Parsimonious Reconstruction of Sequence Evolution and Haplotype Blocks:
Finding the Minimum Number of Recombination Events

Yun S. Song* and Jotun Hein

Department of Statistics, University of Oxford,
1 South Parks Road, Oxford, OX1 3TG, UK
song@stats.ox.ac.uk, hein@stats.ox.ac.uk

Abstract. Under the infinite-sites model of mutation, we consider the problem of finding the minimum number of recombination events which must have occurred in the evolutionary history of sampled DNA sequences. Our approach is deterministic and is based on the combinatorics of leaf-labelled rooted trees. In contrast to previously known approaches, which only yield estimated lower bounds, our approach always gives the exact minimum number of recombination events. Furthermore, our method can be used to reconstruct explicitly evolutionary histories with the minimum number of recombination events. As an additional application, we discuss how our work can be used to define haplotype blocks.

1 Introduction

As well as being the major biological process which can destroy linkage disequilibrium (LD), recombination is one of the principal driving forces which generate genetic variations between different individuals of the same population. As such, recombination can have far-reaching consequences on molecular evolution. Knowing how many and where in the sequence recombination events have occurred can thus be a major contributing factor in unravelling many important questions in genetics. Some recombination events do not change the local phylogeny, however, and therefore it is in general impossible to know exactly how many recombination events have occurred in the evolutionary history of sampled sequences. Nevertheless, within the framework of a chosen model, it is meaningful to ask at least how many recombination events must have occurred in the history.

In [6], Hudson and Kaplan considered, within the framework of the infinite-sites model, the problem of finding a lower bound on the number of recombination events which must have occurred in the history of sampled DNA sequences. With the advent of new technologies which allow us to obtain data at an alarming rate, that particular problem of counting recombination events is currently

* Corresponding Author

receiving a renewed interest. Myers and Griffiths recently proposed an integer linear programming approach for constructing new lower bounds on the number of recombination events [9], whereas the present authors proposed a set theoretical method to define a new lower bound [12]. Neither of these methods can explicitly reconstruct evolutionary histories. Moreover, a common thread that runs through all currently-existing methods is that, for some data set S, the computed lower bound on the number of recombination events may, in fact, be less than the minimum number $\mathcal{R}_{\min}(S)$. Here, the minimum number $\mathcal{R}_{\min}(S)$ is defined by the property that there exists no evolutionary history with less than $\mathcal{R}_{\min}(S)$ recombination events that can generate S under the infinite-sites model. The goal of our present work is thus twofold; to find the exact minimum number $\mathcal{R}_{\min}(S)$ of recombination events and to reconstruct possible evolutionary histories with exactly $\mathcal{R}_{\min}(S)$ recombination events.

The main idea which underlies our algorithm was first laid out by Hein more than a decade ago [4], and a heuristic implementation of the idea was subsequently carried out [5]. An exact implementation of the idea, however, could not be carried out so far due to several difficulties, the major one being the complexity of the combinatorics involved. For example, as we later elaborate, working with trees with many restrictions and computing the distance between two arbitrary such trees can be very complicated. The approach we take in the present paper is to view recombination events as the so-called subtree-prune-and-regraft (SPR) operations on trees. More precisely, if an appropriate definition of a tree is used, the SPR-distance between two trees – defined as the minimum number of SPR operations required to transform one tree to the other – correctly encodes the number of recombination events.

Perhaps the most interesting recent finding in haplotype analysis is the discovery of the possible existence of haplotype block structures in the human genome [1, 7, 3], where a block is roughly characterised by the existence of high LD and limited haplotype diversity. In this paper, using our algorithm for detecting recombination events, we propose a new way of defining haplotype blocks. Unlike previous proposals, we explicitly take possible evolutionary histories into account.

Throughout this paper we assume that the data S consists of binary sequences; phased single nucleotide polymorphism (SNP) data satisfies this criterion, for instance. More precisely, we let $S = \{s_\alpha\}$ be a set of n binary sequences $s_\alpha = c_1^\alpha, c_2^\alpha, \ldots, c_\ell^\alpha$, where $c_i^\alpha \in \{0, 1\}$ for every $\alpha \in \{1, 2, \ldots, n\}$ and $i \in \{1, 2, \ldots, \ell\}$. Note that each sequence is of fixed length ℓ. The entry c_i^α is called the i^{th} *character* of sequence s_α, and $\mathbf{c}_i := (c_i^1, c_i^2, \ldots, c_i^n)$ the i^{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*. We 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 $c_i^\alpha = 0$. As mentioned before, we assume the infinite-sites model of mutation. That is, we assume that at most one mutation event has occurred at each character column.

The organisation of this paper is as follows. In §2 we discuss the combinatorics of rooted trees relevant to our work, as well as laying out some necessary

definitions. Our main ideas and algorithms are described in §3. In §4 we apply our method to analyse Kreitman's 1983 data of the alcohol dehydrogenase locus from 11 chromosomes of *Drosophila melanogaster* [8]. We conclude with some remarks in §5.

2 Trees

In this section, we present some definitions and facts to which we frequently refer in the main part of this paper. The reader is strongly recommended to browse through §2.1 to get familiarised with our notations.

2.1 Definitions

In this paper we consider leaf-labelled rooted binary trees whose branch lengths are not specified. The space of leaf-labelled rooted binary trees with n leaves is denoted by $\mathscr{T}_n^{\mathrm{r}}$. The degree of a vertex v is the number of edges which are incident with v. For $n \geq 2$, a tree in $\mathscr{T}_n^{\mathrm{r}}$ has n labelled degree-1 vertices called *leaves*; $n-2$ unlabelled degree-3 vertices; and a distinguished vertex of degree 2 called the *root*. A 1-leaved tree consists of a single labelled degree-0 vertex which serves as both the root and the leaf. A vertex which is not a leaf is called an *internal* vertex. The leaves of an n-leaved tree are bijectively labelled by a finite set S of n elements. In the remainder of this paper, when we say a tree without any qualification, we shall mean a leaf-labelled rooted binary tree.

A *path* from a vertex v_0 to another vertex v_k is an alternating sequence $v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k$ of vertices v_i and edges e_i, such that (1) e_i joins v_{i-1} and v_i, and (2) all e_i's and v_i's are distinct. In a rooted tree, time flows from the root to the leaves. We say that vertex $v \in T$ is a *descendant* of vertex $u \in T$ if there exists a path from u to v which goes strictly forward in time; u is called an *ancestor* of v. A *subtree* t of a tree $T \in \mathscr{T}_n^{\mathrm{r}}$ is a tree in $\mathscr{T}_{n'}^{\mathrm{r}}$, where $n' \leq n$, and is defined by the property that if a vertex $v \in T$ is contained in t, then so are all its descendants. We say that two vertices $u, v \in T$ are *adjacent* if there exists an edge which joins u and v. In this paper, a subtree whose root is adjacent to the root of T is called an R-subtree of T.

In a rooted binary tree, the set $\{v_1, v_2, \ldots, v_{n-2}\}$ of degree-3 vertices is a *partially* ordered set whose binary relation denoted $<$ is given by ancestral relation. More precisely, $v_i < v_j$ if v_i is a descendant of v_j. Note that, if r denotes the root of a tree T, then $v_i < r$, for all degree-3 vertices v_i of T. Two degree-3 vertices v_i and v_j are *incomparable* if v_i is not in the path to the root from v_j and vice versa.

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 a descendant 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 $i \neq j$. Two trees

equivalent as rooted trees are distinct as ordered trees if the ordering of their degree-3 vertices are different. In a subtree t of an ordered tree T, the ordering of degree-3 vertices in t is determined by their ordering in T. The space of ordered trees with n leaves is denoted by $\mathscr{T}_n^{\mathrm{o}}$.

Let $\{B_i, B_i^c\} = S$ denote the bipartition corresponding to an informative character column $\mathbf{c}_i$, such that s_α and s_β belong to the same subset if and only if $c_i^\alpha = c_i^\beta$. 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 containing the leaves labelled by B_i and the other the leaves labelled by B_i^c. If the column $\mathbf{c}_i$ is not informative, then every n-leaved tree is considered compatible with $\mathbf{c}_i$. In relation to biology, if a tree is compatible with a character column, then at most one mutation event at that column is necessary for the tree to represent the evolutionary history of sampled sequences at that character column, i.e. the tree is consistent with the character column under the infinite-sites model.

NOTE: When we write a symbol (for example, $\mathscr{T}_n$) without a qualifying superscript "r" or "o," it should be understood as referring to both cases.

2.2 SPR Operations

The precise definition of a subtree-prune-and-regraft (SPR) operation depends on the type of tree on which the operation is performed. In general, the more characteristics a tree has, the more restrictive an SPR operation has to be. We begin our discussion with plain leaf-labelled rooted trees. There are three kinds of SPR operations that can be performed on leaf-labelled rooted trees. An illustration of these operations is shown in Figure 1. In what follows, let T (resp. T') denote a tree before (resp. after) an SPR operation. The notation $T \setminus t$ denotes the part of T obtained from removing a subtree t and the edge incident with the root of t but not contained in t. In words the three SPR operations are as follows.

1. An edge e is cut to prune a non-R-subtree t, and t is regrafted onto a pre-existing edge in the remaining part $T \setminus t$ of T, thus creating a new degree-3 vertex. The vertex in $T \setminus t$ where e used to be incident gets removed. The root of T remains the root of T'. (In Figure 1, $T \to T_1$ is an example of this kind. The edge e_b is cut and then regrafted onto the edge e_a.)
2. Let s_1 and s_2 be the two R-subtrees of T, and let e_1 and e_2, respectively, be the edges which join their roots to the root of T. The edge e_1 is cut to prune s_1, and s_1 is regrafted onto a pre-existing edge in s_2. The edge e_2 gets removed and the degree-3 vertex in s_2 where e_2 used to be incident gets replaced by a degree-2 vertex, which becomes the root of T'. (In Figure 1, $T \to T_2$ is an example of this kind. The edge e_c can be cut and regrafted onto e_a. The root of the R-subtree containing t_1, t_2 and t_3 then becomes the root of T_2. In this example, note that T can be transformed into T_2 in several ways. More exactly, the subtree t_1 can be pruned and regrafted onto e_c or

the subtree containing t_2 and t_3 can be pruned and joined onto the root of T; these kinds of SPR operations have already been described above.)

3. An edge e is cut to prune a non-R-subtree t, and t is joined to the root of T. The root of T' is given by creating a new vertex of degree 2 on e. (In Figure 1, $T \to T_3$ is an example of this kind. The edge e_b is cut and then joined to the root of T. A new degree-2 vertex is created on the edge and it serves as the root of T_3.)

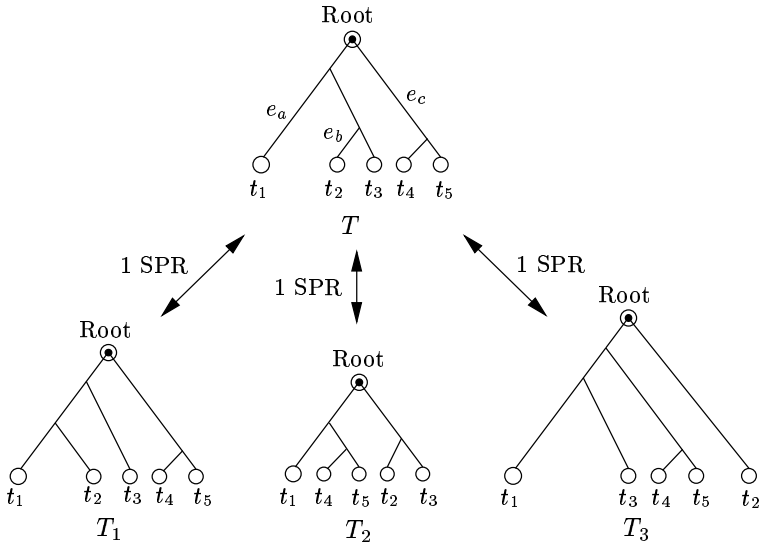


Fig. 1. An illustration of SPR operations. Big open circles ◯ labelled by t_j represent subtrees.

For ordered trees, we impose an additional restriction on the definition of SPR operations. Consider a subtree t of an ordered tree $T \in \mathscr{T}_n^{\mathrm{o}}$. An SPR operation of t which transform $T \in \mathscr{T}_n^{\mathrm{o}}$ to $T' \in \mathscr{T}_n^{\mathrm{o}}$ is defined to satisfy the following additional property: Let v (resp. v') denote the vertex in $T \setminus t$ (resp. $T' \setminus t$) which is adjacent to the root of t. Suppose v_i and v_j are two internal vertices other than v and v'. If $v_i < v_j$ before an SPR operation, then $v_i < v_j$ after the SPR operation, and vice versa; that is, a relation between any two internal vertices other than v and v' should be the same before and after an SPR operation.

2.3 Some Enumerations of Trees

It is well-known [10] that the number of inequivalent leaf-labelled rooted binary trees with n leaves is

$$\tau^{\mathrm{r}}(n) := |\mathscr{T}_n^{\mathrm{r}}| = (2n-3)!! = (2n-3) \times (2n-5) \times \cdots \times 3 \times 1 = \frac{(2n-2)!}{2^{n-1}(n-1)!},$$

whereas the number of inequivalent ordered trees with n leaves is

$$\tau^{\mathrm{o}}(n) := |\mathscr{T}_n^{\mathrm{o}}| = \prod_{m=2}^{n} \binom{m}{2} = \frac{n!(n-1)!}{2^{n-1}} .$$

Note that $\tau^{\mathrm{r}}(n) = (2n-3)\tau^{\mathrm{r}}(n-1)$, while $\tau^{\mathrm{o}}(n) = \frac{n(n-1)}{2}\tau^{\mathrm{o}}(n-1)$. The number $\tau^{\mathrm{o}}(n)$ of ordered trees thus grows much faster than the number $\tau^{\mathrm{r}}(n)$ of plain rooted trees.

The adjacency-set $U(T)$ of a tree T is defined as the set of all trees which are one SPR operation away from T. For rooted trees, the size of $U(T)$ depends on the topology of T. In [11], Song has constructed a closed-form formula for $|U(T)|$, where $T \in \mathscr{T}_n^{\mathrm{r}}$, and has obtained the following result:

Proposition 1. *For leaf-labelled rooted binary trees with n leaves, where $n \geq 3$, the minimum $\delta_{min}(n)$ and the maximum $\delta_{max}(n)$ values of $|U(T)|$ are given by*

$$\delta_{min}(n) = 3n^2 - 13n + 14 ,$$
$$\delta_{max}(n) = 4(n-2)^2 - 2\sum_{m=1}^{n-2} \lfloor \log_2(m+1) \rfloor ,$$

where $\lfloor \cdot \rfloor$ denotes the floor function.

No analogous formulae for ordered trees are known. On the left hand side of Table 1, numerical values of $\tau^{\mathrm{r}}(n), \tau^{\mathrm{o}}(n), \delta_{\min}(n)$ and $\delta_{\max}(n)$ are shown for low values of n. For ordered trees, the reported values of $\delta_{\min}(n)$ and $\delta_{\max}(n)$ have been determined via explicit computation in the implementation of our algorithm. Note that both $\tau^{\mathrm{r}}(n)$ and $\tau^{\mathrm{o}}(n)$ grow much faster than the size of adjacency-sets. The importance of this fact will become clear when we discuss our algorithm.

Also of interest to us is the number of trees which are compatible with a given character column. Let $\{B_i, B_i^c\}$ denote the bipartition of S corresponding to a character column $\mathbf{c}_i$, and suppose $|B_i| = k$ and $|B_i^c| = n-k$, where $1 \leq k \leq n-1$. Let $w^{\mathrm{r}}(n,k)$ (resp. $w^{\mathrm{o}}(n,k)$) denote the number of rooted trees (resp. ordered trees) compatible with the bipartition $\{B_i, B_i^c\}$. From [11] we have the following results:

Proposition 2 (Number of rooted trees compatible with a column). *For $n \geq 4$ and $1 \leq k \leq n-1$,*

$$w^{\mathrm{r}}(n,k) := (2n-3)\,\tau^{\mathrm{r}}(k)\,\tau^{\mathrm{r}}(n-k) .$$

Proposition 3 (Number of ordered trees compatible with a column). *For $n \geq 4$ and $1 \leq k \leq n-1$,*

$$w^{\mathrm{o}}(n,k) := \tau^{\mathrm{o}}(k)\tau^{\mathrm{o}}(n-k)\left[\binom{n}{k-1} + \binom{n}{n-k-1} - \binom{n-2}{k-1}\right] .$$

Numerical values of $w^{\mathrm{r}}(n,k)$ and $w^{\mathrm{o}}(n,k)$ are shown in Table 1 for $n \leq 9$.

Table 1. Numerical summary of tree enumerations. For ordered trees, $\delta_{\min}(n)$ and $\delta_{\max}(n)$ have been determined through explicit computation.

	Rooted Trees			Ordered Trees		
n	$\tau^{\mathrm{r}}(n)$	$\delta_{\min}$	$\delta_{\max}$	$\tau^{\mathrm{o}}(n)$	$\delta_{\min}$	$\delta_{\max}$
4	15	10	12	18	13	13
5	105	24	28	180	35	38
6	945	44	52	2,700	75	81
7	10,395	70	84	56,700	135	146
8	135,135	102	124	1,587,600	220	237
9	2,027,025	140	170	57,153,600	?	?
10	34,459,425	184	224	2,571,912,000	?	?

n	k	$w^{\mathrm{r}}(n,k)$	$w^{\mathrm{o}}(n,k)$
4	2	5	6
5	2, 3	21	36
6	2, 4	135	396
	3	81	216
7	2, 5	1,155	6,660
	3, 4	495	2,484
8	2, 6	12,285	156,600
	3, 5	4,095	44,820
	4	2,925	29,808
9	2, 7	155,925	4,876,200
	3, 6	42,525	1,142,100
	4, 5	23,625	567,000

3 Main Ideas

3.1 Recombination Events as SPR Operations

In a graphical representation of evolutionary history, if sampled sequences have been subjected to recombination, different character columns may be described by different trees. The so-called ancestral recombination graph (ARG) is constructed by putting together the set of trees supported at different character columns. For instance, while a column $\mathbf{c}_i$ is described by $T \in \mathscr{T}_n^{\mathrm{o}}$, the next column $\mathbf{c}_{i+1}$ may be described by a different tree $T' \in \mathscr{T}_n^{\mathrm{o}}$. The tree T can be transformed to the tree T' through SPR operations, and an ARG precisely contains this information; namely, recombinant sequences in the ARG correspond to the leaves in the subtree that gets pruned and regrafted. Furthermore, the number of SPR operations used to transform T to T' corresponds to the number of recombination events occurring between the columns $\mathbf{c}_i$ and $\mathbf{c}_{i+1}$.

3.2 SPR-Distance between Trees

For any pair of trees, say $T, T' \in \mathscr{T}_n$, their SPR-distance $d(T,T')$ 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. It is not very difficult, however, 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 $d(T,T')$ for arbitrary T and T'.

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

$$U(T) := \{T' \in \mathscr{T}_n \mid d(T,T') = 1\}\,.$$

Let $U_0(T) = \{T\}$ and $U_1(T) = U(T)$. Then, for $m \geq 2$, recursively define

$$U_m(T) := \left[\bigcup_{T' \in U_{m-1}(T)} U(T') \right] \setminus [U_{m-1}(T) \cup U_{m-2}(T)] .$$

The distance $d(T, T')$ between T and T' is the value of m which satisfies $T' \in U_m(T)$.

3.3 The Algorithm for Counting Recombination Events

1. To each character column $\mathbf{c}_i$ of S, associate a set W_i^{o} of trees defined as

$$W_i^{\mathrm{o}} := \{T \in \mathscr{T}_n^{\mathrm{o}} \,|\, T \text{ compatible with column } \mathbf{c}_i\} \subseteq \mathscr{T}_n^{\mathrm{o}} .$$

   Let $k(i)$ denote the size of B_i or B_i^c, with $\{B_i, B_i^c\}$ being the bipartition of S corresponding to the column $\mathbf{c}_i$. Then,

$$w_i^{\mathrm{o}} := |W_i^{\mathrm{o}}| = \begin{cases} w^{\mathrm{o}}(n, k(i)), & \text{if } k(i) \geq 2, \\ \tau^{\mathrm{o}}(n), & \text{otherwise,} \end{cases}$$

   where $w^{\mathrm{o}}(n, k)$ is defined in Proposition 3.
2. Construct a weighted graph G as follows. Introduce ℓ clusters, with the i^{th} cluster containing w_i^{o} vertices labelled by the trees in W_i^{o}.
   (a) For all $T \in W_1^{\mathrm{o}}$, let $f_1(T) = 0$.
   (b) For all $1 \leq i < \ell$, recursively determine

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

   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 $f_{i+1}(T_a) - f_i(T_b) = d(T_a, T_b)$, and the weight of the edge is $d(T_a, T_b)$.
3. The number defined as

$$\mathcal{R}_{\mathrm{o}}(S) = \min_{T_a \in W_\ell^{\mathrm{o}}} f_\ell(T_a) \tag{2}$$

   gives the minimum number $\mathcal{R}_{\min}(S)$ 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 $f_\ell(T_b) = \mathcal{R}_{\mathrm{o}}(S)$ is called a *minimal path* in G.

(**Remark:** Why ordered trees, not plain rooted trees, should be used in the above algorithm to obtain $\mathcal{R}_{\min}(S)$ is discussed in §3.6.)

The algorithm described above is a special case of the exact algorithm given in [5]; assuming the infinite-sites model has simplified the algorithm a bit. Because it was not known how the distance $d(T, T')$ for arbitrary ordered trees T and T' could be computed, Hein used unrooted trees in his implementation of

the algorithm [5]. Moreover, he 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. As computing the distance $d(T, T')$ for arbitrary $T, T' \in \mathscr{T}_n^o$ is rather non-trivial, step 2 in the above algorithm is computationally intensive. We will return to this point in §3.8, where we present a new, modified algorithm. For now, using the fact that $d(\cdot, \cdot)$ is a proper distance function on $\mathscr{T}_n^o$, we establish the following result, which allows us to simplify the algorithm further:

Proposition 4. *For all $T_a, T_b \in W_i^o$, where $i \geq 2$, the function f_i defined in (1) satisfies*

$$|f_i(T_a) - f_i(T_b)| \leq d(T_a, T_b)\,. \tag{3}$$

Proof: We shall prove (3) by induction. Since $f_1(T) = 0$ for all $T \in W_1^o$, we have $f_2(T') = \min_{T \in W_1^o}[d(T, T')]$, for all $T' \in W_2^o$. Let $T_a, T_b \in W_2^o$. Suppose $T_a, T_b \in W_1^o$. Then, $f_2(T_a) = 0 = f_2(T_b)$, and therefore (3) is satisfied. Suppose $T_a \notin W_1^o$ and $T_b \in W_1^o$. Then, $f_2(T_b) = 0$, whereas $f_2(T_a) = \min_{T \in W_1^o}[d(T, T_a)]$, which by definition is less than or equal to $d(T_b, T_a)$. Hence, (3) is again satisfied. Lastly, suppose $T_a \notin W_1^o$ and $T_b \notin W_1^o$, with $f_2(T_a) \geq f_2(T_b)$. Let $T_s \in W_1^o$ be a tree which satisfies $d(T_s, T_b) = \min_{T \in W_1^o}[d(T, T_b)]$. Then,

$$\begin{aligned} f_2(T_a) - f_2(T_b) \;=\; \min_{T \in W_1^o}[d(T, T_a)] - d(T_s, T_b) \;&\leq\; d(T_s, T_a) - d(T_s, T_b) \\ &\leq\; d(T_a, T_b), \end{aligned}$$

where the last inequality follows from triangle inequality of the SPR-distance. Hence, we have shown that (3) holds for $i = 2$.

Assume that (3) holds up to $i = k - 1$. We prove the $i = k$ case by contradiction. Let $T_a, T_b \in W_k^o$, with $f_k(T_a) \geq f_k(T_b)$. Suppose we have

$$f_k(T_a) - f_k(T_b) > d(T_a, T_b). \tag{4}$$

Further suppose $T_b \in W_{k-1}^o$. If we had $f_{k-1}(T_b) > f_k(T_b)$, then it would imply that there exists $T \in W_{k-1}^o$ such that $f_{k-1}(T_b) > f_{k-1}(T) + d(T, T_b)$, which would contradict the induction hypothesis. Also, if we had $f_{k-1}(T_b) < f_k(T_b)$, then it would contradict the definition of f given in (1). Hence, we conclude that $f_k(T_b) = f_{k-1}(T_b)$ if $T_b \in W_{k-1}^o$. But, then (4) would imply $f_k(T_a) - f_{k-1}(T_b) > d(T_a, T_b)$, thus contradicting definition (1). Therefore, we must have $T_b \notin W_{k-1}^o$. Now, let $T_c \in W_{k-1}^o$ satisfy $f_k(T_b) = f_{k-1}(T_c) + d(T_b, T_c)$. Then, (4) implies $f_k(T_a) - f_{k-1}(T_c) > d(T_b, T_c) + d(T_a, T_b)$. But, the triangle inequality $d(T_b, T_c) + d(T_a, T_b) \geq d(T_a, T_c)$ implies $f_k(T_a) - f_{k-1}(T_c) > d(T_a, T_c)$, which contradicts definition (1). Thus, (4) cannot be true and this completes our induction. ■

It now follows from the above proposition that step 2(b) in the algorithm can be modified as follows:
For all $1 \leq i < \ell$ and $T_a \in W_{i+1}^o$, recursively determine

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

3.4 Reduction of Data

In performing our algorithm, some character columns do not influence the determination of $\mathcal{R}_{\rm o}(S)$ and therefore can be ignored. 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 $\mathcal{R}_{\rm o}(S)$; i.e., if S' denotes the reduced data, then $\mathcal{R}_{\rm o}(S) = \mathcal{R}_{\rm o}(S')$.

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.

3.5 A Simple Example

To illustrate how our algorithm works, we consider the following very simple example:

$$\begin{array}{l} 1\,1\,0\,0\,0\,0\,0\,0\,0 \\ 1\,1\,0\,1\,1\,1\,0\,1\,1 \\ 1\,0\,1\,0\,0\,1\,1\,1\,0 \\ 1\,1\,0\,1\,1\,1\,0\,1\,0 \\ 1\,0\,1\,1\,0\,0\,0\,0\,0 \\ 1\,0\,1\,1\,0\,0\,0\,0\,0 \end{array} \quad \longrightarrow \quad \begin{array}{c} 1\,0\,0 \\ 1\,1\,1 \\ 0\,0\,1 \\ 0\,1\,0 \end{array} \qquad \begin{aligned} W_1^{\rm o} &= \{T_4, T_9, T_{13}, T_{14}, T_{15}, T_{18}\}\,, \\ W_2^{\rm o} &= \{T_3, T_7, T_8, T_{10}, T_{12}, T_{17}\}\,, \\ W_3^{\rm o} &= \{T_1, T_2, T_5, T_6, T_{11}, T_{16}\}\,. \end{aligned}$$

The original data S is shown on the left hand side of the arrow. After the reduction steps described in §3.4, it reduces to the data S' shown on the right hand side of the arrow. Let $\mathbf{c}_1$, $\mathbf{c}_2$ and $\mathbf{c}_3$ denote the 3 columns of S'. As shown in Figure 2(a), there are 18 inequivalent ordered trees with 4 leaves. Trees T_1 and T_2 are inequivalent as ordered trees but equivalent as plain rooted trees. The same goes true for the pairs T_7, T_8 and T_{13}, T_{14}. The ordered trees compatible (c.f. §2.1) with $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3$ are given by $W_1^{\rm o}, W_2^{\rm o}, W_3^{\rm o}$, respectively.

Applying the algorithm described in §3.3, one can obtain the weighted graph shown in Figure 2(b). In this simple example, all edges have weight 1, and therefore any connected path from a tree $T_a \in W_1^{\rm o}$ to a tree $T_b \in W_3^{\rm o}$ is a solution to the problem. In summary, $\mathcal{R}_{\rm min}(S) = 2$ and there are 132 minimal paths.

3.6 Plain Rooted Trees or Ordered Trees?

In discussing our algorithm so far, we have been careful in using $\mathscr{T}_n^{\rm o}$ and $W_i^{\rm o}$ to refer to ordered trees. That is because coalescent events and recombination events occur at specific points in time, and ignoring the time ordering of the events can lead to contradictions. For example, biologically a recombinant cannot be older than its parents.

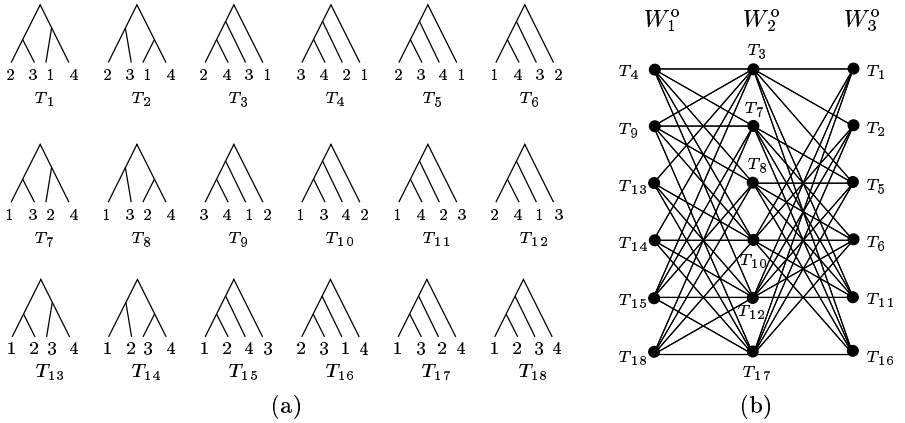


Fig. 2. (a) Inequivalent ordered trees with 4 leaves. (b) A graphical summary of performing our algorithm on the simple example. All edges have weight 1. There are 132 minimal paths, each leading to $\mathcal{R}_{\min}(S) = 2$.

We stress that the algorithm described in §3.3 always gives $\mathcal{R}_{\rm o}(S) = \mathcal{R}_{\min}(S)$. On the contrary, if plain rooted trees are used in the algorithm – i.e. $W_i^{\rm r}$ instead of $W_i^{\rm o}$ are used – to obtain $\mathcal{R}_{\rm r}(S) := \min_{T_a \in W_\ell^{\rm r}} f_\ell(T_a)$, then the number $\mathcal{R}_{\rm r}(S)$ may or may not be equal to the exact minimum $\mathcal{R}_{\min}(S)$. For less than or equal to 9 sequences, however, our investigation shows that $\mathcal{R}_{\rm r}(S) = \mathcal{R}_{\min}(S)$ for most cases of S. A possible explanation of this phenomenon is as follows. When there are only a small number of leaves in a tree, SPR operations usually involve subtrees with few leaves. As a 1-leaved subtree contains no internal vertices, there is no restriction on where the 1-leaved subtree can be regrafted, and therefore if $\mathcal{R}_{\rm r}(S)$ can be obtained through a series of SPR operations each involving a single leaf, then we should have $\mathcal{R}_{\rm r}(S) = \mathcal{R}_{\min}(S)$.

As shown in Table 1, 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, it would be a good strategy to use first plain rooted trees to compute $\mathcal{R}_{\rm r}(S)$ and try to reconstruct history as described in §3.7. If a consistent history can be reconstructed using only $\mathcal{R}_{\rm r}(S)$ recombination events, then we can conclude that $\mathcal{R}_{\rm r}(S) = \mathcal{R}_{\min}(S)$.

3.7 Reconstruction of History

After performing the algorithm from $\mathbf{c}_1$ to $\mathbf{c}_\ell$, we can find out which trees $T \in W_\ell^{\rm o}$ have $f_\ell(T) = \mathcal{R}_{\rm o}(S)$ and obtain minimal paths to those trees. Given a minimal path $T_{i_1}, T_{i_2}, \ldots, T_{i_\ell}$, one can combine the trees in the minimal path into an ARG. More precisely, possible sets of SPR operations that can transform T_{i_m} to $T_{i_{m+1}}$ tell us how the trees can be combined. In general, if $T_{i_m} \neq T_{i_{m+1}}$, there could be more than one way to transform T_{i_m} to $T_{i_{m+1}}$. For instance, as we discussed in §2.2 the tree T shown in Figure 1 can be transformed into

T_2 in several inequivalent ways. Hence, in general more than one ARG may be constructed for a given minimal path. The number of inequivalent ARGs corresponding to a minimal path depends on the topology of the trees involved.

3.8 The Unit-Step Approach

In the algorithm outlined in §3.3, for every $T \in W_{i+1}$ not contained in W_i, one has to compute $d(T, T')$ for all $T' \in W_i$. As computing the SPR-distance between trees is the most computationally intensive part of the algorithm, the problem can quickly become intractable. In what follows, we propose 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 $\mathcal{R}_{\mathrm{r}}(S)$ or $\mathcal{R}_{\mathrm{o}}(S)$, respectively. As mentioned before, the relations $\mathcal{R}_{\mathrm{r}}(S) \leq \mathcal{R}_{\mathrm{o}}(S) = \mathcal{R}_{\min}(S)$ hold true for all S.

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

$$N_r(X) := \left\{ T \in \mathscr{T}_n \,\middle|\, T \in \bigcup_{m=0}^{r} U_m(T') \text{ for some } T' \in X \right\} .$$

In the implementation of our algorithm, we pre-compute the adjacency-set $U(T)$ for all $T \in \mathscr{T}_n$ and store them in a file which can be accessed by our program. Therefore, computing $N_r(X)$ can easily be done. We define the diameter d_n of $\mathscr{T}_n$ as the maximum value of $d(T, T')$ over all trees $T, T' \in \mathscr{T}_n$. As shown in [11], $d_n \leq n - 2$. In the following discussion, define $\mathcal{N}(T, m, i) := N_1(\{T\}) \cap N_m(W_i)$.

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

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

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

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

Here, $\delta_{a,b}$ denotes the Kronecker delta, which is 1 if $a = b$ and 0 if $a \neq b$. The minimum number of recombination events is given by $\min_{T \in W_\ell} f_{\ell,0}(T)$, which is equal to the value $\min_{T \in W_\ell} f_\ell(T)$ defined in §3.3.

There are several advantages to the algorithm just described over that in §3.3. First of all, note that for each tree T, we just need to compare a function evaluated at T with a function evaluated at at most $|N_1(T)| = |U(T)| + 1$ trees. As shown in Table 1, the maximum size of $|U(T)|$ does not grow as fast as the number $w(n, k)$ 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 $1 \leq i < \ell$.

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.

3.9 Haplotype Blocks

For each minimal path our algorithm finds, in addition to knowing which trees are selected, 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, for each minimal path, we define a block as consecutive positions in the sequence where the same tree is supported. As there could be many minimal paths, it could be 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.

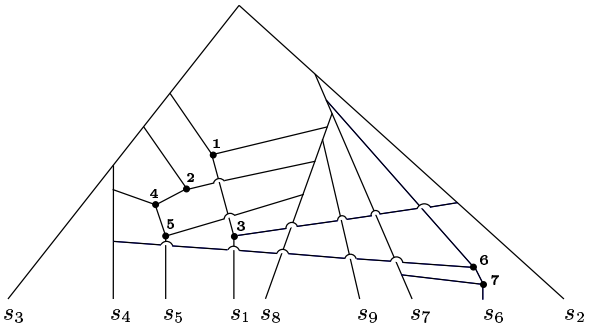
Although in general we cannot find a unique haplotype block structure, we can still ask questions to each of which there exists a unique answer for a given data set S. For example, we can ask the following question: If a block is obtained as described above, what is the minimum number of haplotype blocks that can be defined by a minimal path? We address this question in §4, where we consider a specific application of our method.

4 Application

In this section, we apply the methods discussed in this paper to analyse Kreitman's 1983 data of the alcohol dehydrogenase locus from 11 chromosomes of *Drosophila melanogaster* [8]. The data has been taken from 5 geographically distinct populations and the aligned sequence length is 2800 base-pairs. Ignoring insertions and deletions, there are 43 polymorphic columns in the data. We have transformed the polymorphism data into binary sequences as shown in Figure 3.

Wa-S	=	000	000001100000	0	00110111011111000000	0	0000000
F1-1S	=	001	000000000000	0	00110111011111000000	0	0000000
Af-S	=	000	000000000000	0	00000000000000000000	1	0000101
Fr-S	=	000	000000000000	0	11000000000000000000	1	0011000
F1-2S	=	000	110001011001	1	11000000000000000000	0	1000000
Ja-S	=	001	000000000000	1	00000000000000010101	1	1000010
F1-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

Fig. 3. Kreitman's data in binary form. Also shown is a haplotype block structure with 6 blocks, whose boundaries are indicated by vertical solid lines.



1 : $_L(3,4)_R$
2 : $_L(3,4)_R$
3 : $_L(15,16)_R$
4 : $_R(16,17)_L$
5 : $_L(35,36)_R$
6 : $_R(35,36)_L$
7 : $_R(36,37)_L$

Fig. 4. A minimal ancestral recombination graph for Kreitman's data. Recombination vertices are denoted by •. The notation $_{S_L}(i,j)_{S_R}$ is used to denote the location (i,j) of the break-point and to indicate that S_L part of the recombinant gets descended from the left edge and S_R part from the right edge.

Since the sequences for Fr-F, Wa-F and Af-F are identical, in our analysis we only need to consider 9 distinct sequences. Hence, we relabel the sequences as follows:

$s_1 :=$ Wa-S	$s_2 :=$ F1-1S	$s_3 :=$ Af-S
$s_4 :=$ Fr-S	$s_5 :=$ F1-2S	$s_6 :=$ Ja-S
$s_7 :=$ F1-F	$s_8 :=$ Fr-F = Wa-F = Af-F	$s_9 :=$ Ja-F

As discussed in §3.6, we have performed our analysis on $S = \{s_1, \ldots, s_9\}$ first using rooted trees, obtaining $\mathcal{R}_{\rm r}(S) = 7$. We have then checked that it is indeed possible to construct an ARG with exactly 7 recombination events. In addition, we have enumerated all minimal paths, each of which leads to $\mathcal{R}_{\rm r}(S) = 7$. It turns out that there are about 10 million minimal paths for Kreitman's data; this number is not so surprisingly high, since for each recombination event, there could be numerous choices regarding which subtree undergoes an SPR operation and where in the sequence the event occurs.

The minimum number of haplotype blocks that can be defined by a minimal path is 6. A haplotype block structure with 6 blocks is shown in Figure 3, where block boundaries are indicated by vertical solid lines. A minimal ARG associated to a minimal path which generates such a block structure is shown in Figure 4. Note that 2 recombination events occur between columns 3 and 4, as well as between 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. Figure 5 illustrates where in the actual data the recombination events shown in Figure 4 are supposed to occur.

Let us now compare our result on $\mathcal{R}_{\rm min}(S)$ with that given by some currently-available methods. In [6] Hudson and Kaplan also have examined Kreitman's data. Their algorithm gives 5 as a lower bound on the number of recombination events. If one uses the algorithm developed by Myers and Griffiths [9], one would obtain 6. In [12] the present authors have analysed Kreitman's data using a method based on set theory and have obtained 7 as a lower bound.

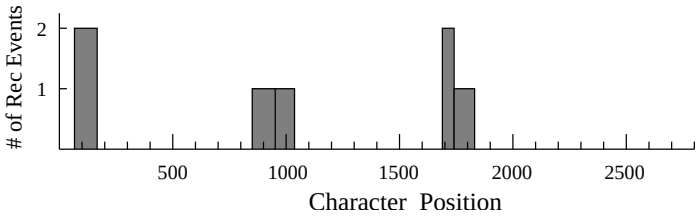


Fig. 5. Recombination locations in the actual data. Shaded areas enclosed by solid lines indicate the regions in which recombination events shown in Figure 4 occur. The number of events for each region is indicated by the height of the shaded area.

5 Concluding Remarks

Under the infinite-sites model of mutation, our algorithm finds the exact minimum number of recombination events in the evolutionary history of sampled sequences. The method introduced in this paper for computing the distance between trees has allowed us to overcome some difficulties which have hitherto prevented an exact implementation of the dynamic programming idea. It is important to note, however, that even our new approach, as it stands, becomes infeasible for 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. In [13], Wang, Zhang and Zhang have shown that, under the infinite-sites model, the problem of reconstructing the evolutionary history with the minimum number of recombination events is NP-hard. Although they have constructed a polynomial-time algorithm for a restricted version of the problem, no polynomial-time algorithm is known for the general case.

Nevertheless, for more than 9 sequences, we can try the following. The algorithm proposed by Myers and Griffiths uses local bounds for small regions to construct a global bound for the entire data [9]. As there may not be so many distinct haplotypes if small regions are considered, we can use our algorithm to compute exact local bounds and use them in Myers and Griffiths' program to find a global bound. Combining our algorithm with that of Myers and Griffiths as just described should perform quite well.

In our future research, we plan to relax the assumption of the infinite-sites model and extend our investigation to consider gene conversion. Also, for a given data set, it would be interesting to infer mutation and recombination rates – for example, using the method developed by Fearnhead and Donnelly [2] – and estimate the probabilities of the minimal ARGs we obtain, as well as the distribution of the number of recombination events.

In this paper we have only begun to consider defining haplotype blocks based on parsimonious reconstruction of sequence evolution. As the example of Kreitman's data shows, in general our algorithm may find many minimal paths and therefore find many inequivalent candidate haplotype block structures. To be able to determine which structures are more probable, we need to be able to as-

sign some sort of confidence level to each block boundary. Clearly there remain many interesting questions to be addressed along this line of research.

Acknowledgments
We thank R. Lyngsø and 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

1. Daly, M.J. *et al.*, *High-Resolution Haplotype Structure in the Human Genome,* Nat. Genet. **29** (2001) 229-232.
2. Fearnhead, P. and Donnelly, P., *Estimating Recombination Rates from Population Genetic Data,* Genetics **159** (2001) 1299-1318.
3. Gabriel, S.B. *et al.*, *The Structure of Haplotype Blocks in the Human Genome,* Science (2002) **296** 2225-2229.
4. Hein, J., *Reconstructing Evolution of Sequences Subject to Recombination Using Parsimony,* Math. Biosci. **98** (1990) 185-200.
5. Hein, J., *A Heuristic Method to Reconstruct the History of Sequences Subject to Recombination,* J. Mol. Evol. **36** (1993) 396-405.
6. Hudson, R.R. and Kaplan, N.L., *Statistical Properties of the Number of Recombination Events in the History of a Sample of DNA Sequences*, Genetics **11** (1985) 147-164.
7. Johnson, G.C. *et al.*, *Haplotype Tagging for the Identification of common Disease Genes,* Nat. Genet. **29** (2001) 233-237.
8. Kreitman, M., *Nucleotide Polymorphism at the Alcohol Dehydrogenase Locus of Drosophila Melanogaster*, Nature **304** (1983) 412-417.
9. Myers, S.R. and Griffiths, R.C., *Bounds on the Minimum Number of Recombination Events in a Sample History,* Genetics **163** (2003) 375-394.
10. Schröder, E., *Vier Combinatorische Probleme,* Zeit. für. Math. Phys, **15**, (1870), 361-376.
11. Song, Y.S., *Notes on the Combinatorics of Rooted Binary Phylogenetic Trees,* submitted to Annals of Combinatorics for publication.
12. Song, Y.S. and Hein, J., *On the Minimum Number of Recombination Events in the Evolutionary History of DNA Sequences,* to appear in J. Math. Biol.
13. Wang, L., Zhang, K. and Zhang, L., *Perfect Phylogenetic Networks with Recombination,* J. Comp. Biol, **8** (2001) 69-78.