CHARACTERIZATION OF THE EXISTENCE OF GALLED-TREE NETWORKS (EXTENDED ABSTRACT)

JÁN MAŇUCH,* XIAOHONG ZHAO, LADISLAV STACHO† AND ARVIND GUPTA‡

School of Computing Science and Department of Mathematics
Simon Fraser University, Canada
Email: arvind,jmanuch,lstacho,xzhao2@sfu.ca

In this paper, we give a complete characterization of the existence of a galled-tree network in the form of simple sufficient and necessary conditions. As a by-product we obtain as simple algorithm for constructing galled-tree networks. We also introduce a new necessary condition for the existence of a galled-tree network similar to bi-convexity.

1. Introduction

With the progress of human genome project[7], large amount of genomic data is available. Analysis of this data requires new methods incorporating events such as recombination, gene conversion, horizontal gene transfer and mobile genetic elements[8,9]. The traditional phylogenetic tree model is not sufficient enymore. In particular, recombination attracts much attention, because of its important role in locating genes influencing complex genetic diseases. A fundamental model which incorporates recombinations, phylogenetic networks, was introduced by Wang et al.[10]. With no restrictions on location of recombinations, they showed that the problem of finding a phylogenetic network with minimum number of recombinations is NP-hard. They also proposed a constrained phylogenetic network model with vertex-disjoint recombination cycles, called a galled-tree network.

Gusfield et al.[6] presented a polytime algorithm for constructing a galled-tree network. The algorithm is based on a number of necessary conditions on the existence of such networks. Some of these conditions are properties of so-called "conflict graph". More necessary conditions were given in the subsequent paper[5]. Surprisingly, unlike in case of phylogenetic trees, no characterization is known for galled-tree networks.

*Research supported in part by PIMS (Pacific Institute for Mathematical Sciences).
†Research supported in part by NSERC (Natural Science and Engineering Research Council of Canada) grant.
‡Research supported in part by NSERC (Natural Science and Engineering Research Council of Canada) grant.

In this paper, we give a complete characterization for the existence of a galled-tree network in the form of simple sufficient and necessary conditions. In particular, we show that two necessary conditions observed by Gusfield et al.[6] are enough to guarantee the existence of a galled-tree network. In our model we assumed that the root of the galled-tree network is labeled by the all-0 sequence. Note that very recently an algorithm for constructing a galled-tree network without any assumption on the label of the root (root-unknown network) was presented[3]. As a by-product, we obtained a simple algorithm for constructing galled-tree networks. Gusfield et al.[6] introduced an interesting necessary condition, called bi-convexity, which they used to design a fast algorithm for the site consistency problem for a matrix A if there exists a galled-tree network explaining A. As another by-product, we present a new necessary condition (bi-inclusiveness) which implies bi-convexity (but not other way around). Gusfield et al.[6] conjectured that the minimum vertex cover of a bi-convex graph can be found in linear time. We show that the cover of a bi-inclusive graph can be found in linear time assuming we know the order of vertices sorted by their degrees. Otherwise we need to add the sorting time to the complexity.

2. Preliminaries

The input to the problem is a haplotype $n \times m$ matrix A with values in $\{0, 1\}$ (binary), where each row represents a haplotype sequence of an individual and each column corresponds to a character (an SNP site in the DNA sequence). The set of characters is assumed to be the set $\{1, \ldots, m\}$. For every character c, the sequence in a row contains in column c the state of character c for that individual. We use the terms "column" and "character" interchangeably.

We will assume that the edges of structures used to explain the input matrix (perfect phylogenies, galled-trees) are directed from the root to leaves. An edge (u, v) is a directed edge from u to v, i.e., u is closer the root than v. We will also assume that root is labeled with the all-0 sequence. We can also assume that no column contains only 0-states, as such columns do not affect solution to any of the considered problems. In the following definition we describe two basic operations on the matrices which we will use frequently.

Definition 2.1. Given an $n \times m$ binary matrix A. Let S be a subset of characters of A. The matrix $A[S]$ is the sub-matrix of A restricted to the columns in S. We will assume that the names of columns in $A[S]$ are the same as in the original matrix A. Let x be a binary sequence of length $|S|$. By $A[S] - x$, we denote the sub-matrix of $A[S]$ from which we remove all rows whose strings are identical to x.

2.1. *Perfect phylogeny*

The main combinatorial tool used in evolutionary biology is the concept of perfect phylogeny (phylogenetic tree). In our considerations phylogenetic trees appear in several places (construction of galls, compressed trees for galled-tree networks).

Definition 2.2. (Perfect phylogeny) Given an $n \times m$ binary matrix A. A *phylogenetic tree* on m characters is a rooted tree having each edge labeled with a unique character in the set $\{1, \ldots, m\}$, i.e., no two edges have the same label. Given a phylogenetic tree, we assign to each vertex a binary sequence of length m in top-down fashion as follows: the root is labeled with the all-0 sequence; for every edge (u, v) labeled with a character c, the label of v is obtained from the label of u by changing 0 at position c to 1 (changing state of character c). We say that a phylogenetic tree T *explains* A if each sequence of A (contained in a row) is a label of some vertex in T. If there is such a tree, we sometimes say A has a perfect phylogeny.

Note that the usual definition of phylogenetic tree T requires the sequences of A to be contained in the leaves of T. However, such a definition allows for unlabeled edges along which labels of end vertices do not change. It is easy to convert our phylogenetic tree to a standard phylogenetic tree. We prefer our definition, as our phylogenetic trees are more compact.

The following is the classical characterization of the existence of the perfect phylogenetic tree rediscovered in many papers. Before stating the result we need the following definition.

Definition 2.3. (Conflicting characters) Given an $n \times m$ binary matrix A. Two characters/columns c and c' *conflict* in A if $A[c, c']$ contains three rows with pairs $[0, 1], [1, 0]$ and $[1, 1]$. A character is *unconflicted* if it does not conflict with any other character.

Theorem 2.1. *Given an $n \times m$ binary matrix A. There exists a phylogenetic tree explaining A if and only if no two characters conflict in A.*

Note that if we drop the requirement in the definition of phylogenetic trees to have the root labeled with the all-0 sequence, the above theorem is still true, although we have to redefine conflicts between characters: c and c' conflict in A if $A[c, c']$ contains all 4-possible pairs (so-called *four-gamete test*).

Definition 2.4. Given a tree. If there is a directed path in the tree containing edges e and e', we say that e and e' are *comparable*. Take the shortest such a path. If e is the first edge on the path, we say that e is an *ancestor* of e', and e' is a *descendant* of e, and write $e \preceq e'$. If there is no such path, we say that e and e' are *incomparable*.

Given an $n \times m$ binary matrix M. Let T be a phylogenetic tree explaining M. Define a map $e : \{1, \ldots, m\} \to E(T)$ returning the edge with label c as follows, for every character c, let $e(c) = e$ where e is the edge with the label c. Since we assume that M has no all-0 columns, the map is defined for every character.

2.2. *Definitions of phylogenetic and galled-tree networks*

Definition 2.5. A *phylogenetic network* N on m characters is a directed acyclic graph containing exactly one vertex (the root) with no incoming edges. Each vertex other than the root has either one or two incoming edges. If it has one incoming edge, the edge is called a *mutation edge*, otherwise it is called a *recombination edge*. A vertex x with two incoming edges is called a *recombination vertex*.

Each integer (character) from 1 to m is assigned to exactly one mutation edge in N and each mutation edge is assigned one character. Each vertex in N is labeled by a binary sequence of length m, starting with the root vertex which is labeled with the all-0 sequence. Since N is acyclic, the vertices in N can be topologically sorted into a list, where every vertex occurs in the list only after its parent(s). Using that list, we can define the labels of the non-root vertices, in order of their appearance in the list, as follows:

(1) For a non-recombination vertex v, let e be the mutation edge labeled c coming into v. The label of v is obtained from the label of v's parent by changing the value at position c from 0 to 1.
(2) Each recombination vertex x is associated with an integer $\mathrm{r}_x \in \{2, \ldots, m\}$, called the *recombination point* for x. Label the two recombination edges coming to x by P and S, respectively. Let $P(x)$ ($S(x)$) be the sequence of the parent of x on the edge labeled P (S). Then the label of x consists of the first $\mathrm{r}_x - 1$ characters of $P(x)$, followed by the last $m - \mathrm{r}_x + 1$ characters of $S(x)$. Hence $P(x)$ contributes a prefix and $S(x)$ contributes a suffix to x's sequence.

Recall that, in this paper, the sequence at the root of the phylogenetic network is always the all-0 sequence, and all results are relative to that assumption. More general phylogenetic networks with unknown root were studied in a recent paper by Gusfield[3]. Note also that there are slight differences in the definition of phylogenetic networks from the original definition[6,10]. We assume that each mutation edge has exactly one label. Every phylogenetic network without this assumption can be easily transformed to our model by replacing every mutation edge with multiple labels by a sequence of edges each having one of these labels, and contracting all mutation edges without a label. Our assumption results in more compact phylogenetic networks, however we cannot require that all sequences of an input matrix appear at the leaves of the network.

Definition 2.6. Given an $n \times m$ binary matrix A, we say that a phylogenetic network N with m characters *explains* A if each sequence of A is a label of some vertex in N.

Definition 2.7. (Galled-tree network) In a phylogenetic network N, let v be a vertex that has two paths out of it that meet at a recombination vertex x (v is

the lowest common ancestor of the parents of x). The two paths together form a *recombination cycle* Q. The vertex v is called the *coalescent vertex*. We say that Q contains a character c, if c labels one of the mutation edges of Q.

A phylogenetic network is called a *galled-tree network* if no two recombination cycles share an edge. A recombination cycle of a galled-tree network is sometimes referred to as a *gall*.

Note that in the original definition of galled-tree network[6,10] it is required that recombination cycles do not share vertices. It is easy to see that our modification is only a minor difference (one can be transformed to the other easily) introduced for technical reasons.

3. Characterization of the existence of a galled-tree network

In this section we will give a complete characterization of the existence of a galled-tree network explaining a given matrix A. We will show that two conditions (Lemma 4 and Theorem 10) in Gusfield et al.[6]) are also sufficient.

Definition 3.1. Given an $n \times m$ binary matrix A. The *conflict graph* G_A has the vertex set $\{1, \ldots, m\}$ and for every two characters c and c', (c, c') is an (undirected) edge of G_A if they conflict.

Our characterization of galled-tree networks is presented in the following theorem.

Theorem 3.1. *Given an $n \times m$ binary matrix A. There exists a galled-tree network explaining A if and only if every nontrivial component (having at least two vertices) K of the conflict graph G_A satisfies the following conditions:*

(1) K is bipartite with partitions L and R such that all characters in L are smaller than all characters in R; and
(2) there exists a sequence $x \neq 0^{|K|}$ such that $A[K] - x$ has no conflicting characters.

In the rest of this section we will prove several results which will imply the theorem. Throughout the rest of the paper, let A be a given $n \times m$ binary matrix.

The following crucial result shows that if the condition (2) of Theorem 3.1 is satisfied then $A[K] - x$ can be explained by a tree with two edge-disjoint branches.

Lemma 3.1. *If a component K of G_A is bipartite with partitions L and R, and $A[K] - x$ has no conflicting characters for some $x \neq 0^{|K|}$, then any phylogenetic tree T explaining $A[K] - x$ has at most two branches. For $i = 0, 1$, let L_i (R_i) be the set of all $c \in L$ ($c \in R$) such that $x[c] = i$. One possible branch contains all edges labeled with characters in $L_1 \cup R_0$, and the other contains all edges labeled*

with characters in $R_1 \cup L_0$. If T has two branches then they do not share any edge (recall that we assume that a phylogenetic tree has all edges labeled by characters).[a]

In the following theorem we will show that if a component of the conflict graph G_A satisfies both conditions of Theorem 3.1 then there is a gall explaining $A[K]$.

Theorem 3.2. *If a component K of G_A is bipartite with partitions L and R, $A[K]-x$ has no conflicting characters for some $x \neq 0^{|K|}$ and all vertices in L are smaller than all vertices in R, then $A[K]$ can be explained by a galled tree containing one recombination cycle (gall) rooted in the node with label $0^{|K|}$ and having x as a label of the recombination vertex.*

Proof. By Lemma 3.1, there is a phylogenetic tree T explaining $A[K]-x$ with at most two branches. Let B_P be the branch containing edges labeled with characters in $L_1 \cup R_0$, and B_S the branch containing edges labeled with characters in $R_1 \cup L_0$. If one of these two sets is empty then one of the branches is empty as well. Furthermore, the vertex labeled $0^{|K|}$ is the only vertex shared by B_P and B_S. Now, we will add a recombination vertex z into T. Let y_P (y_S) be the last vertex on the branch B_P (B_S). Add two recombination edges (y_P, z) labeled P and (y_S, z) labeled S, cf. Figure 1. Set the recombination point r_z to any character in $\{p+1, \ldots, q\}$, where p is the maximum character in L and q is the minimum character in R. We will show that the label of recombination vertex z is x, i.e., the gall explains the matrix $A[K]$.

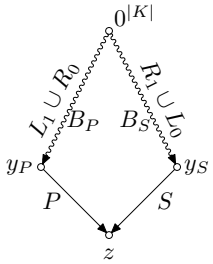

Figure 1. Construction of recombination cycle using two branches B_P and B_S of the phylogenetic tree for $A[K]-x$.

The label of z is formed by concatenating the first $r_z - 1$ characters of $P(z)$ (see Definition 2.5) with the last $|K| - r_z + 1$ characters of S_z. The label $P(z)$ (respectively, $S(z)$) has 0 (respectively, 1) in every position $c \in R_1 \cup L_0$ and 1 (respectively, 0) in every position $c \in L_1 \cup R_0$. The label of z at position $c \in L_0$ comes from $P(z)$, hence it has value 0. Similar arguments show that the label of z agrees with x also on all remaining positions, as required. □

[a] Due to the space limitation the proof will appear in the journal version.

In the following we define a compressed matrix which will be used to build a phylogenetic network. Note that the compressed matrix is similar to the pass-through matrix[4]. However, the pass-through matrix does not contain columns for components of the conflict graph which are singletons.

Definition 3.2. Let $K_1, \ldots, K_k$ be the components of the conflict graph G_A. The *compressed matrix* C_A is the $n \times k$ binary matrix with columns labeled by $K_1, \ldots, K_k$. It has 1 in row $i \in \{1, \ldots, n\}$ and column K_j, $j \in \{1, \ldots, k\}$, if and only if the row i in $A[K_j]$ contains at least one 1.

Lemma 3.2. *The compressed matrix C_A has no conflicting characters.*[b]

It follows that the compressed matrix C_A can be explained by a phylogenetic tree. We will use this tree to construct the galled-tree network explaining A. Recall that a phylogenetic tree with a fixed root is unique up to order of edges labeled with characters having identical columns in the input matrix. From all phylogenetic trees explaining C_A we want to pick one satisfying the following condition:

Definition 3.3. A phylogenetic tree T explaining C_A is called *sorted* if for every two identical columns K_j and $K_{j'}$ such that component K_j is a singleton and component $K_{j'}$ has at least two vertices in the conflict graph, $e(K_j) \prec e(K_{j'})$.

Following lemma shows that sequences in rows of A behave nicely with respect to edges in a sorted phylogenetic tree T explaining the compressed matrix C_A.

Lemma 3.3. *Let T be a sorted phylogenetic tree explaining the compressed matrix C_A. Assume that $e(K_j) \prec e(K_{j'})$ in T for some components K_j and $K_{j'}$ in G_A. Consider all rows containing a 1 in $A[K_{j'}]$, i.e., having 1 in $C_A[K_{j'}]$. Then all sequences in these rows in $A[K_j]$ are identical and different from the all-0 sequence.*[b]

The following algorithm constructs a galled-tree network N_A from a sorted phylogenetic tree for C_A.

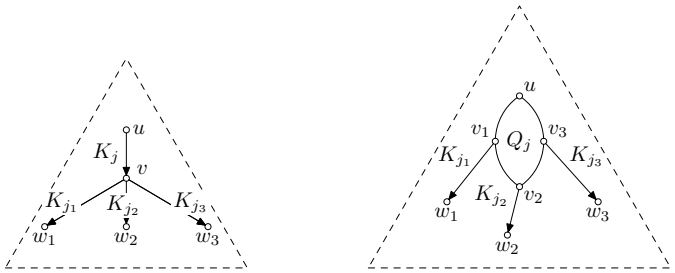

Figure 2. Replacing an edge labeled K_j with a gall Q_j.

[b]Due to the space limitation the proofs will appear in the journal version.

Algorithm 3.1.
Input: An $n \times m$ binary matrix A satisfying assumptions of Theorem 3.2.

(1) Construct a sorted phylogenetic tree T of C_A and for every component K_j, $j \in \{1, \ldots, k\}$, of G_A, construct the gall Q_j explaining $A[K_j]$.
(2) In top-down fashion process every edge (u, v) labeled K_j. If K_j is a singleton, i.e., $K_j = \{c\}$, replace the label of (u, v) by c. Otherwise, replace the edge with a gall Q_j for K_j as follows (cf. Figure 2):
    2.1 Remove edge (u, v).
    2.2 Identify the coalescent node of the gall Q_j with u.
    2.3 For every edge (v, w) labeled $K_{j'}$, consider any row r containing 1 in $C_A[K_{j'}]$. Let s be the sequence in $A[K_j]$ in row r. By Lemma 3.3, $s \neq 0^{|K_j|}$. Since Q_j explains $A[K_j]$, it contains a vertex $v' \neq u$ labeled s. Remove the edge (v, w), add the edge (v', w) and label it $K_{j'}$.
    2.4 Remove vertex v.
(3) To obtain a proper labeling of vertices in N_A, compute new labels of length m using the procedure described in the definition of galled-trees.

The following lemma shows that the algorithm produces essentially unique answer. More precisely,

Lemma 3.4. *After constructing a sorted phylogenetic tree T of C_A and galls Q_j's for every component K_j of G_A in Step 1 of Algorithm 3.1, the remaining construction of the algorithm produces unique result (the resulting galled-tree network depends only on selection of T and Q_j's).*

Proof. The only choice we have in the remaining steps of the algorithm is in Step 2.3 when we can choose any row r containing 1 in $C_A[K_{j'}]$. The selection of vertex v' to which we attach w depends on the sequence s in row r of the matrix $A[K_j]$. However, by Lemma 3.3, for every row r' containing 1 in $C_A[K_{j'}]$, the sequence in row r' of the matrix $A[K_j]$ is also s. □

The question of how many different galls are there for a matrix $A[K_j]$ was studied by Gusfield et al.[6]. It was shown that there are at most three different galls, and if there are enough characters in K_j, there is only one gall explaining $A[K_j]$. Also note that the phylogenetic tree T is unique up to arrangement of characters with identical columns on edges. For our purposes, the fact that Step 2.3 can be performed only in one unique way is sufficient to show that N_A explains A.

Theorem 3.3. *Assume that every non-trivial (with at least two vertices) component K of G_A is bipartite with partitions L and R, $A[K] - x$ has no conflicting characters for some $x \neq 0^{|K|}$ and all vertices in L are smaller than all vertices in R. Then the galled-tree network N_A constructed above explains A.*[c]

[c] Due to the space limitation the proof will appear in the journal version.

It is known that the number of galls in any galled-tree network explaining A is at least the number of non-trivial components in the conflict graph G_A[6]. Since the galled-tree network constructed by Algorithm 3.1 has exactly this number of galls, the constructed network is optimal.

Obviously, by Theorem 3.2, Algorithm 3.1 cannot fail to construct a galled-tree network N_A, and by the above theorem, the constructed network explains A. Hence, we have the following corollary.

Corollary 3.1. *If every non-trivial component K of G_A is bipartite with partitions L and R, $A[K] - x$ has no conflicting characters for some $x \neq 0^{|K|}$ and all vertices in L are smaller than all vertices in R, then there exists a galled-tree network explaining A.*

Combining the above corollary with the results of Gusfield et al.[6], Theorem 3.1 follows.

3.1. *Bi-inclusiveness*

Gusfield et al.[6] introduced an interesting necessary condition for the existence of a galled-tree network, called *bi-convexity*.

Definition 3.4. A bipartite graph K with partitions L and R is called *convex for* R if the vertices in R can be ordered so that, for each vertex $i \in L$, $N(i)$ forms a closed interval in R. That is, i is adjacent to j and $j' > j$ in R if and only if i is adjacent to all vertices in the set $\{j, \ldots, j'\}$. A bipartite graph is called *bi-convex* if sets L and R can be ordered so that it is simultaneously convex for L and convex for R.

They used bi-convexity to design a fast algorithm for the *site consistency problem* for a matrix A if there is a galled-tree network explaining A. The site consistency problem for a matrix A is to find a minimum number of columns whose removal from A results in a perfect phylogeny. The problem was introduced and shown to be NP-complete[1]. The problem reduces to finding a minimum vertex cover in the conflict graph G_A. For bipartite graphs, the vertex cover can be found in polynomial time and for bi-convex graphs in $\mathcal{O}(m^2)$ time (recall that m is the number of vertices in the conflict graph)[2]. It was conjectured by Gusfield et al.[6] that to find a minimum vertex cover of a bi-convex graph can be done in linear time. We present a new necessary condition, *bi-inclusiveness*, which is stronger than bi-convexity (it implies bi-convexity but not other way round) and observe that the minimum vertex cover of a bi-inclusive graph can be found in linear time.

Definition 3.5. We say that a collection of sets forms a chain, if there is an order $S_1, \ldots, S_k$ of sets such that $S_1 \subseteq S_2 \subseteq \cdots \subseteq S_k$. A bipartite graph K with partitions L and R is *bi-inclusive* if the sets $N(i_1), \ldots, N(i_k)$ form a chain, where $N(x)$ denotes the neighborhood of x.

Note that it is easy to check that the swapping of partitions does not change the property whether K is bi-inclusive or not.

The next theorem shows that if a matrix A satisfies sufficient and necessary conditions of Theorem 3.1, i.e., A can be explained by a galled-tree network, then every component of the conflict graph G_A is bi-inclusive.

Theorem 3.4. *Given an $n \times m$ binary matrix A. If a component K of G_A is bipartite and $A[K] - x$ has no conflicting characters for some $x \neq 0^{|K|}$, then K is bi-inclusive.*[d]

Since bi-inclusive graphs are chordal bipartite graphs, a minimum vertex cover of a bi-inclusive graph can be found in linear time given some additional information on the graph[2]. Hence we have the following.

Observation 3.1. A minimum vertex cover in a bi-inclusive graph can be found in $\mathcal{O}(m \log m)$ time and in linear time ($\mathcal{O}(m)$) if the chain order of vertices in one partition is given.

References

1. W. H. Day and D. Sankoff. Computational complexity of inferring phylogenies by compatibility. *Syst. Zool.*, 35(2):224–229, 1986.
2. F. F. Dragan. Strongly orderable graphs: A common generalization of strongly chordal and chordal bipartite graphs. *Discrete Appl. Math.*, 99(1-3):427–442, 2000.
3. D. Gusfield. Optimal, efficient reconstruction of root-unknown phylogenetic networks with constrained and structured recombination. *J. Computer and Systems Sciences*, 70:381–398, 2005.
4. D. Gusfield, S. Eddhu, and C. Langley. Powerpoint slides for: Efficient reconstruction of phylogenetic networks (of SNPs) with constrained recombination. http://wwwcsif.cs.ucdavis.edu/~gusfield/talks.html.
5. D. Gusfield, S. Eddhu, and C. Langley. The fine structure of galls in phylogenetic networks. *INFORMS Journal on Computing*, 16(4):459–469, 2004.
6. D. Gusfield, S. Eddhu, and C. Langley. Optimal, efficient reconstruction of phylogenetic networks with constrained recombination. *Journal of Bioinformatics and Computational Biology*, 2(1):173–213, 2004.
7. L. Helmuth. Genome research: Map of the human genome 3.0. *Science*, 293(5530):583–585, 2001.
8. D. Posada and K. A. Crandall. Intraspecific gene genealogies: trees grafting into networks. *Trends in Ecology and Evolution*, 16(1):37–45, 2001.
9. M. Schierup and J. Hein. Consequences of recombination on traditional phylogenetic analysis. *Genetics*, 156:879–891, 2000.
10. L. Wang, K. Zhang, and L. Zhang. Perfect phylogenetic networks with recombination. In *SAC '01: Proceedings of the 2001 ACM symposium on Applied computing*, pages 46–50, New York, NY, USA, 2001. ACM Press.

[d] Due to the space limitation the proof will appear in the journal version.