# Constructing Minimal Ancestral Recombination Graphs

YUN S. SONG* and JOTUN HEIN

*Department of Statistics, University of Oxford,*
*1 South Parks Road, Oxford, OX1 3TG, UK*

**Abstract**

By viewing the ancestral recombination graph as defining a sequence of trees, we show how possible evolutionary histories consistent with given data can be constructed using the minimum number of recombination events. In contrast to previously-known methods, which only yield estimated lower bounds, our method of detecting recombination always gives the minimum number of recombination events if the right kind of rooted trees are used in our algorithm. A new lower bound can be defined if rooted trees with less constraints are used. As well as studying how often it actually is equal to the minimum, we test how this new lower bound performs in comparison to some other lower bounds. Our study indicates that the new lower bound is an improvement on earlier bounds. Also, using simulated data, we investigate how well our method can recover the actual site-specific evolutionary relationships. In the presence of recombination, using a single tree to describe the evolution of the entire locus clearly leads to lower average recovery percentages than our method. Our study shows that recovering the actual local tree topologies can be done more accurately than estimating the actual number of recombination events.

**Keywords:** ancestral recombination graph, minimum number, lower bound, recombination, tree topology
**Running head:** Constructing Minimal Ancestral Recombination Graphs

## 1 Introduction

A central point in the analysis of within-species genetic variation is recombination. While creating new genetic types in the population, recombination has far-reaching consequences on the genealogy of chromosomes. As a result of recombination, different regions in DNA sequences can have different evolutionary histories and correlation between alleles at different loci can therefore be reduced. Finding out where in the sequence recombination is likely to have occurred and how much recombination rates vary is crucial for many important areas of study such as disease gene mapping.

The primary goal of our work is to construct possible evolutionary histories of DNA sequences which may have undergone recombination. Although reconstructing the true evolutionary history of sample sequences with certainty is not possible, it is possible to find a set of evolutionary histories that are consistent with the observed sequences. Here, one should note that the meaning of the word "consistent" depends on the assumed model of evolution. Furthermore, in general the more constraints the assumed model of evolution has, the more informative the data can be, thus reducing the number of consistent histories. In our work, we assume the infinite-sites model of mutation, which is equivalent to saying that at most one mutation event can occur at each site in the entire evolutionary history.

*Corresponding Author. E-mail: song@stats.ox.ac.uk

As a consequence of this assumption, we limit our study to the case where at most 2 distinct nucleotides occur at every site in the data. Note that, by arbitrarily assigning 0 or 1 to the distinct nucleotides appearing at each site, such data can easily be transformed into binary form, which we shall use in this paper. Throughout this paper, we let $S$ be a set of $n$ binary sequences each of length $\ell$, i.e. $S = \{s_\alpha\}, s_\alpha = c_1^\alpha, c_2^\alpha, \ldots, c_\ell^\alpha$, where $c_i^\alpha \in \{0, 1\}$ for every $\alpha \in \{1, 2, \ldots, n\}$ and $i \in \{1, 2, \ldots, \ell\}$. The entry $c_i^\alpha$ is called the $i^{\text{th}}$ *character* of sequence $s_\alpha$, and the $n$-tuple $\mathbf{c}_i := (c_i^1, c_i^2, \ldots, c_i^n)$ of $i^{\text{th}}$ characters is called the $i^{\text{th}}$ character *column*. A character column $\mathbf{c}_i$ is called *informative* if it contains at least two 0s and two 1s. Otherwise, it is called *non-informative*.

One of the reasons why it is in general impossible to reconstruct the true evolutionary history of sample sequences is that some recombination events are not detectable. That is, some recombination events do not lead to any betokening polymorphism in sampled sequences, and therefore there is no trace in the data that indicates the occurrence of such events in the past. Although it is impossible to know exactly how many recombination events have occurred in the evolutionary history of sample sequences, it is meaningful, however, to ask at least how many recombination events must have occurred in the evolutionary history. The goal of our present work is thus twofold; to find the minimum number $R_{\min}(S)$ of recombination events and to construct possible evolutionary histories with exactly $R_{\min}(S)$ recombination events. For a given data set $S$, the minimum number $R_{\min}(S)$ of recombination events is defined by the property that there exists no evolutionary history with less than $R_{\min}(S)$ recombination events that can generate the observed data $S$ under the infinite-sites model of mutation. We sometimes use the adjective "minimal" to describe evolutionary histories with the minimum number of recombination events.

Finding the minimum number $R_{\min}(S)$ of recombination events for an arbitrary data set $S$ is a difficult problem, and therefore several methods for estimating $R_{\min}(S)$ have been devised. The problem of finding a lower bound on $R_{\min}(S)$ was first considered by Hudson and Kaplan (1985), and that particular problem of counting recombination events is currently receiving a renewed interest. Myers and Griffiths (2003) recently proposed an integer linear programming approach for constructing new lower bounds on the number of recombination events, whereas the present authors proposed a set theoretical method to define a new lower bound (Song and Hein, 2004). None of these methods can always yield the minimum number; for some data set $S$, the computed lower bound may be less than the minimum number $R_{\min}(S)$. In contrast, as we later describe in detail, the method we use in our present work always yields the minimum number of recombination events if the so-called ordered trees are used in our algorithm.

The algorithmic ideas and some technical issues underlying our work have recently been announced at the WABI 2003 conference (Song and Hein, 2003). In the present paper, we further elaborate the method described there and investigate how well our method performs. We have implemented our algorithm in the programming language `C++` and our software, called `RecMinPath`, is available upon request. We define a lower bound based on rooted trees with less constraints than that in ordered trees and show that this lower bound generally performs better than some other presently-available lower bounds. In addition, using simulated data generated under the infinite-sites model of mutation, we investigate how well our method can recover the actual local evolutionary relationships. Our study shows that using a single tree, instead of using a sequence of trees as in our method, to

describe the evolution of the entire locus leads to lower average recovery percentages. Also, we can conclude from our study that recovering the actual local tree topologies can be done more accurately than estimating the actual number of recombination events.

The approach we take in our method is to find a sequence of trees—where the $i^{\text{th}}$ tree in the sequence describes the evolution of the $i^{\text{th}}$ site in the data—such that, as one moves along the sequence, changes in tree topology can be seen as signalling recombination events. To correctly quantify the number of recombination events, we have properly defined in (Song and Hein, 2003) the so-called subtree-prune-and-regraft (SPR) operations on ordered trees, where internal vertices are totally ordered. The SPR-distance between two ordered trees—defined as the minimum number of SPR operations required to transform one tree to the other—correctly encodes the number of recombination events required to put the two trees into an ancestral recombination graph (Griffiths and Marjoram, 1997). In addition, we can explicitly construct evolutionary histories that are consistent with the data by combining the trees which appear in the sequence of trees.

This paper is organised as follows. In Section 2, we briefly sketch the main idea underlying our work. The reader is recommended to browse through this section to obtain a rough picture of what we are trying to achieve. More technical aspects of our work are discussed in Section 3, where, for ease of reference, we also provide the algorithms from (Song and Hein, 2003). The reader may skip this section in the first reading and return to it later. In Section 4, we test the performance of our method on simulated data. We compare our method to some other currently-existing methods, in regard to both detecting recombination events and recovering local evolutionary relationships. Our analysis of Kreitman's 1983 data, consisting of 11 alleles of the alcohol dehydrogenase locus of *Drosophila melanogaster* (Kreitman, 1983), is also discussed in that section. In Section 5, we suggest a few concrete directions for extending our work, and we conclude with some remarks in Section 6.

## 2 An overview of the main idea

Consider the following five sequences:

$$\begin{array}{rcl} s_1 & = & 0\ 0\ 0\ 0 \\ s_2 & = & 0\ 0\ 1\ 1 \\ s_3 & = & 0\ 1\ 0\ 1 \\ s_4 & = & 1\ 1\ 0\ 0 \\ s_5 & = & 1\ 1\ 1\ 1 \end{array} \tag{1}$$

Within the framework of the infinite-sites model of mutation, two character sites are said to be *incompatible* if all four gametic types—00, 01, 10 and 11—appear in the two sites. In the above data, there are four pairs of incompatible sites. Namely, they are sites 1 and 3; sites 1 and 4; sites 2 and 3; and, sites 2 and 4. Whenever two sites $i$ and $j$ are incompatible, it implies that there must have occurred at least one recombination event with its corresponding breakpoint somewhere between $i$ and $j$. In that regard, our goal can be rephrased as finding the minimum number of recombination events needed to explain all incompatibilities present in the data.

The reader is encouraged to check that there exists no evolutionary history with only one recombination event that could have generated the above five sequences under the infinite-

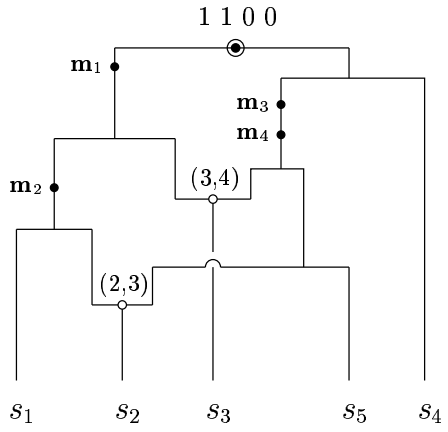


Figure 1: A minimal ancestral recombination graph corresponding to the data $S = \{s_1, s_2, \ldots, s_5\}$ shown in (1). Exactly 2 recombination vertices are present in the graph. Open circles $\circ$ denote recombination events, and the notation $(i, i+1)$ accompanying a recombination event indicates that its corresponding breakpoint occurs between sites $i$ and $i+1$. By convention, when recombination occurs, the part to the left (resp. right) of the breakpoint gets descended from the left (resp. right) edge. Filled circles $\bullet$ indicate mutation events. The notation $\mathbf{m}_i$ denotes a mutation event at character site $i$; it corresponds to a change of 0 to 1 or vice versa. The root is denoted by $\circledcirc$. In this particular ARG, the ancestral sequence at the root is 1 1 0 0.

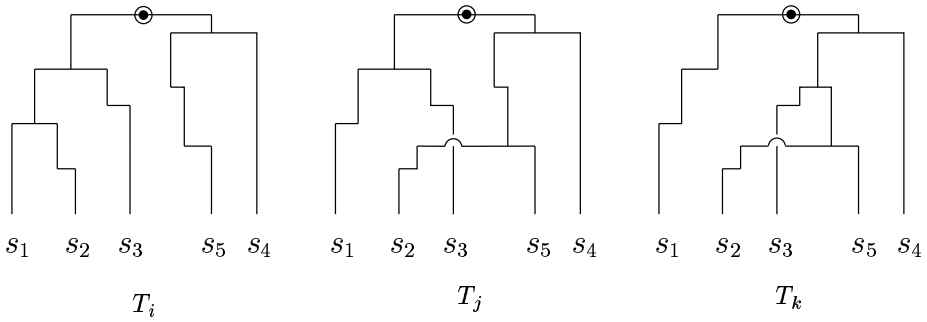


Figure 2: Trees embedded in the ARG shown in Figure 1.

sites model of mutation. In fact, at least two recombination events must have occurred in the evolutionary history of the sequences. Shown in Figure 1 is an ancestral recombination graph (ARG) with exactly two recombination vertices; it represents one of several minimal evolutionary histories consistent with the data $S = \{s_1, s_2, \ldots, s_5\}$.

In our work, we distinguish trees according to their topologies and the ordering of their internal vertices. The space $\mathscr{T}_n$ of leaf-labelled rooted binary trees with $n$ leaves therefore is a discrete space. We defer a more precise definition of a tree until Section 3. An ARG defines a collection of trees and carries the information about which tree describes the evolution of which character site. In our present example, the ARG shown in Figure 1 contains three inequivalent trees, namely the three shown in Figure 2. The evolution of character sites 1 and 2 is represented by the tree $T_i$, whereas $T_j$ describes the evolution of site 3 and $T_k$ that of site 4. Hence, we can think of the ARG as giving a sequence $P = T_i, T_i, T_j, T_k$ of

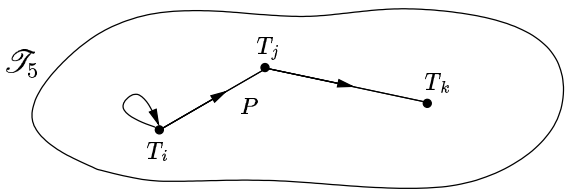


Figure 3: A schematic depiction of a path. This path arises from the sequence of trees associated with the ARG shown in Figure 1. The enclosed region represents the space $\mathscr{T}_5$ of leaf-labelled rooted binary trees with 5 leaves, and a point in this space corresponds to a tree.

trees, which in turn gives rise to a "path," also denoted $P$, in the space $\mathscr{T}_5$ as depicted in Figure 3. In what follows, we sometimes use the two terminologies "sequence of trees" and "path" interchangeably.

In general, to each ARG corresponding to a set of length-$\ell$ DNA sequences, one can associate a unique length-$\ell$ sequence $P$ of trees such that the $i^{\text{th}}$ tree in $P$ describes the evolution of the $i^{\text{th}}$ character site. Suppose that we are given an arbitrary data set $S$ of $n$ sequences and that $H$ is an ARG consistent with $S$. Let $P = T_{i_1}, T_{i_2}, \ldots, T_{i_\ell}$ denote the unique sequence of trees corresponding to $H$. As before, the corresponding path in the space $\mathscr{T}_n$ of trees with $n$ leaves is also denoted by $P$. Now, if we can define a notion of distance between any pair of points in $\mathscr{T}_n$, i.e. if the distance $d(T_a, T_b)$ is defined for all $T_a, T_b \in \mathscr{T}_n$, then we can define the length $L(P)$ of the path $P$ by adding up the pair-wise contributions along the path as follows:

$$L(P) := \sum_{k=1}^{\ell-1} d(T_{i_k}, T_{i_{k+1}}). \tag{2}$$

Let $\mathscr{P}(S)$ denote the set of all paths whose corresponding ARGs are consistent with $S$. The gist of our work lies in finding the right kind of trees (i.e. $\mathscr{T}_n$) and the right kind of metric (i.e. $d(\cdot,\cdot)$) so that the following equality holds:

$$\min_{P \in \mathscr{P}(S)} L(P) = R_{\min}(S). \tag{3}$$

In summary, instead of working with ARGs directly, we can work with sequences of trees or, equivalently, paths in the space of trees as shown in Figure 4. As we discuss later, this change in perspective allows us to use dynamic programming to find solutions to (3). Note that our method does two things for us. First of all, we can find the minimum number of recombination events using (3). Secondly, for each minimal path (i.e. a solution to (3))

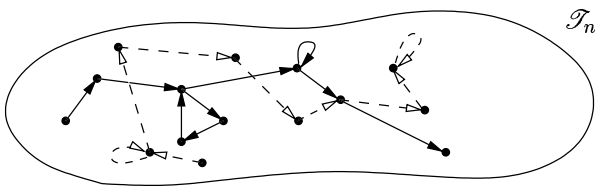


Figure 4: A schematic illustration of paths in the space $\mathscr{T}_n$ of trees with $n$-leaves.

we find, we can construct a set of minimal evolutionary histories (i.e. minimal ARGs) consistent with the data by combining the trees which arise in the minimal path.

# 3 Methods

In this section, we elaborate on the ideas sketched in the previous section. As well as discussing what kind of trees should be used, we describe how the distance between trees should be measured to reflect the number of recombination events. The algorithm for finding minimal paths is restated here for ease of reference. We refer the reader to (Song and Hein, 2003) for precise definitions and for a more in-depth description of our algorithm. As an additional application, we discuss how our work can be used to define haplotype blocks.

## 3.1 Trees

In this paper, we consider leaf-labelled rooted binary trees whose edge lengths are not specified. When we say a tree without any qualification, we shall mean a leaf-labelled rooted binary tree. In a rooted tree, time flows from the root to the leaves. In our way of drawing trees, time should flow vertically from top to bottom; horizontal lines in a tree do not carry any temporal meaning and their lengths are chosen arbitrarily.

The set $\{v_1, v_2, \ldots, v_{n-2}\}$ of degree-3 vertices in an $n$-leaved rooted binary tree is a *partially* ordered set whose binary relation denoted $<$ is given by ancestral relation. More precisely, $v_i < v_j$ if $v_i$ is an ancestor of $v_j$. Two degree-3 vertices $v_i$ and $v_j$ are *incomparable* if there exists no ancestral relation between them. An *ordered tree* is a leaf-labelled rooted binary tree whose corresponding set $\{v_1, v_2, \ldots, v_{n-2}\}$ of degree-3 vertices is a *totally* ordered set; that is, for any two vertices $v_i$ and $v_j$, either $v_i < v_j$ or $v_j < v_i$. In this case, the binary relation $<$ is given by age ordering. As before, $v_i < v_j$ if $v_i$ is an ancestor of $v_j$. If there exists no ancestral relation between $v_i$ and $v_j$, then either $v_i < v_j$ or $v_j < v_i$ is allowed. Furthermore, we impose the condition that $v_i \neq v_j$ if $i \neq j$. The space of plain rooted trees and that of ordered trees are denoted by $\mathscr{T}_n^{\mathrm{r}}$ and $\mathscr{T}_n^{\mathrm{o}}$, respectively.

Two trees equivalent as rooted trees are distinct as ordered trees if the ordering of their degree-3 vertices are different. For example, the two trees shown in Figure 5 are equivalent as plain rooted trees but inequivalent as ordered trees. The number of inequivalent ordered trees corresponding to a fixed plain rooted tree $T$ depends on the topology of $T$.

A tree $T$ is said to be *compatible* with the informative column $\mathbf{c}_i$ if there exists an edge in $T$ such that cutting the edge decomposes $T$ into two connected components, one labelled

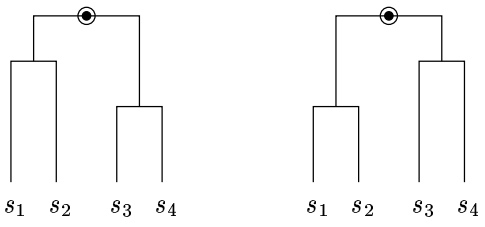


Figure 5: Examples of four-leaved trees. These trees are equivalent as plain rooted trees but inequivalent as ordered trees.

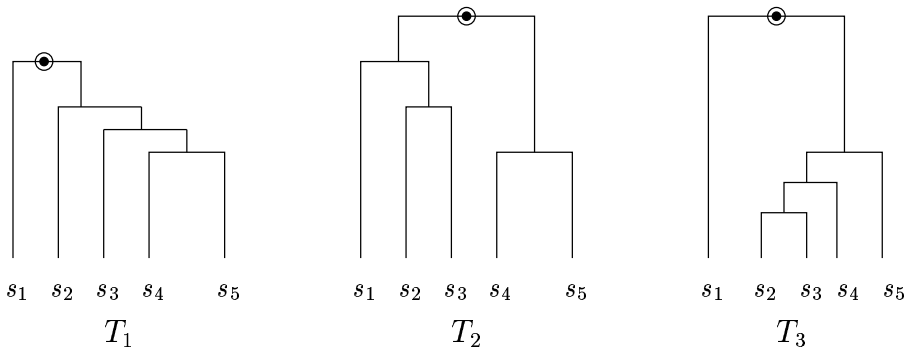


Figure 6: Trees with 5 leaves. If $T_2$ and $T_3$ are viewed as plain rooted trees, then $d_{\text{SPR}}(T_2, T_3) = 1$. If they are viewed as ordered trees, then $d_{\text{SPR}}(T_2, T_3) = 2$.

by the sequences $s_\alpha$ with $c_i^\alpha = 0$ and the other by the sequences $s_\beta$ with $c_i^\beta = 1$. If the column $\mathbf{c}_i$ is not informative, then every $n$-leaved tree is considered compatible with $\mathbf{c}_i$. In relation to evolutionary history, if a tree is compatible with a character column $\mathbf{c}_i$, then at most one mutation event is necessary for the tree to represent the evolutionary history of sample sequences at the character site $i$, i.e. the tree is consistent with the character column under the infinite-sites model of mutation.

### 3.2 SPR operations and the SPR-distance between trees

A type of distance between trees widely used in biology is that defined in terms of certain operations which rearrange trees; the distance between two trees is defined as the minimum number of operations required to transform one tree to the other. A particular kind of operation that we employ in our work is the so-called *subtree pruning and regrafting* (Swofford and Olsen, 1990). In a subtree-prune-and-regraft (SPR) operation, one detaches an edge from a tree $T$, thus "pruning" a subtree $t$ from $T$, and "regrafts" $t$ to somewhere else on the remaining part of $T$. In (Song and Hein, 2003), we properly defined SPR operations for plain rooted trees and ordered trees so that they can be used to quantify recombination events.

The precise definition of an SPR operation depends on the type of the tree on which the operation is being performed. The reader should refer to (Song and Hein, 2003) for details. In general, the more characteristics a tree has, the more restrictive the definition of an SPR operation has to be. For instance, SPR operations on ordered trees are more restricted than SPR operations on plain rooted trees. Consider the trees $T_2$ and $T_3$ shown in Figure 6. If these trees are viewed as plain rooted trees, then the 2-leaved subtree in $T_2$ containing $s_2$ and $s_3$ can be pruned and then regrafted onto the edge incident with the leaf labelled $s_4$. Hence, only one SPR operation is required to transform $T_2$ to $T_3$ if they are thought of as plain rooted trees. If the two trees are viewed as ordered trees, however, the SPR operation just described is not an allowed operation. In this case, at least 2 SPR operations—for example, one moving the leaf labelled $s_2$ and the other moving the leaf labelled $s_3$—are needed to transform $T_2$ to $T_3$, or vice versa.

For any pair of trees, say $T$ and $T'$, their SPR-distance $d_{\text{SPR}}(T, T')$ is a non-negative integer defined as the minimum number of SPR operations necessary to transform $T$ into $T'$.

In practice, determining the SPR-distance between two arbitrary rooted trees can be quite difficult, especially so for ordered trees. A combinatorial analysis of SPR operations on plain rooted trees has been carried out in (Song, 2003), and we are currently investigating analogous questions for ordered trees.

### 3.3 Why SPR operations?

Given a sequence $P = T_{i_1}, T_{i_2}, \ldots, T_{i_\ell}$ of trees, equations (2) and (3) together imply that we should define the distance between two consecutive trees $T_{i_k}$ and $T_{i_{k+1}}$ so that it is equal to the number $d_{\rm rec}(T_{i_k}, T_{i_{k+1}})$ of recombination events needed to combine the two trees into an ARG. As one moves along the sequence, tree topology can change, and such topology changes can be seen as being obtained from SPR operations. A subtree that gets pruned and regrafted in an SPR operation corresponds to the descendant subtree of a recombination vertex in an ARG. Hence, the essential point is that the SPR-distance can correctly quantify the number of recombination events. For example, recall that the trees shown in Figure 2 are embedded in the ARG shown in Figure 1, and that the sequence $P = T_i, T_i, T_j, T_k$ of trees can be associated to the ARG. The tree $T_i$ can be transformed into $T_j$ through a single SPR operation involving the leaf labelled $s_2$. Likewise, the tree $T_j$ can be transformed into $T_k$ through a single SPR operation involving the leaf labelled $s_3$. Thus, the minimum total number of required SPR operations is indeed equal to the number of recombination events in the minimal ARG.

### 3.4 Why ordered trees?

In an ARG, coalescent events and recombination events occur in certain order and ignoring the time ordering of these events can lead to contradictions. For example, biologically a recombinant cannot be older than its parents. Let us return to the trees shown in Figure 6. Suppose that $T_1, T_2, T_3$ describe the evolutionary history of the first, the second and the third character sites, respectively. If these trees are viewed as plain rooted trees, then $d_{\rm SPR}(T_1, T_2) = 1$ and $d_{\rm SPR}(T_2, T_3) = 1$, and therefore we would obtain 2 as the number of recombination events needed for combining the trees into an ARG. But, there exists no proper ARG with only 2 recombination vertices such that it is consistent with the sequence $T_1, T_2, T_3$. The reader is strongly encouraged to think about why this is so. As a specific case, consider the graph shown on the left hand side of Figure 7. This graph does have $T_1, T_2, T_3$ as its associated sequence of plain rooted trees, but the direction of time flow in the edge labelled $e$ is reversed, i.e. the time flow in that edge is from bottom to top instead of from top to bottom. Hence, this graph is not a proper ARG.

On the other hand, if the three trees are viewed as ordered trees, with the particular ordering of internal vertices as shown in Figure 6, then we would conclude that $d_{\rm SPR}(T_1, T_2) = 1$ and that $d_{\rm SPR}(T_2, T_3) = 2$, thus obtaining 3 as the number of recombination events needed. Shown on the right hand side of Figure 7 is a self-consistent ARG with 3 recombination vertices; it correctly describes the change of $T_1$ to $T_2$, which in turn changes to $T_3$.

We emphasise that, in general, using plain rooted trees in our method can lead to an underestimation of the number of recombination events. If ordered trees are used, however, we always have $d_{\rm SPR}(T, T') = d_{\rm rec}(T, T')$.

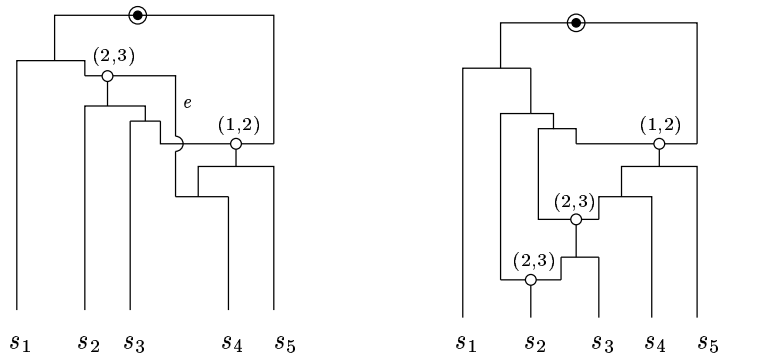


Figure 7: An important difference between using plain rooted trees and using ordered trees. Contradictions can arise if plain rooted trees are used in the algorithm, whereas using ordered trees always leads to consistent ARGs.

### 3.5 The original algorithm

The algorithm for finding minimal paths in the general case was first proposed by one of us in (Hein, 1990). In the case of the infinite-sites model of mutation, the original algorithm can be rephrased as follows. For each character column $\mathbf{c}_i$ of $S$, where $i \in \{1, 2, \ldots, \ell\}$, let $W_i^{\mathrm{o}}$ denote the set of all ordered trees in $\mathscr{T}_n^{\mathrm{o}}$ that are compatible with $\mathbf{c}_i$. The main idea of the algorithm is to evaluate a sequence of functions $f_1, f_2, \ldots, f_\ell$, where $f_i$ is a non-negative-integer-valued function defined on $W_i^{\mathrm{o}}$. The first function $f_1$ is defined to be identically zero, i.e. $f_1(T) = 0$ for all $T \in W_1^{\mathrm{o}}$, and every subsequent function $f_{i+1}$ is recursively defined in terms of $f_i$ and the recombination distance $d_{\mathrm{rec}}(\cdot, \cdot)$. More exactly, the algorithm can be stated as constructing a weighted graph $G$ as follows.

1. Introduce $\ell$ clusters, with the $i^{\mathrm{th}}$ cluster containing $|W_i^{\mathrm{o}}|$ vertices labelled by the trees in $W_i^{\mathrm{o}}$.

   (a) For all $T \in W_1^{\mathrm{o}}$, let $f_1(T) = 0$.

   (b) For all $1 \leq i < \ell$, recursively determine

   $$f_{i+1}(T_a) = \min_{T_b \in W_i^{\mathrm{o}}} [f_i(T_b) + d_{\mathrm{rec}}(T_b, T_a)] \tag{4}$$

   for every tree $T_a \in W_{i+1}^{\mathrm{o}}$.

   (c) In the weighted graph $G$, vertices $T_a \in W_{i+1}^{\mathrm{o}}$ and $T_b \in W_i^{\mathrm{o}}$ are joined by an edge if $f_{i+1}(T_a) - f_i(T_b) = d_{\mathrm{rec}}(T_a, T_b)$, and the weight of the edge is $d_{\mathrm{rec}}(T_a, T_b)$.

2. The number defined as

   $$R_{\mathrm{o}}(S) = \min_{T_a \in W_\ell^{\mathrm{o}}} f_\ell(T_a) \tag{5}$$

   gives the minimum number $R_{\min}(S)$ of recombination events. A connected path from any tree $T_a \in W_1^{\mathrm{o}}$ to a tree $T_b \in W_\ell^{\mathrm{o}}$ with $f_\ell(T_b) = R_{\mathrm{o}}(S)$ is called a *minimal path* in $G$.

Note that a heuristic implementation of the algorithm was made about a decade ago (Hein, 1993). An exact implementation of the idea, however, could not be carried out so far due to several difficulties, the major one being the complexity involved in computing the recombination distance $d_{\rm rec}(T, T')$ between two arbitrary trees $T$ and $T'$. In (Hein, 1993), unrooted trees were used in the implementation of the algorithm and it was assumed that at most one recombination event occurs between any two adjacent character columns. In our present work, we have implemented the above algorithm for ordered trees, without any heuristic assumptions. Note that $d_{\rm SPR}(T, T') = d_{\rm rec}(T, T')$ if $T, T'$ are ordered trees.

We stress that the algorithm described here always gives $R_{\rm o}(S) = R_{\rm min}(S)$. On the contrary, as discussed in Subsection 3.4, if plain rooted trees are used in the algorithm—that is, if $W_i^{\rm r}$, the set of plain rooted trees in $\mathscr{T}_n^{\rm r}$ compatible with $\mathbf{c}_i$, are used instead of $W_i^{\rm o}$—to obtain

$$R_{\rm r}(S) := \min_{T_a \in W_\ell^{\rm r}} f_\ell(T_a), \tag{6}$$

then this number $R_{\rm r}(S)$ may or may not be equal to the minimum $R_{\rm min}(S)$. In general, $R_{\rm r}(S) \le R_{\rm o}(S) = R_{\rm min}(S)$. As we discuss later, however, for less than or equal to 9 sequences, our investigation shows that $R_{\rm r}(S) = R_{\rm min}(S)$ for most cases of $S$. A possible explanation of this phenomenon can be found in Subsection 4.2.

As discussed in (Song and Hein, 2003), it is important to note that the number of ordered trees grows much faster than the number of plain rooted trees. For instance, there are over 57 million 9-leaved ordered trees, whereas there are about 2 million plain rooted trees with 9 leaves. So, for $n \ge 9$, it would be a good strategy to use plain rooted trees first to compute $R_{\rm r}(S)$ and try to construct possible evolutionary histories. If a consistent history can be constructed using only $R_{\rm r}(S)$ recombination events, then we can conclude that $R_{\rm r}(S) = R_{\rm min}(S)$.

### 3.6 A modified algorithm

Since computing the distance $d_{\rm rec}(T, T')$ for arbitrary $T, T' \in \mathscr{T}_n^{\rm o}$ is rather difficult, step 1(b) in the above algorithm is computationally intensive. For example, working with trees with many restrictions and computing the distance between two arbitrary such trees can be very complicated. As described in (Song and Hein, 2003), however, our newly proposed algorithm, together with our way of viewing recombination events as properly defined SPR operations on trees, can allow us to overcome some computational difficulties which have hitherto prevented an exact implementation of the dynamic programming idea.

In (Song and Hein, 2003), we showed that, for all $T_a, T_b \in W_i^{\rm o}$, where $i \ge 2$, the function $f_i$ defined in (4) satisfies $|f_i(T_a) - f_i(T_b)| \le d_{\rm rec}(T_a, T_b)$. It follows from this result that (4) can be replaced by

$$f_{i+1}(T_a) = \begin{cases} f_i(T_a), & \text{if } T_a \in W_i^{\rm o}, \\ \min_{T_b \in W_i^{\rm o}} \left[ f_i(T_b) + d_{\rm rec}(T_b, T_a) \right], & \text{otherwise.} \end{cases}$$

The same result holds if plain rooted trees (i.e. $W_i^{\rm r}$ ) are used in the algorithm. In addition, we proposed an alternative way of carrying out the dynamic programming algorithm. The new method can be applied to either plain rooted trees or ordered trees, depending on whether one wishes to compute $R_{\rm r}(S)$ or $R_{\rm o}(S)$, respectively. As mentioned before, the

relations $R_{\mathrm{r}}(S) \leq R_{\mathrm{o}}(S) = R_{\min}(S)$ hold true for all $S$. In what follows, we describe our modified algorithm using the notations for ordered trees.

Although determining the SPR-distance between two arbitrary rooted trees can be quite difficult, it is not very difficult to determine whether two trees are one SPR operation away. Hence, our approach is to determine first which trees are distance one away from each other and then use that information to compute $d_{\mathrm{SPR}}(T, T')$ for arbitrary $T$ and $T'$.

By the adjacency-set of a tree $T \in \mathscr{T}_n^{\mathrm{o}}$ we mean the set

$$U(T) := \{T' \in \mathscr{T}_n^{\mathrm{o}} \mid d_{\mathrm{SPR}}(T, T') = 1\}.$$

For $X \subset \mathscr{T}_n^{\mathrm{o}}$, let $N_0(X) = X$ and, for $r \geq 1$, define the $r$-neighbourhood of $X$ as

$$N_r(X) := \left\{T \in \mathscr{T}_n^{\mathrm{o}} \,\middle|\, d_{\mathrm{SPR}}(T, T') \leq r \text{ for some } T' \in X\right\}.$$

In the implementation of our algorithm, we pre-compute the adjacency-set $U(T)$ for all $T \in \mathscr{T}_n^{\mathrm{o}}$ and store them in a file which can be accessed by our program, and therefore computing $N_r(X)$ can easily be done. We define the diameter $d_n$ of $\mathscr{T}_n^{\mathrm{o}}$ as the maximum value of $d_{\mathrm{SPR}}(T, T')$ over all trees $T, T' \in \mathscr{T}_n^{\mathrm{o}}$. As shown in (Song, 2003), $d_n \leq n - 2$. In the following discussion, define $\mathcal{N}(T, m, i) := N_1(\{T\}) \cap N_m(W_i^{\mathrm{o}})$. Note that $N_1(\{T\})$ is none other than $\{T\} \cup U(T)$.

Let $f_{1,0}(T) = 0$ for all $T \in \mathscr{T}_n^{\mathrm{o}}$. For all $1 \leq i < \ell$, recursively compute the following quantities:
For all $1 \leq r < d_n$ and $T_a \in N_r(W_i^{\mathrm{o}})$, find

$$f_{i,r}(T_a) = \begin{cases} f_{i,r-1}(T_a), & \text{if } T_a \in W_i^{\mathrm{o}}, \\ \min\limits_{T_b \in \mathcal{N}(T_a, r-1, i)} \left[f_{i,r-1}(T_b) + 1 - \delta_{a,b}\right], & \text{otherwise,} \end{cases} \tag{7}$$

and, for all $T_a \in W_{i+1}^{\mathrm{o}}$, find

$$f_{i+1,0}(T_a) = \begin{cases} f_{i,d_n-1}(T_a), & \text{if } T_a \in W_i^{\mathrm{o}}, \\ \min\limits_{T_b \in \mathcal{N}(T_a, d_n-1, i)} \left[f_{i,d_n-1}(T_b) + 1 - \delta_{a,b}\right], & \text{otherwise.} \end{cases} \tag{8}$$

Here, $\delta_{a,b}$ denotes the Kronecker delta, which is 1 if $a = b$ or 0 if $a \neq b$. The minimum number of recombination events is given by $R_{\mathrm{o}}(S) := \min_{T \in W_\ell^{\mathrm{o}}} f_{\ell,0}(T)$, which is equal to the value $\min_{T \in W_\ell^{\mathrm{o}}} f_\ell(T)$ defined in (5).

There are several advantages to the algorithm just described over the original algorithm. First of all, note that for each tree $T_a$ in (7), we just need to compare previous function values at at most $|N_1(T_a)| = |U(T_a)| + 1$ trees. As discussed in (Song and Hein, 2003), the maximum size of $|U(T)|$ does not grow nearly as fast as the number of trees compatible with a character column. Secondly, we do not need to compute the SPR-distance explicitly; the algorithm effectively computes the SPR-distance for us and correctly updates $f_{i+1,0}(T)$, for all $1 \leq i < \ell$.

Note that a weighted graph $G$, analogous to that described in the original algorithm, can be constructed with straightforward extra bookkeeping. A *minimal path* in $G$ is a connected path from any tree $T_a \in W_1^{\mathrm{o}}$ to a tree $T_b \in W_\ell^{\mathrm{o}}$ with $f_{\ell,0}(T_b) = R_{\mathrm{o}}(S)$.

### 3.7 Data reduction

In performing our algorithm, some character columns do not influence the determination of $R_o(S)$ and therefore can be ignored. Also, identical sequences can be merged into one, thus significantly reducing the complexity of the algorithm. It is straightforward to show that, before one carries out any analysis on $S$, reducing the data as follows does not change the value of $R_o(S)$; i.e., if $S'$ denotes the reduced data, then $R_o(S) = R_o(S')$. Define $\bar{\mathbf{c}}_i := (\bar{c}_i^1, \bar{c}_i^2, \ldots, \bar{c}_i^n)$, where $\bar{c}_i^\alpha = 0$ if $c_i^\alpha = 1$ and $\bar{c}_i^\alpha = 1$ if $c_i^\alpha = 0$.

1. Collapse identical sequences into one.
2. Remove all non-informative columns from $S$. Let $\mathbf{c}'_1, \mathbf{c}'_2, \ldots, \mathbf{c}'_{\ell'}$ denote the character columns in the resulting data.
3. Collapse all consecutive columns $\mathbf{c}'_i, \mathbf{c}'_{i+1}, \ldots, \mathbf{c}'_{i+k}$ where $\mathbf{c}'_{i+j} = \mathbf{c}'_i$ or $\mathbf{c}'_{i+j} = \bar{\mathbf{c}}'_i$, for all $j = 1, 2, \ldots, k$, into a single column $\mathbf{c}'_i$.
4. Sequentially repeat steps $1 \sim 3$ until none of them is possible.

Note that the number of sequences in the reduced data $S'$ can be much less than the number of sequences in $S$. Our current implementation of the algorithm can analyse up to 8 (resp. 9) sequences in the reduced data if ordered (resp. plain rooted) trees are used in the algorithm.

### 3.8 The number of ARGs corresponding to a minimal path

In general, more than one ARG corresponds to a minimal path. There are two reasons for this many-to-one correspondence. First of all, there can be more than one inequivalent set of SPR operations that can transform one tree to another. Secondly, the ordering of internal vertices in an ARG is not completely fixed by the ordering of internal vertices in ordered trees.

Let us demonstrate these points through an explicit example. Consider the path $P = T, T'$ where $T$ and $T'$ are shown in Figure 8. The SPR-distance between $T$ and $T'$ is 1, i.e. the ordered tree $T$ can be transformed into $T'$ by a single SPR operation. Note, however, that there is more than one inequivalent SPR operation that can transform $T$ into $T'$. More exactly, the subtree used can be a 1-leaved subtree—containing either $s_1$ or $s_2$—or it can be the 2-leaved subtree containing $s_3$ and $s_4$.

If the 1-leaved subtree containing $s_1$ is pruned and regrafted, then, as shown in Figure 9(a), there are 2 inequivalent ARGs corresponding to this choice. This results from the

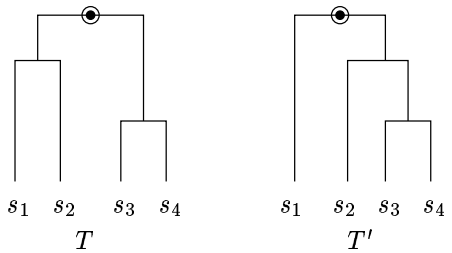


Figure 8: Ordered trees appearing in the path $P = T, T'$.

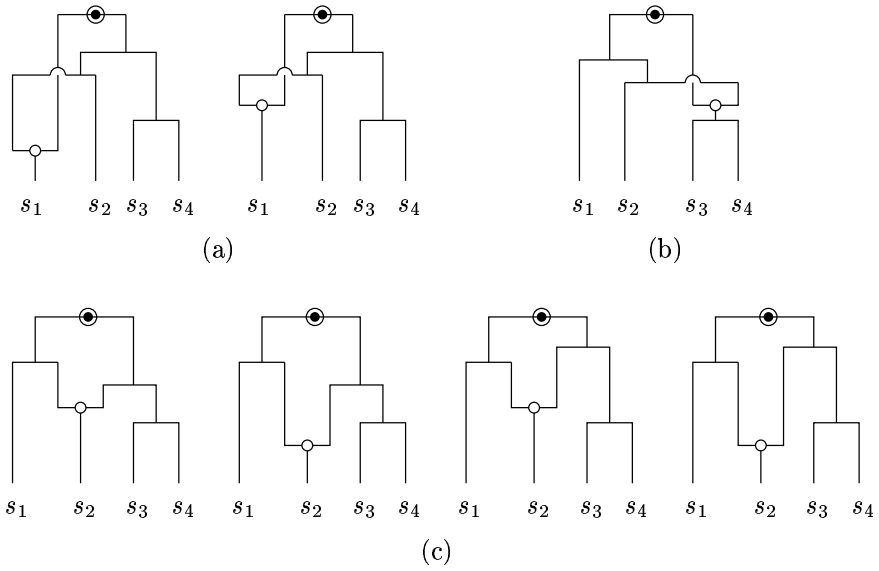


Figure 9: Inequivalent ARGs that can be constructed from combining the trees shown in Figure 8.

fact that there are 2 possibilities in ordering the recombination vertex with respect to the root of the subtree containing $s_3$ and $s_4$. If the 1-leaved subtree containing $s_2$ is pruned and regrafted, then we can construct 4 inequivalent ARGs, which are illustrated in Figure 9(c). In addition to the 2 possible choices of ordering described in the previous case, there is an additional freedom in choosing the ordering of some coalescent vertices. On the contrary, if the 2-leaved subtree containing $s_3$ and $s_4$ undergoes an SPR operation, then there exists a unique corresponding ARG. This unique ARG is shown in Figure 9(b).

In the simple example just described, only a single recombination vertex appears. If more than one recombination vertices are present, then relative ordering of the recombination vertices will, in general, lead to additional inequivalent ARGs.

### 3.9 Haplotype blocks

Perhaps the most interesting recent finding in haplotype analysis is the discovery of the possible existence of haplotype block structures in the human genome (Daly *et al.*, 2001; Johnson *et al.*, 2001; Gabriel *et al.*, 2002), where a block is roughly characterised by the existence of high linkage disequilibrium and limited haplotype diversity. In this subsection, we propose a new way of defining haplotype blocks. Unlike previous proposals, our proposal explicitly takes possible evolutionary histories into account.

For each minimal path our algorithm finds, in addition to knowing which trees are used, we know exactly where in the sequence each tree is supported. Hence, we can associate a candidate haplotype block structure to each minimal path. That is, we define a block as the maximal set of consecutive positions in the sequence where the same tree is supported. As there could be many minimal paths, it is possible that there are many inequivalent candidate haplotype block structures predicted by our algorithm. By studying all inequivalent candidate block structures, however, we may be able to learn something useful. For example, many or all structures may share one or more common blocks, thus indicating the

robustness of those particular blocks.

Although in general we cannot find a unique haplotype block structure, we can still ask the following question, to which there exists a unique answer for a given data set $S$: If a block is obtained as described above, what is the minimum number of haplotype blocks that can be defined by a minimal path? We address this question in Subsection 4.4, where we consider a real biological application of our method.

# 4 Results

In this section, we apply our method to analyse some data, both simulated and real. All simulated data were generated using Hudson's program `ms` (Hudson, 2002). The program `ms` assumes the Wright-Fisher neutral model of evolution and the infinite-sites model of mutation. Input parameters relevant for our purpose are the scaled mutation rate $\theta = 4N_0\mu$ and the scaled recombination rate $\rho = 4N_0 r$. Here, $N_0$ denotes the effective population size and $\mu$ (resp. $r$) denotes the mutation (resp. recombination) rate per generation for the entire locus being simulated. In addition, the program uses a finite-sites uniform recombination model (Hudson, 1983), and therefore the number of sites in-between which recombination can occur must be specified. This number is denoted `nsites`. In our simulation, we used a constant population size with no migration.

## 4.1 Comparisons of lower bounds

Myers and Griffiths (2003) proposed an integer linear programming approach for constructing new lower bounds on the number of recombination events. Their algorithm uses local bounds for sub-intervals to construct a global bound for the entire data. Conforming to their notation, we let $R_{\mathrm{h}}(S)$ denote their "haplotype bound," which is obtained from applying the aforementioned integer linear programming algorithm on a set of inequalities relating the number of recombination events, the number of distinct haplotypes and the number of segregating sites. Their alternative lower bound based on simulation of sample history is denoted here by $R_{\mathrm{s}}(S)$. In general, $R_{\mathrm{s}}(S)$ is a sharper lower bound than $R_{\mathrm{h}}(S)$; that is, $R_{\mathrm{h}}(S) \leq R_{\mathrm{s}}(S) \leq R_{\min}(S)$.

Myers and Griffiths showed that their lower bounds significantly improve on Hudson and Kaplan's earlier bound (Hudson and Kaplan, 1985), which is based on a coarse analysis of incompatibility patterns of character sites. In what follows, we investigate how our lower bound $R_{\mathrm{r}}(S)$ (defined in (6)) performs in comparison to the bounds proposed by Myers and Griffiths. As always, $n$ denotes the number of sequences in the data $S$. In our simulations, we fixed $n = 8$ and $\theta = 15$. Note that, because we are using the neutral model of evolution with the infinite-sites model of mutation, fixing $n$ and $\theta$ fixes the expected number of segregating sites generated (Hudson, 1983). We used 9 different values of $\rho$ and fixed `nsites`=3000 for all simulations. For each value of $\rho$, 1000 runs of simulation were performed. Our investigation here is not meant to be extensive. Rather, our goal is to capture the rough relative behaviour of the lower bounds in comparison.

In computing $R_{\mathrm{s}}(S)$ and $R_{\mathrm{h}}(S)$, we set both the maximum subset size and the maximum width value to 300, which is larger than the length of any simulated data; this high number was chosen to ensure that $R_{\mathrm{s}}(S)$ and $R_{\mathrm{h}}(S)$ are as large as possible. Also, we assumed that

Table 1: A summary of relative performance of the lower bounds on simulated data.

| | $R_{\rm r}$ vs $R_{\rm h}$ | | | $R_{\rm r}$ vs $R_{\rm s}$ | | | $R_{\rm s}$ vs $R_{\rm h}$ | | |
|---|---|---|---|---|---|---|---|---|---|
| $\rho^{\rm a}$ | $>$ | $=$ | $<$ | $>$ | $=$ | $<$ | $>$ | $=$ | $<$ |
| 5 | 34 | 966 | 0 | 12 | 988 | 0 | 22 | 972 | 0 |
| 10 | 108 | 892 | 0 | 52 | 948 | 0 | 59 | 941 | 0 |
| 15 | 163 | 837 | 0 | 94 | 906 | 0 | 72 | 928 | 0 |
| 20 | 202 | 798 | 0 | 122 | 878 | 0 | 89 | 911 | 0 |
| 25 | 271 | 729 | 0 | 186 | 813 | 1 | 99 | 901 | 0 |
| 35 | 372 | 628 | 0 | 282 | 718 | 0 | 123 | 877 | 0 |
| 45 | 433 | 567 | 0 | 341 | 659 | 0 | 121 | 879 | 0 |
| 50 | 456 | 551 | 0 | 368 | 638 | 1 | 126 | 872 | 0 |
| 60 | 511 | 489 | 0 | 413 | 586 | 1 | 139 | 861 | 0 |
| 70 | 552 | 448 | 0 | 477 | 523 | 0 | 125 | 875 | 0 |
| 75 | 601 | 399 | 0 | 530 | 470 | 0 | 130 | 870 | 0 |

Note — Under $R$ vs $R'$, the column labelled by "$>$" reports the number of cases (out of 1000) where the relation $R > R'$ holds. Columns labelled by "$=$" and "$<$" are similarly defined. We fixed $n = 8, \theta = 15$ and `nsites=3000` in all simulations. For ease of notation, we omit the dependence on $S$ and simply write $R_{\rm r}$, etc. In each column labelled "$=$," the number of cases that $R = R' = 0$ is $351, 180, 88, 57, 30, 13, 4, 6, 1, 2, 0$ for $\rho = 5, 10, 15, 20, 25, 35, 45, 50, 60, 70, 75$, respectively.

[a] Scaled recombination rate.

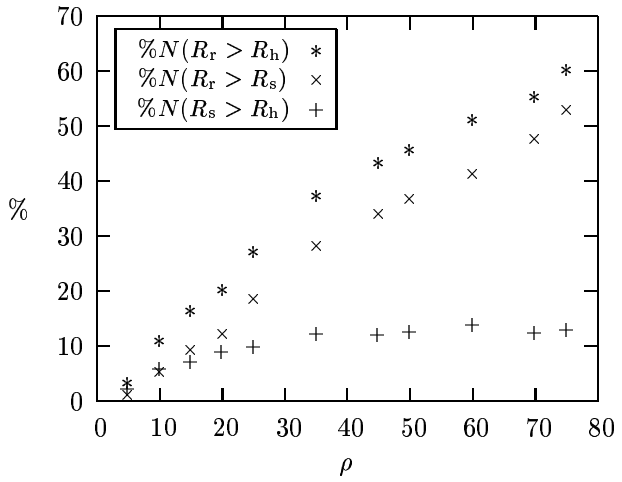


Figure 10: Relative sharpness of the lower bounds for $n = 8$ and $\theta = 15$. These percentages were computed using the numbers from Table 1.

the ancestral allele type is unknown. See (Myers and Griffiths, 2003) for details. For ease of notation, we sometimes omit the dependence on $S$ and simply write $R_{\rm r}$, etc. Results of our study are summarised in Table 1 and are illustrated in Figure 10. We use $N(R > R')$ to denote the number of cases (out of 1000) where the relation $R > R'$ holds. The notations $N(R = R')$ and $N(R < R')$ are similarly defined.

There are several things to be noted from this study. First of all, Table 1 shows that $N(R_{\rm r} < R_{\rm h}) = 0$ for every value of $\rho$ we used. In other words, our lower bound $R_{\rm r}(S)$ is sharper than or equal to the haplotype bound $R_{\rm h}(S)$, i.e. $R_{\rm r}(S) \geq R_{\rm h}(S)$, for all data

Table 2: Mean numbers of detected recombination events in the simulated data for $n = 8$ and $\theta = 15$.

| $\rho^{a}$ | $\mathbb{E}(R_{\rm r})$ | $\mathbb{E}(R_{\rm s})$ | $\mathbb{E}(R_{\rm h})$ |
|---|---|---|---|
| 5 | 1.09 | 1.08 | 1.06 |
| 10 | 1.95 | 1.89 | 1.84 |
| 15 | 2.62 | 2.52 | 2.45 |
| 20 | 3.18 | 3.06 | 2.97 |
| 25 | 3.82 | 3.63 | 3.52 |
| 35 | 4.89 | 4.58 | 4.46 |
| 45 | 5.36 | 4.98 | 4.85 |
| 50 | 5.74 | 5.33 | 5.21 |
| 60 | 6.47 | 5.98 | 5.84 |
| 70 | 7.08 | 6.48 | 6.34 |
| 75 | 7.21 | 6.57 | 6.43 |

Note — $\mathbb{E}(R)$ denotes the mean of the lower bound $R$. All numbers are based on our analysis of the simulated data used in Table 1.

[a] Scaled recombination rate.

$S$ we generated. Furthermore, for each case of $\rho = 25, 50$ and 60, the bound $R_{\rm s}(S)$ from simulation of sample history is sharper than our lower bound $R_{\rm r}(S)$ in only 1 out 1000 simulations; for all other values of $\rho$, we have $R_{\rm r}(S) \geq R_{\rm s}(S)$ for all $S$.

Secondly, Figure 10 shows that, as $\rho$ increases, the advantage of using $R_{\rm s}$ over using $R_{\rm h}$ ceases to depend on $\rho$. More precisely, as $\rho$ increases, the height (%) of the points marked by "+" signs tends to increase initially but fluctuates between 12% and 14% for $\rho > 35$. This seems to indicate that the change in the frequency that $R_{\rm s}$ is sharper than $R_{\rm h}$ has little dependence on the recombination rate $\rho$ for large $\rho$. In contrast, as the points marked by "$*$" and "$\times$" in Figure 10 clearly show, the frequency that $R_{\rm r}$ is sharper than $R_{\rm s}$ or $R_{\rm h}$ significantly increases as $\rho$ increases.

Lastly, we can compare the mean number of recombination events detected. Shown in Table 2 is a summary of this mean for the simulated data. For every value of $\rho$ we used, it is clear that $\mathbb{E}(R_{\rm h}) < \mathbb{E}(R_{\rm s}) < \mathbb{E}(R_{\rm r})$, where $\mathbb{E}(R)$ denotes the mean value of the lower bound $R$. Moreover, as illustrated in Figure 11, the differences $[\mathbb{E}(R_{\rm r}) - \mathbb{E}(R_{\rm s})]$ and $[\mathbb{E}(R_{\rm r}) - \mathbb{E}(R_{\rm h})]$ tend to increase as $\rho$ increases, whereas the difference $[\mathbb{E}(R_{\rm s}) - \mathbb{E}(R_{\rm h})]$ does not vary as much with respect to $\rho$. Hudson and Kaplan (1985) showed that the expected number of actual recombination events is given by

$$\mathbb{E}(R_{\rm a}) = \rho \sum_{j=1}^{n-1} \frac{1}{j}, \tag{9}$$

whereas the expected number of recombination events causing topology changes in local trees, seen as unrooted trees, is given by

$$\mathbb{E}(R_{\rm t}) = 16\rho \sum_{k=4}^{n} \left\{ \frac{1}{(k+1)k^2(k-1)^2} \sum_{i=2}^{k-2} \left[ \frac{1}{i} \sum_{j=2}^{i} j^2(j+1) \right] \right\}. \tag{10}$$

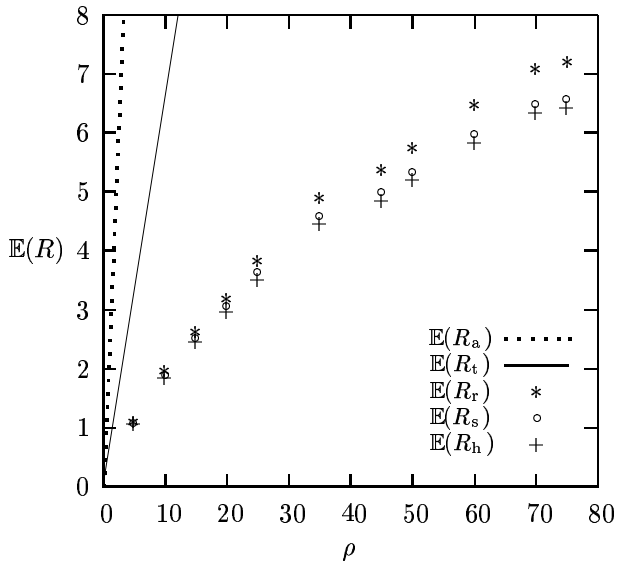


Figure 11: Comparisons of the mean number of detected recombination events for $n = 8$ and $\theta = 15$. The dotted line is the mean number of actual recombination events, whereas the solid line is the mean number of recombination events that change local tree topology.

For $n = 8$, (9) and (10) give $\mathbb{E}(R_a) \approx 2.59\rho$ and $\mathbb{E}(R_t) \approx 0.65\rho$, respectively; these functions are also plotted in Figure 11. In agreement with Hudson and Kaplan's finding regarding their lower bound, Figure 11 shows that in general both $\mathbb{E}(R_a)$ and $\mathbb{E}(R_t)$ are much larger than the mean of any of the lower bounds we consider here. Note that $\mathbb{E}(R_a)$ and $\mathbb{E}(R_t)$ do not depend on $\theta$, whereas the mean number of detected recombination events should increase as $\theta$ increases. We expect the mean number of detected recombination events to get closer to $\mathbb{E}(R_t)$ as $\theta$ increases.

In order to capture how what has been discussed in this subsection depends on $\theta$, we repeated the above study for $\theta = 40$ and $\rho = 5, 10, 20, 35, 45, 60, 70$. For all values of $\rho$, we obtained $R_r(S) \geq R_h(S)$. For $\rho = 10$, there was exactly one case where $R_r(S) < R_s(S)$; for all other values of $\rho$, we obtained $R_r(S) \geq R_s(S)$. Relative sharpness of the lower bounds for $\theta = 40$ are shown in Figure 12, while the mean number of detected recombination events are shown in Figure 13. In comparison to the plots shown in Figures 10 and 11, the points shown in Figures 12 and 13 are shifted upward, but general characteristics are the same as before. Note that the scale of the vertical axes has changed.

### 4.2 $R_o(S)$ versus $R_r(S)$

In the previous subsection, we compared $R_r(S)$ with the lower bounds proposed by Myers and Griffiths. As discussed in Subsection 3.5, $R_o(S)$ is always equal to the minimum number of recombination event, whereas $R_r(S)$ is in general only a lower bound, i.e. $R_r(S) \leq R_o(S) = R_{\min}(S)$ for all $S$. In this subsection, we wish to examine how often $R_r(S)$ actually is equal to the minimum number of recombination events.

To obtain the minimum number of recombination events, we used ordered trees in our program to analyse the simulated data from the previous subsection. We only considered the $\theta = 15$ case to save computation time. For each value of $\rho = 5, 10, 15, 20$ and 25, less

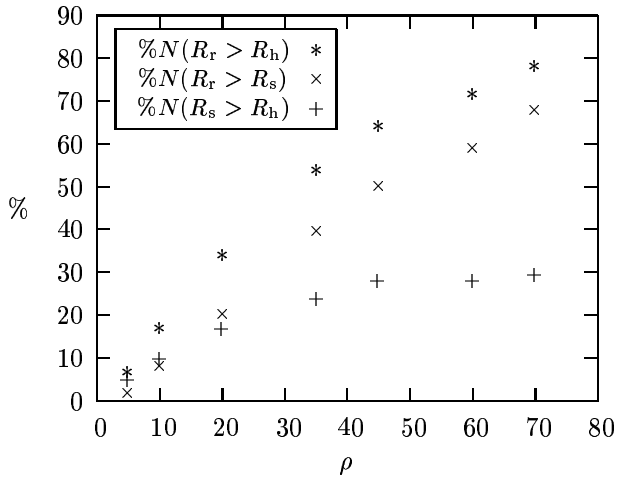


Figure 12: Relative sharpness of the lower bounds for $n = 8$ and $\theta = 40$. This figure should be compared with Figure 10.

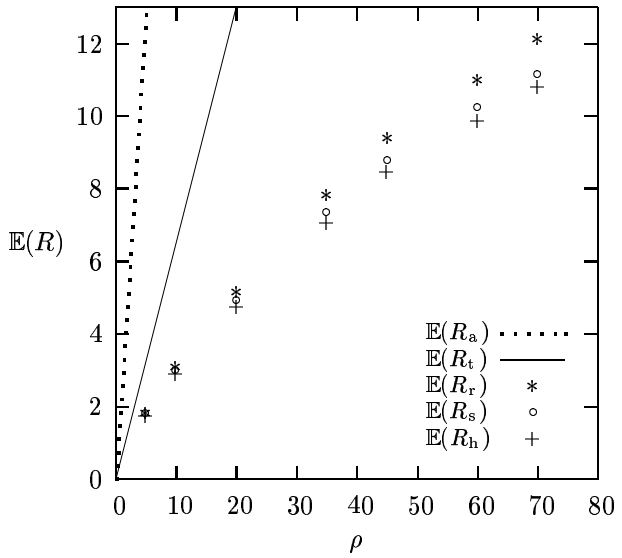


Figure 13: Mean numbers of detected recombination events for $n = 8$ and $\theta = 40$. This figure should be compared to Figure 11.

than or equal to 3 (out of 1000) simulated data actually had $R_{\rm r}(S) < R_{\rm o}(S)$—in terms of the notation introduced in the previous subsection, $N(R_{\rm r} < R_{\rm o}) \leq 3$. For other values of $\rho$ shown in Table 1, $N(R_{\rm r} < R_{\rm o})$ was between 6 and 10. Hence, this study indicates that, for the number of sequences and the mutation rate we used, our lower bound $R_{\rm r}(S)$ is very likely to give the minimum $R_{\rm min}(S)$.

A reason why in general so few cases have $R_{\rm r}(S) < R_{\rm o}(S)$ can be explained as follows. Let $\mathscr{P}^{\rm r}(S)$ (resp. $\mathscr{P}^{\rm o}(S)$) denote the complete set of minimal paths in the space of plain rooted (resp. ordered) trees corresponding to a particular data set $S$. For each path $P^{\rm r} = T^{\rm r}_{i_1}, T^{\rm r}_{i_2}, \ldots, T^{\rm r}_{i_\ell} \in \mathscr{P}^{\rm r}(S)$, we can associate a path $\varphi(P^{\rm r}) = T^{\rm o}_{i_1}, T^{\rm o}_{i_2}, \ldots, T^{\rm o}_{i_\ell}$ in the space of ordered trees such that the following two conditions hold:

(i) For all $1 \leq j \leq \ell$, $T^{\rm o}_{i_j}$ is equivalent to $T^{\rm r}_{i_j}$ as plain rooted trees.

(ii) The ordering of the internal vertices $T^{\mathrm{o}}_{i_j}$ are chosen so that the path length $L(\varphi(P^{\mathrm{r}})) := \sum_{k=1}^{\ell-1} d_{\mathrm{SPR}}(T^{\mathrm{o}}_{i_k}, T^{\mathrm{o}}_{i_{k+1}})$ is as small as it can be.

Note that $L(\varphi(P^{\mathrm{r}}))$ may not be equal to $L(P^{\mathrm{r}})$. If $R_{\mathrm{r}}(S) = R_{\mathrm{o}}(S)$, however, then there must exist a path $P^{\mathrm{r}} \in \mathscr{P}^{\mathrm{r}}(S)$ such that $L(P^{\mathrm{r}}) = L(\varphi(P^{\mathrm{r}}))$. As discussed in Subsection 3.4, the definition of an SPR operation for ordered trees is more restrictive than that for plain rooted trees. When there are only a small number of leaves in a tree, however, SPR operations usually involve subtrees with only few leaves. Furthermore, as a 1-leaved subtree contains no internal vertices, there is no restriction on where a pruned 1-leaved subtree can be regrafted back onto the remaining part of the tree. When the total number of leaves $n$ is small, the property just described often allows one to find a minimal path $P^{\mathrm{r}} \in \mathscr{P}^{\mathrm{r}}(S)$ such that $L(P^{\mathrm{r}}) = L(\varphi(P^{\mathrm{r}}))$ and $\varphi(P^{\mathrm{r}}) \in \mathscr{P}^{\mathrm{o}}(S)$. In general, as $n$ increases, the percentage $\%N(R_{\mathrm{r}} < R_{\mathrm{o}})$ is expected to increase as well.

### 4.3 Recovery of local trees

As mentioned in Section 2, the objective of our work is twofold: to find the minimum number $R_{\min}(S)$ of recombination events and to construct possible evolutionary histories with exactly $R_{\min}(S)$ recombination events (i.e. minimal ARGs). The algorithm described in Subsection 3.5 actually solves these two problems simultaneously. Namely, the quantity shown in (5) gives $R_{\min}(S)$ and minimal ARGs can be obtained from minimal paths; more exactly, trees which arise in a minimal path can be put together to construct a set of minimal ARGs. As discussed in Subsection 4.1, however, one should note that $R_{\min}(S)$ is, in fact, likely to be much less than the number of recombination events which occurred in the actual evolutionary history of the sequences in $S$. Hence, the ARG describing the actual evolution of sample sequences is most likely to be very different from the minimal ARGs one finds using our method. Nevertheless, it is still possible that the trees embedded in a minimal ARG, i.e. the trees in a minimal path, can capture reasonably well how sample sequences are actually related at different character sites. In this subsection, we examine how well our method can recover such local evolutionary relationships.

The program `ms` can include in its output a set of weighted ordered trees which together describe the simulated evolution of the entire locus. We therefore have a complete information about which tree actually describes the evolution of each polymorphic site in the simulated data. The trees we use in our method are unweighted, i.e. their branch lengths are not specified. We therefore ignore branch lengths when we compare local evolutionary relationships. Furthermore, we shall ignore the position of the root and only compare the sets of bipartitions defined by tree topologies.

The leaves of an $n$-leaved tree are bijectively labelled by a finite set $\mathcal{L}$ of $n$ elements. Cutting an edge in a tree $T$ decomposes $T$ into two connected components, one containing the leaves labelled by $X \subsetneq \mathcal{L}$ and the other the leaves labelled by $\mathcal{L} \setminus X$. Hence, to each edge in a tree, one can associate a bipartition of the label set $\mathcal{L}$ into two proper subsets. Let $\mathcal{B}(T)$ denote the set of all inequivalent bipartitions defined by the edges in $T$. For an $n$-leaved binary tree, there are $2n-3$ inequivalent bipartitions. We define the *bipartition match score* between two $n$-leaved binary trees $T$ and $T'$ as

$$\beta(T, T') = \frac{|\mathcal{B}(T) \cap \mathcal{B}(T')| - n}{n-3}. \tag{11}$$

Here, $|\mathcal{B}(T) \cap \mathcal{B}(T')|$ gives the number of identical bipartitions in $\mathcal{B}(T)$ and $\mathcal{B}(T')$, whereas "$-n$" subtracts from it the $n$ always-occurring bipartitions, each consisting of a single leaf $\{l\}$ and its complement $\mathcal{L} \setminus \{l\}$. Note that (11) is normalised so that $\beta(T, T') = 1$ if and only if $\mathcal{B}(T) = \mathcal{B}(T')$ and $\beta(T, T') < 1$ otherwise.

For a general data set $S$, there can be many minimal paths—in some cases as many as a few millions—and therefore it may not be practical to store all such paths in memory, although enumerating them can be done without much difficulty. Hence, we have written our program `RecMinPath` so that it randomly chooses only one minimal path for each $T \in W_\ell$ that satisfies $f_\ell(T) = \min_{T' \in W_\ell} f_\ell(T')$.

Suppose we perform $r$ number of simulations to generate $r$ independent sets of samples $S_1, S_2, \ldots, S_r$. In what follows, for each $a \in \{1, 2, \ldots, r\}$, $\ell_a$ denotes the number of polymorphic sites in the sample $S_a$ and the actual local trees describing the evolution of the polymorphic sites are denoted by $t^{\mathtt{ms}}_{a,1}, t^{\mathtt{ms}}_{a,2}, \ldots, t^{\mathtt{ms}}_{a,\ell_a}$. We use $k_a$ to denote the number of minimal paths reported in the output of `RecMinPath`. For every $i \in \{1, 2, \ldots, k_a\}$, a minimal path we find is of the form $P_{a,i} = T_{a,i,1}, T_{a,i,2}, \ldots, T_{a,i,\ell_a}$. Now, we define the *average local tree recovery percentage* for our method as

$$\overline{\beta}_{\text{RMP}} := \frac{1}{r} \sum_{a=1}^{r} \left\{ \frac{1}{k_a} \sum_{i=1}^{k_a} \left[ \frac{1}{\ell_a} \sum_{j=1}^{\ell_a} \beta(T_{a,i,j}, t^{\mathtt{ms}}_{a,j}) \right] \right\} \times 100\%, \tag{12}$$

where $\beta(T_{a,i,j}, t^{\mathtt{ms}}_{a,j})$ is defined in (11). For the results reported in this subsection, plain rooted trees were used in `RecMinPath`.

If one does not take recombination events into account and tries to describe the evolution of all polymorphic sites by a single tree, then the corresponding average local tree recovery percentage is defined as

$$\overline{\beta}_{\text{ST}} := \frac{1}{r} \sum_{a=1}^{r} \left[ \frac{1}{\ell_a} \sum_{j=1}^{\ell_a} \beta(\tau_a, t^{\mathtt{ms}}_{a,j}) \right] \times 100\%,$$

where $\tau_a$ is the single optimal tree used to describe the evolution of the sequences in $S_a$. In what follows, we compare our method to the single tree method based on parsimony. The program `RecPars`, originally written by Jotun Hein (Hein, 1993) and later rewritten by Kim Fisker, was used to obtain the most parsimonious tree for each data set.

For $n = 7$, we used $\theta = 15, 25, 50, 75$ and $\rho = 10, 20, 50$ in our simulations. We fixed `nsites`=3000 for the finite-sites recombination model. For each pair of $\theta$ and $\rho$ values, we generated 1000 sets of simulated data, each with $R_{\min}(S) > 0$. Just to get an idea of how $\overline{\beta}_{\text{RMP}}$ and $\overline{\beta}_{\text{ST}}$ may depend on the number of sequences, we performed a similar set of tests for $n = 8$. Numerical values of $\overline{\beta}_{\text{RMP}}$ and $\overline{\beta}_{\text{ST}}$ for these simulations are shown in Table 3, and, for ease of comparison, these results are illustrated in Figure 14. We clearly see that $\overline{\beta}_{\text{RMP}} > \overline{\beta}_{\text{ST}}$ for every pair of $\theta$ and $\rho$. Moreover, the gain in $\overline{\beta}_{\text{RMP}}$ as $\theta$ increases is clearly larger than that in $\overline{\beta}_{\text{ST}}$. As expected, for fixed $\theta$ and $\rho$, both $\overline{\beta}_{\text{RMP}}$ and $\overline{\beta}_{\text{ST}}$ tend to decrease as $n$ increases from 7 to 8.

Suppose we have $r$ simulated data sets each with exactly $\ell$ polymorphic sites. Then, $\overline{\beta}_{\text{RMP}}$ defined in (12) can be written as $\overline{\beta}_{\text{RMP}} = \frac{1}{\ell} \sum_{j=1}^{\ell} \overline{\beta}_{\text{RMP},j}$, where

$$\overline{\beta}_{\text{RMP},j} := \frac{1}{r} \sum_{a=1}^{r} \left[ \frac{1}{k_a} \sum_{i=1}^{k_a} \beta(T_{a,i,j}, t^{\mathtt{ms}}_{a,j}) \right] \times 100\%.$$

Table 3: Average local tree recovery percentages.

| | | $\rho = 10$ | | $\rho = 20$ | | $\rho = 50$ | |
|---|---|---|---|---|---|---|---|
| $n$ | $\theta$ | $\overline{\beta}_{\mathrm{RMP}}$ | $\overline{\beta}_{\mathrm{ST}}$ | $\overline{\beta}_{\mathrm{RMP}}$ | $\overline{\beta}_{\mathrm{ST}}$ | $\overline{\beta}_{\mathrm{RMP}}$ | $\overline{\beta}_{\mathrm{ST}}$ |
| 7 | 15 | 74 | 60 | 68 | 50 | 58 | 41 |
| | 25 | 78 | 61 | 72 | 52 | 63 | 43 |
| | 50 | 84 | 64 | 79 | 54 | 70 | 43 |
| | 75 | 87 | 65 | 81 | 56 | 74 | 44 |
| 8 | 15 | 73 | 57 | 65 | 47 | 55 | 38 |
| | 25 | 77 | 61 | 69 | 48 | 59 | 38 |
| | 50 | 82 | 62 | 75 | 52 | 65 | 39 |
| | 75 | 87 | 65 | 81 | 54 | 68 | 39 |

Note — This is a comparison between our method and the method which finds the single most parsimonious tree. The notations $\overline{\beta}_{\mathrm{RMP}}$ and $\overline{\beta}_{\mathrm{ST}}$ are defined in the text.

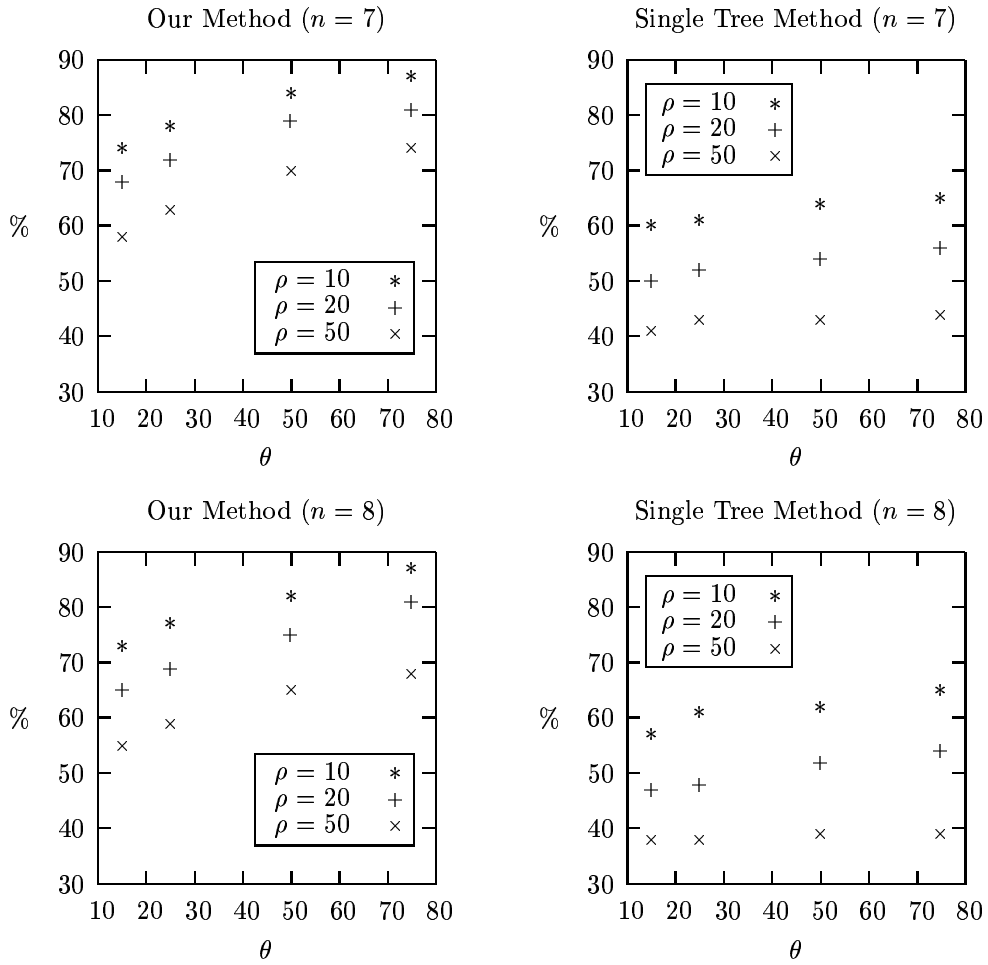


Figure 14: A comparison of average local tree recovery percentages.

Whereas $\overline{\beta}_{\mathrm{RMP}}$ reflects an average local tree recovery percentage first averaged over the entire locus, $\overline{\beta}_{\mathrm{RMP},j}$ gives an average local tree recovery percentage per site. To investigate how $\overline{\beta}_{\mathrm{RMP},j}$ behaves, we first need to decide on the fixed length $\ell$, for which we use the

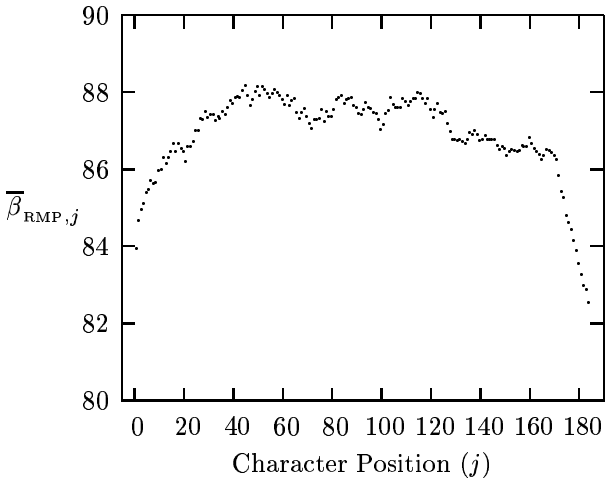


Figure 15: Average local tree recovery percentage per site. This result is based on 2000 simulations, each with 184 segregating sites. The parameters we used are $n = 7$, $\rho = 10$, $\theta = 75$ and `nsites=3000`.

formula $\mathbb{E}(s) = \theta \sum_{j=1}^{n-1} 1/j$ for the expected number of segregating sites (Hudson, 1983). For $n = 7$ and $\theta = 75$, $\mathbb{E}(s) \approx 184$.

Using $n = 7$, $\theta = 75$, $\rho = 10$ and `nsites=3000`, we ran the program `ms` until 2000 simulated data sets each with 184 polymorphic sites were generated. The result of our study of $\overline{\beta}_{\mathrm{RMP},j}$ is shown in Figure 15. A notable feature of this figure is the arch-like shape of the plot, indicating that the recovery percentage is higher around the middle than near the ends of the sequence. We can offer two related plausible explanations of this phenomenon.

First of all, note that recovering evolutionary relationships, or tree topology, is not possible without mutation events that generate polymorphism in the data. Furthermore, to be able to recover all bipartitions associated to a tree, we need at least one informative polymorphic site for every internal edge. Now, near the left (resp. right) end of the sequence, neighbouring informative polymorphic sites are concentrated only on the right (resp. left) hand side. In contract, at a position towards the middle of the sequence, neighbouring informative polymorphic sites appear both to its left and right. Secondly, dynamic programming itself can be partially responsible. In the dynamic programming algorithm described in Subsection 3.5, there is a forward propagation of information because the $(i+1)^{\mathrm{th}}$ column is analysed by looking at the information from the $i^{\mathrm{th}}$ column, and there is a backward propagation of information because whether a tree in $W_i^{\mathrm{o}}$ appears in a minimal path depends on what happens in the $(i+1)^{\mathrm{th}}$ column. Hence, near the ends of the sequence, there is not as much information from dynamic programming as in the middle of the sequence. An interplay between the two factors mentioned above might explain the arch-like shape shown in Figure 15.

### 4.4 Kreitman's data

In this subsection, we apply our method to study a classic biological data set. This particular example was used at the WABI 2003 conference as well (Song and Hein, 2003). We

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Wa-S | = | 000 | 000001100000 | 0 | 00110111011111000000 | 0 | 0000000 |
| Fl-1S | = | 001 | 000000000000 | 0 | 00110111011111000000 | 0 | 0000000 |
| Af-S | = | 000 | 000000000000 | 0 | 00000000000000000000 | 1 | 0000101 |
| Fr-S | = | 000 | 000000000000 | 0 | 11000000000000000000 | 1 | 0011000 |
| Fl-2S | = | 000 | 110001011001 | 1 | 11000000000000000000 | 0 | 1000000 |
| Ja-S | = | 001 | 000000000000 | 1 | 00000000000000010101 | 1 | 1000010 |
| Fl-F | = | 001 | 000000000000 | 1 | 00000000000000111111 | 0 | 1000000 |
| Fr-F | = | 111 | 110001011100 | 1 | 00000000000000111111 | 0 | 1100000 |
| Wa-F | = | 111 | 110001011100 | 1 | 00000000000000111111 | 0 | 1100000 |
| Af-F | = | 111 | 110001011100 | 1 | 00000000000000111111 | 0 | 1100000 |
| Ja-F | = | 111 | 111111000010 | 1 | 00001000100000111111 | 0 | 1000000 |

Figure 16: Kreitman's data in binary form. Also shown is a minimal haplotype block structure we found. There are 6 blocks, whose boundaries are indicated by vertical solid lines.

consider Kreitman's 1983 data which consist of 11 alleles of the alcohol dehydrogenase locus of *Drosophila melanogaster* (Kreitman, 1983). The alleles were sampled from 5 geographically distinct populations; they were taken from Washington (Wa), Florida (Fl), Africa (Af), France (Fr) and Japan (Ja). Kreitman's work was the first to study genetic variation in alleles obtained from nature. The aligned sequence length is 2800 base-pairs, of which, ignoring insertions and deletions, 43 sites are polymorphic. We have transformed the polymorphism data into binary sequences as shown in Figure 16. Note that Fr-F, Wa-F and Af-F are identical; there are 9 distinct sequences in the reduced data (c.f. Subsection 3.7). Hence, we relabel the sequences as follows:

$$
\begin{array}{lll}
s_1 := \text{Wa-S} & s_2 := \text{Fl-1S} & s_3 := \text{Af-S} \\
s_4 := \text{Fr-S} & s_5 := \text{Fl-2S} & s_6 := \text{Ja-S} \\
s_7 := \text{Fl-F} & s_8 := \text{Fr-F} = \text{Wa-F} = \text{Af-F} & s_9 := \text{Ja-F}
\end{array}
$$

As discussed in Subsection 3.5, we performed our analysis on $S = \{s_1, \ldots, s_9\}$ first using plain rooted trees, obtaining $R_{\mathrm{r}}(S) = 7$. We then checked that it is indeed possible to construct an ARG with exactly 7 recombination events. A minimal ARG constructed from combining the trees arising in a minimal path is shown in Figure 17. Recombination breakpoints are shown in the caption of the figure. As always, our convention is that when a recombination event occurs, the part to the left (resp. right) of the breakpoint gets descended from the left (resp. right) edge. Note that 2 recombination events occur between character columns 3 and 4, as well as between character columns 35 and 36. Positions $3, 4, 15, 16, 17, 35, 36, 37$ in $S$ correspond to positions $63, 170, 847, 950, 1030, 1691, 1730, 1827$, respectively, in the actual data.

As mentioned in Subsection 3.9, we can ask what the minimum number of haplotype blocks that can be defined by a minimal path is. For Kreitman's data, the answer is 6. We remind the reader that we define a block as consecutive positions in the sequence where the same tree is supported. A haplotype block structure with 6 blocks is shown in Figure 16, where block boundaries are indicated by vertical solid lines. For there to be only 6 blocks, only the second boundary is allowed to vary; all other block boundaries shown in Figure 16 must be fixed. The second block boundary occurs between character columns 15 and 16.

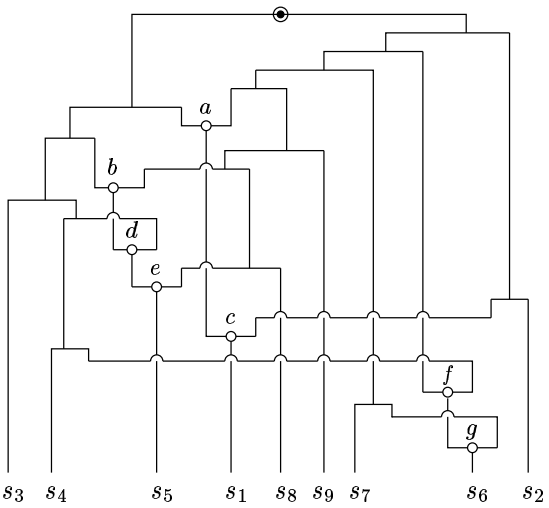


Figure 17: A minimal ancestral recombination graph for Kreitman's data. Recombination vertices are denoted by $\circ$ and their corresponding breakpoints are as follows. $a$: $(3, 4)$, $b$: $(3, 4)$, $c$: $(15, 16)$, $d$: $(16, 17)$, $e$: $(35, 36)$, $f$: $(35, 36)$, $g$: $(36, 37)$.

This block boundary can shift to left up to 6 columns. This freedom is not surprising at all since every character column between 10 and 15 is compatible with column 16.

In regard of detecting recombination events in Kreitman's data, we can compare our method with other currently-existing methods. Hudson and Kaplan's algorithm described in (Hudson and Kaplan, 1985) gives 5 as a lower bound on the number of recombination events. If one uses the method developed by Myers and Griffiths (2003) one would obtain 6 as both the haplotype bound $R_{\mathrm{h}}(S)$ and the bound $R_{\mathrm{s}}(S)$ based on simulation of sample history. In (Song and Hein, 2004), the present authors analysed Kreitman's data using a method based on set theory and obtained 7 as a lower bound. Note that none of these alternative methods can explicitly construct possible evolutionary histories consistent with the data.

# 5 Possible extensions

In this section, we describe some possible extensions of our work. It should be easy to modify our source code for `RecMinPath` to incorporate the first two extensions.

## 5.1 Missing data

In our method, working with missing data is straightforward. The reader should refer back to Subsection 3.5 for cross reference. Suppose that a character column $\mathbf{c}_i$ has some missing characters, i.e. it is not known whether $c_i^\alpha$ is 0 or 1 for some $\alpha \in \{1, 2, \ldots, n\}$—where $n$ denotes the total number of sequences present in the data set $S$. Let $\tilde{\mathbf{c}}_i$ consist of the characters in $\mathbf{c}_i$ that are known. Then, in the algorithm, for each character column $\mathbf{c}_i$ with missing characters, use the following set of trees:

$$M_i^{\mathrm{o}} = \{T \in \mathscr{T}_n^{\mathrm{o}} \,|\, T \text{ is compatible with } \tilde{\mathbf{c}}_i\}\,.$$

For character columns $\mathbf{c}_j$ without any missing character, use $W_j^{\text{o}}$ as usual. We can then carry out the rest of the algorithm without any further change. Note that the only practical change in the algorithm is that the size $|M_i^{\text{o}}|$ is larger than what $|W_i^{\text{o}}|$ would be if the missing characters in $\mathbf{c}_i$ were actually known.

### 5.2 Finite-sites model of mutation

As in the original algorithm for finding the most parsimonious history (Hein, 1990; Hein, 1993), the dynamic programming idea can be extended to consider back and recurrent mutations. In the case of the finite-sites model of mutation, one has to include in the dynamic programming algorithm mutation cost $m(i, T)$, which could be defined in terms of the minimum number of mutation events required for the tree $T$ to represent the evolutionary history of character site $i$. More generally, a weighted mutation cost $w(i, T)$ can be used to reflect its relative weight with respect to the recombination cost. Note that the infinite-sites model is equivalent to setting $w(i, T) = 0$ if $T$ is compatible with $\mathbf{c}_i$ and $w(i, T) = \infty$ otherwise.

The algorithm for the finite-sites model of mutation can be stated as constructing a weighted graph $G$ as follows.

1. Introduce $\ell$ clusters, with the $i^{\text{th}}$ cluster containing $|\mathscr{T}_n^{\text{o}}|$ vertices labelled by the trees in $\mathscr{T}_n^{\text{o}}$. Denote the $i^{\text{th}}$ cluster by $A_i^{\text{o}}$. (The letter "$A$" stands for "all" trees.)
   (a) For all $T \in A_1^{\text{o}}$, let $f_1(T) = w(1, T)$.
   (b) For all $1 \leq i < \ell$, recursively determine
   $$f_{i+1}(T_a) = w(i+1, T_a) + \min_{T_b \in A_i^{\text{o}}} [f_i(T_b) + d_{\text{rec}}(T_b, T_a)]$$
   for every tree $T_a \in A_{i+1}^{\text{o}}$.
   (c) In the weighted graph $G$, vertices $T_a \in A_{i+1}^{\text{o}}$ and $T_b \in A_i^{\text{o}}$ are joined by an edge if $f_{i+1}(T_a) - f_i(T_b) = d_{\text{rec}}(T_a, T_b) + w(i+1, T_a)$, and the weight of the edge is $d_{\text{rec}}(T_a, T_b) + w(i+1, T_a)$.
2. The number defined as
   $$C_{\text{o}}(S) = \min_{T_a \in A_\ell^{\text{o}}} f_\ell(T_a)$$
   gives the minimum cost. A minimal path in $G$ is a connected path from any tree $T_a \in A_1^{\text{o}}$ to a tree $T_b \in A_\ell^{\text{o}}$ with $f_\ell(T_b) = C_{\text{o}}(S)$.

As we have done for the infinite-sites model of mutation, we can use plain rooted trees to obtain a lower bound. If plain rooted trees are used in the above algorithm, then the minimum cost would be defined as $C_{\text{r}}(S) = \min_{T_a \in A_\ell^{\text{r}}} f_\ell(T_a)$. In general, $C_{\text{r}}(S) \leq C_{\text{o}}(S)$. A comparison between the number $|W_i^{\text{o}}|$ of trees compatible with a column $\mathbf{c}_i$ and the total number of $|\mathscr{T}_n^{\text{o}}|$ of $n$-leaved trees can be found in (Song and Hein, 2003). Since most of $|W_i^{\text{o}}|$ are only very small fractions of $|\mathscr{T}_n^{\text{o}}|$, using the finite-sites model of mutation is significantly more computationally intensive than using the infinite-sites model.

Note that the idea of using adjacency-sets, as discussed in Subsection 3.6, can be applied to the present case as well. Firstly, let $f_{1,0}(T) = w(1, T)$ for all $T \in A_1^{\text{o}}$. Then, instead of (7) and (8), we just need to perform the following steps recursively for all $1 \leq i < \ell$:

For all $1 \leq r < d_n$ and $T_a \in A_i^{\circ}$, find

$$f_{i,r}(T_a) = \min_{T_b \in N_1(\{T_a\})} [f_{i,r-1}(T_b) + 1 - \delta_{a,b}],$$

and, for all $T_a \in A_{i+1}^{\circ}$, find

$$f_{i+1,0}(T_a) = w(i+1, T_a) + \min_{T_b \in N_1(\{T_a\})} [f_{i,d_n-1}(T_b) + 1 - \delta_{a,b}].$$

### 5.3 Combining with Myers & Griffiths's algorithm

One limitation of our method is that, as it stands, it becomes infeasible to analyse more than 9 sequences in the reduced data; when there are many trees, it takes an inordinate amount of memory to store the adjacency-sets. As mentioned before, the algorithm proposed by Myers and Griffiths (2003) uses local lower bounds for small regions to construct a global bound for the entire data. Hence, for more than 9 sequences, we can try the following. If we focus on small regions, there may not be so many distinct sequences in the reduced data, and therefore we can use our algorithm to compute exact local bounds, which can then be used in Myers and Griffiths's integer linear programming algorithm to find a global bound. Combining the two methods as just described should perform quite well.

## 6 Conclusion

As well as finding the minimum number of recombination events which must have occurred in the evolutionary history of sampled sequences, our algorithm can be used to construct minimal evolutionary histories that are consistent with the data. The approach we take is to view recombination events as inducing SPR operations on local trees. If the right kind of trees are used, the SPR-distance between two such trees correctly encodes the number of recombination events. The method introduced in (Song and Hein, 2003) for computing the SPR-distance between trees has allowed us to overcome some difficulties which have hitherto prevented an exact implementation of the dynamic programming idea.

When there are not too many sequences in the reduced data so that our method can be applied, we have shown that our lower bound $R_{\mathrm{r}}(S)$ is an improvement on the haplotype bound $R_{\mathrm{h}}(S)$ and the bound $R_{\mathrm{s}}(S)$ from simulation of sample history (Myers and Griffiths, 2003). As discussed in Subsection 4.3, however, the minimum $R_{\min}(S)$ is likely to be much less than the number of recombination events which occurred in the actual evolutionary history. Hence, merely counting the number of detectable recombination events does not seem to be sufficient for inferring what actually happened in the evolutionary history. Nevertheless, we have shown in this paper that local evolutionary relationships can be recovered more accurately. In other words, the average local tree recovery percentage $\overline{\beta}_{\mathrm{RMP}}$ based on our method is much larger than the percentage of $\mathbb{E}(R_{\min})$ relative to $\mathbb{E}(R_{\mathrm{a}})$.

In (Wang *et al.*, 2001), it was shown that, under the infinite-sites model of mutation, the problem of constructing possible evolutionary histories with the minimum number of recombination events is NP-hard. Moreover, the authors of that paper considered a restricted version of the problem, investigating which data are representable by an ARG where all recombination cycles are node-disjoint. This restricted problem was recently completely

solved by Gusfield *et al.* (2004), who constructed a polynomial-time algorithm for obtaining such a restricted ARG when it exists. Not surprisingly, no polynomial-time algorithm is known for the general case. Insightful results from (Wang *et al.*, 2001) and (Gusfield *et al.*, 2004) may prove useful for devising an algorithm for the general case that is more efficient than the one presented in the present paper.

In recent years, several fairly strong evidences have emerged supporting that gene conversion may play an important role in human evolutionary genetics (Ardlie *et al.*, 2001; Jeffreys and May, 2004). In light of these important findings, it would be interesting to generalize our method to study gene conversion, possibly in conjunction with the study of single cross-over recombination considered in this paper. We believe that much of what we laid out in this paper can be carried over to that case.

## Acknowledgments

We thank S. Myers for useful discussions and the Oxford Supercomputing Centre for allowing us to use their CPU time. This research is supported by EPSRC under grant HAMJW and by MRC under grant HAMKA. Y.S.S. is partially supported by a grant from the Danish Natural Science Foundation (SNF-5503-13370).

## References

Ardlie, K., Liu-Cordero, S.N., Eberle, M.A., Daly, M., Barrett, J., Winchester, E., Lander, E.S., and Kruglyak, L. 2001. Lower-than-expected linkage disequilibrium between tightly linked markers in human suggests a role for gene conversion. *Am. J. Hum. Genet.* 69, 582–589.

Daly, M.J., Rioux, J.D., Schaffner, S.F., Hudson, T.J., and Lander, E.S. 2001. High-resolution haplotype structure in the human genome. *Nat. Genet.* 29, 229–232.

Gabriel, S.B., Schaffner, S.F., Nguyen, H., Moore, J.M., Roy, J., Blumenstiel, B., Higgins, J., DeFelice, M., Lochner, A., Faggart, M., Liu-Cordero, S.N., Rotimi, C., Adeyemo, A., Cooper, R., Ward, R., Lander, E.S., Daly, M.J., and Altshuler, D. 2002. The structure of haplotype blocks in the human genome. *Science* 296, 2225–2229.

Gusfield, D., Eddhu, S., and Langley, C. 2004. Optimal, efficient reconstruction of phylogenetic networks with constrained recombination. *J. Bioinf. Comp. Biol.* 2, 173–213.

Griffiths, R.C. and Marjoram, P. 1997. An ancestral recombination graph, 257–270. In Donnelly, P. and Tavaré, S., eds., *Progress in Population Genetics and Human Evolution. IMA Volumes in Mathematics and its Applications 87,* Springer-Verlag, Berlin.

Jeffreys, A.J. and May, C.A. 2004. Intense and highly localized gene conversion activity in human meiotic crossover hot spots. *Nat. Genet.* 36, 151–156.

Johnson, G.C., Esposito, L., Barratt, B.J., Smith, A.N., Heward, J., Di Genova, G., Ueda, H., Cordell, H.J., Eaves, I.A., Dudbridge, F., Twells, R.C., Payne, F., Hughes, W., Nutland, S., Stevens, H., Carr, P., Tuomilehto-Wolf, E., Tuomilehto, J., Gough, S.C.,

Clayton, D.G., and Todd, J.A. 2001. Haplotype tagging for the identification of common disease genes. *Nat. Genet.* 29, 233–237.

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

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

Hudson, R.R. 1983. Properties of a neutral allele model with intragenic recombination. *Theor. Pop. Biol.* 23, 183–201.

Hudson, R.R. 2002. Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics* 18, 337–338.

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

Kreitman, M. 1983. Nucleotide polymorphism at the alcohol dehydrogenase locus of Drosophila melanogaster. *Nature* 304, 412–417.

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

Song, Y.S. 2003. On the combinatorics of rooted binary phylogenetic trees. *Annals of Combinatorics* 7, 365–379.

Song, Y.S. and Hein, J. 2003. Parsimonious reconstruction of sequence evolution and haplotype blocks: Finding the minimum number of recombination events, 287–302. In Benson, G. and Page, R., eds., *Algorithms in Bioinformatics. Lecture Notes in Bioinformatics,* Springer-Verlag, Berlin.

Song, Y.S. and Hein, J. 2004. On the minimum number of recombination events in the evolutionary history of DNA sequences. *J. Math. Biol.* 48, 160–186.

Swofford, D.L. and Olsen, G.J. 1990. Phylogeny reconstruction, 411–501. In Hillis, D.M., Moritz, C., and Mable, B.K., eds., *Molecular Systematics,* Sinauer Associates, Massachusetts.

Wang, L., Zhang, K., and Zhang, L. 2001. Perfect phylogenetic networks with recombination. *J. Comp. Biol.* 8, 69–78.