Algorithms to Distinguish the Role of Gene-Conversion from Single-Crossover Recombination in the Derivation of SNP Sequences in Populations

Y.S. Song[1]*, Z. Ding[1], D. Gusfield[1], C.H. Langley[2], and Y. Wu[1]

[1] Department of Computer Science, University of California, Davis, CA 95616, USA
[2] Section of Evolution and Ecology, University of California, Davis, CA 95616, USA

Abstract. Meiotic recombination is a fundamental biological event and one of the principal evolutionary forces responsible for shaping genetic variation within species. In addition to its fundamental role, recombination is central to several critical applied problems. The most important example is "association mapping" in populations, which is widely hoped to help find genes that influence genetic diseases [3, 4]. Hence, a great deal of recent attention has focused on problems of inferring the historical derivation of sequences in populations when both mutations and recombinations have occurred. In the algorithms literature, most of that recent work has been directed to single-crossover recombination. However, *gene-conversion* is an important, and more common, form of (two-crossover) recombination which has been much less investigated in the algorithms literature.

In this paper we explicitly incorporate gene-conversion into discrete methods to study historical recombination. We are concerned with algorithms for identifying and locating the extent of historical crossing-over and gene-conversion (along with single-nucleotide mutation), and problems of constructing full putative histories of those events. The novel technical issues concern the incorporation of gene-conversion into recently developed discrete methods [20, 26] that compute *lower* and *upper-bound* information on the amount of needed recombination without gene-conversion. We first examine the most natural extension of the lower bound methods from [20], showing that the extension can be computed efficiently, but that this extension can only yield weak lower bounds. We then develop additional ideas that lead to higher lower bounds, and show how to solve, via integer-linear programming, a more biologically realistic version of the lower bound problem. We also show how to compute effective upper bounds on the number of needed single-crossovers and gene-conversions, along with explicit networks showing a putative history of mutations, single-crossovers and gene-conversions.

We validate the significance of these methods by showing that they can be effectively used to distinguish simulation-derived sequences generated without gene-conversion from sequences that were generated with gene-conversion. We apply the methods to recently studied sequences of *Ara-*

* To whom correspondence should be addressed. E-mail: yssong@cs.ucdavis.edu

bidopsis thaliana, identifying many more regions in the sequences than were previously identified [22], where gene-conversion may have played a significant role. Demonstration software is available at www.cs.ucdavis.edu/∼gusfield.

1 Introduction

Sequence variations in populations (in a single species) are caused in part by mutations at single nucleotide sites, and in part by recombination during meiosis, which creates a chimeric genome in an individual from the genomes of the individual's two parents. Sites where two alleles (states) occur in a population with a frequency above some threshold are called Single Nucleotide Polymorphism (SNP) sites. Much recent attention has focused on problems of inferring the historical derivation of SNP sequences in populations when both mutations and recombinations have occurred. In the algorithms literature, most of that work has been directed to single-crossover recombination. (Previous methods for single-crossover recombination appear in [1, 9–14, 20, 24–26].) However, *gene-conversion* is a form of two-crossover recombination that has large biological significance, and there has been much less algorithmic work devoted to the study of models that incorporate gene-conversion as well as mutation and single-crossover recombination. Some exceptions are the papers [6, 9, 19], and statistical methods have also been developed [5, 23, 27, 30] to address gene-conversion.

Tools to study gene-conversion are important because gene-conversion is a fundamental biological process [18] that is not fully understood (partly because fine-scale data is needed which is only now becoming available, and partly because of the lack of algorithmic tools); because gene-conversion is a cause of genomic sequence variation in populations [8, 21]; and because gene-conversion has the potential to cause problems in association-mapping [15, 29]. Association mapping depends on understanding the structure of linkage disequilibrium (LD) in population data: "Standard population genetics models of recombination generally ignore gene conversion, even though crossovers and gene conversions have different effects on the structure of LD" [29].

In this paper, we extend recently developed tools for the study of historical (single-crossover) recombination and mutation, to explicitly incorporate gene-conversion events. We validate the biological significance of these methods by showing that the methods can be effectively used to distinguish sequences that were generated without gene-conversion from sequences that were generated with gene-conversions, and we apply these methods to identify regions in *Arabidopsis thaliana* sequences where gene-conversions may have played a significant role, in comparison to single-crossover recombination, in the derivation of the sequences.

In contrast to our methods, existing statistical methods (for example [23]), do not provide information on the necessary amount of recombination in the history of the sequences, or produce an explicit derivation of those sequences using mutation and recombination. Those methods also do not assess the relative importance of single-crossover recombination compared to gene-conversion. Those methods are based on patterns in the sequences rather than on how well a full history can be obtained to explain the derivation of the sequences, or on how much recombination is needed.

2 Recombination: Crossing-Over and Gene-Conversion

There are two major forms of recombination that occur during meiosis: single-crossover recombination (called "crossing-over" in the genetics literature), and gene-conversion. We will use "crossing-over" and "single-crossover recombination" interchangeably, and use "recombination" to refer to either crossing-over or gene-conversion.

Meiotic Crossing-Over: The best studied form of recombination is *crossing-over*, where during meiosis two equal length sequences produce a third sequence of the same length consisting of some prefix of one of the sequences, followed (at the "breakpoint") by a suffix of the other sequence.

Gene-Conversion: The other major form of meiotic recombination, called "gene-conversion", involves two crossovers at two breakpoints. In gene-conversion, a new sequence is formed from a prefix of one sequence, followed by an internal segment of a second sequence, followed by a suffix of the first sequence. All three sequences are of the same length. The endpoints of the internal segment are the "breakpoints" of the gene-conversion. Gene-Conversion is a small-scale meiotic event; the internal segment (called a "conversion-tract", or "tract") is short, around 50 to 2000 base pairs. Gene-conversion has been hard to study in populations because of the lack of analytical tools and the lack of fine-scale data. For example, little is known about the distribution of tract lengths. However, genomic data produced over the next several years should allow quantification of the fundamental parameters of gene conversion, and the contribution of gene conversion to the overall patterns of sequence variations in a population.

3 Minimizing the total number of recombination events

Given a set M of binary (SNP) sequences, we would like to determine the true history of mutations, crossing-over events and gene-conversions that derived the sequences from some ancestral sequence. This is of course impossible and instead previous research has focused on computing or estimating the *minimum* number, denoted $R_{\min}(M)$, of crossing-over events needed to derive the sequences from some known or unknown ancestral sequence, when only one mutation is allowed per site in the entire history of the sequences. Although the true history of the sequences may have involved more than $R_{\min}(M)$ recombinations, $R_{\min}(M)$ and particular *lower-bounds* on $R_{\min}(M)$, have proven to be useful reflections of the true historical number, for example allowing or contributing to the identification of recombination hotspots in genomic sequences [2, 7, 28].

In this paper we move from a focus on $R_{\min}(M)$, to incorporate gene-conversions. We define $T_{\min}(M)$ as the minimum total number of recombination events needed to derive M from an ancestral sequence (either known or unknown in different versions of the problem) where a recombination event is *either* a crossing-over or a gene-conversion event. Because gene-conversion tract length is typically small, we will often bound its permitted length, and define $T_{\min}(M,t)$ as the minimum number of recombination events needed to derive M, where each gene-conversion has tract length at most t (nucleotides). In the next section we discuss a practical method to compute *lower bounds* on $T_{\min}(M,t)$, and in Section 5 we discuss a practical method to compute an actual sequence of events that derives M. Since $T_{\min}(M) = T_{\min}(M,t)$ when t is sufficiently large (for example, the physical distance between the first and the last sites in M), these methods can be used to compute bounds on $T_{\min}(M)$.

4 Lower Bounds on Crossing-Over and Gene-Conversion

Since the effect of one gene-conversion can be obtained by two crossing-over events, $LBCO(M)/2$ is a valid lower-bound on $T_{\min}(M,t)$, where $LBCO(M)$ is any lower bound $R_{\min}(M)$. Several such lower bounds on $R_{\min}(M)$ have been developed and extensively studied [1, 12, 17, 20]. We will prove that when t is unconstrained, the most natural extensions of these methods to include gene-conversions, yield only weak lower bounds. However, we introduce additional ideas to increase these lower bounds, and show how to obtain higher lower bounds when t is bounded.

Our methods to compute lower bounds on $T_{\min}(M,t)$ are based on an a general approach developed by Myers and Griffiths [20] to compute lower bounds on $R_{\min}(M)$. Their approach has two essential parts: methods to compute *local* lower bounds for intervals of sites, and a simple, polynomial-time method to combine those local bounds into a composite *global* lower bound. All of the known methods (HK [17], Haplotype and History [20], and Connected Component [1, 12]) to compute local lower bounds on crossing-over extend immediately to the case that gene-conversions are allowed, but the issue of how to combine those local bounds into a composite global bound on $T_{\min}(M,t)$ is more complex.

The Haplotype local bound: Due to its centrality, we discuss the local Haplotype bound of Myers and Griffiths [20] in detail. Consider the set of sequences M arrayed in a matrix, and an interval I of sites. Let $M(I)$ be the sequences M restricted to the sites of I. Then $h(M(I))$ is defined as the number of *distinct* rows of M, minus the number of *distinct* columns of M, minus one. $h(M(I))$ is a valid lower bound on $R_{\min}(M(I))$, and in fact a lower bound on the number of breakpoints that must be located *inside* interval I. To see this, assume first that all the columns of M are distinct, since removal of (duplicate) columns cannot increase the number of crossing-over events needed, so any lower-bound computed for the reduced matrix will be a lower-bound for the original M. Next, consider the derivation of $M(I)$ using $R_{\min}(I)$ crossing-over events and one mutation per site in I, and consider an actual specification of the relative order that the events and mutations occur (this is called a "history"). Any mutation of a site in I can only occur once in the history, and it can result in only one new sequence (not yet derived in the history). Similarly, each crossing-over event with breakpoint in I can only create one new sequence. A crossing-over event with breakpoint outside of I, or a mutation outside of I cannot create a new sequence in the set $M(I)$. The history must create all the distinct sequences in $M(I)$ starting from some ancestral sequence which might itself be in $M(I)$. It follows that there must be at least $h(M(I))$ crossing-over events whose breakpoint is (strictly) in I, so $h(M(I)) \leq R_{\min}(M(I))$. Clearly, $h(M(I))$ is also a lower bound on $T_{\min}(M(I), t)$ for any t, because a single gene-conversion can also create at most one new sequence in $M(I)$.

In [20], local haplotype lower bounds for an interval I were raised by considering subsets of sites in the interval. That approach was further explored in [2] and optimized in [26]. In our software, we use the latter approach to obtain the highest possible local haplotype bounds for each interval I.

The Composite global lower bound: We are interested in a lower bound on $T_{\min}(M, t)$, not just a bound on $T_{\min}(M(I), t)$ for a single interval I. Of course, $h(M)$ (the haplotype bound applied to the interval consisting of all the sites) is a lower bound on $T_{\min}(M, t)$, but in the computations we have done, it is a very poor lower bound, often a negative number. The same is true of $R_{\min}(M)$, but a much better composite global lower bound on $R_{\min}(M)$ can be obtained from the local lower bounds. We say a point p "covers" an interval I if p is contained in I with at least one SNP site on each side of p. Then we obtain a composite global bound on $R_{\min}(M)$ by solving the **Crossover Coverage Problem**: Find the smallest set of points B so that each interval I is covered by at least $h(M(I))$ points of B. $|B|$ is a valid lower bound on $R_{\min}(M)$, and B can be found using a simple polynomial-time algorithm[20]. However, $|B|$ is not necessarily a lower bound on $T_{\min}(M, t)$. To use the local bounds to obtain a composite bound on $T_{\min}(M, t)$, we next formulate a natural generalization of the Crossover Coverage Problem.

We say a line-segment "covers" an interval I if at least one end p of the line-segment covers I. Note that a line-segment that strictly contains I does not cover it, and that a line-segment covers I only once even if both of its endpoints are in I. The intuitive meaning is that a line-segment represents a gene-conversion, and a line-segment covers an interval I only if the action of the gene-conversion it represents could create a new sequence in $M(I)$. Then we obtain a composite bound on $T_{\min}(M, t)$ by solving the **Gene-Conversion Coverage Problem**: Find the smallest set consisting of points P, and line segments S with length at most t, so that each interval I is covered by at least $h(M(I))$ elements of $P \cup S$. Each point in P represents a crossing-over and each line-segment in S represents a gene-conversion. Clearly, a derivation of M using exactly $T_{\min}(M, t)$ recombinations, where the conversion tract of any gene-conversion is of length at most t, defines a set of points and line-segments that cover each interval I at least $h(M(I))$ times. Therefore, a solution to the Gene-Conversion Coverage Problem is a valid lower bound on $T_{\min}(M, t)$.

4.1 A special case of the Gene-Conversion Coverage Problem

We first show that when t is unbounded, the Gene-Conversion Coverage Problem has a simple, yet disappointing solution. For the following discussion, let B be a minimum-sized set of breakpoints that solves the Crossover Coverage Problem, and let $|B| = n$. Number the breakpoints in B left to right choosing an arbitrary ordering of breakpoints that lie on the same point. For any $k \leq \lfloor \frac{n}{2} \rfloor$, let $P(B, k)$ be a pairing of the leftmost k breakpoints to the rightmost k breakpoints of B under the mapping $i \rightarrow n - k + i$, and create a line-segment between the two endpoints of each pair in $P(B, k)$. Let S be the set of these k line-segments, and let P be the set $n - 2k$ unpaired points in B. We will show that there is a solution of the Gene-Conversion Coverage Problem that has this form for some k, and show that the best k can be easily obtained.

Note that in $P(B, k)$, if $i < j \leq k$, then i maps to a breakpoint to the left of the breakpoint that j maps to, a property that we call "monotonicity". Define $L(I)$ as the number of breakpoints in B to the left of I, and $R(I)$ as the number of breakpoints in B to the right of I. We say a line-segment is "contained in" I if both of its ends are contained in I. Define the "coverage" of interval I as the number of elements of $P \cup S$ that cover I.

Lemma 1. *Let I be any interval where some line-segment in S is contained in I. Then exactly $k - (L(I) + R(I))$ line-segments in S are contained in I.*

Proof. First, if a line-segment (a, b) in S is contained in I, then $k \geq L(I) + 1$ so breakpoint $L(I) + 1$ (the leftmost breakpoint in I) must be the left end of some line-segment in S. Moreover, by monotonicity, the right endpoint of that segment (which is at breakpoint $n - k + L(I) + 1$) must be at or to the left of b, and hence in I. Since the pairing $P(B, k)$ involves the rightmost k breakpoints in B, all the breakpoints in I to the right of $n - k + L(I) + 1$ must be right endpoints of some line-segment in S, and again by monotonicity, their paired left endpoints must be to the right of $L(I) + 1$, and hence must be in I. The rightmost breakpoint in I is $n - R(I)$, so there are exactly $n - R(I) - [n - k + L(I) + 1] + 1 = k - (L(I) + R(I))$ line-segments in S that are contained in I. □

Lemma 2. *For $k \leq n - \max_I(h(M(I)))$, the coverage of any interval I is at least $h(M(I)))$.*

Proof. Let $B(I)$ be the number of breakpoints in B that are contained in I. The coverage of I is exactly $B(I)$ minus the number of line-segments in S contained in I. Since, $B(I) \geq h(M(I)))$ for all I, we only need to examine intervals where some line-segment in S is contained in the interval. Let I be such an interval. By assumption, $k \leq n - \max_I h(M(I))) \leq n - h(M((I)))$, so

$$k - (L(I) + R(I)) \leq n - h(M((I))) - (L(I) + R(I)) = B(I) - h(M((I))).$$

Therefore $h(M((I))) \leq B(I) - [k - (L(I) + R(I))]$, and by the Lemma 1, the coverage of I is at least $h(M(I)))$. □

Corollary 1. *If* $k = \min(\lfloor \frac{n}{2} \rfloor, n - \max_I h(M(I)))$*, then the coverage of* I *is at least* $h(M(I))$*, for each interval* I*.*

Theorem 1. *If* B *is a minimum sized set of breakpoints (of size* n*) solving the Crossover Coverage Problem, then the optimal solution to the Gene-Conversion Coverage Problem has size exactly* $\max(\lceil \frac{n}{2} \rceil, \max_I h(M(I)))$*.*

Proof. By the Corollary, if we set k to $\min(\lfloor \frac{n}{2} \rfloor, n - \max_I h(M(I)))$, then every interval I has coverage at least $h(M(I))$, and $|S \cup P|$ is exactly $n - k = \max(\lceil \frac{n}{2} \rceil, \max_I h(M(I)))$. But both of those terms are trivial lower bounds on the number of needed line-segments and single breakpoints in any solution to the Gene-Conversion Coverage Problem, and hence that choice of k gives the optimal solution. □

So when t is unbounded, we have a simple, efficient algorithm for the Gene-Conversion Coverage Problem: solve the Crossover Coverage Problem, yielding set B, and then apply Theorem 1. Note that Theorem 1 holds regardless of which (optimal) solution B is used, and provides a lower bound on $T_{\min}(M, t)$ for any t, as well as for $T_{\min}(M)$. It can also be shown that Theorem 1 holds even if we use HK or history or connected component local lower bounds, instead of Haplotype local lower bounds.

4.2 Improving the Bounds

Theorem 1 establishes trivial lower bounds on $T_{\min}(M, t)$ and $T_{\min}(M)$. It's importance is that it proves that the natural extension of the way that good lower bounds on $R_{\min}(M)$ were obtained, will not yield non-trivial lower bounds when gene-conversion is included. To get higher bounds we have to use additional constraints. The first such constraint is to bound the permitted tract length to t in any solution to the Gene-Conversion Coverage Problem. We do not have a polynomial-time algorithm for this version of the problem, but next show how to effectively solve it using integer linear programming.

An ILP formulation for bounded t: We define $\phi(i)$ as the physical position in the chromosome of site i. Given an input matrix M with m sites, and a bound t, we define an integer-valued variable $K_{i,j}$ for each pair of integers i, j where $0 < i \le m - 1$, $0 < j \le m - 1$, $i \le j$, and either $i = j$ or $\phi(j) - \phi(i + 1) < t$. The value that variable $K_{i,j}$ takes on in the ILP solution specifies the number of line-segments $[i, j]$ (whose two endpoints are between sites $i, i + 1$ and sites $j, j + 1$) that will be used in the solution. For an interval $I = [a, b]$, we define the set $A(I) = \{K_{i,j} : a \le i < b \ \textit{ or } \ a \le j < b\}$. Set $A(I)$ is the set of the variables that can specify a line-segment that covers I. We allow $i = j$ to indicate a single point. Then the following ILP solves the Gene-Conversion Coverage Problem when t is bounded:

Minimize $\sum_{(i,j)} K_{i,j}$
Subject to
$\sum_{K_{i,j} \in A(I)} K_{i,j} \ge h(M(I))$, for each interval I.

Further improvements: We can often increase the composite global lower bound on $T_{\min}(M,t)$ with the following observation. We say sites p, q are *incompatible* when all four binary combinations $00, 01, 10, 11$ appear at those two sites. If p, q are incompatible, then there must be at least one breakpoint in the interval $[p, q]$, and this can introduce additional constraints on possible gene conversions. For example, consider the following four sequences: $000, 011, 110, 101$. All three sites are pairwise incompatible. Let a, b, c denote the first, second, and third sites, respectively. Intervals $[a, b], [b, c], [a, c]$ all have a local bound of 1. We can cover those intervals using a single line-segment with one endpoint between a and b, and the other endpoint between b and c. That single segment covers all three of the intervals. However, it is easy to see that a gene conversion corresponding to that single segment cannot make sites a and c incompatible. Thus, there must be at least *two* gene-conversion or crossing-over events for this example. Those additional constraints can increase the resulting global lower bound, and can be incorporated into the ILP with the following constraints:

$\sum_{p\leq a<q, b\geq q} K_{a,b} + \sum_{a<p, p\leq b<q} K_{a,b} + \sum_{p\leq a<q} K_{a,a} \geq 1$, for each pair of incompatible sites p, q.

Note that $K_{a,a}$ defines a crossing-over event rather than gene-conversion.

The above ILP formulation can be solved reasonably fast for data of the size of current biological interest. Some timing details will be presented in Section 6. This approach has been implemented in the program HapBound-GC. A demonstration version of HapBound-GC uses the free GNU GLPK package to solve the ILP.

Another way to raise the composite lower bound involves the interaction of local bounds and global bounds that use those local bounds. Consider a subset of sites K that span an interval I, and let $M(K)$ be the sequences M restricted to the sites in K. The action of a gene-conversion can create a sequence in $M(K)$ differing from both of the parent sequences only if there are sites in K to the left and right of one of the two breakpoints of the gene-conversion. Moreover, if the two ends of a gene-conversion are in the interval spanned by K, then the gene-conversion can create a new sequence in $M(K)$ only if there is a site in K between those two breakpoints. Those observations constrain where we must place points and line-segments in a solution to the Gene-Conversion Coverage Problem. In fact, such constraints are used in the ILP for the subsets described in Section 4 that yield the highest local haplotype bounds. However, we can further raise the composite global bound by enumerating each subset of sites K up to a certain size, and computing the haplotype lower bound on the sequences $M(K)$. Then, we generate constraints for the ILP requiring that the number of points and line-segments covering K must be at least the computed haplotype bound for $M(K)$, and requiring that the selection of covering points and line-segments be constrained as described above. These additional ideas result in larger lower-bounds at the cost of increasing the size of the ILP and the time needed to solve it. Our experience shows that when CPLEX is used, the ILP formulation can still be solved reasonably fast when enumerating all size-3 (or even 4) subsets of sites. Unfortunately, the free GNU GLPK package is much slower than CPLEX when we enumerate all size-3 subsets with data containing more than 30 sites.

5 Computing upper bounds on $T_{\min}(M, t)$: A practical algorithm to derive M explicitly

In this section we describe our algorithm that produces an explicit sequence of recombination and mutation events to generate an input set of sequences M. This of course gives an upper bound on $T_{\min}(M, t)$.

The overall framework of our upper bound on $T_{\min}(M, t)$ is similar to that of the upper bound described in [26] where only mutations and crossing-over events are allowed. The main distinction lies in that we here use the following more general *derivation cost*: Given a row r in a binary matrix A and a specified maximum tract length t, $c(r|A - r, t)$ is defined as the minimum total number of crossing-over events and gene-conversions with tract length at most t that are needed to derive row r from some other rows in A. In [26], only single-crossovers were allowed in defining the derivation cost, denoted $w(r|A - r)$.

The following procedure, called **Procedure History**, is the key component of our method to compute an upper bound on $T_{\min}(M, t)$.

```
Step 0. Set $A = M$.
Step 1. Set $W$=0.
Step 2. Repeat Steps 2a and 2b until neither operation is possible.
  Step 2a. Collapse two identical rows of $A$ into one row.
  Step 2b. Remove any column $k$ of $A$ containing less than two 0s or less
           than two 1s.
Step 3. If $A$ is empty, then stop. Otherwise, remove a row from $A$, say row
        $r$, set $W \leftarrow W + c(r|A - r, t)$, and go to step 2.
```

The final upper bound $\tau(M, t)$ on $T_{\min}(M, t)$ is defined as the *minimum* final value of W over all possible executions of Procedure History. (Two inequivalent executions have different sequences of row removals in Step 3.)

Of course, if we explicitly explore all possible executions, then the method would only be practical for very small problem instances. Instead, we use branch and bound ideas to find $\tau(M, t)$ without explicitly exploring all possible executions of Procedure History. The details of the branch and bound method are similar to those in [26], and their use results in dramatic speedups allowing practical computation of $\tau(M, t)$ for moderate size data. Some timing details are presented in Section 6. We implemented this method into a program called SHRUB-GC which is available on the web.

The fact that $\tau(M, t)$ is an upper bound on $T_{\min}(M, t)$ follows from the observation that every execution of Procedure History, with final cost denoted W^*, specifies backwards in time a series of W^* recombination events (along with one mutation per site) that derive M. These events can be represented in an directed acyclic graph (DAG), and our program can produce this DAG. More specifically, an execution of Step 2a creates a new node with two directed edges out of it (this is a "coalescent" event); an execution of Step 2b creates a directed edge on which site k mutates; and removing the row r in Step 3 corresponds to creating $c(r|A-r,t)$ crossing-over and gene-conversion events, along with the needed coalescent nodes, to derive row r from the sequences in $A-r$.

The generalized derivation cost $c(r|A-r,t)$ can be computed by an $O(nmt^2)$ time algorithm recently presented in [19], where n and m are the number of sequences and the length of each sequence in M. There may exist several distinct combinations of $c(r|A-r,t)$ crossing-over and gene-conversion events that produce r from $A-r$. We modified the algorithm in [19] to generate r using the minimum number of gene-conversions possible over all histories that use exactly $c(r|A-r,t)$ crossing-over and gene-conversion events to generate r from $A-r$. That derivation of r can be used in the construction of the larger DAG that derives M.

6 Application: Distinguishing Gene-Conversion from Crossing-Over

One of the key motivations for the development of our lower and upper bound methods is to use them to estimate the relative extent that gene-conversion, compared to crossing-over, was involved in the true historical generation of a set of sequences. Further, the successful use of our methods to distinguish the average behavior of sets of sequences whose true generation involved moderate to high levels of gene-conversion, from that of sets that used a low level, is also a validation of the biological relevancy of the objective function $T_{\min}(M,t)$ and bounds on it. In this section we highlight some key empirical results.

Define $\Delta\tau(M,t) := \tau(M,0) - \tau(M,t)$, where both $\tau(M,0)$ and $\tau(M,t)$ denote either our lower or upper bounds on $T_{\min}(M,t)$ (context will determine which one). Note that $\Delta\tau(M,t) \geq 0$ for all M and t, since $\tau(M,0)$ corresponds to the case when no gene-conversion is allowed.

There may exist several distinct combinations of single-crossovers and gene-conversions that produce $\tau(M,t)$. We use $\gamma(M,t)$ to denote the minimum number of gene-conversions over all such combinations of single-crossovers and gene-conversions. As mentioned above, we modified the algorithm in [19] so that $\gamma(M,t)$ can be computed for the upper bound. Computing it in the lower bound can be done by an easy modification of the ILP presented in Section 4.

The general idea of the simulations reported in this section is to generate sets of sequence data with varying amounts of gene-conversion. We expect that $\gamma(M,t)$ and $\Delta\tau(M,t)$ will increase as t increases, but will do so faster and will be larger for data actually generated with gene-conversions than for data without gene-conversions. That is, as we allow our methods to try to use more gene-conversions, they will be able to do so more effectively on sequences that were actually generated using gene-conversions. Another reflection of the same intuition is that the proportion of total recombinations that are gene-conversions should grow at t grows, but in a more pronounced way for data that was generated using gene-conversions.

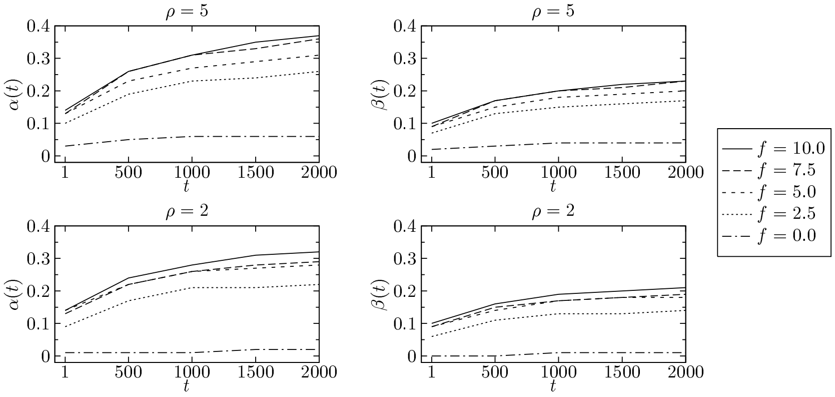

Fig. 1. Upper bound expectations $\alpha(t) := \mathbb{E}[\gamma(M,t)/\tau(M,t) \,|\, \tau(M,0) \neq 0]$ and $\beta(t) := \mathbb{E}[\Delta\tau(M,t)/\tau(M,0) \,|\, \tau(M,0) \neq 0]$, for $n = 20, s = 30, k = 5000$ and $\lambda = 500$. We carried out computations for $t = 1, 500, 1000, 1500, 2000$ and joined the corresponding consecutive points by a straight line to generate these plots.

6.1 Simulation Study

Our main empirical result is that the expected behavior of both $\gamma(M,t)$ and $\tau(M,t)$ as t increases, does depend critically on the extent that gene-conversions were used to generate M. We examined $\alpha(t) := \mathbb{E}[\gamma(M,t)/\tau(M,t) \,|\, \tau(M,0) \neq 0]$ and $\beta(t) := \mathbb{E}[\Delta\tau(M,t)/\tau(M,0) \,|\, \tau(M,0) \neq 0]$ in our study. The first expectation reflects the fraction of all events that are gene-conversions, in solutions that minimize the total number of events. The second expectation summarizes the reduction in the number of total recombination events for a given t, compared to the number when no gene-conversion is allowed. If our methods truly reflect the extent of gene-conversion used to generate the sequences, and properly reflect the tract lengths used, then both of these summary statistics should increase with increasing f (defined below) and t.

We used Hudson's program MS [16] to generate simulated data. MS uses a finite-sites uniform recombination model, and the user specifies the number k of sites to be considered in the model. We used $k = 5000$ in our study. In humans, the genome-wide average of the scaled recombination rate ρ is about 0.4 per kb, which translates to $\rho \approx 2$ for a region of 5000 bps long. Instead of specifying the mutation rate, we specified the number s of polymorphic sites to be generated. With g being the probability per generation per sequence that gene conversion initiates between a pair of adjacent sites and r the probability per generation per sequence that crossing-over occurs between a pair of adjacent sites, f is defined as g/r, for $r \neq 0$. That is, f specifies the relative rate of gene-conversions used by MS, compared to crossing-over, in the generation of sequences M. Larger f specifies a higher rate of gene-conversions. It is believed that in humans, f is in the range 2 to 10 [15]. Program MS assumes that the conversion tract length is geometrically distributed with mean conversion tract length λ, provided by the user. We analyzed 500 simulated datasets for each set of parameters.

Figure 1 illustrates our upper bound results on datasets generated with $n = 20, s = 30, k = 5000, \lambda = 500$ and $\rho = 2$ or 5. An important thing to note is that, for $f = 0$ (i.e., when no gene-conversions were used in the generation of M), both $\alpha(t)$ and $\beta(t)$ remain close to zero as t changes. That is, when the data were generated without gene-conversions, the allowance of gene-conversions in the bounding methods does not reduce the bounds very much. In effect, the methods cannot "make up" gene-conversions that did not actually occur in the generation of the sequences. Similarly, for fixed f, both $\alpha(t)$ and $\beta(t)$ first increase rapidly as t increases from zero, but increasing t beyond 1500 does not seem to influence them very much. This behavior is consistent with the fact that, for $\lambda = 500$, the probability of the tract length being less than or equal to 1500 is about 95%. These characteristics are a very strong validation of the biological relevancy of our methods.

As f increases, both $\alpha(t)$ and $\beta(t)$ grow, and this growth is more pronounced for larger t. In general $\alpha(t)$ is more sensitive to changes in f and t than is $\beta(t)$. For fixed f and t, both $\alpha(t)$ and $\beta(t)$ tend to increase as the recombination rate ρ increases, with $\alpha(t)$ more so than $\beta(t)$. The general behavior of our lower bound is quite similar to that of our upper bound.

In general, as t, f or ρ increases, the running times of our programs increase. The average running time per dataset of SHRUB-GC ranged from a fraction of a second to a bit over a minute on a 2 GHz pentium PC. HapBound-GC is faster than SHRUB-GC, with the average running time per dataset being less than a second.

6.2 Gene-Conversion Presence (GCP) Test

Based on the simulation results of our methods, one can devise various tests for determining whether gene-conversion was used to generate a given dataset. We here suggest a simple test involving $\gamma(M, t)$: For a given maximum tract length t, we say that $\gamma(M, t) > 0$ indicates the presence of gene-conversion. Percentages of simulated datasets with $\gamma(M, t) > 0$ are summarized in Table 1, for mean tract length $\lambda = 500$. Percentages for $f = 0$ can be regarded as false positive rates, whereas percentages for $f > 0$ can be regarded as sensitivity. Results for three different methods are shown in the table: (U) requiring $\gamma(M, t) > 0$ in the upper bound method only, (L) requiring $\gamma(M, t) > 0$ in the lower bound method only, and (U&L) requiring $\gamma(M, t) > 0$ in both upper and lower bound methods.

The outcome of these tests depends on the value of t used, but our results indicate that increasing t beyond a certain point does not change the percentages by a considerable amount; that is, in our simulations using $\lambda = 500$, the difference between $t = 1$ and $t = 1000$ is much more significant than that between $t = 1000$ and $t = 2000$. In practice, the user is advised to decide on an appropriate t based on what is believed to be the mean tract length for the species being studied.

Results in Table 1 suggest that for small ρ (say, $\rho \leq 2$), false positive rates are low and method U seems to work better than method L or method U&L. For $\rho = 5$, however, both methods U and L lead to somewhat high false positive rates. Since using the combined method U&L reduces the false positive rate, a

Table 1. Percentage of datasets with $\gamma(M,t) > 0$ for $n = 20, s = 30, k = 5000$ and $\lambda = 500$. "U" denotes having $\gamma(M,t) > 0$ in the upper bound method, "L" having $\gamma(M,t) > 0$ in the lower bound method, and "U&L" having $\gamma(M,t) > 0$ in both upper and lower bound methods.

		$t = 1$			$t = 500$			$t = 1000$			$t = 1500$			$t = 2000$		
ρ	f	U	L	U&L	U	L	U&L	U	L	U&L	U	L	U&L	U	L	U&L
2	0.0	0.8	0.6	0.6	0.8	1.0	0.8	1.2	1.6	1.2	1.6	3.0	1.6	1.8	3.0	1.8
	2.5	20.6	15.0	12.4	32.6	30.2	25.2	36.6	35.6	29.8	37.4	38.4	31.0	38.6	40.4	32.6
	5.0	38.6	31.4	24.8	56.8	55.0	46.8	61.2	61.2	53.4	63.8	63.0	56.0	64.6	65.2	58.0
	7.5	46.8	37.0	29.8	62.6	61.0	52.0	65.8	67.0	57.2	68.2	70.4	60.6	69.6	72.0	62.2
	10.0	57.0	45.4	36.0	74.4	72.6	64.4	78.8	78.2	71.0	79.8	80.4	73.0	81.8	81.2	74.8
5	0.0	4.6	4.6	2.8	8.4	9.4	6.0	10.2	12.6	7.4	11.8	16.2	8.8	12.6	18.4	9.6
	2.5	40.6	37.2	25.0	63.6	64.6	54.4	68.0	71.8	61.2	69.6	72.6	62.8	71.8	73.8	64.6
	5.0	64.4	50.6	42.2	81.4	82.6	74.2	87.4	88.6	81.6	89.8	90.4	85.2	91.0	90.6	86.2
	7.5	72.6	63.2	51.4	92.8	92.4	88.0	96.0	94.4	92.2	96.8	95.4	93.8	97.0	95.4	94.0
	10.0	81.6	69.4	61.8	95.8	93.6	91.4	97.2	96.8	94.8	97.4	97.0	95.2	98.0	97.0	95.8

conservative strategy would be to use that method if $\rho > 2$ or if ρ is unknown. All three methods perform significantly better with increasing f (i.e., high sensitivity can be achieved for $f \geq 5$). Further, we remark that, although not shown here, our GCP test performs better with increasing number of segregating sites, given that all other parameters remain fixed.

6.3 GCP Test on *Arabidopsis thaliana* Data

We applied the above GCP Test to the *Arabidopsis thaliana* data of Plagnol et al.[22] To be conservative, we used method U&L. The data consist of 96 samples broken up into 1338 short fragments, each of length between 500 and 600 bps.

Most fragments contain a significant fraction of missing data, which were handled in [22] as follows. Given a fragment, they first found the set of all pairs of columns containing $00, 01, 10$ and 11. Then, if there were less than or equal to ten missing data restricted to that set, they tried all possible assignments of values to those missing data and declared that the fragment contains a clear gene-conversion event if their test produced an affirmative answer for any assignment. In our approach, instead of assigning values to missing data, we removed certain columns and rows so that the remaining dataset was free of missing data.

Typically $\rho < 1$ for each fragment, as estimated in [22], so the above simulation results imply that our GCP test should have a low false positive rate, provided that the actual evolution of *Arabidopsis thaliana* has been consistent with the model used for simulation. Since we ignored homoplasy events (recurrent or back mutations) in our simulations, we decided to account for their possibility as follows. First, we ran our GCP test using $t = 1$, thus allowing for at most one SNP in the conversion tract. Note that a gene-conversion event in such a case has similar effects as does a homoplasy event. Second, when we performed GCP tests for $t > 1$, we ignored those datasets that had affirmative GCP test results for $t = 1$.

Plagnol et al. [22] identified four fragments as containing clear signals for gene-conversion, with potential tracts being $55, 190, 200$ and 400 bps long. In contrast, 22 fragments passed our test when the maximum tract length was set to 200. (Increasing t beyond 200 did not change our results.) Of these 22 fragments, three coincided with those found in [22]. We believe that the fact that we handled missing data differently is responsible for our not detecting any signal for gene-conversion in the remaining one fragment (whose potential tract length is 400 bps) identified in [22]. All in all, our detection methods are more general than the method used in [22], and we believe that that led us to identify many more fragments than they did. Effectively, the method of Plagnol et al. [22] can only detect fragments with $\tau(M,t) = 1$ and $\gamma(M,t) = 1$. All 19 additional fragments we identified had $\tau(M,t) > 1$ and $\gamma(M,t) \geq 1$.

Acknowledgment

We thank Vincent Plagnol for providing us with the *Arabidopsis thaliana* data and for communicating with us regarding the data. This research is supported by the National Science Foundation grants EIA-0220154 and IIS-0513910.

References

1. V. Bafna and V. Bansal. The number of recombination events in a sample history: conflict graph and lower bounds. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1:78–90, 2004.
2. V. Bafna and V. Bansal. Improved recombination lower bounds for haplotype data. In *Proceedings of RECOMB 2005*, pages 569–584, 2005.
3. C. Carlson, M. Eberle, L. Kruglyak, and D. Nickerson. Mapping complex disease loci in whole-genome association studies. *Nature*, 429:446–452, 2004.
4. A. G. Clark. Finding genes underlying risk of complex disease by linkage disequilibrium mapping. *Curr. Opin. Genet. Dev.*, 13:296–302, 2003.
5. G. Drouin, F. Prat, M. Ell, and G.D. Clarke. Detecting and characterizing gene conversion between multigene family members. *Mol. Bio. Evol.*, 16:1369–1390, 1999.
6. N. El-Mabrouk. Deriving haplotypes through recombination and gene conversion pathways. *J. Bioinformatics and Computational Biology*, 2(2):241–256, 2004.
7. P. Fearnhead, R.M. Harding, J.A. Schneider, S. Myers, and P. Donnelly. Application of coalescent methods to reveal fine scale rate variation and recombination hotspots. *Genetics*, 167:2067–2081, 2004.
8. L. Frisse, R.R. Hudson, A. Bartoszewicz, J.D. Wall, J. Donfack, and A. Di Rienzo. Gene conversion and different population histories may explain the contrast between polymorphism and linkage disequilibrium levels. *Am. J. Hum. Genet.*, 69:831–843, 2001.
9. D. Gusfield. Optimal, efficient reconstruction of Root-Unknown phylogenetic networks with constrained and structured recombination. *JCSS*, 70:381–398, 2005.

10. D. Gusfield, S. Eddhu, and C. Langley. The fine structure of galls in phylogenetic networks. *INFORMS J. on Computing, special issue on Computational Biology*, 16:459–469, 2004.
11. D. Gusfield, S. Eddhu, and C. Langley. Optimal, efficient reconstruction of phylogenetic networks with constrained recombination. *J. Bioinformatics and Computational Biology*, 2(1):173–213, 2004.
12. D. Gusfield, D. Hickerson, and S. Eddhu. An efficiently-computed lower bound on the number of recombinations in phylogenetic networks: Theory and empirical study. To appear in *Discrete Applied Math, special issue on Computational Biology.*
13. J. Hein. Reconstructing evolution of sequences subject to recombination using parsimony. *Math. Biosci*, 98:185–200, 1990.
14. J. Hein. A heuristic method to reconstruct the history of sequences subject to recombination. *J. Mol. Evol.*, 36:396–405, 1993.
15. J. Hein, M. Schierup, and C. Wiuf. *Gene Genealogies, Variation and Evolution: A primer in coalescent theory.* Oxford University Press, UK, 2004.
16. R. Hudson. Generating samples under the Wright-Fisher neutral model of genetic variation. *Bioinformatics*, 18(2):337–338, 2002.
17. R. Hudson and N. Kaplan. Statistical properties of the number of recombination events in the history of a sample of DNA sequences. *Genetics*, 111:147–164, 1985.
18. A. J. Jeffreys and C. A. May. Intense and highly localized gene conversion activity in human meiotic crossover hot spots. *Nature Genetics*, 36:151–156, 2004.
19. M. Lajoie and N. El-Mabrouk. Recovering haplotype structure through recombination and gene conversion. *Bioinformatics*, 21(Suppl 2):ii173–ii179, 2005.
20. S. R. Myers and R. C. Griffiths. Bounds on the minimum number of recombination events in a sample history. *Genetics*, 163:375–394, 2003.
21. B. Padhukasahasram, P. Marjoram, and M. Nordborg. Estimating the rate of gene conversion on human chromosome 21. *Am. J. Hum. Genet.*, 75:386–97, 2004.
22. V. Plagnol, B. Padhukasahasram, J. D. Wall, P. Marjoram, and M. Nordborg. Relative influences of crossing-over and gene conversion on the pattern of linkage disequilibrium in Arabidopsis thaliana. *Genetics*, in press. Ahead of Print: 10.1534/genetics.104.040311.
23. S. Sawyer. Statistical tests for detecting gene conversion. *Mol. Biol. Evol.*, 6:526–538, 1989.
24. Y.S. Song and J. Hein. Parsimonious reconstruction of sequence evolution and haplotype blocks: Finding the minimum number of recombination events. In *Proc. of 2003 Workshop on Algorithms in Bioinformatics*, pages 287–302, 2003.
25. Y.S. Song and J. Hein. On the minimum number of recombination events in the evolutionary history of DNA sequences. *J. Math. Biol.*, 48:160–186, 2004.
26. Y.S. Song, Y. Wu, and D. Gusfield. Efficient computation of close lower and upper bounds on the minimum number of needed recombinations in the evolution of biological sequences. *Proc. of ISMB 2005, Bioinformatics*, 21:i413–i422, 2005.
27. J. C. Stephens. Statistical methods of DNA sequence analysis: Detection of intragenic recombination or gene conversion. *Mol. Bio. Evol.*, 2:539–556, 1985.
28. The International HapMap Consortium. A haplotype map of the human genome. *Nature*, 437:1299–1320, 2005.
29. J. D. Wall. Close look at gene conversion hot spots. *Nat. Genet.*, 36:114–115, 2004.
30. T. Wiehe, J. Mountain, P. Parham, and M. Slatkin. Distinguishing recombination and intragenic gene conversion by linkage disequilibrium patterns. *Genet. Res., Camb.*, 75:61–73, 2000.