# An Analytical Upper Bound on the Minimum Number of Recombinations in the History of SNP Sequences in Populations

Yufeng Wu

Department of Computer Science and Engineering

University of Connecticut

Storrs, CT 06269, U.S.A.

ywu@engr.uconn.edu

**Keywords: algorithm, combinatorial problems, population genetics, recombination, ancestral recombination graph.**

## 1 Introduction

A major development in genetics is the on-going collection of large scale population-scale genetic variation data, such as single nucleotide polymorphisms (SNPs), microsatellites, and structural variations, in several large projects (e.g. international HapMap project [8, 9]). An important question is how we can better utilize and analyze these variation data to provide more biological insights. This paper focuses on one such application: understand meiotic *recombination* using population variation data: More specifically, we are given a *binary* matrix $M$ with $n$ sequences (i.e. rows) and $m$ sites (i.e. columns). A column corresponds to a SNP site where two of the four possible nucleotides appear in the population with a frequency above some set threshold. Thus, we can use 0 and 1 to encode the two shown nucleotides. We are interested in *estimating* the level of historical recombination in the evolutionary history of the set of binary sequences.

A mutation at a site is a change of state at a single site (either from 0 to 1 or vice versa). A common assumption is the *infinite sites model* in population genetics, i.e., that any site (in the study) can mutate at most once in the entire history of the sequences. This implies that each site in any of the studied sequences can take on only two states (also called *alleles*), and hence the sequences we see today are binary sequences. Although non-binary population sequences do exist, the infinite sites model may be justified if the population being studied is not very old (and thus the chance for more than one mutation occurring at the same site is small). In fact, most SNPs in the international HapMap project [8, 9] are binary. Throughout this paper, we assume the infinite sites model holds for every SNP site we considered.

If recombination is assumed to be absent, the genealogical history of sequences is modeled as a single *tree* or perfect phylogeny (see e.g. [3]), and can be quite easily reconstructed under the one mutation per site model. The situation changes when recombinations occur. Meiotic recombination is an important genetic process, shaping the genetic diversity within a population. Meiotic recombination takes two equal length sequences and produces a third sequence of the same length consisting of some *prefix* of one of the sequences, followed by a *suffix* of the other sequence. With recombination, the genealogical history should be modeled as a *network* (called *ancestral recombination graph* or ARG as introduced in Section 1.1), instead of a tree. An important example where ARGs can be useful is "association mapping" which is widely hoped to help find genes that influence genetic diseases. Recombination in the history of a population is the key to that approach, and understanding or estimating the effects of recombination is very important to making association mapping work.

In studying recombination, a common underlying problem is to determine the *minimum* number of recombinations needed to generate a given set of molecular sequences from an ancestral sequence (which may or may not be known), assuming the one mutation per site model. We let $R_{\min}(M)$ denote the minimum number of recombinations needed to generate the sequences $M$ from any ancestral sequence, allowing only one mutation per site over the entire history of the sequences. The problem of computing or estimating $R_{\min}(M)$ has been studied in a number of papers (e.g. [10, 11, 6, 12, 1, 13]). A variation to the problem occurs when a specific ancestral sequence is known in advance. No polynomial-time algorithm for either problem is known, and the second problem is known to be NP-hard [14, 2]. Since $R_{\min}(M)$ is not easy to compute exactly, an alternative is to ask for an upper bound on $R_{\min}(M)$. Song, et al. [13] developed an algorithm to compute upper bounds on $R_{\min}(M)$ (i.e. derive sequences in $M$ with close to $R_{\min}(M)$ recombinations). The method is heuristic: to get the best upper bound, the running time is exponential in terms of $n$.

This article is focused on a *different* type of question on estimating recombination: we want to derive an *analytical* (i.e. mathematical) estimate on the needed recombinations. Note that this is different from the computational upper bounds (e.g. the method in [13]) which take a particular dataset. Here, we ask for the *range* of the number of recombinations needed for *arbitrary* data with fixed size. Analytical bounds on recombination are very easy to compute (by hand) and also theoretically interesting. This motivates the following problem, which is the focus of the current work.

**Analytical upper bound** problem: for binary matrix with $n$ rows and $m$ sites, find an analytical upper bound on $R_{\min}(M)$ in terms of input data characteristics (e.g. $n$ and $m$).

Analytical upper bound gives a mathematical bound on $R_{\min}(M)$, based on properties of $M$ (e.g. $m$ and $n$). Such an analytical upper bound holds for any $M$ with certain properties, and thus gives very quick estimate on how many recombinations are needed for data with known sizes. Before the current work, only an almost trivial analytical upper bound is known [7, 6]. Here, we present the **first** non-trivial analytical upper bound on $R_{\min}(M)$, which is $O(\log_2(n))$ smaller than the previously known upper bound, where $n$ is the number of the input sequences. Moreover, we show that the new upper bound is asymptotically optimal.

## 1.1 More definitions

When we say that $M$ is "generated" by mutations and recombinations with one mutation per site, that is shorthand for a more complete model of how sequences are generated on a network or an ancestral recombination graph (ARG) in the population genetics literature. An ARG $N$, generating $n$ sequences of $m$ sites, is a directed acyclic graph containing exactly one node (the root) with no incoming edges, and exactly $n$ leaves with one incoming edge each and no outgoing edges. Every other node has one or two incoming edges. A node with two incoming edges is called a "recombination" node. Each site (integer) from 1 to $m$ is assigned to exactly one edge in $N$, and none are assigned to any edge entering a recombination node. Each node in $N$ is labeled by an $m$-length binary sequence, starting with the root node which is labeled with an "ancestral" sequence. The label for any non-recombination node $v$ is obtained from the label of $v$'s parent $p(v)$ by changing the state of any site assigned to the edge $(p(v), v)$. This implements mutations on that edge. Each recombination node $a$ is associated with an integer $r_a$, called the "crossover point" for $a$. See Figure 1 for an illustration on the ARG. See Gusfield, et al. [6] and Gusfield [4] for more

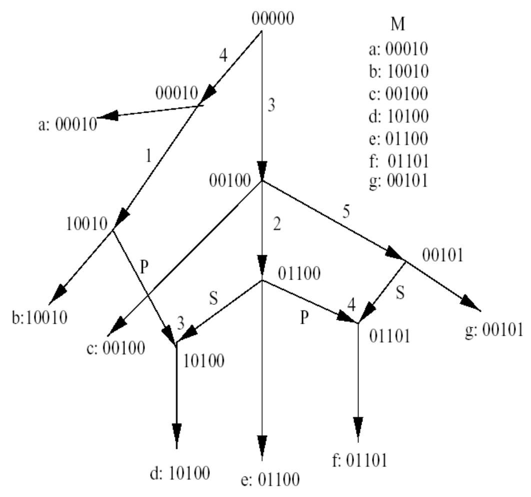

Figure 1: A phylogenetic network (ARG) with two recombination nodes (taken from [6]). The matrix that this ARG derives is shown to the right.

elaborated explanation.

When we say $r$ is a *lower* bound on $R_{\min}(M)$, this means that $R_{\min}(M) \geq r$. The "haplotype bound", $h(M)$, was introduced in [11]. The haplotype bound is one of best lower bounds on $R_{\min}(M)$. Consider the set of sequences $M$ arrayed in a matrix. Then $h(M)$ is the number of distinct rows of $M$, minus the number of distinct columns (denoted as $m$) of $M$, minus one. It is easy to establish its correctness (see [11]). When used with the composite method [11], it leads to high lower bounds.

A commonly-used technique of generating a sequence in the history is to use recombination to concatenate segments that already appear in the history before. In more detail, consider a sequence $r$. Suppose $r$ can be broken into non-overlapping blocks $b_1, b_2, \dots b_k$. Here, a block is a contiguous segment of sites of $M$. Further assume for each $b_i$ there exists a sequence $r_i$ already in the history such that $r_i$ matches $r$ when restricted to $b_i$. That is, the segment $b_i$ is already present in the history. Then, we can easily concatenate the blocks $b_1, \dots, b_k$ through recombination between blocks. To see this, we maintain a recombinant sequence $r_p$ with an increasingly longer prefix of $r$, which is initialized to be identical to $r_1$ (i.e. the existing sequence matching $b_1$). Then, for each block $b_i$ ($i \geq 2$), we perform recombination between $r_p$ and $r_i$ to bring block $b_i$ into $r_p$. Of course, no recombination is needed if $r_p$ already contains $b_i$. The number of recombination needed is no more than $k - 1 \leq m - 1$.

## 2 Improved analytical upper bound on $R_{min}$

In this section, we present the first non-trivial analytical upper bound on the minimum number of recombinations $R_{min}$. This bound improves a trivial bound by a factor of $\log_2(n)$ asymptotically. We start by describing the previous trivial bound $UB_0 = (n-1) \times (m-1)$. A simple proof of the validity of $UB_0$ as an upper bound was given in [7]. For completeness, we give a short proof here.

**Proposition 2.1** *[7, 6]* $R_{min} \leq (n-1) \times (m-1)$.

**Proof** We start from an arbitrary sequence $r_1$ in $M$, and generate a *complementary* sequence $r'_1$ through mutations at each site. That is, $r_1[j] + r'_1[j] = 1$ for each site $s_j$. Now, for any of the remaining $n-1$ sequences $r_i$ in $M$, it takes at most $m-1$ recombinations between $r_1$ and $r'_1$ to derive $r_i$. Therefore, we derive all sequences with at most one mutation per site. Thus, we need no more than $(n-1) \times (m-1)$ recombinations in deriving $M$. ■

We now demonstrate a *non-trivial* upper bound:

$$UB_1 = 2mn/log_2(n)$$

This gives $\mathrm{O}(\log_2(n))$ improvement upon $UB_0$. The *key* idea of $UB_1$ is to exploit local sequence *similarity* shared by the input sequences within *short* intervals. Here, interval refers to a segment of *consecutive* SNPs. For any interval with $k$ sites, there are at most $2^k$ unique sequences with this interval. Thus, when $k = \mathrm{O}(\log_2(n))$, there are at most $n$ possible unique sequences. Note that there are no more than $n$ unique *input* sequences within any interval. But it may take more than one recombination per sequence when generating these $n$ input sequences. The important property here is, for short interval, we can afford to derive *each* of the $n$ possible sequences (thus also the input sequences segments within the interval) with no more than one recombination per sequence. Thus, it takes no more than $n$ recombinations to generate the input sequences within such interval.

To exploit the local sequence similarity, we apply a divide-and-conquer approach: divide the input sequences into small segments of columns, such that each segment can contain only a small number of unique sequences. The method is a two-stage approach: first derive sequences for each segment with mutations and recombinations, and then combine the segments to generate entire data with recombinations. When properly divided, the number of recombinations needed will not be very large.

We need the following lemma for our upper bound.

**Lemma 2.2** *For a matrix $M'$ with $m$ sites and un-specified number of rows, $R_{\min}(M') \leq 2^m - m - 1$.*

**Proof** First $R_{\min}(M')$ remains the same when duplicate rows are removed. That is, we only consider unique rows in $M'$. Then, the key fact for $M'$ is that the number of unique rows is at most $2^m$. This is because all rows are binary with $m$ bits. In the following, we show a simple procedure that generates *all* the binary sequences with $m$ bits (and so all sequences in $M'$ are included).

We start from the all-zero sequence as the root sequence. Note that we can start from any sequence with $m$ bits and the following procedure works by re-coding the input. Then generate $m$ sequences with exactly one 1 allele, through mutations at proper site from the root. From now on, only recombinations are needed for the remaining $2^m - m - 1$ sequences. We divide the remaining sequences into groups, where sequences in a group have the same number of 1 allele. We generate the sequences group by group, with increasing number of 1-allele.

We generate sequence $r$ in a group with $k$ 1-allele by a recombination of two sequences: the first being the sequence $r_1$ with a *single* 1 allele at the leftmost position $j_0$ where $r[j_0] = 1$, and the second being the sequence $r_2$ identical to $r$ but $r_2[j_0] = 0$. Note $r_2$ must have already been generated since it has *fewer* number of 1 alleles. Since we generate each of the remaining $2^m - m - 1$ sequences with exactly one recombination, the lemma follows. ■

Now, we show how to generate a matrix $M$ of $n$ rows and $m$ sites by dividing $M$ into segments of columns.

1. Divide $M$ into $m/\log_2(n)$ blocks of $n$ rows by $\log_2(n)$ sites each, as illustrated in Figure 2.

2. Starting from an arbitrarily fixed row $r_0$ (the root) in $M$, derive all the above blocks. That is, for block $b_i$, generate $m$-bit sequences in block $b_i$ such that all the $\log_2(n)$-bit sequences within $b_i$ are created. And these sequences are identical to $r_0$ outside block $b_i$. In other words, the derivation of these blocks shares the same root.

3. For each sequence in $M$ (except $r_0$), use recombinations to concatenate the segments generated in the previous step.

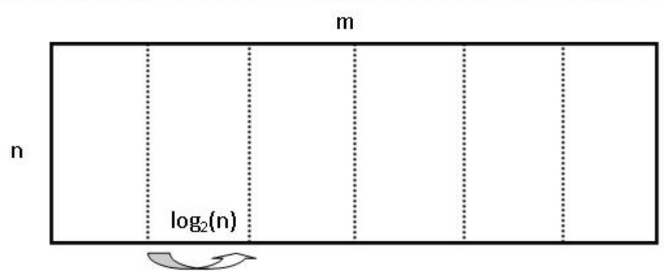

Figure 2: Divide input sequences into blocks, each with $\log_2(n)$ sites.

Note that the number of sites in each of the $m/\log_2(n)$ blocks is $\log_2(n)$. Due to Lemma 2.2, it needs at most $n - log_2(n) - 1$ recombinations to generate sequence segments in each block. Thus, we need no more than $(n - log_2(n) - 1) \times m/log_2(n) \le nm/log_2(n)$ recombinations at Step 2. Note that any sequence in $M$ is composed of segments in blocks already generated at Step 2. Since there are $m/\log_2(n)$ blocks, each sequence needs no more than $m/\log_2(n) - 1$ recombinations to link these blocks. See Section 1.1 on how to concatenate the segments into the target sequence. Since there are $n$ input sequences, the number of recombinations used at Step 3 is no more than $nm/\log_2(n)$. Therefore, the total number of recombinations needed by the above method is no more than $2mn/\log_2(n)$. This leads to a new upper bound $UB_1$, which is factor of $\log_2(n)$ improvement over $UB_0$ asymptotically. Thus, we have the following main result.

**Theorem 2.3** $R_{min} \le 2mn/log_2(n) = UB_1$.

A natural question is whether there exists better upper bounds than $UB_1$. As shown in the following, the room for further improvement is not very large: there exists matrix $n$ by $m$ matrix $M_1$ (for *any* $n$ and $m$) where $R_{min}(M_1)$ is at least (roughly) $nm/\log_2(n)$. In other words, $UB_1$ is no more than $2R_{\min}(M_1)$ for $M_1$ (with some approximation).

We again let the $n$ by $m$ matrix $M_1$ divided into n by $\log_2(n)$ non-overlapping blocks. Each of the $m/\log_2(n)$ blocks has exactly the same content (and with same row order): **all** $n$ possible unique sequences of length $\log_2(n)$. Then we restrict our attention to each block. Note that each block of $\log_2(n)$ bits has $n$ unique sequences as constructed. From the haplotype bound, we need at least $n - log_2(n) - 1$ recombinations to generate sequences within the block. Moreover, the breakpoints of the recombinations to generate sequences within a block are *inside* the block. The total number of recombinations for the entire data must be at least as large as the sum of the haplotype lower bound on $R_{min}$ of these blocks. This is because the blocks are spatially disjoint, and thus the recombinations used to generate sequence segments in one block can *not* generate novel sequence segments in another block [1]. That is, we need at least $m(n - log_2(n) - 1)/log_2(n)$ recombination. This is roughly $nm/\log_2(n)$, which is about the half of $UB_1$.

**Remarks.** We have demonstrated that $UB_1$ is asymptotically smaller than the trivial upper bound $UB_0$. Also, it is unlikely to have a upper bound that is asymptotically much better than $UB_1$. However, this does not necessarily mean the end of story: there might exist analytical upper bound on $R_{\min}(M)$ using additional characteristics of $M$ than just $n$ and $m$. For example, the connected components and incompatibility graph (as introduced in [6]) played an important role in studying recombination (e.g. [5]). So, there may be improved analytical upper bounds that are based on, say, number of connected components of the incompatibility graph, in addition to $n$ and $m$.

## Acknowledgments

I would like to thank Dan Gusfield for useful discussions. The research is supported by grant IIS-0803440 from National Science Foundation. and a grant from the Research Foundation of University of Connecticut. Part of work was performed when I was with University of California, Davis, and was supported by grants CCF-0515278 and IIS-0513910 from National Science Foundation.

## References

[1] V. Bafna and V. Bansal, Inference about Recombination from Haplotype Data: Lower Bounds and Recombination Hotspots. J. of Comp. Bio., v.13, p.501-521, 2006.

[2] M. Bordewich and C. Semple: On the computational complexity of the rooted subtree prune and regraft distance. Annals of Combinatorics, v.8, p.409-423, 2004.

[3] D. Gusfield: Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology, Cambridge University Press, 1997.

[1]This also follows from the recombination lower bound composition method [11]

[4] D. Gusfield: Optimal, efficient reconstruction of Root-Unknown phylogenetic networks with constrained and structured recombination. JCSS, 70, 381-398, 2005.

[5] D. Gusfield and V. Bansal and V. Bafna and Y. Song: A decomposition theory for phylogenetic networks and incompatible characters, Journal of Computational Biology, v. 14, pages 1247-1272, 2007.

[6] D. Gusfield, S. Eddhu and C. Langley: Optimal, efficient reconstruction of phylogenetic networks with constrained recombination. J. Bioinformatics and Computational Biology, v. 2, 173-213, 2004.

[7] J. Hein, M. Schierup and C. Wiuf: Gene Genealogies, Variation and Evolution: A primer in coalescent theory. Oxford University Press, 2005.

[8] International HapMap Consortium: A haplotype map of the human genome. Nature, 437, p1299-1320, 2005.

[9] International HapMap Consortium: A second generation human haplotype map of over 3.1 million SNPs, Nature, 449, p.851861, 2007.

[10] R. Hudson and N. Kaplan: Statistical properties of the number of recombination events in the history of a sample of DNA sequences. Genetics, v.111, p.147-164, 1985.

[11] S. R. Myers and R. C. Griffiths: Bounds on the minimum number of recombination events in a sample history. Genetics, v. 163, p.375-394, 2003.

[12] Y. S. Song and J. Hein: Constructing Minimal Ancestral Recombination Graphs. J. of Comp. Bio., 2005, 12, p159-178.

[13] 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. Bioinformatics, v. 421, p.i413-i422, Proceedings of ISMB 2005.

[14] L. Wang, K. Zhang and L. Zhang: Perfect Phylogenetic Networks with Recombination. J. of Comp. Bio., v.8, p.69-78, 2001.