Journal of Bioinformatics and Computational Biology
Vol. 4, No. 4 (2006) 807–832

ICP Imperial College Press
www.icpress.co.uk

RECONSTRUCTING AN ULTRAMETRIC GALLED PHYLOGENETIC NETWORK FROM A DISTANCE MATRIX*

HO-LEUNG CHAN†, JESPER JANSSON‡, TAK-WAH LAM§ and SIU-MING YIU¶

†,§,¶*Department of Computer Science, The University of Hong Kong*
Pokfulam Road, Hong Kong, P. R. China
†*hlchan@cs.hku.hk*
§*twlam@cs.hku.hk*
¶*smyiu@cs.hku.hk*

‡*Department of Computer Science and Communication Engineering*
Kyushu University, 6-10-1 Hakozaki, Higashi-ku, Fukuoka, Japan
jj@tcslab.csce.kyushu-u.ac.jp

Received 20 February 2006
Revised 25 February 2006
Accepted 25 February 2006

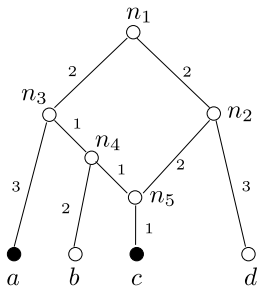
Given a distance matrix M that specifies the pairwise evolutionary distances between n species, the phylogenetic *tree* reconstruction problem asks for an edge-weighted phylogenetic tree that satisfies M, if one exists. We study some extensions of this problem to rooted phylogenetic *networks*. Our main result is an $O(n^2 \log n)$-time algorithm for determining whether there is an ultrametric galled network that satisfies M, and if so, constructing one. In fact, if such an ultrametric galled network exists, our algorithm is guaranteed to construct one containing the minimum possible number of nodes with more than one parent (*hybrid* nodes). We also prove that finding a largest possible submatrix M' of M such that there exists an ultrametric galled network that satisfies M' is NP-hard. Furthermore, we show that given an incomplete distance matrix (i.e. where some matrix entries are missing), it is also NP-hard to determine whether there exists an ultrametric galled network which satisfies it.

Keywords: Phylogenetic network; ultrametric galled network; distance-based reconstruction; algorithm.

1. Introduction

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 nontreelike, for example, due to recombination events such as hybrid speciation or horizontal gene

*A preliminary version of this paper has appeared in the Proceedings of the 30th International Symposium on Mathematical Foundations of Computer Science (MFCS 2005).
‡Supported in part by JSPS (Japan Society for the Promotion of Science)



M

	a	b	c	d
a	0	6	6	10
b		0	10	10
c			0	6
d				0

Fig. 1. The (galled and ultrametric) phylogenetic network on the left satisfies the distance matrix M on the right. There are two evolutionary paths (a, n_3, n_4, n_5, c) and $(a, n_3, n_1, n_2, n_5, c)$ with lengths 6 and 10, respectively, connecting a and c. The entry $M(a, c)$ corresponds to the first path. Note that there does not exist any phylogenetic tree that satisfies M.

transfer[14,22–24,27] or to represent several conflicting phylogenetic trees at once in order to identify parts where the trees disagree.[4,17,18]

To develop efficient methods for inferring phylogenetic networks is an important topic in computational biology. In particular, one promising category of methods which includes methods such as Neighbor-Net[4] and several others (see Ref. 24 for a survey) is known as *distance-based*. Here, the input consists of a (symmetric and nonnegative) distance matrix which specifies the pairwise evolutionary distances between the species. To infer a phylogenetic *tree* from such a matrix is a well-studied problem,[5,7,9,12,25,26,28] the basic objective being to construct an edge-weighted phylogenetic tree such that for any two species, the length of the path between them in the tree equals the corresponding entry in the matrix. It should be noted that in a phylogenetic tree, the path between two specified leaves is always unique. On the other hand, due to recombination events, for any two species in a phylogenetic network, there can be more than one path connecting them with different path lengths. The entry in the input matrix may correspond to one of these paths only. Hence, in some cases, there may exist a phylogenetic network that satisfies the given distance matrix (see the definition below), while no such phylogenetic tree exists (e.g. see Fig. 1). In this paper, we consider some natural extensions of the distance-based variant of the phylogenetic tree reconstruction problem to phylogenetic networks and present a new algorithm.

Problem definitions: A *rooted phylogenetic network* for a set S of species is a rooted, connected, directed acyclic graph such that: (i) exactly one node (the *root*) has indegree 0 and all other nodes have indegree 1 or 2; (ii) any node with indegree 2 (called a *hybrid node*) has outdegree 1 and all other nodes have outdegree 0 or 2; and (iii) each node with outdegree 0 (a *leaf*) is labeled with a distinct species from S. A rooted phylogenetic network is called a *galled phylogenetic network*, or *galled network* for short,[a] if all cycles in the underlying undirected graph (i.e. where edge

[a]Galled networks are also known in the literature as *topologies with independent recombination events*,[27] *galled-trees*,[14] *gt-networks*,[23] and *level*-1 *phylogenetic networks*.[6,21]

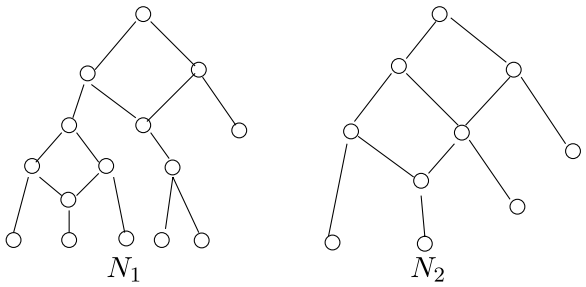


Fig. 2. N_1 is a galled network, while N_2 is not. (The leaf labels have been omitted for clarity.)

orientations are ignored) are node-disjoint. For example, the phylogenetic network in Fig. 1 and the network N_1 in Fig. 2 are galled networks. From here on, we only consider phylogenetic networks that are *edge-weighted*, i.e. where each edge has a positive length. In analogy with the standard usage of the term "ultrametric" for phylogenetic trees, we say that a galled network is *ultrametric*, if every directed path from the root to a leaf has the same length.

For any rooted phylogenetic network N, an *evolutionary path* between two leaves a and b is a simple path which goes up (i.e. moving in a child-to-parent direction) from a to a common ancestor u of a and b, and then down (i.e. moving in a parent-to-child direction) from u to b. It is observed that even if N is galled and ultrametric, there can be more than one evolutionary path between a and b, and moreover, these paths may have different lengths (again, see Fig. 1). However, in an ultrametric galled network, there can exist at most two different evolutionary path lengths between each pair of leaves, since each pair of leaves has at most two different lowest common ancestors in N.

A *distance matrix* for a set S of n species is a symmetric, nonnegative $(n \times n)$-matrix M such that $M(a, a) = 0$ for every $a \in S$. Intuitively, for each $a, b \in S$, $M(a, b)$ contains the measured evolutionary distance between a and b. A rooted phylogenetic network N for S *satisfies* M if, for every $a, b \in S$, it holds that N contains an evolutionary path between a and b of length equal to $M(a, b)$. In this case, we also say that M is *satisfied by* N.
We are now ready to define the problem which is the main focus of this paper.

Problem statement: Given a distance matrix M for a set S of n species, return an ultrametric galled network for S satisfying M, if one exists; otherwise, return *fail*.

Motivation: The rationale behind the way we define the problem is as follows. There are a number of methods to estimate the evolutionary distance between two species. One common approach is to align the DNA sequences for some related genes from the species. The alignment score usually provides a reasonable estimation on the evolutionary distance between the species. However, if recombination events had occurred, there might exist more than one common ancestor (at different evolutionary distances) for a pair of species. Thus, depending on which common

ancestor the selected genes were inherited from, the measured evolutionary distance may reflect only one of the possible evolutionary paths. Therefore, for any two species in the phylogenetic network, we only require one of their evolutionary paths to satisfy the matrix entry.

If there are no restrictions on the topological structure of the constructed phylogenetic network, it may not make sense from a biological point of view. Therefore, we concentrate on galled networks, a very useful class of rooted phylogenetic networks which despite their simple structure are powerful enough to describe evolutionary history when the frequency of recombination events is moderate or when most of the recombination events have occurred recently.[14] See Ref. 14 for a discussion on the importance of galled networks. Finally, the biological meaning of the ultrametric assumption is that the species have evolved according to a constant rate; see, e.g. Refs. 5, 9, 12, 26, 28 and the references therein for justification of this assumption.

Our contributions: Our main result in this paper is an $O(n^2 \log n)$-time exact algorithm to determine if there exists an ultrametric galled network satisfying a given distance matrix M, and constructing such a network if one exists. When a solution exists, our algorithm always outputs one having as few hybrid nodes as possible. In practical studies, it is desirable to find the simplest explanation that is consistent with the observed distances. Thus, although recombination events (corresponding to hybrid nodes) may occur, it is crucial to find a satisfying network containing the minimum number of hybrid nodes. On the other hand, given a matrix M with no exact solution, we prove that finding a largest possible submatrix M' of M such that there exists an ultrametric galled network that satisfies M' is an NP-hard problem. We also show that given an incomplete distance matrix (i.e. where some matrix entries are missing), it is NP-hard to determine whether there exists an ultrametric galled network which satisfies it.

Related work: In the context of reconstructing a phylogenetic network from distance data, the most related work is the *Neighbor-Net* method, developed by Bryant and Moulton,[4] which outputs a planar, unrooted phylogenetic network from a given distance matrix. Neighbor-Net is based on the well-known Neighbor-Joining method for trees.[25] Earlier proposed distance-based methods for reconstructing phylogenetic networks include those described in Ref. 8 and others described in Ref. 24. However, all of these approaches are heuristics-based and there is no guarantee that the output is a phylogenetic network that satisfies the given matrix exactly, even when a galled network exists. Also, Neighbor-Net runs in $O(n^3)$ time, which is slower than the method we present here.

Some other models of computation for reconstructing phylogenetic networks (i.e. assuming other types of input) are reviewed in Ref. 24. Recently, in addition to distance-based methods, researchers have also studied *character-based* and

supertree-based methods for inferring phylogenetic networks. In character-based methods, each species is represented by a set of characters obtained from, e.g. the DNA sequences or morphological data. These methods work under a parsimony framework and try to construct a network with the minimum number of evolutionary events; related work includes those described in Refs. 13, 14, 27. Supertree-based methods assume that information concerning the evolutionary relationships for some subsets of the species set is available (usually represented as a set of trees) and then try to merge this input into a phylogenetic network; examples can be found in Refs. 15, 17–20, 23.

To reconstruct a phylogenetic *tree* with n species consistent with a given distance matrix (if one exists), can easily be done in $O(n^2)$ time (see Refs. 9, 12, 26). However, when an exact solution does not exist, obtaining a tree that is as "close" as possible to the matrix has been shown to be NP-hard on several closeness metrics.[5,7,9,28]

Organization of the paper: In Sec. 2, we first introduce some additional terminology that is used throughout the paper to describe our techniques and results, and then examine some fundamental structural properties of all valid solutions. In Sec. 3, we present our efficient algorithm (whose design is based on the findings of Sec. 2) for determining if there exists an ultrametric galled network satisfying M, and if so, constructing such a network. Next, we prove in Sec. 4 that the two related problems mentioned above are NP-hard. Finally, we discuss an implementation of our algorithm in Sec. 5. In Sec. 5, we also discuss how to extend our basic algorithm to nonultrametric inputs.

2. Preliminaries

2.1. *Terminology*

Let N be a galled network. It should be remembered that a node h in N is called a *hybrid node*, if the indegree of h is equal to 2. Let s be an ancestor of h such that there are two edge-disjoint paths from s to h. Then s is called *the split node of* h. In a galled network, each split node is a split node of exactly one hybrid node, and each hybrid node has exactly one split node (see Lemma 1 in Ref. 21). The two paths from s to h are *the merge paths of* h, and they form a *galled loop* rooted at s. The galled loop rooted at s is *skew*, if one of its two merge paths consists of a single edge from s to h; otherwise, it is *nonskew*. Nodes other than h and s on the merge paths of h are called *side nodes*, and a node is called a *tree node*, if it is not on any galled loop. For any node u in N, the *subnetwork* rooted at u is the minimal subgraph of N including all nodes and directed edges reachable from u, and is denoted by N_u. Finally, N_u is a *side network*, if the parent of u belongs to a merge path P in N, but u itself is not on P.

For any internal node u of an ultrametric galled network N, every directed path from u to a leaf under u has the same length. We call this length the *height* of u and denote it by $height(u)$. For any leaf a, $height(a) = 0$. It should be noted that

the length of any edge (a, b) can be calculated from $height(a)$ and $height(b)$. Thus, to find a network for M, we only need to determine the heights of all internal nodes and the parent–child relations between nodes.

2.2. *Basic structural observations*

In any galled network, the smallest possible galled loop is skew and consists of exactly three nodes (a split node, a hybrid node, and a side node). By simple induction, one can prove that a galled network with n leaves contains at most $3n - 3$ internal nodes. This property is useful to our algorithm.

Lemma 1. *Let N be a galled network with n leaves. There are at most $3n - 3$ internal nodes in N.*

We now derive some properties of any ultrametric galled network satisfying the given distance matrix M. For simplicity, we say that M is *satisfiable*, if there exists an ultrametric galled network which satisfies it, and we refer to an ultrametric galled network as a *network*. Also, for any $S' \subseteq S$, if a network N for S' satisfies the submatrix of M induced by the species in S', then we say that N satisfies S'.

Consider any two species a and b in S. To satisfy M, the network must contain an evolutionary path between a and b with length equal to $M(a, b)$. We notice that this path starts from a, goes up to a common ancestor of height $M(a, b)/2$, and then goes down to b. Let D_S be the maximum distance between two species in S as specified by M. If M is satisfiable, then there is a network satisfying M whose root has height $D_S/2$.

We have the following observation about the internal nodes of N.

Observation 1. Assume that M can be satisfied by a network N. For any node u that is a tree node or a split node, let N_u be the subnetwork rooted at u, and let S_u be the set of species in N_u.

- For any two species $a, b \in S_u$, $M(a, b) = 2 \times height(v)$ for some internal node v in N_u, and hence $M(a, b) \leq 2 \times height(u)$.
- For any species $a \in S_u$ and $c \in S - S_u$, $M(a, c) > 2 \times height(u)$.

Observation 1 motivates the following definition.

Definition 1. For any set of species $S' \subseteq S$, S' is called a *cluster*, if there exists a value x such that for any two species $a, b \in S'$, $M(a, b) \leq x$ and for any species $a \in S'$ and $c \in S - S'$, $M(a, c) > x$.

S itself is the biggest cluster. It should be noted that clusters are nested, i.e. two clusters are always either disjoint or one is a subset of the other. Observation 1 states that each tree node or split node in N induces a cluster. In fact, the reverse is also true.

Lemma 2. *Assume that M can be satisfied by some network. Then there exists such a network N' in which, for every cluster $S' \subseteq S$, N' has a tree node or a split node u such that all species in S' are in the subnetwork N'_u, and no species in $S - S'$ are in N'_u.*

A detailed proof of Lemma 2 is given in Sec. 2.3. To prove the lemma, we let N be any network satisfying M. If N does not satisfy the conditions in Lemma 2, we prove that we can always modify N to obtain another network N' which does, and moreover, that N' has the same number of hybrid nodes as N.

We call a network satisfying the conditions in Lemma 2 a *well-structured* network. Well-structured networks have the following very nice property. Consider any $S' \subseteq S$ that is a cluster, and let $S_1, S_2, \ldots, S_t$ be all the maximal clusters which are proper subsets of S' (note that $S' = S_1 \cup S_2 \cup \cdots \cup S_t$). We call $S_1, S_2, \ldots, S_t$ the *side clusters* of S'. Then:

Lemma 3. *Let S' be a cluster with side clusters $S_1, \ldots, S_t$. Let N' be any well-structured network satisfying S' (w.r.t. the submatrix of M induced by S'). N' consists of a root node u, with the networks satisfying $S_1, \ldots, S_t$ attached to u, or attached to a galled loop rooted at u.*

Proof. Since N' is well-structured, for each side cluster S_i, there is a tree node or a split node v whose subnetwork contains all and only species in S_i. We notice that on the path from v to u, there is no tree node or split node other than u and v (otherwise, let w be that intermediate node; the species under the subnetwork rooted at w forms a cluster S'' and $S_i \subsetneq S'' \subsetneq S'$, meaning that S_i is not a side cluster of S'). Thus, v is directly attached to u or a galled loop rooted at u. It means that N' is formed by attaching the networks for $S_1, \ldots, S_t$ to u, or to a galled loop rooted at u. □

2.3. *Proof of Lemma 2*

This subsection proves Lemma 2. (Readers who are interested in our algorithm may skip ahead to the next section.) Let M be a matrix for a set S of n species, and let N be a network satisfying M. We say that a cluster $S' \subseteq S$ *occupies* a tree node or a split node u, if S' is the set of species in the subnetwork rooted at u. Lemma 2 states that if M can be satisfied by some network N, then there is one such network N' such that each cluster S' occupies a tree node or a split node.

It should be remembered that a network which meets the conditions in Lemma 2 is called a *well-structured* network. The rest of this subsection is devoted to the proof of the following lemma, which then immediately implies Lemma 2.

Lemma 4. *Let N be a network satisfying the matrix M. If N is not well-structured, we can modify N into a well-structured network N' satisfying M such that N' has the same number of hybrid nodes as N.*

As mentioned below, suppose that N is a network satisfying M and that N is not well-structured. Let $S' \subseteq S$ be a cluster such that S' does not occupy a tree node or split node in N, and let v be the node with minimum height such that the subnetwork rooted at v contains all species from S'.

Lemma 5. *v is a side node.*

Proof. We prove the lemma by a simple case analysis:

If v is a hybrid node then the child of v is a node with smaller height whose subnetwork contains all species from S', which contradicts the definition of v.

If v is a split node, consider the galled loop rooted at v. This galled loop must be non-skew (otherwise, the topmost side node on the galled loop has a smaller height than v but contains all species in S', which is a contradiction). Thus, there are two species $a, b \in S'$ where a (resp. b) is in some side network attached to a side node on the left (resp. right) merge path. The evolutionary path between a and b has length $2 \times \mathit{height}(v)$, meaning that $M(a, b) = 2 \times \mathit{height}(v)$ as N satisfies M. For any species c in the subnetwork rooted at v, the evolutionary path between a and c has length at most $2 \times \mathit{height}(v)$, so $M(a, c) \leq M(a, b)$ and c belongs to S'. Therefore, all species in the subnetwork rooted at v belong to S', contradicting that S' does not occupy a split node.

Similarly, a contradiction occurs, if v is a tree node.

Combining the three cases, we conclude that v must be a side node. □

Consider the galled loop containing v. Let s be the split node. Without loss of generality, let v be a side node on the left merge path. Let h be the hybrid node and let $v_1, v_2, \ldots, v_\ell$ be the nodes on the merge path from h to v, where v_1 is the node immediately next to h and $v_\ell = v$. For each $i \in \{1, 2, \ldots, \ell\}$, denote the side network attached to v_i by $S(v_i)$. Define $S(h)$ similarly.

Lemma 6. (i) *For $i \in \{1, 2, \ldots, \ell\}$, all species from $S(v_i)$ are in S'.* (ii) *Either all species from $S(h)$ are in S', or none of the species from $S(h)$ are in S'.*

Proof. First, observe that there are two species $a, b \in S'$, where a is in the subnetwork under the left child of v, and b is in the subnetwork under the right child of v. The evolutionary path between a and b has length at least $2 \times \mathit{height}(v)$, meaning that $M(a, b) \geq 2 \times \mathit{height}(v)$. It should be noted that for any species c, if $M(a, c) \leq M(a, b)$, then c is in S' due to the definition of a cluster.

(i) For any species c from $S(v_i)$, where $i \in \{1, 2, \ldots, \ell\}$, the evolutionary path between a and c has length at most $2 \times \mathit{height}(v)$. It means that $M(a, c) \leq M(a, b)$ and c is in S'. Thus, all species from $S(v_i)$, for all $i \in \{1, 2, \ldots, \ell\}$, are in S'.

(ii) If there is a species $d \in S(h)$ which belongs to S', let c be any other species from $S(h)$. Any evolutionary path between c and d has the highest node below the hybrid node h, so the length is less than $2 \times \mathit{height}(v)$. It means that $M(c, d) \leq M(a, b)$ and c is in S'. Therefore, in this case, all species from $S(h)$ are

in S'. Hence, either all species from $S(h)$ are in S' or none of the species from $S(h)$ are in S'. □

Lemma 7. *If none of the species from $S(h)$ are in S', then we can modify N to a new network N' such that N' satisfies S and S' occupies a tree node in N'. Furthermore, every cluster that occupies a tree node or a split node in N still occupies a tree node or a split node in N', and N' has the same number of hybrid nodes as N.*

Proof. Note that $S(v_1), S(v_2), \ldots, S(v_\ell)$ together contain exactly all species from S'. Also, $\ell \geq 2$ (otherwise, $S(v_1)$ contains exactly all species from S', and S' occupies the root of $S(v_1)$, which is a tree node or a split node). Since $S(h)$ does not contain any species from S', for any species $a \in S'$ and $c \in S(h)$, we have $M(a,c) > 2 \times height(v_\ell)$ and $M(a,c) = 2 \times height(s)$. See Fig. 3 for an illustration.

We modify N into N' by merging the side networks $S(v_1), S(v_2), \ldots, S(v_\ell)$ into a single side network as shown in Fig. 3. More precisely, let $v_{\ell+1}$ be the node on the left merge path immediately above v_ℓ (possibly with $v_{\ell+1} = s$). We first create a node w on the left merge path whose height equals $(height(v_\ell) + height(v_{\ell+1}))/2$. Next, we remove the node v_1 and all its incident edges. Finally, we insert an edge from w to h, and an edge from v_2 to the root of $S(v_1)$. Note that we do not change the heights of any existing nodes. For any two species x and y, if there is an evolutionary path between them with length α in N, then there is still an evolutionary path between them with length α in N', except for the case that $x \in S'$ and $y \in S(h)$. In that case, N' still satisfies the distance requirement of $M(x,y)$ as $M(x,y) = 2 \times height(s)$. Thus, N' satisfies S and S' occupies the tree

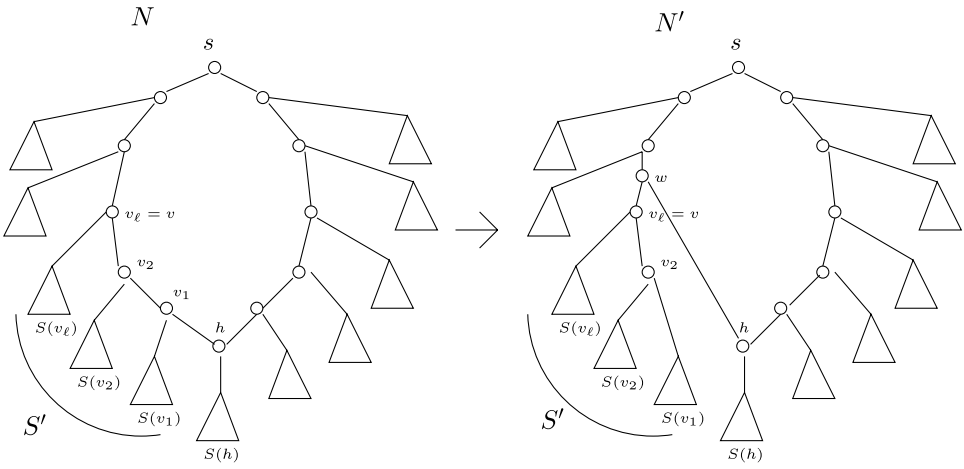


Fig. 3. If none of the species from $S(h)$ are in S', we can modify N to obtain another network N' which is well-structured.

node v_ℓ. Furthermore, it is straightforward to verify that if a cluster occupies a tree node or split node in N, the cluster still occupies the same node in N'. □

Lemma 8. *If all species from $S(h)$ are in S', then we can modify N to a new network N' such that N' satisfies S and S' occupies a tree node in N'. Furthermore, every cluster that occupies a tree node or split node in N still occupies a tree node or split node in N', and N' has the same number of hybrid nodes as N.*

Proof. Observe that $S(v_1), S(v_2), \ldots, S(v_\ell)$ and $S(h)$ together contain exactly all species from S'. Since all species from $S(h)$ are in the cluster S', for any species $a \in S(v_i)$, where, $i \in \{1, 2, \ldots, \ell\}$ and any species $b \in S(h)$, $M(a, b) \leq 2 \times height(v_\ell)$. Furthermore, there is an evolutionary path between a and b in N which does not pass through s and has length equal to $M(a, b)$. Let $u_1, u_2, \ldots, u_r$ be the nodes on the right merge path, where u_1 is the node immediately above h and $u_r = s$ (note that if the galled loop is skew, we have $u_1 = u_r = s$). There are two cases: (1) $height(v_\ell) < height(u_1)$; and (2) $height(v_\ell) \geq height(u_1)$.

(1) If $height(v_\ell) < height(u_1)$, we can modify N into N' as shown in Fig. 4. To be precise, first create a node w on the left merge path with height $(\min\{height(u_1), height(v_{\ell+1})\} + height(v_\ell))/2$, where $v_{\ell+1}$ is the node immediately above v_ℓ on the left merge path. Next, remove h and all its incident edges. Finally, insert an edge from u_1 to w, and an edge from v_1 to the root of $S(h)$. It should be noted that N' is a valid network. Furthermore, for any pair of species $x, y \in S$, by checking the cases of whether x, y are in S', it is easy to see that N' satisfies S, and moreover, S' occupies the tree node v_ℓ in N'. Also, every cluster

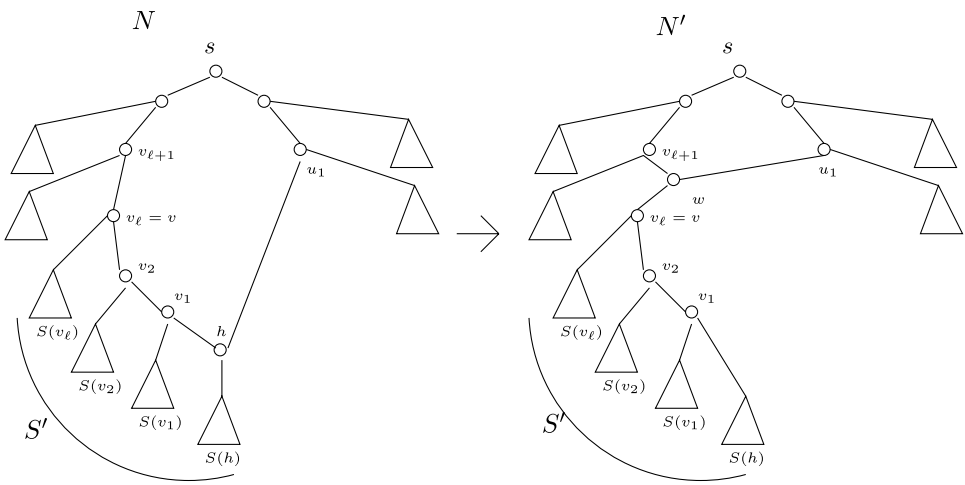


Fig. 4. If all species from $S(h)$ are in S' and $height(v_\ell) < height(u_1)$, we can modify N to obtain another network N' which is well-structured.

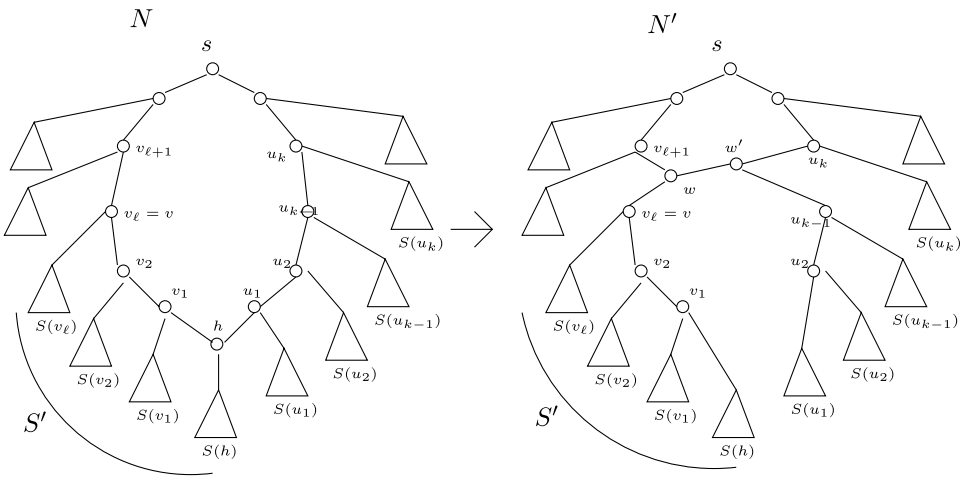


Fig. 5. If all species from $S(h)$ are in S' and $height(v_\ell) \geq height(u_1)$, we can modify N to obtain another network N' which is well-structured.

that occupies a tree node or a split node in N still occupies a tree node or a split node in N'. Thus, N' is a network satisfying the lemma.

(2) If $height(v_\ell) \geq height(u_1)$, let u_k be the lowest node on the right merge path such that $height(v_\ell) < height(u_k)$. Let a be a species from $S(h)$ and let c be a species from $S(u_i)$, where $i \in \{1, 2, \ldots, k-1\}$. a is in the cluster S', while c is not. Thus, $M(a, c) > 2 \times height(v_\ell) \geq 2 \times height(u_i)$, meaning that $M(a, c) = 2 \times height(s)$. Then, we can modify N into a network N' as shown in Fig. 5. More precisely, first create a node w on the left merge path with height $(\min\{height(u_k), height(v_{\ell+1})\} + height(v_\ell))/2$, where $v_{\ell+1}$ is the node immediately above v_ℓ on the left merge path. Then, create a node w' on the right merge path with height equal to $(height(u_k) + height(w))/2$. Next, remove h and u_1 and all their incident edges. Finally, insert an edge from w' to w, an edge from v_1 to the root of $S(h)$, and an edge from u_2 to the root of $S(u_1)$. It should be noted that N' is a valid network. Furthermore, for any pair of species $x, y \in S$, by checking the cases of whether x, y are in S' or in $S(u_i)$, $i \in \{1, 2, \ldots, k-1\}$, it is easy to verify that N' satisfies S. Also, S' occupies the tree node v_ℓ in N', and every cluster that occupies a tree node or a split node in N still occupies a tree node or a split node in N'. Thus, N' is a network satisfying the lemma. □

Finally, we can prove Lemma 4.

Proof of Lemma 4. Given a network N that satisfies the matrix M. If N is not well-structured, we pick any cluster that does not occupy any tree node or split node in N. We modify N according to Lemma 7 or 8. There are at most $2n-1$ different clusters and each round of modification causes at least one more cluster to occupy

a tree node or a split node. Thus, after at most $2n-1$ rounds of modification, we will obtain a network that is well-structured, and has the same number of hybrid nodes as N. □

3. The Algorithm

In this section, we present our $O(n^2 \log n)$-time algorithm for determining if there exists an ultrametric galled network satisfying a given $(n \times n)$-distance matrix M for a set S of n species, and constructing such a network if one exists. (In fact, whenever a solution exists, the algorithm will always output a well-structured network.) The algorithm is named GalledNet. We first give an outline of the algorithm and analyze its overall running time in Sec. 3.1. Then, in Sec. 3.2, we present the more technical details of Step 2c (procedure ConnectingSideClusters). Next, in Sec. 3.3, we prove that any network constructed by GalledNet is optimal in the sense that it has the minimum number of hybrid nodes among all possible networks satisfying M, including all well-structured and all nonwell-structured networks.

For short, we say *network* to refer to an ultrametric galled network.

3.1. *Framework of the algorithm*

According to Lemma 2, if M can be satisfied by some network then there exists a well-structured network which satisfies M. Therefore, in the following, we only consider well-structured networks and show how to find a well-structured network for S that satisfies M, if one exists.

Lemma 3 states that we can construct a network for a cluster by connecting the networks for its side clusters. Thus, our algorithm takes a bottom-up approach, which continuously identifies subsets of S that are clusters, starting from smaller ones to bigger ones. It maintains an invariant that as soon as a cluster S' is found, a subnetwork satisfying S' is constructed. For the base case, a set containing only a single species is a cluster, and the corresponding network is a single leaf for this species. Since S is the biggest cluster, the algorithm will eventually find a network satisfying S.

To efficiently identify and keep track of clusters, our algorithm uses an auxiliary graph G as follows. Initially, G consists of n isolated nodes, each representing a species in S. Edges which represent the distance among the species are added in rounds, where two nodes u, v will be connected by an edge of length $M(u, v)$. In the ith round, all edges with the ith shortest length are added. Suppose that after the ith round, a connected component of G becomes a clique. Then, the species inside this connected component form a cluster for which a network is built immediately.

The algorithm is named GalledNet and is outlined below. A detailed explanation of how to construct a network for a cluster, i.e. Step 2c, is provided in Sec. 3.2.

Algorithm 1 GalledNet

1: **[Step 1.] Sort the entries in M and let $m_1 < m_2 < \cdots < m_r$ be the distinct positive values in M. If $r > 3n - 3$, return failure.**
2: **[Step 2.] Build the networks while inserting edges into the auxiliary graph G. Initially, G contains n isolated nodes representing the n species. For $i = 1, 2, \ldots, r$,**
   a. Add all edges of length m_i to G.
   b. Identify all newly formed cliques.
   c. For each new clique, let $S' \subseteq S$ be the corresponding cluster. Run the procedure ConnectingSideClusters (described in Sec. 3.2) which constructs a network satisfying S', if S' is satisfiable. This is done by creating a new internal node u, and attaching the networks for side clusters of S' to u, or to a galled loop rooted at u.

It should be noted that Algorithm GalledNet terminates immediately, if there are more than $3n-3$ different positive values in M. The reason is that any galled network for n species can contain at most $3n-3$ internal nodes according to Lemma 1, and the length of any evolutionary path is $2 \times height(u)$ for some internal node u. Thus, if there are more than $3n - 3$ distinct positive values in M, no network can satisfy M.

We analyze the running time of GalledNet as follows. Step 1 takes $O(n^2 \log n)$ time. Step 2a takes $O(n^2)$ time over the whole algorithm. With some straightforward bookkeeping (which takes $O(1)$ time for each edge added), Step 2b can be done in $O(n)$ time in each iteration and $O(n^2)$ time in total. We will show in Sec. 3.2 that Step 2c, which calls ConnectingSideClusters, takes a total of $O(n^2)$ time. Thus, the whole algorithm takes $O(n^2 \log n)$ time.

To summarize:

Theorem 9. *Algorithm GalledNet determines if there exists an ultrametric galled network satisfying M, and if so, constructs such a network, in $O(n^2 \log n)$ time. Moreover, any network constructed by Algorithm GalledNet is well-structured.*

3.2. *Attaching side clusters to a galled loop*

This subsection explains how to perform Step 2c of GalledNet. Let S be a satisfiable cluster with side clusters $S_1, S_2, \ldots, S_t$. Suppose that we have constructed side networks for these side clusters. As mentioned below, we overload S_i to also denote the side network corresponding to side cluster S_i, and for convenience, we denote the height of the root node of this subnetwork by $height(S_i)$. To build a network for S, we need to determine how its side networks $S_1, S_2, \ldots, S_t$ should be attached to a new root node or to a galled loop, and compute the heights of the new root and all nodes on the galled loop.

First, we consider the simple case when $t = 2$. Suppose that the distance matrix of S contains $c \geq 1$ distinct distances $\{d_1, d_2, \ldots, d_c\}$ between a species in S_1 and a species in S_2. It should be remembered that S can be satisfied by a network with S_1 and S_2 attached either to a root or to a galled loop. The first case implies that

$c = 1$, and in the latter case $c = 2$. Thus, we do not need to consider c bigger than 2.

Lemma 10. *Assume that S contains only two side clusters S_1 and S_2.*

- *If $c = 1$, then S can be satisfied by a network N in which both S_1 and S_2 are attached to a root u and height$(u) = d_1/2$.*
- *Assume $c = 2$ and $d_1 < d_2$. Then S can be satisfied by a network N with a (skew) galled loop containing a root u, a side node v and a hybrid node h such that S_1 is attached to h and S_2 to v, and height$(u) = d_2/2$; height$(v) = d_1/2$; height$(h) = \alpha$, where α is any value in the range $(\text{height}(S_1), d_1/2)$.*

Proof. If $c = 1$, the distance from any species in S_1 to any species in S_2 is exactly d_1. The network N constructed in the lemma can obviously satisfy the distance matrix of S.

For $c = 2$, consider the network constructed in the lemma. It should be noted that the distance from any species in S_1 to any species in S_2 can only be d_1 or d_2. If the distance is d_1, then the shorter evolutionary path that passes through h and v satisfies the distance. Otherwise, if the distance is d_2, the evolutionary path passing through h, u, and v satisfies the distance. □

The rest of this subsection is devoted to the general case where S has three or more side clusters, i.e. $t \geq 3$. In this case, we need to build a galled loop to accommodate the corresponding side networks S_i's. We focus on the network N that satisfies S and we show that the structure of N can be determined from the relations between the side clusters. It should be remembered that N has a galled loop at the top. Let S_h be the side cluster attached to the hybrid node. Let LEFT(S_h) be the group of side clusters attached to the side nodes on the left merge path. Define RIGHT(S_h) similarly. The following lemma tells how to identify the side clusters in the two groups. For simplicity, we say a species a is in LEFT(S_h) (resp. RIGHT(S_h)), if a belongs to some side cluster in LEFT(S_h) (resp. RIGHT(S_h)).

Lemma 11 (Partitioning the side clusters according to the two merge paths). *Let D_S be the maximum distance between two species in S.* (i) *For any two species a, b in LEFT(S_h), $M(a, b) < D_S$; similarly, for any two species a, b in RIGHT(S_h), $M(a, b) < D_S$; and* (ii) *for any species a in LEFT(S_h) and c in RIGHT(S_h), $M(a, c) = D_S$.*

Proof. (i) Consider two species a, b in the same group (i.e. either LEFT(S_h) or RIGHT(S_h)). If a and b belongs to different side clusters, the highest node on their evolutionary path must be some side node on the corresponding merge path, whose height is less than the split node. The height of the split node is $D_S/2$, so $M(a, b)$, which equals the length of the evolutionary path, is less than D_S. The case for a and b belonging to the same side cluster is obvious.

(ii) For any two species a, c in different groups, the highest node on their evolutionary path must be the root of N. Thus, $M(a, c)$, which equals the length of the evolutionary path, is exactly D_S. □

Assume that LEFT(S_h) contains ℓ side clusters, and that their side networks are attached to side nodes $v_1, v_2, \ldots, v_\ell$ on the left merge path of N, where v_i is the ith node from the hybrid node. Let r be the root. Denote the side cluster (as well as the side network) attached to v_i by $S(v_i)$. That is, LEFT(S_h) = $\{S(v_i) \mid 1 \leq i \leq \ell\}$.

The following lemmas provide some structural characteristics of each side network $S(v_i)$, which allow us to identify them easily. For each side cluster S' of S, let $inter\_dist(S')$ denote the minimum distance $M(x, y)$ between a species x in S' and a species y in $S - S'$.

Lemma 12 (Identifying the order of side clusters). (i) $inter\_dist(S(v_1)) \leq inter\_dist(S(v_2))$, *and* $inter\_dist(S(v_2)) < inter\_dist(S(v_3)) < \cdots < inter\_dist(S(v_\ell))$;
(ii) $height(v_i) = inter\_dist(S(v_i))/2$ *for* $i = 2, \ldots, \ell$.

Proof. We prove (ii) first. For any $i = 2, \ldots, \ell$, for any species $a \in S(v_i)$ and any species $b \in S - S(v_i)$, the evolutionary path between a and b passes the node v_i. Thus, $M(a, b)$, which equals the length of some evolutionary path, is at least $2 \times height(v_i)$. It means that $inter\_dist(S(v_i)) \geq 2 \times height(v_i)$. On the other hand, for any species a in $S(v_i)$ and any species c in $S(v_1)$, the highest node on the evolutionary path between a and c must be v_i. Thus, $M(a, c) = 2 \times height(v_i)$, meaning that $inter\_dist(S(v_i)) \leq 2 \times height(v_i)$. This completes the proof.

For (i), for any species a in $S(v_1)$ and any species b in $S(v_2)$, the highest node on their evolutionary path must be v_2. Thus, $inter\_dist(S(v_1)) \leq 2 \times height(v_2) = inter\_dist(S(v_2))$. $inter\_dist(S(v_2)) < inter\_dist(S(v_3)) < \cdots < inter\_dist(S(v_\ell))$ follows directly from (ii). □

Lemma 12(i) allows us to identify which side cluster in LEFT(S_h) is attached to each v_i, except when $inter\_dist(S(v_1)) = inter\_dist(S(v_2))$. In this case, we exploit the relationship with S_h to distinguish the side clusters attached to v_1 and v_2. It should be noted that a species x in $S(v_2)$ and a species y in S_h are connected by two evolutionary paths, with r and v_2 as the highest node, respectively. Since N satisfies S, the distance of x and y (i.e. $M(x, y)$) must equal the length of either path, i.e. $2 \times height(r)$ or $2 \times height(v_2)$. The latter value is strictly less than $2 \times height(r) = D_S$.

Lemma 13 (Resolving ambiguity). (i) *If* $inter\_dist(S(v_1)) = inter\_dist(S(v_2))$, *then* $S(v_2)$, *but not* $S(v_1)$, *contains a species* x *whose distance to some species* y *in* S_h (*i.e.* $M(x, y)$) *is less than* D_S, *and* $height(v_1)$ *can be any value in the range* $(height(S_h), inter\_dist(S(v_2))/2)$.
(ii) *Otherwise,* $height(v_1) = inter\_dist(S(v_1))/2$.

Proof. (i) Note that $height(v_1) < height(v_2)$. If $S(v_1)$ contains a species x whose distance to some species y in S_h is less than D_S, then $inter\_dist(S(v_1)) = 2 \times height(v_1) < 2 \times height(v_2) = inter\_dist(S(v_2))$, which leads to a contradiction. If $S(v_2)$ does not contain a species x with the required property, then any species x in $S(v_1) \cup S(v_2)$ and any species y in $S - S(v_1) \cup S(v_2)$ have distance (i.e. $M(x, y)$) greater than $2 \times height(v_2)$. It means that $S(v_1) \cup S(v_2)$ is a cluster, which leads to a contradiction that $S(v_1)$ and $S(v_2)$ are not side clusters of S (i.e. maximal clusters contained by S).

Since any species x in $S(v_1)$ and any species y in S_h have distance exactly D_S, the evolutionary paths with highest node at the root will satisfy $M(x, y)$. Thus, $height(v_1)$ can be any value in $(height(S_h), inter\_dist(S(v_2))/2)$, and the resulting network is valid and still satisfies the distance requirement between a species in $S(v_1)$ and a species in $S - S(v_1)$.

(ii) This is the case for $inter\_dist(S(v_1)) < inter\_dist(S(v_2))$. Note that $inter\_dist(S(v_1))$ is either D_S or $2 \times height(v_1)$, and D_S is not less than $inter\_dist(S(v_2))$. Thus, $inter\_dist(S(v_1)) = 2 \times height(v_1)$ and the lemma follows. □

The above lemmas explain how the side clusters are attached to the merge paths, once the side cluster under the hybrid node is known. The following lemma shows that we can in fact find the side cluster attached to the hybrid node easily.

Lemma 14 (Finding S_h). (i) *$inter\_dist(S_h) \leq inter\_dist(S_i)$ for any side cluster S_i of S; and* (ii) *there can be at most five side clusters S_i of S such that $inter\_dist(S_i) = inter\_dist(S_h)$.*

Proof. (i) We will first show that $inter\_dist(S_h) \leq inter\_dist(S(v_1))$. We consider the two cases of Lemma 13. If $inter\_dist(S(v_1)) = inter\_dist(S(v_2))$, then there is a species x in $S(v_2)$ whose distance to some species y of S_h is $2 \times height(v_2)$, meaning that $inter\_dist(S_h) \leq inter\_dist(S(v_2)) = inter\_dist(S(v_1))$. If $inter\_dist(S(v_1)) < inter\_dist(S(v_2))$, then there is a species x in $S(v_1)$ whose distance to some species y of S_h is $2 \times height(v_1)$, meaning that $inter\_dist(S_h) \leq inter\_dist(S(v_1))$. Thus, $inter\_dist(S_h)$ is no greater than $inter\_dist(S_i)$ for all side clusters S_i in $\mathrm{LEFT}(S_h)$. We can repeat the same argument for side clusters in $\mathrm{RIGHT}(S_h)$, and conclude that $inter\_dist(S_h) \leq inter\_dist(S_i)$ for all side cluster S_i of S.

(ii) There are at most two side clusters in $\mathrm{LEFT}(S_h)$ (which are $S(v_1)$ and $S(v_2)$) having the minimum $inter\_dist$ value as S_h. The same is true for side clusters in $\mathrm{RIGHT}(S_h)$. Together with S_h itself, there are at most five side clusters S_i such that $inter\_dist(S_i) = inter\_dist(S_h)$. □

On the basis of the above lemmas, we can construct a galled loop to connect the side clusters for S, as follows. By Lemma 14, there are at most five candidates for the side cluster attached to the hybrid node. We try to build the network using each of the candidate according to Lemmas 11–13. We verify each network constructed

and return the one that satisfies S. The details of the algorithm are shown below. It builds a network for S, if and only if S is satisfiable.

Algorithm 2 ConnectingSideClusters $(S, S_1, S_2, \ldots, S_t)$

1: **[Step 1.] If S has only two side clusters, i.e. $t = 2$, construct the network according to Lemma 10.**
2: **[Step 2.] Otherwise, find $inter\_dist(S_i)$ for each side cluster S_i and sort the side clusters according the $inter\_dist$ value. If there are more than five side clusters having the minimum $inter\_dist$ value, return failure. Otherwise, for each side cluster S_h with the minimum $inter\_dist$ value, try to build a network which attaches S_h to hybrid node, as follows.**
   a. **Divide the remaining side clusters into two groups LEFT(S_h) and RIGHT(S_h) that satisfy Lemma 11.**
   b. **Sort the side clusters in LEFT(S_h) according to the $inter\_dist$ value and attach the side clusters to the left merge path according to Lemmas 12 and 13. Repeat it for the side clusters in RIGHT(S_h).**
   c. **Let h_l and h_r be the lowest side node on the left and right merge path, respectively. Set $height(h)$ to any value in $(height(S_h), \min\{h_l, h_r\})$.**
   d. **Verify that for every pair of species a, b, there is an evolutionary path between a and b with distance $M(a, b)$. Return the network if this condition is true.**

3.2.1. *Implementation of ConnectingSideClusters*

It is straightforward to implement the procedure ConnectingSideClusters$(S, S_1, S_2, \ldots, S_t)$ in $O(t \log t + \phi)$ time, where ϕ is the number of species pair (x, y) where x and y are species belonging to two different side clusters of S. The key ideas for the procedure are given in the following.

- Finding $inter\_dist(S_i)$ for all side clusters can be done by checking $M(x, y)$ for all species x, y that belongs to different side clusters, which takes $O(\phi)$ time.
- For Step 2a, we arbitrarily pick a side cluster S_i to LEFT(S_h). For each other side cluster S_j, we put S_j to RIGHT(S_h) if some species in S_i has distance D_S with some species in S_j; and put S_j to LEFT(S_h) otherwise. Then, we verify if the partition satisfies Lemma 11, by checking $M(x, y)$ for every species x, y belonging to different side clusters.
- For Step 2d, for any two species x and y belonging to different side clusters, the length of their evolutionary paths can be calculated in $O(1)$ time, so the total verification time is $O(\phi)$. We do not need to check the distance for species within the same side cluster, because the side network for the side cluster already satisfies the required distance.

To measure the total running time of all procedure calls to ConnectingSideClusters over the whole execution of GalledNet, we need the following lemma.

Lemma 15. *Let T be the total number of side clusters considered by all procedure calls to ConnectingSideClusters over the whole execution of GalledNet, and let Φ be*

the total number of species pairs considered by all calls to ConnectingSideClusters that belong to two different side clusters. It holds that $T \leq 2n-1$ *and* $\Phi \leq \frac{n(n-1)}{2}$.

Proof. It should be remembered that the input to GalledNet is a matrix M for a set of n species. Because clusters are nested, there can be at most $2n-1$ different clusters. Throughout the execution of GalledNet, each cluster will be counted as a side cluster of another cluster only once, so $T \leq 2n-1$.

It should be noted that the species pairs considered by different calls to ConnectingSideClusters are disjoint, and there are only $\frac{n(n-1)}{2}$ different species pairs, so $\Phi \leq \frac{n(n-1)}{2}$. □

Thus, the total running time for ConnectingSideClusters over the whole execution of GalledNet is $O(n \log n + \frac{n(n-1)}{2}) = O(n^2)$.

3.3. *The minimality of the number of hybrid nodes*

Here, we prove that the network for M produced by GalledNet has the minimum number of hybrid nodes among all networks satisfying M. We begin with the following observation.

Lemma 16. *The network produced by GalledNet has the minimum number of hybrid nodes among all well-structured networks satisfying* M.

Proof. Let N be the well-structured network produced by GalledNet and let N' be any well-structured network satisfying M. For any cluster $S' \subseteq S$, let $N(S')$ be the smallest subnetwork in N containing all species in S'. Define $N'(S')$ similarly. Since N and N' are well-structured, both $N(S')$ and $N'(S')$ contain exactly all species in S'. On the basis of the definition of the procedure ConnectingSideClusters, it can be easily proved by induction that $N(S')$ has no more hybrid nodes than $N'(S')$. Thus, when $S' = S$, we conclude that N has no more hybrid nodes than N'. □

Next, let N^* be a network satisfying M with the minimum number of hybrid nodes. In case N^* is not well-structured, according to Lemma 4, we can always transform N^* into a well-structured network satisfying M with the same number of hybrid nodes. Thus, we obtain the following theorem.

Theorem 17. *Given a satisfiable matrix* M, *the network produced by GalledNet has the minimum number of hybrid nodes among all networks satisfying* M.

4. NP-Hardness Results

In this section, we prove that two problems related to reconstructing an ultrametric galled network from a given distance matrix are NP-hard. We first show a useful

lemma. In the following, we say that a distance matrix M *admits* an ultrametric galled network, if there exists such a network which satisfies M.

Lemma 18. *Let M be a distance matrix for a set S. If there exist $a, b, c, d \in S$ such that $M(a,b) = M(a,c) = M(a,d) = M(b,c) = M(b,d) = M(c,d)$ then M does not admit any ultrametric galled network.*

Proof. Suppose M admits an ultrametric galled network. Write $X = M(a,b)$. Since $M(a,b) = M(a,c) = M(b,c)$ and every internal node in a galled network has degree at most two, the leaves a, b, c must belong to three different side networks of some split node having height $\frac{X}{2}$, where one leaf (without loss of generality, assume b) is a descendant of a hybrid node h and the other two leaves belong to two different merge paths. If d is a descendant of h, then $M(b,d) < X$, which is impossible. Thus, d must be attached to the same merge path as one of a and c. But then, either $M(a,d) < X$ or $M(c,d) < X$, which is a contradiction. Thus, M does not admit any ultrametric galled network. □

4.1. *NP-hardness of finding a maximum submatrix admitting an ultrametric galled network*

Here, we prove that finding a maximum submatrix M' of a given distance matrix M such that M' admits an ultrametric galled network is an NP-hard problem. Our proof consists of a polynomial-time reduction from the independent set problem which is known to be NP-hard (see Ref. 11).

4.1.1. *The independent set problem*

Instance: An undirected graph $G = (V, E)$ and a positive integer $I \leq |V|$.
Question: Is there a subset V' of V with $|V'| = I$ such that V' is an independent set, i.e. such that no two vertices in V' are joined by an edge in E?

4.1.2. *The maximum submatrix admitting an ultrametric galled network problem, decision problem version (MSGN-d)*

Instance: A set S, a distance matrix M for S, and a positive integer $K \leq |S|$.
Question: Is there a subset S' of S with $|S'| = K$ such that M restricted to S' admits an ultrametric galled network?

The following shows the reduction of the independent set problem to MSGN-d. Let (G, I) be any given instance of the independent set problem. For convenience, write $n = |V|$ and $V = \{v_1, v_2, \ldots, v_n\}$. Construct an instance (S, M, K) of MSGN-d as follows. Let $S, = V \cup P \cup Q$, where $P = \{p_1, p_2, \ldots, p_n\}$ and $Q = \{q_1, q_2, \ldots, q_n\}$ are two disjoint sets of elements not in V, and set $K = I + 2n$. Next, let M be a distance matrix for S satisfying, for every $i, j \in \{1, 2, \ldots, n\}$: $M(p_i, p_j) = \max\{i, j\}$; $M(q_i, q_j) = \max\{i, j\}$; $M(p_i, q_j) = n + 1$; $M(v_i, v_j) = \max\{i, j\}$ if the edge $\{v_i, v_j\}$

does not belong to E and $M(v_i, v_j) = n + 1$ if the edge $\{v_i, v_j\}$ belongs to E; $M(v_i, p_j) = n + 1$; and $M(v_i, q_j) = n + 1$.

Lemma 19. *M has a submatrix of size $K \times K$ which admits an ultrametric galled network if and only if G has an independent set of size I.*

Proof. Suppose that G has an independent set W of size I. Let $S' = W \cup P \cup Q$ and let M' be the $(K \times K)$-submatrix of M induced by S'. Let C_P be the binary caterpillar tree[b] whose leaves are labeled (in the order of nonincreasing distance from the root) by $p_1, p_2, \ldots, p_n$. For each internal node u of C_P, set the height of u equal to $\frac{i}{2}$, where p_i is the leaf descendant of u with the maximum index. Define C_Q and C_W analogously (using Q and W, respectively). Now, let N' be a galled network consisting of a root node at height $\frac{n}{2} + \frac{1}{2}$ having two children s and t, both at height $\frac{n}{2} + \frac{1}{4}$. Let s be the parent of C_P and a hybrid node h, and let t be the parent of C_Q and h. Finally, let h be the parent of C_W and set the height of h to $\frac{n}{2} + \frac{1}{8}$. It is easy to verify that N' satisfies M' and that N' is ultrametric.

Conversely, suppose there exists a subset S' of S of size K such that the submatrix of M induced by S' admits an ultrametric galled network. Since $K > 2n$, S' contains at least one element p_i from P and at least one element q_j from Q. Next, observe that for each $\{v_x, v_y\} \in E$, we have $M(v_x, v_y) = n + 1$, while $M(v_x, p_i) = M(v_x, q_j) = M(v_y, p_i) = M(v_y, q_j) = M(p_i, q_j) = n+1$. By Lemma 18, at most one of x and y can be included in S'. Thus, $W = S' \cap V$ is an independent set in G and $|W| \geq K - 2n = I$. □

Hence, MSGN-d is NP-hard and it follows that MSGN is NP-hard.

Theorem 20. *MSGN is NP-hard.*

4.2. *NP-hardness of the incomplete distance matrix case*

Next, we prove that it is NP-hard to determine whether a given incomplete distance matrix admits an ultrametric galled network. For the reduction, we make use of the NP-hard 3-coloring problem (see Ref. 11).

4.2.1. *3-coloring*

Instance: A connected, undirected graph $G = (V, E)$.

Question: Can G be 3-colored, i.e. can V be partitioned into three disjoint subsets in such a way that E contains no edge between two vertices in the same subset?

[b]A *caterpillar tree* is a rooted tree such that every internal node has at most one child which is not a leaf (see, e.g. Ref. 3).

4.2.2. *The incomplete distance matrix admitting an ultrametric galled network problem (IDGN)*

Instance: A set S and an incomplete distance matrix M (i.e. where some entries are missing) for S.

Question: Is there an ultrametric galled network which satisfies all of the nonempty entries in M?

Let G be any given instance of 3-coloring with at least two vertices. Construct an instance (S, M) of IDGN by setting $S = V$ and defining the $(|S| \times |S|)$-matrix M as follows: for every $i \in V$, let $M(i,i) = 0$; and for every edge $\{i,j\} \in E$, let $M(i,j) = M(j,i) = 1$. For every pair of vertices i, j in V such that $\{i,j\} \notin E$, leave the matrix entries $M(i,j)$ and $M(j,i)$ empty.

Lemma 21. *G is 3-colorable if and only if there exists an ultrametric galled network which satisfies all of the nonempty entries in M.*

Proof. Suppose G is 3-colorable. Partition V into three disjoint subsets V_1, V_2, V_3 such that each V_i is an independent set and build an ultrametric galled network N as follows. Let the root node r of N have height $\frac{1}{2}$, and let r's two children s and t have height $\frac{1}{4}$. Let s and t be the parents of a hybrid node h with height $\frac{1}{8}$. For $k \in \{1,2,3\}$, build an arbitrary tree T_k distinctly leaf-labeled by V_k. Attach T_1 as a child of s, T_2 as a child of h, and T_3 as a child of t, and assign arbitrary valid heights to the internal nodes of each T_i so that the resulting N is ultrametric. Now, every nonempty matrix entry in M equal to 0 is of the form $M(i,i)$ and, therefore, trivially satisfied by N. Next, consider any nonempty matrix entry $M(i,j)$ with $i \neq j$. By the construction above, we have $M(i,j) = 1$, and furthermore, E must contain the edge $\{i,j\}$. This means that i and j belong to different sets V_k and thus different side networks in N, so there exists a path of weight 1 in N from i to j passing through r. Therefore, all nonempty matrix entries in M are satisfied by N.

Now, we want to show that if there exists an ultrametric galled network which satisfies all of the nonempty entries in M, then G is 3-colorable. Let N be an ultrametric galled network that satisfies all the nonempty entries in M with the minimum possible number of hybrid nodes, let r be the root of N, and denote the two children of r by c_1 and c_2. It should be noted that it is not possible to have $height(r) < \frac{1}{2}$, since $|V| \geq 2$ and G is connected. As mentioned below, we describe how to use N to construct a partition (V_1, V_2, V_3) of V which induces a 3-coloring of G. There are two cases:

- r is not a split node. In this case, the height of r is always equal to $\frac{1}{2}$. (To see this, suppose on the contrary that $height(r) > \frac{1}{2}$. Pick any leaf descendant a of c_1 and any leaf descendant b of c_2, and consider any path $p_0(= a), p_1, p_2, \ldots, p_x(= b)$ in G connecting a and b. By the construction of M, $M(p_{i-1}, p_i) = 1$ for all $1 \leq i \leq x$, which implies that for every $1 \leq i \leq x$, p_i is contained in the subnetwork rooted

at c_1, contradicting that b is contained in the subnetwork rooted at c_2.) Let V_1 and V_2 be the (disjoint) sets of leaf descendants of c_1 and c_2, respectively, and let $V_3 = \emptyset$.

- r is a split node. Denote the hybrid node corresponding to r by h, and observe that $height(h) < \frac{1}{2}$ (if not, then by removing one edge ending at h and then for every node with resulting outdegree 1, contracting its outgoing edge, we would obtain another galled network with one less hybrid node than N which still satisfies M, contradicting the minimality of N). Define V_1 as the set of leaf descendants of h and let P and Q be the two merge paths from r to h. Define V_2 as the union of: (1) the set of leaf descendants of side networks attached to P at nodes with height greater than or equal to $\frac{1}{2}$; and (2) the set of leaf descendants of side networks attached to Q at nodes with height less than $\frac{1}{2}$ that were not already included in V_1. Finally, define V_3 in the same way as V_2, but using the upper part of Q and the lower part of P.

In either case, for any two vertices a, b in the same subset V_i, the distance in N between a and b is never equal to 1, which gives us $\{a, b\} \notin E$. Hence, each V_i forms an independent set in G, i.e. G is 3-colorable. □

Hence, the following theorem follows.

Theorem 22. *IDGN is NP-hard.*

5. Concluding Remarks

In this paper, we have introduced the problem of inferring an ultrametric galled network that satisfies a given distance matrix, and provided an efficient algorithm named GalledNet to solve it. Moreover, we have shown that two closely related problems are NP-hard.

We have implemented GalledNet using Java on a PC with a standard configuration (2.4 GHz with 512 MB memory).[c] To visualize the constructed network, we make use of two freely available graph drawing tools: Graph Visualization (Graphviz[1]) and Visualizing Graphs with Java (VGJ[2]). We output the network from our program in two representations, the dot language[10] and the graph modeling language (GML[16]), which are the input formats for Graphviz and VGJ, respectively. A sample output produced by Graphviz with 15 species is given in Fig. 6. The program is fast and can produce the output in seconds for a few dozens of species.

As a remark, in real cases, the input distance matrix might not be satisfiable by any ultrametric galled network due to the following. The estimation of the evolutionary distance based on alignment scores may not be accurate, or there may be errors in the sequences of the selected genes. Also, the set of selected genes

[c]The programs are available upon request.

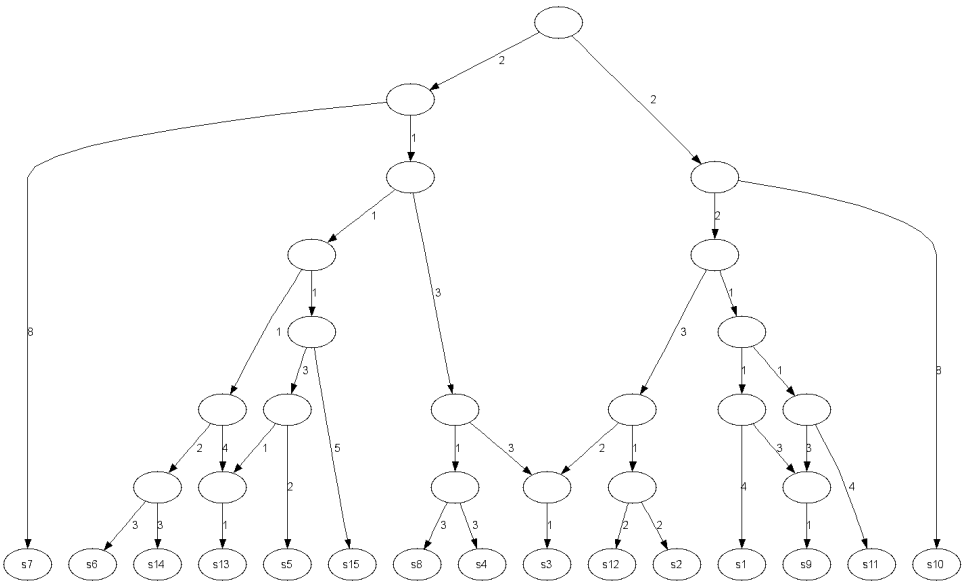


Fig. 6. A sample output for 15 species.

might not all have evolved from the same common ancestor. Thus, the estimated evolutionary distance between two species may not correspond exactly to the length of an evolutionary path between them. On the other hand, it is likely that the estimated distance between any pair of species should fall between the lengths of the shortest and longest evolutionary paths connecting them. To handle this issue, one can extend our problem to construct an ultrametric galled network N from the given distance matrix M such that for every pair a, b of species, the interval defined by the lengths of the shortest and longest evolutionary paths between a and b in N contains the value of $M(a, b)$.

Algorithm GalledNet can be modified to report such a network, if one exists. For the case of nonultrametric inputs, the major observation is that Lemma 2 still holds. That is, every cluster will still occupy a split node or tree node. Thus, we can solve the new problem by following the same bottom-up approach as in Sec. 3, but removing the old stopping condition in Step 1 and relaxing the requirements for combining side clusters into a galled loop in procedure ConnectingSideClusters as follows. Like before, S_h must have minimum *inter_dist* and we only need to test at most five candidates for the side cluster attached to the hybrid node in Step 2 because Lemma 14 is still true. In Steps 2a and 2b, we divide the side clusters and sort them as before (Lemmas 11 and 12 still hold); however, resolving ambiguity requires a relaxed Lemma 13:

(i) If $inter\_dist(S(v_1)) = inter\_dist(S(v_2))$, then we cannot resolve the ambiguity, and swapping the positions of the two side clusters still yields a network

satisfying the distance matrix. In this case, $height(v_1)$ can be any value in the range $(height(S_h), inter\_dist(S(v_2))/2)$.

(ii) Otherwise, $height(v_1) = inter\_dist(S(v_1))/2$.

In the last step of ConnectingSideClusters, we verify every constructed network using the relaxed requirement.

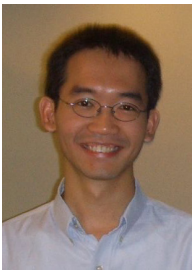
Acknowledgments

We thank Wing-Kin Sung for introducing this problem to us and for his useful comments. We also thank C. F. Li for implementing the algorithm.

References

1. AT&T Bell Laboratories, Graphviz — graph visualization software, http://www.graphviz.org.
2. Auburn University, Drawing graphs with VGJ. http://www.eng.auburn.edu/department/cse/research/graph_drawing/graph_drawing.html.
3. Bryant D, *Building trees, hunting for trees, and comparing trees: theory and methods in phylogenetic analysis*, Ph.D. Thesis, University of Canterbury, Christchurch, New Zealand, 1997.
4. Bryant D, Moulton V, Neighbor-Net: an agglomerative method for the construction of phylogenetic networks, *Mol Biol Evol* **21**:255–265, 2004.
5. Chen H-F, Chang M-S, An efficient exact algorithm for the minimum ultrametric tree problem, *Proc. 15th Int Symp Algorithms Comput (ISAAC 2004)*, Vol. 3341 of *LNCS*, Springer, pp. 282–293, 2004.
6. Choy C *et al.*, Computing the maximum agreement of phylogenetic networks, *Theor Comput Sci* **335**:93–107, 2005.
7. Day WE, Computational complexity of inferring phylogenies from dissimilarity matrices, *Bulletin Math Biol* **49**:461–467, 1987.
8. Diday E, Bertrand P, An extension of hierarchical clustering: the pyramidal representation, *Pattern Recogn in Practice II*, 411–424, 1986.
9. Farach M, Kannan S, Warnow T, A robust model for finding optimal evolutionary trees, *Algorithmica* **13**:155–179, 1995.
10. Gansner E, Koutsofios E, North S, Drawing graphs with *dot*, http://www.research.att.com/sw/tools/graphviz/dotguide.pdf.
11. Garey M, Johnson D, *Computers and Intractability — A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, New York, 1979.
12. Gusfield D, *Algorithms on Strings, Trees, and Sequences*, Cambridge University Press, New York, 1997.
13. Gusfield D, Bansal V, A fundamental decomposition theory for phylogenetic networks and incompatible characters, *Proc 9th Annual Int Conf Res Comput Mol Biol* (RECOMB 2005), Vol. 3500 of *LNCS*, Springer-Verlag, Berlin, Heidelberg, pp. 217–232, 2005.
14. Gusfield D, Eddhu S, Langley C, Optimal, efficient reconstruction of phylogenetic networks with constrained recombination, *J Bioinf Computat Biol* **2**:173–213, 2004.
15. He Y-J *et al.* Inferring phylogenetic relationships avoiding forbidden rooted triplets, *J Bioinf Comput Biol* **4**:59–74, 2006.

16. Himsolt M, GML: a portable graph file format, http://infosun.fmi.uni-passau.de/Graphlet/ GML/gml-tr.html.
17. Holland B, Moulton V, Consensus networks: a method for visualising incompatibilities in collections of trees, *Proc 3rd Workshop Algorithms Bioinf* (WABI 2003), Vol. 2812 of *LNCS*, Springer-Verlag, Berlin, Heidelberg, pp. 165–176, 2003.
18. Huson DH *et al.*, Phylogenetic super-networks from partial trees, *Proc 4th Workshop Algorithms Bioinfor* (WABI 2004), Vol. 3240 of *LNCS*, Springer, pp. 388–399, 2004.
19. Huynh TND *et al.*, Constructing a smallest refining galled phylogenetic network, *Proc 9th Annual Int Conf Res Comput Mol Biol* (RECOMB 2005), Vol. 3500 of *LNCS*, Springer-Verlag, Berlin, Heidelberg, pp. 265–280, 2005.
20. Jansson J, Nguyen NB, Sung W-K, Algorithms for combining rooted triplets into a galled phylogenetic network, *SIAM J Comput* **35**:1098–1121, 2006.
21. Jansson J, Sung Y, The maximum agreement of two nested phylogenetic networks, *Proc 15th Int Symp Algorithms Comput* (ISAAC 2004), Vol. 3341 of *LNCS*, Springer-Verlag, Berlin, Heidelberg, pp. 581–593, 2004.
22. Linder CR *et al.*, Network (reticulate) evolution: biology, models, and algorithms, Tutorial presented at *the 9th Pac Symp Biocomputing* (PSB 2004), 2004.
23. Nakhleh L, Warnow T, Linder CR, Reconstructing reticulate evolution in species — theory and practice, *Proc 8th Annual Int Conf Res Comput Mole Biol* (RECOMB 2004), pp. 337–346, 2004.
24. Posada D, Crandall KA, Intraspecific gene genealogies: trees grafting into networks, *TRENDS Ecolo & Evol* **16**:37–45, 2001.
25. Saitou N, Nei M, The neighbor-joining method: a new method for reconstructing phylogenetic trees, *Mole Biol Evol* **4**:406–425, 1987.
26. Setubal JC, Meidanis J, *Introduction to Computational Molecular Biology*, PWS Publishing Company, Boston, 1997.
27. Wang L, Zhang K, Zhang L, Perfect phylogenetic networks with recombination, *J Comput Biol* **8**:69–78, 2001.
28. Wu BY, Chao K-M, Tang CY, Approximation and exact algorithms for constructing minimum ultrametric trees from distance matrices, *J Comb Optimization* **3**:199–211, 1999.



Ho-Leung Chan received his B.E. in Computer Engineering from the University of Hong Kong in 2002. He is now a Ph.D. student in the Department of Computer Science at the University of Hong Kong. His research interests include design and analysis of algorithms.

Tak-Wah Lam graduated with a B.Sc. in Computer Science from the Chinese University of Hong Kong in 1984 and received his M.S. & Ph.D. in Computer Science from the University of Washington in 1988. He is now an associate professor in the Department of Computer Science at the University of Hong Kong.

He is interested in the design and analysis of algorithms for different applications. His recent work is in the areas of computational biology, online scheduling, and compressed text indexing.

Siu-Ming Yiu received his B.Sc. degree in Computer Science from the Chinese University of Hong Kong, a M.S. degree in Computer and Information Science from Temple University, and a Ph.D. degree in Computer Science from the University of Hong Kong. He is now a Research Assistant Professor in the Department of Computer Science at the University of Hong Kong. Computational Biology is currently one of his major research interests.

Jesper Jansson received his Ph.D. degree in Computer Science from Lund University, Sweden, in 2003.

He is currently a postdoc researcher at Kyushu University, Japan, and has previously worked at National University of Singapore and The University of Hong Kong.

His main research interests include efficient graph algorithms and their applications to bioinformatics; in particular, supertree methods, inferring and comparing phylogenetic networks, clustering binary data, and fast algorithms for computing alignments between labeled trees for RNA secondary structure comparison and prediction.