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

Article *in* Genetics · February 2003

Source: PubMed

2 authors:

Simon R Myers
University of Oxford
172 PUBLICATIONS **34,210** CITATIONS

SEE PROFILE

Robert C Griffiths
University of Oxford
135 PUBLICATIONS **6,589** CITATIONS

SEE PROFILE

Bounds on the Minimum Number of Recombination Events in a Sample History

Simon R. Myers[1] and Robert C. Griffiths

Department of Statistics, University of Oxford, Oxford OX1 3TG, United Kingdom

Manuscript received May 7, 2002
Accepted for publication October 7, 2002

ABSTRACT

Recombination is an important evolutionary factor in many organisms, including humans, and understanding its effects is an important task facing geneticists. Detecting past recombination events is thus important; this article introduces statistics that give a lower bound on the number of recombination events in the history of a sample, on the basis of the patterns of variation in the sample DNA. Such lower bounds are appropriate, since many recombination events in the history are typically undetectable, so the true number of historical recombinations is unobtainable. The statistics can be calculated quickly by computer and improve upon the earlier bound of Hudson and Kaplan (1985). A method is developed to combine bounds on local regions in the data to produce more powerful improved bounds. The method is flexible to different models of recombination occurrence. The approach gives recombination event bounds between all pairs of sites, to help identify regions with more detectable recombinations, and these bounds can be viewed graphically. Under coalescent simulations, there is a substantial improvement over the earlier method (of up to a factor of 2) in the expected number of recombination events detected by one of the new minima, across a wide range of parameter values. The method is applied to data from a region within the lipoprotein lipase gene and the amount of detected recombination is substantially increased. Further, there is strong clustering of detected recombination events in an area near the center of the region. A program implementing these statistics, which was used for this article, is available from http://www.stats.ox.ac.uk/mathgen/programs.html.

RECOMBINATION is one of the major influences on genetic diversity in many organisms. Understanding well the part it has to play is crucial to applications including disease association mapping and to many population genetic analyses. Of particular interest in humans is answering the question of how much evolutionary recombination has occurred in different genomic regions; this has major implications for the effort required, and difficulties involved, in disease-mapping studies. This question is very difficult to answer through traditional pedigree-based methods used to estimate recombination rates, because we are often interested in recombination within a relatively short region. For such a region the likelihood of a recombination event in any one generation is likely to be extremely small, although recombination events over much longer genealogical timescales can still have a strong influence on the population ancestry. In this case, the size of pedigree study required to observe sufficient recombinations within the region, to estimate the rate of occurrence of these events accurately, is often prohibitively large. The rapid expansion occurring in available sequence data sets enables a different approach based on the analysis of such data. Although all genetic variation is ultimately created through mutation, recombination can create new variants by combining types already present in the population. Detecting such historical recombinations is important in understanding the role of recombination in the creation of the patterns of variability that we observe. These detections enable us to reconstruct events that have shaped the history of a present-day sample.

In this article we develop methods that can be applied to DNA sequence data to obtain lower bounds on the number of recombination events that have occurred in the genealogy describing the joint history of a sample. Such lower bounds are useful since they give a measure of the extent this genealogy must differ from a tree structure (which is what we would see in the absence of recombination). Further, since some recombination events are fundamentally undetectable (for example, both parents of the recombinant are of the same type in the region sequenced) it makes sense to consider the minimum number that might have occurred in the history. The new methods also allow us to look at the pattern of detection across a region, giving an idea of where more historical recombination events may have occurred.

The bound R_m introduced by Hudson and Kaplan (1985) already gives a minimum number of recombination events in the history of a sample. It is based on the four-gamete test, which infers a recombination event between pairs of diallelic loci at which all four possible gametic types are present. Such an event must have

[1]*Corresponding author:* Department of Statistics, 1 S. Parks Rd., Oxford OX1 3TG, England. E-mail: myers@stats.ox.ac.uk

occurred if both loci have mutated at most once since the ancestor of the sample. R_m is constructed through testing all pairs of loci in the region sequenced and using the four-gamete test to construct a collection of intervals, within each of which recombination is detected. Under the conservative assumption that overlapping intervals signal the same recombination event, the algorithm of Hudson and Kaplan (1985) finds the largest subset of *non*overlapping intervals from the original collection. R_m is then the number of such intervals (each of which must indicate a different recombination event). R_m has the advantage of being very fast to compute, but it is not optimal; there is often no possible history for the data in which there are R_m recombinations. Further, it is known that R_m misses most of the recombination events in the sample history (Hudson and Kaplan 1985), particularly where the mutation data are limited or the rate of recombination is high. Thus there is reason to hope that an improved minimum could offer detection of more recombination events.

Hein (1990, 1993) also introduced and developed an algorithm that can be used to give a minimum number of recombination events for a set of sequences, using a dynamic programming algorithm that works along the polymorphic sites. Unfortunately this method becomes computationally infeasible for larger numbers of sequences, for example, more than eight, due to the fact that the method must consider a large number of possible evolutionary trees at each locus. For these cases an approximating algorithm must be used that no longer guarantees that the solution is a true minimum. Currently no known algorithm will rapidly compute the true minimum number of recombinations that are needed to construct a history for a given data set.

Here we develop two new statistics, R_h and R_s, which give lower bounds on the number of recombination events in the history of a sample. A computer program that can calculate both R_h and R_s upon input of a set of sequence data has been written. For any given data set, $R_s \geq R_h \geq R_m$ so both minima offer an improvement over R_m. Both statistics employ a technique developed here that enables the combining of local recombination bounds to create a better overall bound on a larger region. The difference is on the way these local bounds are obtained, and indeed future improved methods of obtaining these bounds could result in further increases in recombination detection. The first new bound R_h is based on bounding the number of recombination events by calculating the difference between the number of observed types in the sample and the number of segregating sites; at least this number of types must have been created at recombination, rather than mutation, events. The second statistic R_s bounds the number of recombinations by approximating the history of the data using a simplified version of recombination events, in such a way that any true history for the data has more recombination events than one of these approximate histories. By minimizing the number of such simplified recombinations over all possible approximate histories, a lower bound on the number of true recombinations is obtained. R_h is rapidly computable for large data sets, so its statistical properties can be estimated by simulation under different models. For every pair of loci, the minimum number of recombination events between that pair is easily obtainable, and the collection of such bounds can be viewed graphically, so regions with more detectable recombinations can be visually identified. This can be used in conjunction with similar visual aids, such as the pairwise linkage disequilibria and incompatibility matrix plots.

The statistic R_h is applied to a real data example, the lipoprotein lipase (LPL) data set sequenced by Nickerson *et al.* (1998). For this data we compare the number of recombinations detected by R_h and R_m and look at whether the pattern of detections shows clustering of recombination events along the 10-kb region sequenced (as suggested by Templeton *et al.* 2000a using a different method). The findings are discussed in the light of the assumptions under which R_h is valid. Readers most interested in the practical application of the new statistics could go straight to this section.

The following gives a guide to the subsequent sections in this article; the first few sections detail the method and its implementation, followed by a section on simulation results under a neutral coalescent model of evolution. Next is a section devoted to the LPL data set application, before the final discussion. Some of the earlier sections contain results that require proof; the simpler results are proved within the text, while two longer proofs are placed in separate appendices.

First, in combining local recombination bounds we explain the central mathematical concept of the approach offered here. Given a collection of bounds on how many recombination events are needed for each of a collection of continuous subregions of a larger region, we provide an algorithm (Algorithm 1) whereby these bounds can be combined in an efficient manner to give the best possible overall bound for the parent region. This gives a general method to combine recombination information from different subregions. Within local recombination event bounds we first consider an existing method and then derive two new methods to obtain such subregion bounds, for input into the algorithm; the first new method results in the statistic R_h, and the second the improved statistic R_s (which is not so rapid to compute), upon combining the bounds.

These two sections give enough for a basic implementation of the method; the simple example (of a small constructed data set) of the following section illustrates this and shows that the bounds R_m, R_h, and R_s can give different numbers of detected events. A section (implementation of the bounds) that details some technical results important to implementing the methods efficiently follows; the reader less interested in this

aspect could omit this. Following this, in SIMULATED PROPERTIES we investigate how well all the minima perform under coalescent simulations, both to suggest sensible parameter choices in obtaining R_h (where we can choose input values, increasing which improves the bound but lengthens the time taken by the algorithm) and to look at how much improvement is possible over R_m, for different scaled mutation and recombination rates and using the new statistics.

After the LPL application described above, in the DISCUSSION we talk about aspects of the results found and possible future applications of the statistics (for example, estimating the recombination parameter ρ) as well as possible further developments and extensions to the methods introduced here.

We assume throughout that each mutation observed in the sample is the result of a single mutation event in the sample history; this holds, for example, in the so-called "infinite-sites" model of mutation. Thus the sample types may be represented in binary form at each locus. Further, it is assumed that there is no gene conversion in the sample history, all recombination events take the form of reciprocal crossover events, and the DNA region that we analyze is sufficiently short that for every recombination event within it, exactly one endpoint lies within the region and the other outside. Thus we make the same assumptions as used in the derivation of R_m by Hudson and Kaplan. If gene conversion is possible, the statistics R_m, R_h, and R_s will each give valid lower bounds on the number of reciprocal crossover/gene conversion event endpoints within the region. In the event of repeat mutation the bounds produced may no longer be valid; the degree to which this distorts the results obtained will depend on the amount of repeat mutation that has occurred.

COMBINING LOCAL RECOMBINATION BOUNDS

In this section we derive an algorithm to combine sets of local regional bounds on the number of past recombinations to obtain a bound for a longer stretch; in the next section we see how such local bounds may be obtained. The use of this algorithm is central to obtaining all the minima developed in this article. The general approach used here is to consider bounding the number of recombination events as a mathematical optimization problem. Suppose we have phased haplotype data for a sample of size n with S consecutively labeled segregating sites at s_l for $l = 1, 2, \ldots, S$. The data provide information on recombination events between pairs of segregating sites, in other words on the number of recombination events r_l in the region (s_l, s_{l+1}), for $l = 1, \ldots, S-1$. Now suppose we have some method of obtaining a local bound B_{ij} on the number of recombinations in the region (s_i, s_j) for $i < j$; possible ways of obtaining such bounds are discussed in the next section. Then we may write this bound mathematically in the form of a constraint on the r_l's:

$$\sum_{l=i}^{j-1} r_l \geq B_{ij}. \tag{1}$$

The total number of recombination events in the whole region is $\Sigma_{l=1}^{S-1} r_l$. We wish to find a minimum number of recombination events, so must choose a vector $(r_1, r_2, \ldots, r_{S-1})$ giving the inferred recombination events in each interval, which minimizes this sum while satisfying all the local bounds. Thus for a collection (B_{ij}) of such nonnegative bounds for the sample (some of which may be zero), the minimum bound satisfying all the constraints corresponds to the solution of the optimization problem: Minimize $\Sigma_{l=1}^{S-1} r_l$ over nonnegative integers $\{r_l\}$ such that

$$\sum_{l=i}^{j-1} r_l \geq B_{ij} \quad \text{for } 1 \leq i < j \leq S. \tag{2}$$

This is an example of an integer linear programming (ILP) problem. Various algorithms have been developed to solve such problems, although these can be computationally costly for large systems. Fortunately, in this case, we often have a relatively small set of constraints. More importantly, the fact that the constraints and objective function are of the same form enables a simple dynamic solution, which means we can easily use combinations of bounds in this form. In fact, there is an efficient method of solving (2), which can be used to give the minimum number of recombinations between every pair of segregating sites in the sample. This results in a minimum bound matrix R with R_{st} equal to the minimum number of recombination events between sites s and t needed to satisfy the bound system. This can be more informative than reporting a single realization of a minimal solution since it incorporates uncertainty as to where the detected recombinations occur in the region. R may be constructed using the following algorithm.

ALGORITHM 1. *Define a matrix R, where for $1 \leq s < t \leq S$, R_{st} is the optimal value for the objective function of the integer linear programming problem: Minimize $\Sigma_{l=s}^{t-1} r_l$ over nonnegative integers $\{r_l\}$ such that*

$$\sum_{l=i}^{j-1} r_l \geq B_{ij} \quad \textit{for } 1 \leq i < j \leq S. \tag{3}$$

Then we may construct R by the following algorithm:

1. *Set $R = 0$ and $k = 2$.*
2. *(Maximization step) For $j = 1, 2, \ldots, k-1$ set*

$$R_{jk} = \max\{R_{ji} + B_{ik} : i = j, j+1, \ldots, k-1\}.$$

3. *(Incrementing step) If $k < S$, increment k by 1 and go to step 2.*

In particular, the minimum for the whole region is given by R_{1S}.

Proof. See APPENDIX A. ■

This algorithm has been used to investigate the patterns of detectable recombination in a human data set from the LPL locus; the results from the analysis are discussed later, in LPL DATA APPLICATION. R may be viewed graphically, in a similar manner to the incompatibility matrix; this is useful for visualizing the pattern of detection along a length of sequence.

Thus we have a general solution enabling us to combine different bounds on the number of recombinations that occur in the history. Algorithm 1 is really a dynamical programming algorithm, which efficiently gives an optimal bound for every pair of sites. Further, if only the overall bound is required, this can be obtained easily by performing the algorithm but fixing $j = 1$ in the maximization step. This gives the required bound after looking at every element B_{ij} only once, so is an efficient solution to the system (2). A particular optimal solution vector can then be recovered by subtracting consecutive elements of the row $R_{1\cdot}$ produced. It should be noted that this particular solution results in the placement of all recombination events as far to the right along the sequence as possible; this rightward bias can obviously mean that the particular solution produced is not necessarily a good indicator of where recombinations actually occurred along the sequence. By working in the other direction, we can equally subtract consecutive elements of the column $R_{\cdot S}$ to give another (not the same in general) optimal solution. This will have a corresponding leftward bias.

The optimization approach is also useful if we wish to include known or model-based information about the history of the sample, provided this can be phrased as constraints on the r_l's. For example, if we wish to impose $r_k = 0$ then we need only remove the term in r_k from each constraint. This results in a system of the same form as before, which may readily be solved through the same algorithm; the new system will have no solution if $B_{k(k+1)} > 0$, corresponding to a known recombination in the interval (s_k, s_{k+1}). An extension of this is the case where recombination is allowed only between a few sites or even a single adjacent pair of sites; this again results in the same form for the system of equations, which may be solved using the method of Algorithm 1. It is clear that more complicated constraints could also be introduced.

LOCAL RECOMBINATION EVENT BOUNDS

The usefulness of the approach above is dependent on the quality of the collection of local bounds obtained. Here we suggest one existing and two new methods that give sets of local bounds, with later methods always improving on the earlier ones. The first method simply uses the four-gamete test (employed by HUDSON and KAPLAN 1985 to calculate R_m) and incorporates the results of this test into the new framework; unsurprisingly, the minimum this results in is equal to R_m, so with these conditions Algorithm 1 gives an alternative method of obtaining this statistic. More interestingly, the two new methods result in statistics R_h and R_s, respectively, when Algorithm 1 is employed, and these two offer improved detection over R_m.

The first of the two new methods creates a "haplotype bound" for a local region on the basis of counting the number of different types in the sample produced by various subsets of the full collection of segregating sites in that region and deducing how many of these types must be created by recombination. The second, described in *Bounds from simulation of the sample history*, works by evolving the sample back, coalescing, and mutating until a recombination event is needed, and then choosing the recombinant from the remaining sequences. The chosen recombinant is then removed and the process repeated until a single ancestral sequence remains. The bound produced is the minimum number of removals over the different choices of recombinants. To obtain a local bound for a region, information from segregating sites outside that region is temporarily disregarded in creating the history.

The bound that is possible depends on whether the ancestral type is known or unknown at each locus; in the case where we have an outgroup, the type is known and this lends us extra information. If ancestral types are known, then for convention we assume that the ancestral type is designated as a 0 in the binary sequence data.

Hudson and Kaplan's R_m: The conditions obtained by HUDSON and KAPLAN (1985) can be viewed in this framework as the set of equations obtained using the four-gamete test for recombination in the sample history. This uses the fact that if mutations occur only once in the sample history, a recombination event can be inferred in (s_i, s_j) provided all four possible gametic types 00, 01, 10, and 11 are present within the sample at the respective loci s_i and s_j. In this case we say that sites i and j are *incompatible*. In the form of the system (2) we can view the result of the four-gamete test in terms of a bound B_{ij} for each pair $i < j$ of sites, where $B_{ij} = 1$ if sites i and j are incompatible (so there must be at least one recombination between sites i and j) and $B_{ij} = 0$ otherwise. Thus in this case, viewing the system of constraints as a matrix, it is just the usual incompatibility matrix of pairs. The algorithm given in HUDSON and KAPLAN (1985) that gives the minimum bound subject to these constraints involves "pruning" of the set of incompatible pairs and seems rather different from the solution of Algorithm 1. However, it is clear on closer inspection that both methods lead to the same bound R_m. In fact, the right endpoints of the disjoint intervals remaining in Hudson and Kaplan's method correspond exactly to the only nonzero values for r_k obtained in evaluating the particular optimal solution

from the top row $R_{1.}$. Thus with these bounds the new approach gives a new algorithm that will obtain the same bound as the older method.

The bounding method above holds whether ancestral types are known or unknown. However, in the case of known types the bound can be improved slightly through a refinement of the test. In this case, a recombination can be inferred whenever the *three* types 01, 10, and 11 are present in the sample (Gusfield 1991, for example). This is because we can effectively infer the existence of the ancestor type 00. Thus we can obtain a *types known* incompatibility matrix and corresponding equation system, the solution of which improves the bound in certain cases; both the types-known and the types-unknown bounds are referred to as R_m from here on. The knowledge of ancestral types is not expected to improve R_m much in practice, since for large sample sizes the ancestral type 00 will be present for most pairs of sites considered. Note that the improved set of equations obtained when types are known corresponds to adding an all-zero type to the top of our list of sequences and then proceeding as if types were unknown.

Haplotype bounds: The method above clearly throws away information in the data about the recombination history, since it considers only pairwise comparisons. This results in a nonoptimality of the bound; it is not always possible to construct a history for the sample with only R_m recombinations. In fact, the incompatibility bound can be viewed as a special case of a wider class of bounds. Suppose that there are $H \leq n$ distinct haplotypes in our sample of size n. Consider any possible history describing the evolution of the sample backward in time. The ancestors of our sample may undergo recombination, mutation, or coalescence events in the history, before all lineages eventually coalesce to a single ancestor of the whole segment of DNA being studied (Griffiths and Marjoram 1996b, for example), called the most recent common ancestor (MRCA). At a time t back from the present, let H_t be the number of types present in these ancestors, so $H_0 = H$. We can make the following observations:

1. H_t eventually declines to 1.
2. H_t remains unchanged by a coalescence of two lines (they must be identical types to coalesce).
3. H_t decreases by at most 1 at each mutation or recombination event in the history.

Looking forward in time from the ancestor to the present, this means that each recombination or mutation event in the ancestry creates at most one new type of the H that must be created altogether to make up the present-day sample. The original ancestor makes up at most one of these types depending on whether or not it is present in the sample. Then the total number E of recombination or mutation events in the sample history must satisfy $E \geq H - 1$ to create the observed number of types. Further, if the number of such events equals $H - 1$, then one of the sample members is of the same type as the sample MRCA. But we know the number of mutations in the sample history; this is just S as each segregating site is assumed to have mutated exactly once since the MRCA. Hence, if R is the number of recombinations in the history, we have $R = E - S$ and this gives the bound

$$R \geq \begin{cases} H - S - 1: & \text{Ancestor type present in the sample} \\ H - S: & \text{Otherwise.} \end{cases} \quad (4)$$

This may seem like a very weak bound, since whenever there are more segregating sites than distinct haplotypes it will give a negative answer, leading to a zero bound on the number of recombinations. However, it becomes much more powerful when we note that there is no reason why we may not apply it to *any* subset of sites in the data, not merely the full set of sites. Thus from the original S sites we may choose any $S' \leq S$ say (for example, the first, third, and fourth sites), consider the types at these sites only, and obtain a corresponding local bound that holds for the region between the endpoint sites in our set (sites one and four in our example). Examining many subsets will typically give a much-improved bound collection! For a given local region, the best "haplotype bound" is the maximum bound obtained from applying (4) to subsets with the same endpoint sites as that region. This collection of local bounds gives the overall bound R_h when Algorithm 1 is applied.

If we do not know ancestral types in the history, we must use the first bound of (4) since the ancestral type may or may not be present. If types are known, we know which set of conditions in (4) is satisfied, depending on whether or not there is an all-zero (ancestral) type within the sample. Then if we always add an all-zero type (before any other analysis) to the front of the list of types in this case, as before it is clear that applying the type's unknown bound will always give the correct lower bound here. Finally, if we consider only the subsets consisting of pairs of elements (*i.e.*, the endpoint pairs) the haplotype bound is equivalent to the three- or four-gamete test, and so the incompatibility test can be viewed as a special case of (4).

If there are S sites then there are a total of 2^S subset collections of these S. For large S it is thus impractical to consider all subsets of the original site collection, particularly since we need to count types for each such collection. The approach taken to address this problem by the current implementation of the minimum is to introduce parameters that reduce the set of site collections considered in producing R_h. There are two parameters, and these set (1) the maximal subset size S' that is to be considered and (2) the maximal distance apart of the endpoint sites (after removing compatible sites).

Increasing either parameter will increase the number of subsets used to give local bounds and can thus increase the minimum produced. This improvement is at

the cost of more computation time. The second parameter sets the maximal width after removing sites compatible with all others (which cannot affect R_h) from the data set; further details of why this is done are given in IMPLEMENTATION OF THE BOUNDS. Note that setting a low value for this parameter means that local bounds are computed only for the number of recombination events between site pairs that are close together. Choosing sensible values for these parameters often means the bound obtained is as good as if all subsets were considered; sample sizes of 1000 with hundreds of segregating sites can easily be analyzed on an average Pentium PC. In practice the best approach for a given data set is to start with reasonable small values for the parameters and then increase these until either the bound no longer improves or (for a very large data set) the run time of the program becomes too long. The implementation also performs further steps that reduce computation time; some of these are described further in the section IMPLEMENTATION OF THE BOUNDS.

Bounds from simulation of the sample history: The previous two bounds have had a somewhat loose genealogical interpretation in that they do not attempt to directly reconstruct the history of the sample in any way. Improvements to the bounds they offer can be made through searching over such histories; this, however, comes at a time cost relative to the previous bounding techniques. The idea is to search over possible histories of the sample.

The following proposition gives us a general condition under which we may simplify our sequence data to produce a smaller data set, which requires the same number of recombination events in the sample history.

PROPOSITION 2. *For a sample S of size n, the minimum number R_S of recombinations in the sample history until the MRCA is equal to $R_{S'}$, where the new sample S' is formed from S by either of the following two events:* (1) *coalescence of two sample members of the same type and* (2) *removal of a noninformative site. If the ancestral type at a segregating site is known, it is defined as noninformative if and only if exactly one member of the sample is of the mutant type. If the ancestral type is unknown, the site is noninformative if and only if all but one of the sample members are of the same type at that site.*

Proof. See APPENDIX B. ■

The proposition above enables us to produce data set S', a smaller data set than S, which nevertheless requires the same number of recombinations in its history, under certain conditions; in other words, to simplify the problem of finding a minimum. More importantly it motivates the following algorithm, which gives a lower bound for the number of recombinations needed in a region to explain the history of a data set S consisting of a list of sequence types at segregating sites.

ALGORITHM 3. *A lower bound for the number of recombination events in the history of a data set S is given by the following algorithm:*

1. *Initially put $B = \infty$ and set $R = 0$.*
2. *If two sequences in S are of the same haplotype, coalesce them. If a site s is noninformative, remove the data corresponding to s from S. Repeat while it is still possible to perform such an event and more than one sequence remains in S.*
3. *If only one sequence remains at this point, record the number of removal events R in the current history and if $R < B$ put $B = R$.*

   Otherwise more than one sequence remains; choose one of the remaining sequences and remove it from the list S. Add 1 to R and go to step 2.
4. *If there remains some possible sequence of removal events not yet tried, start a new history, set $R = 0$, and return to step 2 with the original data set S. Otherwise return B as the required bound.*

Proof. The algorithm returns the minimum number of removal events over possible histories of S, which consist of coalescence, mutation, and removal events. We use induction on the number of sequences n. Now the algorithm is valid for $n = 2$ since no recombinations are ever needed to construct a history for a sample size of 2 or less. Supposing it gives a true lower bound up to a sample size of $n - 1$, for a data set S with a sample size of n let D_S be the *true* minimum number of recombinations required to construct a history for S. We need to show that when applied to S, the algorithm returns a number that is no larger than D_S.

Considering beginning the algorithm, we know by Proposition 2 that performing the second step of the algorithm does not change the number of required recombinations in the history. This step then produces a modified data set S' say, which also requires D_S recombinations in its history. If S' has $n - 1$ sequences or less, the algorithm then obtains a valid bound by assumption. Otherwise, note that the first event back in time must be a recombination to an ancestor of the sample, since no other event type is possible. Letting H be some optimal history of the data set S', with D_S recombination events in total, this first recombination occurs to the ancestor of sample member a say. Define a set S'' as S' with a removed from the list. Then S'' is a set of $n - 1$ sequences. Further, if we follow the evolution of the ancestors of the members of S'' using H, we obtain a history for this modified data set. This history cannot contain the first recombination event, so has at most $D_S - 1$ recombination events. Now our inductive hypothesis tells us that applying the algorithm to S'' will give a bound of at most $D_S - 1$.

Since the algorithm tries all possible chains of removal events, at some point sequence a will be chosen as the first removal. The resulting data set is then S''. Because

the algorithm gives a bound of at most $D_S - 1$ for this data set, some chain of removals for S'' then includes at most $D_S - 1$ removals before step 4 of the algorithm is reached. But adding a to the front of this chain gives a list of removals for the original S with at most $D_S - 1 + 1 = D_S$ removals altogether, and as the bound given is the minimum over such chains, the bound returned by the algorithm is at most D_S, the true minimum. ■

The basic idea of the algorithm is to look over possible histories, using the proposition to perform events that do not change the minimum number of recombinations, where this is possible. If at some point before a single ancestor is reached we must stop, it is necessary to have a recombination event; we choose a member of the current ancestor set to "recombine." However, to simplify things instead of recombining this sequence we simply remove it from the list and continue. This results in a simpler data set (which then needs no more recombinations) than the one we should have if we included ancestors of the recombinant. It also speeds things up since there is no need to look over possible breakpoint positions. After enough removal, coalescence, and mutation events have been performed, it is clear the data set will evolve to a single ancestor state. Minimizing removals over every possible such "history" then ensures the bound obtained is valid; however, it may not be optimal because of the incomplete consideration of recombination events. Although the proposition greatly reduces the class of histories that need be considered, the number of such histories can become prohibitively large if there are many types and little simplification is possible.

To obtain a local bound for the region between sites i and j, we can construct a data set corresponding to the types of all the sample members at just the sites within that region. A bound B_{ij} will correspond to the subset $\{s_i, s_{i+1}, \ldots, s_j\}$ of mutant loci. Unlike for the haplotype bounds there is no point in taking further subsets of this set, since the bound obtained is obviously increasing in the number of segregating sites in the sample. Thus there is a total of $S(S-1)/2$ subsets to be tried in general. Further, the bound obtained through this method is easily seen to always equal or increase the haplotype bound for the corresponding region; it models the history more carefully. Using Algorithm 1 on the local bounds results in the statistic R_s, which often improves the minimum, through incorporating information about the positions of recombination events along the sequence. For large data sets it is not always possible to calculate R_s, due to a very long computation time, and in this case using the haplotype bound R_h is the only option. This is more likely to be necessary if the number of recombinations required for a history is large. The best approach in practice might be to calculate the (quick) haplotype bound and then try to improve this by using R_s if the original local bounds obtained from this are not too large, certainly no larger than about nine.

We now have a sequence of increasing bounds; for any given data set $R_s \geq R_h \geq R_m$; as we might expect, there seems to be a trade-off between the quality of the bound obtained and the computational time to acquire the bound.

EXAMPLE

The three methods of obtaining local bounds above result in three different bounds, R_m, R_h, and R_s, on inputting these local bounds into Algorithm 1. The following example illustrates the bounds and shows that they are all different in general. Consider a sample of only eight sequences, where the two types at each site are expressed in binary notation and with known ancestral type at each site, denoted by 0:

Site	1	2	3	4
a	0	0	0	0
b	0	1	0	1
c	1	1	0	0
d	0	1	1	0.
e	1	1	1	1
f	1	1	0	1
g	1	1	1	0
h	1	0	0	1

The incompatible site pairs are (1, 2), (1, 3), (1, 4), (2, 4), and (3, 4). This gives an incompatibility matrix B_I as below, and using Algorithm 1 or the original algorithm of Hudson and Kaplan (1985) gives $R_m = 2$, with a solution vector of (1, 0, 1) for the number of recombinations between adjacent sites. Looking at counting types, sites (1, 2, 3) considered together give six distinct types and so a bound of $6 - 3 - 1 = 2$ recombinations in this region; the same is true for sites (2, 3, 4). Sites (1, 2, 3, 4) give eight types and thus a bound of $8 - 4 - 1 = 3$ recombinations, and no subset containing both endpoints 1 and 4 improves this bound. At this point we have obtained a set of local bounds that can be expressed as a matrix B_H, shown alongside the incompatibility matrix B_I:

$$B_I = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad B_H = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{5}$$

Now we can again use Algorithm 1 to find R_h, the minimal number of recombinations subject to the haplotype bounds. This results in a value of $R_h = 3$ recombinations for the whole region, with a minimal solution vector (1, 1, 1) for the number of recombinations between

successive sites. Note that even if we restricted our subset search so the endpoints were separated by at most one site [so the local bound for site pair (1, 4) drops to zero], Algorithm 1 still recovers the same solution since $B_{13} + B_{34} = 3$, and $R_h = 3$ still. This illustrates how the algorithm can improve over using the local bounds alone.

If we use the improved local bounds from searching over sample histories (which will give R_s) the bound matrix becomes

$$B_S = \begin{pmatrix} 0 & 1 & 2 & 4 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (6)$$

For an illustration, the top right bound of 4 (corresponding to the local bound between sites 1 and 4) may be attained by removing sequences *b*, *f*, *h* (viewed as "recombinants" by the algorithm) and then removing the mutation at site 4 (which is a singleton after these three sequences are removed). No further simplification is possible at this point, so sequence *e* is removed from the list. At this point mutations and coalescences can be performed until a single ancestor is reached, so no further recombinations are needed. All other possible simulated histories require at least four recombinations so this is the local bound given for the number of recombinations between sites 1 and 4. Using Algorithm 1 with the bounds of B_S gives a minimum bound of $R_S = 4$ and an optimal solution vector (1, 1, 2) in this case.

In this example we then have $2 = R_m < R_h < R_s = 4$, so the three bounds are all different, and the new methods improve the detection over R_m. In fact, we can easily construct a history with exactly four recombinations in this case, so four is the true minimum. No other algorithm could detect more than four past recombination events.

IMPLEMENTATION OF THE BOUNDS

We describe here some technical aspects of how the bounds derived in COMBINING LOCAL RECOMBINATION BOUNDS and LOCAL RECOMBINATION EVENT BOUNDS are actually implemented. The methods given in this section give ways to reduce the time taken to obtain the corresponding bounds, particularly R_h.

With the haplotype bounds, the number of subsets of the set of sites is very large for a reasonably large number of segregating sites. However, the implementation is aided by the following fact about the number of haplotypes added by a given mutation:

LEMMA 4. *For mutation data S for a sample of size n, define H_S as the number of different types in the sample. For a given mutation labeled s, construct data S′ by removing the mutation data corresponding to site s from the data S. Then*

$$H_{S'} \leq H_S. \quad (7)$$

Further, if the mutation s is compatible with all other segregating sites in the region, S′ is such that

$$H_{S'} \geq H_S - 1. \quad (8)$$

Proof. Suppose the total set of sites is *C*. Then the mutation data over $C - s$ (*i.e.*, without site *s*) partitions the set of *n* sequences into a set of $H_{S'}$ equivalence classes according to which type each belongs to. Adding the mutation at *s* to form the list *C* results in a refinement of these equivalence classes since if two sequences are the same type for every mutation in *C*, they are certainly of the same type for every mutation in $C - s$. The new total number of classes is H_S by definition. Then trivially $H_S \geq H_{S'}$, the first inequality.

To obtain the second, for a contradiction suppose $H_S \geq H_{S'} + 2$. Then at least two classes, E_0 and E_1 say, are split through adding the extra mutation; this follows since the mutation data are binary and so when the list of equivalence classes is refined through adding data at a single mutation site, each previous class is divided into at most two disjoint classes corresponding to types 0 and 1 at the added site. Now since E_0 and E_1 are *different* classes initially, without loss of generality there exists some segregating site *t* at which all members of E_0 are type 0 and all members of E_1 are type 1; in the reverse case we can simply swap the labels of the classes.

Since the mutation at *s* splits both E_0 and E_1, there exist members of both types 0 and 1 at *s* in each set. But then all types 00, 01, 10, and 11 are present in the sample at *t*, *s*, respectively, and these sites form an incompatible pair. This contradicts the assumption of *s* being compatible with *all* other sites, so $H_S \leq H_{S'} + 1$, giving the lemma. ■

COROLLARY 5. *For any given list C of segregating sites, define R_C as the haplotype-based bound* (4) *obtained using the mutation data from all sites in C. Now form a new list C′ by removing from C those sites that are not incompatible with any others. Then*

$$R_{C'} \geq R_C \quad (9)$$

so the set C′ leads to an improved bound compared to C.

Proof. Consider taking the list *C* and removing the nonincompatible sites one at a time. Then from Lemma 4, each successive removal reduces the number of types by either one or zero. However, since each removal reduces the number of segregating sites in the list by exactly one, the haplotype bound 4 is either unchanged or increased by one after each removal, giving the corollary. ■

Thus we need look only at *incompatible* sites to obtain the best bound for a region; so the incompatibility matrix itself is still very valuable in this case. This means we can often greatly reduce the number of site subsets that need considering to obtain local haplotype bounds,

since many sites need never be considered for such subsets. Our implementation calculates R_h by first constructing the incompatibility matrix (and R_m) and then removing those sites that are compatible with all others, before obtaining bounds using subset search over the reduced set. Further use of the condition is made by choosing viable subsets; for example, checking the endpoints is not compatible with the whole of the set. The algorithm also saves time, using a "branch-and-bound" method; only subsets of a size up to the maximum that could improve the current bound obtained for the section are considered, and this maximal size is updated as the current bound increases.

The bounds from simulation of the sample history have also been implemented. Using Algorithm 3, the implementation tries all possible histories, coalescing or removing mutations from lines whenever possible (at removal events the program first tries removing the first remaining sequence, and then the second, and so on until all possibilities are exhausted). It, too, uses a branch-and-bound approach, simulating histories only to the previous best recombination depth for a region.

SIMULATED PROPERTIES

Having developed methods that can potentially detect more recombinations than the existing statistic R_m, a question that is of obvious interest is how much improvement we can expect to see. Another question is the effect of different parameter choices on the value of R_h we obtain. Here we investigate both of these questions under the simple neutral population genetic model of evolution.

The properties of the various statistics giving a minimum number of recombinations are difficult to calculate analytically under even simple models. However, they may be estimated under different population genetic models through simulation. To do this for the neutral model of evolution, standard coalescent simulations (Hudson 1983) were run with various recombination and mutation rates and different sample sizes. A constant population size and no population subdivision were assumed. Data sets were produced using R. R. Hudson's program ms, which simulates mutation data under this model (among others), and the statistical properties of the new minima were estimated. The effect of known or unknown ancestor types was also tested. Under this model, the evolution of a sample of n sequences back in time is governed by a stochastic process, with rates dependent only on the scaled mutation and recombination rates $\theta = 4N\mu$ and $\rho = 4Nr$. Here N is the effective population size and μ, r are the per gene, per generation mutation and recombination rates, respectively.

Figure 1 gives three charts showing the effect of different choices of maximal subset size and maximal number of sites spanned on the new haplotype-based minimum, compared to Hudson's minimum, for the particular parameter choice $\theta = \rho = 10$ and sample sizes $n = 25$, 50, 100. Each number is based on a single data set containing 10,000 simulated samples, with ancestral types assumed unknown. It is clear that increasing the subset size has a diminishing-returns effect; the increase in number of recombination events detected diminishes as the subset size gets bigger. The effect of increasing this subset size on the average minimum obtained appears to be stronger than the effect of increasing the maximal width considered; changing the latter parameter can greatly increase the run time of the program. As might be expected, using larger and wider spanning subsets of the data yields the most benefit for large sample sizes, where more information about the recombination history of the data is available. The run times for all 10,000 data sets were typically of the order of 10 min for subsets of size at most five, spanning at most 12 inconsistent sites. Thus for a single data set the program will run very fast with much less conservative settings; however, to learn properties of the distribution of the statistic for different parameters (and so examine different models in the light of the data, for example), a fast run time is important. Obviously, to get the best bound for a single data set one should use the largest width and subset size possible.

The remaining simulations (except those for Table 2) were run with a maximal subset size of five and maximal width of 12 sites, which seems to be a reasonable compromise for speed and quality. This produces a minimum, which is referred to as R_h from here on. Figure 2 shows two histograms of the distribution of the two statistics with $\theta = 10$, $\rho = 10$, 20 and $n = 50$, on the basis of 100,000 simulated samples and unknown ancestral types. It is clear that the new statistic R_h often detects substantially more recombination than does R_m. In fact, from these runs the statistics coincided 0.33 of the time for $\rho = 5$ (not shown on histogram), 0.09 of the time for $\rho = 10$, and only 0.01 of the time for $\rho = 20$. While both distributions are shifted to the right through increasing ρ, the distribution corresponding to the haplotype minimum shows a more pronounced separation of the peaks of the two distributions across the range of recombination parameters considered. Thus it may be that the new minimum carries more information about the true amount of recombination in the history than does the old.

To investigate the difference between the minima for a wider range of parameter values, Table 1 shows the mean and coefficient of variation (standard deviation divided by the mean) of the two estimates for different values of θ and ρ, for $n = 25$, $n = 100$, respectively, and all ancestral types known (types were assumed unknown for the previous two simulations). Each data point is based on 10,000 samples from the neutral distribution. Run times for the program were strongly dependent on the parameters used and varied from just a few tens of

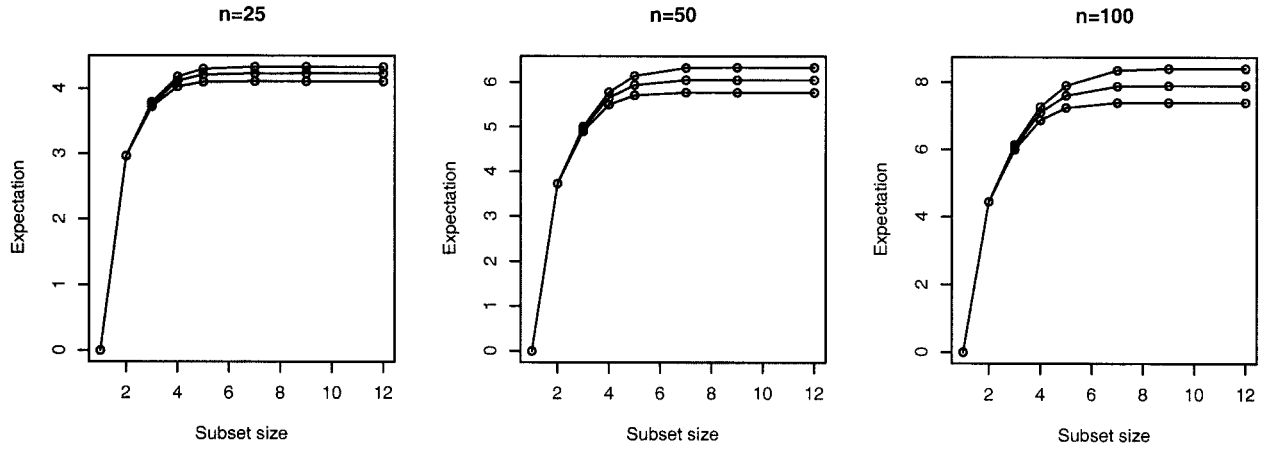

FIGURE 1.—The effect of parameter choice on the expected minimum for $\theta = \rho = 10$. The three lines on each graph correspond to maximal widths for subsets of 10 (bottom line), 12 (middle line), and 15 (top line) incompatible sites; the horizontal axis gives the maximal number of members of each subset. For each graph, the point corresponding to a subset size of 2 shows the expectation of R_m.

seconds for the $\theta = 1$ values to several hours for the $\theta = 100$, $\rho = 100$ case to complete all 10,000 runs (corresponding to a couple of seconds on average for each simulated data set in the latter case). For a given θ and ρ pair the $n = 25$ simulations took around one-quarter of the time of the $n = 100$ simulations. Comparing the expectation of R_h for $\theta = \rho = 10$ with the types-unknown case gives an increase in expectation from 4.21 to 5.28 and a change in scaled relative error from 0.46 to 0.43, for $n = 25$. For $n = 100$ the types-unknown, types-known values are 7.60, 8.70 for expectation and 0.34, 0.33 for scaled relative error. Finally, for the estimator R_m the types-unknown numbers are 2.96, 0.46 ($n = 25$) and 4.45, 0.37 ($n = 100$), which compare to mean and standard errors in the types-known case of 3.65, 0.44 ($n = 25$) and 5.08, 0.36 ($n = 100$). Thus, using information about the ancestral types where it is available can lead to significant improvements in terms of the number of recombinations detected by either statistic, particularly for small sample sizes (where the original ancestral type is less likely to be present).

The most striking point is that the expectation of the new minimum R_h is much more sensitive than R_m to changes in recombination rate; so for a given value of θ and increasing ρ, the gap between the two increases on average. The coefficient of variation, though, remains broadly similar between the two estimates across a wide range of parameter values, with R_h having a slightly smaller relative error compared to R_m. For increasing θ and fixed ρ, n, the expectations of R_m and R_h, will converge to the same limiting value; for a given history this corresponds to the number of different tree topologies across the stretch of DNA and the expectation is the expected number of such topologies. This value can be calculated; it is $\sim$2.04 for $\rho = 1$ and $n = 25$, the only case for which the limit is approached over the range of θ values considered, and in this case there is clear convergence of R_h and R_m for large θ. The limiting value

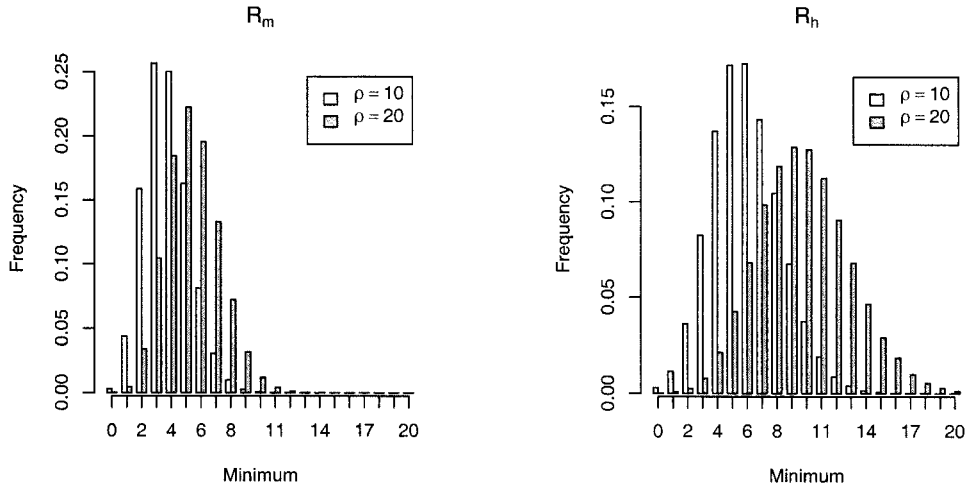

FIGURE 2.—Histograms of R_m and R_h for $\theta = 10$, $n = 50$.

TABLE 1

Estimates of the mean and coefficient of variation (standard deviation divided by the mean) of R_m and R_h, for samples of sizes 25 and 100

		θ													
ρ	R_m, R_h	1		5		10		20		30		50		100	
							$n = 25$								
1	Mean	0.08	0.08	0.49	0.54	0.76	0.85	1.03	1.12	1.19	1.28	1.36	1.43	1.59	1.63
	Coeff. var.	3.63	3.68	1.34	1.38	1.06	1.09	0.91	0.93	0.85	0.86	0.81	0.82	0.76	0.76
5	Mean	0.27	0.31	1.55	2.08	2.45	3.22	3.51	4.34	4.17	4.93	5.07	5.68	6.27	6.66
	Coeff. var.	1.93	2.04	0.70	0.73	0.55	0.54	0.46	0.46	0.44	0.43	0.41	0.41	0.39	0.39
10	Mean	0.41	0.50	2.27	3.40	3.65	5.28	5.37	7.18	6.53	8.27	8.15	9.65	10.43	11.47
	Coeff. var.	1.57	1.70	0.57	0.57	0.44	0.43	0.38	0.36	0.36	0.35	0.33	0.33	0.31	0.31
20	Mean	0.57	0.78	3.11	5.20	5.11	8.16	7.83	11.47	9.69	13.39	12.49	15.93	16.68	19.43
	Coeff. var.	1.34	1.51	0.48	0.48	0.37	0.35	0.31	0.29	0.29	0.28	0.27	0.27	0.25	0.25
30	Mean	0.65	0.92	3.68	6.52	6.14	10.40	9.57	14.90	12.04	17.64	15.64	21.13	21.38	26.01
	Coeff. var.	1.26	1.43	0.44	0.43	0.33	0.31	0.28	0.27	0.25	0.25	0.23	0.23	0.22	0.22
50	Mean	0.75	1.10	4.39	8.30	7.54	13.74	11.96	20.14	15.36	24.31	20.49	29.82	28.85	37.46
	Coeff. var.	1.17	1.37	0.41	0.40	0.30	0.28	0.24	0.23	0.22	0.22	0.20	0.20	0.18	0.19
100	Mean	0.89	1.39	5.40	10.90	9.54	18.85	15.75	29.27	20.58	36.27	28.29	46.04	41.39	60.07
	Coeff. var.	1.11	1.33	0.37	0.37	0.27	0.26	0.21	0.20	0.18	0.18	0.16	0.16	0.15	0.15
							$n = 100$								
1	Mean	0.14	0.15	0.80	0.97	1.21	1.46	1.67	1.93	1.88	2.10	2.19	2.34	2.57	2.65
	Coeff. var.	2.64	2.72	0.98	1.03	0.78	0.80	0.65	0.66	0.63	0.63	0.60	0.59	0.58	0.58
5	Mean	0.44	0.58	2.20	3.58	3.49	5.36	5.06	6.85	6.09	7.69	7.47	8.69	9.38	10.10
	Coeff. var.	1.46	1.63	0.56	0.55	0.43	0.40	0.37	0.35	0.34	0.34	0.32	0.32	0.30	0.30
10	Mean	0.68	1.02	3.20	5.96	5.08	8.72	7.52	11.31	9.20	12.70	11.68	14.57	15.30	17.22
	Coeff. var.	1.19	1.38	0.45	0.43	0.36	0.33	0.31	0.29	0.28	0.27	0.26	0.26	0.24	0.24
20	Mean	0.87	1.48	4.28	9.08	7.04	13.66	10.73	18.17	13.38	20.72	17.24	23.84	23.69	28.60
	Coeff. var.	1.06	1.28	0.39	0.38	0.30	0.28	0.26	0.24	0.23	0.22	0.22	0.22	0.20	0.20
30	Mean	1.02	1.86	5.02	11.45	8.27	17.27	12.91	23.66	16.27	27.19	21.45	31.85	30.04	38.34
	Coeff. var.	1.00	1.22	0.37	0.35	0.28	0.26	0.23	0.21	0.21	0.20	0.19	0.19	0.18	0.18
50	Mean	1.18	2.33	5.94	14.75	10.09	23.10	15.97	32.25	20.59	38.18	27.55	45.35	39.51	55.11
	Coeff. var.	0.92	1.16	0.34	0.33	0.25	0.24	0.21	0.19	0.19	0.18	0.17	0.16	0.15	0.15
100	Mean	1.39	2.95	7.25	19.86	12.68	32.57	20.82	48.06	27.26	58.16	37.54	71.71	55.73	89.91
	Coeff. var.	0.86	1.13	0.31	0.32	0.23	0.22	0.18	0.17	0.16	0.15	0.14	0.13	0.12	0.12

For each set of parameter values, the first column refers to R_m and the second to R_h. Each value is estimated from 10,000 simulated data sets.

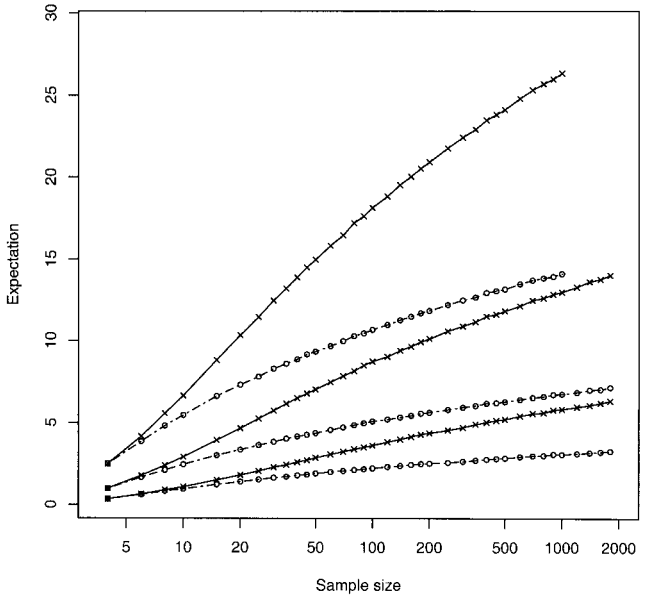

FIGURE 3.—The effect of sample size on the expected minima for three values of $\theta = \rho$. Each point is estimated from 10,000 simulated samples. The dotted lines with circles give values for $\mathbb{E}(R_m)$, and the solid lines with crosses give values for $\mathbb{E}(R_h)$. The three pairs then correspond to $\theta = \rho = 5$ (bottom pair), 10 (middle pair), and 20 (top pair).

here corresponds to detection of ~54% of the expected total number of recombination events in the sample history and is in fact the true minimum given the collection of ancestral tree topologies along the sequence; this is more data than we can possess in practice, so many recombination events will inevitably go undetected. Although the minimum can give a guide to the amount of recombination in the history, any minimum will miss the greater proportion of recombination events that have occurred and thus should not be used directly to estimate the recombination parameter.

For bigger values of ρ, it is clear that θ needs to be very large indeed for the two estimators to become close to one another, and the improvement from using R_h over R_m is greater for the larger sample size of 100 for given parameter values. Increasing the length of DNA sequence for which data exist corresponds, for uniform per base rates of recombination and mutation, to increasing ρ and θ in proportion. For such increases, the expected values of both minima increase approximately linearly with the parameters, and the ratio of the two remains approximately constant (though dependent on the ratio of recombination to mutation rates).

The effect of increasing the sample size on the statistics is also of interest; Figure 3 shows the expected value of each statistic as a function of n for different values of $\rho = \theta$, and ancestral types are assumed known. Each data point is based on 10,000 samples. The expected number of recombinations detected increases slowly with sample size in both cases, with the new minimum seeming to perform better relative to R_m as n increases; the ratio of the two expectations is increasing with the sample size. The increase in detection is, however, slow in both cases; this reflects the fact that the total number of recombination events in the sample history increases only as $\rho \log n$ with sample size n and that mutations in the correct portions of the ancestry are needed to detect these few extra recombinations.

TABLE 2

Estimates of the means of the different minima for samples of size 10

		ρ					
θ	Statistic	1	5	10	20	50	100
5	$\mathbb{E}(R_m)$	0.27	0.97	1.48	2.09	2.97	3.64
	$\mathbb{E}(R_h)$	0.28	1.10	1.76	2.64	3.97	5.01
	$\mathbb{E}(R_s)$	0.29	1.13	1.83	2.74	4.08	5.11
10	$\mathbb{E}(R_m)$	0.43	1.58	2.45	3.54	5.24	—
	$\mathbb{E}(R_h)$	0.45	1.79	2.92	4.45	7.00	—
	$\mathbb{E}(R_s)$	0.46	1.89	3.10	4.70	7.30	—

Dashes indicate where data were not generated due to a very long expected run time. The data were generated assuming known types with $n = 10$ and a maximal subset size of 12 and width 12 for the R_h estimates. For each set of parameters, all three expectations are estimated from the same simulated data set.

The statistical properties of the history-based bound are much more difficult to estimate, since the bound is impossible to compute in the current implementation when the number of detected recombinations is large, although for smaller obtained bounds, it may be rapidly computed. This means that the algorithm tends to "freeze" on occasional data sets, even when the simulation parameters are quite moderate. Thus unfortunately the simulation results of Table 2 correspond to only 10 sequences and varying rates of recombination and mutation. They compare the means of all three types of bounds, with types known, across this range of parameters in the model. The maximal subset size and widths for the R_h algorithm were both chosen here to be 12, to observe as many recombinations as possible through number of haplotypes only, for the purposes of the comparison. The mean is a reasonable summary of the difference between the minima since for any sample, $R_m \leq R_h \leq R_s$. The bounds from generation of the history always took much longer to obtain than the haplotype-based bounds (typically many hours for R_s, compared to <5 min for R_h, for the full 10,000 simulated samples). It appears from the simulations that the minimum based on simulating histories offers little improvement over the haplotype-based bound; however, this may be misleading since the improvement offered is probably much greater for larger sample sizes, where there is more information about the unobserved history.

TABLE 3

The expectation and coefficient of variation for R_h for a model with a recombination hotspot in the center of the region (for details see text) and uniform mutation rates of 5, 10, and 20

		Subset size			
θ	Statistic	2	3	5	12
5	$\mathbb{E}$ (R_h)	1.48	2.20	2.93	3.00
5	Coeff. var.	0.55	0.51	0.55	0.57
10	$\mathbb{E}$ (R_h)	2.21	3.17	4.24	4.40
10	Coeff. var.	0.49	0.42	0.39	0.41
20	$\mathbb{E}$ (R_h)	3.35	4.46	5.55	5.70
20	Coeff. var.	0.45	0.39	0.35	0.35

The sample size is fixed at 50 and the ancestral type is assumed known at each locus. The columns correspond to different subset parameters for the bound; thus the first column actually refers to R_m; for the other columns, a maximal subset width of 12 was used. For each parameter set, 10,000 simulated samples were created and the estimates for the different widths are all based on this same data set.

This is reflected in the fact that the improvement in detection over R_m is only moderate for $n = 10$.

The theoretical behavior of the different bounds relative to one another depends on the number S of segregating sites in the sample. If $S = 2$, Hudson and Kaplan's minimum R_m is obviously the optimal lower bound for the number of recombinations. If $S = 3$, this is no longer true ($R_m \leq 2$ but $R_h \leq 4$ here); however, a search over all possible sets of genes with this number of segregating sites reveals that the haplotype bound R_h is in fact always equal to the bound R_s; though a history for the sample requiring exactly R_h recombinations cannot necessarily be constructed. This does not hold for $S \geq 4$, since the history-based bound R_s is not equal to R_h in some cases here; it is not clear whether R_s is optimal for these four sites (there are many possible sets of genes to look at in this case).

The next simulations here briefly consider what can happen under a different model of recombination. They give the means for the bound R_m of Hudson and Kaplan and the new haplotype bound R_h, in a model where there is variability in the recombination rate across the region: a hotspot of recombination with an otherwise uniform recombination rate. In fact, the simulations use a model where the central 10% of the region has a recombination rate of 100 times the average elsewhere (having a cumulative recombination rate $\rho = 10$), and the overall recombination rate is 10.9 for the whole length of the sequence. Uniform mutation rates of 5, 10, and 20 across the sequence were then used, with a sample size of 50. Table 3 gives the results under this model for different subset sizes. It shows that with these parameters, the improvement in expected minimum over R_m (the first column) of the new method is about a factor of two, corresponding to detection of twice as many recombinations within the region, using the largest subset size of 12. This larger-than-usual improvement (compared with a flat recombination rate) corresponds to the fact that the new minimum often detects multiple recombinations within the hotspot, by looking across it. The coefficients of variation seem fairly similar except for the case where the mutation rate is 20; here the coefficient of variation of the new minimum is somewhat lower. The fact that looking across the hotspot leads to increased detection is also confirmed by the increase in detection that occurs as the maximal subset size increases through 3, 5, and 12.

Finally, one unexplored question is the effect of departures from the given assumptions on the results obtained for the minimum. All the statistics here will give a true minimum under any model of selection, recombination rates, or population structure (although their properties will be strongly dependent on the particular model chosen), provided the two assumptions of no gene conversion and no repeat mutations hold. However, if these are violated, then such events in the history can be mistaken for recombinations by the algorithms, and so the sensitivity to departures from the given assumptions is important. Here we look only briefly at the repeat mutation case. Data were simulated under a neutral model, for 50 sequences with a uniform recombination rate of 0 and 10 across the region, in a finite-sites model with 10,000 sites. The scaled mutation rate was chosen to be 10 across the region; however, on top of this underlying rate 5 or 10 randomly selected sites were chosen to be hypermutable with mutation rates 200 times the regional average. This corresponds to an expected number of 0.90 mutations per hypermutable site; so a proportion of such sites will undergo repeat mutation. The results of Table 4 demonstrate that under this moderate amount of repeat mutation, some multiple mutation contributes to the minimum number of recombinations, so that even if $\rho = 0$ there is some false "detection." We can measure the number of such erroneous detections by the increase over the (in parentheses) expected number of recombinations detected with no repeat mutations but the same rates. It seems that the erroneous detections are almost as great for a subset size of 2 (*i.e.*, R_m) as for R_h with a subset size of 12. Thus the relative contribution of the repeat mutations is reduced slightly with the new minima. The relative errors seem almost unaffected by the hypermutable sites in the $\rho = 10$ case.

Although this is encouraging, it should be noted that in theory the absolute contribution of such sites can be worse for the new minimum than for R_m, if they have mutated more than twice. This means that sites with extremely high mutation rates could have serious biasing effects on R_h; however, these may be visible in practice through a lack of linkage disequilibrium with nearby sites, or more than two types may be present at these sites. If such a site is strongly suspected to be hypermuta-

TABLE 4

The expectation and coefficient of variation for R_h for a model with 5 or 10 hypermutable sites in the region (for details see text) and an otherwise uniform mutation rate of 10, with ρ = 0 or 10

ρ	No. hypermutable	Statistic	Subset size 2	3	5	12
0	5	$\mathbb{E}$ (R_h)	1.37 (0.00)	1.51 (0.00)	1.53 (0.00)	1.53 (0.00)
		Coeff. var.	1.07 (0.00)	1.10 (0.00)	1.11 (0.00)	1.11 (0.00)
	10	$\mathbb{E}$ (R_h)	2.43 (0.00)	2.70 (0.00)	2.77 (0.00)	2.77 (0.00)
		Coeff. var.	0.80 (0.00)	0.82 (0.00)	0.82 (0.00)	0.82 (0.00)
10	5	$\mathbb{E}$ (R_h)	5.30 (4.64)	6.92 (6.12)	8.21 (7.33)	8.38 (7.49)
		Coeff. var.	0.39 (0.38)	0.37 (0.37)	0.36 (0.35)	0.35 (0.35)
	10	$\mathbb{E}$ (R_h)	6.14 (4.89)	7.93 (6.44)	9.28 (7.67)	9.46 (7.81)
		Coeff. var.	0.39 (0.37)	0.37 (0.36)	0.36 (0.35)	0.35 (0.35)

The sample size is fixed at 50 and the ancestral type is assumed known at each locus. The columns again correspond to different subset parameters for the bound; thus the first column actually refers to R_m; for the other columns, a maximal subset width of 12 was used. For each parameter set, 10,000 simulated samples were created and the estimates for the different widths are all based on this same data set. The term in parentheses in each column gives the estimated number under an infinite-sites model with the same recombination rate and the same overall mutation rate (including the contribution from the hypermutable sites).

ble, the best course might be to remove it before calculating the minimum.

LPL DATA APPLICATION

As an application of the new minimum, we examined data corresponding to 9.7 kb of genomic DNA sequence from the human LPL gene. For this data, we calculated two lower bounds on the number of recombinations in the sample history: R_m of Hudson and Kaplan (1985) and one of the new statistics, R_h, derived in this article. R_h was calculated using the methods described in combining local recombination bounds and local recombination event bounds, and the associated bound matrix R for the data was also calculated. Algorithm 1 was used to give R, which for every pair of sites gives a lower bound for the number of recombination events in the sample history that occurred between those two sites. R_h is the bound for the whole region, corresponding to the endpoint sites.

The 9.7-kb region was sequenced by Nickerson *et al.* (1998) in 71 individuals from Jackson, Mississippi; North Karelia, Finland; and Rochester, Minnesota. Here we look at the data for each location in turn, as well as the combined data set. Sites where phase was not determined (at frequencies of 1 or 2 in the whole sample) were excluded, as were sites where the type was not determined for every individual in the sample. It should be possible to modify the algorithms so that missing data of the latter type can be handled, but this has not yet been implemented.

The amount of recombination detectable for this data set has previously been analyzed by Clark *et al.* (1998), using Hudson and Kaplan's R_m, and Templeton *et al.* (2000a), who used a parsimony-based method to infer likely recombinations. Their method is not based on finding a minimal number of recombinations, but on identifying sample members who seem likely to be recombinants through a statistical approach. This method suggested that 29 recombination events were detectable in the history of the larger sample, which includes all three populations. Further, Templeton *et al.* (2000a,b) concluded that repeat mutation might have a significant role to play in the ancestry of the region. In particular, they suggested that repeat mutations were likely to be common in CpG base pairs. This conclusion was questioned by Przeworski and Wall (2001), who found that with plausible higher mutation rates at CG dinucleotides, the prior expectation was that there should be only around one repeat mutation event at the CpG sites in the sample. This issue is of interest here, since such repeat mutations can be misinterpreted as recombination events. Recent work by Fearnhead and Donnelly (2002) again suggests that repeat mutations are unlikely to have played a major role in the evolution of the sample and specifically that it was likely that only around one to five repeat mutations were in the sample history.

Templeton *et al.* (2000a) found that the 29 recombination events found by their method were clustered strongly near the center of the region, approximately between sites 2987 and 4872, and suggested that this was due to an elevated rate of recombination in this area. If this is the case, we would expect a similar clustering of the recombination events detected using the new methods between sites 2987 and 4872. Of the recombination events suggested by Templeton *et al.* (2000a), 21 of them fell within the suggested hotspot and only 8 outside. Clark *et al.* (1998) looked at the spread of recombination events detected using the algorithm of Hudson and Kaplan (1985) and did not find any such

TABLE 5

The number of detected recombination events for the three data sets in the different site ranges, calculated using R_h and Algorithm 1

	Site range			
Region	106–2987	2987–4872	4872–9721	Full region
Jackson	10 (0.00347)	9 (0.00477)	13 (0.00268)	36 (0.00374)
Finland	2 (0.00069)	13 (0.00690)	11 (0.00227)	27 (0.00281)
Rochester	1 (0.00035)	13 (0.00690)	7 (0.00144)	21 (0.00218)
Combined	12 (0.00417)	22 (0.01167)	28 (0.00577)	70 (0.00728)

Pairs of entries give the number of detections and (in parentheses) the detections divided by the relevant distance. The middle interval (sites 2987–4872) corresponds to the suggested recombination hotspot.

clustering. Although TEMPLETON *et al.* (2000a) argued that this was due to false positives caused by repeat mutations, an alternative explanation is that R_m simply could not pick up the increased amount of recombination within a hotspot by looking only at incompatible pairs.

The minimum number of recombination events found using R_h for each geographic location and the distribution of detections within the region sequenced are summarized in Table 5. The statistic was calculated using a maximal set width and subset size of 25 each (see *Haplotype bounds* in LOCAL RECOMBINATION EVENT BOUNDS for details on these parameters, the increasing of which improves the quality of the bound produced). Table 5 also shows the number of detections *per site* for the different regions for each location and the combined data, calculated by simply dividing the number of detections by the distance between sites. For every region the density of detection is higher in the suggested hotspot, although the signal is weaker for the Jackson data set (for which a number of sites within the hotspot were removed due to missing data). For the Finland and Rochester data, this increase in detection is very substantial, and for the overall data there is support for more recombination in the central region.

The new minima are all much higher than the corresponding R_m's, which vary between 13 and 17 for the three data sets, with an overall R_m of 22 for the combined data. This compares to $R_h = 70$ when the data from all three regions are taken into account, so the increase in detection for this data set is very substantial. One important proviso is that this could be an overestimate of the minimum number if there has been repeat mutation; however, each repeat mutation event could be mistaken for only two recombinations at most. The Jackson data differ by having more detection at the 3′ end of the sequence and less within the hotspot than the other areas. However, there is an additional signal for more detectable recombination within the central region when we consider pairs of sites spanning sites 2987–4872.

In fact, Figure 4 shows the number of detections between all pairs of sites, scaled by the distance between sites, for all three regions and for the combined data. This is a scaled version of the matrix R calculated as in Algorithm 1. There is a clear tendency for increased density of detection between pairs of sites where at least one site is within the central zone (between the white and black pairs of lines) or the sites span this region (above and to the right of the black lines). This increase is sustained even for sites some distance either side of the central region. If there were an overwhelming amount of repeat mutation (unless this was clustered in certain regions) it would probably be expected to cloud rather than enhance any signal of increased detection in certain regions, by causing false detections throughout the region; thus the results here may be a conservative estimate.

DISCUSSION

Assuming no repeat mutation, and with no gene conversion, the new minima R_h and R_s presented here can give substantially improved bounds over the incompatibility-based measure R_m of the minimum number of recombinations in the history of a sample. The bounds developed here use more information about the sample history than does the pattern of pairwise incompatibilities. Moreover, with appropriate parameter choices the bound R_h is extremely fast to compute, while still giving a good quality of bound; R_s always gives an equal or improved bound, for data sets sufficiently small for it to be used. Compared to Hudson and Kaplan's R_m statistic the improvement is greatest when there is a substantial amount of recombination in the history of a sample or a large sample size. The new bounds are not the best that might be theoretically obtained since it is not always possible to construct a data history with R_h or R_s recombinations; however, they may come close to this optimum for real data samples.

These minima can be used to obtain an instant visual impression of where and how much recombination is

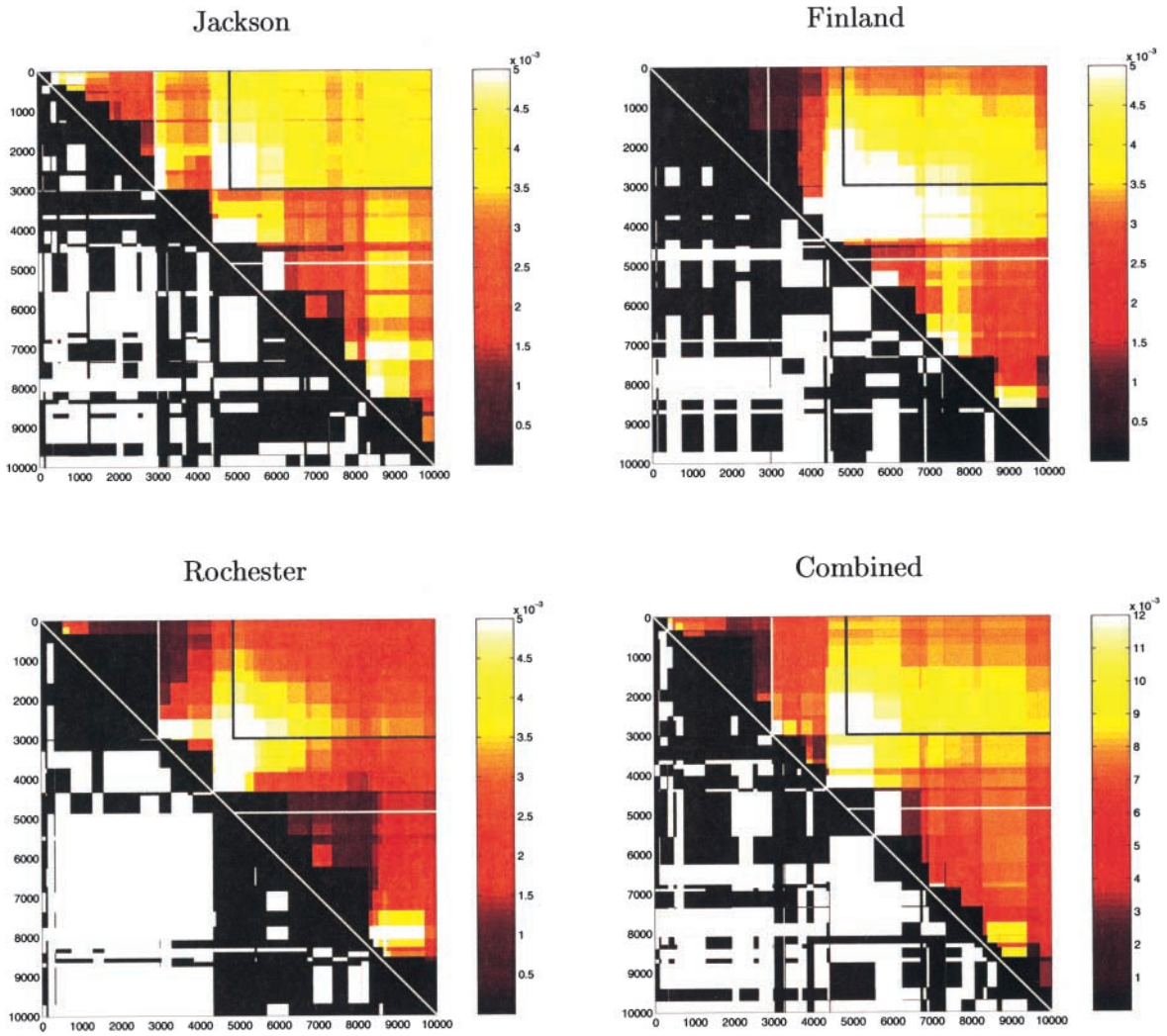

FIGURE 4.—Density of detected recombinations for the different regions. For each plot, the incompatibility matrix is shown below the diagonal. Above the diagonal the matrix R is shown as calculated by Algorithm 1, with each entry scaled by the distance between sites. Thus the color at the point (x, y) shows the number of recombinations detected between sites x and y divided by the distance between x and y. The black and white lines show the boundaries of the suggested hotspot region (colors chosen for contrast). Points above and to the right of these lines correspond to pairs of sites within or across the central region.

occurring in a given region. The pairwise incompatibility matrix is still important in this respect; it gives a set of intervals in which we *must* put recombination events to construct any history of the data. Viewing any system of bounding Equations 2 as a generalized incompatibility matrix, recombination hotspots may be indicated by large recombination bounds across the hotspot location itself compared to neighboring regions. This picture is enhanced when Algorithm 1 is employed to give a matrix bounding the number of recombination events between every pair of segregating sites. Scaling each entry by the distance between sites gives the number of detected events per site, and the resulting matrix can be viewed graphically (Figure 4). The idea of this scaling is that the number of detections should increase approximately linearly with distance if the recombination rate is constant across a region. Although this is obviously not perfect, the amount of visible recombination is subject to stochastic variation, and the distribution of segregating sites within the region will also affect the pattern of detection; we would on the whole expect the detections per site to be increased both within and across any hotspot region. This should provide a clearer signal than incompatibilities alone would give, before the use of more sophisticated methods (for example, simulation studies or a likelihood analysis), to check whether an observed pattern was really significant in a statistical sense. The development of such methods will be an important tool in the reliable detection of local heterogeneity in recombination rates.

The method was applied to a recent data set of 9.7 kb from the lipoprotein lipase gene. The new method detected more than three times the amount of recombination as R_m, subject to the question of whether repeat mutations have an important role to play in this region. Further, the pattern of detection across the region

showed substantial support for the presence of an elevated recombination rate in the central region, which has been suggested by other authors; this was most strongly signaled between pairs of sites encompassing this region, while the signal within the region itself was more variable among the data sets from different geographic locations. Although the possibility of repeat mutation is a concern, the presence of a few repeat mutation events would not bias the minima too severely, since many recombination events can be inferred for these data.

It is important to note, however, that the new bounds and perhaps any parsimony bound on the number of recombinations in the history will miss the majority of recombination events in a sample history for realistic mutation rates; most recombinations will go undetected. Thus the minimum should not be taken as an indicator of the number of recombinations that really occurred in the history. This is a result of the fact that many recombinations do not create a novel haplotype in the population across the region of consideration, so may be omitted from a parsimonious reconstruction of the history. Thus if we wish to estimate the rate of recombination ρ itself in the region, the full-likelihood methods introduced and developed by Griffiths and Marjoram (1996a), Nielsen (2000), Kuhner *et al.* (2000), and Fearnhead and Donnelly (2001) will be the best if the data set is suitable for their use. For larger data sets where this is currently impossible, one approach suggested and implemented by Wall (2000) estimates the rate of recombination by performing maximum-likelihood estimation on summary statistics of the data. This can be much quicker than performing the full-likelihood inference and enables properties of the estimators produced to be investigated, as long as the summary statistics used can be computed very rapidly for a given data set. The increased sensitivity of the new minima to the recombination rate, in tandem with the speed of computation, may mean that in conjunction with others they are useful statistics for incorporation into such a scheme, and future work will investigate this possibility.

Both of the two minima R_h and R_s introduced here use an algorithm that can be used to combine a set of local bounds on recombination events, to produce an optimal overall bound. This framework is profitable since *any* valid method of bounding can be immediately incorporated into the system. Improvements over the two new methods presented here of obtaining these bounds will lead to improvements in the overall minimum given; one obvious improvement over R_h might be a hybrid of the history and haplotype-based bounds, with the bound method choice dependent on the particular subset being considered. The method is very flexible to changing the particular model of recombination points being considered and should be capable of gaining reasonable bounds under a wide range of such models; of particular interest for human data may be models where there is much more recombination than average at particular areas within the region. As mentioned above, the recombination bounds are able to give at least some indication of the comparative level of recombination across such hotspots relative to elsewhere in the region.

Another question that needs studying is the effect of departures from the given assumptions, no repeat mutations and no gene conversion, on the minima produced. The example here shows that repeat mutation can lead to erroneously inferred recombination events. More generally, if either can occur, the obtained minimum may not be a true lower bound on recombination events, and so the sensitivity to some violation of these assumptions is of importance. Where the assumptions do not hold true, R_m and R_h are still valid statistics for performing inference, as long as the appropriate true underlying model of the process is used for judging the values given, but such departures may reduce the information they carry about the parameters of interest. In principle it should be possible to adapt the approach used here to include the possibility of inferring repeat mutations at certain sites (instead of recombination events), although this has not yet been attempted; this might make it possible to extract more information about the recombination history of a sample where a significant number of sites may have mutated more than once.

The calculations of R_m, R_h, and R_s for this article were performed by a C program called RecMin.c, which implements all three statistics and is available from http://www.stats.ox.ac.uk/mathgen/programs.html.

We thank Carsten Wiuf, Jotun Hein, and one anonymous referee for very helpful comments and criticisms, and Gil Mcvean for supplying one of the simulation programs. This work was supported in part by an Engineering and Physical Sciences Research Council doctoral studentship.

LITERATURE CITED

Clark, A. G., K. M. Weiss, D. A. Nickerson, S. L. Taylor, A. Buchanan *et al.*, 1998 Haplotype structure and population genetic inferences from nucleotide sequence variation in human lipoprotein lipase. Am. J. Hum. Genet. **63:** 595–612.

Fearnhead, P., and P. Donnelly, 2001 Estimating recombination rates from population genetic data. Genetics **159:** 1299–1318.

Fearnhead, P., and P. Donnelly, 2002 Approximate likelihood methods for estimating local recombination rates. J. R. Stat. Soc. Ser. B **64:** (Pt. 4): 1–24.

Griffiths, R. C., and P. Marjoram, 1996a Ancestral inference from samples of DNA sequences with recombination. J. Comput. Biol. **3:** 479–502.

Griffiths, R. C., and P. Marjoram, 1996b An ancestral recombination graph, pp. 257–270 in *IMA Volume on Mathematical Population Genetics*, edited by P. Donnelly and S. Tavaré. Springer-Verlag, Berlin.

Gusfield, D., 1991 Efficient algorithms for inferring evolutionary trees. Networks **21:** 19–28.

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., 1983 Properties of a neutral allele model with intragenic recombination. Theor. Popul. Biol. **23:** 183–201.
Hudson, R., and N. Kaplan, 1985 Statistical properties of the number of recombination events in the history of a sample of DNA sequences. Genetics **111:** 147–164.
Kuhner, M. K., J. Yamato and J. Felsenstein, 2000 Maximum likelihood estimation of recombination rates from population data. Genetics **156:** 1393–1401.
Nickerson, D. A., S. L. Taylor, K. M. Weiss, A. G. Clark, R. G. Hutchinson *et al.*, 1998 DNA sequence diversity in a 9.7-kb region of the human *lipoprotein lipase* gene. Nat. Genet. **19:** 233–240.
Nielsen, R., 2000 Estimation of population parameters and recombination rates from single nucleotide polymorphisms. Genetics **154:** 931–942.
Przeworski, M., and J. D. Wall, 2001 Why is there so little intragenic linkage disequilibrium in humans? Genet. Res. **77:** 143–151.
Templeton, A. R., A. G. Clark, K. M. Weiss, D. A. Nickerson, E. Boerwinkle *et al.*, 2000a Recombinational and mutational hotspots within the human lipoprotein lipase gene. Am. J. Hum. Genet**. 66:** 69–83.
Templeton, A. R., K. M. Weiss, D. A. Nickerson, E. Boerwinkle and C. F. Sing, 2000b Cladistic structure within the lipoprotein lipase gene and its implications for phenotypic association studies. Genetics **156:** 1259–1275.
Wall, J. D., 2000 A comparison of estimators of the population recombination rate. Mol. Biol. Evol. **17:** 156–163.

Communicating editor: J. Hey

APPENDIX A: PROOF OF ALGORITHM 1

Before proving the optimality of the algorithm, we derive a few results that are needed in the proof. The first proposition gives an alternative algorithm that can be used to find an optimal solution vector.

Proposition 6. *An optimal vector for the system of* (2) *is given by the vector* $(r_1^*, r_2^*, \ldots, r_{S-1}^*)$ *generated by the following algorithm.*

1. *Let* $k = 1$.
2. *(Maximization step) Set* $r_k^* = \max\{B_{i(k+1)}: i = 1, 2, \ldots, k\}$.
3. *(Updating step) For* $i = 1, 2, \ldots, k$ *and* $j = k + 2, k + 3, \ldots S$, *subtract* r_k^* *from* B_{ij}.
4. *(Incrementing step) If* $k < S - 1$, *increment* k *by* 1 *and go to step* 2.

Proof. We prove the result using the following lemma, which gives the proposition immediately when applied to the case $k = S - 1$. ■

Lemma 7. *For any* $k = 1, 2, \ldots, S - 1$ *the vector* $(r_1^*, r_2^*, \ldots, r_k^*)$ *obtained using the algorithm of Proposition* 6 *minimizes the sum* $R_k = \sum_{l=1}^{k} r_l$ *subject to those conditions of* (2) *satisfying* $j \leq k + 1$.

Proof. We use induction on k. The lemma is clearly true for $k = 1$ as in this case, only a single bound B_{12} is used to give r_1^*, and $r_1^* = B_{12}$ is clearly optimal with respect to this bound. Further, the only condition of (2) satisfying $j \leq 2$ is on B_{12}.

Suppose the lemma is true up to and including the $(k - 1)$th case. Then consider evaluating r_k^* using the algorithm of the proposition. We must show that

1. The vector $(r_1^*, r_2^*, \ldots, r_k^*)$ satisfies all the conditions of (2) with $j \leq k + 1$.
2. The vector minimizes R_k over vectors satisfying these conditions.

For condition 1, note that by the inductive hypothesis, only bounds corresponding to elements $B_{i(k+1)}$ may fail to be satisfied. By the algorithm definition, we subtract r_l^* from such a term $B_{i(k+1)}$ whenever $i \leq l \leq k - 1$. Then as we go into the iteration to obtain r_k^*, the original $B_{i(k+1)}$ is replaced by $B'_{i(k+1)} = B_{i(k+1)} - \sum_{l=i}^{k-1} r_l^*$ in the current system. Hence the original condition will be satisfied if and only if $r_k^* \geq B'_{i(k+1)}$. But by definition $r_k^* = \max\{B'_{i(k+1)} : i = 1, 2, \ldots, k\}$ and thus the original conditions are satisfied for all i, giving 1. Further, there exists m such that $r_k^* \geq B'_{m(k+1)}$, implying

$$\sum_{l=m}^{k} r_l^* = B_{m(k+1)}. \tag{A1}$$

For condition 2, consider any vector $(r_1, r_2, \ldots, r_k)$ satisfying the conditions defined in the lemma. Now since $(r_1^*, r_2^*, \ldots, r_{m-1}^*)$ is minimal (taking this to be the empty vector if $m = 1$) subject to a subset of these conditions, we have

$$\begin{aligned}\sum_{l=1}^{k} r_l &= \sum_{l=1}^{m-1} r_l + \sum_{l=m}^{k} r_l \\ &\geq \sum_{l=1}^{m-1} r_l^* + B_{m(k+1)} \\ &= \sum_{l=1}^{k} r_l^*,\end{aligned}$$

where the last line follows from (A1). Thus $(r_1^*, r_2^*, \ldots, r_k^*)$ minimizes R_k subject to the appropriate bounds and the lemma follows by induction. ■

The lemma is useful in itself, since it means that for any k, our solution minimizes the number R_k of recombination events up to site $k + 1$, subject to the set of bounds. The algorithm of Proposition 6 acts by sweeping through the equation system from left to right, and at the maximization step setting successive r_k^* values equal to the smallest value needed to satisfy all the conditions whose last term is r_k, the "minimum necessary" at each stage. Then we decrease the remaining bounds where there is a term in r_k by r_k^*, reflecting the fact that r_k has already been chosen, in the updating step. The choice of starting from the left rather than from the right is arbitrary; a right-to-left algorithm would lead to the same minimum, although the actual solution itself might be different as there is no guarantee of uniqueness. If we arrange the B_{ij}'s in (upper triangular) matrix form, then the algorithm results in a sweep across the columns from left to right, at each step maximizing for the current column and then decrementing the

appropriate entries to the right of this column. Algorithm 1 offers a faster and more comprehensive solution to the system, so is preferred; here we use Proposition 6 as a stepping stone to derive the better method. In fact, Lemma 7 now has the following corollary, which will be useful in the proof.

COROLLARY 8. *The minimum bound R_n satisfies the following equality:*

$$R_n = \max_{1=i_1<i_2<\ldots<i_k=n} \left(\sum_{j=1}^{k-1} B_{i_j i_{j+1}} \right).$$

Proof. Consider any collection $1 = i_1 < i_2 < \ldots < i_k = n$. Since the number of recombinations in the interval (i_j, i_{j+1}) cannot be below $B_{i_j i_{j+1}}$ and the intervals (i_j, i_{j+1}) and (i_l, i_{l+1}) are disjoint for $j \neq l$, we must have

$$R_n \geq \sum_{j=1}^{k-1} B_{i_j i_{j+1}}.$$

Maximizing over all such collections gives

$$R_n \geq \max_{1=i_1<i_2<\ldots<i_k=n} \left(\sum_{j=1}^{k-1} B_{i_j i_{j+1}} \right). \tag{A2}$$

Now note that in the proof of the lemma above we work along the columns $B_{\cdot k}$ from left to right. At the kth column we always choose some constraint $B_{m_k k}$, where $m_k < k$ to be tight for the minimal solution produced. Then for the particular case $k = n$ we may write

$$R_n = B_{m_n n} + R_{m_n}$$

for some $1 \leq m_n < n$.

Define $j_1 = n$ and recursively $j_{l+1} = m_{j_l}$ for $l \geq 1$. This will produce a strictly decreasing series of integers bounded below by 1 and commencing at n; hence, for some $s \leq n$, $j_s = 1$. Then for any $l \leq (s - 1)$ we can similarly write

$$R_{j_l} = B_{j_{l+1} j_l} + R_{j_{l+1}}$$

and so

$$R_n = \sum_{l=1}^{s-1} B_{j_{l+1} j_l}$$

since $R_1 = 0$. But $1 = j_s < j_{s-1} < \ldots < j_1 = n$ and thus

$$R_n = \sum_{l=1}^{s-1} B_{j_{l+1} j_l} \leq \max_{1=i_1<i_2<\ldots<i_k=n} \left(\sum_{j=1}^{k-1} B_{i_j i_{j+1}} \right). \tag{A3}$$

Equations A2 and A3 together give the required result. ■

Thus we can write the optimal solution as a sum of bounds corresponding to disjoint intervals. Note that this decomposition is not necessarily unique. Since these bounds are exactly satisfied by one optimal solution, they must be satisfied in all optimal solutions.

Proof of Algorithm 1. Note that the algorithm again works by moving along columns from left to right, and each column of R is assigned as that column is reached. The algorithm recursively assigns

$$R_{jk} = \max\{R_{ji} + B_{ik} : i = j, j + 1, \ldots, k - 1\}. \tag{A4}$$

For the purposes of induction, suppose this results in optimal $R_{\cdot 2}, \ldots, R_{\cdot(k-1)}$ so the first $k - 1$ columns of R are set correctly. This is certainly true for $k = 3$ since clearly $R_{12} = B_{12}$ as set by the algorithm. By definition R_{jk} minimizes

$$\sum_{l=j}^{k-1} r_l \tag{A5}$$

subject to the bounds of (2). Now those bounds B_{pq} with $p < j$ or $q > k$ can obviously always be satisfied by setting r_p or r_q, respectively, to be large enough, and such terms will not contribute to the sum of (A5). Then in fact minimizing this sum subject to the whole set of bounds is equivalent to minimizing subject to those constraints corresponding to B_{pq}, where $j \leq p < q \leq k$. This results in an ILP of the same form as the original system; so we may apply Corollary 8 to this new problem to give immediately

$$\begin{aligned} R_{jk} &= \max_{j=i_1<i_2<\ldots<i_l=k} \left(\sum_{m=1}^{l-1} B_{i_m i_{m+1}} \right) \\ &= \max_{j\leq i\leq k-1} \left(\max_{j=i_1<i_2<\ldots<i_l=i} \left(\sum_{m=1}^{l-1} B_{i_m i_{m+1}} \right) + B_{ik} \right) \qquad \text{(A6)} \\ &= \max_{j\leq i\leq k-1} \left(R_{ji} + B_{ik} \right), \qquad \text{(A7)} \end{aligned}$$

where (A6) follows by repeating the previous argument with k replaced by i. But we know that $R_{\cdot i}$ is set correctly by the algorithm for all $i \leq k - 1$ and now comparing with (A4), we can see the algorithm assigns each R_{jk} and hence $R_{\cdot k}$ correctly also. Then by induction, the whole matrix R produced by the algorithm is correct. ■

If we seek a particular solution vector that is optimal, it is clear from the proof of the algorithm that we recover the solution given by Proposition 6 by subtracting the consecutive elements of $R_{1\cdot}$ and setting $r_j = R_{1(j+1)} - R_{1j}$ for $j = 1, 2, \ldots, S - 1$.

APPENDIX B: PROOF OF PROPOSITION 2

Proof. Consider any history $\mathcal{H}$ for the sample $\mathcal{S}$, consisting of an ancestral graph for the sample, with mutation events placed on the branches, resulting in the sample configuration seen in the present. We can construct a history for $\mathcal{S}'$ from this as follows. If $\mathcal{S}'$ is formed from $\mathcal{S}$ through a coalescence of two type a sample members, then a history $\mathcal{H}'$ for $\mathcal{S}'$ is created from $\mathcal{H}$ by choosing some sample member j of $\mathcal{S}$ of type a and removing material only ancestral to j from the graph. This gives a history for a reduced set of sequences, of

types identical to the members of $\mathcal{S}'$, which may then be viewed as a possible history of $\mathcal{S}'$. Otherwise, $\mathcal{S}'$ is formed from $\mathcal{S}$ through the removal of a noninformative mutation at m say from $\mathcal{S}$. Since by assumption mutations arise only once in the history, there exists a unique branch in $\mathcal{H}$ containing a mutation at m. Removing this mutation from the appropriate branch will result in a history for a sample with types identical to the members of $\mathcal{S}$ except at m, where there are no mutants, and so this adapted history may again be regarded as a history for $\mathcal{S}'$. Now $\mathcal{H}$ and $\mathcal{H}'$ contain the same number of recombination events; minimizing this number of recombinations over all possible histories $\mathcal{H}$ immediately gives $R_{\mathcal{S}'} \leq R_{\mathcal{S}}$.

Conversely, suppose we have a history $\mathcal{H}'$ for the sample $\mathcal{S}'$. Then we may always construct a history $\mathcal{H}$ for $\mathcal{S}$ as follows. If $\mathcal{S}'$ is formed from $\mathcal{S}$ through a coalescence of type a, sample members then form $\mathcal{H}$ through first coalescing two members of type a back in time. This results in a set of sequences identical to the members of $\mathcal{S}'$ who may then be evolved back to their ancestor according to the history $\mathcal{H}'$ (with starting point shifted slightly backward in time). Finally, if $\mathcal{S}'$ is formed from $\mathcal{S}$ through the removal of a noninformative mutation at m from $\mathcal{S}$, then there exists a unique sequence l in $\mathcal{S}$, which is the only representative of one of the two types present at m. l corresponds to some line l' in $\mathcal{S}'$, where l' is identical in type to l except at m. Form $\mathcal{H}$ through first mutating the ancestral line to l, at m. The ancestor to l is now of the same type as l'; so again we obtain a set identical to $\mathcal{S}'$ and may use $\mathcal{H}'$ exactly as before to give the history $\mathcal{H}$ of the original data. Then in the same way as above we can deduce $R_{\mathcal{S}} \leq R_{\mathcal{S}'}$, giving the proposition. ■