An algorithm for reconstructing ultrametric tree-child networks from inter-taxa distances

M. Bordewich, N. Tokac*

School of Engineering and Computing Sciences, Durham University, Lower Mountjoy, South Road, Durham, DH1 3LE, UK

Abstract

Traditional "distance based methods" reconstruct a phylogenetic tree from a matrix of pair-wise distances between taxa. A phylogenetic network is a generalization of a phylogenetic tree that can describe evolutionary events such as reticulation and hybridization that are not tree-like. Although evolution has been known to be more accurately modelled by a network than a tree for some time, only recently have efforts been made to directly reconstruct a phylogenetic network from sequence data, as opposed to reconstructing several trees first and then trying to combine them into a single coherent network. In this work we present a generalisation of the UPGMA algorithm for ultrametric tree reconstruction which can accurately reconstruct ultrametric tree-child networks from the set of distinct distances between each pair of taxa.

Keywords:
UPGMA, Phylogenetic Networks, Ultrametric Networks

1. Introduction

The evolutionary history of organisms is generally represented by a phylogenetic tree. One popular and fast method for reconstructing phylogenetic tree from DNA or protein sequence data is to first compute a matrix of pairwise distances between the taxa, and then infer the phylogenetic tree from

*Corresponding author
Email addresses: m.j.r.bordewich@durham.ac.uk (M. Bordewich), nihan.tokac@durham.ac.uk (N. Tokac)

this distance matrix. Such approaches are called *distance-based* methods, and they are very widely used due to their simplicity and speed. The two most well known and long standing approaches are UPGMA [1] and Neighbor Joining [2]. In recent years several variants of these and new approaches have been suggested, including Least Squares [3], BioNJ [4] and Balanced Minimum Evolution [5]. The properties and accuracy of distance based methods have been widely studied, see for example [6, 7, 8].

In this paper, we consider the task of reconstructing phylogenetic networks from distance data. A phylogenetic network is a generalization of a phylogenetic tree, which can be used to describe the evolutionary history of a set of species that is non-tree like because of reticulation events such as hybridization, horizontal gene transfer or recombination. The reconstruction of restricted classes of phylogenetic network from inter-taxa distances have been studied in a number of recent papers. A key feature of this problem is that in a network there is no longer a unique distance between a pair of taxa (as there is in a tree), so one must work with shortest distances, average distances or sets or subsets of distances. Chan et al. [9] take a matrix of inter-taxa distances and reconstruct an ultrametric galled network (more commonly called a galled tree or a level-1 network) such that there is a path between each pair of taxa having the weight given in the matrix, if such network exists. Willson [10] studied the problem of determining the network given the average distance between taxa, where each reticulation vertex assigns a probability to its two incoming arcs. He manages the reconstruction of phylogenetic networks which have a single reticulation cycle from such distances in polynomial time [11]. In a recent paper [12], Bordewich and Semple showed that (unweighted) tree-child phylogenetic networks may be reconstructed from the multi-set of path lengths between taxa and that temporal, tree-child, phylogenetic networks may be reconstructed from the set of path lengths between taxa, each in polynomial time in the size of the input.

In this paper, which builds on and extends the approach of [12], we present a polynomial-time algorithm (which we have called NetworkUPGMA) that reconstructs an ultrametric tree-child network from the set of distances between each pair of taxa. Our algorithm offers an improvement over previous works in two ways. First ultrametric tree-child networks are a much wider class of networks than networks with only a single reticulation or ultrametric galled networks, which are a subclass of ultrametric tree-child networks. In particular note that: the total number of reticulations in a tree-child network on n taxa can be as large as $n - 1$ [13], whereas a galled network

has at most $n/2$ reticulations; and the interrelation of reticulations may be more complex, as each 2-connected component of our networks may contain many reticulations (again linear in the number of taxa), whereas in a galled network there can only be one reticulation in each 2-connected component. Second, the algorithm takes the *set of distances* between each pair of taxa as input, where Bordewich and Semple [12] required the *multiset of path lengths* (for unweighted tree-child networks). This is an important distinction: the distance matrices come from estimating evolutionary distance based upon sequence data of some type. Real phylogenies are weighted: edge weights correspond to some measure of genetic difference. Furthermore, while it is quite conceivable that by sampling different genes or regions of the genome one might build up an accurate picture of the set of different evolutionary path weights between a given pair of taxa, it seems hard to imagine how one might manage to measure the number of distinct evolutionary paths of a given observed weight. Thus the set of distances seems a much more reasonable input for an algorithm in practice. (Note, however, that only through study of the multi-set problem did we gain the understanding needed to tackle this newer work).

2. Definitions and Statement of Results

In this section we formally define the central concepts of phylogenetic networks and give further definitions which we shall require in order to present our algorithm and proof. Throughout the paper, standard notation and terminology follows Semple and Steel [14]. X denotes a non-empty finite set of taxa. A rooted phylogenetic X-tree $\mathcal{T}$ is a rooted tree with no degree-two vertices, except possibly the root which has degree at least two, and whose leaf set is X. An X-tree is binary if either $|X| = 1$ or the root has degree two and every other interior vertex has degree three.

2.1. Ultrametric tree-child networks

A phylogenetic network $\mathcal{N}$ on X is a rooted, connected, directed acyclic graph with the following properties:

(i) exactly one node (the root) has in-degree 0 and all other nodes have in-degree 1 or 2,

(ii) any node with in-degree 2 (called a reticulation) has out-degree 1 and all other nodes have out-degree 0 (called leaves) or 2 (called tree vertices), and

(iii) each node with out-degree 0 is labelled with a distinct element of X (taxon).

Note that, what we have called a phylogenetic network is sometimes referred to as a *binary* phylogenetic network.

A network $\mathcal{N}$ is *weighted* if there is a positive weighting (or length) associated with each arc, which is strictly positive for all tree arcs (those arcs whose head is a tree vertex or leaf). For arc $e = (u, v)$ the weight is denoted by l_e or $l(u, v)$. The weight of a path is the sum of the weights of arcs it contains. An *ultrametric network* is a weighted phylogenetic network such that every directed path from the root to any leaf has the same weight [15, 9]. This implies that for any vertices u, v such that there is a directed path from u to v in $\mathcal{N}$, every path from u to v has the same weight, which we denote $d_{u,v}$.

Let $\mathcal{N}$ be a phylogenetic network on X. For any two vertices u and v in $\mathcal{N}$ that are joined by an arc (u, v), we say u is a parent of v and, conversely, v is a child of u. Cardona et. al. [13] discussed "tree-child" networks, in which every vertex that is not a leaf has a child that is a tree vertex or leaf. We say an ultrametric network is *ultrametric tree-child network* if every non-leaf has a child which is either a tree vertex or a leaf. For vertices u, v such that there is a directed path from u to v in $\mathcal{N}$, we say the path is a *tree-path* if every vertex on the path, except possibly u, is a tree vertex or a leaf. Note that in a tree-child network every vertex has a tree-path to a leaf.

2.2. Distance matrices

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

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

and

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

The weight of an up-down path is the sum of the weights of the two directed paths it contains. The vertex v_i is called the peak of the up-down path. In any rooted network $\mathcal{N}$, a *least common ancestor* of two vertices x and y is a vertex v such that there is an up-down path from x to y with v the peak of

the path. By this definition there are multiple least common ancestors for x and y. However for each, the paths v to x and v to y are arc-disjoint, so there could be some genetic inheritance from the root of the network to x and y that has a common path as far as v and then diverges.

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

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

2.3. Equivalent networks

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

Two ultrametric tree-child networks $\mathcal{N}_1, \mathcal{N}_1'$ are said to be *equivalent up to weights at reticulations* (denoted $\equiv_1$) if the underlying unweighted networks are isomorphic and: at each reticulation v with incoming arcs e_1 and e_2 and outgoing arc e_3, the weight of the path e_1, e_3 is the same in $\mathcal{N}_1$ and $\mathcal{N}_1'$, and also the weight of the path e_2, e_3 is the same in $\mathcal{N}_1$ and $\mathcal{N}_1'$. Thus if arcs e_1, e_2, e_3 have weights l_1, l_2, l_3 respectively, any network $\mathcal{N}'$ formed by changing the weights of arcs e_1, e_2, e_3 to $l_1 - \epsilon, l_2 - \epsilon, l_3 + \epsilon$ respectively for some $\epsilon \in (-l_3, \min\{l_1, l_2\})$ is equivalent to $\mathcal{N}_1$ up to weights at reticulations. We define a class representative for each equivalence class as the network in which one of the incoming edge weights is zero at every reticulation. E.g. for the network $\mathcal{N}_1$, the class representative would have arcs e_1, e_2, e_3 with weights $l_1 - \epsilon, l_2 - \epsilon, l_3 + \epsilon$ where $\epsilon = \min\{l_1, l_2\}$. In Fig. 1 networks $\mathcal{N}_1 \equiv_1 \mathcal{N}_1'$ and $\mathcal{N}_2 \equiv_1 \mathcal{N}_2'$. Moreover $\mathcal{N}_1'$ and $\mathcal{N}_2'$ are class representatives.

We next define a second equivalence relation, denoted $\equiv_2$, on the class representatives. A reticulation vertex whose two parents are also parent and child is said to be an *immediate reticulation*, i.e. v is an immediate reticulation if it is a reticulation node with parents u and w such that w is

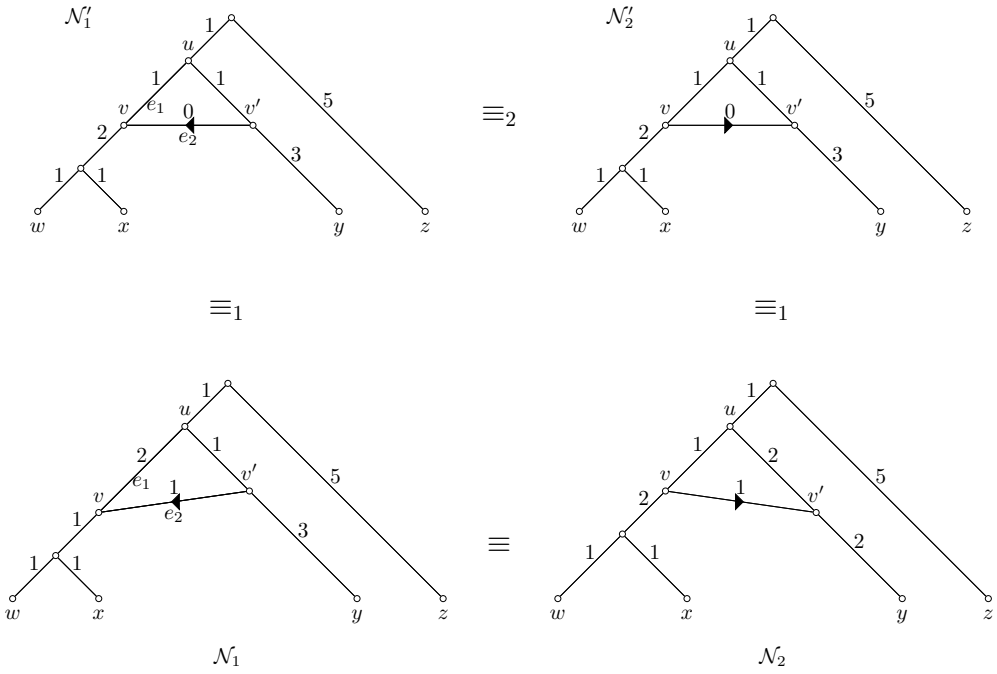

Figure 1: Four ultrametric tree-child networks each containing an immediate reticulation. Networks $\mathcal{N}_1$ and $\mathcal{N}_1'$ (and $\mathcal{N}_2$ and $\mathcal{N}_2'$) are equivalent up to weights at reticulations. Networks $\mathcal{N}_1'$ and $\mathcal{N}_2'$ are equivalent up to direction of immediate reticulations. Thus networks $\mathcal{N}_1$ and $\mathcal{N}_2$ are equivalent under $\equiv$.

also a child of u. In the case that the network is a class representative, the arc (w, v) has weight 0. An immediate reticulation occurs when a parent species immediately recombines with its own offspring. In each network shown in Fig. 1 the reticulation is an immediate reticulation. We say two class representative phylogenetic networks $\mathcal{N}_1', \mathcal{N}_2'$ are *equivalent up to direction of immediate reticulations* if for some set of immediate reticulations R in $\mathcal{N}_1'$ such that v_i in R has parents u_i, w_i where (u_i, w_i) is an arc, then the network $\mathcal{N}_2'$ is formed by removing the arcs (w_i, v_i) and inserting the arcs (v_i, w_i) (so that w_i is now a immediate reticulation with parents u_i, v_i), where the new arcs have weight 0. Note that $\mathcal{N}_1'$ and $\mathcal{N}_2'$ display the same set-distance matrix. In Fig. 1, the network $\mathcal{N}_1' \equiv_2 \mathcal{N}_2'$.

Finally we define the equivalence relation $\equiv$ on phylogenetic networks, where $\mathcal{N}_1 \equiv \mathcal{N}_2$ if the class representatives (under $\equiv_1$) for $\mathcal{N}_1$ and $\mathcal{N}_2$ are equivalent under $\equiv_2$. For example in Fig. 1, $\mathcal{N}_1 \equiv \mathcal{N}_2$ since $\mathcal{N}_1 \equiv_1 \mathcal{N}_1' \equiv_2 \mathcal{N}_2' \equiv_1 \mathcal{N}_2$. Observe that if $\mathcal{N}_1 \equiv \mathcal{N}_2$ and $\mathcal{N}_1$ is an ultrametric tree-child network, then $\mathcal{N}_2$ is an ultrametric tree-child network. Also, $\mathcal{N}_1$ and $\mathcal{N}_2$ will display the same set-distance matrix.

2.4. Cherry reductions

Let $\mathcal{N}$ be an ultrametric tree-child network on X. A 2-element subset $\{x, y\}$ of X is a *cherry* in $\mathcal{N}$ if the parents of x and y are the same. Note that the distances from this parent to x and y are the same. Fig. 2 (a) depicts a cherry $\{x, y\}$. *Reducing a cherry* $\{x, y\}$ is the operation replacing the cherry with a single new node while keeping the ultrametric property, see Fig. 2(d). Note that the number of leaves in the resulting network is reduced by one, but the number of reticulations is unchanged.

A two-element subset $\{x, y\}$ of X is a *reticulated cherry* in $\mathcal{N}$ if there is an up-down path consisting of three edges, say $(x, u), (u, v), (v, y)$, between x and y where u is a tree vertex, and v is a reticulation vertex. Necessarily, the arc joining u and v is directed from a tree vertex to the reticulation. This arc is referred to as the reticulation arc of the reticulated cherry. The leaf adjacent to the tree vertex is called the tree leaf of the reticulated cherry, and the leaf adjacent to the reticulation is the reticulation leaf of the reticulated cherry. Fig. 2 (b) depicts a reticulated cherry $\{x, y\}$. Note that the distance between u and x is equal to the distance between u and y because of the ultrametric property. *Reducing a reticulated cherry* $\{x, y\}$ is the operation of deleting the incoming arc to the reticulation vertex that is not

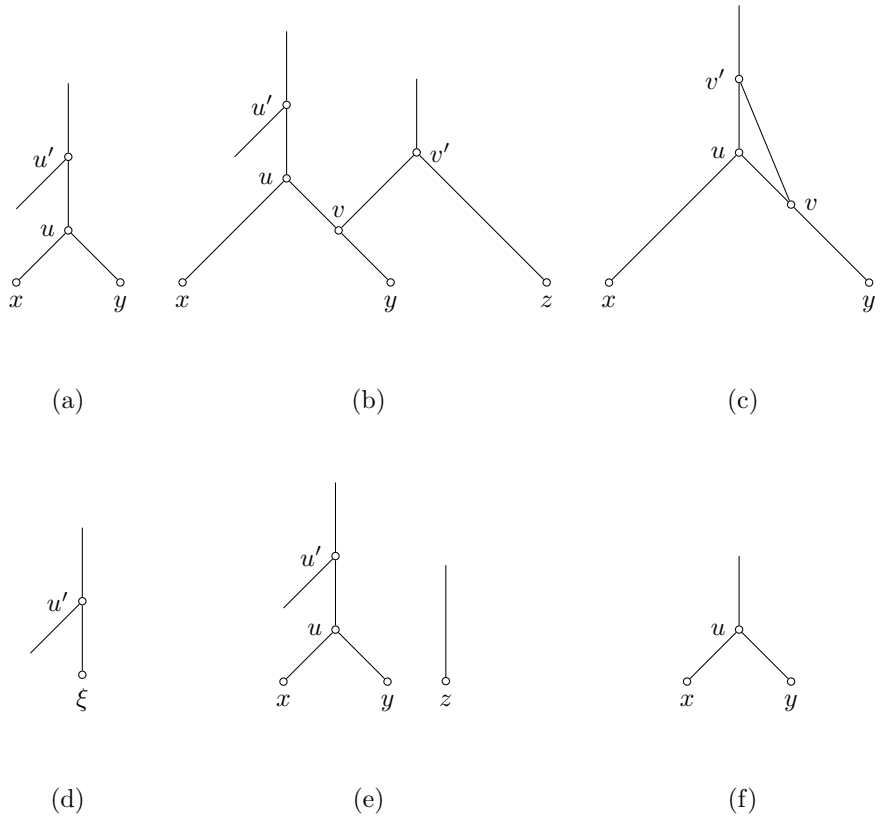

Figure 2: Reduction of (a) cherry, (b) reticulated cherry, (c) reticulated cherry with immediate reticulation

part of the reticulated cherry (i.e. the incoming arc that is not (u, v)) and suppressing the degree-two vertices resulting from the deletion, see Fig 2(e). Observe that, by reducing a reticulated cherry, the number of reticulations in the resulting network is reduced by one, but the number of leaves and, in particular, the leaf set, is unchanged. An immediate reticulation is a special case of a reticulation, and the reduction of an immediate reticulation is the same as for a normal reticulation. Fig. 2(c) shows an immediate reticulation, and Fig. 2(f) shows the result of reducing this immediate reticulation.

Note that the above definition is different from the reticulated cherry reduction used in [12] since they delete the arc (u, v).

2.5. Main result

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

It was already known that a tree-child network with every edge weight 1 can be reconstructed from the *multiset matrix* [12] (up to the direction of an immediate reticulation at the root). Our result generalizes this to arbitrary positive edge weights and reconstructing from the set-distance matrix; however, this comes at the cost of restricting attention to only *ultrametric* tree-child networks.

Theorem 1. *Given a set-distance matrix $\mathcal{D}$ on X, if there is an ultrametric tree-child network $\mathcal{N}$ that displays $\mathcal{D}$, then $\mathcal{N}$ is the unique such network (up to $\equiv$) and may be found in polynomial time.*

The remainder of the paper is organised as follows. In Section 3, we describe the algorithm NetworkUPGMA that is central to the paper. In Section 4 we show that the algorithm is correct, and in Section 5 we show that the algorithm's running time is polynomial in the number of taxa $|X|$.

3. The Algorithm NetworkUPGMA

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

For a set X and a set-distance matrix $\mathcal{D}$ of distances on X, the algorithm NetworkUPGMA applied to input X and $\mathcal{D}$ works by recursively finding

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

Lemma 2. *Let $\mathcal{N}$ be an ultrametric tree-child network on X, and let $\mathcal{D}$ be the set-distance matrix of inter-taxa distances of $\mathcal{N}$. A pair of leaves x, y form a cherry or reticulated cherry if and only if there is a leaf z such that*

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

Moreover such a pair x, y:

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

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

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

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

Now we are in a position to present NetworkUPGMA formally. The main body of the NetworkUPGMA algorithm looks for a pair $\{x, y\}$ which form a cherry or reticulated cherry. If such a pair is found, the algorithm forms a set of elements X' and a set-distance matrix $\mathcal{D}'$ resulting from reducing this cherry or reticulated cherry. It then makes a recursive call to NetworkUPGMA$(X', \mathcal{D}')$. If this yields a suitable network $\mathcal{N}'$ displaying

$\mathcal{D}'$ then a subroutine ReverseReduction is called, which reconstructs $\mathcal{N}$ by reversing the cherry reduction on $\mathcal{N}'$. Finally we need to check that the resulting network does display $\mathcal{D}$ before returning the network $\mathcal{N}$ (see Fig. 6 in Section 4 for an example illustrating why).

The pseudocode of NetworkUPGMA is given in Algorithm 1, and the pseudocode of the subroutine ReverseReduction is given in Algorithm 2.

4. Proof that NetworkUPGMA is correct

The following lemmas establish that the various steps in the algorithm work and can be accomplished in polynomial time. The first lemma, from [12], shows that every tree-child network contains either a cherry or reticulated cherry. After that, we present the proof of Lemma 2, which shows that we can recognise a cherry, immediate reticulation or reticulated cherry in an ultrametric tree-child network. Then we present lemmas showing that we can modify the set-distance matrix appropriately to effect an appropriate reduction in each case, and that we can also reverse the reduction once we have a network displaying the reduced set-distance matrix.

Lemma 3. *[12] Let $\mathcal{N}$ be a tree-child network on X. If $|X| \geq 2$, then $\mathcal{N}$ contains either a cherry or a reticulated cherry.*

The above lemma establishes that every tree-child network contains either a cherry or reticulated cherry; Lemma 2 stated that we can identify a pair of leaves involved in a cherry or reticulated cherry (with or without immediate reticulation), and moreover which of the cases it is. We now present the proof of that lemma.

Proof (Proof of Lemma 2). If $\{x, y\}$ do form a cherry, immediate reticulation or reticulated cherry, then it is easy to verify that the claimed conditions do hold. We therefore concentrate on proving that if the stated conditions hold, then $\{x, y\}$ must indeed be a cherry/immediate reticulation/reticulated cherry.

Let vertex v_1 be a least common ancestor of leaves x and y such that v_1 is at minimal distance from the root. Suppose there is a descendant leaf z

Algorithm 1 NetworkUPGMA

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

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

if $|X| = 1$ **then**
  return $\mathcal{N}$: a single vertex labelled with the element of X
else if $|X| = 2$ **and** $|\mathcal{D}_{x,y}| = 1$ **then**
  return $\mathcal{N}$: a cherry on two leaves with both arcs of weight $d_{x,y}/2$
else if $|X| = 2$ **and** $|\mathcal{D}_{x,y}| = 2$ **then**
  let $\{x, y\} = X$ and $\{d_1, d_2\} = \mathcal{D}_{x,y}$ such that $d_1 < d_2$
  return $\mathcal{N}$ on two leaves $\{x, y\}$ as given in Fig. 3
else if $|X| = 2$ **and** $|\mathcal{D}_{x,y}| > 2$ **then**
  return "Network not found"
  if there is a pair $x, y \in X$ such that $\{x, y\}$ forms a cherry **then**
    $X' = (X - \{x, y\}) \cup \{\xi\}$, where $\xi \notin X$
    ▷ Create the set-distance matrix $\mathcal{D}'$ on X' as follows:
    $\mathcal{D}'_{v,w} = \mathcal{D}_{v,w}$ if $v, w \in X - \{x, y\}$
    $\mathcal{D}'_{\xi,v} = \mathcal{D}'_{v,\xi} = \mathcal{D}_{x,v}$ if $v \in X - \{x, y\}$.
  else if there is a pair $x, y \in X$ such that $\{x, y\}$ forms a reticulated cherry with an immediate reticulation **then**
    $X' = X$
    ▷ Create the set-distance matrix $\mathcal{D}'$ on X' as follows:
    $\mathcal{D}'_{x,y} = \{d_{x,y}\}$
    $\mathcal{D}'_{v,w} = \mathcal{D}'_{v,w}$ for all pairs $\{v, w\} \neq \{x, y\}$.
  else if there is a pair $x, y \in X$ such that $\{x, y\}$ forms a reticulated cherry with y the reticulation leaf **then**
    $X' = X$
    ▷ Create the set-distance matrix $\mathcal{D}'$ on X' as follows:
    for all $v \in X - \{x, y\}$ **do**
      let $\{d_1, d_2, ..., d_k\} = \mathcal{D}_{x,v}$ and $\{d'_1, d'_2, ..., d'_l\} = \mathcal{D}_{y,v} - \mathcal{D}_{x,v}$
      $\mathcal{D}'_{v,w} = \mathcal{D}_{v,w}$ if $v, w \in X - \{y\}$
      $\mathcal{D}'_{y,v} = \mathcal{D}'_{v,y} = \mathcal{D}_{x,v}$ if $v \in X - \{y\}$
      $\mathcal{D}'_{x,y} = \mathcal{D}'_{y,x} = d_{x,y}$.
    end for
  else
    return "Network not found".
  end if
end if

Algorithm 1 NetworkUPGMA (continued)

```
30: if NetworkUPGMA(X', D') == "Network not found" then
31:     return "Network not found"
32: else
33:     let N' = NetworkUPGMA(X', D')
34:     let N = ReverseReduction(D, N', X', D', x, y).
35:     if N displays D then
36:         return N
37:     else
38:         return "Network not found"
39:     end if
40: end if
```

of v_1 such that $z \notin \{x, y\}$ as shown in Fig. 4 (a). Then there is an up-down path from x to z that has peak either v_1 or a descendant of v_1. Thus by the ultrametric property, $d_{x,z} \leq 2d_{x,v_1} = d_{x,y}$. This contradicts the condition $d_{x,y} < d_{x,z} : \forall z \notin \{x, y\}$, thus we may conclude that if the condition holds, there are no descendant leaves of v_1 except x, y.

Suppose there is a tree vertex v_2 that is a descendant of v_1, and without loss of generality take v_2 to be a tree vertex at maximal distance from v_1. Since in a tree-child network every tree vertex has a tree-path to a leaf, the two children of v_2 must either be leaves or have tree-paths to leaves. They cannot have tree-paths to the same leaf, since it would require a reticulation where the paths meet. Since there are no descendant leaves of v_1 except x and y, it must be that one child of v_2 has a tree-path to x and the other a tree-path to y and there are no other descendant leaves of v_2. If v_2 is on the path from v_1 to x, then directed path from v_2 to y must join the path from v_1 to y at a reticulation w as shown in Fig. 4 (b). By the tree-child property, the child of w must be a tree vertex or leaf, and by v_2's maximality of distance from v_1, it must be a leaf, thus y. Again by the tree-child property, the other child of v_2 is a tree vertex or leaf, and by maximality of distance from v_1, it must be a leaf, therefore x. There can be no other tree vertex that is a descendant of v_1, as it could not have a tree-path to y since w is the parent of y; thus w is an immediate reticulation with parents v_1 and v_2. Since v_2 is a descendant of v_1, the paths with peaks v_1 and v_2 have different weights and so $|\mathcal{D}_{x,y}| = 2$. If v_2 is on the path from v_1 to y, then it gives the equivalent

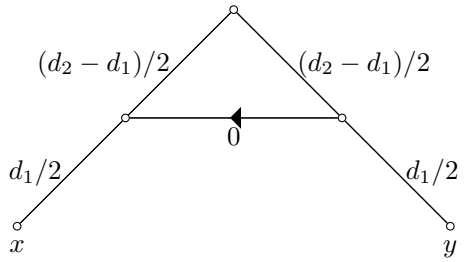

Figure 3: The unique (up to $\equiv$) ultrametric tree-child network on two leaves x, y, such that there are two distinct distances d_1, d_2 between the leaves. See Algorithm 1, Line 7.

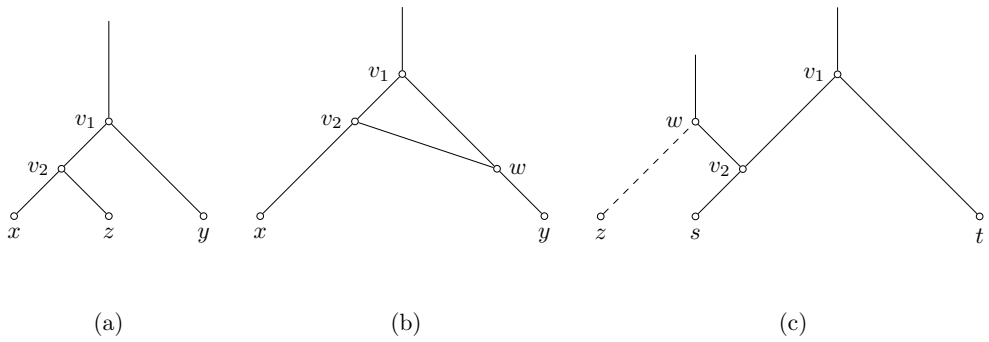

Figure 4: The situation when: (a) there is a descendant leaf z of v_1 such that $z \notin \{x, y\}$; (b) there is a tree vertex v_2 that is a descendant of v_1; (c) there is no tree vertex that is a descendant of v_1 but there is a reticulation vertex v_2 that is a child of v_1, also note that $\{s, t\} = \{x, y\}$.

network obtained by reversing the direction of the arc (v_2, w).

Suppose now that there is no tree vertex that is a descendant of v_1 but that there is a reticulation vertex v_2 that is a child of v_1. Observe that by the tree-child property applied to v_1, there can be reticulations only down the path to x or the path to y, not both, and by the tree-child property applied v_2, there can be no other reticulations that are descendants of v_1. Thus x and y must form a reticulated cherry. Let s denote the element of $\{x, y\}$ such that v_2 is on the path from v_1 to s. Let w be the other parent of v_2 (i.e. $w \neq v_1$), and let z be a leaf that is reached by a tree-path from w as shown in Fig. 4 (c). Observe that w cannot be a parent of v_1 because then

Algorithm 2 ReverseReduction

Input: A set-distance matrix $\mathcal{D}$ on a finite set X, and a set-distance matrix $\mathcal{D}'$ on a finite set X', a phylogenetic network $\mathcal{N}'$, a pair of leaves x, y

Output: Phylogenetic network $\mathcal{N}$, or **Network not found**

if $X' = (X - \{x, y\}) \cup \{\xi\}$ **and** $l_{(\xi', \xi)} > d_{x,y}/2$ where the parent of ξ is ξ' **then**
  ▷ Reversing a cherry reduction
  form $\mathcal{N}$ from $\mathcal{N}'$ by appending leaves x, y as a children of ξ
  set $l_{\mathcal{N}}(\xi, x) = l_{\mathcal{N}}(\xi, y) = d_{x,y}/2$
  set $l_{\mathcal{N}}(\xi', \xi) = l_{\mathcal{N}'}(\xi', \xi) - d_{x,y}/2$
  for all other edges e set $l_{\mathcal{N}}(e) = l_{\mathcal{N}'}(e)$
  return $\mathcal{N}$
else if $|\mathcal{D}_{x,y}| = 2$ **and** $\mathcal{D}_{x,z} = \mathcal{D}_{y,z}$ for all $z \in X - \{x, y\}$ **then**
  ▷ Reversing an immediate reticulated cherry reduction
  form $\mathcal{N}$ from $\mathcal{N}'$ as follows:
  let the common parent of x, y in $\mathcal{N}'$ be u, and its parent be u'
  subdivide the arc (u', u) with a new vertex v'
  subdivide the arc (u, y) with a new vertex v
  add an arc (v', v)
  let $d^*_{x,y} = \mathcal{D}_{x,y} - d_{x,y}$
  $l_{\mathcal{N}}(u, v) = 0$
  $l_{\mathcal{N}}(u, x) = l_{\mathcal{N}}(v, y) = d_{x,y}/2$
  $l_{\mathcal{N}}(v', v) = l_{\mathcal{N}}(v', u) = (d^*_{x,y} - d_{x,y})/2$
  $l_{\mathcal{N}}(u', v') = l_{\mathcal{N}'}(u', u) - l_{\mathcal{N}}(v', u)$
  for all other edges e set $l_{\mathcal{N}}(e) = l_{\mathcal{N}'}(e)$
  return $\mathcal{N}$

Algorithm 2 ReverseReduction (continued)

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

w would be a least common ancestor of x and y at shorter distance from the root than v_1. Then every path from s to z is either the (unique) path P that starts s, v_2, w, or is a path via v_1. Note also that any path via v_1 must also pass through w and therefore (by the ultrametric property) is longer than the path P. Thus $|\mathcal{D}(s,z)| = |\mathcal{D}(t,z)| - 1$, where t is the element in $\{x,y\} - \{s\}$.

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

The next lemmas establish the effect of reducing a cherry or a reticulated cherry on the set-distance matrices. Recall that two networks are equivalent under $\equiv$ if one can be obtained from the other by adjusting weights at reticulations and flipping the direction of immediate reticulations (see Section 2.3). We show that the reducing a cherry or reticulated cherry has a deterministic effect on the set-distance matrix, and moreover if, up to equivalence under $\equiv$, there is a unique ultrametric tree-child network that displays the reduced set-distance matrix, then there is, up to equivalence under $\equiv$, a unique ultrametric tree-child network that displays the original set-distance matrix.

Lemma 4. *Let $\mathcal{N}$ be an ultrametric tree-child network on $|X| > 2$. Let $\mathcal{D}$ be the set-distance matrix of inter-taxa distances of $\mathcal{N}$. Let $\{x,y\}$ be a cherry of $\mathcal{N}$ with common parent v, so that $d_{x,y} = 2 \times d_{v,x}$. Let $X' = (X - \{x,y\}) \cup \{\xi\}$ and $\mathcal{D}'$ be the set-distance matrix of inter-taxa distances on X' given by ${\mathcal{D}'}_{z,z'} = \mathcal{D}_{z,z'}$ if $z, z' \in X - \{x,y\}$, and ${\mathcal{D}'}_{z,\xi} = \mathcal{D}_{z,x}$ if $z \in X - \{x,y\}$. Then the following hold:*

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

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

Proof. Let w be the parent of v. We reduce the cherry by deleting leaf y and its incident edge, and suppressing the degree two vertex v and relabelling

the leaf x as ξ. Set the weight of the edge (w, ξ) to be $d_{w,x}$ to obtain $\mathcal{N}'$. Thus, for all $z, z' \in X - \{y\}$ the set of path distances between z and z' is unchanged by the reduction. Hence $\mathcal{D}'$ is displayed by the network $\mathcal{N}'$ on X'.

For (ii), suppose $\mathcal{N}'$ is the unique (up to $\equiv$) ultrametric tree-child network displaying $\mathcal{D}'$, and let $\mathcal{N}_1$ be an ultrametric tree-child network on X displaying $\mathcal{D}$. By Lemma 2, $\{x, y\}$ is a cherry in $\mathcal{N}_1$. Furthermore, by (i), the network ${\mathcal{N}'}_1$ on X obtained from $\mathcal{N}_1$ by reducing the cherry $\{x, y\}$ also displays $\mathcal{D}'$. Therefore, by the assumption in the statement of part (ii), ${\mathcal{N}'}_1 \equiv \mathcal{N}'$. Since the pair x, y are not involved in any reticulations, it follows that $\mathcal{N}_1 \equiv \mathcal{N}$. □

Lemma 5. *Let $\mathcal{N}$ be an ultrametric tree-child network, and let $\mathcal{D}$ be the set-distance matrix of inter-taxa distances of $\mathcal{N}$. Let $\{x, y\}$ be a reticulated cherry in which the reticulation vertex is an immediate reticulation, with v the reticulation, u the parent and sibling of v, and v' the parent of u and v (See Fig. 2(c)). Let $\mathcal{D}'$ be the set-distance matrix of inter-taxa distances on X given by*

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

and

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

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

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

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

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

For (ii), suppose $\mathcal{N}'$ is the unique (up to $\equiv$) ultrametric tree-child network displaying $\mathcal{D}'$, and let $\mathcal{N}_1$ be an ultrametric tree-child network on X displaying $\mathcal{D}$. By Lemma 2, $\{x, y\}$ is a reticulated cherry with immediate reticulation in $\mathcal{N}_1$. Furthermore, by (i), the network $\mathcal{N}'_1$ on X obtained from $\mathcal{N}_1$ by reducing the reticulated cherry $\{x, y\}$ also displays $\mathcal{D}'$. Therefore, by the assumption in the statement of (ii), $\mathcal{N}'_1 \equiv \mathcal{N}'$. Since each of these networks was formed by the removal of a single arc subdividing the incoming arcs to y and to its parent, it follows that $\mathcal{N}_1 \equiv \mathcal{N}$. □

Lemma 6. *Let $\mathcal{N}$ be an ultrametric tree-child network, and let $\mathcal{D}$ be the set-distance matrix of inter-taxa distances of $\mathcal{N}$. Let $\{x, y\}$ be a reticulated cherry of $\mathcal{N}$ with y the reticulation leaf, and not part of an immediate reticulation. Let $\mathcal{D}'$ be the set-distance matrix of inter-taxa distances on X given by $\mathcal{D}'_{x,y} = \{d_{x,y}\}$, $\mathcal{D}'_{y,z} = \mathcal{D}_{x,z}$ for $z \in X - \{x, y\}$ and $\mathcal{D}'_{z,z'} = \mathcal{D}_{z,z'}$ for $z, z' \in X - \{y\}$. Then the following hold:*

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

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

Proof. Let u be the parent of x and v be the parent of y in $\mathcal{N}$, as shown in Fig. 5. Since u is a tree vertex, it has a unique parent u'. Since v is a reticulation vertex, it has a parent v' additional to u, and v' has a tree-path to a leaf z. The reduction of the reticulated cherry involves removing the arc (v', v) and suppressing the resulting degree 2 vertices v and v'. Intuitively, we delete v and v', and their incident arcs, and introduce arcs (u, y) and (b, a), where b is the parent of v', and a is the other child of v'.

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

Now consider the up-down paths between y and $z \neq x$ in $\mathcal{N}$. Every up-down path between y and z does exactly one of the following: either passes

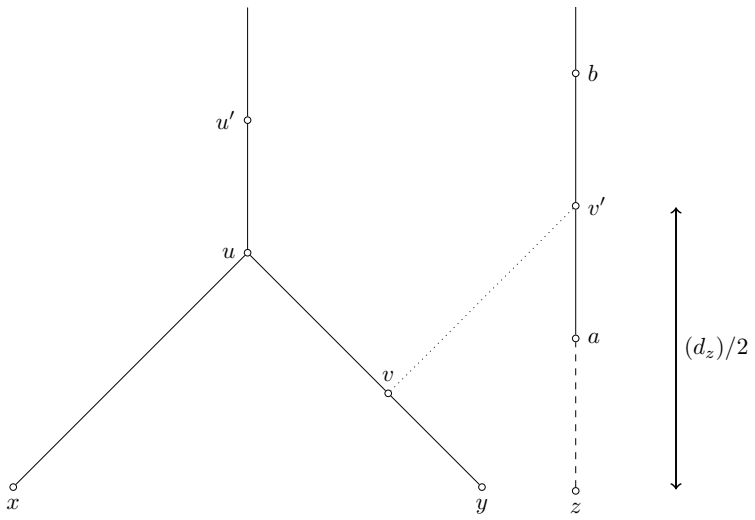

Figure 5: A reticulated cherry $\{x, y\}$. Reducing the reticulated cherry involves deleting the arc (v', v), shown with dotted lines, and suppressing the degree-2 vertices v', v. Since there is a tree-path in $\mathcal{N}$ from v' to a leaf z, there is an additional distance d_z in $\mathcal{D}_{y,z}$ that is not in $\mathcal{D}_{x,z}$.

through arc (v', v), in which case the path is not in $\mathcal{N}'$, or it passes through arc (u', u) in which case the weight of the path is equal to some $d \in d_{x,z}$. Moreover, for any path from x to z, there is an equal weight path from y to z because $d_{u,x} = d_{u,y}$. This concludes the proof of part (i).

For the proof of (ii), let $\mathcal{N}_1$ be some ultrametric tree-child network that displays $\mathcal{D}$. Then by Lemma 2, $\{x, y\}$ form a reticulated cherry in $\mathcal{N}_1$ where there is no immediate reticulation and y is the reticulation leaf. Thus if we reduce this reticulated cherry we will obtain a network $\mathcal{N}_1'$ which displays $\mathcal{D}'$ and is therefore equivalent to $\mathcal{N}'$. Observe that $\mathcal{N}_1'$ differs from $\mathcal{N}_1$ by the removal of a single arc (v_1', v_1) whose head v_1 was a parent of y. We now show that there is only one possible position in $\mathcal{N}_1'$ for the tail v_1' of this arc to have been in order for the matrix $\mathcal{D}$ to be displayed by $\mathcal{N}'$.

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

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

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

Lemma 7. *Let $\mathcal{N}$ be an ultrametric tree-child network displaying set-distance matrix $\mathcal{D}$, such that leaves x, y form a cherry or reticulated cherry in $\mathcal{N}$. Let $\mathcal{D}'$ and X' be as formed by lines 10-25 of* NetworkUPGMA*, corresponding to reducing the cherry or reticulated cherry $\{x, y\}$, and let $\mathcal{N}'$ be an ultrametric tree-child network displaying $\mathcal{D}'$. Then Algorithm* ReverseReduction *applied to $\mathcal{D}, X, \mathcal{D}', X', \mathcal{N}', \{x, y\}$ returns a network equivalent to $\mathcal{N}$ under $\equiv$.*

Proof. First suppose that x, y form a cherry in $\mathcal{N}$. Then by Lemma 2 and Lemma 4, $\mathcal{N}'$ is equivalent to a network obtained by reducing the cherry $\{x, y\}$ in $\mathcal{N}$. Hence $|X'| = |X| - 1$, and so lines 2-6 are executed. By construction the arc (ξ', ξ) in $\mathcal{N}'$ has weight greater than $d_{x,y}/2$, and lines

2-5 of ReverseReduction correctly reconstruct a network $\mathcal{N}_1 \equiv \mathcal{N}$ by Lemma 4.

Secondly suppose that x, y form a reticulated cherry with immediate reticulation in $\mathcal{N}$. Then by Lemma 2 and Lemma 5, $\mathcal{N}'$ is equivalent to a network obtained by reducing the reticulated cherry $\{x, y\}$ in $\mathcal{N}$. Thus $X' = X$ and $\{x, y\}$ form a cherry in $\mathcal{N}'$. Therefore, by Lemma 2, lines 9-19 of ReverseReduction are executed, and the resulting network displays $\mathcal{D}$. So by Lemma 5 the reconstructed $\mathcal{N}_1 \equiv \mathcal{N}$.

Thirdly suppose that x, y form a reticulated cherry without immediate reticulation in $\mathcal{N}$. Then by Lemma 2 and Lemma 6, $\mathcal{N}'$ is equivalent to a network obtained by reducing the reticulated cherry $\{x, y\}$ in $\mathcal{N}$. Thus $X' = X$ and $\{x, y\}$ form a cherry in $\mathcal{N}'$. Therefore, by Lemma 2, lines 21-39 of ReverseReduction are executed. Since $\mathcal{N}'$ is equivalent to the network $\mathcal{N}''$ obtained by reducing the reticulated cherry $\{x, y\}$ in $\mathcal{N}$, then the set Z at line 21 of ReverseReduction (that is the set of leaves z which have a single extra distance in $\mathcal{D}_{y,z}$ that is not present in $\mathcal{D}'_{y,z}$) is non-empty, and moreover for at least one $z \in Z$ the arc (a, b) exists (since in $\mathcal{N}$ there is a tree-path to a leaf z from the vertex v' at the tail of the deleted arc). The question remains of whether we can detect the arc given $\mathcal{N}'$ instead of $\mathcal{N}''$. There are two reasons for possible failure. First, the arc (a, b) satisfies $d_{b,z} < d^*_{y,z}/2 < d_{a,z}$ in $\mathcal{N}''$ but not in $\mathcal{N}'$, which may occur if a is a reticulation and under $\equiv_1$ the weights of the arcs into and out of a have been adjusted. However we can easily determine the class representative for $\mathcal{N}'$ under $\equiv_1$, which will satisfy this condition if $\mathcal{N}''$ does, and use that in place of $\mathcal{N}'$. Second, the vertex b may be a reticulation in $\mathcal{N}'$ but not in $\mathcal{N}''$ if it is an immediate reticulation that has been created by reversing an arc in $\mathcal{N}''$ under $\equiv_2$. However we can again easily identify when this occurs, and reverse the incoming arc to b that is not (a, b) in the case that the only reticulation on the path b to z is an immediate reticulation at b. Once such a z is found, there can be only one place to insert an arc to reverse the reduction, by Lemma 6, and so we correctly reconstruct a network $\mathcal{N}_1 \equiv \mathcal{N}$. □

Theorem 8. *Algorithm* NetworkUPGMA *is correct. Moreover, if Algorithm* NetworkUPGMA *returns a network $\mathcal{N}$ on input $\mathcal{D}$ then $\mathcal{N}$ is, up to $\equiv$, the unique ultrametric tree-child network displaying $\mathcal{D}$.*

Proof. The proof is by induction on $|X|$. It is straightforward to verify that if $|X| \leq 2$ then NetworkUPGMA takes the correct action. Assume

now that $|X| > 2$ and the algorithm is correct on inputs with fewer than $|X|$ leaves. By Lemma 2, we can determine if one of the three cases in lines 10, 14, 18 applies, and by Lemmas 4, 5, and 6, assuming there is a network that displays $\mathcal{D}$, then the correct $\mathcal{D}', X'$ are created corresponding to a network after the appropriate reduction. By Lemma 3, if there is a tree-child network displaying $\mathcal{D}$, then it contains a cherry or reticulated cherry, so if none is found then we are correct to return "Network not found" in line 27 of NetworkUPGMA algorithm.

Again by Lemmas 4, 5, and 6, assuming there is an ultrametric tree-child network that displays $\mathcal{D}$, then the recursive call in line 30 would return a valid network, so we are correct to return "Network not found" if the recursive call does not return a network. Finally, in the case that a network $\mathcal{N}'$ is returned, we call ReverseReduction. If there is a network that displays $\mathcal{D}$, then by Lemma 7, we reconstruct a valid network displaying $\mathcal{D}$ from $\mathcal{N}'$, and hence return a correct answer. If there is not an ultrametric tree-child network that displays $\mathcal{D}$, then the check in line 38 fails and we correctly return "Network not found". Hence in all cases NetworkUPGMA is correct.

Finally observe that when NetworkUPGMA returns a network, it is built up from a network on one or two leaves by successively reversing reductions. Since there is a unique possible network for each case when $|X| = 1$ or $|X| = 2$, and by Lemmas 4, 5, and 6, each reduction reversal results in a unique network (up to $\equiv$), it must be that $\mathcal{N}$ is also unique up to $\equiv$. □

Note that the final check that the network displays $\mathcal{D}$ is required. Fig. 6 gives an example of where given input corresponding to a non-ultrametric phylogenetic tree the algorithm would correctly identify and reduce a cherry, reconstruct a network $\mathcal{N}'$ displaying $\mathcal{D}'$, and then reverse the reduction, but there is no valid ultrametric network that displays $\mathcal{D}$. This example also serves to illustrate the extent to which NetworkUPGMA generalises UPGMA. Here we note that UPGMA takes as input a matrix of distances, whereas NetworkUPGMA takes a matrix of sets of distances, however when there is a unique distance between each pair of taxa, we ignore the distinction between the set containing the distance and the distance itself. Given a set-distance matrix with each set of size 1 that corresponds to an ultrametic tree, then both UPGMA and NetworkUPGMA will return the same correct tree, by Theorem 8. However, given data that is a set-distance matrix with each set of size 1 that does not correspond to an ultrametric tree, our algorithm will halt with "Network not found", whereas UPGMA

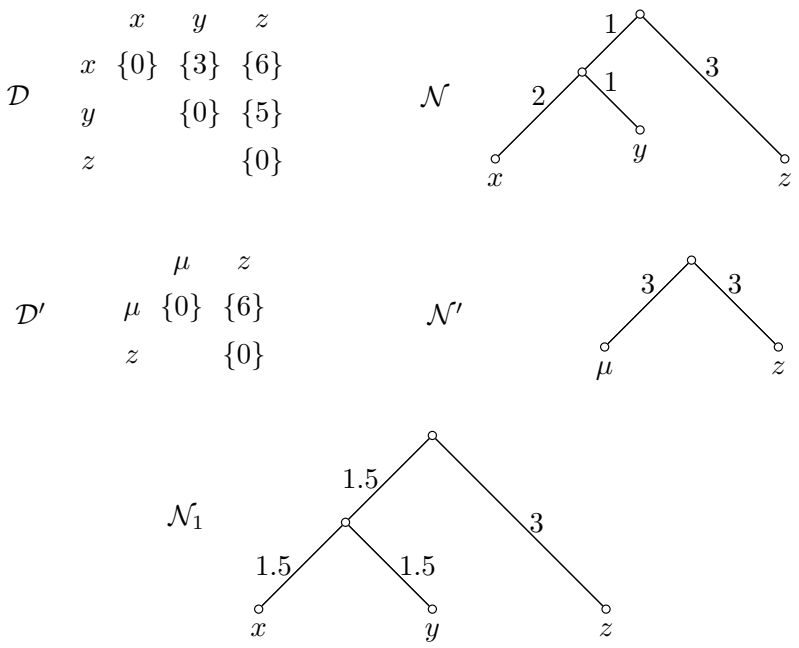

Figure 6: An example of an input $\mathcal{D}$ corresponding to a non-ultrametric phylogenetic tree $\mathcal{N}$, the reduced set-distance matrix $\mathcal{D}'$ and corresponding tree $\mathcal{N}'$, and the ultrametric tree $\mathcal{N}_1$ created in the algorithm by reversing the reduction, which is then rejected in the test at line 38 of NetworkUPGMA.

will output a phylogenetic tree that may or may not be close to displaying the distances in the data, but will not display them exactly.

5. Running Time of NetworkUPGMA Algorithm

In this section, we analyse the running time of NetworkUPGMA. First we consider the size of the input. Typically in phylogenetic algorithms the running time is given in terms of $|X|$, the number of taxa under consideration. Here, the actual input is a set X and a $|X|$ by $|X|$ set-distance matrix $\mathcal{D}$ of inter-taxa distances on X. For all $x, y \in X$, we will assume that each entry $\mathcal{D}_{x,y}$ is presented as a sorted list of distances. The size of each set $\mathcal{D}_{x,y}$ is linear in $|X|$, as the ultrametric condition means that the weight of any up-down path between x and y is twice the distance from x to the peak of the up-down path, and there are less than $3|X|$ internal vertices in $\mathcal{N}$ that

could be the peak since $\mathcal{N}$ is tree-child (see [13], Proposition 1). Thus the input (essentially $\mathcal{D}$) has size $O(|X|^3)$, at least when the input is displayed by some tree-child network; excessively large inputs could be rejected out of hand before running the algorithm if need be.

Theorem 9. *Given a set X and a set-distance matrix $\mathcal{D}$, the algorithm* NetworkUPGMA *runs in time $O(|X|^4)$.*

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

Finally we consider algorithm NetworkUPGMA. Lines 1-7 deal with constant sized X and can be accomplished in constant time. Determining which case to undertake in the next **if** statement (lines 10, 14, 26) can be done in time $O(|X|^3)$, by applying Lemma 2. Also creating X' and $\mathcal{D}'$ take at most $O(|X|^3)$ steps. The call to ReverseReduction is also at most $O(|X|^3)$, and finally checking whether $\mathcal{D}$ is displayed by $\mathcal{N}$ is $O(|X|^3)$, since we need only check for each internal vertex of $\mathcal{N}$ whether it is an ancestor of each leaf, and its height, in order to determine the set-distance matrix displayed by $\mathcal{N}$. Thus the work done in NetworkUPGMA outside of the recursive call, takes at most $O(|X|^3)$ steps. In each recursive call $\mathcal{D}$ has strictly smaller size, thus the whole algorithm takes at most $O(|X|^6)$ steps. However we can do better than this. Any reduction of a reticulated cherry results in x, y being a cherry in $\mathcal{N}'$, so in fact every other reduction (at least) on an input that is displayed by a tree-child network reduces the size of $|X|$ by one. Thus there are only $2|X|$ recursions, so the entire algorithm completes in $O(|X|^4)$ steps. An additional check that we do not make two reticulated cherry reductions

in a row, else return "Network not found", would be needed in the algorithm to obtain this running time for all inputs. □

Combining Theorems 8 and 9 gives Theorem 1.

6. Conclusion

We have presented a generalisation NetworkUPGMA of the widely used ultrametric tree reconstruction algorithm UPGMA to ultrametric tree-child networks. This expands the class of weighted networks that can be directly reconstructed from inter-taxa distance information to include much more complex networks than galled trees or single reticulation networks. This work gives rise to the open problems of determining what accuracy guarantees can be given for the new algorithm, and testing its performance on real or simulated data.

A further interesting open problem is the following. Suppose that a set-distance matrix D on X is not displayed by any ultrametric tree-child network, then is it possible to determine the largest subset $Y \subset X$ such that there is an ultrametric tree-child network on Y where the set of distances between any $y, z \in Y$ is given by $D_{y,z}$? This maximisation problem is clearly harder than the simple decision problem of determining whether a given subset of X may be displayed on an ultrametric tree-child network, which could be answered by the algorithm in this paper.

Our algorithm suffers the same drawbacks as the original UPGMA algorithm does for trees: it relies on the assumption that the target network is ultrametric. It is clear than this assumption is not always valid, and this is part of the reason that Neighbor Joining (NJ) has proved to be an even more popular and robust method for reconstructing phylogenetic trees than UPGMA. It is the subject of ongoing research to see whether the ideas presented in this paper can be used to create a network generalisation of NJ.

Acknowledgment

The first author gratefully acknowledges the scholarship supplied to her from the Republic of Turkey, Ministry of National Education.

References

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

[2] N. Saitou, M. Nei, The neighbor-joining method: a new method for reconstructing phylogenetic trees, Mol. Biol. Evol. 4 (4) (1987) 406–425.

[3] W. M. Fitch, E. Margoliash, et al., Construction of phylogenetic trees, Science 155 (3760) (1967) 279–284.

[4] O. Gascuel, BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data., Mol. Biol. Evol. 14 (7) (1997) 685–695.

[5] R. Desper, O. Gascuel, Theoretical foundation of the balanced minimum evolution method of phylogenetic inference and its relationship to weighted least-squares tree fitting, Mol. Biol. Evol. 21 (3) (2004) 587–598.

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

[7] M. Bordewich, R. Mihaescu, Accuracy guarantees for phylogeny reconstruction algorithms based on balanced minimum evolution, IEEE/ACM Trans. Comput. Biol. Bioinform. 10 (3) (2013) 576–583.

[8] O. Gascuel, M. Steel, Neighbor-joining revealed, Mol. Biol. Evol. 23 (11) (2006) 1997–2000.

[9] H. Chan, J. Jansson, T. Lam, S. Yiu, Reconstructing an ultrametric galled phylogenetic network from a distance matrix, J. Bioinform. Comput. Biol. 4 (4) (2006) 807–832.

[10] S. J. Willson, Tree-average distances on certain phylogenetic networks have their weights uniquely determined, Algorithms Mol. Biol. 7 (2012) 13.

[11] S. J. Willson, Reconstruction of certain phylogenetic networks from their tree-average distances, Bull. Math. Biol. 75 (10) (2013) 1840–1878.

[12] M. Bordewich, C. Semple, Determining phylogenetic networks from inter-taxa distances, Journal of Mathematical Biology (2015) 1–21. URL http://dx.doi.org/10.1007/s00285-015-0950-8

[13] G. Cardona, F. Rossello, G. Valiente, Comparison of tree-child phylogenetic networks, IEEE/ACM Trans. Comput. Biol. Bioinform. 6 (4) (2009) 552–569.

[14] C. Semple, M. A. Steel, Phylogenetics, Vol. 24, Oxford University Press, 2003.

[15] A. Apostolico, M. Comin, A. W. Dress, L. Parida, et al., Ultrametric networks: a new tool for phylogenetic analysis, Algorithms Mol. Biol. 8 (1) (2013) 1–10.