Available online at www.sciencedirect.com

ScienceDirect

Discrete Applied Mathematics 155 (2007) 806–830

DISCRETE APPLIED MATHEMATICS

www.elsevier.com/locate/dam

An efficiently computed lower bound on the number of recombinations in phylogenetic networks: Theory and empirical study

Dan Gusfield, Dean Hickerson, Satish Eddhu

University of California, Davis, CA 95616, USA

Received 30 June 2004; received in revised form 1 May 2005; accepted 1 May 2005
Available online 30 October 2006

Abstract

Phylogenetic networks are models of sequence evolution that go beyond trees, allowing biological operations that are not tree-like. One of the most important biological operations is recombination between two sequences. An established problem [J. Hein, Reconstructing evolution of sequences subject to recombination using parsimony, Math. Biosci. 98 (1990) 185–200; J. Hein, A heuristic method to reconstruct the history of sequences subject to recombination, J. Molecular Evoluation 36 (1993) 396–405; Y. Song, J. Hein, Parsimonious reconstruction of sequence evolution and haplotype blocks: finding the minimum number of recombination events, in: Proceedings of 2003 Workshop on Algorithms in Bioinformatics, Berlin, Germany, 2003, Lecture Notes in Computer Science, Springer, Berlin; Y. Song, J. Hein, On the minimum number of recombination events in the evolutionary history of DNA sequences, J. Math. Biol. 48 (2003) 160–186; L. Wang, K. Zhang, L. Zhang, Perfect phylogenetic networks with recombination, J. Comput. Biol. 8 (2001) 69–78; S.R. Myers, R.C. Griffiths, Bounds on the minimum number of recombination events in a sample history, Genetics 163 (2003) 375–394; V. Bafna, V. Bansal, Improved recombination lower bounds for haplotype data, in: Proceedings of RECOMB, 2005; Y. Song, Y. Wu, D. Gusfield, Efficient computation of close lower and upper bounds on the minimum number of needed recombinations in the evolution of biological sequences, Bioinformatics 21 (2005) i413–i422. Bioinformatics (Suppl. 1), Proceedings of ISMB, 2005, D. Gusfield, S. Eddhu, C. Langley, Optimal, efficient reconstruction of phylogenetic networks with constrained recombination, J. Bioinform. Comput. Biol. 2(1) (2004) 173–213; D. Gusfield, Optimal, efficient reconstruction of root-unknown phylogenetic networks with constrained and structured recombination, J. Comput. Systems Sci. 70 (2005) 381–398] is to find a phylogenetic network that derives an input set of sequences, minimizing the number of recombinations used. No efficient, general algorithm is known for this problem. Several papers consider the problem of computing a *lower bound* on the number of recombinations needed. In this paper we establish a new, efficiently computed lower bound. This result is useful in methods to estimate the number of needed recombinations, and also to prove the optimality of algorithms for constructing phylogenetic networks under certain conditions [D. Gusfield, S. Eddhu, C. Langley, Optimal, efficient reconstruction of phylogenetic networks with constrained recombination, J. Bioinform. Comput. Biol. 2(1) (2004) 173–213; D. Gusfield, Optimal, efficient reconstruction of root-unknown phylogenetic networks with constrained and structured recombination, J. Comput. Systems Sci. 70 (2005) 381–398; 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]. The lower bound is based on a structural, combinatorial insight, using only the site conflicts and incompatibilities, and hence it is fundamental and applicable to many biological phenomena other than recombination, for example, when *gene conversions* or *recurrent or back mutations* or *cross-species hybridizations* cause the phylogenetic history

E-mail address: gusfield@cs.ucdavis.edu (D. Gusfield).

0166-218X/$ - see front matter
doi:10.1016/j.dam.2005.05.044

to deviate from a tree structure. In addition to establishing the bound, we examine its use in more complex lower bound methods, and compare the bounds obtained to those obtained by other established lower bound methods.

Keywords: Molecular evolution; Phylogenetic networks; Ancestral recombination graph; Recombination; Hybridization; SNP

1. Introduction to phylogenetic networks

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 [25,26]. Recombination is particularly important in deriving chimeric sequences in a population of individuals of the same species, rather than across species. Recombination in populations is the key element underlying techniques that are widely hoped to locate genes influencing genetic diseases. *Hybridization* is a similar phenomenon, creating chimeric sequences, but operates between species [19,21,22,24].

Hein [14,15] introduced the *phylogenetic network problem* (*with recombination*): construct a phylogenetic network that derives a given set of binary sequences, minimizing the number of recombinations used, although the model for allowed mutations in not same in all of the papers. No efficient, general algorithm is known for the problem, and when only a single mutation is allowed per site over the history of the sequences (the model we consider in this paper), the problem is NP-hard [32]. The minimization criterion is motivated by the general utility of parsimony in biological problems, and because most evolutionary histories are thought to contain a small number of observable recombinations. At the population level, the assumption that the sequences are binary is motivated today by the importance of single nucleotide polymorphism (SNP) data, where each site can take on at most two states (alleles) [4]. At the species level, the assumption that the sequences are binary is motivated by the evolution of complex traits [5]. The assumption that only a single mutation is allowed per site comes from the "infinite-sites" assumption in population genetics [30], which is strongly believed to be appropriate in the case of human SNP data [16].

An exact, superexponential-time, method has been developed [28] for the phylogenetic networks problem, but it is practical only on small problem instances. The problem can be solved in polynomial time when the sequences can be derived on a "galled-tree" [8,12,13], a condition that mostly often occurs when the recombination rate is low.

Since there is no efficient solution to the phylogenetic network problem, several papers have developed methods to compute *lower bounds* on the number of needed recombinations [18,20,27]. We will discuss some of these methods in Section 6.

In this paper, we develop a new, fundamental, efficiently computed lower bound on the number of needed recombinations, which has both theoretical and practical utility. The lower bounds are mainly discussed in the context of recombination, but the results are based on structural, combinatorial properties of site conflicts and incompatibilities, and hence apply to many biological phenomena that cause a deviation from the pure (perfect phylogeny) tree model.

Formal definition of a phylogenetic network: There are four components needed to specify a phylogenetic network (see Fig. 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, and conversely, an edge may be assigned more than one integer. 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

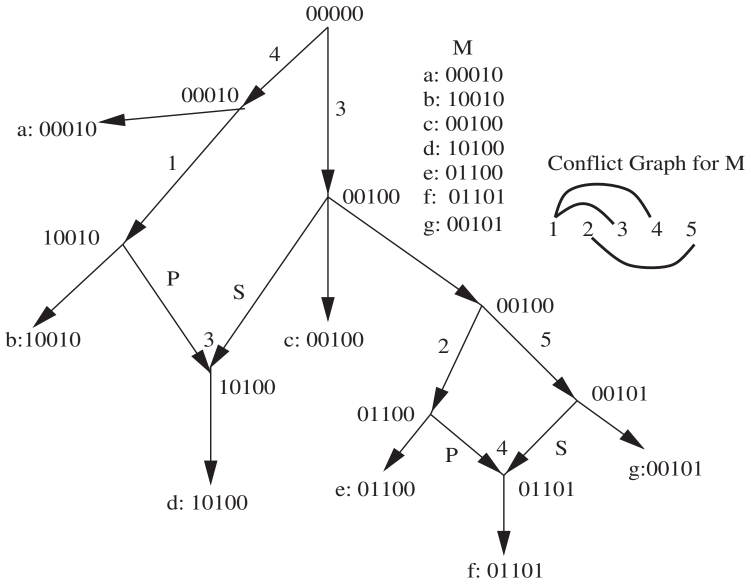

Fig. 1. A phylogenetic network that derives the set of sequences *M*. The two recombinations shown are single-crossover recombinations, and the crossover point is written above the recombination node. In general, the recombinant sequence exiting a recombination node may be on a path that reaches another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

set of 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) at site i, for every integer i on edge e. This corresponds to a mutation at site i occurring on edge e. The requirement that each site appears on one edge corresponds to the requirement that a site can mutate at most once in the history of the sequences [30,16].

(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 state of i in X is equal to the character at site i in either Z or Z'. The creation of X from Z and Z' is called a "multiple-crossover recombination".

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.

What we have defined here as a phylogenetic network is the digraph part of the stochastic process called an "ancestral recombination graph" in the population genetics literature (see [23] for example). With these definitions, the classic "perfect phylogeny" [6] is a phylogenetic network without any recombinations.

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" (or "crossing-over" in genetics) since it uses exactly one crossover. 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 and often occurs due to "gene-conversion" in meiosis. In a different biological context, what we have defined as two-crossover recombination models the biological phenomena of "lateral gene-transfer" and "hybridization speciation".

The phylogenetic network problems studied in [13–15,18,20,27,28] assume that recombination is a single-crossover recombination. The lower bounds we prove in this paper are for multiple-crossover recombination, and hence also hold for single-crossover recombination and gene-conversion.

1.1. Rooted and root-unknown problems

The phylogenetic network problem is to construct a network (as defined above) that derives the input set of sequences, *M*, minimizing the number of single-crossover recombinations used. That problem 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 problem is to select an ancestral sequence *R*, so that a phylogenetic network for *M* with ancestral sequence *R* minimizes the number of recombination nodes over all phylogenetic networks for *M*, and over all choices of ancestral sequence.

2. Introduction to tools

The main tools that we use are two graphs representing "incompatibilities" and "conflicts". We introduce these graphs here.

Given a set of input 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 Fig. 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.

Given a sequence *R*, two columns *i* and *j* in *M* are said to *conflict* (*relative to R*) if and only if columns *i* and *j* contain all three of the above four pairs that differ from the i, j pair in *R*. We call this the "three-gamete test".

Clearly, if a pair of columns i, j are incompatible, then i, j conflict relative to any sequence *R*. However, i, j may conflict relative to some sequence *R*, even though i, j are not incompatible.

2.1. The incompatibility and conflict graphs

We define the "incompatibility graph" $G(M)$ for *M* 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* are incompatible. Similarly, given a sequence *R*, we define the "conflict graph" $G_R(M)$ for *M* (relative to *R*) 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 *R*. Fig. 1 shows the conflict graph relative to the all-zero sequence *R*. 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 Fig. 1 has two components. We let $cc_R(M)$ and $cc(M)$ be the number of *non-trivial* components in $G_R(M)$ and $G(M)$, respectively.

For any *M* and any *R*, the edge set of $G_R(M)$ contains the edge set of $G(M)$, and this containment may be strict. Given a specified ancestral sequence *R*, let $M + R$ be the matrix *M* with the sequence *R* appended as a new row. Clearly, a pair of columns i, j are incompatible in $M + R$ if and only if they conflict (relative to *R*) in *M*. Therefore, $cc_R(M) = cc(M + R)$.

We define $m(M)$ as the minimum number of recombination nodes needed by any phylogenetic network that derives *M*. Note that the choice of the ancestral sequence is not specified in advance, but is chosen to minimize the number of recombination nodes. Given a specified ancestral sequence *R*, we define $m_R(M)$ as the minimum number of recombination nodes needed by any phylogenetic network that derives *M*, using the ancestral sequence *R*. Note that a multiple-crossover recombination is allowed at each recombination node.

Clearly, for any *M* and any *R*, $m(M) \leqslant m_R(M)$.

The conflict graph was introduced in [11,13] and exploitation of its connected components was the key idea in obtaining an optimal, efficient solution to the specialized phylogenetic networks problem considered there. That approach was studied in more detail in [12] and extended in [7]. The connected components are also exploited in [1] and in [9]. This paper further develops and exploits the importance of the connected components of those graphs, and further establishes that structural properties of phylogenetic networks can be efficiently discovered through structural properties of the conflict and incompatibility graphs.

3. Main results

Theorem 3.1. *For a set of sequences* M, $m(M) \geqslant cc(M)$.

Theorem 3.2. *For a set of sequences* M *and an ancestral sequence* R, $m_R(M) \geqslant cc_R(M)$.

We will sometimes refer to these two bounds collectively as "*cc*" bounds.

Theorem 3.2 was first announced at the CSB 2003 conference [10]. Independently, Bafna and Bansal [1] also obtained these results. Theorem 3.2 is a generalization of [27, Proposition 1], established by Song and Hein. That proposition establishes (in other terminology) that if $cc(M)$ is greater than one, then $m(M)$ is greater than one.

Note that Theorem 3.2 follows from Theorem 3.1. In particular, if Theorem 3.1 holds, then $m_R(M) \geqslant m(M + R) \geqslant cc(M + R) = cc_R(M)$, and Theorem 3.2 also holds. However, there is no fixed relationship between $cc(M)$ and $cc_R(M)$. For example, for M shown below, $cc(M)$ is 2, but if we require that the root sequence R be 1111, then $cc_R(M)$ is 1, and $cc(M) > cc_R(M)$ in this case.

12	34
00	00
01	00
10	00
11	00
00	01
00	10

Conversely, for M shown below, $cc(M)$ is 0, but if we require that the root sequence R be 00, then $cc_R(M)$ is 1, and $cc_R(M) > cc(M)$ in this case.

12
01
10

Now since $m(M) \leqslant m_R(M)$, even when an ancestral sequence R is known, $cc(M)$ is a correct lower bound for $m_R(M)$. So when R is specified, both $cc(M)$ and $cc_R(M)$ should be computed, and the maximum of the two used as a lower bound on $m_R(M)$.

4. Proof of the main result

We first establish a theorem which will be used to prove Theorem 3.1 and several extensions of it to other biological models. We expect that this theorem will have additional applications in the study of phylogenetic networks.

Note that a pair of incompatible columns in M may become compatible after the removal of one of the sequences from M. In that case, we say that the sequence removal "breaks" an incompatibility.

Theorem 4.1. *The removal of a single sequence from* M *can break incompatibilities in at most one connected component of* $G(M)$.

Proof. For contradiction, suppose there is a matrix M where $G(M)$ has at least two non-trivial connected components, and the removal of a single sequence A from M results in breaking the incompatibilities of column pairs in two different components of $G(M)$. In particular, suppose the removal of A breaks the incompatibilities of r, s and of t, u, where edges (r, s) and (t, u) are in different connected components of $G(M)$.

We use V, W, X, Y, respectively, to denote the values in row A of matrix M, in columns r, s, t, u. Because columns r and s are incompatible, there must also be rows denoted B, C, D in M which contain the three state-pairs for columns r

and s not in row A. We let a, b, c, d, e, f denote the entries in the columns t, u in rows B, C and D. Similarly, because columns t and u are incompatible, there must be rows denoted E, F, G which contain the three state-pairs for columns t and u not in row A. We let g, h, i, j, k, l denote the entries in columns r, s in rows E, F and G. Note that a row could be in the set $\{B, C, D\}$ and also in $\{E, F, G\}$. We will consider that possibility after analyzing the cases where the rows B–G are all distinct. So, until stated otherwise, we assume that the rows B–G are distinct. We let V', W', X', Y' denote $(V+1) \bmod 2$, $(W+1) \bmod 2$, $(X+1) \bmod 2$, $(Y+1) \bmod 2$, respectively. So, assuming rows A–G are distinct, they can be pictured without loss of generality as follows:

	r	s	t	u
A	V	W	X	Y
B	V'	W'	a	b
C	V'	W	c	d
D	V	W'	e	f
E	g	h	X'	Y'
F	i	j	X'	Y
G	k	l	X	Y'

Now, because the removal of row A is assumed to break the r, s incompatibility, the state-pair V, W that appears in row A in columns r, s cannot appear in any other row in the columns r, s. Similarly, the state-pair X, Y that appears in row A in columns t, u cannot appear in any other row in columns t, u.

We now consider two cases: either element b is equal to Y' or it is equal to Y.

Case 1: Entry b is equal to Y'. Then columns r, u contain V, Y and V', Y' (in rows A and B), so to avoid incompatibility between columns r and u (which are assumed to be in different components of $G(M)$), either V', Y or V, Y' must be avoided in those columns. So either (1.1) $d = Y'$ and $i = V$, to avoid V', Y or (1.2) $f = Y$ and $g = k = V'$, to avoid V, Y'.

In case (1.1), j must be W' to avoid an additional V, W in r, s. Then we have

	r	s	t	u
A	V	W	X	Y
B	V'	W'	a	Y'
C	V'	W	c	Y'
D	V	W'	e	f
E	g	h	X'	Y'
F	V	W'	X'	Y
G	k	l	X	Y'

and columns s and u are incompatible, a contradiction.

In case (1.2), e must be set to X' to avoid X, Y in t, u. Then we have

	r	s	t	u
A	V	W	X	Y
B	V'	W'	a	Y'
C	V'	W	c	d
D	V	W'	X'	Y
E	V'	h	X'	Y'
F	i	j	X'	Y
G	V'	l	X	Y'

and r and t are incompatible, a contradiction. So it is not possible for b to equal Y' in the assumed M.

Case 2: Entry b is equal to Y. Then entry a must be equal to X', to avoid an additional X, Y in columns t, u.

	r	s	t	u
A	V	W	X	Y
B	V'	W'	X'	Y
C	V'	W	c	d
D	V	W'	e	f
E	g	h	X'	Y'
F	i	j	X'	Y
G	k	l	X	Y'

Now columns r, u contain V, Y and V', Y (in rows A and B), so to avoid conflict either V', Y' or V, Y' must be avoided in those columns. So either (2.1) $d = Y$ and $g = k = V$, to avoid V', Y' or (2.2) $f = Y$ and $g = k = V'$ to avoid V, Y'. In case (2.1) we have

	r	s	t	u
A	V	W	X	Y
B	V'	W'	X'	Y
C	V'	W	c	Y
D	V	W'	e	f
E	V	h	X'	Y'
F	i	j	X'	Y
G	V	l	X	Y'

But then l must be W' to avoid an additional V, W in columns r, s, and c must be X' to avoid an additional X, Y in t, u. So we have

	r	s	t	u
A	V	W	X	Y
B	V'	W'	X'	Y
C	V'	W	X'	Y
D	V	W'	e	f
E	V	h	X'	Y'
F	i	j	X'	Y
G	V	W'	X	Y'

And now columns s and t are incompatible, a contradiction.

In case (2.2) we have

	r	s	t	u
A	V	W	X	Y
B	V'	W'	X'	Y
C	V'	W	c	d
D	V	W'	e	Y
E	V'	h	X'	Y'
F	i	j	X'	Y
G	V'	l	X	Y'

Then e must be X' to avoid an additional X, Y in columns t, u

	r	s	t	u
A	V	W	X	Y
B	V'	W'	X'	Y
C	V'	W	c	d
D	V	W'	X'	Y
E	V'	h	X'	Y'
F	i	j	X'	Y
G	V'	l	X	Y'

And now columns r, t are incompatible, a contradiction. So b cannot be equal to Y, and the lemma is proved when the rows B–G are distinct.

We now consider the situation when rows B–G are not all distinct. For example, consider the case that rows B and E are the same row. In that situation, for a given set of values V, W, X, Y, entries a and b have the fixed values of X', Y', and entries g and h have the fixed values V', W', as shown below.

	r	s	t	u
A	V	W	X	Y
B,E	$g = V'$	$h = W'$	$a = X'$	$b = Y'$
C	V'	W	c	d
D	V	W'	e	f
F	i	j	X'	Y
G	k	l	X	Y'

Note that entries a and b are still in a row where the r, s entries are V', W' and entries g and h are still in a row where the t, u entries are X', Y', and that is all that was assumed about entries a, b, g and h in the earlier argument, when rows B–G were distinct. So if we now apply the earlier argument to this configuration, we see that every step of the argument either still holds, or is a step where a (now) fixed value is assumed or deduced to have a different value in the earlier argument. That simply allows the argument to stop sooner, by leading to a contradiction sooner. For example, case 2 (where entry b is assumed to be equal to Y) can be avoided because the value of b is now fixed at Y'. In general, no matter how rows $\{B, C, D\}$ and $\{E, F, G\}$ intersect, we can use the earlier argument, but truncate any part of the argument where it assumes or deduces a value for some entry a–l that is different from its now fixed value. The key again is that each entry a–f is still in a row that has the same r, s state-pair as was shown when rows B–G were assumed to be distinct, and similarly each entry g–l is still in a row that has the same t, u state pair as shown earlier. Therefore, Theorem 4.1 is proven. □

Note that neither the statement of Theorem 4.1 nor its proof mention recombination, but only the incompatibility of columns. Further, the order that the column appear is never used. These facts will be important in applying Theorem 4.1 to a wide variety of biological models. We can now prove Theorem 3.1.

Proof of Theorem 3.1. Suppose the claim in the theorem is untrue. Pick a "minimal" counterexample M: first, among all counterexamples, select those for which $m(M)$ is minimal. Among those, pick one for which the number of columns of matrix M is minimal. Since M is a counterexample to the theorem, the incompatibility graph $G(M)$ has $k \geqslant m(M)+1$ non-trivial connected components.

Now consider an ancestral sequence R which solves the root-unknown phylogenetic network problem for M, and let N be the resulting phylogenetic network. Because N is a directed acyclic graph, if it has any recombination nodes it has one, v, which does not lead to another recombination node. So the subgraph of N rooted at v contains no cycles (in the underlying undirected graph of N), and must be a tree, denoted T_v. With multiple cross-over recombination, the state of any site i in a recombinant sequence is taken from one of the two i-states in the recombining sequences, so if the state is the same in both parental sequences, it will be the same in the recombinant sequence. It then follows that when a site can label only one edge of N, any two incompatible sites must occur together on some cycle in N (after directions are deleted). This is easy to prove, and a formal proof appears in [13]. Hence, no site assigned to an edge in T_v can be incompatible with any other site, and hence no site assigned to an edge in T_v can be in a non-trivial connected component of $G(M)$. Therefore, any site i assigned to an edge in T_v could be removed from M and from N without changing $G(M)$ or the number of recombinations in the resulting N. Hence by the minimality of M, such a site i cannot exist, and v must lead directly to a leaf, and the edge out of v cannot contain a site. (Note that this conclusion would not hold if a "recombination" allowed the state of a site i to be set arbitrarily, independent of the parental sequences.)

Let A denote the sequence labeling node v, and hence labeling the leaf it points to. Therefore, A is a sequence in M. If we delete A from M, and we delete leaf A and node v from N, we have a set of sequences $M - A$ which can be derived on a phylogenetic network using at most $m(M) - 1$ recombination nodes. So by the minimality of the counterexample, the incompatibility graph $G(M - A)$ for $M - A$ has at most $m(M) - 1$ non-trivial connected components. But graph $G(M)$ had at least $m(M) + 1$ non-trivial connected components, so the removal of sequence A from M must have resulted in the removal of edges in at least two distinct non-trivial connected components of $G(M)$. That is, the removal of A must have broken incompatibilities represented in at least two distinct connected components of $G(M)$. But Theorem 4.1 makes this impossible. □

It is worth noting that the bound $cc_R(M)$ is exact when M can be derived on a phylogenetic network N with ancestral sequence R, where all the "recombinations cycles" in N are disjoint [13,7]. Similarly, the bound $cc(M)$ is exact when M can be derived on a phylogenetic network N where the ancestral sequence of N is not pre-selected, and all the recombinations cycles in N are disjoint [7].

5. Extension to other biological models

The main biological motivation for Theorems 3.2 and 3.1 comes from single-crossover recombination and gene-conversion in populations (individuals from the same species). Gene conversion [3,31] can be viewed as a multiple-crossover operation with exactly two crossovers. At the species level, "hybridization" causes the movement of genetic material between two species, and mathematically (but not biologically) looks like a multiple-crossover recombination. Hence if we interpret $m(M)$ and $m_R(M)$ to be the minimum number of hybridization events needed in a phylogenetic network (with unknown root, and with root R, respectively), then the theorems apply to phylogenetic networks with hybridization as well. The theorems also apply to hybridizations where the linear order of the columns (characters) is not fixed [21,19].

Theorems 3.2 and 3.1 can also be extended to apply to biological phenomena that do not initially look like recombination or hybridization. For example, in the Dollo model of complex-character evolution [5], a character is assumed to be created at most once in an evolutionary history, but can subsequently be lost at any point in the history. Specifically, if 0 is the ancestral state of a character i, then it can change from 0 to 1 at most once, but a change from 1 back to 0 is permitted in any sequence where the state of i is 1. If we define $m(M)$ and $m_R(M)$ as the minimum number of such "back mutations" needed in a phylogenetic tree (no cycles in the underlying graph) with unknown and known ancestral sequence, respectively, then Theorems 3.1 and 3.2 continue to hold, and provide lower bounds on the minimum number of back mutations needed to derive the sequences. A simple way to see this is to note that the state-change caused by a back mutation at a site i in a sequence S can be created with one (two-crossover) recombination, with breakpoints before and after site i, between the ancestral sequence and sequence S. With similar ideas, we can model other biological events with a number of (multiple-crossover) recombinations equal to the number of those events, and hence Theorems 3.2 and 3.1 establish lower bounds on the needed number of such events.

6. Comparison with other lower bound methods

The lower bounds established in Theorems 3.2 and 3.1 are primarily of interest because they further develop the view that structural properties of phylogenetic networks can be efficiently discovered from efficiently observed structural properties of the conflict and incompatibility graphs. The theorems are also of practical interest because they provide simple proofs that the algorithms in [13,7] use the minimum possible number of recombinations when they produce a phylogenetic network. However, it is of interest to see how well the lower bounds perform, compared to other known lower bounds, when used alone and when embedded in more complex lower bounding approaches. Here we discuss such comparisons to previously published methods.[1]

Several prior lower bound methods have been published [18,27,20], along with one meta-method that shows how to obtain a single composite bound from disparate lower bounds computed over different subsets of sites in M [20]. Two of the lower bound methods run in polynomial time, while two have exponential running times. The running time needed to compute the composite bound depends on which subsets are used, and will take exponential time unless the subsets are suitably restricted. We will consider two ways polynomial-time restrictions of the general composite methods. Some of the lower bound methods depend on a fixed linear order of the sites, and hence do not extend easily to biological phenomena other than recombination, while some of the methods do not depend on order and hence extend to a wide range of biological phenomena.

6.1. The Hudson–Kaplan bound

The first efficient, and most widely used, lower bound method was proposed in [18] and (in other terminology) is the following: arrange the nodes of $G(M)$ on the real line, in the order that the sites appear in the underlying chromosome. Then compute the minimum number of points on the real line needed so that each edge in the embedded graph $G(M)$ crosses at least one of the selected points. Call this number $HK(M)$. To see that $HK(M)$ is a lower bound on $m(M)$, note that for any phylogenetic network N for M, and for any pair of incompatible sites i, j, there must be at least one recombination in N that occurs at a point on the chromosome between i and j. $HK(M)$ can be computed by a greedy left-to-right sweep in time that is linear in the number of edges of $G(M)$. $HK(M)/2$ is also a lower bound on the number of gene-conversions needed, but the *HK* bound cannot be used to bound the number of hybridizations, where the linear order of the sites is not fixed, and it cannot be used in the Dollo model.

For any k, it is easy to construct examples where $cc(M) > HK(M) + k$, and conversely, examples where $HK(M) > cc(M) + k$, so these lower bound are incomparable. But since both bounds can be efficiently computed, it is worth using both in practice. We will see below that a polynomial-time variant of the composite method, using cc bounds, is guaranteed to give a lower bound that is always larger or equal to $HK(M)$.

6.2. The haplotype and history bounds

The second efficiently computed lower bound, called the "haplotype bound", was developed in [20]: first, remove every column from M that is compatible with every other column; the haplotype bound, $h(M)$, is the number of distinct rows of M, minus the number of distinct columns of M, minus one. It is easy to construct examples where $cc(M) > h(M)$, and examples where $h(M) > cc(M)$, so those bounds are incomparable. Simulations using sequences generated by the program *MS* [17] show that $h(M)$ by itself is a very poor bound, often a negative number, and never observed (in these simulations) to be larger than $HK(M)$. However, when used inside the composite method (explained below), haplotype bounds lead to very good lower bounds.

The other two published lower bounds take exponential time to compute, but seem to be better than $HK(M)$, $cc(M)$ and $h(M)$ on the small problem instances where the bounds can be computed [20,27]. One of the exponential-time lower bounds [20], called the "history lower bound", is computed through multiple executions of an algorithm where in each execution, rows (sequences) and columns of M are removed until M becomes empty. Some of the row (sequence) removals break incompatibilities, relative to the current M, and all incompatibilities are broken at or before the step in which M becomes empty. Each row (sequence) removal either removes a sequence that is a duplicate of another sequence in the current M, or removes a sequence that is distinct. Clearly, no removal of the first kind will break an incompatibility, and a row removal of the second kind might not break an incompatibility. The history lower bound established in [20] is equal to the number of row removals of the second kind, in the execution of the algorithm that minimizes the number of row removals of the second kind. Hence, by Theorem 4.1, we have

Theorem 6.1. *The history lower bound is always larger or equal to* $cc(M)$.

[1] Two new, methods [29,2], were published after the submission of this paper, and are not compared.

	0.5	1	3	5	10	20	50	100
CC > HK	0,0,0	0,0,0	0,0,0	0,0,0	1,50,1	1,100,1	0,0,0	1,33,1
CCG > HK	0,0,0	0,0,0	0,0,0	0,0,0	2,75,1	2,100,1	3,100,1	4,71,1
CC > H	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,100,1	0,0,0	0,0,0
CCG > H	0,0,0	0,0,0	0,0,0	0,0,0	1,100,1	2,100,1	1,100,1	0,0,0
H > CC	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	3,83,1	4,94,1
H > CCG	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,50,1	2,37,1
CHG > all	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CC > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CCG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
H > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CHG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
Rh > CC	0,0,0	0,0,0	0,0,0	0,0,0	1,100,1	1,100,1	4,112,1	5,80,1
Rh > CCG	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	2,50,1	3,33,1
Rh > H	0,0,0	0,0,0	0,0,0	0,0,0	1,100,1	2,100,1	2,75,1	1,25,1
Rh > CHG	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,50,1	1,25,1

Fig. 2. $n = 10$, $m = 10$.

	0.5	1	3	5	10	20	50	100
CC > HK	0,0,0	0,0,0	0,0,0	0,0,0	1,100,1	0,0,0	1,25,1	0,0,0
CCG > HK	0,0,0	0,0,0	0,0,0	1,100,1	4,87,1	5,60,1	5,52,1	7,31,1
CC > H	0,0,0	0,0,0	0,0,0	0,0,0	1,100,1	0,0,0	0,0,0	0,0,0
CCG > H	0,0,0	0,0,0	0,0,0	0,0,0	1,100,1	3,50,1	3,58,1	2,27,1
H > CC	0,0,0	0,0,0	0,0,0	1,100,1	3,83,1	2,75,1	2,33,1	7,38,2
H > CCG	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,14,1	3,33,1
CHG > all	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,75,3
CC > Rh	0,0,0	0,0,0	0,0,0	0,0,0	1,100,1	0,0,0	0,0,0	0,0,0
CCG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	1,100,1	0,0,0	0,0,0	0,0,0
H > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CHG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	1,100,1	0,0,0	0,0,0	0,0,0
Rh > CC	0,0,0	0,0,0	0,0,0	1,100,1	3,83,1	5,60,1	5,60,1	9,48,2
Rh > CCG	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	2,32,1	5,37,2
Rh > H	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	3,50,1	4,56,1	5,28,1
Rh > CHG	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,50,1	3,22,1

Fig. 3. $n = 10$, $m = 20$.

Despite Theorem 6.1, $cc(M)$ is of value because the computation of the history bound requires trying all possible executions. That takes time that grows superexponentially with the size of M, making it practical only for small problem instances. The history lower bound has been implemented as an option in the computer program RECMIN

	0.5	1	3	5	10	20	50	100
CC > HK	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	4,18,1	1,12,1
CCG > HK	0,0,0	0,0,0	0,0,0	2,100,1	4,58,1	6,38,1	9,38,2	6,17,1
CC > H	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,12,1	1,12,1
CCG > H	0,0,0	0,0,0	0,0,0	1,100,1	4,58,1	3,46,2	4,21,1	3,10,1
H > CC	0,0,0	0,0,0	0,0,0	1,100,1	0,0,0	4,22,1	6,30,2	4,20,1
H > CCG	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,10,1
CHG > all	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CC > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,12,1	0,0,0
CCG > Rh	0,0,0	0,0,0	0,0,0	1,100,1	0,0,0	0,0,0	1,12,1	0,0,0
H > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CHG > Rh	0,0,0	0,0,0	0,0,0	1,100,1	0,0,0	0,0,0	1,12,1	0,0,0
Rh > CC	0,0,0	0,0,0	0,0,0	1,100,1	4,58,1	6,38,1	8,40,2	6,29,2
Rh > CCG	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	4,12,1	4,20,1
Rh > H	0,0,0	0,0,0	0,0,0	0,0,0	4,58,1	3,46,2	7,17,1	6,17,1
Rh > CHG	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	4,12,1	3,24,2

Fig. 4. $n = 10$, $m = 50$.

	0.5	1	3	5	10	20	50	100
CC > HK	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,14,1	3,13,1
CCG > HK	0,0,0	0,0,0	0,0,0	2,75,1	2,27,1	6,31,1	9,25,2	10,29,3
CC > H	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,14,1	1,8,1
CCG > H	0,0,0	0,0,0	0,0,0	2,42,1	1,33,1	4,37,1	8,17,1	10,17,2
H > CC	0,0,0	0,0,0	0,0,0	1,50,1	1,20,1	2,17,1	4,20,1	5,16,2
H > CCG	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CHG > all	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CC > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CCG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,17,1	1,10,1	1,9,1
H > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CHG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,17,1	1,10,1	1,9,1
Rh > CC	0,0,0	0,0,0	0,0,0	2,75,1	2,27,1	5,38,1	9,28,2	9,30,3
Rh > CCG	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,17,1	3,13,1	3,9,1
Rh > H	0,0,0	0,0,0	0,0,0	2,42,1	1,33,1	4,37,1	7,25,2	9,21,2
Rh > CHG	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,17,1	3,13,1	3,9,1

Fig. 5. $n = 10$, $m = 70$.

[20], but the program can compute the history bound only for problem instances much smaller than those of current interest, and when $m(M)$ is small. Recently, it was shown that the history bound can be computed in exponential time [2].

	0.5	1	3	5	10	20	50	100
CC > HK	0,0,0	0,0,0	0,0,0	1,33,1	2,37,1	2,14,1	2,12,1	3,8,1
CCG > HK	0,0,0	0,0,0	1,100,1	3,61,1	6,49,1	6,32,1	9,22,2	10,15,2
CC > H	0,0,0	0,0,0	0,0,0	1,33,1	2,37,1	1,11,1	0,0,0	1,10,1
CCG > H	0,0,0	0,0,0	1,100,1	2,67,1	5,48,1	6,29,1	5,15,1	6,8,1
H > CC	0,0,0	0,0,0	0,0,0	1,50,1	1,50,1	0,0,0	7,13,1	8,12,2
H > CCG	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,7,1
CHG > all	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,20,2	2,23,3
CC > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CCG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	3,56,1	1,20,1	4,10,1	0,0,0
H > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CHG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	3,56,1	1,20,1	4,10,1	0,0,0
Rh > CC	0,0,0	0,0,0	1,100,1	2,75,1	1,50,1	5,28,1	8,18,2	9,22,3
Rh > CCG	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	2,10,1	6,10,1
Rh > H	0,0,0	0,0,0	1,100,1	2,67,1	2,37,1	5,31,1	4,13,1	8,13,2
Rh > CHG	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,11,1	4,9,1

Fig. 6. $n = 10$, $m = 100$.

	0.5	1	3	5	10	20	50	100
CC > HK	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,100,1
CCG > HK	0,0,0	0,0,0	0,0,0	1,100,1	2,100,1	1,100,1	8,66,1	9,52,1
CC > H	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CCG > H	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,50,1
H > CC	0,0,0	0,0,0	0,0,0	1,100,1	2,150,1	3,58,1	9,81,1	8,100,3
H > CCG	0,0,0	0,0,0	0,0,0	0,0,0	1,50,1	2,37,1	4,37,1	7,49,2
CHG > all	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,80,4
CC > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CCG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
H > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CHG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
Rh > CC	0,0,0	0,0,0	0,0,0	1,100,1	2,250,2	3,58,1	9,92,2	9,99,3
Rh > CCG	0,0,0	0,0,0	0,0,0	0,0,0	2,75,1	2,37,1	5,47,1	7,54,2
Rh > H	0,0,0	0,0,0	0,0,0	0,0,0	2,42,1	0,0,0	2,33,1	3,26,1
Rh > CHG	0,0,0	0,0,0	0,0,0	0,0,0	2,42,1	0,0,0	2,33,1	1,14,1

Fig. 7. $n = 20$, $m = 10$.

6.3. *The composite method*

Myers and Griffiths [20] introduced a meta-method, we call the "composite method", to combine lower bounds that have been computed (by any method) over a family $\mathscr{F}$ of subsets of sites. The composite method is a generalization of the method that computes $HK(M)$. When there is no restriction on the subsets in $\mathscr{F}$, the method can consider a

	0.5	1	3	5	10	20	50	100
CC > HK	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,50,1	2,42,1	2,37,1
CCG > HK	0,0,0	0,0,0	0,0,0	3,83,1	4,75,1	6,56,1	8,66,2	9,43,2
CC > H	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,50,1	0,0,0	0,0,0
CCG > H	0,0,0	0,0,0	0,0,0	2,75,1	1,50,1	3,57,1	1,12,1	0,0,0
H > CC	0,0,0	0,0,0	0,0,0	1,100,1	4,67,1	5,56,2	9,71,2	10,75,4
H > CCG	0,0,0	0,0,0	0,0,0	0,0,0	1,20,1	2,58,1	5,37,2	10,37,2
CHG > all	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CC > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CCG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
H > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CHG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
Rh > CC	0,0,0	0,0,0	0,0,0	5,70,1	6,95,2	7,62,2	9,96,3	10,99,5
Rh > CCG	0,0,0	0,0,0	0,0,0	2,50,1	5,39,1	3,47,1	8,42,2	10,57,3
Rh > H	0,0,0	0,0,0	0,0,0	4,62,1	4,60,1	4,49,1	7,20,1	9,15,1
Rh > CHG	0,0,0	0,0,0	0,0,0	2,50,1	4,44,1	1,25,1	7,18,1	9,15,1

Fig. 8. $n = 20$, $m = 20$.

	0.5	1	3	5	10	20	50	100
CC > HK	0,0,0	0,0,0	4,87,1	2,42,1	3,53,1	4,23,1	2,16,1	5,10,1
CCG > HK	0,0,0	0,0,0	6,75,1	3,44,1	9,53,1	9,34,1	9,34,2	10,33,3
CC > H	0,0,0	0,0,0	3,78,1	2,42,1	1,25,1	2,25,1	0,0,0	0,0,0
CCG > H	0,0,0	0,0,0	4,71,1	3,44,1	3,42,1	5,31,1	4,18,1	1,10,1
H > CC	0,0,0	0,0,0	1,50,1	0,0,0	5,57,2	4,28,1	8,31,2	10,51,5
H > CCG	0,0,0	0,0,0	0,0,0	0,0,0	1,60,3	1,12,1	4,14,1	8,25,3
CHG > all	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,57,4	3,135,14
CC > Rh	0,0,0	0,0,0	1,100,1	2,42,1	1,25,1	1,33,1	0,0,0	0,0,0
CCG > Rh	0,0,0	0,0,0	1,100,1	2,42,1	2,50,1	1,33,1	0,0,0	0,0,0
H > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CHG > Rh	0,0,0	0,0,0	1,100,1	2,42,1	2,50,1	1,33,1	0,0,0	1,6,1
Rh > CC	0,0,0	0,0,0	2,50,1	1,50,1	6,51,2	7,42,2	9,65,5	10,78,8
Rh > CCG	0,0,0	0,0,0	0,0,0	0,0,0	1,60,3	2,42,2	9,28,3	10,40,5
Rh > H	0,0,0	0,0,0	3,61,1	1,50,1	1,25,1	6,32,1	9,31,3	9,21,3
Rh > CHG	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	2,36,2	8,22,2	9,17,3

Fig. 9. $n = 20$, $m = 50$.

number of subsets that grows exponentially with the number of sites. However, the size of $\mathscr{F}$ is quadratic when the sites are embedded on the real line, ordered as they are in a chromosome, and each subset of sites in $\mathscr{F}$ is a contiguous interval of sites. Alternatively, the method is polynomial when a constant bound is placed on the size of the subsets.

	0.5	1	3	5	10	20	50	100
CC > HK	0,0,0	0,0,0	2,50,1	0,0,0	1,33,1	1,29,2	5,13,1	6,9,1
CCG > HK	0,0,0	2,100,1	3,61,1	5,51,1	7,46,2	8,27,1	10,36,3	10,33,4
CC > H	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,12,1	0,0,0	0,0,0
CCG > H	0,0,0	1,100,1	1,33,1	4,38,1	4,37,1	4,21,1	3,16,2	1,7,1
H > CC	0,0,0	1,100,1	1,33,1	1,100,1	4,40,1	4,45,2	9,42,3	10,40,5
H > CCG	0,0,0	0,0,0	0,0,0	0,0,0	1,25,1	2,25,2	4,29,2	8,15,3
CHG > all	0,0,0	0,0,0	0,0,0	0,0,0	1,50,2	0,0,0	3,124,13	7,366,47
CC > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CCG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
H > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CHG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
Rh > CC	0,0,0	2,100,1	2,33,1	7,56,1	8,60,2	9,33,2	10,64,6	10,67,9
Rh > CCG	0,0,0	0,0,0	0,0,0	4,29,1	5,32,1	5,20,2	10,29,3	10,33,6
Rh > H	0,0,0	1,100,1	1,33,1	6,49,1	7,43,1	5,27,2	8,27,3	10,20,3
Rh > CHG	0,0,0	0,0,0	0,0,0	4,29,1	3,39,1	3,16,1	8,17,2	9,16,3

Fig. 10. $n = 20$, $m = 70$.

	0.5	1	3	5	10	20	50	100
CC > HK	0,0,0	0,0,0	2,75,1	2,37,1	3,31,1	4,14,1	5,13,1	8,7,1
CCG > HK	0,0,0	0,0,0	2,75,1	5,37,1	7,33,1	9,46,3	10,38,4	10,35,5
CC > H	0,0,0	0,0,0	1,100,1	1,25,1	1,33,1	0,0,0	0,0,0	0,0,0
CCG > H	0,0,0	0,0,0	1,100,1	3,26,1	5,24,1	9,23,2	5,11,1	1,10,2
H > CC	0,0,0	0,0,0	0,0,0	2,25,1	2,20,1	5,21,2	10,31,3	10,33,5
H > CCG	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	2,31,4	7,7,1
CHG > all	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	4,173,19	8,321,52
CC > Rh	0,0,0	0,0,0	0,0,0	2,37,1	1,33,1	0,0,0	0,0,0	0,0,0
CCG > Rh	0,0,0	0,0,0	0,0,0	4,32,1	2,24,1	2,26,1	0,0,0	0,0,0
H > Rh	0,0,0	0,0,0	0,0,0	1,50,1	0,0,0	0,0,0	0,0,0	0,0,0
CHG > Rh	0,0,0	0,0,0	0,0,0	4,32,1	2,24,1	2,26,1	0,0,0	0,0,0
Rh > CC	0,0,0	0,0,0	0,0,0	3,33,1	4,38,2	8,44,3	10,59,7	10,67,11
Rh > CCG	0,0,0	0,0,0	0,0,0	1,50,1	2,14,1	5,12,1	10,23,3	10,31,6
Rh > H	0,0,0	0,0,0	1,100,1	1,50,1	4,26,1	8,28,2	10,23,3	10,26,5
Rh > CHG	0,0,0	0,0,0	0,0,0	1,50,1	2,14,1	5,12,1	9,13,2	10,19,4

Fig. 11. $n = 20$, $m = 100$.

For an interval *I*, let $M(I)$ denote the matrix *M* restricted to the sites in *I*. For each interval $I \in \mathscr{F}$, let $L(I)$ denote the highest lower bound computed (somehow) for $M(I)$. Each $L(I)$ is called an "interval" or "local" bound. Then, the *composite* lower bound is computed from these interval bounds by picking the smallest number of points on the real line, so that for any interval $I \in \mathscr{F}$, at least $L(I)$ of the selected points are contained in interval *I*. The selection of the points can be computed in linear time by a greedy left-to-right sweep of the intervals. In particular, whenever the

	0.5	1	3	5	10	20	50	100
CC > HK	0,0,0	0,0,0	0,0,0	0,0,0	1,100,1	1,100,1	0,0,0	3,108,1
CCG > HK	0,0,0	0,0,0	0,0,0	1,50,1	7,93,1	5,63,1	6,58,1	7,74,1
CC > H	0,0,0	0,0,0	0,0,0	0,0,0	1,100,1	0,0,0	0,0,0	0,0,0
CCG > H	0,0,0	0,0,0	0,0,0	0,0,0	1,100,1	0,0,0	0,0,0	0,0,0
H > CC	0,0,0	0,0,0	1,50,1	1,50,1	6,125,1	4,142,3	9,168,4	9,126,4
H > CCG	0,0,0	0,0,0	1,50,1	0,0,0	2,50,1	3,78,2	9,95,3	9,83,3
CHG > all	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CC > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CCG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
H > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,12,1	4,22,1
CHG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,12,1	4,22,1
Rh > CC	0,0,0	0,0,0	1,50,1	1,50,1	6,142,1	4,154,3	9,174,4	8,125,4
Rh > CCG	0,0,0	0,0,0	1,50,1	0,0,0	3,50,1	3,89,3	9,99,3	8,81,3
Rh > H	0,0,0	0,0,0	0,0,0	0,0,0	2,75,1	1,17,1	3,16,1	1,20,1
Rh > CHG	0,0,0	0,0,0	0,0,0	0,0,0	1,50,1	1,17,1	3,16,1	1,20,1

Fig. 12. $n = 50, m = 10$.

	0.5	1	3	5	10	20	50	100
CC > HK	0,0,0	0,0,0	0,0,0	1,33,1	1,50,1	1,33,1	3,48,1	1,20,1
CCG > HK	0,0,0	0,0,0	1,50,1	4,33,1	5,62,1	7,45,1	8,53,2	10,58,3
CC > H	0,0,0	0,0,0	0,0,0	0,0,0	1,50,1	0,0,0	0,0,0	0,0,0
CCG > H	0,0,0	0,0,0	0,0,0	2,33,1	2,75,1	1,50,1	0,0,0	0,0,0
H > CC	0,0,0	0,0,0	1,50,1	1,33,1	3,69,2	8,143,4	10,140,6	10,216,11
H > CCG	0,0,0	0,0,0	0,0,0	0,0,0	1,25,1	8,88,3	10,92,5	10,104,8
CHG > all	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CC > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CCG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
H > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,9,1	1,20,1	5,11,2
CHG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,9,1	1,20,1	5,11,2
Rh > CC	0,0,0	0,0,0	1,50,1	4,50,1	4,104,2	10,142,4	10,160,7	10,199,10
Rh > CCG	0,0,0	0,0,0	0,0,0	2,42,1	3,32,1	9,99,3	10,108,6	10,94,7
Rh > H	0,0,0	0,0,0	0,0,0	3,56,2	5,43,1	7,27,1	7,16,2	1,7,1
Rh > CHG	0,0,0	0,0,0	0,0,0	2,42,1	3,22,1	6,23,1	7,16,2	1,7,1

Fig. 13. $n = 50, m = 20$.

right endpoint of an interval I is reached, if $z < L(I)$ points in I have already been selected, then select an additional $L(I) - z$ points as far right in I as possible.

Let $CC(M)$ be the lower bound obtained by the composite method, choosing $\mathscr{F}$ to be the set of all possible intervals and using $cc(M(I))$ for the lower bound in each interval I. To compare $CC(M)$ to $HK(M)$, note that if i and $j > i$ are two incompatible sites and $I = [i, j]$, then $cc(M(I)) \geqslant 1$. Hence we have

	0.5	1	3	5	10	20	50	100
CC > HK	1,100,1	0,0,0	3,72,1	3,61,1	2,25,1	3,24,1	4,17,1	4,16,2
CCG > HK	1,100,1	1,100,1	5,70,1	7,57,1	8,33,1	10,44,2	10,34,3	10,35,4
CC > H	0,0,0	0,0,0	3,58,1	0,0,0	1,17,1	0,0,0	0,0,0	0,0,0
CCG > H	0,0,0	1,100,1	5,60,1	2,25,1	3,21,1	2,15,1	0,0,0	0,0,0
H > CC	0,0,0	0,0,0	0,0,0	6,39,1	7,39,2	9,43,2	10,83,7	10,122,16
H > CCG	0,0,0	0,0,0	0,0,0	3,28,1	3,29,2	3,18,1	10,45,5	10,76,13
CHG > all	0,0,0	0,0,0	0,0,0	0,0,0	1,50,3	2,73,6	0,0,0	1,142,17
CC > Rh	0,0,0	0,0,0	1,100,1	0,0,0	1,17,1	0,0,0	0,0,0	0,0,0
CCG > Rh	0,0,0	0,0,0	1,100,1	0,0,0	1,17,1	0,0,0	0,0,0	0,0,0
H > Rh	0,0,0	0,0,0	0,0,0	1,33,1	1,20,1	0,0,0	0,0,0	2,7,2
CHG > Rh	0,0,0	0,0,0	1,100,1	1,33,1	2,18,1	0,0,0	0,0,0	2,9,2
Rh > CC	0,0,0	1,100,1	3,40,1	7,50,2	7,71,3	10,74,4	10,112,10	10,139,18
Rh > CCG	0,0,0	0,0,0	0,0,0	3,47,2	5,45,3	9,34,3	10,69,8	10,89,15
Rh > H	0,0,0	1,100,1	4,50,1	3,47,2	6,32,2	9,31,2	10,17,3	6,16,4
Rh > CHG	0,0,0	0,0,0	0,0,0	2,40,2	5,25,2	7,31,3	10,17,3	6,16,4

Fig. 14. $n = 50$, $m = 50$.

	0.5	1	3	5	10	20	50	100
CC > HK	1,100,1	3,117,1	3,44,1	5,25,1	2,23,1	2,19,1	6,19,2	3,6,1
CCG > HK	2,100,1	3,117,1	6,61,1	8,50,1	9,42,2	10,41,3	10,44,4	10,34,6
CC > H	1,100,1	2,50,1	2,42,1	1,33,1	0,0,0	0,0,0	0,0,0	0,0,0
CCG > H	1,100,1	2,50,1	4,54,1	3,56,1	3,47,1	1,12,1	0,0,0	0,0,0
H > CC	1,100,1	0,0,0	2,67,1	4,33,1	7,50,3	10,48,4	10,79,9	10,111,19
H > CCG	0,0,0	0,0,0	1,33,1	1,12,1	7,20,1	5,19,2	10,38,6	10,61,13
CHG > all	0,0,0	0,0,0	0,0,0	0,0,0	1,75,3	4,186,16	0,0,0	1,106,19
CC > Rh	0,0,0	1,50,1	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CCG > Rh	0,0,0	1,50,1	1,25,1	1,33,1	0,0,0	0,0,0	0,0,0	0,0,0
H > Rh	0,0,0	0,0,0	0,0,0	0,0,0	1,20,1	0,0,0	0,0,0	0,0,0
CHG > Rh	0,0,0	1,50,1	1,25,1	1,33,1	1,40,2	0,0,0	0,0,0	0,0,0
Rh > CC	1,100,1	0,0,0	4,67,1	7,49,2	10,57,3	10,76,6	10,112,13	10,135,23
Rh > CCG	0,0,0	0,0,0	1,67,2	5,21,1	7,30,2	9,33,4	10,63,9	10,78,17
Rh > H	1,100,1	1,50,1	5,42,1	8,26,1	7,32,1	8,25,3	10,18,4	10,12,4
Rh > CHG	0,0,0	0,0,0	1,25,1	5,19,1	5,16,1	8,19,2	10,18,4	10,11,4

Fig. 15. $n = 50$, $m = 70$.

Theorem 6.2. *$CC(M) \geqslant HK(M)$ and can be arbitrarily larger. Further, $CC(M)$ can be computed in polynomial time.*

The haplotype lower bound $h(M(I))$ can also be used in the composite method for each interval I, and we call the resulting composite bound $H(M)$. It is again easy to establish [20] that $H(M) \geqslant HK(M)$, and it again holds that $H(M)$ can be computed in polynomial time.

	0.5	1	3	5	10	20	50	100
CC > HK	1,100,1	3,83,1	2,33,1	5,45,1	2,18,1	6,18,1	8,14,2	9,9,2
CCG > HK	1,100,1	4,87,1	5,32,1	8,52,2	8,27,2	10,39,3	10,39,5	10,39,7
CC > H	0,0,0	2,75,1	1,33,1	2,27,1	1,17,1	0,0,0	0,0,0	0,0,0
CCG > H	0,0,0	3,83,1	3,31,1	4,34,2	6,22,1	5,14,1	1,14,2	0,0,0
H > CC	0,0,0	0,0,0	1,33,1	3,33,2	3,22,2	8,29,2	9,56,8	10,82,16
H > CCG	0,0,0	0,0,0	0,0,0	0,0,0	1,11,1	2,25,2	9,23,4	10,43,10
CHG > all	0,0,0	0,0,0	0,0,0	1,43,3	0,0,0	0,0,0	3,240,34	2,152,32
CC > Rh	0,0,0	1,50,1	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CCG > Rh	0,0,0	2,75,1	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
H > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,6,2
CHG > Rh	0,0,0	2,75,1	0,0,0	0,0,0	0,0,0	0,0,0	1,3,1	1,6,2
Rh > CC	0,0,0	0,0,0	4,37,1	5,44,2	10,27,2	10,58,5	10,77,11	10,105,21
Rh > CCG	0,0,0	0,0,0	2,25,1	2,14,1	5,18,1	9,29,3	10,41,7	10,60,15
Rh > H	0,0,0	1,100,1	4,37,1	6,27,1	10,21,1	10,28,3	9,21,4	9,15,5
Rh > CHG	0,0,0	0,0,0	2,25,1	1,17,1	5,16,1	9,23,2	9,18,4	9,14,5

Fig. 16. $n = 50$, $m = 100$.

	0.5	1	3	5	10	20	50	100
CC > HK	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,50,1	1,100,1	1,33,1
CCG > HK	0,0,0	0,0,0	2,100,1	3,78,1	5,80,1	5,48,1	7,58,1	8,56,1
CC > H	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CCG > H	0,0,0	0,0,0	0,0,0	0,0,0	1,100,1	0,0,0	0,0,0	0,0,0
H > CC	0,0,0	0,0,0	2,100,1	3,89,1	4,75,1	5,102,3	8,220,5	9,224,6
H > CCG	0,0,0	0,0,0	0,0,0	1,25,1	0,0,0	4,57,2	8,127,4	9,121,5
CHG > all	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CC > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CCG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
H > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,8,1	2,17,1
CHG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,8,1	2,17,1
Rh > CC	0,0,0	0,0,0	2,100,1	4,79,1	5,80,1	5,128,3	9,197,4	9,216,5
Rh > CCG	0,0,0	0,0,0	0,0,0	2,37,1	0,0,0	5,67,3	9,115,3	9,116,4
Rh > H	0,0,0	0,0,0	0,0,0	1,50,1	1,100,1	3,32,1	1,33,1	1,12,1
Rh > CHG	0,0,0	0,0,0	0,0,0	1,50,1	0,0,0	3,32,1	1,33,1	1,12,1

Fig. 17. $n = 70$, $m = 10$.

6.4. Using galled-tree bounds to boost the lower bound

An additional idea that has been implemented and tested is to use the composite method over intervals, as in the computation of $CC(M)$, but also include lower bounds obtained by using the galled-tree program from [7]. That program determines in polynomial time whether a set of sequences M can be derived on a phylogenetic network, called

	0.5	1	3	5	10	20	50	100
CC > HK	0,0,0	0,0,0	1,50,1	0,0,0	4,67,1	5,48,1	2,27,1	4,22,1
CCG > HK	0,0,0	1,100,1	4,87,1	3,72,1	6,74,1	9,60,2	10,42,2	10,41,2
CC > H	0,0,0	0,0,0	0,0,0	0,0,0	1,100,1	0,0,0	0,0,0	0,0,0
CCG > H	0,0,0	1,100,1	2,75,1	0,0,0	2,67,1	1,20,1	0,0,0	0,0,0
H > CC	0,0,0	0,0,0	2,67,1	4,62,1	5,60,2	9,99,4	10,198,8	10,183,12
H > CCG	0,0,0	0,0,0	0,0,0	1,33,1	3,57,2	8,66,3	10,123,7	10,119,10
CHG > all	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CC > Rh	0,0,0	0,0,0	0,0,0	0,0,0	1,100,1	0,0,0	0,0,0	0,0,0
CCG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	1,100,1	0,0,0	0,0,0	0,0,0
H > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	2,22,1	6,19,2	8,19,3
CHG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	1,100,1	2,22,1	6,19,2	8,19,3
Rh > CC	0,0,0	1,100,1	4,79,1	4,71,1	7,74,2	9,113,4	10,174,7	10,147,9
Rh > CCG	0,0,0	0,0,0	1,25,1	2,27,1	5,67,2	9,66,3	10,103,6	10,91,8
Rh > H	0,0,0	1,100,1	3,58,1	1,20,1	6,27,1	4,32,2	2,11,1	0,0,0
Rh > CHG	0,0,0	0,0,0	1,25,1	1,20,1	5,25,1	4,25,1	2,11,1	0,0,0

Fig. 18. $n = 70$, $m = 20$.

	0.5	1	3	5	10	20	50	100
CC > HK	0,0,0	1,100,1	4,67,1	3,36,1	3,94,1	5,19,1	4,10,1	4,10,1
CCG > HK	0,0,0	3,100,1	6,65,1	8,56,1	9,54,1	10,37,2	10,37,3	10,47,4
CC > H	0,0,0	0,0,0	2,75,1	1,33,1	0,0,0	0,0,0	0,0,0	0,0,0
CCG > H	0,0,0	1,100,1	3,83,1	4,68,1	1,20,1	0,0,0	0,0,0	0,0,0
H > CC	0,0,0	2,75,1	1,25,1	4,37,1	6,52,2	10,48,3	10,134,13	10,210,20
H > CCG	0,0,0	1,50,1	0,0,0	2,27,1	3,38,2	6,30,2	10,79,10	10,120,17
CHG > all	0,0,0	0,0,0	0,0,0	0,0,0	1,75,3	2,71,5	0,0,0	0,0,0
CC > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CCG > Rh	0,0,0	0,0,0	0,0,0	1,17,1	0,0,0	0,0,0	0,0,0	0,0,0
H > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,8,1	2,6,1	7,11,3
CHG > Rh	0,0,0	0,0,0	0,0,0	1,17,1	0,0,0	1,8,1	2,6,1	7,11,3
Rh > CC	0,0,0	3,150,2	4,49,1	7,71,1	8,74,3	10,84,6	10,144,14	10,191,19
Rh > CCG	0,0,0	2,100,2	2,35,1	3,41,1	7,46,2	10,47,4	10,86,11	10,107,15
Rh > H	0,0,0	3,72,1	5,64,1	5,74,1	6,34,2	9,31,3	7,9,2	2,6,2
Rh > CHG	0,0,0	2,58,1	2,35,1	2,33,1	6,27,1	9,28,3	7,9,2	2,6,2

Fig. 19. $n = 70$, $m = 50$.

a *galled-tree*, where all the cycles are disjoint. It has been proven (using Theorem 3.1) that when M can be derived on such a network, the network produced by the program uses exactly $m(M)$ recombinations. Modifying the program, we can efficiently compute, for each site i, the *longest* interval I starting at site i with the property that the sequences in $M(I)$ can be derived on a galled-tree. Then, whenever an interval I is found where $M(I)$ can be derived using *exactly* one recombination, it follows that the interval consisting of I together with one extra site on its right has a lower bound, called $g(I)$, of two recombinations. Similarly, if from a position i say, the longest interval containing a

	0.5	1	3	5	10	20	50	100
CC > HK	0,0,0	1,50,1	1,33,1	6,34,1	5,38,2	3,16,1	5,20,2	7,13,2
CCG > HK	0,0,0	2,75,1	5,63,1	8,42,2	9,48,2	9,33,2	10,45,4	10,41,5
CC > H	0,0,0	1,50,1	1,33,1	1,17,1	2,22,1	0,0,0	0,0,0	0,0,0
CCG > H	0,0,0	2,75,1	4,54,1	5,23,1	5,18,1	1,11,1	0,0,0	0,0,0
H > CC	0,0,0	0,0,0	1,100,1	1,17,1	6,51,3	10,42,3	10,103,11	10,159,23
H > CCG	0,0,0	0,0,0	0,0,0	0,0,0	4,28,2	8,20,2	9,59,9	10,100,19
CHG > all	0,0,0	0,0,0	0,0,0	0,0,0	1,50,4	4,255,20	2,102,12	0,0,0
CC > Rh	0,0,0	0,0,0	0,0,0	0,0,0	1,20,1	0,0,0	0,0,0	0,0,0
CCG > Rh	0,0,0	0,0,0	0,0,0	1,14,1	1,20,1	0,0,0	0,0,0	0,0,0
H > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	3,10,2	4,7,3
CHG > Rh	0,0,0	0,0,0	0,0,0	1,14,1	1,20,1	0,0,0	3,10,2	4,7,3
Rh > CC	0,0,0	1,100,1	4,71,1	7,27,1	8,58,3	10,78,6	10,126,14	10,156,23
Rh > CCG	0,0,0	0,0,0	0,0,0	4,22,1	8,25,2	10,44,4	10,70,11	10,98,19
Rh > H	0,0,0	2,75,1	4,54,1	8,23,1	7,22,2	10,26,3	7,25,5	4,5,2
Rh > CHG	0,0,0	0,0,0	0,0,0	4,22,1	5,14,1	10,21,2	7,21,5	4,5,2

Fig. 20. $n = 70$, $m = 70$.

	0.5	1	3	5	10	20	50	100
CC > HK	1,100,1	3,83,1	3,33,1	5,36,1	9,19,1	7,26,2	6,17,2	6,11,2
CCG > HK	1,100,1	4,71,1	6,46,1	6,49,2	10,46,2	10,45,3	10,46,6	10,35,6
CC > H	1,100,1	0,0,0	2,37,1	1,20,1	0,0,0	0,0,0	0,0,0	0,0,0
CCG > H	1,100,1	1,33,1	6,41,1	3,21,1	5,26,2	2,13,1	0,0,0	0,0,0
H > CC	0,0,0	0,0,0	0,0,0	2,27,1	5,26,2	9,38,4	10,83,12	10,118,23
H > CCG	0,0,0	0,0,0	0,0,0	0,0,0	1,25,2	6,20,2	10,38,7	10,72,18
CHG > all	0,0,0	0,0,0	0,0,0	3,93,5	3,127,7	1,62,5	2,208,28	1,105,21
CC > Rh	1,100,1	0,0,0	1,25,1	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CCG > Rh	1,100,1	0,0,0	1,25,1	0,0,0	1,33,2	1,11,1	0,0,0	0,0,0
H > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,7,1	0,0,0	3,8,3
CHG > Rh	1,100,1	0,0,0	1,25,1	0,0,0	1,33,2	2,9,1	0,0,0	3,8,3
Rh > CC	0,0,0	1,33,1	5,49,1	7,34,2	10,40,2	9,55,5	10,110,17	10,127,25
Rh > CCG	0,0,0	0,0,0	3,21,1	6,21,1	7,25,2	8,28,3	10,58,12	10,79,20
Rh > H	0,0,0	1,33,1	6,49,1	7,28,1	9,27,2	6,20,3	9,16,5	7,11,5
Rh > CHG	0,0,0	0,0,0	3,21,1	4,18,1	6,17,1	5,17,2	9,15,4	7,10,4

Fig. 21. $n = 70$, $m = 100$.

perfect phylogeny extends to position j, and the longest interval starting at i that contains a galled-tree also ends at j, we can deduce that the interval $I = [i, j+1]$ has a lower bound, $g(I)$, of two. Formally, in any interval I where no such bound is obtained, $g(I)$ is set to 0. We can combine the bounds $g(I)$ with the $cc(M(I))$ bounds, taking the maximum in every interval. This can lead to improved interval lower bounds, and hence to an improved composite bound. The composite lower bound computed in this way is called $CCG(M)$. A natural extension is to set $L(I)$, for each interval I, to the *maximum* of $h(M(I))$, $cc(M(I))$ and $g(I)$. Let $CHG(M)$ denote the resulting lower bound. Note that $CHG(M)$ is always at least as large as the maximum of $CC(M)$, $H(M)$, and $CCG(M)$, but can be strictly larger than all of

	0.5	1	3	5	10	20	50	100
CC > HK	0,0,0	0,0,0	0,0,0	0,0,0	2,100,1	0,0,0	0,0,0	1,50,1
CCG > HK	0,0,0	1,100,1	0,0,0	0,0,0	5,100,1	6,78,1	8,56,1	10,48,1
CC > H	0,0,0	0,0,0	0,0,0	0,0,0	2,100,1	0,0,0	0,0,0	0,0,0
CCG > H	0,0,0	0,0,0	0,0,0	0,0,0	3,100,1	0,0,0	0,0,0	0,0,0
H > CC	0,0,0	1,100,1	0,0,0	1,50,1	2,200,2	6,158,3	10,174,5	10,206,7
H > CCG	0,0,0	0,0,0	0,0,0	1,50,1	2,50,1	5,55,2	9,109,5	10,116,5
CHG > all	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CC > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CCG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
H > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	2,22,3	6,22,2
CHG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	2,22,3	6,22,2
Rh > CC	0,0,0	1,100,1	0,0,0	1,50,1	3,167,2	7,140,2	10,171,5	10,168,5
Rh > CCG	0,0,0	0,0,0	0,0,0	1,50,1	2,50,1	6,51,2	9,104,5	10,89,4
Rh > H	0,0,0	0,0,0	0,0,0	0,0,0	3,100,1	1,33,1	2,16,1	0,0,0
Rh > CHG	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	1,33,1	2,16,1	0,0,0

Fig. 22. $n = 100$, $m = 10$.

	0.5	1	3	5	10	20	50	100
CC > HK	0,0,0	0,0,0	2,75,1	1,50,1	1,100,1	1,50,1	1,33,1	0,0,0
CCG > HK	0,0,0	0,0,0	4,87,1	2,75,1	5,83,1	10,58,2	9,41,2	10,39,2
CC > H	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CCG > H	0,0,0	0,0,0	1,100,1	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
H > CC	0,0,0	0,0,0	2,67,1	2,100,2	7,138,2	10,168,5	10,215,10	10,384,19
H > CCG	0,0,0	0,0,0	1,33,1	1,100,3	7,60,2	10,77,4	10,131,9	10,249,17
CHG > all	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CC > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CCG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
H > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	4,21,2	5,21,4	9,41,7
CHG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	4,21,2	5,21,4	9,41,7
Rh > CC	0,0,0	0,0,0	3,78,1	5,67,1	8,137,2	10,182,6	10,183,9	10,249,13
Rh > CCG	0,0,0	0,0,0	1,33,1	4,58,1	8,69,2	10,83,4	10,111,7	10,152,11
Rh > H	0,0,0	0,0,0	1,100,1	3,44,1	3,34,1	5,25,2	2,13,1	0,0,0
Rh > CHG	0,0,0	0,0,0	0,0,0	3,44,1	3,34,1	5,25,2	2,13,1	0,0,0

Fig. 23. $n = 100$, $m = 20$.

them. This provides another practical reason for using *cc* bounds, despite the fact that $CC(M)$ by itself may be a weak bound.

6.5. Program RECMIN

The program RECMIN [20] uses haplotype bounds in the composite method in a different way than described above. In RECMIN, the user specifies two parameters s and w. Then, RECMIN identifies every *subset* of s or fewer sites,

	0.5	1	3	5	10	20	50	100
CC > HK	0,0,0	2,75,1	3,42,1	5,41,1	6,38,1	1,17,1	5,18,1	6,11,1
CCG > HK	1,100,1	2,75,1	4,56,1	9,50,2	9,43,2	10,44,2	10,44,4	10,36,4
CC > H	0,0,0	0,0,0	1,25,1	0,0,0	1,33,1	0,0,0	0,0,0	0,0,0
CCG > H	0,0,0	0,0,0	1,25,1	1,17,1	3,23,1	0,0,0	0,0,0	0,0,0
H > CC	1,100,1	0,0,0	1,100,1	6,42,2	5,49,3	10,84,5	10,164,15	10,251,28
H > CCG	0,0,0	0,0,0	0,0,0	2,19,1	5,24,2	8,37,3	10,97,12	10,177,25
CHG > all	0,0,0	0,0,0	0,0,0	1,40,2	1,100,4	3,244,15	0,0,0	0,0,0
CC > Rh	0,0,0	0,0,0	0,0,0	0,0,0	1,33,1	0,0,0	0,0,0	0,0,0
CCG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	1,33,1	0,0,0	0,0,0	0,0,0
H > Rh	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	6,15,4	9,25,8
CHG > Rh	0,0,0	0,0,0	0,0,0	0,0,0	1,33,1	0,0,0	6,15,4	9,25,8
Rh > CC	1,100,1	1,50,1	6,46,1	10,62,2	8,72,4	10,122,7	10,150,14	10,191,21
Rh > CCG	0,0,0	1,50,1	5,36,1	9,36,2	8,46,3	10,57,5	10,88,11	10,128,18
Rh > H	0,0,0	1,50,1	6,34,1	9,32,2	7,38,3	8,29,3	4,10,2	1,14,4
Rh > CHG	0,0,0	1,50,1	5,36,1	9,28,1	6,35,2	8,24,3	4,10,2	1,14,4

Fig. 24. $n = 100$, $m = 50$.

	0.5	1	3	5	10	20	50	100
CC > HK	0,0,0	0,0,0	1,100,1	3,28,1	4,39,2	6,18,1	7,18,2	6,14,2
CCG > HK	1,100,1	0,0,0	7,50,1	7,37,1	9,41,2	10,36,3	10,50,5	10,45,6
CC > H	0,0,0	0,0,0	0,0,0	1,33,1	1,40,2	0,0,0	0,0,0	0,0,0
CCG > H	1,100,1	0,0,0	3,39,1	2,33,1	2,27,1	2,9,1	0,0,0	0,0,0
H > CC	0,0,0	0,0,0	3,56,1	3,58,2	9,32,2	8,49,5	10,138,18	10,214,34
H > CCG	0,0,0	0,0,0	1,25,1	1,25,1	5,18,1	6,25,3	10,78,13	10,135,28
CHG > all	0,0,0	0,0,0	0,0,0	1,60,3	1,100,7	5,253,22	0,0,0	0,0,0
CC > Rh	0,0,0	0,0,0	0,0,0	1,33,1	0,0,0	0,0,0	0,0,0	0,0,0
CCG > Rh	0,0,0	0,0,0	0,0,0	1,33,1	0,0,0	0,0,0	0,0,0	0,0,0
H > Rh	0,0,0	0,0,0	0,0,0	0,0,0	1,12,1	0,0,0	5,9,2	10,16,6
CHG > Rh	0,0,0	0,0,0	0,0,0	1,33,1	2,10,1	0,0,0	5,9,2	10,16,6
Rh > CC	1,100,1	0,0,0	7,62,2	6,68,2	10,42,3	10,66,6	10,131,17	10,171,27
Rh > CCG	0,0,0	0,0,0	5,27,1	5,38,2	8,24,2	10,36,4	10,73,13	10,103,22
Rh > H	1,100,1	0,0,0	5,50,1	6,33,1	4,46,3	7,32,4	2,8,2	0,0,0
Rh > CHG	0,0,0	0,0,0	4,27,1	5,29,1	4,29,2	7,21,3	2,8,2	0,0,0

Fig. 25. $n = 100$, $m = 70$.

provided that no pair of the s columns are more than w positions apart. For any such subset S, the program computes the haplotype lower bound on the submatrix of M restricted to the sites in S. If I is an interval whose endpoints are w or fewer positions apart, then $L(I)$ is taken as the largest haplotype lower bound computed over all the subsets of size s or less whose extreme left and right points coincide with the endpoints of interval I. Those bounds are then used in the composite method to obtain an overall lower bound, called R_h. When s and w are fixed, independent of the size of M, then R_h is computed in polynomial time. Overall, RECMIN is a very impressive, efficient program for computing lower bounds on $m(M)$.

	0.5	1	3	5	10	20	50	100
CC > HK	2,75,1	0,0,0	5,37,1	6,51,1	7,30,2	8,22,1	8,9,1	8,11,2
CCG > HK	3,83,1	4,62,1	8,45,1	10,45,1	10,39,2	10,53,4	10,41,6	10,45,8
CC > H	0,0,0	0,0,0	1,67,2	3,37,1	1,18,2	0,0,0	0,0,0	0,0,0
CCG > H	1,100,1	1,50,1	4,41,1	4,41,1	1,27,3	0,0,0	1,6,1	0,0,0
H > CC	0,0,0	3,100,1	2,27,1	4,32,1	8,25,2	10,46,4	10,91,14	10,165,33
H > CCG	0,0,0	1,50,1	0,0,0	1,33,1	4,14,1	5,26,3	9,51,11	10,99,27
CHG > all	0,0,0	0,0,0	0,0,0	0,0,0	3,127,11	6,304,28	2,206,31	0,0,0
CC > Rh	0,0,0	0,0,0	2,29,1	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
CCG > Rh	0,0,0	0,0,0	3,31,1	1,17,1	0,0,0	0,0,0	0,0,0	0,0,0
H > Rh	0,0,0	0,0,0	2,33,1	0,0,0	0,0,0	0,0,0	0,0,0	8,8,4
CHG > Rh	0,0,0	0,0,0	3,31,1	1,17,1	1,12,1	0,0,0	0,0,0	8,8,4
Rh > CC	1,100,1	4,87,1	6,30,1	6,30,1	10,43,3	10,80,6	10,116,18	10,150,30
Rh > CCG	0,0,0	1,50,1	3,18,1	2,27,1	9,27,2	10,38,4	10,64,13	10,88,24
Rh > H	1,100,1	1,50,1	6,32,1	4,45,1	8,26,2	9,27,3	10,15,4	2,2,1
Rh > CHG	0,0,0	0,0,0	3,18,1	1,20,1	8,20,2	8,22,2	9,14,4	2,2,1

Fig. 26. $n = 100$, $m = 100$.

6.6. New bounds

We mention here that there are two new, unpublished at the time of submission, lower bound methods that are not included in our comparisons. In [29], for each interval I in M, we use integer linear programming to find a subset S of columns of $M[I]$, giving the largest haplotype bound over all subsets of columns of M. The integer linear program has one variable for each row and column of M, and has only n^2 inequalities, where n is the number of rows of M. These bounds are then used in the composite method to obtain an overall lower bound. That bound is equivalent to what RECMIN would produce if the parameters s and w were set to their maximum values. Several heuristic ideas are implemented to speed up the computations without changing the resulting bound. A method for computing upper bounds is also discussed in [29], and the computations discussed there show that the lower and upper bounds obtained are often very close or matching.

A lower bound developed in [2] is similar to the one in [29], but instead of using integer programming to find the best local bound for each interval, they use a greedy algorithm to approximate the integer programming result in each interval. This leads to a faster computation, but somewhat lower composite bounds.

6.7. Empirical results

We have conduced extensive simulations to compare the lower bounds $HK(M)$, $CC(M)$, $H(M)$, $CCG(M)$, $CHG(M)$, and R_h (using the RECMIN default settings of $s = 6$ and $w = 12$). Programs computing the first five bounds are available at wwwcsif.cs.ucdavis.edu/∼gusfield.

The sequences used in the simulations were produced using the program MS [17], which is widely used for simulating the evolution of binary sequences under the neutral coalescent model with recombination. In the results we report here, we varied the number of sequences n (using $n = 10, 20, 50, 70, 100$), the number of polymorphic sites m (using $m = 10, 20, 50, 70, 100$), and the MS recombination parameter r (using $r = 0.5, 1, 3, 5, 10, 20, 50, 100$). The mutation rate is indirectly specified through the number of specified polymorphic sites.[2] The range of choices for the recombination parameter is consistent with ranges explored in other empirical studies of recombination (for example [20]), and is believed to be broad enough to simulate recombination rates occurring in human populations. For each

[2] MS also allows the mutation rate to be explicitly specified, but then the number of polymorphic sites is not fixed, making the results more difficult to organize and compare.

of the 400 combinations of n, m and r, we used MS to generate 10 sets of sequences and then we computed the six lower bounds listed above. While 10 sets is too small to allow precise quantitative conclusions about any single (n, m, r) combination, general qualitative conclusions about ranges of parameters can be made, and those conclusions are consistent with more extensive but more focussed simulations we have done. Moreover, we do not believe that conclusions about a specific (n, m, r) combination would lead to useful suggestions about which lower bounds to use in practice, since the recombination rate is generally unknown.

We show below one table for 25 combinations of n and m. Each column in a table specifies a choice of the recombination parameter r, and each row compares two specific lower bounds, or compares $CHG(M)$ to the best of $CCG(M)$ and $H(M)$ ($HK(M)$ and $CC(M)$ are omitted because $CCG(M)$ is guaranteed to be as good or better than $CC(M)$). Each cell contains three numbers. In a row labeled "$X > Y$", the entry u, v, w in a cell indicates that the lower bound X is *strictly* larger than the lower bound Y u times, that the average *percentage* increase in those cases (rounded to the nearest integer) is v%, and that the average *absolute* increase in those cases (rounded to the nearest integer) is w. For example, consider the entry 3, 78, 1 in the row labeled "$CC > H$" in column labeled 3 in the table for $n=20$, $m=50$. The entry indicates that out of the 10 data sets generated for $n=20$, $m=50$ and $r=3$, $CC(M)$ is *strictly* larger than $H(M)$ three times; that the average percentage increase in the lower bound in those three sets (rounded to the nearest integer) is 78%; and that the average absolute increase (rounded to the nearest integer) in those three cases is 1.

The tables below in Figs. 2–26 summarize our experiments. More extensive simulations were done, using a denser survey of choices for the parameter, and using more data sets for certain combinations of parameters, but the results shown here are consistent with those simulations and are sufficient for the general conclusions we are able to draw from all the data. Due to space limitations we present only the selected tables.

6.8. *Conclusions drawn from the data*

There are several qualitative conclusions that can be drawn from the simulations.

First, as expected the $HK(M)$ bound is consistently below the other bounds, and should be avoided.

Second, $CCG(M)$ is very frequently higher than $CC(M)$. Although that comparison is not shown directly in the tables, we can infer this conclusion by the fact that $CCG(M) > HK(M)$ more frequently than $CC(M) > HK(M)$, and also from the fact that $CCG(M) > H$ more frequently than $CC(M) > H$.

Third, $CHG(M)$ is often larger than $CCG(M)$ and $H(M)$. For example, see the table $n = 20, m = 70$. This conclusion is seen more clearly in the results for individual data sets (not shown) where $H(M) > CCG(M)$ and yet $CHG(M) > H(M)$. These results shows the utility of using all of computing multiple bounds in intervals, even bounds that are weak when used alone, to obtain the overall composite bound.

Fourth, $CC(M)$ and $CCG(M)$ can be larger than H and R_h, particularly in the mid-ranges of the recombination parameter r, when n is less than m. For example, see the tables for $n = 20, m = 50, 70$ and for $n = 50, m = 100$. Although, the overall frequencies with which $CC(M)$ and $CCG(M)$ are larger than H and R_h are not high, they are high in certain parameter ranges, and we view the overall frequencies as justification for the development and use of $CC(M)$ or $CCG(M)$ (in addition to H and R_h), since they are both efficiently computed lower bounds.

Fifth, when r and n are large, H is frequently larger than R_h, and this is observed for lower values of r when n is large compared to m. For example, see the tables for $n = 50, m = 10$; $n = 70, m = 20$; $n = 70, m = 50$; and $n = 100, m = 20, 50, 70$. The frequency and the pattern that the lower bound H is larger than R_h is perhaps the most surprising empirical observation.

Finally, as n, m and r increase, there is increasing variability about which bound is largest. This argues for the use of all the efficiently computed bounds. No single bound is universally superior, although if only one bound can be used, the best choice would be R_h.

Acknowledgments

Research partially supported by NSF Grants EIA-0220154 and CCR 0515378. We would like to thank Yun Song and Charles Langley for helpful conversations.

References

[1] V. Bafna, V. Bansal, The number of recombination events in a sample history: conflict graph and lower bounds, IEEE/ACM Trans. Comput. Biol. Bioinform. 1 (2004) 78–90.

[2] V. Bafna, V. Bansal, Improved recombination lower bounds for haplotype data, in: Proceedings of RECOMB, 2005.

[3] A. Berry, A. Barbadilla, Gene conversion is a major determinant of genetic diversity at the DNA level, in: R.S. Singh, C.B. Krimbas (Eds.), Evolutionary Genetics: From Molecules to Morphology, Cambridge University Press, Cambridge, 1999, pp. 102–123.

[4] A. Chakravarti, It's raining SNP's, hallelujah?, Natur. Genetics 19 (1998) 216–217.

[5] J. Felsenstein, Inferring Phylogenies, Sinauer, Sunderland, MA, 2004.

[6] D. Gusfield, Efficient algorithms for inferring evolutionary history, Networks 21 (1991) 19–28.

[7] 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.

[8] D. Gusfield, Optimal, efficient reconstruction of root-unknown phylogenetic networks with constrained and structured recombination, J. Comput. Systems Sci. 70 (2005) 381–398.

[9] D. Gusfield, V. Bansal, A fundamental decomposition theory for phylogenetic networks and incompatible characters, in: Proceedings of RECOMB, 2005.

[10] D. Gusfield, S. Eddhu, C. Langley, Powerpoint slides for: efficient reconstruction of phylogenetic networks (of SNPs) with constrained recombination ⟨http://wwwcsif.cs.ucdavis.edu/~gusfield/talks.html⟩.

[11] D. Gusfield, S. Eddhu, C. Langley, Efficient reconstruction of phylogenetic networks (of SNPs) with constrained recombination, in: Proceedings of second CSB Bioinformatics Conference, Los Alamitos, CA, 2003, IEEE Press, New York.

[12] D. Gusfield, S. Eddhu, C. Langley, The fine structure of galls in phylogenetic networks, INFORMS J. Comput. 16 (2004) 459–469 (special issue on Comput. Biol.).

[13] D. Gusfield, S. Eddhu, C. Langley, Optimal, efficient reconstruction of phylogenetic networks with constrained recombination, J. Bioinform. Comput. Biol. 2 (1) (2004) 173–213.

[14] J. Hein, Reconstructing evolution of sequences subject to recombination using parsimony, Math. Biosci. 98 (1990) 185–200.

[15] J. Hein, A heuristic method to reconstruct the history of sequences subject to recombination, J. Molecular Evoluation 36 (1993) 396–405.

[16] D. Hinds, L. Stuve, G. Nilsen, E. Halperin, E. Eskin, D. Gallinger, K. Frazer, D. Cox, Whole-genome patterns of common DNA variation in three human populations, Science 307 (2005) 1072–1079.

[17] R. Hudson, Generating samples under the Wright–Fisher neutral model of genetic variation, Bioinformatics 18 (2) (2002) 337–338.

[18] R. Hudson, N. Kaplan, Statistical properties of the number of recombination events in the history of a sample of DNA sequences, Genetics 111 (1985) 147–164.

[19] B. Moret, L. Nakhleh, T. Warnow, C.R. Linder, A. Tholse, A. Padolina, J. Sun, R. Timme, Phylogenetic networks: modeling, reconstructibility, and accuracy, IEEE/ACM Trans. Comput. Biol. Bioinform. (2004) pp. 13–23.

[20] S.R. Myers, R.C. Griffiths, Bounds on the minimum number of recombination events in a sample history, Genetics 163 (2003) 375–394.

[21] L. Nakhleh, D. Ringe, T. Warnow, Perfect phylogenetic networks: a new methodology for reconstructing the evolutionary history of natural languages, Language (Journal of the Linguistic Society of America) 81 (2) (2005) 382–420.

[22] L. Nakhleh, J. Sun, T. Warnow, C.R. Linder, B.M.E. Moret, A. Tholse, Towards the development of computational tools for evaluating phylogenetic network reconstruction methods, in: Proceedings of the eighth Pacific Symposium on Biocomputing (PSB 03), 2003, pp. 315–326.

[23] M. Norborg, S. Tavare, Linkage disequilibrium: what history has to tell us, Trends Genetics 18 (2002) 83–90.

[24] R. Page, Tangled trees: Cospeciation and coevolution, University of Chicago Press, Chicago, IL, 2002.

[25] D. Posada, K. Crandall, Intraspecific gene genealogies: trees grafting into networks, Trends Ecology Evolution 16 (2001) 37–45.

[26] M.H. Schierup, J. Hein, Consequences of recombination on traditional phylogenetic analysis, Genetics 156 (2000) 879–891.

[27] Y. Song, J. Hein, On the minimum number of recombination events in the evolutionary history of DNA sequences, J. Math. Biol. 48 (2003) 160–186.

[28] Y. Song, J. Hein, Parsimonious reconstruction of sequence evolution and haplotype blocks: finding the minimum number of recombination events, in: Proceedings of 2003 Workshop on Algorithms in Bioinformatics, Berlin, Germany, 2003, Lecture Notes in Computer Science, Springer, Berlin.

[29] Y. Song, Y. Wu, D. Gusfield, Efficient computation of close lower and upper bounds on the minimum number of needed recombinations in the evolution of biological sequences, Bioinformatics 21 (2005) i413–i422 Bioinformatics (Suppl. 1), Proceedings of ISMB, 2005.

[30] S. Tavare, Calibrating the clock: using stochastic processes to measure the rate of evolution, in: E. Lander, M. Waterman (Eds.), Calculating the Secretes of Life, National Academy Press, Washington, DC, 1995.

[31] J.D. Wall, Close look at gene conversion hot spots, Natur. Genetics 36 (2004) 114–115.

[32] L. Wang, K. Zhang, L. Zhang, Perfect phylogenetic networks with recombination, J. Comput. Biol. 8 (2001) 69–78.