UC Davis Computer Science Technical Report CSE-2003-29[1]

Optimal, Efficient Reconstruction of Phylogenetic Networks with Constrained Recombination

Dan Gusfield, Satish Eddhu, Charles Langley

November 24, 2003

[1]To appear in Journal of Bioinformatics and Computational Biology.

Optimal, Efficient Reconstruction of Phylogenetic Networks with Constrained Recombination

Dan Gusfield and Satish Eddhu
Department of Computer Science
University of California, Davis
{gusfield,eddhu}@cs.ucdavis.edu

Charles Langley
Division of Evolution and Ecology
University of California, Davis
chlangley@ucdavis.edu

Keywords: Molecular Evolution, Phylogenetic Networks, Ancestral Recombination Graph, Recombination, SNP, bi-convex graph

Abstract

A phylogenetic network is a generalization of a phylogenetic tree, allowing structural properties that are not tree-like. With the growth of genomic data, much of which does not fit ideal tree models, there is greater need to understand the algorithmics and combinatorics of phylogenetic networks [17, 18]. However, to date, very little has been published on this, with the notable exception of the paper by Wang et al.[21]. Other related papers include [9, 10, 12, 20, 19].

Wang et al. [21] studied the problem of constructing a phylogenetic network for a set of n binary sequences derived from the all-0 ancestral sequence, when each site in the sequence can mutate from 0 to 1 at most once in the network, and recombination between sequences is allowed. They gave a polynomial-time algorithm that was intended to determine whether the sequences could be derived on such a phylogenetic network where the recombination cycles are node disjoint. In this paper, we call such a phylogenetic network a "galled-tree". That work is seminal in focusing on galled-trees, and for its assertion that reconstruction of such networks can be done in polynomial time. Unfortunately, the algorithm in [21] is incomplete and does not constitute a necessary test for the existence of a galled-tree for the data.

In this paper, we completely solve the problem of determining whether a set of binary sequences can be derived on a galled-tree. By more deeply analyzing the combinatorial constraints on cycle-disjoint phylogenetic networks, we obtain an efficient algorithm that is guaranteed to be both a necessary and sufficient test for the existence of a galled-tree for the data. If there is a galled-tree, the algorithm constructs one which is optimal, minimizing the number of recombinations over all phylogenetic networks for the data (using the all-0 ancestral sequence), even phylogenetic networks that are not restricted to be galled-trees, and even if their recombination events allow multiple-crossover recombinations. We also prove that when there is a galled-tree for the data, the galled-tree minimizing the number of recombinations is "essentially unique", with only limited modifications permitted. We also note two additional results: first, any set of sequences that can be derived on a galled tree can be derived on a true tree (without recombination cycles), where at most one back mutation is allowed per site; second, the site compatibility problem (which is NP-hard in general) can be solved in polynomial time for any set of sequences that can be derived on a galled tree.

The combinatorial constraints we develop apply (for the most part) to node-disjoint cycles in any phylogenetic network (not just galled-trees), and can be used for example to prove that a given site cannot be on a node-disjoint cycle in any phylogenetic network. Perhaps more important than the specific results about galled-trees, we introduce an approach that can be used to study recombination in phylogenetic networks that go beyond galled-trees.

This paper greatly extends the conference version that appears in [7]. PowerPoint slides of the conference talk can be found at [6].

1 Introduction to phylogenetic networks and galled-trees

A phylogenetic network is a generalization of a phylogenetic tree, allowing structural properties that are not tree-like. With the growth of genomic data, much of which does not fit ideal tree models, and the increasing appreciation of the genomic role of such phenomena as recombination, recurrent and back mutation, horizontal gene transfer, gene conversion, and mobile genetic elements, there is greater need to understand the algorithmics and combinatorics of phylogenetic networks [17, 18]. Recombination is particularly important, because it is the key element needed for techniques that are widely hoped to locate genes influencing genetic diseases. The key to locating these genes is to understand and use the patterns of recombination in the genetic "experiments" done by nature and history. However, to date, very little has been published on the combinatorial structure of phylogenetic networks, with the notable exception of the paper by Wang et al.[21]. Other related papers include [9, 10, 12, 20, 19].

1.1 Formal definition of a phylogenetic network

There are five components needed to specify a phylogenetic network: a directed acyclic graph (no directed cycles, but the underlying undirected graph can have cycles); an assignment of mutations or sites (integers) to edges; an assignment of a sequence to each non-recombination node; a specification of a prefix-sequence and a suffix-sequence for each recombination node; and an assignment of a recombination point and a recombinant sequence to each recombination node. We will define each of these components in turn. See Figure 1 for an example of a phylogenetic network.

Definition: An (n, m)-phylogenetic network N is built on a directed acyclic graph containing exactly one node (the root) with no incoming edges, a set of internal nodes that have both incoming and and outgoing edges, and exactly n nodes (the leaves) with no outgoing edges. Each node other than the root has either one or two incoming edges. A node x with two incoming edges is called a "recombination" node.

Each integer (site) from 1 to m is assigned to exactly one edge in N, but for simplicity of exposition, none are assigned to any edge entering a recombination node. It is also possible that other edges will receive no integer assignment. We use the terms "column" and "site" interchangeably.

Each node in N is labeled by an m-length binary sequence, starting with the root node which is labeled with the all-0 sequence. Since N is acyclic, the nodes in N can be topologically sorted into a list, where every node occurs in the list only after its parent(s). Using that list, we can constructively define the sequences that label the non-root nodes, in order of their appearance in the list, as follows:

a) For a non-recombination node v, let e be the single edge coming into v. The sequence labeling v is obtained from the sequence labeling v's parent by changing from 0 to 1 the value at position i, for every integer i assigned to edge e. This corresponds to a mutation at site i occurring on edge e.

b) Each recombination node x is associated with an integer r_x (denoted r, when x is clear by context) between 2 and m inclusive, called the "recombination point" for x. For the recombination at node x, one of the two sequences labeling the parents of x must be designated P and the other designated S. Then the sequence labeling x consists of the first $r_x - 1$ characters of P, followed by the last $m - r_x + 1$ characters of S. Hence P contributes a *Prefix* and S contributes a *Suffix* to x's sequence. The resulting sequence that labels x is called a "recombinant sequence".

The sequences labeling the leaves of N are the extant sequences, i.e., the sequences that can be observed. In this paper, the ancestral sequence (at the root of the phylogenetic network) is always the all-0 sequence, and all results are relative to that assumption. We will sometimes restate this for emphasis.

Definition 1.1 *An (n, m)-phylogenetic network N derives (or explains) a set of n sequences M if and only if each sequence in M labels exactly one of the leaves of N.*

What we have defined here as a phylogenetic network is often referred to as an "ancestral recombination graph" in the population genetics literature (see [16] for a typical example).

The biological interpretation of a phylogenetic network N that derives M is that N is a possible history of the evolution of the sequences in M, under the assumptions that there is a single, known ancestral sequence (assumed to be all-0 for convenience); that for any site in the sequences there is exactly one point in the history (recorded on an edge) where that state of that site mutates (due to a point-mutation) from 0 to 1; and that two sequences are permitted to recombine in an equal-crossover event. Recombinations occur at a recombination node and this distinguishes a change of state due to recombination from a change of state due to mutation. With these definitions, a classic perfect phylogeny is a phylogenetic network which is topologically a directed, rooted tree, i.e., lacking any cycles in the underlying (undirected) graph.

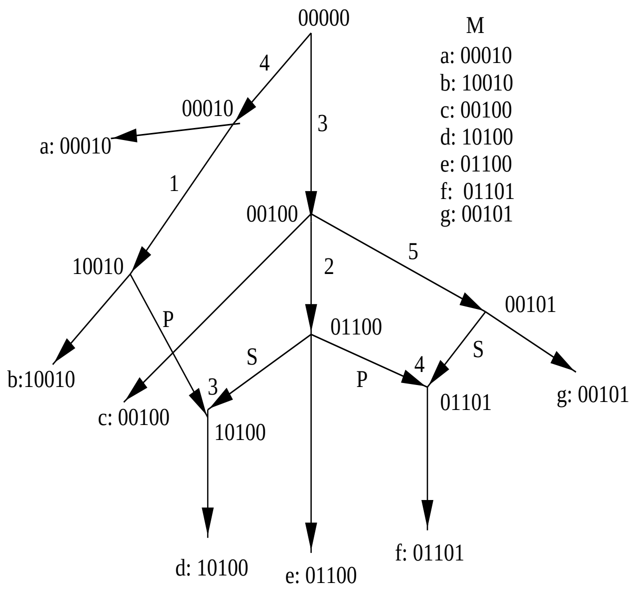


Figure 1. A phylogenetic network N with two recombination nodes. The matrix of sequences M that are derived by N is shown at the right. Note that the node with sequence label 01100 is sequence S for the left recombination node, and is sequence P for the right recombination node. The recombination points are 3 and 4 for the left and right recombination nodes respectively, and are written just above the recombination nodes . In this example, every label of an interior node also labels a leaf, but that is not a general property of phylogenetic networks.

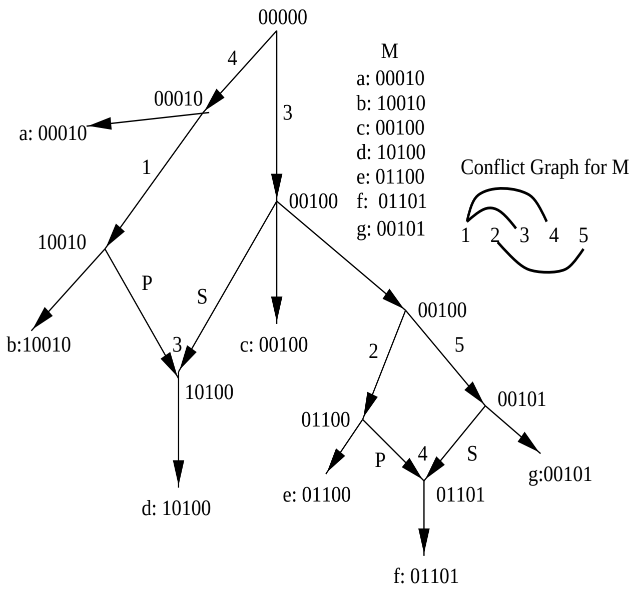


Figure 2. A galled-tree deriving the same sequences as the phylogenetic network in Figure 1. Unlike the example shown here, in general the recombinant sequence exiting a gall may be on a path that reaches another gall.

It is important to note that when recombination is allowed, the order of the sites in the sequences constrain the possible outcomes of a recombination event, and so the given order of the sites in a sequence is a critical feature of a problem instance. This is in contrast to the classic perfect phylogeny problem, and to recent work on perfect phylogenies with hybridization [15], where the order of the characters does not matter.

Interest in phylogenetic networks comes partly from a desire to reconstruct the evolutionary history under a model that is more biologically complete than the perfect phylogeny model. But there also more applied uses of phylogenetic networks. For example, in a population of "unrelated" individuals, we want to determine which parts of the individuals genomes came from a common ancestor. This determination helps locate regions in the genome associated with genes contributing to an observable trait (for example, a disease). Recombination in the population is key to this determination, and understanding the history of the recombinations is the key to doing this kind of mapping.

Motivation for binary sequences comes from a number of sources, but the strongest current motivation comes from data where each site is a single SNP (single nucleotide polymorphism), i.e., a site where two of the four possible nucleotides appear in the population with a frequency above some set threshold.

1.2 Which Phylogenetic Networks are Biologically Informative?

It is easy to show that for every binary matrix M, there is a phylogenetic network N that derives M using $O(nm)$ recombination nodes, but that is not of great interest because in most evolutionary histories the number of observable recombinations is thought to be relatively small. Hence a more biologically informative problem is to find, for input M, a phylogenetic network that derives M, and that either has some biologically-motivated structure, or uses the *minimum* number of recombinations, over all phylogenetic networks with all-0 ancestral sequence. We call that number m_M. Wang et al. [21] showed that the general problem of computing m_M is NP-hard, and Hudson and Kaplan [11], Myers and Griffiths [14], and Song and Hein [19] give combinatorial methods for computing lower-bounds on m_M.

1.2.1 Galled-trees: A biological and algorithmically motivated structural restriction

Given the NP-hardness of the problem of computing m_M, Wang et al. suggested a structural restriction on the permitted phylogenetic networks which has both biological and algorithmic appeal.

Definition 1.2 *In a phylogenetic network N, let w be a node that has two paths out of it that meet at a recombination node x. Those two paths together define a "recombination cycle" Q. Node w is called the "coalescent node" of Q, and x is the recombination node of Q.*

Definition 1.3 *A recombination cycle in a phylogenetic network that shares no nodes with any other recombination cycle is called a "gall" (imagine a wasp's gall in a tree). We say a site i "is on" or "appears" on a gall Q if i labels one of the edges of Q. When i appears on Q, we also say that "Q contains i". We use the term "recombination cycle" for phylogenetic networks.*

Definition 1.4 *A phylogenetic network is called a "galled-tree" if every recombination cycle is a gall. See Figure 2.*

Galled-Tree Problem: Given a set M of n binary sequences, each of length m, determine if there exists a galled-tree T that derives M, and if there is one, construct one.

Wang et al. [21] give an $O(nm + n^4)$-time algorithm that was intended to solve the Galled-Tree Problem. This work is seminal as it is the first paper to introduce a biologically motivated structural restriction for a phylogenetic network that allows a polynomial time algorithm. Unfortunately, the algorithm in [21] is incorrect, and only provides a sufficient test for the existence of a galled-tree for M.

1.3 Main results

In this paper we develop a faster algorithm ($O(nm + n^3)$-time) that completely solves the Galled-Tree Problem. In particular, the algorithm produces a galled-tree of a particular form, called a "reduced galled-tree".

We will show that if there is a galled-tree for M, then there is a reduced galled-tree for M, and that every reduced galled-tree for M uses exactly m_M galls and recombinations. Moreover, when there is a galled-tree for M, even if multiple-crossover events are allowed at each recombination event, there is no phylogenetic network that derives M using fewer recombination events. We show that reduced galled-trees are "essentially-unique", allowing only easily characterized variation. Thus if the sequences did derive historically on a galled-tree, the algorithm will correctly capture the essential features of that history.

The algorithm can be used to count and represent all the reduced galled-trees for M, and to produce each one efficiently. We also show that if M can be derived on a galled-tree, then it can be derived on a true tree (without underlying cycles) with at most one back mutation per site, and that the problem of removing the minimum number of sites of M, so that the remaining sites have a perfect phylogeny (an NP-hard problem in general) can be solved in polynomial time.

In obtaining these results, we develop combinatorial constraints that apply to galls in any phylogenetic network (whether a galled-tree or not). This is useful as a first step in understanding phylogenetic networks in general, and for specific tasks, such as proving that a given site cannot be on any gall in any phylogenetic network. Some of these general constraints are more fully examined in [8].

1.4 Motivation for Galled-Trees

There are several reasons for interest in galled-trees. A galled-tree defines a phylogenetic history where the recombination cycles are node-disjoint, using a modest number of recombinations, at most $m/2$. A phylogenetic network is likely to be a galled-tree if the level of recombination is moderate, or if most of the observable recombinations are recent. In Human populations, both conditions are believed to hold. Other examples of galled-trees arise in the data reported in [13]. The simplest situation is the case of an interval in the genome where only a single recombination has occurred. In that case, the true history of the sequences in that interval takes the form of a galled-tree, and our algorithm will correctly reconstruct the essential features of that history. More generally, it is important (in disease association studies, for example) to find regions of the genome where the subsequences in a population exhibit moderate recombination, and the galled-tree algorithm can be used to find such regions.

Further motivation for galled-trees comes from the fact that if M can be derived on a galled-tree, then it can be derived on a true tree (no underlying undirected cycles) with at most one back mutation per site. A tree with limited back mutations is another model of interest that deviates from the perfect phylogeny model. But perhaps the most compelling motivation for galled-trees comes from the main results in this paper, namely that when a set of sequences M can be derived on a galled-tree, an efficient algorithm will find such a galled-tree, and the galled-tree constructed uses the minimum number of recombinations over any phylogenetic network for M (with the all-0 ancestral sequence), even phylogenetic networks that allow multiple-crossover events at each recombination node. Hence, galled-trees provide the only known non-trivial case where optimal (with respect to the number of recombinations) phylogenetic networks can be efficiently constructed from a set of sequences.

2 Combinatorial definitions and observations

We organize M into a matrix, where each row contains a sequence in M, and assume there are no duplicate columns, and that each column has at least one entry that is 1.

2.1 Combinatorial Background and Major Combinatorial Tool

Definition 2.1 *Two columns (or sites) in M are said to "conflict" if and only if the two columns contain three rows with the pairs 1,1; 0,1; and 1,0. A site is called "conflicted" if it is involved in at least one conflict, and is otherwise called "unconflicted".*

Recall that a perfect phylogeny is a phylogenetic network without recombinations. Hence, as a graph, it is a directed rooted tree. The following is the classic necessary and sufficient condition for the existence of a perfect phylogeny deriving a set of sequences M. See [3, 4] for one exposition.

Theorem 2.1 *There is a perfect phylogeny deriving M if and only if matrix M contains no conflicted sites. Further, if there is a perfect phylogeny for M and all columns of M are distinct, then there is a unique perfect phylogeny for M, and each edge is labeled by at most one site. If there are identical columns then the perfect phylogeny is unique up to any ordering given to multiple sites that label the same edge.*

Hence it is the existence of conflicts in M that require a deviation from the perfect phylogeny model, and in this paper, require recombinations in order to derive a history of M. We will show that when there is a galled-tree for M, there is a galled tree of a particular form, as follows:

Definition 2.2 *A galled-tree is called "reduced galled-tree" if every gall contains at least one conflicted pair of sites, and contains no unconflicted sites.*

Major Tool: The Conflict Graph and its Connected Components

The central technical contribution of this paper is to observe that there is combinatorial structure in the pattern of conflicts between columns, and that this structure can be represented and exploited to obtain insights about recombination in phylogenetic networks. We now introduce the *conflict graph*, which represents and exposes some of the combinatorial structure.

Definition 2.3 *The conflict graph G contains one node for each site in M. We label each node of G by the site it represents. Two nodes i and j are connected by an undirected edge if and only if sites i and j conflict. See Figure 2.*

Overview: The connected components of G reveal important structural information about galled-trees. We will show that there is a one-one correspondence between the non-trivial connected components of G and the galls in a reduced galled-tree. More generally, every gall in any phylogenetic network that contains some conflicted sites, contains all the sites of one non-trivial connected component, and contains no sites from another non-trivial connected component. Further, no gall need contain any unconflicted sites. It follows that every reduced galled-tree for M (if there is a galled-tree ofr M) uses the same number of galls, and the same number of recombinations. We will show that this number is m_M, the minimum number of recombinations needed by any phylogenetic network that derives M, and has the all-0 ancestral sequence.

2.2 Combinatorial Constraints on Galls

In order to prove the claims made in the overview, we begin an examination of the combinatorial constraints on galls and galled-trees.

Definition 2.4 *If a node q is reachable from a node p via a directed path, then p is an ancestor of q, and q is a descendant of p.*

Lemma 2.1 *Let Q be a gall in a phylogenetic network N, and assume that Q contains a site i. Let v be a node on Q and define N' as the subnetwork of N consisting of all nodes and edges reachable by directed paths from v, not using any edges in Q. That is, N' is the maximal subnetwork of N branching off of Q at v. Then the state (0 or 1) of site i at every node in N' is the same as at node v.*

Proof Suppose that at some node in N', the state of i is different than it is at v. Let x' be such a node with the property that at every ancestor of x' in N', the state of i is the same as at node v. Since i only mutates once, and not on an edge in N', the state of i cannot change in N' due to mutation, and can therefore only change due to recombination. Hence, x' must be a recombination node. Now if both parents of x' were in N', then by the choice of x', the state of i at both parents would be the same as the state at v, and that state would be unchanged at x' regardless of where the recombination point $r_{x'}$ is. So one of the parents of x', call it p, must be outside of N'. By definition of x' and N', all paths from the root of N to x' either avoid v, or go through the edge on Q out of v. In the first case, let y be the last ancestor of v on that path. Then the path from y to x' that goes through p, together with the path from y to x' through v (and N') form a recombination cycle that shares an edge with Q. In the second case, let y be the child of v on Q. Then the path from v to x' in N' together with the path from v to x' through y form a recombination cycle that shares an edge with Q. Both contradict the assumption that Q is a gall, and so the state of site i at every node in N' must be the same as at node v. $\square$

In a similar way, we can prove

Lemma 2.2 *If a site mutates on a directed edge (v, v') that is not on any gall, then the state of i is the same at any node reachable from v'.*

Definition 2.5 *Let C be a set of sites on a gall Q, and let the matrix $M(C)$ be matrix M restricted to the sites in C. Let $M(C) - 0$ denote the sequences in $M(C)$ that are unequal to the all-0 sequence. Given a phylogenetic network N for M, let $S_v(C)$ denote the sequence labeling node v in N, restricted to the sites in C.*

Note that $M(C) - 0$ equals $M(C)$ if the all-0 sequence is not in $M(C)$. Lemma 2.1 implies the following

Corollary 2.1 *For the coalescent node w of Q, $S_w(C)$ is the all-0 sequence. A sequence is in $M(C) - 0$ if and only if it is the sequence $S_v(C)$ for some node $v \neq w$ on Q. Stated differently, the node labels of the non-coalescent nodes on Q, restricted to sites in C, are exactly the sequences in $M(C) - 0$.*

Proof Clearly, $S_w(C)$ is the all-0 sequence, since no sites on C mutate on the path(s) from the root of N to w. The sequences in $M(C)$ are the sequences labeling the leaves of N, restricted to C. If a leaf z is reachable from a node v in Q, not using an edge in Q, then by Lemma 2.1, $S_z(C)$ and $S_v(C)$ are the same. If leaf z is not reachable from any node v in Q, then it must have state 0 for every site i that mutates on Q. In that case $S_z(C)$ is all zeros. □

Corollary 2.1 is important because it says that information about the node labels on a gall is reflected in the sequences at the leaves, and hence that information is contained in extant sequences. This is a property of galls that does not generalize to every non-gall recombination cycle, and is intuitively one of the reasons why problems concerning galls and galled-trees have efficient solutions.

Definition 2.6 *A node v on a recombination cycle Q is called a "branching node" if there is a directed edge (v, v') where v' is not on Q. The edge (v, v') is called a "branching edge".*

The following theorem is the technical key to most of the analysis of the combinatorial structure of galled-trees.

Theorem 2.2 *Let T be a galled-tree for matrix M. Two sites i and $j > i$ in M conflict if and only if the following conditions hold:*

a) i and j are together on the same gall (call it Q) in T, with recombination node x, and $i < r_x \leq j$.

b) Sites i and j are arrayed on Q in one of the following three ways (see Figure 3):

W1: Site i is on the P-side and j is on the S-side of Q, and there is a branching node between i and x, and a branching node between j and x. Note: In this case, the i, j state-pair in the recombinant sequence is 1,1.

W2: Sites i and j are both on the P-side with j above i (i.e., j mutates before i does), and there is a branching node between j and i, and a branching node between i and x. In this case the i, j state-pair in the recombinant sequence is 1,0.

W3: Sites i and j are both on the S-side with i above j, and there is a branching node between i and j, and a branching node between j and x. The state-pair in this case is 0,1.

Proof of Theorem 2.2 We first establish the necessary direction, i.e., starting with the assumption that i and j conflict in M. We prove a general fact, namely that if i and j conflict and N is a phylogenetic network for M (not necessarily a galled-tree), then i and j must be together on some recombination cycle in N. That fact, specialized to galled-trees, implies the first claim in the necessary direction of part a). The proof is by contradiction.

Suppose the sites i and j are not together on some recombination cycle in N. This could either be because they are on separate recombination cycles, or because one or both are on edges outside of any recombination cycle. Let us remove all the sites from N except for i and j. Certainly, the resulting phylogenetic network derives the sequences M, restricted to the two sites i and j. Now contract any edge e unless e is labeled with i or j, or e is on a recombination cycle that contains either i or j. The resulting network N' again correctly derives the sequences of M, restricted to sites i and j. Network N' is either a perfect phylogeny (if it contains no recombination cycles), or it is a phylogenetic network with recombination, where each recombination cycle contains exactly one mutation, either i or j.

If N' contains recombination cycles, we further modify it. Suppose i is on a recombination cycle Q with recombination node x. If the state of i is 0 at x (through recombination), then we remove the edge into x on the side of Q that i is on. If the state of i is 1 at x, then we remove the edge into x that is on the side opposite the side that i is on. In either case, the result is a network with one fewer recombination cycles whose leaves are labeled exactly as in N'. Continue to apply this transformation (for i or j) to any remaining recombination cycle in N', until there are no remaining recombination cycles. The result is a perfect phylogeny which correctly derives the sequences in M, restricted to sites i and j. But a perfect phylogeny exists for a set of sites if and only if no pair of those sites conflict. This implies that i and j do not conflict, a contradiction. Hence, i and j must be together on some recombination cycle in N. Specialized to the case when M is derived on a galled-tree T, sites i and j must be together on one gall of T. This proves the first claim in the necessary direction of part a).

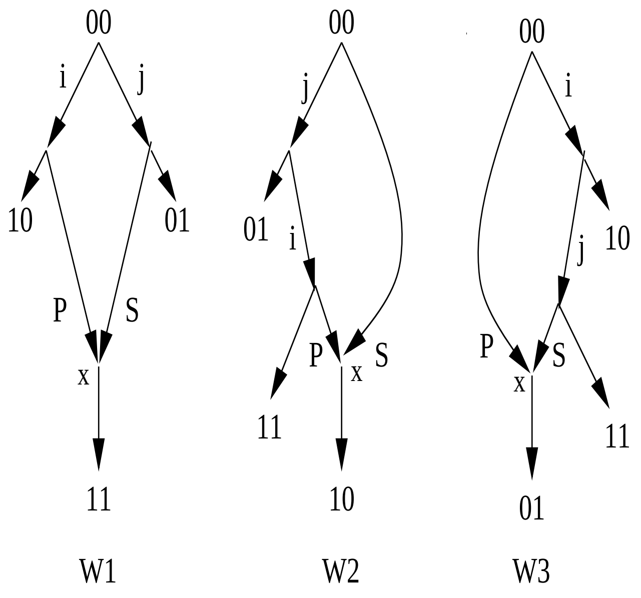


Figure 3. The three cases for Theorem 2.2. In each case, the recombination point r_x is between i and $j > i$.

Let r be the recombination point of gall Q. We will show that $i < r \leq j$ is forced. First, by Corollary 2.1 and the assumption that i and j conflict, and the fact (just established) that i and j are together on some gall Q, it follows that all three of the required (i,j) state-pairs must occur at branching nodes of Q. It also follows that the (i,j) state-pair at the recombination node x of Q must be different from the (i,j) state-pairs at any other node on Q, for those nodes can only have state-pairs 0,0 and two of the three pairs 0,1; 1,0; and 1,1, since i and j each only mutate once. Now suppose for contradiction that $i < j < r$. If i and j are both on the P-side of Q, then the (i,j) state-pair at x will be $(1,1)$ which already occurs at some branching node above x. If both are on the S-side, then the state-pair at x will be 0,0. If i and j are on different sides of Q, then at x, the state-pair will be identical to what it is at the last node on the P-side, above x. In all cases, the state-pair at x will be identical to some state-pair at a different node on Q, contradicting the assumption that i and j conflict. The argument for when i and j are larger than r is symmetric. Hence it must be that $i < r \leq j$. This completes the proof of the necessary direction for part a).

Now we prove the necessary direction for part b). Since sites i and j conflict, there must be three sequences in M, one with (i,j) state-pair 0,1; one with pair 1,0; and one with pair 1,1. We established above that each of these three (i,j) state-pairs must appear in a sequence labeling some node on Q. So we consider how these pairs could be placed on Q. If both i and j are on the P-side of Q, and i appears before j, then the nodes other than x will have the (i,j) state-pairs 1,0 or 1,1, and node x will also have pair 1,0, so the pair 0,1 will be missing. Similarly, if i and j both appear on the S-side, then i must appear above j, or else the pair 1,0 will be missing. If i and j appear on different sides, with j on the P-side, and i on the S-side, then state-pairs 0,1 and 1,0 will appear on nodes of Q, but pair 1,1 will be missing. Hence the only possibilities for the relative placement of mutations i and j are the ones given in statements $W1, W2$ and $W3$. To finish proving the necessary side of b) we need to establish that the branching nodes on Q must be as claimed in statements $W1, W2$ and $W3$. Consider statement $W1$. If the claimed branching node between j and the recombination node x is missing, then no edge out of Q would pass on the (i,j) state-pair 0,1, and so by Corollary 2.1, no leaf sequence would have that pair, contradicting the assumption that i,j conflict. Similarly if there is no branching node between i and x, then the pair 1,0 would not appear at any leaf. Hence the branching nodes claimed in statement $W1$ must exist. The proofs for statements $W2$ and $W3$ are similar, and left to the reader. This concludes the proof of the necessary side of b).

To prove the sufficient direction of the theorem, assume i and j are together on some gall Q with recombination node x, where $i < r_x \leq j$, and i and j are arrayed in one of the three ways enumerated in statement b). Note (by inspection) that in each of the relative placements of i and j enumerated by $W1, W2$ and $W3$, all three (i,j) state-pairs 0,1; 1,0; and 1,1 are found at nodes on Q. Then, with the specified branching edges off of Q, and Corollary 2.1, each of these pairs is found at a leaf of T, and hence in some sequence in M. Therefore, no matter whether the (i, j) configuration is as given in $W1, W2$ or $W3$, sites i and j will conflict in M. $\Box$

Note that in all cases, the i, j state-pair at x differs from the i, j state-pair at every other node on Q.

In the case of phylogenetic networks (not necessarily galled-trees), parts of Theorem 2.2 continue to hold.

Theorem 2.3 *Let N be a phylogenetic network for M. Suppose sites i and $j > i$ are together on some gall Q (with recombination node x) in N. Then sites i and j conflict if and only if $i < r_x \leq j$, and one of the conditions W1, W2 or W3 hold.*

We will not use Theorem 2.3 in this paper, and leave the proof to the reader. A deeper analysis of galls in general phylogenetic networks is developed in [8].

The algorithm in [21] is only a sufficient test for the existence of a galled-tree that derives M, because it (implicitly) assumes that a pair of sites can conflict only due to arrangement W1. Equivalently, the algorithm in [21] correctly determines whether or not the input sequences can be derived on a galled-tree T having the following added constraint: for each site i, if site i mutates on an edge e, then the state of i remains set at 1 at all nodes which are reachable from the end of e. Hence once the state of i mutates from 0 to 1, it never returns to 0, even through the action of recombination. That is a severe restriction compared to what is allowed by the general definition of a galled-tree. In the galled-tree in Figure 2, the state of site 4 mutates from 0 to 1, but then is returned to 0 through recombination in the gall shown on the left.

We now develop the theorems leading to the one-one correspondence between connected components in the conflict graph for M and the galls in a galled-tree for M.

Theorem 2.4 *For any non-trivial connected component C of the conflict graph, and any galled tree T for M, all the sites in C must occur together on a single gall in T.*

Proof This follows by transitivity from the necessary direction of part a) of Theorem 2.2, and the fact that for any pair of sites i and j in C, there must be a path connecting i to j in C. $\Box$

The following theorem is the complement to Theorem 2.4.

Theorem 2.5 *Let T be a galled-tree for M. If sites i and i' are on different non-trivial connected components of the conflict graph, then they must appear on different galls of T.*

We prove Theorem 2.5 by using the following lemma, which is of interest in its own right.

Lemma 2.3 *Let Q be a gall in T with recombination node x, and recombination point r, and let i, i', j, j' be sites on Q, where i conflicts with $j > i$ and i' conflicts with $j' > i'$. Then either i conflicts with j', or i' conflicts with j.*

Proof First, by Theorem 2.2, part a) i, i' must both be smaller than the recombination point r of Q, and j, j' must both be greater or equal to r.

Using Theorem 2.2, we consider the three ways that i and j can be arrayed on Q. In each case we will show that either i', j must conflict, or i, j' must conflict.

Case 1) Sites i and j are arrayed as in W1, so site i is on the P-side, and site j is on the S-side of Q, and there is a branching node below each, but above x. Sites i' and j' conflict, so no matter where they are on Q, there must be a branching node below i' and one below j', but above x. Now to avoid conflict with j, site i' must be on the S-side of Q. But then to create conflict between i' and j', site j' must also be on the S-side. That puts i on the P-side and j' on the S-side, with a branching node below each, and hence by the sufficient direction of Theorem 2.2 case W1, sites i and j' must conflict.

Case 2) Sites i and j are arrayed as in W2, so i and j are both on the P-side, with j above i, and there is a branching node between them. Since j' conflicts with i', there must be a branching node below j'. Hence, by Theorem 2.2, to avoid conflict with i, site j' must be on the P-side, either below i, or above i but below the branching node between j and i. Either way, j' must be below j on the P-side. But then by Theorem 2.2, because i' and j' conflict, site i' must be on the P-side below j', and there must be a branching node below i'. Therefore, j and i' are arrayed on Q as in case W2, and hence by Theorem 2.2 they conflict.

Case 3) Sites i and j are arrayed as in W3, so both are on the on the S-side, with i above j, and there is a branching node between them. Since i' conflicts with j', there must be a branching node below i'. Hence, by Theorem 2.2, to avoid conflict with j, site i' must be on the S-side, either below j, or above j but below the branching node between i and j. Either way,

i' must be below i on the S-side. But then by Theorem 2.2, because i' and j' conflict, site j' must be on the S-side below i', and there must be a branching node below j'. Therefore, i and j' are arrayed on Q as in case W3, and hence they conflict. □

Lemma 2.3 can be generalized to the case where no ancestral sequence has been specified in advance. This has been done (independently) in Proposition 1 in [19].

Proof of Theorem 2.5: Let i and $j > i$ be a conflicting pair of sites on one non-trivial connected component, and i' and $j' > i'$ be a conflicting pair of sites on another non-trivial connected component. If these four sites are all together on a single gall Q, then with respect to the recombination point r of that gall, i and i' are below r, and j and j' are each equal to or above r. So by Lemma 2.3, either i conflicts with j' or i' conflicts with j. But that contradicts the assumption that i and j are on a different connected component of the conflict graph than are i' and j'. Hence i and j are on one gall and i' and j' are on another. But by Theorem 2.4, all sites on the same connected component are together on a single gall, so any two sites on two different connected components are on different galls. □

Theorems 2.4 and 2.5 together imply

Theorem 2.6 *In any galled-tree T for M, there is a one-one correspondence between the non-trivial connected components of the conflict graph and the galls in T containing conflicted sites. Each such gall in T contains all the sites of one non-trivial connected component (due to Theorem 2.4), and contains no sites from a different non-trivial connected component (due to Theorem 2.5).*

Theorem 2.6 is crucial for the efficiency and optimality of the algorithm to construct galled-trees, when possible. It also leads to the following

Theorem 2.7 *If the conflict graph for M has k non-trivial connected components, and there is a galled-tree for M, then any galled-tree that minimizes the number of recombinations (over all galled-trees) uses exactly k recombinations. Moreover, any galled-tree for M where each gall contains some conflicted sites, uses exactly k recombinations.*

Proof Let T be a galled-tree for M. If there is a gall Q in T that only contains unconflicted sites, the sequences labeling the nodes on Q can be derived on a perfect phylogeny where the root of the phylogeny is the sequence labeling the coalescent node of Q. Replacing Q with the perfect phylogeny results in a galled-tree for M that uses one fewer recombinations than T. Hence in any galled-tree using the minimum number of recombinations, every gall contains at least one conflicting pair of sites. Therefore, the number of galls is exactly the number of non-trivial connected components in the conflict graph. □

Theorems 2.4 and 2.5 generalize to phylogenetic networks as follows.

Theorem 2.8 *Let N be a phylogenetic network for sequences M. Every gall (if there is one) in N that contains conflicted sites contains all the sites of one non-trivial connected component of the conflict graph, and contains no sites from another non-trivial connected component.*

3 Arranging the gall Q

The one-one correspondence between non-trivial connected component and galls in a reduced galled-tree greatly simplifies the task of creating a galled-tree for M. We can focus independently on each non-trivial connected component C of the conflict graph, to determine how the sites on that component are arrayed on the gall Q, and how to select the recombination point for Q. In this section we show how to efficiently accomplish these tasks.

3.1 Selecting the recombination point r on Q

Lemma 3.1 *If there is a galled-tree for M, then every non-trivial connected component C of the conflict graph must be bipartite, and the bipartition is unique: the (indices of the) sites on one side of the bipartite graph must be strictly smaller than the sites on the other side.*

Proof All the sites on C must mutate on a single gall Q, and Q has only a single recombination point r. By Theorem 2.2a), $i < r \leq j$ for any conflicting pair i, j in C where $i < j$. Therefore, each edge in C connects one site whose index is below r and one site whose index is at or above r. □

Constructively, it is easy to find the bipartition described in Lemma 3.1: let p be the largest (by index) node in C which is connected only to larger nodes in C, and let q be the smallest node in C which is connected only to smaller nodes. Then all the sites with index p or less are on one side of the bipartite graph, and all the sites with index q or larger are on the other side of the graph. Moreover, the recombination point r for the gall associated with the connected component C, can be chosen to be any point in the interval $[p+1, q]$.

Definition 3.1 *The interval* $[p+1, q]$ *is called the "recombination interval" of* C.

Lemma 3.1 gives a necessary condition that can be used to prove that certain sets of sequences cannot be derived on a galled-tree. For example, see Figures 4 and 5.

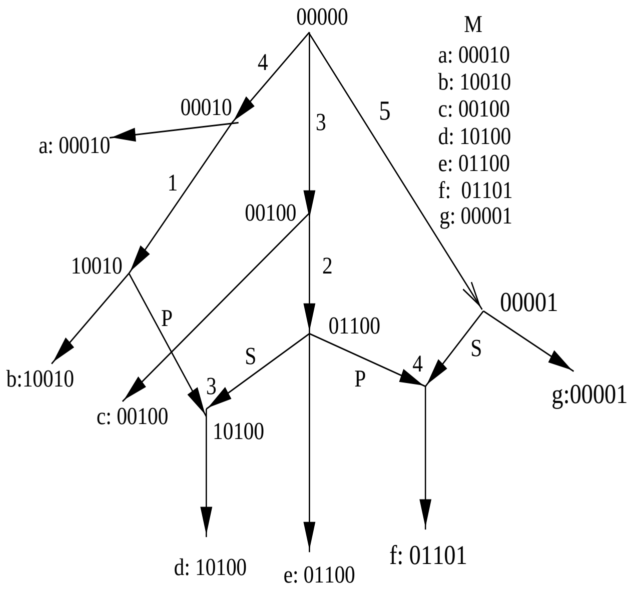


Figure 4. The phylogenetic network derives the sequences M shown to the right. Although this network is similar to the one shown in Figure 1, and only sequence g is different, M cannot be derived on a galled-tree. See Figure 5.

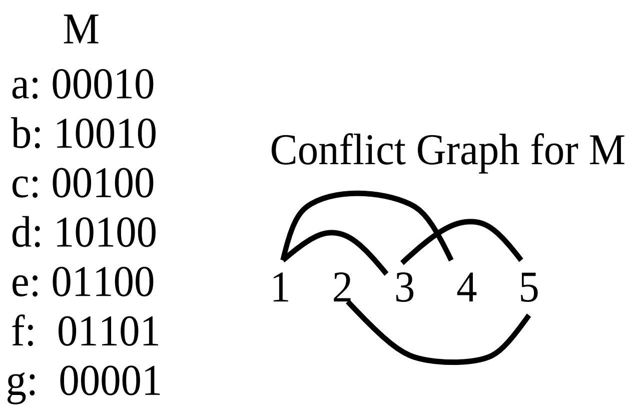


Figure 5. The conflict graph of the sequences M from Figure 4 is bipartite, and the bipartition is unique, but it does not have the required properties stated in Lemma 3.1.

By a much more detailed analysis of the combinatorial structure of galls, in [8], we prove a stronger result than Lemma 3.1:

Theorem 3.1 *Let N be an arbitrary phylogenetic network for M. The sites in a connected component C can appear on a gall in N only if C is a bipartite graph with the bipartition described in Lemma 3.1, and C is a* **bi-convex** *graph. A bipartite graph is bi-convex if the nodes of the graph can be renumbered so that for any node v, the set of nodes that v is adjacent to form a contiguous interval in the new node numbers.*

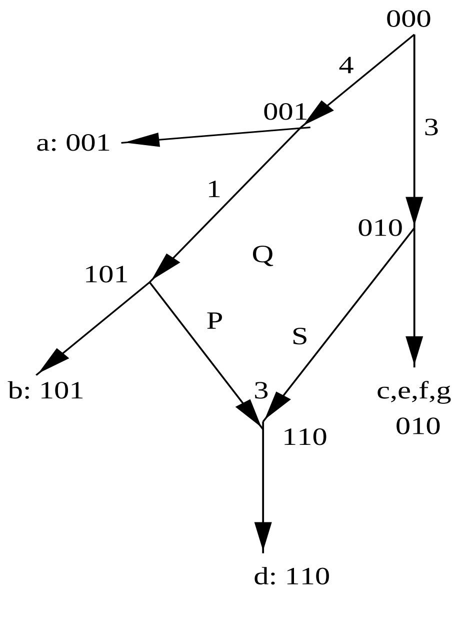
This is a very useful, general theorem since it allows us to identify more connected components whose sites cannot appear on a gall in any phylogenetic network. One can determine if a graph with m nodes is bi-convex in $O(m^2)$ time and also find a minimum *node cover* of a bi-convex graph in $O(m^2)$ time [2]. We conjecture that a linear time bound is obtainable. It is easy to see that the minimum number of columns to remove from M so that no conflicts remain, is given by the minimum node cover of the conflict graph. This is called the "site consistency" problem, and it is NP-hard in general [1]. However, the node cover problem can be solved in polynomial time (by network flow) on any bipartite graph. So when there is a galled-tree for M, the site-consistency problem can be solved by network flow in polynomial time, and even faster by exploiting the fact that each connected component must be bi-convex.

3.2 Arranging the sites of C on Q

We now describe how to arrange the sites of C on a gall Q. Corollary 2.1 will be a central tool.

To understand the method for arranging the sites on a gall, consider a fixed galled-tree T for M, and focus on the arrangement of sites of C on gall Q in isolation of the rest of T. Now remove the recombination node x from Q, and the two edges entering x The resulting graph consists of one or two directed paths starting at the coalescent node of Q, and containing all the sites in C. If it only contains one path, then denote the coalescent node as y, and the single end node as u; otherwise let u and y be the two end nodes of the two paths. For each node v other than the coalescent node, add an edge from v branching off of Q, and label its leaf end with $S_v(C)$. The result is a perfect phylogeny, denoted $T(Q)$, that of course, derives the sequences labeling the leaves of $T(Q)$. Further, $S_x(C)$ can be formed by a recombination of the sequences $S_u(C)$ and $S_y(C)$ at the recombination point r determined from C.

Now by Corollary 2.1, the leaf labels of $T(Q)$, restricted to the sites in C, are exactly the sequences in $M(C)$, other than the sequence $S_x(C)$. That is, tree $T(Q)$, with sites restricted to the sites in C, is a perfect phylogeny for all the sequences in $M(C)$ other than $S_x(C)$. This fact is the key to how we can actually find $T(Q)$ knowing only $M(C)$. See Figures 6 and 7.



M

a:	00010
b:	10010
c:	00100
d:	10100
e:	01100
f:	01101
g:	00101

M(C)

a:	001
b:	101
c:	010
d:	110
e:	010
f:	010
g:	010

Figure 6. Q is the gall for the connected component C in Figure 2 containing sites 1,3, and 4. The node labels shown on the gall are the node labels from Figure 2, restricted to the sites in C. The sequences in M, restricted to the sites in C are also shown. Note that the set of node labels and the set of sequences in $M(C)$ are identical, which is assured by Corollary 2.1.

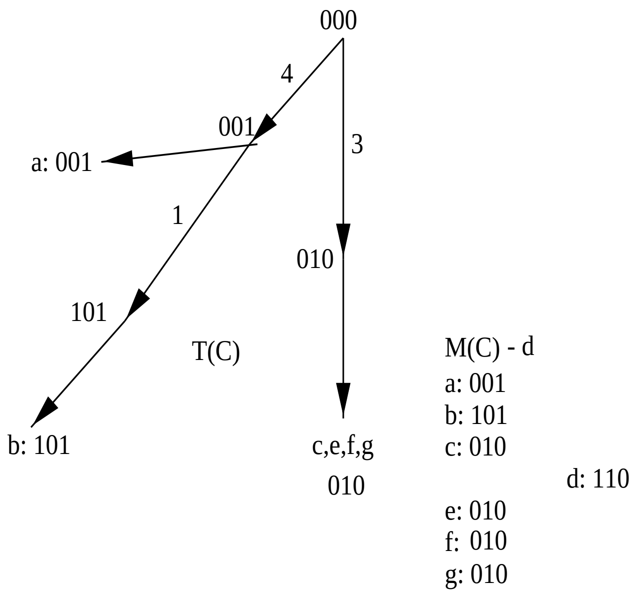


Figure 7. After removal of the recombinant node from gall Q of Figure 6, the resulting graph is a perfect phylogeny for the sequences $M(C) - 110$. Moreover, the mutations on the perfect phylogeny are organized into two paths, and the removed sequence 110 can be created by recombining the two sequences at the ends of those two paths, using the recombination point 3 determined earlier from C, as shown in Figure 6.

Hence, we have

Theorem 3.2 *There is a sequence X in $M(C)$, such that after removal of all copies of X, there is a perfect phylogeny for the resulting matrix; the labeled edges of that perfect phylogeny contain all sites in C organized into one or two paths; and the recombination of the two "end" sequences (from either the root sequence of the perfect phylogeny and the single leaf sequence, or the two leaf sequences) at the recombination point r creates sequence X.*

Recall that the recombination point r can be computed from C, so Theorem 3.2 suggests an effective procedure. However, since C might contain a strict subset of the sites in M, the meaning of r has to adjusted to correctly indicate the correct crossover point for the sites in C. When applied to a subset of sites C, any index in C which is less than r will be to the left of the recombination, and any index in C which is equal or larger than r will be to the right of the recombination.

Now, by Theorem 2.1, any matrix which has a perfect phylogeny has a unique perfect phylogeny as long as no ordering is given to multiple sites on the same edge. Hence, given M and C, if we could guess X, we could create the correct, unique, perfect phylogeny and exactly recreate the arrangement of sites on Q as given in T.

However, since we do not know X, if we remove from $M(C)$ all copies of a different sequence Y, and yet there is a (unique) perfect phylogeny for the resulting matrix, where all the sites in C are contained in one or two paths, and the recombination of the two end sequences at point r creates Y, then this new perfect phylogeny can also be used to arrange the site on Q. To see this, note that the node labels on both arrangements are exactly the same: restricted to C, both arrangements label the nodes with the sequences in $M(C)$, and only sites in C appear on Q. The state of each site not in C is identical at every node on Q. Hence, the two arrangements of sites on C induce a permutation of the nodes on Q and of the edges out of Q, but are indistinguishable outside of Q. Note that in all arrangements on Q, the sequence labeling the coalescent node contains only zeros at the sites in C. For example, removal of sequence b instead of d from $M(C)$ in Figure 6 leaves a set of sequences with the perfect phylogeny shown in Figure 8, and the arrangement of gall Q shown in Figure 9.

Hence, all arrangements of the sites in C on a gall Q can be found by the following algorithm:

Site-Arrangement Algorithm

0) Determine the recombination point r from C.

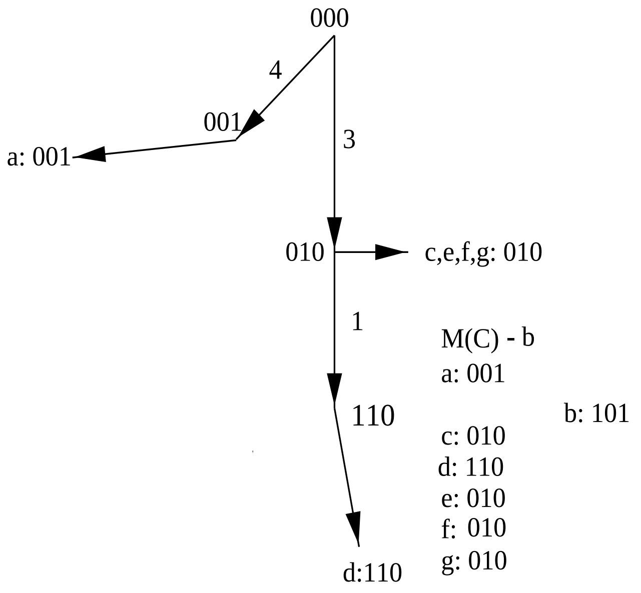


Figure 8. The perfect phylogeny for $M(C) - b$.

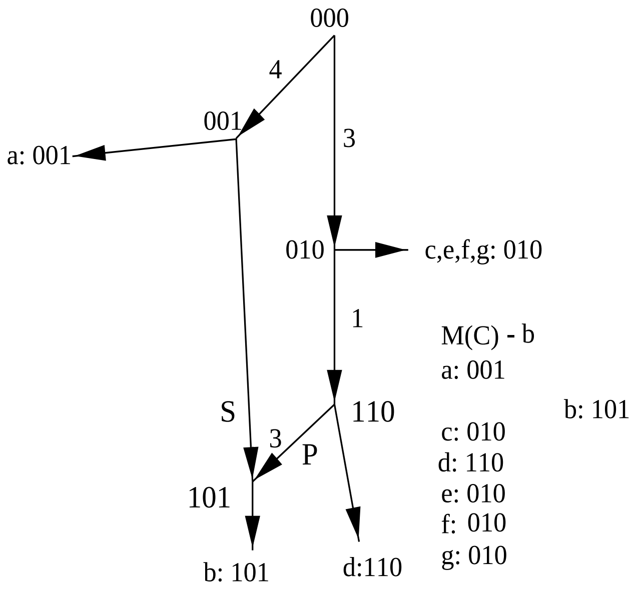


Figure 9. The arrangement of gall Q derived from the perfect phylogeny for $M(C) - b$.

1) Let $M(C)$ be matrix M restricted to the sites in C.
2) For each distinct sequence X in $M(C)$ do:
3) Let $M(C, X)$ be $M(C)$ after the removal of all rows with sequence X. Check if there is a perfect phylogeny for $M(C, X)$, and if so, check if all sites on C are contained in one or two paths whose end sequences can be recombined at point r to create X.
If the answer is "yes", then output an arrangement of the sites on Q consistent with this perfect phylogeny.

Specializing to galled-trees, we have established,

Theorem 3.3 *Assuming that there is a galled-tree for M, every arrangement of the sites in C on Q that is used in some galled-tree for M, can be found as above. The set of sequences labeling nodes of Q, restricted to C, is invariant over all the arrangements, and all the galled-trees for M.*

Time Analysis: Given the matrix $M(C, X)$ with n rows and at most $|C|$ columns, the perfect phylogeny algorithms in [3, 4] can determine if there is a perfect phylogeny for $M(C, X)$, and construct it, in $O(n|C|)$ time. So, all the arrangements of Q that are used in any galled-tree for M can be found in $O(n^2|C|)$ time, and over all the galls, the time to find all the arrangements that appear on any galled-tree is $O(n^2 m)$. It was established in [21] that m can be at most $2n$ if there is a galled-tree for M, so the total time is $O(n^3)$.
By a more detailed analysis of how the sites on a gall can be arranged, we have developed an alternative algorithm [8] for arranging the sites on a gall, whose running time is only $O(n^2)$.

We have claimed that the galled-tree for M is "essentially-unique". The one-one correspondence between connected-components and galls establishes the first part of that claim. We now show that the number of arrangements of the sites on a gall is very limited, further establishing the "essential-uniqueness".

Theorem 3.4 *Let C be a non-trivial connected component of the conflict graph whose sites can be arranged on a gall Q in some phylogenetic network for input M. The sites in C can be arranged on Q in at most three distinct ways, as long as no ordering is given to multiple sites on the same edge.*

Proof We have already established that each distinct arrangement of sites on Q is associated with one distinct sequence X in $M(C)$, with the property that when all copies of X are removed from $M(C)$, the remaining sequences in $M(C)$ can be derived on a perfect phylogeny. Moreover, when no ordering is given to multiple sites on edges, a sequence X is associated with at most one arrangement of sites on Q. Hence, when all copies of X are removed from $M(C)$, all conflicts between pairs in C are broken, and we can bound the number of distinct arrangements of sites on Q by bounding the number of distinct sequences in $M(C)$ whose removal (of all identical copies) breaks all conflicts in C.
Let i, j be a conflicting pair of columns in C. In order for the removal of X to break the i, j conflict, the row for X in $M(C)$ must contain one of the three state-pairs 0,1 or 1,0 or 1,1 in columns i, j, and no other row in $M(C)$ can contain that state-pair in columns i, j. It follows that there can be at most three rows in $M(C)$ whose removal can break the i, j conflict, and hence there can be at most three distinct arrangements of the sites in C on Q. $\square$

We next show that C can actually be arranged on Q in three ways, only when C has two sites, as in Theorem 2.2. Otherwise, Q can only be arranged in two ways, and typically will only have one arrangement. For simplicity, we assume that all but one copy of any duplicate row has been deleted. These results are obtained using the following

Lemma 3.2 *Let C be a connected component in the conflict graph with at least three sites i, j, j', where i conflicts with both j and j'. Suppose the sites in C can be arranged on a gall Q in a phylogenetic network for the input. If X_1 is a sequence whose removal breaks all the conflicts in $M(C)$, and the i, j state-pair for X_1 is 0,1, then there is only one way to arrange the sites of C on Q.*

Proof First, the i, j' state-pair must also be 0,1 for otherwise it would be 0,0 and the removal of a 0,0 pair would not break the i, j' conflict in $M(C)$. Now suppose X_2 is another sequence whose removal breaks all conflicts. The value of entry (X_2, i) must be 1, since if it were 0, then the i, j state-pair for X_2 would either be 0,1 or 0,0. The first possibility would contradict the assumption that removing X_1 breaks the i, j conflict, and the second possibility would contradict the assumption that removing X_2 breaks the i, j conflict. So let $1, x, y$ denote the i, j, j' state-triple for X_2. Since i and j are in conflict, there must also be another row X_3 whose i, j state-pair is $1, \bar{x}$, where $\bar{x}$ is $x + 1 \bmod 2$. Also, the i, j' state-pair must be $1, \bar{y}$, since otherwise it would be $1, y$, and then the removal of X_2 would not break the i, j' conflict.

If $x = y$, then columns j and j' are identical in rows X_1, X_2 and X_3. Any other row S must have the j, j' state-pair of 0,0 or $\bar{x}, \bar{y}$. To see this, consider the state of i in row S. When i is 0, the j, j' state-pair must be 0,0 for otherwise either i, j or i, j' will be 0,1 and the removal of X_1 would not break all the conflicts. When i is 1 in S, the j, j' state-pair must be $\bar{x}, \bar{y}$ or else one of the i, j and i, j' state-pairs in S would be identical to the corresponding pair in X_2, and the removal of X_2 would not break all conflicts. So when $x = y$, columns j and j' are identical, which contradicts the assumption that all columns in the input are distinct.

But if $x \neq y$ (and hence $\bar{x} \neq \bar{y}$), then the j, j' columns would have all three pairs 1,1; 0,1; and 1,0 and so be in conflict. That is a contradiction, because j and j' are both in conflict with i, and hence must be on the same side of the (bipartite) conflict graph.

The conclusion is that if i is in conflict with two columns j and j', and if X_1 is a sequence whose removal breaks all the conflicts in $M(C)$, and the i, j state-pair for X_1 is 0,1, then X_1 is the only sequence whose removal breaks all conflicts in $M(C)$, and hence the arrangement of C on Q is unique. $\square$

Theorem 3.5 *Let C be a connected component of the conflict graph whose sites can be arranged on a gall Q in a phylogenetic network for the input. If C has at least three sites, then the sites can be arranged on Q in at most two distinct ways, and if C has at least two sites on each side of bipartite graph (i.e., on each side of the recombination point for C), then the sites can be arranged on Q in only one way.*

Proof When there are at least three sites in C, there must be at least two sites on one side of the bipartite graph C, and since C is connected, there must be at least one site, call it i that is connected to two distinct sites j and j'. Suppose there are three distinct sequences (rows) X_1, X_2 and X_3 in $M(C)$, such that the removal of any of these sequences breaks all the conflicts in $M(C)$. Then one of those three rows must contain the pair $0, 1$ for columns i, j. Applying Lemma 3.2 leads to a contradiction, so when C has at least three sites, there can be at most two arrangements of the sites of C on Q.

Now assume that each side of the bipartite graph C has at least two sites, and suppose there are two ways to arrange the sites of C on Q. Hence there are two rows X_1 and X_2 whose removal breaks all conflicts in $M(C)$. Since C is connected, there must be an edge (i, j) such that i is also adjacent to a node $j' \neq j$, and j is adjacent to a node $i' \neq i$. Applying Lemma 3.2 to the i, j pair, the i, j state-pair cannot be 0,1 (and it can't be 0,0) in either row X_1 or X_2. Further, the i, j state-pair cannot be the same in X_1 and X_2. So in one of those two rows, the i, j state-pair must be 1,0. But node j is also adjacent to node i', so (after relabeling) we can apply Lemma 3.2 to obtain a contradiction to the assumption that the removal of both X_1 and X_2 break all conflicts. $\square$

Hence, except in degenerate cases, there is a unique permitted arrangement of the conflicted sites on a gall, and otherwise, all the arrangements can be compactly represented and easily generated.

For example, in Figures 2 and 9, we showed two arrangements of the sites in the connected component C containing $\{1, 3, 4\}$. These two arrangements were associated with rows b and d in $M(C)$. However, there is no other row of $M(C)$ that can be deleted to break all the conflicts in C.

4 Connecting the galls in a galled-tree

Now we explain how to connect the galls together into a single galled-tree. Let T be a particular galled-tree for M and let Q and Q' be two galls in T. Gall Q is an "ancestor" of a gall Q' in T if there is a directed path in T from some node on Q to the coalescent node of Q'. If neither gall is an ancestor of the other, then we say that they are "incomparable". The algorithm to connect the galls will first deduce the ancestry relations between pairs of galls. We will see that the ancestry relations are invariant over all the galled-trees for M.

Since T is a particular galled-tree for M, the arrangement of sites on Q and Q' is determined. In that arrangement, let f_P and f_S be the first sites on the P and S sides respectively on Q. Define f'_P and f'_S similarly for Q'. Assume, without loss of generality, that f_S and f'_S exist. The analysis is symmetric for the other three combinations, one of which must exist. Also, let i, j be a pair of sites on Q that conflict with each other. Note that at the recombination node for Q, the state of at least one of i or j is set to 1, say i. Note that i might be f_P or f_S. Similarly, there is a site i' that has state 1 at the recombination node for Q'.

Now let Z' be any row of M with a 1 in column f'_S (there must be one since f'_S is involved in a conflict). If Q is an ancestor of Q' then Z' must have a 1 in at least one of the columns for f_S, f_P or i. Similarly, let Z be any row of M with a 1 in column f_S. If Q' is an ancestor of Q then Z must have a 1 in at least one of the columns for f'_S, f'_P or i'. So Q and Q' are incomparable if and only if neither of these conditions hold for rows Z' and Z.

With the proper data structure for M, and after simple $O(nm)$-time preprocessing of M, rows Z' and Z can be found in constant time, and in constant time we can check those (up to six) entries in rows Z' and Z. So in constant time, we can determine whether Q and Q' are incomparable or not. If comparable, then we have found a row which has a 1 in a column q for a site that appears on Q and a 1 in a column q' for a site that appears on Q'. That means that there is a path from the root of T that passes through both the edges where sites q and q' mutate. We claim that Q is an ancestor of Q' if and only if site q has strictly more 1's in its column than does column q'. To see this, note first that by Lemma 2.1, site q appears before q' on the path if and only if site q has a 1 in every row where site q' has a 1. Moreover, for any conflicted site on a gall, there must be at least two nodes on that gall where that site has state 1, so a tie for the largest number of 1's in columns q and q' is not possible. Hence,

Theorem 4.1 *After $O(nm)$ preprocessing time, we can determine if Q and Q' are comparable in constant time, and if comparable, determine which is the ancestor of the other. There are at most $O(m) = O(n)$ galls [21], so over all the pairs of galls, we can determine all the ancestry relations in $O(nm + n^2) = O(n^2)$ time.*

A gall Q is called the "immediate ancestor" of a gall Q' in T if Q is an ancestor of Q' and no descendent of Q is an ancestor of Q'. Every gall in T that has an ancestor in T, has a unique immediate ancestor, and the ancestor relation computed above is the transitive closure of the immediate-ancestor relation. Hence, given a fixed arrangement of the sites on each gall, to find the immediate-ancestor (if any) of each gall, we find the transitive-reduction of the ancestor relation. This can be done in $O(n^2)$ time (which is faster than for general transitive-reduction) because the transitive reduction is a tree in this case, so each gall has a unique immediate ancestor. One approach then is to sort the galls by the number of ancestors they have. Then the gall Q' with the largest number of ancestors is a "leaf gall" in the transitive reduction, and we identify its immediate ancestor as the ancestor of Q' which has exactly one fewer ancestors than does Q'.

This identifies for each gall Q', its immediate ancestor Q in T, or determines that Q' has no ancestor. Since we are ignoring unconflicted sites, every site appears on some gall, so in T a gall is connected to its immediate ancestor by a single edge (rather than a path). If Q is the immediate ancestor of Q', we next want to determine the specific node, call it $v(Q,Q')$, on Q which is connected by a single edge to the coalescent node of Q' in T.

Let i,j be a conflicting pair on Q and let F be a sequence in M with a 1 for f'_S or f'_P. We claim that the i,j state-pair (0,1; 1,0; or 1,1) at the recombination node x of Q is found at no other node on Q. This was noted after the proof of Theorem 2.2, and follows from the details of that proof. Hence $v(Q,Q')$ is node x if and only if F has the same i,j state-pair that is found at x. If that determination finds that $v(Q,Q')$ is not x, then F must have a 1 for exactly one of f_P or f_S, which identifies the side of Q that $v(Q,Q')$ is on. We then walk from the coalescent node of Q along that side until either encountering the edge e into the recombination node, or encountering the first edge e containing a site i such that F has state 0 for i. Node $v(Q,Q')$ is the node at the head of edge e. Since each of the $O(n)$ sites is on at most one gall, the total time to find all these nodes is $O(n)$.

Definition 4.1 *Let $\tilde{T}$ denote the digraph determined to this point, i.e., consisting of all the arranged galls connected by the immediate ancestry edges.*

Note that $\tilde{T}$ might be a forest of galled-trees, rather than a single galled-tree.

Now the above exposition and determination of ancestry relations was based on assuming a particular arrangement of sites on each gall. But from Theorem 3.3, different arrangements of the sites on a gall merely permute the positions of the nodes, and the branching edges attached to them. This clearly does not change the ancestry and immediate ancestry relations. Therefore, we can use any permitted arrangement of the sites on the galls to determine $\tilde{T}$, and

Theorem 4.2 *The digraph $\tilde{T}$ is unique up to the different permitted arrangements of nodes inside the galls.*

Theorem 4.2 is a further reflection of the "essential uniqueness" of the reduced galled-trees that derive M.

Corollary 4.1 *Ignoring the issue of how to place the unconflicted sites, if there are k connected components of the conflict graph, and the sites on component C_q can be arranged in A_q different ways, then the number of reduced galled-trees for M is exactly $\Pi_{q=1}^{q=k} A_q$, where $A_q \leq 3$.*

We will show in the next section that unconflicted sites need not be on any galls, so algorithmically, Theorem 4.2 implies

Corollary 4.2 *Given the galls, and an arrangement of the sites on the galls, $\tilde{T}$ can be determined in $O(n^2)$ time. Further, if there is a galled-tree for M, any $\tilde{T}$ determined at this point can be extended to a galled-tree for M, by placing the unconflicted sites on edges of $\tilde{T}$ between galls, and possibly adding new edges containing unconflicted sites, or new edges leading to leaves.*

A very different approach to connecting the galls is outlined in the powerpoint slides [6].

5 Unconflicted sites need not be on galls

So far, we have only detailed how to handle conflicted sites. Now we focus on unconflicted sites.

First, if Q is a gall containing only unconflicted sites, then there is a perfect phylogeny for the sites on Q, and Q can be replaced by that perfect phylogeny, creating a phylogenetic network for M with one fewer gall. So we consider the case that every gall contains some conflicted sites.

Definition 5.1 *Consider a gall Q with recombination point r, and a particular arrangement of sites on Q. We define L_r to be all the sites on Q with index less than r, and R_r to be all the sites on Q with index equal or greater to r. We define P_L to be the conflicted sites in L_r on the P-side of Q; P_R to be the conflicted sites in R_r on the P-side of Q; S_L to be the conflicted sites in L_r on the S-side of Q; and S_R to be the conflicted sites in R_r on the S-side of Q.*

Lemma 5.1 *Let N be a phylogenetic network deriving M with a gall Q containing no conflicted sites on its S-side. Then there is a phylogenetic network N' deriving M with gall Q replaced by gall Q' whose S-side contains no sites and no branching nodes. A symmetric result holds if the P-side contains no conflicted sites.*

Proof Since all conflicted sites are on the P-side, P_L and P_R are non-empty. Moreover, by Theorem 2.2 there must be a branching edge below the last conflicted site on P.

Let s be an unconflicted site in R_r on the S-side and suppose there is another site t below s. If t is in R_r, then there would have to be a branching node between them, in order for the columns in M for s and t to be different. But then s would be in conflict with any conflicted site in P_L, a contradiction. If t is in L_r, but there is a branching node below t, then there is a branching node below s and again s would be in conflict with any site in P_L. But if there is no branching node below t, and since the value of t is set to 0 at the recombination node of Q, the column for t must be all zeros, a contradiction. Hence, if the S-side has only unconflicted sites and contains a site s in R_r, s must be the last site on S, and it must not have a branching node below it on Q.

Now, the value of site s is 1 at the recombination node x of Q. If x has a single edge out of it, we can move s off of Q and place it on that single edge out of x. If x has more than one edge out of it, we create a new, single edge e out of x attached to all the edges that had previously been out of x. We then move s to e, creating a modified phylogenetic network that derives M.

After the removal of site s from Q, the only remaining sites (if any) on the S-side of Q are in L_r. If there are none, then the transformation is complete. Otherwise, all of those sites have value 0 at the recombination node x, so the sequence label at x is maintained even if we remove all those sites from Q. But we need to maintain the sequence label of any branching node v on the S-side, and the subgraph branching off of Q at v. Hence, we can modify Q as follows: remove the edge into x on the S-side, and add a new edge directly from the coalescent node w of Q to node x. The result is a modified phylogenetic network which derives M, where the S-side of the modified gall Q has no sites and no nodes. □

In the transformation above, the part of the old S-side of Q that contains all the sites and nodes, is now a path branching from the coalescent node w. For the modifications of Q used to prove the next theorem, it is useful to maintain the assertion that there is no edge branching off Q at w. So we further modify Q as follows. Let $e = (w', w)$ be the edge into w. Expand e into two edges (w', w'') and (w'', w) by inserting a new node w'' on e, below any sites that are on e. This creates a new edge into w with no sites. Then disconnect any branching edge out of w and reconnect it to w''. The resulting phylogenetic network clearly derives exactly the same sequences as before.

Theorem 5.1 *Let N be a phylogenetic network deriving M with a gall that contains both unconflicted and conflicted sites. Then there is also a phylogenetic network N' with the same number of galls as N, where each gall only contains conflicted sites.*

Proof We consider a single gall Q, and examine the P-side of Q in detail. The argument for the S-side is symmetric. If the P-side only contains unconflicted sites, then these are removed using Lemma 5.1. So assume that the P-side contains both conflicted and unconflicted sites.

Let v be the last branching node on the P-side of Q. By Theorem 2.2, any sites below v must be unconflicted. There can be no sites in R_r below v, for any such site would have an all-0 column in M, a contradiction. Any sites in L_r below v have value 1 at the recombination node x and can be moved to the edge out of x, as in the proof of Lemma 5.1. Hence, we assume that all sites on P are above node v, and so there is a branching edge below every site on the P-side of Q.

If there are no additional unconflicted sites on the P-side of Q, then the transformation is complete. Otherwise, let p be an unconflicted site in L_r on the P-side of Q. If there is any site q in S_R, then by Theorem 2.2 p and q would conflict, a contradiction. So, if there is an unconflicted site in L_r on the P-side, then there can be no site in S_R, and hence no site in S_L (recall that by definition, such sites are conflicted), and so there must not be any conflicted sites on the S-side of Q. But then by Lemma 5.1, all the unconflicted sites and branching edges can be moved off of Q, and we assume that has been done.

Now consider p^*, the highest unconflicted site in L_r on the P-side. There cannot be a conflicted site i in P_L above p^*, since by Theorem 2.2, i would have to be in conflict with a site $j \in P_R$ above i with a branching node between j and i. That would create the conditions for p^* and j to be in conflict, a contradiction. So p^* must be above all sites in P_L.

Similarly, if there is any site j in R_r above p^*, then there can be no branching node between it and p^*. This follows because there is a branching edge below p^*, so by Theorem 2.2, sites j and p^* would be in conflict, a contradiction. So there can be no branching nodes above p^* and no branching node (or sites) on the S-side of Q. Therefore, we can move p^* to the edge e entering the coalescent node of Q, creating a modified phylogenetic network N' the derives M and contains one fewer unconflicted sites.

If there is a branching node v on the P-side of Q with no sites above v on Q, then we further modify N' to prepare for additional site removals. Let $e = (w', w)$ be the edge into the coalescent node w. Expand e into two edges (w', w'') and (w'', w) by inserting a new node w'' on e, below any sites that are on e. Then disconnect any branching edges out of v and reconnect these edges to w''.

Iterating as above, we can remove all unconflicted sites in L_r on the P-side of Q. We let Q' denote the resulting gall at this point, and now discuss how to handle any unconflicted sites in R_r on the P-side of Q'.

We make two observations. First, the only sites in L_r on the P-side of Q' are conflicted sites, i.e., in P_L. Second, every unconflicted site p in R_r must be below every site j in P_R. To see this, note that j must be in conflict with a site $i \in P_L$ below j on the P-side of Q', so there must be a branching node between j and i, and a branching edge below i. Hence if p were above j, the conditions would be satisfied for p to conflict with i, a contradiction. So either all the unconflicted sites on the P-side of Q' are together at the bottom of the P-side of Q', just above the last branching node v, or there is an unconflicted site p in R_r above a conflicted site i in P_L.

In the later case, we transform Q' to conform to the former case, as follows. Assume that p is the first (topmost) unconflicted site in R_r on the P-side, and i is the last site in P_L. There cannot be a branching node between p and i, or else they would conflict. However, if p is not immediately above i, consider a site q between p and i. If q is in R_r, there would have to be a branching node between p and q or they would have identical columns in M. If q is in P_L, there would have to be a branching node between q and i or else they would have identical columns in M. Hence if there is an unconflicted site $p \in R_r$ above a site in $i \in P_L$, there only be one such pair, and the two sites must be adjacent with no branching node between them. But then, we can flip the positions of p and i and still have a phylogenetic network that derives M. At that point, all the unconflicted sites in R_r are together at the bottom of the P-side of Q', just above the last branching node v.

Let k be the site (if any) just above the first unconflicted site p in R_r on the P-side of Q'. If there is a node between k and p, call it v', and otherwise create a node v' between sites k and p. If there is no site k above p, then set v' to the coalescent node w of Q'. Then remove the edge into x from node v, and create an edge from v' to x containing no sites. The result is a phylogenetic network that derives M where gall Q has been transformed to a gall Q' that has only conflicted sites, and all other galls remain unchanged.

The theorem is proven by repeating this transformation for every gall containing unconflicted sites. □

Corollary 5.1 *If there is a galled-tree for M, then there is a reduced galled-tree for M. Moreover, the number of recombinations used by any reduced galled-tree for M is the number of non-trivial connected components in the conflict graph, and this is the minimum number of recombinations possible over all galled-trees for M.*

We will later show that any reduced galled-tree for M uses exactly m_M recombinations.

6 Adding the leaf sequences and the unconflicted sites

We assume there is a galled-tree T that derives M. To finish constructing a galled-tree for M from $\tilde{T}$, we must extend $\tilde{T}$ by adding in any unconflicted sites, and place the sequences of M at specific leaves, possibly adding additional tree edges outside of any galls. We do this in three main phases.

6.1 Phase 1

Let Z be a sequence that has state 1 for a conflicted site i. Then in any galled-tree T for M, the leaf labeled with Z must be below the gall containing i. Conversely, any sequence labeling a leaf in T below a gall, must have a 1 for at least one site on that gall (recall that there is no branching off the gall from the coalescent node). Therefore, we can divide the sequences into those that have at least one 1 for a conflicted site, and those that don't. The sequences in the second set (if any) must be derivable on a unique perfect phylogeny U that must be the "upper part" of any galled-tree for M. This follows from Theorem 2.1. If the second set is empty, then consider U to be a single node, which will be the root node of any galled-tree for M.

We can efficiently construct U, and then (assuming there is a galled-tree for M) determine where each gall in $\tilde{T}$ resides relative to U. For example, let Q be a "maximal" gall (one with no ancestor) in $\tilde{T}$, let i be either the first site on the P-side or the S-side of Q, and let Z be a sequence which has state 1 for site i. Then walk along a unique path from the root of U following edges labeled with sites which have state 1 in Z. If that walk ends at node v (either a leaf in U or Z has state 0 for every site on the edges out of v), then gall Q must be a descendant of v in galled-tree T. So the sub-galled-tree rooted at Q hangs off of U at node v. We modify $\tilde{T}$ by finding the attachment node v on U for each maximal gall Q, and connecting the coalescent node of Q to v. $\tilde{T}$ now is a single galled-tree rather than a forest, since U is a single tree. Moreover, $\tilde{T}$ correctly derives the sequences in M, restricted to the conflicted sites and the sites in U. Note that in the full galled-tree T for M, Q might continue to be connected to v by a single edge, or may be connected by a path of unconflicted sites that will be added in later, as detailed below.

6.2 Phase 2

Let Z be a sequence which has state 1 for at least one conflicted site, and hence the leaf for Z in T must be a descendant of some gall in T. Let M' denote the set of such sequences. For each such sequence Z in M', we will find the node v_Z in $\tilde{T}$ such that in any galled-tree for M, v_Z is the last node in $\tilde{T}$ on the path from the root to leaf Z.

To do this, we do a bottom up traversal of $\tilde{T}$, only traversing a gall after traversing all its descendents. At the start of the bottom up traversal, all sequences in M' are unmarked. Let i, j be a conflicting pair that appears on a gall Q. We traverse gall Q as follows. Declare the recombination node x to be node v_Z for every unmarked sequence Z in M' which has same i, j state-pair as does node x; mark every such sequence Z. Then do a bottom-up traversal of one side of Q, and for each site p encountered (just above a node v), declare node v to be v_Z for each unmarked sequence Z in M' which has state 1 for site p; again, mark every such sequence. Then do a bottom-up traversal of the other side of Q in a similar manner. The traversal over $\tilde{T}$ takes $O(n^2)$ time and finds the node v_Z for each sequence Z in M'. We modify $\tilde{T}$ by extending an edge from v_Z to a leaf labeled Z, for each sequence Z in M'. This places all the leaves for the sequences in M either in U, or at a leaf connected by an edge to a node in $\tilde{T}$. Note that v_Z will not be a coalescent node for any gall.

For exposition purposes, we further modify $\tilde{T}$ as follows. For a node v on a gall Q, let V be the set of nodes off of Q that are children of v. If $|V| > 1$, create a new edge from v to a new node v', and connect v' to every node in V. The effect is that every node v on a gall will have only one edge branching off the gall from v. Note that v can be a recombination node.

Next, with another bottom-up traversal (or folded into the previous traversal), label each *edge* e with (the index of) one sequence, call it $Z(e)$, such that the leaf for Z is below e. In case there is more than one to choose from, choose arbitrarily.

6.3 Phase 3

Now we turn to the issue of adding in the remaining unconflicted sites.

Sites that are part of U need no further attention. For any other unconflicted site i, do the following: Find a sequence Z in M' which has a 1 for site i, and start a walk on a path from the leaf in $\tilde{T}$ labeled with Z towards the root of $\tilde{T}$. We know that site i must be placed somewhere along that path, although we may need to create a new edge that it will label. The main idea is to move up along an edge on that path once we have determined that i must be above the current edge in the walk. Otherwise, i must be placed on the current edge . However there are some subtle details.

During the walk for i, let v be a node entered along an edge e. There are four cases: 1) v is a node in the perfect phylogeny U; or 2) v is is a recombination node of a gall Q; or 3) v is on a gall Q, but is not the recombination node of Q; or 4) v is not on U and not on any gall.

In case 1) site i must be placed on edge e. We will only be in this case if we have already determined that all sequences at leaves below e have state 1 for site i. However, site i cannot be placed higher, for then there would be a sequence at a leaf in U with a 1 for site i, and hence i would already have been placed in U.

Cases 2) and 3) are similar. In either case i should be placed on edge e, or i should be placed somewhere on the path from the root of $\tilde{T}$ to the coalescent node of Q (we know we need not place i on any edge in Q). To determine which is the proper placement in case 2), let p be a branching node on Q (there must be one since Q has some conflicting sites) and let e' be a branching edge off of Q at p. To determine the proper placement in case 3) let e' be the edge on Q just below v. Then in either case 2) or 3), site i should be on edge e if and only if the sequence $Z(e')$ has state 0 for site i. When $Z(e')$ has state 1 for site i, the walk jumps to the coalescent node of Q and resumes from that point.

In case 4) (i.e, when v is not on a gall or on U), examine the sequence $Z(e')$ for *each* edge e' out of v other than e. If none of these sequence have a 1 for site i, then place site i on edge e. If every one of the sequences have a 1 for site i, then continue the upward walk from v. If some of the sequences have a 1 for site i, and some do not, let p be the parent node of v on $\tilde{T}$. Create a new node p' and a new directed edge (p, p'). Then disconnect from v every edge e' whose sequence $Z(e')$ has a 0 for site i, and reconnect e' to node p'. Then place the site i on the (p, v) edge. $\tilde{T}$ again denotes the modified galled-tree.

The correctness of algorithm follows from Corollary 4.2 and Theorem 5.1 and the observation that all decisions made by the algorithm are forced if no unconflicted sites are to be placed on galls. The time for placing the unconflicted sites is $O(n)$ per site, so $O(n^2)$ overall.

7 Time bound and Correctness

All of the results given above assume the existence of a galled-tree for the input M. These results imply the correctness of the algorithm derived from them, when there is a galled-tree for M. When there isn't one, the algorithm either will not be able to execute a required step, or it will run to completion producing some galled-tree. At termination, the algorithm checks whether or not the galled-tree it produced does derive M and that each site is on only one edge. If not, it correctly reports that there is no galled-tree for M. Note that the algorithm produces a reduced galled-tree since each gall only contains conflicted sites.

The overall time bound for the algorithm is $O(nm+n^3)$. If $m > 2n$, then we first find and remove all duplicate columns in $O(nm)$ time using radix sort (considering each column to be a binary number). If the number of remaining columns is more than $2n$ then M has no galled tree [21]. Next we build the conflict graph in $O(n^3)$ time and find the non-trivial connected components. The time to arrange all of the galls was shown to be $O(n^3)$, but that can be reduced to $O(n^2)$ time [8]. All other steps require $O(n^2)$ time.

8 Optimality: Every reduced galled-tree uses exactly m_M recombinations

Let M be a set of sequences that are derivable on a galled-tree, and suppose the conflict graph for M has k non-trivial connected components. Recall that m_M is the minimum number of recombinations needed in any phylogenetic network (with all-0 ancestral sequence) that derives M. We have already established (Corollary 5.1) that if there is a galled-tree for M, then there is a reduced galled-tree for M, and that every reduced galled-tree for M uses exactly k galls and hence k recombinations. We now prove that $m_M = k$, i.e., no phylogenetic network for M uses fewer than k recombinations[1]. To prove this, we exploit a (modified) method from Myers and Griffiths [14] that computes a lower bound on the number of recombinations needed by any phylogenetic network for M. The modification here specializes their algorithm to the case when the ancestral sequence is known and assumed to be all-0. Other small changes are made for simplicity of exposition. Their method successively modifies the input matrix M, and we use $\overline{M}$ to denote the modified matrix at any point in the algorithm. Initially $\overline{M}$ is set to M.

[1]This claim was made first in [21], but the proof proposed there only relates to the algorithm in that paper, and also seems incomplete in that it only seems to establish that (in our language) any galled-tree for M needs at least k recombinations. Thus it does not seem to establish that no phylogenetic network can use fewer recombinations

The Myers and Griffiths lower bound method uses three types of operations on $\overline{M}$: *column deletion, sequence deletion, and sequence removal*. A column deletion of column i from $\overline{M}$ is allowed if column i contains zero or one 1's. A sequence deletion of sequence Z from $\overline{M}$ is allowed if sequence Z is identical to some other sequence in $\overline{M}$. A sequence removal of an arbitrary sequence Z from $\overline{M}$ is allowed if neither a column deletion nor a sequence deletion is possible. The distinction between a sequence "deletion" and a sequence "removal" is critical and the words will never be used interchangeably. (We could define a fourth permitted operation, deleting any column that contain no 0's. The method would be correct with this added operation, but it is not needed.) Matrix M is said to be "eliminated" when $\overline{M}$ is empty. The following algorithm eliminates M.

Elimination Algorithm

```
Until $M$ is eliminated:

1) Successively perform sequence and column deletion operations
on $\overline{M}$ in any order until no further deletions
are possible.

2) Arbitrarily choose one remaining sequence $Z$ in $\overline{M}$
and remove sequence $Z$ from $\overline{M}$.

3) Return to step 1)
```

Note that this algorithm is non-deterministic in that it allows choices in both steps 1) and 2). Hence different executions of the algorithm can result in a different number sequence removals, i.e., different number of times that step 2) is performed. An execution of the Elimination Algorithm is called a "minimum execution" if it uses the minimum number of sequence removals over all possible executions of the algorithm. We use R_s to denote that minimum number of sequence removals.

Theorem 8.1 *(Myers and Griffiths [14]) $R_s \leq m_M$, so R_s is a lower bound on m_M.*

Theorem 8.1 is actually a specialization (to the case when the ancestral sequence is known) of what is proven in [14], because [14] concerns phylogenetic networks where no ancestral sequence is known, and m_M is the minimum over all phylogenetic networks with the all-0 ancestral sequence. Theorem 8.1 can be proven by induction on the number of distinct sequences in M, or by induction on m_M. Although not stated in [14], the proof can be used to establish a somewhat stronger result.

Definition 8.1 *Given two sequences Z and Z' of equal length, say m, a sequence X of length m is created at at recombination node by a "multiple-crossover event" of Z and Z', if for every position k, $1 \leq k \leq m$, the value of sequence X at position k is either equal to the value at position k in Z, or to the value at position k in Z'. That is, at any position k, the value in X is taken either from Z or from Z'.*

A multiple-crossover event generalizes a single recombination. In a single recombination with recombination point r, sequence X is created by taking the values from one sequence (which we denoted as P) in positions 1 through $r-1$, and then taking the values from the other sequence (S) in positions r through m. Clearly, for any M, if a multiple-crossover event is permitted at each recombination node, then the number of recombination nodes needed to derive M is less than or equal to m_M.

Without observing this, the proof in [14] actually establishes that R_s is a lower bound on the minimum number of multiple-crossover events needed to derive M on any phylogenetic network. We use the phrase "generalized phylogenetic network" to refer to networks that allow a multiple-crossover event at any recombination node.

When M is small enough, R_s can be computed by a direct application of its definition, but in general, such an enumerative approach to finding a minimum execution is impractical. However, the method can be used as a conceptual device, and we use it that way to prove that any reduced galled-tree for M uses exactly m_M recombinations.

8.1 A more graphical view of the Elimination Algorithm

It is useful to have a graphical interpretation of the operations in the Elimination Algorithm. Such an interpretation can be developed in general, but we specialize it here to reduced galled-trees. As $\overline{M}$ changes during the algorithm, we let $\overline{T}$ denote a reduced galled-tree that derives the current $\overline{M}$. We also let $\overline{G}$ be the conflict graph representing the conflicting pairs of columns in $\overline{M}$. We make the following claims:

Lemma 8.1 *If there is only a single 1 entry in column i in $\overline{M}$, then site i cannot be on a gall in $\overline{T}$.*

Proof In a reduced galled-tree, the only sites on any gall are conflicted sites. Any conflicted site must have at least two 1's in its column. □

Lemma 8.2 *When a column i with one 1 is deleted from $\overline{M}$ in the Elimination Algorithm, a reduced galled-tree for the new remaining sequences can be created by deleting site i from $\overline{T}$, and removing the i'th character from each sequence labeling a node in $\overline{T}$. A column deletion never results in the deletion of a site on a gall in $\overline{T}$, and it does not result in the deletion of an edge in $\overline{G}$.*

Proof Clearly the modified galled-tree derives the remaining sequences. By Lemma 8.1 no sites on any gall are deleted, and since no edges are removed, Theorem 2.2 implies that all the conflicting pairs in $\overline{T}$ continue to conflict in the modified galled tree, hence no edge in $\overline{G}$ is affected, and the modified galled-tree is a reduced galled-tree. □

Definition 8.2 *A node v is a lowest common ancestor of two leaves Z and Z' if v is an ancestor of both Z and Z', and no descendant of v has that property.*

Note that in a galled-tree, there must be a *unique* lowest common ancestor for any two leaves Z and Z'.

Lemma 8.3 *If two sequences Z and Z' in M are identical, and v is their unique lowest common ancestor, then no path from v to the leaf labeled Z in a reduced galled-tree $\overline{T}$ can contain an edge with a site or an edge on a gall. This is symmetrically true for Z'. Further, the path from v to Z is unique, as is the path from v to Z'.*

Proof Suppose a path from v to Z contains an edge on a gall Q and let e be the last such edge. If e is not an edge into the recombination node of Q, then e contains a site i above a branching node, and by Lemma 2.1, sequence Z will have a value of 1 for i. However, no path from v to Z' can share an edge with a v to Z path since v is the lowest common ancestor of Z and Z'. So the v to Z' path will not contain e and will have a 0 for i, contradicting the assumption that Z and Z' are identical. If e is an edge into the recombination node x for Q, let i, j be any conflicting pair of sites on Q. Since $\overline{T}$ is reduced there will be such a pair i, j. Now the i, j state-pair at x is found at no other nodes of Q, and by Lemma 2.1, Z will have that state-pair. But no v to Z' path can contain x since v is the lowest common ancestor of Z and Z', so Z' will not have the i, j state pair at x, again contradicting the assumed equality of Z and Z'. Hence the v to Z path cannot use an edge on a gall. It follows that the path from v to Z is unique.

Now suppose that the unique path from v to Z contains a site i but no edge on a gall. By Theorem 2.2, Z will have a 1 for i, and since no path from v to Z' can contain an edge of the v to Z path, Z' will have a 0 for i. This is again a contradiction. □

Lemma 8.4 *Let v be the unique lowest common ancestor of leaves Z and Z'. When a sequence Z is deleted from $\overline{M}$ in the Elimination Algorithm, a reduced galled-tree for the remaining sequences can be created from $\overline{T}$ in the following way: delete the leaf Z, and then successively delete edges on the Z to v (backwards) path until reaching a node q which is the ancestor of some remaining leaf in $\overline{T}$. No edge in any gall in $\overline{T}$ will be deleted, and no edge in $\overline{G}$ will be deleted.*

Proof The network created by the edge deletions in $\overline{T}$ clearly derives all the remaining sequences, since an edge is deleted only if there is no remaining leaf reachable from it's end. Node q is either node v or a descendant of node v, because the leaf labeled by sequence Z' is reachable from v. So by Lemma 8.3, no edges on any gall are deleted, and if v is not on a gall, then no edges branching from a gall are deleted, and the resulting galled-tree remains reduced by Theorem 2.2. The edges in $\overline{G}$ remain unchanged as well. However, if v is a node on a gall Q and $q = v$, then v has at least two edges branching off of Q (one on the path to Z and a different one on the path to Z'). So even if the entire v to Z path is deleted, every node on Q will continue to have an edge branching from it, so by Theorem 2.2 all galls continue to contain the same conflicted pairs of sites, and the new galled-tree is reduced, and no edges in $\overline{G}$ are deleted. □

Lemma 8.5 *When a sequence Z is removed from $\overline{M}$ in the Elimination Algorithm, a galled-tree for the remaining sequences can be created from $\overline{T}$ by removing leaf Z and every edge e for which Z is the only leaf below e in $\overline{T}$. There is at most one gall Q in $\overline{T}$ such that any of the removed edges are in Q or are incident with a node in Q. Edges in $\overline{G}$ are removed from at most one non-trivial connected component.*

Proof Another way to view the transformation of $\overline{T}$ is that we remove the leaf Z, and then successively remove edges that are ancestral to Z, following any such path backwards, until each path reaches a node which is the ancestor of some remaining leaf in T. It is again clear that the resulting galled-tree derives the remaining sequences, since Z is the only leaf sequence in $\overline{T}$ that the removed edges could reach via a directed path. Any edge that could reach a different leaf sequence remains in the galled-tree. Now if the transformation removes an edge (s,t) on a gall Q in $\overline{T}$, then node t must be the recombination node x of Q. Otherwise, t would be a branching node and there would be two disjoint paths from t to the leaves of $\overline{T}$ (one using the branching edge off of Q at t, and using the edge below t on Q). Moreover, one side of Q, at least, must have a branching node (or else $\overline{T}$ would not be reduced), so if the edges into x are removed, no edge in $\overline{T}$ above the coalescent node of Q can be removed. Hence, the only edges on a gall that could be eliminated are the two edges entering its recombination node, and only edges from one gall can be removed in the transformation. Similarly, there can be at most one gall with a node incident to a branching edge that is removed. Since only one gall is affected, only one non-trivial connected component of $\overline{G}$ is affected. □

The key point of Lemma 8.5 is that a sequence removal from $\overline{M}$ results in the removal of edges from $\overline{G}$ in at most one non-trivial connected component.

Note that Lemma 8.5 does not claim that the new galled-tree is a reduced galled-tree. However, we need to have a reduced galled-tree that derives the new sequences in order to be able to continue to apply the Lemmas which assume a reduced galled-tree. But any subset of a set of sequences that can be derived on a galled-tree can also be derived on a galled-tree, and we showed above that if $\overline{G}$ has k non-trivial connected components, then the conflict graph for the remaining sequences has either k or $k-1$ non-trivial connected components. Hence,

Lemma 8.6 *Let $\overline{T}$ be a reduced galled-tree with k galls for a set of sequences $\overline{M}$. After a sequence removal in the Elimination Algorithm, there is a reduced galled-tree that derives the remaining sequences using either k or $k-1$ galls.*

Theorem 8.2 *Let M be a set of sequences that can be derived on a reduced galled-tree using k galls. Then $R_s \geq k$.*

Proof Consider any execution of the Elimination Algorithm. After every step of the algorithm we have a reduced galled-tree $\overline{T}$ that derives the current set of sequences $\overline{M}$, and a conflict graph $\overline{G}$ that represents the conflicts in $\overline{M}$, as detailed above. At the start of the execution, $\overline{M}$ is M, $\overline{T}$ is T, and $\overline{G}$ is G. At the end of the execution $\overline{M}$ and $\overline{G}$ are empty, and $\overline{T}$ is a single node.

Lemmas 8.2 and 8.4 show that row and column deletion operations do not result in the deletion of edges in any non-trivial connected component of the conflict graph. Lemma 8.5 shows that each sequence removal results in the removal of edges in at most one non-trivial connected component of the conflict graph. Hence, there must be at least k removal operations in any execution of the Elimination Algorithm, and $R_s \geq k$. □

Theorem 8.3 *When M can be derived on a reduced galled-tree using k galls, then $R_s = m_M = k$. Moreover, even if we allow a multiple-crossover event at any recombination node, there is no phylogenetic network deriving M that uses fewer than k recombination nodes. Hence, a reduced galled-tree for M is optimal in this stronger way as well.*

Proof R_s is a lower bound on the number of needed recombinations, and every reduced galled-tree derives M using only k galls, so $R_s \leq k$. Together with Theorem 8.2, we have $R_s = k$, and so $m_M = k = R_s$. As noted earlier, R_s is a lower bound on the number of multiple-crossover events needed in any generalized phylogenetic network that derives M, so no generalized phylogenetic network can use fewer recombination nodes than does a reduced galled-tree, when M can be derived on a galled-tree. □

It is interesting to note that the above argument also gives an indirect way to prove that all reduced galled-trees for M have the same number of galls, a fact that we have established earlier.

It may be easy to misinterpret Theorem 8.3. It says that *if* M can be derived on a galled-tree, then no generalized phylogenetic network can have fewer recombination nodes than does a reduced galled-tree for M. However, that does not imply that multiple-crossover events are never better than single recombinations. It certainly is possible to construct examples where there is no galled-tree for M, and a generalized phylogenetic network for M uses fewer recombination nodes than does any phylogenetic network (i.e., using only single crossovers) for M.

We note also that when the ancestral sequence need not be the all-0 sequence, there are examples where one can find a different ancestral sequence that allows a galled-tree for M, using fewer than m_M recombinations. However, it can be shown that the number of recombinations used is at least $m_M - 1$. A polynomial-time algorithm to choose the best ancestral sequence will be described in a future paper.

9 Duplicate Columns

For simplicity of exposition, we have assumed throughout that the input matrix M contains no duplicate columns or rows. The assumption of no duplicate rows is without loss of generality, but the assumption of no duplicate columns is a true restriction. If i and i' are two identical conflicted columns (sites) in M which are not adjacent to each other, it can happen that there is a galled-tree for M when i (say) is removed, but no galled-tree for the complete original M. This situation may be important in applications to real data. However, it is easy to modify the algorithm to correctly handle this possibility.

Theorem 9.1 *Let i be a conflicted site, and assume there are duplicates of column i in the full input M. Let M^* be the matrix obtained from M be removing all but one copy of i. Suppose there is a galled-tree T for M^*, let Q be the gall containing i, and let $[p+1, q]$ be the recombination interval of Q. Then there is a galled-tree for the full input M, if and only if all copies of column i are at sites strictly above p, or all are at sites strictly below q. That is, if and only if all the copies of i can be assigned to the same side of some permitted recombination point for Q. Moreover, if there is a galled-tree for M, it must have all copies of i together on the same edge, and placing all the copies on the edge in T containing the single i, is a permitted arrangement.*

We can apply Theorem 9.1 inductively to handle the case of several distinct columns that have duplicate copies. We leave the proof of this and of Theorem 9.1 to the reader.

10 Non-Zero ancestral sequences

For simplicity of exposition, we have assumed that the ancestral sequence is the all-zero sequence. That assumption is not necessary. What is necessary is that the ancestral sequence is specified at the time the input matrix M is specified. Let A represent the ancestral sequence. If A is not the all-zero sequence then for every site i which has value 1 in A, reverse the value of every entry in column i in M, and then solve the galled-tree problem where the ancestral sequence is assumed to be the all-zero sequence. If a galled-tree T is produced, then change the value back to 1 for every site whose value was changed, in every sequence labeling a node of T.

11 Relation to the back-mutation model

Another deviation from the perfect phylogeny model that is of interest is to allow a limited number of back-mutations, but no recombinations. A back-mutation is a mutation from state 1 back to state 0 that occurs on an edge, i.e., it is not a change due to recombination.

Theorem 11.1 *Any set of sequences M that can be derived on a galled-tree, can be derived on a true tree (no recombinations and hence no underlying undirected cycles) with at most one mutation and one back-mutation per site.*

Proof We take a galled-tree T for M and transform each gall Q separately, so that no cycles remain, but all the node labels are preserved. The simplest case is that Q has one side, say S, which has no mutations (sites). Remove the S-side (which consists of just a single edge into x) from Q. Let p denote the P-side parent of x. Then for any site i which has state 1 at p, but has state 0 at x, write a back-mutation for i on the (p, x) edge. Hence Q no longer is a cycle, but all the node labels on Q remain unchanged. The more complex case is that both the S and P sides have at least one mutation. In this case, remove the first edge on Q out of the coalescent node, on either the P or the S-side, say the S-side, and reverse the direction of all the remaining edges on the S-side. Next, for every site i that has state 1 at p but state 0 at x, write a back-mutation for i on the (p, x) edge. For every site i that has state 0 at p but state 1 at x, write the mutation i on edge (p, x). Let s denote the parent of x on the S-side of Q. For every site i that has state 1 at s, but state 0 at x, write the mutation i on the (x, s) edge (which now runs from x to s). For every site i that has a state 1 at x, but state 0 at s, write the back-mutation for i on the (x, s) edge. Finally, convert each original mutation on a remaining edges of the S-side to a back mutation. The result is that Q is no longer a gall, but all the node labels are preserved. Processing each gall in this way creates a true tree that derives M using at most one back-mutation per site. See Figure 10 for an example. □

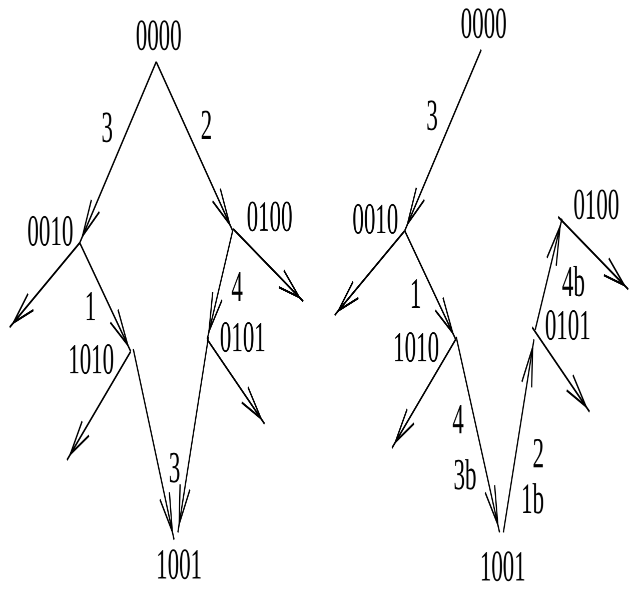


Figure 10. Gall Q is shown on the left and the result of the transformation is shown on the right. The recombination point for Q is 3, written above the recombination node. A number written on an edge is a mutation; a number followed by the letter b denotes a back-mutation.

12 Program, Future Work and Open Questions

Our interest in phylogenetic networks with recombination (and other non-tree-like properties) continues beyond the content of this paper.

Program Gall.pl The method in this paper, combined with ideas from [8], has been implemented in a Perl program that constructs a reduced galled-tree for input M, or determines that no galled-tree is possible. The program will be available at `wwwcsif.cs.ucdavis.edu/˜gusfield/`.

Future Papers: The key ideas introduced in this paper are the one-one correspondence of connected components of the conflict graph and galls in a galled-tree, and the fact that the sites on a connected component C can appear on a gall in any phylogenetic network only if C obeys certain structural constraints. More generally, properties of phylogenetic networks more complex than galled-trees can also be elucidated through structural properties of the conflict graph. Some of these results are reported in [8]. Other results, such as lower bounds on the number of recombinations required in any phylogenetic network for M (where the recombination cycles can intersect in unconstrained ways), will be detailed in a future paper. Another future paper will detail a polynomial-time solution to the *root-unknown* galled-tree problem: If no ancestral sequence is known in advance, find (if one exists) a sequence Z so that there is a galled-tree for M with ancestral sequence Z, and if such a Z exists, find one where the resulting galled-tree minimizes the number of recombinations over all phylogenetic networks for M.

Open questions There are many open questions. Two important questions are how to handle missing data in M, and how to solve the perfect phylogeny haplotyping problem [5] when the underlying sequences (haplotypes) were derived on a galled-tree. This latter problem was the original motivation for beginning our study of galled-trees and phylogenetic networks.

Acknowledgments Research partially supported by NSF grant EIA-0220154. We would like to thank Katherine St. John for LaTeX advice. Thanks also to Dean Hickerson for suggesting consideration of multiple-crossover events in recombination lower bounds.

References

[1] W. H. Day and D. Sankoff. Computational complexity of inferring phylogenies by compatibility. *Syst. Zool.*, 35:224–229, 1986.

[2] F. Dragan. Strongly orderable graphs: A common generalization of strongly chordal and chordal bipartite graphs. *Discrete Applied Math*, pages 427–442, 2000.

[3] D. Gusfield. Efficient algorithms for inferring evolutionary history. *Networks*, 21:19–28, 1991.

[4] D. Gusfield. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.

[5] D. Gusfield. Haplotyping as Perfect Phylogeny: Conceptual Framework and Efficient Solutions (Extended Abstract). In *Proceedings of RECOMB 2002: The Sixth Annual International Conference on Computational Biology*, pages 166–175, 2002.

[6] 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.

[7] D. Gusfield, S. Eddhu, and C. Langley. Efficient reconstruction of phylogenetic networks (of SNPs) with constrained recombination. In *Proceedings of 2'nd CSB Bioinformatics Conference.* IEEE Press, 2003.

[8] D. Gusfield, S. Eddhu, and C. Langley. The fine structure of galls in phylogenetic networks with recombination. Technical report, UC Davis, Department of Computer Science Techreport CSE-03-28, 2003.

[9] J. Hein. Reconstructing evolution of sequences subject to recombination using parsimony. *Math. Biosci*, 98:185–200, 1990.

[10] J. Hein. A heuristic method to reconstruct the history of sequences subject to recombination. *J. Mol. Evol.*, 36:396–405, 1993.

[11] R. Hudson and N. Kaplan. Statistical properties of the number of recombination events in the history of a sample of DNA sequences. *Genetics*, 111:147–164, 1985.

[12] J. D. Kececioglu and D. Gusfield. Reconstructing a history of recombinations from a set of sequences. *Discrete Applied Math.*, pages 239–260, 1998.

[13] J.Z. Lin, A. Brown, and M. T. Clegg. Heterogeneous geographic patterns of nucleotide sequence diversity between two alcohol dehydrogenase genes in wild barley (Hordeum vulgare subspecies spontaneum). *PNAS*, 98:531–536, 2001.

[14] S. R. Myers and R. C. Griffiths. Bounds on the minimum number of recombination events in a sample history. *Genetics*, 163:375–394, 2003.

[15] L. Nakhleh, D. Ringe, and T. Warnow. Perfect phylogenetic networks: A new methodology for reconstructing the evolutionary history of natural languages. preprint, April 14, 2002.

[16] M. Norborg and S. Tavare. Linkage disequilibrium: what history has to tell us. *Trends in Genetics*, 18:83–90, 2002.

[17] D. Posada and K. Crandall. Intraspecific gene genealogies: trees grafting into networks. *Trends in Ecology and Evolution*, 16:37–45, 2001.

[18] M. H. Schierup and J. Hein. Consequences of recombination on traditional phylogenetic analysis. *Genetics*, 156:879–891, 2000.

[19] Y. Song and J. Hein. On the minimum number of recombination events in the evolutionary history of DNA sequences. *Journal of Mathematical Biology (to appear)*, 2003.

[20] Y. Song and J. Hein. Parsimonious reconstruction of sequence evolution and haplotype blocks: Finding the minmimum number of recombination events. In *Proc. of 2003 Workshop on Algorithms in Bioinformatics*. Springer-Verlag LNCS, 2003.

[21] L. Wang, K. Zhang, and L. Zhang. Perfect phylogenetic networks with recombination. *Journal of Computational Biology*, 8:69–78, 2001.