A Fundamental Decomposition Theory for Phylogenetic Networks and Incompatible Characters

Dan Gusfield[1] and Vikas Bansal[2]

[1] Department of Computer Science, University of California, Davis
gusfield@cs.ucdavis.edu
[2] Department of Computer Science and Engineering, University of California, San Diego
vibansal@cs.ucsd.edu

Keywords: Molecular Evolution, Phylogenetic Networks, Perfect Phylogeny, Ancestral Recombination Graph, Recombination, Gene-Conversion, SNP

Abstract. Phylogenetic networks are models of evolution that go beyond trees, allowing biological operations that are not consistent with tree-like evolution. One of the most important of these biological operations is recombination between two sequences (homologous chromosomes). The algorithmic problem of reconstructing a history of recombinations, or determining the minimum number of recombinations needed, has been studied in a number of papers [10–12, 23–25, 16, 13, 14, 6, 9, 8, 18, 19, 15, 1]. In [9, 6, 10, 8, 1] we introduced and used "conflict graphs" and "incompatibility graphs" to compute lower bounds on the minimum number of recombinations needed, and to efficiently solve constrained cases of the minimization problem. In those results, the non-trivial connected components of the graphs were the key features that were used. In this paper we more fully develop the structural importance of non-trivial connected components of the incompatibility graph, to establish a fundamental decomposition theorem about phylogenetic networks. The result applies to phylogenetic networks where cycles reflect biological phenomena other than recombination, such as recurrent mutation and lateral gene transfer. The proof leads to an efficient $O(nm^2)$ time algorithm to find the underlying maximal tree structure defined by the decomposition, for any set of n sequences of length m each. An implementation of that algorithm is available. We also report on progress towards resolving the major open problem in this area.

1 Introduction to Phylogenetic Networks and Problems

With the growth of genomic data, much of which does not fit ideal evolutionary-tree models, and the increasing appreciation of the genomic role of such phenomena as recombination, recurrent and back mutation, horizontal gene transfer, cross-species hybridization, gene conversion, and mobile genetic elements, there is greater need to understand the algorithmics and combinatorics of phylogenetic networks on which extant sequences were derived [20]. Recombination is particularly important in deriving chimeric sequences in a population of individuals of the same species. Recombination in populations is the key element underlying techniques that are widely hoped to locate genes influencing genetic diseases.

Formal definition of a phylogenetic network

There are four components needed to specify a phylogenetic network that allows multiple-crossover recombination (see Figure 1).

A 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 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. There may be additional edges that are assigned no integers. 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 some sequence R, called the "root" or the "ancestral" 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 the state (from 0 to 1, or from 1 to 0) of the value at site i, for every integer i on edge e. This corresponds to a mutation at site i occurring on edge e.

b) For the recombination at node x, let Z and Z' denote the two m-length sequences labeling the parents of x. Then the "recombinant sequence" X labeling x can be any m-length sequence provided that at every site i, the character in X is equal to the character at site i in (at least) one of Z or Z'.

The "event" that creates X from Z and Z' is called a "multiple-crossover recombination". To fully specify the event, we must specify for every position i whether the character in X "comes from" Z or Z'. This specification is forced when the characters in Z and Z' at position i are different. When they are the same, a choice must be specified. For a given event, we say that a *crossover* occurs at position i if the characters at positions $i-1$ and i come from different parents. It is easy to determine the minimum number of crossovers needed to create X by a recombination of Z and Z'.

The sequences labeling the leaves of N are the extant sequences, i.e., the sequences that can be observed. We say that an (n, m)-phylogenetic network N *derives (or explains)* a set of n sequences M if and only if each sequence in M labels one of the leaves of N.

With these definitions, the classic "perfect phylogeny" [4] is a phylogenetic network without any recombinations. That is, each site mutates exactly once in the evolutionary history, and these is no recombination between sequences.

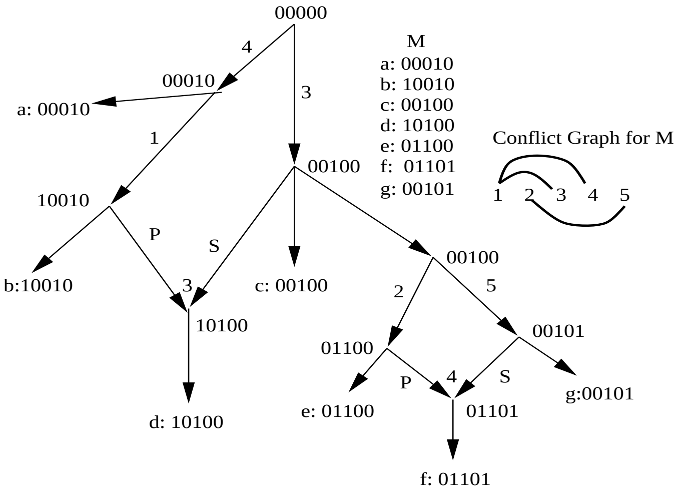

Fig. 1. A phylogenetic network that derives the set of sequences M. The two recombinations shown are single-crossover recombinations, and the crossover point is written above the recombination node. In general the recombinant sequence exiting a recombination node may be on a path that reaches another recombination node, rather than going directly to a leaf. Also, in general, not every sequence labeling a node also labels a leaf.

There are two restricted forms of recombination that are of particular biological interest. One is where X is formed from a *prefix* of one of its parent sequences (Z or Z') followed by a *suffix* of the other parent sequence. This is called "single-crossover recombination" since it uses exactly one crossover, and it is the definition of recombination used in [9, 8]. The other case is when X is formed from a prefix of one parent sequence, followed by an internal segment of the other parent sequence, followed by a suffix of the first parent sequence. This is a two-crossover recombination, which occurs during "gene-conversion" in meiosis, and during some forms of "lateral gene-transfer". Multiple-crossover recombination allows the modeling of complex biological phenomena, and hence the main result in this paper applies to many causes of incompatibility besides recombination.

What we have defined here as a phylogenetic network with single-crossover recombination is the digraph part of the stochastic process called an "ancestral recombination graph (ARG)" in the population genetics literature.

In the context of meiotic recombination, the assumption that the sequences are binary is motivated today by the importance of SNP data, where each site can take on at most two states (alleles) [2]. In the context of macroevolution, complex evolutionary characters are usually considered to be binary (either present or absent)[3].

Rooted and Root-Unknown problems Problems of reconstructing phylogenetic networks, given an input set of binary sequences M, can be addressed either in the rooted case, or the root-unknown case. In the *rooted* phylogenetic network problem, a required root or ancestral sequence R for the network is specified in advance. In the *root-unknown* phylogenetic network problem, no ancestral sequence is specified in advance, and the algorithm must select an ancestral sequence.

2 A Fundamental Decomposition Theory for Phylogenetic Networks and Incompatible Characters

In this section we define and derive the main result of this paper, that for any input M, there always is a phylogenetic network of an important, natural structure. We believe this to be a very fundamental fact about phylogenetic networks that will have many applications. We now begin the needed definitions that lead to the statement of the main result.

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. In Figure 1, the nodes labeled 00000 and 00100 are coalescent nodes of two different recombination cycles.

If a recombination cycle in a phylogenetic network N is not isolated (a "gall" in the terminology of [9]), it shares at least one edge with some other recombination cycle. We can add another cycle to that blob if the new cycle shares an edge with at least one cycle already on the blob. Continuing in this way, we ultimately get a maximal set of recombination cycles in N that form a single connected subgraph of N, and each cycle shares at least one edge with some other cycle in the set. We call such a maximal set of cycles a "blob".

Clearly, because of maximality, the blobs in a phylogenetic network N are well-defined. Moreover, if we contract each blob in N to a single point, the resulting network is a directed tree T'. This follows because if the resulting graph had a cycle (in the underlying undirected graph) that cycle would correspond to a recombination cycle which should have been contracted. We call T' a "tree of blobs" or a "blobbed tree". So every phylogenetic network N can be viewed as a blobbed tree. The edges in T' are called "tree edges" of N.

2.1 The main tools

The main tools that we used in [9, 10, 1] and other papers were two graphs representing "incompatibilities" and "conflicts" between sites. We introduce these graphs here.

Given a set of binary sequences M, two columns i and j in M are said to be *incompatible* if and only if there are four rows in M where columns i and j contain all four of the ordered pairs 0,1; 1,0; 1,1; and 0,0. For example, in Figure 1 columns 1 and 3 of M are incompatible because of rows a, b, c, d. The test for the existence of all four pairs is called the "four-gamete test" in the population genetics literature. A site that is not involved in any incompatibility is called a "compatible site".

Given a sequence S, two columns i and j in M are said to *conflict (relative to S)* if and only if columns i and j contain all three of the above four pairs that differ from the i, j pair in S.

The classic Perfect Phylogeny Theorem (in the terminology of this paper) is that there is a root-unknown phylogenetic network without any recombination cycles, that derives a set of binary sequences M, if and only if there is no incompatible pair of columns. Similarly, there is a phylogenetic network with ancestral sequence S, without any recombination cycles, that derives M, if and only if there is no pair of columns that conflict relative to S. For one exposition of this classic result, see [5].

Incompatibility and Conflict Graphs

We define the "incompatibility graph" $G(M)$ for M as a graph containing one node for each column (site) in M, and an edge connecting two nodes i and j if and only if columns i and j are incompatible. Similarly, given a sequence S, we define the "conflict graph" $G_S(M)$ for M (relative to S) as a graph containing one node for each column in M, and an edge connecting two nodes i and j if and only if columns i and j conflict relative to S. Figure 1 shows the conflict graph relative to the all-zero sequence S. This conflict graph is also the incompatibility graph for M.

A "connected component" (or "component" for short), C, of a graph is a maximal subgraph such that for any pair of nodes in C there is at least one path between those nodes in the subgraph. A "trivial" component has only one node, and no edges. The conflict graph in Figure 1 has two components.

2.2 Main Result

Theorem 1. *Let $G(M)$ be the incompatibility graph for M. Then, there is a phylogenetic network N that derives M where every blob contains all and only the sites of a single non-trivial connected component of $G(M)$, and every compatible site is on a tree edge of N.*

Stated another way, for any input M, there is a blobbed-tree that derives M, where the blobs are in one-one correspondence with the non-trivial connected components of $G(M)$, and if B is the blob corresponding to component C, then B contains all and only the sites in C. We call a network "fully-decomposed" if it has the structure specified in Theorem 1.

Theorem 1 is an extension of the stronger theorem proved in [9] about galled-trees. In the case of galled-trees, *every* reduced galled-tree for M *must* be fully-decomposed. A galled-tree is "reduced" if every recombination cycle contains some incompatible sites. When there is a galled-tree for M, there is a reduced galled-tree for M, and the program galledtree.pl will produce one (see Section 3).

There is an analogous theorem to Theorem 1 in the case that the ancestral sequence S is known in advance. In that case, there is a phylogenetic network N that derives M, with ancestral sequence S, where the blobs in N are in one-one correspondence with the non-trivial connected components of $G_S(M)$, and any non-conflicting site is on a tree edge of N.

3 Proof of Theorem 1

Let C and C' be two connected components in the incompatibility graph $G(M)$. Note that either C or C' or both may be a trivial connected component, i.e., consist of only a single node.

For any $i \in C, i' \in C'$ let $(X, \overline{X})$ and $(Y, \overline{Y})$ be the respective bipartitions (of the rows of M), associated with sites i and i'. The two bipartitions cannot be identical, for otherwise sites i and i' would have exactly the same incompatibilities and so be in the same connected component. Each of the four subsets $X, \overline{X}, Y, \overline{Y}$ is called a "class" of the bipartition it is part of. Sites i and i' are not incompatible, so one class of the i bipartition must strictly contain one class of the i' bipartition, and the other class of the i' bipartition must strictly contain the other class of the i bipartition. Without loss of generality, suppose $X \supset Y$ and $\overline{Y} \supset \overline{X}$. We say that X is the "dominant" class of i, and $\overline{X}$ is the "dominated" class, with respect to the pair i, i'. Similarly, $\overline{Y}$ is the dominant class of i', and Y is the dominated class, with respect to the pair i, i'.

Lemma 1. *Let i, i', X, and Y be as above. Let j' be any site in C', and let $(Z, \overline{Z})$ be the bipartition associated with j'. Then, the dominant class of i with respect to the pair i, j' is the dominant class of i with respect to the pair i, i'. That is, either $X \supset Z$ or $X \supset \overline{Z}$.*

Proof. The Lemma is vacuously true if C' is a trivial connected component, so assume C' is non-trivial, and consider a site $k' \in C'$ that is incompatible with i'. Such a site k' must exist since C' is connected. Let $(W, \overline{W})$ be the bipartition defined by site k' If X is not dominant with respect to i, k', then $\overline{X}$ is dominant with respect to i, k', and so either $\overline{X} \supset W$ or $\overline{X} \supset \overline{W}$. Suppose that $\overline{X} \supset \overline{W}$, so $W \supset X$. But then $W \supset Y$ since $X \supset Y$, and so $Y \cap \overline{W} = \emptyset$, and i and k' can't be incompatible, which is a contradition. Similarly, if $\overline{X} \supset W$, then $\overline{W} \supset X$, so $\overline{W} \supset Y$, and $W \cap \overline{Y} = \emptyset$, a contradiction. So the dominant class, X, with respect to i, i' is the dominant class with respect to i, k', where k' is any site that is incompatible with i'. The Lemma now follows by transitivity, because C' is a connected component, so from i' it is possible to reach any $j' \in C'$ by a series of incompatibility relations. □

Lemma 1 establishes that for any $i \in C$, one class of i is dominant with respect to *all* sites in C', and symmetrically, for any $i' \in C'$ one class of i' is dominant with respect to *all* sites in C. So, with respect to the (C, C') pair of connected components, each site in $C \cup C'$ has a well-defined dominant class, and a well-defined dominated class.

Now return focus to the sequences in M and the sites in C and C'. For a site $i \in C$, the bipartition $(X, \overline{X})$ is encoded with 0's and 1's, where all the rows in X have one character at site i and all the rows in $\overline{X}$ have the other character at site i. So, with respect to the (C, C') pair of connected components, and a specific set of sequences M, each site in C has a well-defined *dominant character* (either 0 or 1). For example, in Figure 2, the dominant character is 0 in all sites except 3, where the dominant character is 1.

For $i \in C$, let $D(i)$ be the rows in the dominated class with respect to (C, C'), and similarly, for $i' \in C'$, let $D(i')$ be the rows in the dominated class with respect to (C, C'). Let $D[C, C']$ be the union of the rows in any dominated class of C, with respect to (C, C'). Similarly, let $D[C', C]$ be union of the rows in any dominated class of C', with respect to (C, C').

Let $M(C)$ and $M(C')$ be the sequences in M, restricted to the sites in C and C' respectively. Then Lemma 1 implies

Theorem 2. *Every row in $D[C, C']$ has the same sequence in $M(C')$. In particular, in each row of $D[C, C']$, every site $i' \in C'$ has the dominant character with respect to (C, C'). Similarly, every row in $D[C', C]$ has the same sequence in $M(C)$. In particular, in each row of $D[C', C]$, every site $i \in C$ has the dominant character with respect to (C, C').*

Given Theorem 2, we can define the *dominant sequence* in $M(C)$ with respect to (C, C') as the sequence in $M(C)$ where each site has the dominant character with respect to (C, C'). Similarly, we can define the dominant sequence in $M(C')$ with respect to (C, C').

Corollary 1. *Let C and C' be two connected components of $G(M)$. There is no row in M which contains both a non-dominant sequence in $M(C)$ and a non-dominant sequence in $M(C')$ with respect to (C, C').*

Figure 2 illustrates Lemma 1, Theorem 2 and Corollary 1. Note that a row can have the dominant sequence in $M(C)$ and the dominant sequence in $M(C')$. Row c in Figure 2 is an example of this.

	1	3	4	2	5
a	0	0	1	0	0
b	1	0	1	0	0
d	1	1	0	0	0
c	0	1	0	0	0
e	0	1	0	1	0
f	0	1	0	1	1
g	0	1	0	0	1

Fig. 2. The sites in the two connected components from Figure 1. We denote the component with sites $\{1, 3, 4\}$ as C, and the component with sites $\{2, 5\}$ as C'. The dominant sequence for C is 010, and the dominant sequence for C' is 00. The rows in $D[C, C']$ are $\{a, b, d\}$, and the rows in $D[C', C]$ are $\{e, f, g\}$. Note that row c is in neither $D[C, C']$ nor $D[C', C]$, since row c has the dominant sequence in both its C and C' sides.

Lemma 1, Theorem 2 and Corollary 1 establish a structure that exists in M, imposed by the partition of the columns of M by the connected components of $G(M)$, the incompatibility graph of M. We begin now to exploit that structure to prove the main theorem. We will create a new, binary, matrix MB from

M and $G(M)$. Let C be a connected component of $G(M)$ and let $M(C)$ be the sequences in M restricted to the sites in C. Create one column in MB for each *distinct* sequence in $M(C)$. Each such new column, associated with a sequence $S \in M(C)$ say, encodes a bipartition of the rows of M, where one side of the bipartition contains all the rows that have sequence S in $M(C)$, and the other side of the bipartition contains the remaining rows. More specifically, and without loss of generality, in the new column we assign value 1 to each row which contains sequence S in $M(C)$, and assign value 0 to each row that does not. The new column defines a binary character derived from M and $G(M)$. Note that if C is a trivial connected-component, so it only contains one site, then MB will have two columns derived from that one site, but those columns define the same bipartition. That will cause no problems, and one can be removed for simplicity.

Matrix MB is defined by the columns described above, over the set of all connected components in $M(G)$. We call a character (column, site) of MB a "super-character". We want to use these super-characters to build a tree that will prove Theorem 1. We start by showing

Lemma 2. *No pair of super-characters are incompatible.*

Proof. Let p and q be super-characters in MB, and let $(P, \overline{P})$ and $(Q, \overline{Q})$ be the bipartitions associated with p and q. If p and q originate from the same connected component C in $G(M)$, then without loss of generality, the rows in P all have the same sequence in $M(C)$, and the rows in Q all have the same sequence in $M(C)$, and those two sequences are different. Hence, $P \cap Q = \emptyset$, and so p and q are not incompatible.

Now suppose p and q originate from two different connected components C and C' in $G(M)$. If p and q both originate from non-dominant sequences of C and C', then Corollary 1 guarantees that there is no row with $1, 1$ in columns p and q, and so p and q cannot be incompatible. Symmetrically, if p and q both originate from dominant sequences in C and C', then there is no row with $0, 0$ in columns p and q. If p originates from the dominant sequence of C and q originates from a non-dominant sequence of C', then there can be no $0, 1$ in columns p and q. The remaining case is symmetric. □

Hence, by the Perfect Phylogeny Theorem, there is a *unique* perfect phylogeny $\overline{T}$ where each super-character labels an edge in $\overline{T}$, and each edge is labeled by one or more super-characters[3].

We now develop the structure of $\overline{T}$ to both complete the proof of Theorem 1, and to constructively show how to build a network N for M from $\overline{T}$. A "split of edge e" is defined as the bipartition of the leaves resulting from the removal of edge e from $\overline{T}$. Note that all the splits of the edges in $\overline{T}$ are distinct. The removal of any edge e in $\overline{T}$ creates two connected subtrees, whose leaves correspond to

[3] It is of independent interest to note that we have established that the super-characters defined by the connected components of $G(M)$ generalize the standard (tree) characters and play a role in the theory of phylogenetic networks, that tree characters play in the theory of phylogenetic trees.

the two classes of the split of edge e. If e is labeled by super-character C, we define the "1-side" of e as the subtree of $\overline{T} - e$ that contains the leaves for rows in MB that have value 1 for super-character C. The other side is called the "0-side" of the split.

Lemma 3. *In $\overline{T}$, there is a node v_C such that all the edges labeled by super-characters that originate from the same connected component C in $G(M)$ are incident with v_C. That is, these edges form a star around a single central node v_C. Further, v_C is on the 0-side of each split defined by every super-character that originates from C.*

Proof. First, the Lemma is trivialy true if C is a trivial component. Note, however, that any non-trivial connected component has at least four distinct super-characters. Consider such a connected component C and any three of its super-chars, and let e_1, e_2, e_3 be the three edges in $\overline{T}$ labeled with those super-chars. Note that every row in MB has value 1 in exactly one column of $MB(C)$, so every leaf of $\overline{T}$ is on the 1-side of exactly one edge labeled by a super-character that derives from C. Hence, no leaf in $\overline{T}$ can be on the 1-side of two of the edges e_1, e_2, e_3.

If e_1 and e_2 are incident with each other, sharing a node v, then there must be another edge incident with node v, and hence there must be a leaf l_v that is reachable from v without going through e_1 or e_2. If this were not true, then e_1 and e_2 would define the same splits in $\overline{T}$, which is not possible. If e_1 and e_2 are not incident with each other, then there is a unique shortest path P from an endpoint of e_1 to an endpoint of e_2. Clearly, path P does not contain edge e_1 or e_2. There must be a node v on P and a leaf l_v that is reachable from v via a path that does not go through e_1 or e_2. If this were not true, then again there would be two adjacent edges that define the same splits in $\overline{T}$.

Now we claim that node l_v must be on the 0-side of both e_1 and e_2. We have already established that it cannot be on the 1-side of both, since no leaf can be on the 1-side of two splits derived from the super-characters of C. However, suppose without loss of generality, that l_v is on the 1-side of e_1 and the 0-side of e_2. Then consider the endpoint u of e_2 that is on the 1-side of e_2, and consider a leaf l_u that is reachable from u without going through e_2. Leaf l_u would be on the 1-side of both e_1 and e_2, which is not possible. Hence the 1-sides of both e_1 and e_2 point "away" from each other. It also follows that path P cannot go through edge e_3. If it did, then some leaf on the 1-side of e_3 would also be on the 1-side of e_1 or e_2.

So edges e_1 and e_2 are either incident with each other, or there is an edge e which is incident with e_1 on path P, where e is not labeled by a super-character from C. We will show that such an edge e cannot exist. All internal edges in $\overline{T}$ are labeled by some super-character, so suppose e exists and is labeled by a super-character that derives from a connected component C'. Let v be the common endpoint of e_1 and e. As above, there must be a leaf l_v that is reachable from v without going through either edge e or e_1, for otherwise e and e_1 define the same split and should not be separate edges in $\overline{T}$. Recall that each super-character and each split in $\overline{T}$ that derives from C or C' corresponds to a sequence in $M(C)$ or $M(C')$, and with respect to the pair (C, C'), there is a dominant sequence S in $M(C)$ and a dominant sequence S' in $M(C')$. Let $e(S)$ be the edge in $\overline{T}$ labeled by the super-character for S, and let $e(S')$ be the edge in $\overline{T}$ labeled by the super-chararacter for S'. Now e_1 is either $e(S)$ or not, and e is either $e(S')$ or not, so we have four cases to consider.

Case 1: Suppose e_1 is $e(S)$ and e is $e(S')$. We know that $l(v)$ is on the 0-side of e_1, so it must be on the 1-side of e to obey Corollary 1. But then, all leaves on the 1-side of e will be on the 0-side of both e and e_1, which contradicts Corollary 1. The other cases are similar. So e cannot exist, and hence e_1 and e_2 are incident with each other. The three other cases are similar and omitted.

Since e_1 and e_2 were arbitrary edges labeled by super-characters derived from connected component C, every pair of edges labeled by super-characters from C must be incident with each other. But in a tree, that is only possible if all those edges share exactly one endpoint, and so form a star around a single center. That endpoint is the claimed node v_C. Also, we established that if there are two distinct edges labeled with super-characters derived from C, then the 1-sides of these edges point away from each other. This holds for any pair of edges labeled with super-characters derived from C, so v_C is on the 0-side of every such edge. □

To finish the proof of Theorem 1, we first arbitrarily select a leaf in $\overline{T}$ to act as the root node, and we direct every edge away from that leaf. This also defines an ancestral sequence for the phylogenetic network we will construct. Next, we need to inflate each node v_C in $\overline{T}$ that is the central node of the star associated with the super-characters of a *non-trivial* connected component of $M(G)$. We can identify such central-star nodes by the fact that for some non-trivial connected component C, all of the edges labeled by the super-characters that derive from C are incident with a node v, and hence that node must be v_C. Each such edge may also be labeled with the super-character that derives from another connected component or with a compatible character. However, every leaf is on the 1-side of exactly one super-character that derives from C, and v_C is on the 0-side of each such super-character, so there can be no no edge $e = (v_C, v')$ in $\overline{T}$ that is labeled only by a compatible site. If there was such an edge, then a leaf reached from the v' without going through v_C would not be on the 1-side of any super-character that derives from C.

Note that each central-star node v_C has exactly one edge directed into it. We call the sequence in $M(C)$ on that edge the "ancestral sequence" of v_C. Now, any sequence in $M(C)$ can be derived from the ancestral sequence of v_C using at most one mutation per site, if enough recombinations are allowed. So, each central-star node v can be inflated into a blob B_v containing one node labeled by each distinct sequence in $M(C)$ (and other nodes if needed). Then for each distinct sequence in $M(C)$ we connect the node in B_v labeled with that sequence to the edge (incident with v_C) that is labeled by the super-character for that sequence in $M(C)$.

After inflating each central-star node in $\overline{T}$, the end result is a phylogenetic network N where each blob contains all and only the sites from one connected component of $G(M)$. Every compatible site labels a tree edge of N. This completes the proof of Theorem 1.

Uniqueness We leave the proof to the reader, but it is also true that if N is a fully-decomposed network and T' is created by contracting each blob of N to a single node, then after the directed edges in T' are made undirected, the resulting tree is necessarily $\overline{T}$. So $\overline{T}$ is the *invariant* underlying structure of any fully-decomposed phylogenetic network for M.

Programs The above proof of the existence of $\overline{T}$ can be converted into an efficient, constructive method[4] for finding $\overline{T}$ from any input M. The program galledtree.pl, available at wwwcsif.cs.ucdavis.edu/~gusfield/ takes in a set of sequences M and tries to build a galled-tree for M. If it succeeds, then it has produced a complete phylogenetic network for M where each blob is a single cycle, and the cycles are node disjoint. Hence, the program produces a fully-decomposed phylogenetic network for M. If the program determines that there is no galled-tree for M, then it outputs the tree $\overline{T}$ for M. The running time for the program is $O(nm^2 + m^3)$, but the time used to build $\overline{T}$ is just $O(nm^2)$.

4 What is the "most tree-like" phylogenetic network?

When a set of sequences M fails the four-gametes test and hence cannot be generated on a perfect phylogeny, one would still like to derive the sequences on a phylogenetic network that is the "most tree-like". There is no accepted definition of "treeness", and under many natural definitions, the problem of finding the most tree-like network would likely be computationally difficult. In this section, we introduce a measure of treeness and relate it to Theorem 1.

Given a phylogenetic network N we first modify N so that no two blobs share a node. The only way that two blobs can share a node v is if v is the "root" of one of the blobs, so we can always add a new edge to separate the two blobs. We can also assume that N has no node with in and out-degrees that are both one. Then if each blob is contracted to a single node, the number of edges in the resulting directed tree measures the "treeness" of N. In other words, the "treeness" of N is measured by the size of the tree in the underlying tree structure of N. For example, if all the sites in M are in a single blob in N, then N is less tree-like than a network where the sites are distributed between several blobs, connected by several edges in a tree structure.

With the above definition of "treeness", we claim that a phylogenetic network N is "the most tree-like" if and only if $\overline{T}$ is the resulting undirected tree, after the blobs of N are contracted, and all the edges are made undirected. This follows from Theorem 1 and that fact that all the sites in a single non-trivial con-

[4] It may seem that $\overline{T}$ can be obtained by simply building a perfect phylogeny T using one site from each connected component of $G(M)$. This does not work because the edge structure of T may be very different from that of $\overline{T}$. For example, in the tree T created from sites 1 and 2 in Figure 1, the two edges labeled with those sites are adjacent, while they are not adjacent in $\overline{T}$.

nected component of $G(M)$ *must* be together in a single blob in any phylogenetic network. This second fact is proven in [9].

This definition of "most tree-like" is somewhat crude because it does not consider any details inside of a blob, but it has the advantage of being easy to compute and allowing a clear identification of the most tree-like networks. Further, it seems reasonable that any other natural definition of "most tree-like" would identify a *subset* of the networks identified by the definition considered here.

5 Alternative Proofs of Theorem 1

We believe that Theorem 1 has not previously been stated in any published literature, but Mike Steel has pointed out that Theorem 1 can be proven by using Buneman graphs [21], and the details of this approach have been worked out by Yufeng Wu at UC Davis. However, it takes exponential time in worst case to build a Buneman graph from M, and so this is not an efficient constructive approach. Andreas Dress and Hiroshi Hirai have also pointed out that Theorem 1 can be derived from the framework of block-decompositions in T-theory. Finally, Daniel Huson and Mike Steel have (subsequent to the development of Theorem 1) recently developed a related decomposition theory for splits graphs, where the input to the problem is not a set of sequences, but a set of trees that must be subtrees in a constructed phylogenetic network. Problems of that type have been studied in [19, 15].

6 Theorem 1 Applies in Diverse Biological Contexts

Theorem 1 was proven in the context of multiple crossover recombination when explicit binary sequences are given as input. This is most directly motivated by the evolution of sequences of SNPs (single nucleotide polymorphisms). SNP sequences evolve in a population by site mutation and by (meiotic) recombination of homologous chromosomes (single crossover recombination), and by gene conversion (a specific kind of two crossover recombination). However, multiple crossover recombination can be considered as a mathematical operation on binary sequences, rather than a biological event, and can be used to model biological events that don't explicitly involve recombination. As a consequence, Theorem 1 holds in many biological contexts where diverse biological events cause incompatibility between sites (or more generally, incompatibility between binary evolutionary "characters"). Three such biological events are "back-mutation", "recurrent-mutation", and "lateral gene transfer", and we consider the first two of those below.

6.1 Back and Recurrent Mutation

"Back-mutation" occurs when the state of a site mutates back from its derived state to its ancestral state. "Recurrent-mutation" occurs when the state of a site is permitted to mutate from its ancestral state more than once in an evolutionary history. Because there is no explicit recombination, the underlying graph of a network with back or recurrent mutation is a tree. Generally, when back or recurrent mutation is the cause of incompatibility, we seek an evolutionary tree that derives a given set of sequences using as few back or recurrent mutations as possible. Such a tree is called a "maximum parsimony tree" and it is a solution to the maximum parsimony problem [3, 21].

While biologically unrelated to recombination, each occurrence of back-mutation or recurrent-mutation of a site i in a sequence S can be *modeled* as a two-crossover recombination between S and some appropriate sequence, in the intervals $i-1, i$ and $i, i+1$. Modeling back and recurrent mutations in this way explicitly creates recombination cycles and blobs, and shows explicitly how Theorem 1 applies when back-mutation and/or recurrent mutation cause incompatibilities. The consequence is that one can derive M using a separate tree for the sites in each non-trivial connected component C of $G(M)$. The tree for each C derives the sequences in $M(C)$ using recurrent and/or back mutation if needed, and the separate trees can be connected using $\overline{T}$.

Note that when back or recurrent-mutation is modeled in this way, each recombination only changes a single site, so the linear order of the site has no impact on the permitted recombinations, and the ordering of the sites can be arbitrary. This allows Theorem 1 to apply to (binary) "evolutionary characters" which may experience back and/or recurrent mutation, but have no natural order.

7 Open Question and Conjecture

The main open question related to Theorem 1 is the following

> **Decomposition Optimality Conjecture**: For any M, there is always a fully-decomposed phylogenetic network for M that minimizes the number of recombinations used, over all possible phylogenetic networks for M.

Note, that the conjecture does not say that the minimum number of recombinations is equal to $\sum_C cc(C)$, where $cc(C)$ is the minimum number of recombinations needed in a phylogenetic network for $M(C)$. Such a stronger claim has been shown to be false [22]. The difficulty is that the separate solutions may choose ancestral sequences that cannot be combined into a single network.

The Decomposition Optimality Conjecture can be proven when the recombinations model recurrent and back mutations, as discussed earlier (one proof is based on the Buneman graph of M). Because of this result, when incompatibilities are caused by recurrent and/or back mutation, one can solve the parsimony problem separately for each connected component of $M(G)$, and then connect the trees as specified by $\overline{T}$. Since the parsimony problem is itself NP-hard, and the only known methods to solve it take exponential time in worst-case, decomposing the problem into several smaller problems may allow larger problems to be solved in practice.

If the Decomposition Optimality Conjecture is true in general (for any multiple-crossover recombinations), we could follow a similar approach to finding phylogenetic networks that minimize the number of recombinations. It is easiest to exploit the conjecture (if proved) in the case that an ancestral sequence A is given. In that case, we know the root of $\overline{T}$ and hence the ancestral sequence for each of the blobs in the network for M. Hence we could solve a single (rooted) problem for each component of $G_A(M)$. When no ancestral sequence is known in advance, this approach would need to be repeated for each choice of root position in $\overline{T}$. If the conjecture is true, it would also follow that we could compute lower bounds on the number of needed recombinations by computing bounds separately for the sites on each connected component of $M(G)$, and then add these bounds together for a correct overall bound. This would be correct no matter what lower bound method is used. This approach has been proven correct for two specific lower bounds [1], strengthening the belief that the above conjecture is true.

Progress on Proving the Conjecture We have recently proven [7] a weaker version of the Decomposition Optimality Conjecture. We say that a node v in a phylogenetic network N for M is "visible" if the sequence labeling node v in N is a sequence in M.

Theorem 3. *If every node v in N is visible, then there is a fully-decomposed network for M which uses the same number of recombinations, or fewer, than does N.*

The theorem can be proven with somewhat weaker conditions than the visibility of all nodes in N. Also, a sufficient (but not necessary) condition for the visibility of all nodes is that the "haplotype lower bound" [16] on the minimum number of recombinations equals the true minimum. Simon Myers has shown [17] that under the neutral coalescent model with recombination, the expected difference between the haplotype bound and the true minimum is bounded by a *constant* as the number of sequences goes to infinity. Thus, there may be optimal phylogenetic networks where all nodes are visible, more often than might at first be assumed.

8 Acknowledgements

We would like to thank S. Eddhu for writing the program to construct $\overline{T}$. We thank Dean Hickerson, Chuck Langley, Vineet Bafna, Yun Song, Yufeng Wu, Zhihong Ding and Gabriel Valiente for helpful discussions and comments. Research of the first author is partly supported by NSF grant EIA-0220154.

References

1. V. Bafna and V. Bansal. The number of recombination events in a sample history: conflict graph and lower bounds. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1:78–90, 2004.
2. A. Chakravarti. It's raining SNP's, hallelujah? *Nature Genetics*, 19:216–217, 1998.
3. J. Felsenstein. *Inferring Phylogenies.* Sinauer, Sunderland, MA., 2004.
4. D. Gusfield. Efficient algorithms for inferring evolutionary history. *Networks*, 21:19–28, 1991.
5. D. Gusfield. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology.* Cambridge University Press, Cambridge, UK, 1997.

6. D. Gusfield. Optimal, efficient reconstruction of Root-Unknown phylogenetic networks with constrained recombination. Technical report, Department of Computer Science, University of California, Davis, CA, 2004.
7. D. Gusfield. On the decomposition optimality conjecture for phylogenetic networks. Technical report, UC Davis, Department of Computer Science, 2005.
8. D. Gusfield, S. Eddhu, and C. Langley. The fine structure of galls in phylogenetic networks. *INFORMS J. on Computing, special issue on Computational Biology*, 16:459–469, 2004.
9. D. Gusfield, S. Eddhu, and C. Langley. Optimal, efficient reconstruction of phylogenetic networks with constrained recombination. *J. Bioinformatics and Computational Biology*, 2(1):173–213, 2004.
10. D. Gusfield and D. Hickerson. A new lower bound on the number of needed recombination nodes in both unrooted and rooted phylogenetic networks. Report UCD-ECS-06. Technical report, University of California, Davis, 2004.
11. J. Hein. Reconstructing evolution of sequences subject to recombination using parsimony. *Math. Biosci*, 98:185–200, 1990.
12. J. Hein. A heuristic method to reconstruct the history of sequences subject to recombination. *J. Mol. Evol.*, 36:396–405, 1993.
13. 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.
14. J. D. Kececioglu and D. Gusfield. Reconstructing a history of recombinations from a set of sequences. *Discrete Applied Math.*, 88:239–260, 1998.
15. B. Moret, L. Nakhleh, T. Warnow, C.R. Linder, A. Tholse, A. Padolina, J. Sun, and R. Timme. Phylogenetic networks: Modeling, reconstructibility, and accuracy. *IEEE/ACM Transactions on Computatational Biology and Bioinformatics*, pages 13–23, 2004.
16. S. R. Myers and R. C. Griffiths. Bounds on the minimum number of recombination events in a sample history. *Genetics*, 163:375–394, 2003.
17. Simon Myers. *The detection of recombination events using DNA sequence data.* PhD thesis, University of Oxford, Oxford England, Department of Statistics, 2003.
18. L. Nakhleh, J. Sun, T. Warnow, C.R. Linder, B.M.E. Moret, and A. Tholse. Towards the development of computational tools for evaluating phylogenetic network reconstruction methods. In *Proc. of 8'th Pacific Symposium on Biocomputing (PSB 03), pages 315-326*, 2003.
19. L. Nakhleh, T. Warnow, and C.R. Linder. Reconstructing reticulate evolution in species - theory and practice. In *Proc. of 8'th Annual International Conference on Computational Molecular Biology*, pages 337–346, 2004.
20. D. Posada and K. Crandall. Intraspecific gene genealogies: trees grafting into networks. *Trends in Ecology and Evolution*, 16:37–45, 2001.
21. C. Semple and M. Steel. *Phylogenetics.* Oxford University Press, UK, 2003.
22. Y. Song. Personal Communication.
23. 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*, Berlin, Germany, 2003. Springer-Verlag LNCS.
24. Y. Song and J. Hein. On the minimum number of recombination events in the evolutionary history of DNA sequences. *Journal of Mathematical Biology*, 48:160–186, 2004.
25. L. Wang, K. Zhang, and L. Zhang. Perfect phylogenetic networks with recombination. *Journal of Computational Biology*, 8:69–78, 2001.