_2| - 1 \leq k - 3$. But we

could replace B_1 and B_2 by $B_1 \cup B_2$ plus one other element, without changing any of the structure below, and that has size $k-2$. The claim follows.

Hence the maximum value of $f(T) = \frac{n^2+3n-2}{2} - (2n-1) = \frac{(n-1)(n-2)}{2}$.

We can then observe that as we take any shortest undirected path from S to a caterpillar tree, the value of $f(T)$ increases by 1 each time. □

Corollary 4.3.6. *If $|X| = n$ and the diameter of $RP(X)$ under $\leq_{HP}$ is Δ_{HP}, then*

$$\frac{(n-1)(n-2)}{2} + 1 \leq \Delta_{HP} \leq (n-1)(n-2).$$

In particular, the diameter is $O(n^2)$.

Proof. As previously observed, one can always get from T to T' by a sequence of down-moves to the star tree, then up-moves to T'. Hence for any $T, T' \in RP(X)$, by Theorem 4.3.3 we have $d_{HP}(T, T') \leq f(T) + f(T') - 2f(S)$. It follows by Theorem 4.3.5 that $\Delta_{HP} \leq (n-1)(n-2)$.

We can also observe that for any caterpillar tree C with inclusion-maximal proper cluster $X \backslash \{a\}$, any tree T with a single proper cluster ab for some leaf b does not have a hierarchy-preserving map into C, and hence a shortest path from C to T must go from C to the star tree to T, for a distance of $d(C, T) = f(T) - f(S) + 1 = \frac{(n-1)(n-2)}{2} + 1$. Therefore $\frac{(n-1)(n-2)}{2} + 1 \leq \Delta_{HP}$ and the corollary holds.

Note that at least the upper bound can certainly be improved on, since no shortest path between a pair of binary trees with more than 3 leaves includes the star tree. □

The size of the up-neighbourhood and down-neighbourhood of a given tree varies with the structure of the tree. We now investigate the maximum sizes of these neighbourhoods.

Theorem 4.3.7. *Let $T \in RP(X)$, where $|X| = n$. Then the up-neighbourhood of T contains at most $\frac{n(n-1)}{2}$ trees, with this value achieved only by the star tree.*

Proof. We will show that deleting a proper cluster from $H(T)$ will increase the size of the up-neighbourhood of T. It follows that the tree with the largest up-neighbourhood is the star tree S, and we can observe that the up-neighbourhood of S consists of the trees with a single proper cluster which is size 2 - those obtained by binding any two leaves together. As there are n leaves, there are $\binom{n}{2} = \frac{n(n-1)}{2}$ in the up-neighbourhood of S.

We can now show that deleting a proper cluster from $H(T)$ will increase the size of the up-neighbourhood of T. Suppose that we have some hierarchy $H(T)$, with some cluster C. Let D be the cluster that C is inclusion-maximal in (with the possibility $D = X$). Suppose D has k inclusion-maximal subclusters and that C has j inclusion-maximal subclusters. Then, suppose that T has a total of x possible bindings that do not include the inclusion-maximal clusters of C or D. Suppose first that $k = 2$. Then the inclusion-maximal subclusters of D cannot bind (as they would

form a cluster already in $H(T)$), for a total of $x + \binom{j}{2}$ trees in the up-neighbourhood of T, or just x if $j = 2$. But if we delete C to form T', we now have $x + \binom{j+1}{2}$ trees in the up-neighbourhood (that is, all of the previous bindings plus all of the bindings involving the inclusion-maximal subclusters of C), and is larger since $j > 1$.

Now suppose $k > 2$. We can then immediately see that T has a total of $x + \binom{j}{2} + \binom{k}{2}$ possible bindings, or just $x + \binom{k}{2}$ if $j = 2$. However, once we have deleted C to form T', we have $x + \binom{j+k-1}{2}$ possible bindings, which is larger, as $j > 1$.

The result follows. □

Theorem 4.3.8. *Let $T \in RP(X)$, where $|X| = n$. Then the down-neighbourhood of T contains at most $2^{n-2} - 1$ trees, with this value achieved only by trees with a single proper cluster, and that cluster is of the form $X\backslash\{a\}$, for some leaf a.*

Proof. Suppose T has some proper cluster D with an inclusion-maximal proper subcluster C. Denote the inclusion-maximal subclusters of C by $C_1, \ldots, C_k$. Let x be the number of valid unbindings of clusters that are not C or D, y be the number of valid unbindings of D, and z the number of valid unbindings of C, so T has a total of $x + y + z$ unbindings - that is, a down-neighbourhood of size $x + y + z$. Now, if we remove C from $H(T)$, we claim that this increases the number of unbindings. This does not affect the number of valid unbindings of clusters that are not C and D, so there are x bindings of this type in $H(T)\backslash C$. Now, note that every valid unbinding of D in $H(T)$ is a valid unbinding in $H(T)\backslash C$, as if C is in a given partition, we can construct the same partition using the inclusion-maximal subclusters of C. Given that there is at least one partition here that we could not do before (deleting D and replacing it by C and $D\backslash C$), there are at least $y + 1$ possible unbindings of D. We can also identify the z unbindings of C with z unbindings of D in the following way. Suppose C is partitioned into A and B in $H(T)$. Then D partitioned into A and $B \cup (D\backslash C)$ is also a valid partition. It follows that there are at least $x + y + z + 1$ trees in the down-neighbourhood of $H(T)\backslash C$, so the number of unbindings has been increased.

We can therefore consider only the hierarchies in which no proper cluster has a proper subcluster, that is, no proper subclusters intersect. Supposing there are k proper clusters of size $i_1, \ldots, i_k$ where $i_j \geq 2$ for all j and $i_1 + \cdots + i_k \leq n$, the number of unbindings of such a tree will be

$$\sum_{j=1}^{k} \left\{ {i_j \atop 2} \right\}.$$

Observe in particular that for trees with a single proper cluster, and that cluster is of the form $X\backslash\{a\}$, this becomes $\left\{ {n-1 \atop 2} \right\}$, and it follows from basic properties of the Stirling numbers of the second kind that

$$\sum_{j=1}^{k} \left\{ {i_j \atop 2} \right\} \leq \left\{ {n-1 \atop 2} \right\}.$$

Hence trees of the form described have the largest possible number of unbindings, $\left\{ {n-1 \atop 2} \right\} = 2^{n-2} - 1$, and the result follows. □

Corollary 4.3.9. *The maximum neighbourhood size of a tree T (the sum of the up- and down-neighbourhoods of T) is $O(2^{n-2})$.*

4.4 An upper bound on d_{HP}

In this section we present an algorithm for calculating an upper bound e_{HP} on the distance $d_{HP}(T, T')$, because an exact calculation seems to be computationally expensive as the authors are yet to find an algorithm with subexponential runtime. We will also show that the upper bound is quite often equal to the true distance (despite not being a metric itself — see Observation 4.4.12). For instance, computational experiments show that $e_{HP} = d_{HP}$ in over 80% of cases of pairs of trees on nine leaves (Section 4.5).

The method to find the upper bound depends on finding $\leq_{HP}$-maximal trees that have a hierarchy-preserving map into both T and T', and then finding a minimum path between T and T' that goes through one of these. Of course, a geodesic path between T and T' need not visit any such $\leq_{HP}$-maximal tree, which is why this is only an upper bound.

Definition 4.4.1. Let T, T' be a pair of trees, and $\max_{\leq_{HP}}(T, T')$ be the set of trees T_i in $RP(X)$ that are $\leq_{HP}$-maximal subject to the condition that $T_i \leq_{HP} T$ and $T_i \leq_{HP} T'$. Then $e_{HP}(T, T')$ is defined to be $\min(f(T) + f(T') - 2f(T_i))$ across all trees in $\max_{\leq_{HP}}(T, T')$.

To find these, we will look at hierarchy-preserving maps in a different way, involving the following new definitions.

Definition 4.4.2. A *multi-hierarchy* $\mathcal{M}$ on a set X is a set of tuples (A, i) (referred to as *multi-clusters*) where $A \subseteq X$, and i is a positive integer, with the following properties:

1. $\mathcal{M}$ contains both the tuple $(X, 1)$ and all singleton tuples $(\{x\}, 1)$ for $x \in X$.
2. Let $(H_1, i), (H_2, j)$ be a pair of tuples in $\mathcal{M}$. Then $H_1 \cap H_2 = \varnothing$, $H_1 \subseteq H_2$ or $H_2 \subseteq H_1$.
3. The set of elements in $\mathcal{M}$ that share the same first entry A, say, $(A, i_1), \ldots, (A, i_k)$ are numbered sequentially from 1 to k in the second entry.

The set of multihierarchies on a set X will be denoted $MRP(X)$. If $(A, i), (B, j) \in MRP(X)$, we write $(A, i) \subseteq_M (B, j)$ if either $A \subset B$, or $A = B$ and $i \leq j$. In the latter case, if $i = j$, we write $(A, i) =_M (B, j)$. Define $(A, i) \subset_M (B, j)$ similarly except $i \neq j$.

Finally, if there is a multi-cluster $(A, i) \in \mathcal{M}$ where A is a proper cluster on X, call (A, i) a *proper multi-cluster*.

Note in particular that for any multi-hierarchy on X, there is a hierarchy on X obtained by taking the *support* of $\mathcal{M}$, denoted $supp(\mathcal{M})$ and defined by

$$supp(\mathcal{M}) = \{A \mid (A, 1) \in \mathcal{M}\}.$$

This is of course not a one-to-one correspondence as there can be many multi-hierarchies with the same support.

Definition 4.4.3. Let $T \in RP(X)$ and $\mathcal{M} \in MRP(X)$. Then $\delta : H(T) \to \mathcal{M}$ is a *multi-hierarchy-preserving* map if the following properties hold for all $A, B \in H(T)$:

1. **Enveloping:** If $\delta(A) = (A', i)$, then $A \subseteq A'$, and
2. **Subset-Preserving:** $A \subset B$ implies that $\delta(A) \subset_M \delta(B)$.

The set of trees with a multi-hierarchy-preserving map into $\mathcal{M}$ is denoted $MHP(\mathcal{M})$.

The reason for introducing these definitions is that for an algorithm to compute potential $\leq_{HP}$-maximal elements of $HP(P)$, we require a systematic way of finding them. We will do this by taking certain intersections (see the algorithm below) of the clusters of $H(T)$ and $H(T')$, which unfortunately will not necessarily be a hierarchy. Observe that many of our results for hierarchy-preserving maps have an equivalent result for multi-hierarchy-preserving maps, proven in much the same way.

Lemma 4.4.4 (Multi-hierarchy equivalent of Lemma 4.2.10)**.** *Let $T \in RP(X), \mathcal{M} \in MRP(X)$, with a multi-hierarchy-preserving map $\delta : H(T) \to \mathcal{M}$. Suppose A and B are distinct inclusion-maximal subclusters of some third distinct cluster D in $H(T)$, where $D \neq A \cup B$ and $\delta(A) \subseteq_M \delta(B)$. Then the binding $T^D_{A \cup B} \in MHP(\mathcal{M})$.*

4.4.1 Forming a multi-hierarchy from two trees

Algorithm 1 takes the hierarchies of two trees to produce a multi-hierarchy.

Algorithm 1 MAKEMULTI: producing a multi-hierarchy from two trees

Require: T, T' trees.
$\mathcal{M} \leftarrow \emptyset$.
while $H(T)$ and $H(T')$ are non-empty **do**
  for all pairs of maximal clusters $A_i \in H(T)$ and $B_j \in H(T')$ **do**
    if $C = A_i \cap B_j$ is non-empty **then**
      $\mathcal{M} \leftarrow \mathcal{M} \cup \{(C, k)\}$, where k indicates the k-th occurrence of C
    end if
  Delete all inclusion-maximal clusters of $H(T)$ and $H(T')$
  end for
end while

We note here that as a tree has at most $2n$ clusters, the multi-hierarchy will contain at most $4n^2$ multi-clusters. In fact, this will generally not be a strict upper bound as we are only taking intersections of inclusion-maximal clusters with inclusion-maximal clusters, but it is sufficient for later showing that the algorithm has polynomial time complexity.

Proposition 4.4.5. *The set $\mathcal{M}$ obtained from T, T' using MAKEMULTI is a multi-hierarchy.*

Proof. It is easily seen that $\mathcal{M}$ contains $(X, 1)$ and all singleton tuples. The second entry of repeated elements being sequential from 1 to k is also obvious. Hence we just have to check requirement (2) of Definition 4.4.2.

Let (A, i) and (B, j) be two multi-clusters of $\mathcal{M}$ produced by the algorithm, and suppose that $A \cap B$ is non-empty. Suppose (A, i) was obtained by taking the intersection of A_1 and B_1, and that B was obtained by taking the intersection of A_2 and B_2. Now, since $A \cap B$ is non-empty, it follows that A_1 and A_2 have a non-empty intersection, and similarly for B_1 and B_2. It follows that either $A_1 \subseteq A_2$ or $A_1 \supset A_2$. Without loss of generality, suppose $A_1 \subseteq A_2$. Then A was obtained on either the same step as B, or a subsequent step. If produced on the same step, it follows that $A_1 = A_2$ and $B_1 = B_2$, as inclusion-maximal elements have non-empty intersection with each other. Therefore $A = B$. Otherwise, if A was obtained on a subsequent step, then $A_1 \subseteq A_2$ and $B_1 \subseteq B_2$ and so $A_1 \cap B_1 \subseteq A_2 \cap B_2$, and thus $A \subseteq B$. It follows that the set of clusters in $\mathcal{M}$ is a multi-hierarchy. □

As the resulting set of tuples from the algorithm is a multi-hierarchy, determination of a $\leq_{HP}$-maximal element of $HP(T, T')$ can be equivalently recognised as determination of a $\leq_{HP}$-maximal tree in $MHP(\mathcal{M})$, where $\mathcal{M}$ is the multi-hierarchy obtained from T, T'.

Example 4.4.6. Consider the trees T and T' on the set $X = \{a, b, c, d, e, f, g\}$ so that $P(T) = ab, abcde, abcdef$ and $P(T') = ab, abcde, abcdeg$. Then the proper multi-clusters of the multi-hierarchy obtained from T, T' are $\{(abcde, 1), (abcde, 2), (ab, 1)\}$ and the proper clusters in $supp(\mathcal{M})$ are $\{abcde, ab\}$.

Example 4.4.7. Suppose $\mathcal{M}$ is obtained via the algorithm from T, T' and has a support corresponding to the hierarchy $H(T)$ of the tree T. Then if $\mathcal{M} = \{(A, 1) | A \in H(T)\}$, the unique $\leq_{HP}$-maximal tree in $MHP(\mathcal{M})$ is T itself, and so $e_{HP}(T, T') = d_{HP}(T, T) + d_{HP}(T', T) = f(T) + f(T') - 2f(T)$.

Lemma 4.4.8. *Let $\mathcal{M}$ be the multi-hierarchy consisting only of $\{(A, 1), \ldots, (A, k)\}$ for $A \neq X$. Then, the maximum value of $f(T)$ for $T \in MHP(\mathcal{M})$ is*

$$f(T) = \begin{cases} k|A| - \frac{k(k+3)}{2}, & \text{if } |A| > k \\ \frac{(|A|-1)(|A|-2)}{2}, & \text{if } |A| \leq k. \end{cases}$$

Proof. Let $T \in MHP(\mathcal{M})$. First suppose there is some cluster C with more than two inclusion-maximal subclusters. Let two of them be A, B, and we can immediately see by Theorem 4.2.11 that $T^D_{A \cup B} \in MHP(\mathcal{M})$ and $T \leq_{HP} T^D_{A \cup B}$, so T is not $\leq_{HP}$-maximal. We can therefore assume every cluster of T has at most 2 inclusion-maximal subclusters.

Now, suppose that C is an inclusion-minimal cluster of T with respect to the requirement that C has two inclusion-maximal clusters, neither of which is a singleton. Let the two inclusion-maximal clusters be A and B. It follows that $f(T|_C) =$

$\frac{(|A|-1)(|A|-2)}{2} + \frac{(|B|-1)(|B|-2)}{2}$, which is maximised if $|A| = 1$ or $|B| = 1$. Therefore T can only have maximal $f(T)$ if there is no non-singleton cluster that does not have a singleton subcluster.

Therefore, the maximal possible value of $f(T)$ is achieved by mapping A into $(A, 1)$, then removing one element from A for each mapping into $(A, 2), (A, 3)$, etc. The result follows. □

Example 4.4.9. If $\mathcal{M}$ is the multi-hierarchy obtained from T, T', then, perhaps counterintuitively, it is not true in general that there exists a $\leq_{HP}$-maximal element of $HP(T, T')$ that is a refinement of $supp(\mathcal{M})$ that has maximal f. Consider $\mathcal{M} = \{(abcdef, 1), (abcdef, 2), (abcdef, 3), (ab, 1), (cd, 1), (ef, 1)\}$. Then the maximum value of $f(T)$ is 23 with e.g. $\{abcdef, abcde, abcd, ab, cd\}$, but the maximum value achievable with T a refinement of $supp(\mathcal{M})$ is $f(T) = 20$ with e.g. $\{abcdef, abcd, ab, cd, ef\}$.

We use Lemma 4.4.8 as inspiration for the next algorithm, in Section 4.4.2. In particular, that whenever MAKEMULTI produces a repeated cluster (i.e. a multi-cluster (A, i) with $i > 1$), we must delete one leaf from our cluster.

4.4.2 Finding a $\leq_{HP}$-maximal tree in $HP(T, T')$ using the multi-hierarchy of T, T'.

Algorithm 2 MAXTREE: an algorithm to find a maximal tree in $HP(T, T')$ with maximal rank.

Require: The multi-hierarchy $\mathcal{M}$ obtained from T and T'.

1: $T'' \leftarrow$ star tree.
2: **for all** $(A, i) \in \mathcal{M}$ **do**
Let A' be the unique largest subcluster of A for which $H(T'') \cup \{A'\}$ is a hierarchy.
3: **if** $A' \notin H(T'')$ **then**
4: $H(T'') \leftarrow H(T'') \cup \{A'\}$.
5: **else if** $A' \in H(T'')$ **then**
6: **if** $|A'| > 1$ **then** choose $x \in A'$
7: $H(T'') \leftarrow H(T'') \cup \{A' \setminus \{x\}\}$.
8: **end if**
9: **end if**
10: **end for**
By iterating over all possible choices in line 6, we will find all $\leq_{HP}$-maximal trees in $HP(T, T')$ (or equivalently $MHP(\mathcal{M})$), and we take the tree with the highest rank.

We will show this algorithm has polynomial run-time in the proof of the following proposition.

Proposition 4.4.10. *The algorithmic complexity of determining the upper bound e_{HP} to $d_{HP}(T, T')$ is polynomial.*

Proof. Calculation of the rank $f(T)$ of T is linear because there are at most n clusters in a tree.

Calculation of the multi-hierarchy via MAKEMULTI (Algorithm 1) involves a linear number of intersections, and intersections can be done in linear time. Hence calculation of the multi-hierarchy is quadratic.

The only part of MAXTREE (Algorithm 2) that allows for choice is determining which elements to remove when we have repeated clusters. There are at most $4n^2$ multi-clusters in a multi-hierarchy obtained from two trees, and each cluster has a maximum of n elements that we can choose to remove. Hence there is a maximum of $4n^3$ possible choices for a given multi-hierarchy, so iterating over all possible choices and checking $f(T)$ for each one will be polynomial in time complexity. □

Example 4.4.11. Unfortunately, e_{HP} is not equal to d_{HP} in general, as the following example demonstrates. Let T and T' be trees on $X = \{a, b, c, d, e, f, g\}$ with $P(T) = \{abc, de\}$ and $P(T') = \{ae, bdf\}$. Then the star tree is the unique tree with a hierarchy preserving map into both T and T', so the algorithm gives a distance of $e_{HP}(T, T') = d_{HP}(T, S) + d_{HP}(T', S) = 3 + 3 = 6$. However, it is not difficult to find a path of length 4 from T to T' in $\mathcal{H}(X)$. For example, let U_1, U_2, U_3 be trees with $P(U_1) = \{ab, de\}, P(U_2) = \{abde\}$ and $P(U_3) = \{ae, bd\}$. Then the path T, U_1, U_2, U_3, T' is one such path.

Observation 4.4.12. The above example also shows that e_{HP} is not a metric, because it fails the triangle inequality: we have $e_{HP}(T, U_2) = e_{HP}(U_2, T') = 2$, but $e_{HP}(T, T') = 6$.

4.5 Computational results

We have implemented the algorithms required to compute e_{HP}, and in this section present some preliminary results. Because MCMC algorithms often examine only binary trees, we explore both all of $RP(X)$ and also $BRP(X)$, the set of binary trees.

A naïve algorithm to calculate the true distance d_{HP} (by checking all trees along all possible paths shorter than e_{HP}, with some optimisations) can be used for trees on up to nine leaves, although the same approach for ten or more leaves can be very slow. The algorithm, implemented in Python, can be found at [23]. In short, it sequentially generates the sets of trees within $\lfloor \frac{e_{HP}}{2} \rfloor$ bindings/unbindings of T_1 and T_2 (ignoring duplicates), while checking whether there is any intersection between each set. If any tree appears in both sets, the true distance is shorter than e_{HP}. In the worst cases, this has exponential runtime (due to the potential size of the neighbourhood of each tree), but with $n < 10$ the coefficient of the exponential is sufficiently small that calculations took at most slightly over thirty minutes.

4.5.1 Comparison of the upper bound e_{HP} with the true distance d_{HP}.

Figure 4.5 shows the results of an experiment on 100 random pairs of trees with 9 leaves. The data indicate that the upper bound is reasonably accurate, with e_{HP} and d_{HP} being equal in 77% of cases. The mean upper bound distance in this simulation was 9.87, while the mean true distance was 9.39. The biggest difference between the upper bound and the true distance was 4.

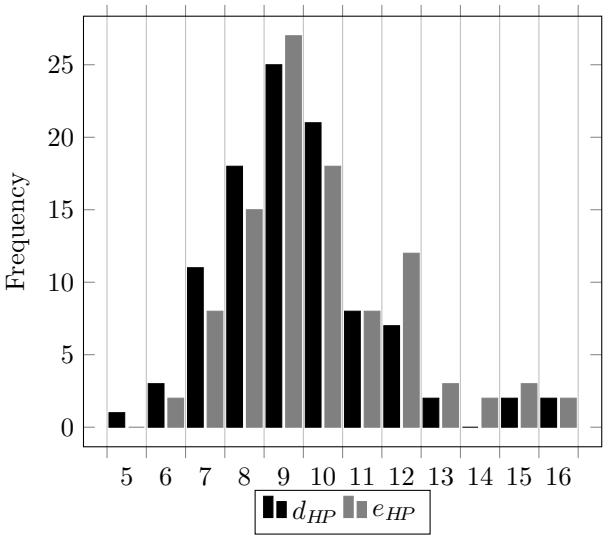


Figure 4.5: A comparison of e_{HP} with d_{HP}, on trees with $n = 9$ leaves.

On the same data set, we also investigated how the proportion of e_{HP} values of a given distance were related to the value of e_{HP}, with results given in Figure 4.6. Overall it appears that the larger the e_{HP}, the more likely that e_{HP} differs from d_{HP}, with the abrupt increase at distances 15 and 16 likely due to small sample sizes at this distance. We were unable to confirm this due to the exponential time that it takes our current algorithm to find d_{HP}.

4.5.2 Experimental results on the upper bound e_{HP}.

Table 4.1 shows some representative distance statistics for the upper bound e_{HP} on the distance.

The Average Distance column indicates the average e_{HP} between pairs, to three decimal places. These are provided as a baseline from which to judge the distance for a given pair of trees.

The Maximum Distance column shows the maximum recorded e_{HP} between a pair of trees. Note that all trees that are the result of simulations only provide a lower bound on the maximum e_{HP}, which is again an upper bound on the true e_{HP}.

In particular, note that in Table 4.1, both the average and maximum e_{HP} on $BRP(X)$ are larger than those on all of $RP(X)$. Indeed, for $n = 40$ on binary trees

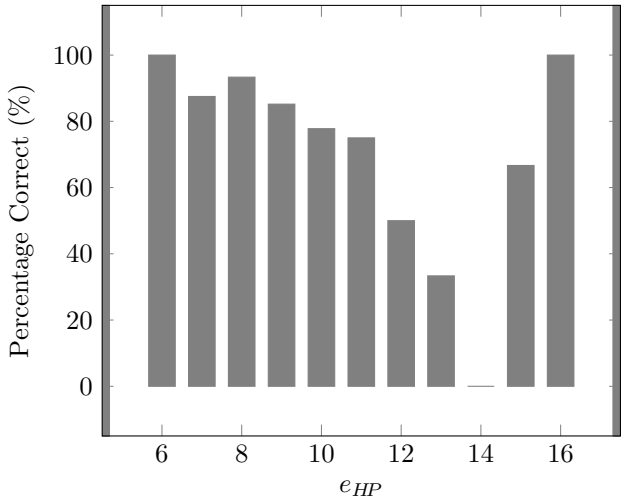


Figure 4.6: A comparison of e_{HP} with the proportion of values of e_{HP} for which $e_{HP} = d_{HP}$, on trees with $n = 9$ leaves.

the average distance is larger than the maximum distance obtained for $n = 40$ on all trees! For such large trees the distributions of distances seem to radically diverge, as seen in Figure 4.7, which shows distances for 20,000 randomly selected pairs of trees.

Of course, the distributions don't *actually* diverge, because after all the binary trees $BRP(X)$ are a subset of the set of all trees $RP(X)$. However the binary trees sit along the top of the very large Hasse diagram, since they are all of maximal rank (Prop 4.2.12), so the range of potential distances between them is therefore higher than any pair of nonbinary trees (Corollary 4.3.4). It is therefore, heuristically at least, unsurprising that the distances are correspondingly higher.

Part of the explanation for the apparent divergence of the distributions seen in Figure 4.7 in the 40 leaf case is that the binary trees are such a small proportion of the total number of trees that when selecting a pair of random trees one almost never selects a pair of binary trees.

In the sampling, trees are selected by randomly partitioning the set of leaves, and successively partitioning the components of the partition until all components have cardinality 1 (the leaves). To select a binary tree, each successive partition must be a partition into exactly two components. The probability of doing this is the number of partitions of 40 into two parts divided by the total number of partitions into any number of parts k. These are counted by the Stirling numbers of the second kind, $\left\{ {n \atop k} \right\}$. So the probability of even the first partition (immediately below the root) being binary is just

$$\frac{\left\{ {40 \atop 2} \right\}}{\sum_{k=2}^{40} \left\{ {40 \atop k} \right\}},$$

which is approximately 3.49×10^{-24}. To select a fully binary tree one would need to continue to choose further partitions into two parts at each point.

It is perhaps worth noting the symmetry of the distributions shown in Figure 4.7,

	RP(X)		BRP(X)	
n	Average e_{HP}	Maximum e_{HP}	Average e_{HP}	Maximum e_{HP}
4	2.587	4	3.0	4
5	4.645	8	5.525	8
6	5.294	12	8.440	12
7	6.990	16	10.123	16
8	8.752	17	12.900	19
9	10.708	21	15.883	24
10	12.695	24	18.983	29
20	35.719	57	56.344	74
40	91.662	123	151.527	176

Table 4.1: Distance statistics for pairs of trees with each number of leaves. For $|X| \leq 6$ (resp. $|X| \leq 5$) these statistics represent calculations over *all* pairs of trees in $RP(X)$ (resp. $BRP(X)$). For larger leaf sets the results are the outcome of testing a sample of $20,000$ random pairs of trees.

which suggest that the metric e_{HP} avoids the skew that affects the Robinson-Foulds metric.

4.6 Discussion

The new metric on phylogenetic tree space introduced in this chapter has several interesting properties that may make it valuable for biological applications.

First of all, it is a cluster-similarity metric, so the notion of distance between two trees corresponds to the similarity of their hierarchies. This in itself is a valuable property in terms of comparisons of trees that have arisen under related processes, such as gene trees in the presence of incomplete lineage sorting.

Second, in contrast to other cluster-similarity metrics, this metric has a simple local operation to move around tree space, ensuring easy calculation of neighbourhoods. This feature, coupled with the cluster-similarity property, can be expected to help with MCMC searches of tree-space around trees of similar hierarchies.

And third, the distribution of distances on a given tree space appears to be quite symmetric, and also to have a reasonable spread of values. This will be valuable in choosing trees from a set that are closest to each other or to a special tree (such as a purported species tree), in a way consistent with their hierarchies, and also makes it capable of distinguishing trees in a way required in the biological studies mentioned in the Introduction.

A primary goal for future study would be to either find an efficient method for calculating d_{HP} exactly, or a proof of NP-hardness. If found to be NP-hard, results regarding the accuracy of the upper bound e_{HP} would prove useful, as would a determination of whether the problem is FPT. It would also be interesting to find

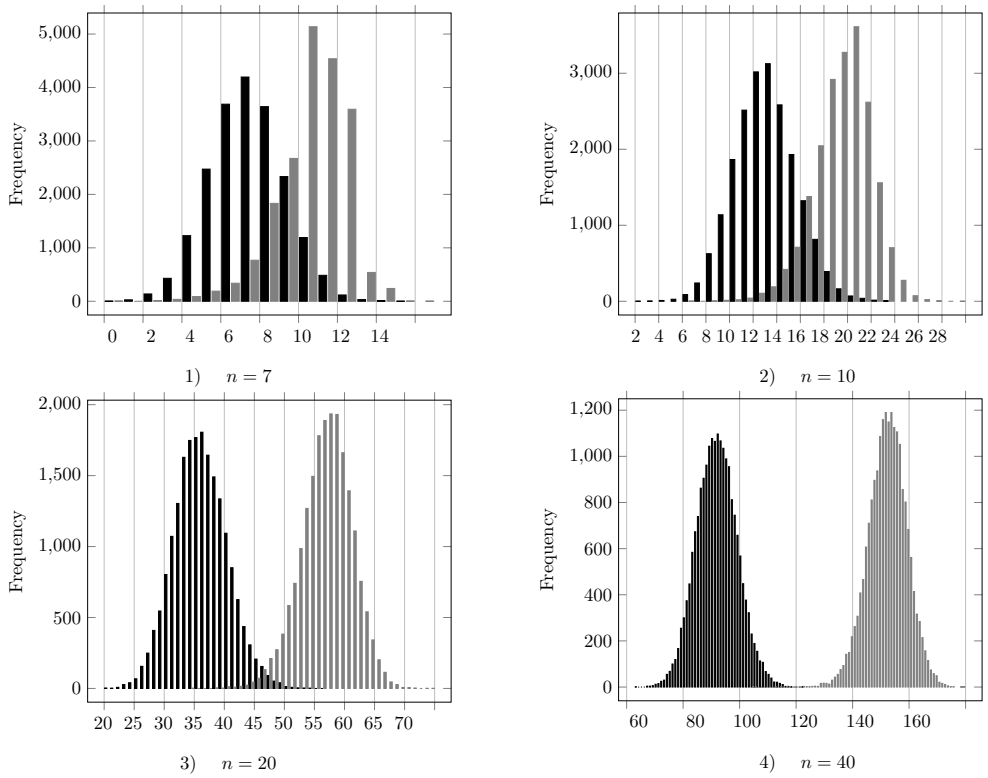


Figure 4.7: Histograms of e_{HP} under 20,000 simulations of random pairs of trees with n leaves. Simulations using trees randomly selected from all $RP(X)$ in black, and $BRP(X)$ in grey.

tighter bounds for many of the results in this chapter. For instance, under d_{HP}, the diameter of $RP(X)$ and the neighbourhood size of a given tree T can almost certainly be given better bounds.

It may be that the ranks of trees are able to provide additional information for estimating tree distances. For instance, Corollary 4.3.4 allows one to estimate distances between trees quite well if one or both trees have small rank. Further, it is not difficult to show that for any pair of binary trees T, T', the distance $d_{HP}(T, T') < f(T) + f(T')$ - note the strict inequality. Hence further research into the relationship between the ranks of trees and the distances between them may be fruitful.

Finally, the notion of hierarchy-preserving maps may be of independent mathematical interest. It is one of many possible generalisations of refinement, and as such is compatible with the notion. To our knowledge, the induced partial order and the concept of binding are both also new and may provoke further interest in the mathematical community.

Bibliography

[1] R. Alberich, G. Cardona, F. Rosselló, and G. Valiente, *An algebraic metric for phylogenetic trees*, Applied Mathematics Letters **22** (2009), no. 9, 1320–1324.

[2] J. O. Andersson, *Lateral gene transfer in eukaryotes*, Cellular and Molecular Life Sciences **62** (2005), no. 11, 1182–1197.

[3] D. Bogdanowicz and K. Giaro, *On a matching distance between rooted phylogenetic trees*, International Journal of Applied Mathematics and Computer Science **23** (2013), no. 3, 669–684.

[4] R. L. Brooks, *On colouring the nodes of a network*, Mathematical Proceedings of the Cambridge Philosophical Society, vol. 37, Cambridge University Press, 1941, pp. 194–197.

[5] P. Buneman, *The recovery of trees from measures of dissimilarity*, Mathematics in the Archaeological and Historical Sciences (1971).

[6] G. Cardona, M. Llabrés, F. Rosselló, and G. Valiente, *Nodal distances for rooted phylogenetic trees*, Journal of Mathematical Biology **61** (2009), no. 2, 253–276.

[7] S. R. Cole, D. F. Wright, and W. I. Ausich, *Phylogenetic community paleoecology of one of the earliest complex crinoid faunas (Brechin Lagerstätte, Ordovician)*, Palaeogeography, Palaeoclimatology, Palaeoecology **521** (2019), 82–98.

[8] E. Corel, P. Lopez, R. Mheust, and E. Bapteste, *Network-thinking: Graphs to analyze microbial complexity and evolution*, Trends in Microbiology (2016).

[9] T. Dagan, Y. Artzy-Randrup, and W. Martin, *Modular networks and cumulative impact of lateral transfer in prokaryote genome evolution*, Proceedings of the National Academy of Sciences **105** (2008), no. 29, 10039–10044.

[10] T. Dagan and W. Martin, *The tree of one percent*, Genome Biol **7** (2006), no. 10, p. 118.

[11] W. F. Doolittle and E. Bapteste, *Pattern pluralism and the tree of life hypothesis*, Proceedings of the National Academy of Sciences **104** (2007), no. 7, pp. 2043–2049.

[12] B. W. van Ee and P. E. Berry, *Croton section Pedicellati (Euphorbiaceae), a novel new world group, and a new subsectional classification of Croton section Lamprocroton*, Systematic Botany **36** (2011), no. 1, 88–98.

[13] J. Felsenstein, *Inferring phylogenies*, Sinauer Press, 2004.

[14] J. Fischer and D. H. Huson, *New common ancestor problems in trees and directed acyclic graphs*, Information Processing Letters **110** (2010), no. 8-9, 331–335.

[15] M. Fischer, M. Galla, L. Herbst, Y. Long, and K. Wicke, *Non-binary treebased unrooted phylogenetic networks and their relations to binary and rooted ones*, arXiv:1810.06853.

[16] A. Francis, K. T. Huber, and V. Moulton, *Tree-based unrooted phylogenetic networks*, Bulletin of Mathematical Biology **80** (2018), no. 2, 404.

[17] A. Francis, C. Semple, and M. Steel, *New characterisations of tree-based networks and proximity measures*, Advances in Applied Mathematics **93** (2018), 93–107.

[18] A. Francis and M. Steel, *Tree-like reticulation networks: When do tree-like distances also support reticulate evolution?*, Mathematical Biosciences **259** (2015), 12–19.

[19] A. Francis and M. Steel, *Which phylogenetic networks are merely trees with additional arcs?*, Systematic Biology **64** (2015), no. 5, 768–777.

[20] M. Guo, J. Li, and Y. Liu, *A topological transformation in evolutionary tree search methods based on maximum likelihood combining p-ECR and neighbor joining*, BMC Bioinformatics **9** (2008), no. Suppl 6, S4.

[21] J. Hein, *Reconstructing evolution of sequences subject to recombination using parsimony*, Mathematical Biosciences **98** (1990), no. 2, 185–200.

[22] M. Hendriksen, *Tree-based unrooted nonbinary phylogenetic networks*, Mathematical Biosciences **302** (2018), 131–138.

[23] ———, *Clustermetric. https://github.com/mahendriksen/clustermetric*, 2019, GitHub repository.

[24] M. Hendriksen and A. Francis, *Tree-metrizable HGT networks*, Mathematical Biosciences **318** (2019), 108283.

[25] ———, *A partial order and cluster-similarity metric on rooted phylogenetic trees*, Journal of Mathematical Biology **80** (2020), no. 5, 1265–1290.

[26] D. H. Huson and D. Bryant, *Application of phylogenetic networks in evolutionary studies*, Molecular Biology and Evolution **23** (2005), no. 2, 254–267.

[27] D. H. Huson, T. Dezulian, T. Klopper, and M. A. Steel, *Phylogenetic supernetworks from partial trees*, IEEE/ACM Transactions on Computational Biology and Bioinformatics **1** (2004), no. 4, 151–158.

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

[29] H. Jeong, B. Arif, G. Caetano-Anollés, K. Kim, and A. Nasir, *Horizontal gene transfer in human-associated microorganisms inferred by phylogenetic reconstruction and reconciliation*, Scientific Reports **9** (2019), no. 1.

[30] L. Jetten and L. van Iersel, *Nonbinary tree-based phylogenetic networks*, IEEE/ACM Transactions on Computational Biology and Bioinformatics (2016).

[31] K. Kuhls, E. Cupolillo, S. O. Silva, C. Schweynoch, M. C. Boité, M. N. Mello, I. Mauricio, M. Miles, T. Wirth, and G. Schönian, *Population structure and evidence for both clonality and recombination among Brazilian strains of the subgenus Leishmania (Viannia)*, PLoS Neglected Tropical Diseases **7** (2013), no. 10, e2490.

[32] M. K. Kuhner and J. Yamato, *Practical performance of tree comparison metrics*, Systematic Biology **64** (2014), no. 2, 205–214.

[33] W. Martin, *Early evolution without a tree of life*, Biology Direct **6** (2011), no. 1, 1.

[34] G. W. Moore, M. Goodman, and J. Barnabas, *An iterative approach from the standpoint of the additive hypothesis to the dendrogram problem posed by molecular data sets*, Journal of Theoretical Biology **38** (1973), no. 3, 423–457.

[35] V. Moulton and T. Wu, *A parsimony-based metric for phylogenetic trees*, Advances in Applied Mathematics **66** (2015), 22–45.

[36] K. Okumura, Y. Shimomura, S. Murayama, J.Yagi, K. Ubukata, T. Kirikae, and T. Miyoshi-Akiyama, *Evolutionary paths of streptococcal and staphylococcal superantigens*, BMC Genomics **13** (2012), no. 1, 404.

[37] M. D. Pirie, A. M. Humphreys, N. P. Barker, and H. P. Linder, *Reticulation, data combination, and inferring evolutionary history: an example from Danthonioideae (Poaceae)*, Systematic Biology **58** (2009), no. 6, 612–628.

[38] J. C. Pons, C. Semple, and M. Steel, *Nonbinary tree-based networks: characterisations, metrics, and support trees*, arXiv:1710.07836 (2017).

[39] D. F. Robinson and L. R. Foulds, *Comparison of phylogenetic trees*, Mathematical Biosciences **53** (1981), no. 1-2, 131–147.

[40] K. H. Rosen, *Handbook of discrete and combinatorial mathematics*, Chapman and Hall/CRC, 2017.

[41] V. Rougeron, T. De Meeûs, M. Hide, E. Waleckx, J. Dereure, J. Arevalo, A. Llanos-Cuentas, and A. Banuls, *A battery of 12 microsatellite markers for genetic analysis of the Leishmania (Viannia) guyanensis complex*, Parasitology **137** (2010), no. 13, 1879–1884.

[42] C. Semple, *Phylogenetic networks with every embedded phylogenetic tree a base tree*, Bulletin of Mathematical Biology **78** (2016), no. 1, 132–137.

[43] G. Sevillya and S. Snir, *Synteny footprints provide clearer phylogenetic signal than sequence data for prokaryotic classification*, Molecular Phylogenetics and Evolution **136** (2019), 128–137.

[44] L. Shuguang and L. Zhihui, *Algorithms for computing cluster dissimilarity between rooted phylogenetic trees*, The Open Cybernetics and Systemics Journal **9** (2015), no. 1, 2218–2223.

[45] M. Steel, *Phylogeny : Discrete and random processes in evolution*, Society for Industrial and Applied Mathematics, 2016.

[46] C. R. Woese, *On the evolution of cells*, Proceedings of the National Academy of Sciences **99** (2002), no. 13, 8742–8747.

[47] L. Zhang, *On tree-based phylogenetic networks*, Journal of Computational Biology **23** (2016), no. 7, 553–565.

[48] L. Zhang, P. Ma, Y. Zhang, C. Zeng, L. Zhao, and D. Li, *Using nuclear loci and allelic variation to disentangle the phylogeny of Phyllostachys (Poaceae, Bambusoideae)*, Molecular Phylogenetics and Evolution **137** (2019), 222–235.