Genome-wide compatible SNP intervals and their properties

Jeremy Wang
Dept. of Computer Science
University of North Carolina
Chapel Hill, NC 27599, USA

Kyle J. Moore
Dept. of Computer Science
University of North Carolina
Chapel Hill, NC 27599, USA

Qi Zhang
Dept. of Computer Science
University of North Carolina
Chapel Hill, NC 27599, USA

Fernando Pardo-Manual de Villena
Dept. of Genetics
University of North Carolina
Chapel Hill, NC 27599, USA

Wei Wang
Dept. of Computer Science
University of North Carolina
Chapel Hill, NC 27599, USA

Leonard McMillan
Dept. of Computer Science
University of North Carolina
Chapel Hill, NC 27599, USA

ABSTRACT

Intraspecific genomes can be subdivided into blocks with limited diversity. Understanding the distribution and structure of these blocks will help to unravel many biological problems including the identification of genes associated with complex diseases, finding the ancestral origins of a given population, and localizing regions of historical recombination, gene conversion, and homoplasy.

We present methods for partitioning a genome into blocks for which there are no apparent recombinations, thus providing parsimonious sets of compatible genome intervals based on the four-gamete test. Our contribution is a thorough analysis of the problem of dividing a genome into compatible intervals, in terms of its computational complexity, and by providing an achievable lower-bound on the minimal number of intervals required to cover an entire data set. In general, such minimal interval partitions are not unique. However, we identify properties that are common to every possible solution. We also define the notion of an interval set that achieves the interval lower-bound, yet maximizes interval overlap. We demonstrate algorithms for partitioning both haplotype data from inbred mice as well as outbred heterozygous genotype data using extensions of the standard four-gamete test. These methods allow our algorithms to be applied to a wide range of genomic data sets.

1. INTRODUCTION

The local block-structure of genotypes within a population sheds light on many biological questions [10]. Genotype blocks are central to quantifying and localizing recombinations (both recent and historical) [37, 36, 40], are widely used to identify informative marker sets [46], and are building blocks for constructing genetic maps [35]. Genotype-block structure also underlies many genome-wide association methods[45], provides biochemical evidence for selection [18], and offers a tool for ascertaining ancestral origins [9].

ACM-BCB 2010, Niagara Falls, NY, USA

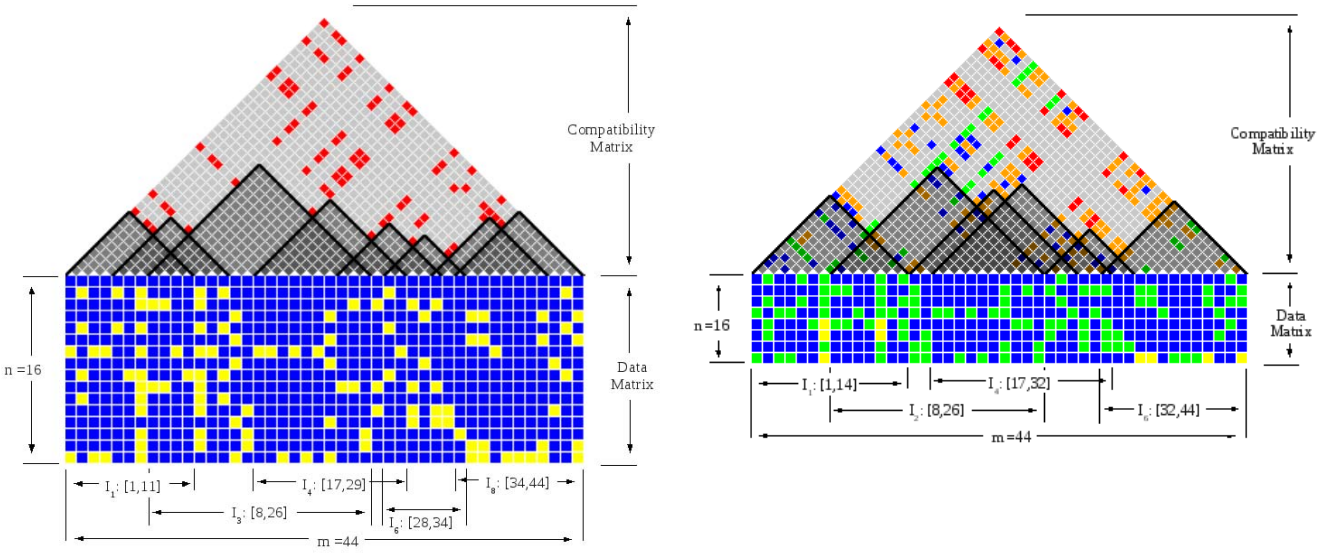
The task of decomposing a genome into meaningful blocks, however, has proven to be ill-defined, inconsistent, and often ambiguous [31, 34]. In part, the problem resides in the ad hoc definition of what constitutes a genotype block. Genotype blocks are often defined to serve a specific purpose. Examples include the minimum number of *tagging* SNPs sufficient to capture informative genotypes [32, 46], intervals of SNPs that exceed a given threshold of Linkage Disequilibrium (LD) [33], and maximal regions whose genotype diversity falls below a threshold [10]. Partitioning genotypes into blocks supporting perfect phylogenies [37, 17], and, the related, selection of blocks lacking evidence for recombination [41] are also used to construct Ancestral Recombination Graphs (ARGs).

We propose unambiguous definitions for haplotype and genotype blocks and efficient methods for computing them. Where ambiguity is unavoidable, we have uncovered properties common to all solutions. Our haplotype block definition directly supports, and has been used for, association mapping [30], construction of genetic maps [48], and determining ancestral origins within local genomic regions [49]. Our proposed genotype blocks can be used in much the same way.

Dense genotype data sets that are homozygous at every allele are available for many inbred mammal [15] and plant [6, 28] models commonly used for association mapping. However, haplotype data is not directly available for use in human studies. In our approach, it is unnecessary to phase such data sets, yet, we can still identify blocks important for exploring the local diversity structures [3, 24, 27, 29], and ancestral origins [44]. Like others [17, 41, 40], our blocks are chosen for their lack of historical recombination evidence.

Our methods can be used as an alternative to other block methods such as those in [13, 47, 16]. Particularly, the genotype block methods we introduce may be used to inform phasing [25] and to extend these methods to unphased genotype data. Block association methods such as Blossoc [26] and QBlossoc [2], which utilize small regions that admit perfect phylogenies could potentially benefit from our methods.

We define our blocks in terms of SNP compatibility according to the Four-Gamete Test (FGT) [22]. The FGT is of interest because of its close relation to perfect phylogeny [23]. Specifically, a neces-



(a) $C(h)_{Uber}$ (b) $C(g)_{Uber}$

Figure 1: Example data sets and Uber-cover. The data sets used in this figure will be used as running examples. The lower portion of the figure is the data matrix. The columns correspond to SNPs, and the rows correspond to haplotypes (a) and genotypes (b) derived by pairing adjacent haplotypes. Blue and yellow boxes represent homozygous alleles normalized such that the first sample is homozygous blue. Green boxes represent heterozygous alleles. The large triangles above the data matrices are the compatibility matrices, which show the compatibility of each pair of SNPs. Gray boxes imply that a pair of SNPs definitely does not violate the four-gamete rule. Red boxes imply that a pair of SNPs creates four gametes. Green boxes imply that a pair must be *in-phase* to be compatible. Orange boxes imply that a pair must be *out-of-phase* to be compatible. Blue boxes imply that a pair must be either *in-phase* or *out-of-phase* to be compatible.

sary and sufficient condition for a perfect phylogeny is that all pairs of SNPs satisfy the FGT [21]. For unphased genotype data, we define the notion of *optimistic* and *pessimistic* compatibility based on if a region *possibly* or *necessarily* passes the FGT. We partition the genome into a set of potentially overlapping, maximal compatible intervals, each of which admits a perfect phylogeny, and whose union covers the full data set. We address the question of what is the fewest number of such intervals required, and we also identify suspect SNPs whose removal reduces the overall complexity of the block structure (perhaps indicating genotyping errors, homoplasy, or gene conversions).

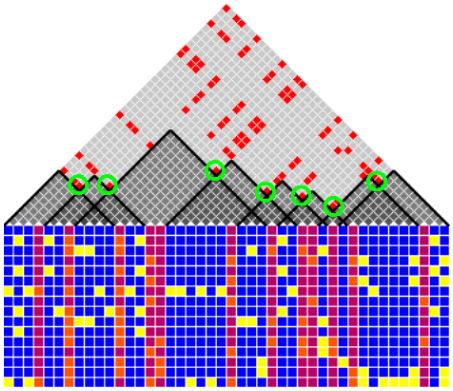
Our contribution is an analysis of the problem of dividing a genome into compatible intervals based on genotypes and its complexity. We provide an achievable lower-bound on the number of such intervals. While in general there are numerous ways of dividing a genome into a minimum number of compatible intervals (a fact overlooked by others [26, 40, 42]), we also identify non-overlapping *core* subintervals common to all valid solutions. We also define an interval set that achieves the interval lower-bound, yet maximizes the block overlap, thus minimizing the number of perfect phylogeny trees, while providing the richest possible set of SNPs to each tree.

2. RELATED WORK

There are three common approaches for partitioning haplotypes into blocks. The first employs LD measures [16, 33] and assigns blocks to regions with high pairwise LD within, and low LD between, blocks. A second class assigns blocks to regions of low sequence diversity [32]. Lastly, there are approaches that look for direct evidence of recombination, by either applying the FGT [22] and defining blocks as regions free of apparent recombination or homoplasy, or during the construction of ARGs, denoting supporting regions' component subtrees [37]. Schwartz et al. [34] performed an analysis of approaches and concluded that the block assignments of various methods differed markedly. Of these methods, the block boundaries of the FGT were better correlated to both the LD and diversity-based methods than these two methods were to each other.

Our approach partitions the genome into blocks satisfying the FGT. This is not new. The seminal work of Hudson and Kaplan provides a sketch of a greedy algorithm that processes SNPs in sequence order looking for runs of compatible intervals that are broken at points of incompatibility. This method appears to be widely used [26, 34, 40, 42]. A disconcerting feature of this approach is that one arrives at a different interval set if the genome is scanned in the reverse order (Fig. 4). Alternative sets of compatibility intervals arise when the region is grown maximally around each SNP [26]. Moreover, there appears to be many other possible partitions, begging the question of which block sets have the fewest intervals, and, of these sets, which minimizes haplotype diversity. In our model, each block is compatible with a perfect phylogeny (a side effect of the FGT) and overlaps between adjacent intervals are allowed.

We extend our methods to unphased genotype data. Little work has been done on partitioning genotype data without first phasing, however, there has been considerable work on the related topic of phasing by perfect phylogeny [1, 4, 14, 19, 12]. Such methods determine if a given genotype block admits a perfect phylogeny. Our



(a) *Flagging SNPs*

Figure 2: Uber intervals with flagging SNPs highlighted in red. Incompatibilities between flagging SNPs of adjacent maximal compatible intervals are highlighted with green circles.

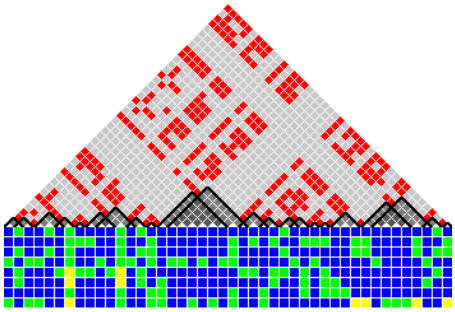
contribution is to apply the basic insights of these methods to extend the notion of a haplotype "scan" to the genotype case. Similar work has been done [11] in which local phylogenies are built over unphased genotype data to inform association mapping, however this work does not take full advantage of compatible blocks, employing a single-marker approach rather than a global block structure.

Past attempts at using perfect phylogeny to analyze genotypes assume they are given a region which admits a perfect phylogeny or does so within an error model. Most previous work ([12, 19, 14, 38, 1]) determines if the given data set does in fact admit a perfect phylogeny, and then solves the Perfect Phylogeny Haplotyping problem (PPH) ([19]) for the given instance. Recent extensions allow data to fall within some error model and subsequently handles cases where the data does not fit a perfect phylogeny. Error models include Missing Data (MD) and Character-removal (CR), and the algorithms attain a global perfect phylogeny while dealing with erroneous point cases ([21, 20]).

No previous approach considers the possibility of different PPH solutions as determined by the choice of block partition. While we do not propose haplotyping by perfect phylogeny, we use related techniques to partition the genome into blocks which satisfy a perfect phylogeny that could, in practice, then be haplotyped using any one of several previous algorithms. Introducing a genome-wide approach to perfect phylogeny rather than filtering out data as in [21, 20] considers many biological factors previously overlooked. The notion of recombination-free blocks in the genome is well-documented in humans and mice, as well as other species [16, 47, 40, 30, 28]. In many cases, regions of the genome on either side of a recombination point should realistically admit different phylogenies based on hybridization between subspecies. Simply removing presumed erroneous data and forcing regions separated by historical recombination into a global phylogeny ignores their biological relevance and produces a misleading solution. Our method of partitioning allows for biologically meaningful, though limited, regions with which to perform further analyses.

3. PRELIMINARIES

Throughout this paper we assume a data set of m SNPs spanning n haplotypes (or genotypes) that are represented as a binary data



(a) $C(p)_{Max}$

Figure 3: Pessimistic Max-k cover over the example set of genotypes. In the compatibility matrix, red indicates that the corresponding SNP pair is *possibly* incompatible, gray indicates that the pair is definitely compatible.

matrix S or ternary matrix S_g where each column corresponds to a SNP, and each row is a haplotype or genotype (Fig. 1). Alleles 0 and 1 represent alternative homozygous alleles and 2 represents heterozygous alleles.

A *compatible interval* over a set of haplotypes is a sequence of contiguous SNPs over S for which there are no violations of the FGT between any SNP pair. A compatible interval, $I_x = [b_x, e_x]$, includes all SNPs between the starting SNP s_{b_x} and ending SNP, s_{e_x}. Fig. 1(a) shows a data set of 16 haplotypes and 44 SNPs, together with eight compatible intervals, I_1 through I_8. Each interval covers a consecutive set of SNPs. For example, I_3 covers from s_8 to s_{26}. The triangular matrix above the SNP matrix is the pairwise compatibility matrix. If two SNPs exhibit four gametes, the corresponding matrix element is marked incompatible (red). Darkened triangles indicate sub-matrices corresponding to SNP pairs in the compatible intervals. Note that no triangles enclose red elements.

Compatible intervals over genotypes (S_g) are less straightforward due to ambiguities caused by heterozygous alleles. We define the notion of *optimistic* and *pessimistic* compatibility, whether genotypes are *possibly* or *nessesarily* compatible, respectively. Resolving genotype intervals requires more considerations when performing the FGT. Pairs of SNPs are evaluated to determine which gametes phasing could produce. In cases of homozygous-homozygous and homozygous-heterozygous pairs, the possible gametes are trivially determined. For example, the 0-0 produces only the 0-0 gamete and 0-2 produces the 0-0 and 0-1 gametes. Ambiguity is caused only by the 2-2 case– when there exist heterozygous alleles in the same sample at two different loci. These cases can produce two different sets of gametes, either 0-0 and 1-1, which we call *in phase* gametes, or 0-1 and 1-0, which we call *out of phase* gametes. There are three compatibility cases if 2-2 pairings exist. The ambiguous cases must be in phase in order to be compatible (if one of the 0-1 or 1-0 gametes are not present), must be out of phase (if one of the 0-0 or 1-1 gametes are not present), or truly ambiguous when all 2-2 pairs must simply be produce the *same* set of gametes since it is always possible to produce four gametes with opposite phasings of two 2-2 pairs.

The optimistic algorithm forms a graph with a vertex representing each locus and an edge representing a phasing (in phase, out of phase, or ambiguous) between two vertices. An interval is optimistically compatible iff there exists a bipartition of the graph into

two sets A and B such that no edge within A is out of phase, no edge within B is out of phase, no edge between sets A and B is in phase, and all ambiguous edges are uniquely resolvable to as either in phase or out of phase. We use an algorithm similar to [14] to partition the genome into blocks of genotypes which admit a perfect phylogeny. Similar to the haplotype "scan", we introduce SNPs one-by-one and test whether the resulting interval is internally compatible. For haplotypes, this is accomplished by pairwise comparisons of previous SNPs with every newly introduced SNP. For the genotype case, we define two interval types. For *optimistic* intervals, we use the idea of what Eskin et al [14] refer to as *equal* and *unequal* resolution to create a bipartite graph for each proposed interval. We "scan" SNPs as described in the haplotype case, adding these SNPs as vertices to the graph until the graph is no longer realizeable, thus ending an interval. *Pessimistic* intervals are unambiguously compatible regardless of the choice of phasing. When the scan is performed, an interval is ended as soon as it reaches a SNP which is possibly incompatible with any previous SNP in the interval. This is equivalent to considering all non-gray points as incompatible (red) and performing a haplotype scan to produce the *pessimistic* genotype intervals (see Fig. 3).

Fig. 1(b) shows a data set of 8 genotypes and 57 SNPs, together with four of its optimistic compatible intervals. It remains true that no interval may enclose an incompatible SNP pair. However, unlike the haplotype case, intervals are not necessarily bounded by red elements. As described, SNPs may be implicitly incompatible with a given interval if their addition forms an unrealizeable graph.

A compatible interval is *maximal* if it cannot be extended in either direction. All intervals in Fig. 1(a) (I_1, I_3, I_4, I_6, and I_8) are maximal, since further extension includes one or more incompatible SNP pairs. We denote the set of all maximal compatible intervals as C_{Uber}. Throughout, we will denote a cover over a genome generically by C. A cover of a set of haplotypes will be represented by $C(h)$. An optimistic cover of a set of genotypes will be represented by $C(g)$ and a pessimistic cover by $C(p)$. The darkened triangles in Fig. 1(a) depict $C(h)_{Uber}$. The two SNPs adjacent to a maximal compatible interval, s_{b_x-1} and s_{e_x+1}, are the *flagging SNPs* of the interval (Fig. 2). Note that flagging SNPs are incompatible with at least one SNP of the maximal compatible interval that it flanks.

A *cover*, $C_{x,y}$, is an ordered set of intervals, $C_{x,y}= \{I_1, I_2, \ldots, I_c\}$, where $b_i \leq b_{i+1}$, and every SNP in the range $[x, y]$ is covered by some interval in C but no SNP outside $[x, y]$ is covered by any interval in $C_{x,y}$. $C_{x,y}$ also satisfies $e_i \leq e_{i+1}$, since otherwise I_{i+1} is a fully contained subset of I_i. We call $C_{1,m}$ a *complete cover* of S, and $\|C\|$ is its cardinality. We will frequently refer to a cover, C, where $\|C\| = c$, as a *c-interval cover*, or simply as a *c-cover*. In addition, we will refer to special instances of complete covers by using descriptive subscripts, in which case a range from $[1, m]$ is implied. For example, in Fig. 1(a), $\{I_1, I_3, I_4\}$ is a $C(h)_{1,29}$ cover, $\{I_1, I_3, I_4, I_6, I_8\}$ is a complete 5-cover. In this paper, we are particularly interested in complete k-*cover*s, where k is a reachable lower bound on the number of intervals for the given SNP set.

In the following sections we provide an effective method for finding minimum-length complete covers for a given SNP set. Thus, establishing k as a tight lower bound. In general, there is no one unique k-cover for a given data set. We provide several algorithms that generate various k-covers in time linear to the number of genotypes. In addition, we examine features which are common to all k-covers of a given data set. We then present a linear-time algorithm for finding a cover composed entirely of maximal compatible intervals from C_{Uber}, where $\|C_{Uber}\| \geq k$. Finally, we present an algorithm for finding the k-cover with maximal overlap, the *Maximal-k-Cover* (C_{Max}). The cover C_{Max} is of particular interest since it leads to the construction of a parsimonious set of perfect phylogeny trees where each incorporates maximal information (i.e. the maximum number of SNPs per tree). Finally, we present an algorithm for finding critical SNPs in S whose removal reduces, $\|C_{Max}\|$, from k to $k-1$ or smaller, using a number of tests that are proportional to $\|C_{Uber}\|$ rather than m.

4. A LOWER BOUND ON THE NUMBER OF INTERVALS IN A COMPLETE COVER

4.1 Left-to-Right and Right-to-Left Covers

We first define two non-overlapping covers, the Left-to-Right cover ($C(h)_{LR}$), and the Right-to-Left cover ($C(h)_{RL}$). A simple greedy algorithm, LRScan, whose pseudocode is given in Appendix B, finds $C(h)_{LR}$ over a set of haplotypes and it has been previously described in [40]. It begins at the leftmost SNP (s_1), and either extends or terminates the current active interval as it considers each SNP in sequence order. LRScan performs FGTs of the candidate SNP against those SNPs already in the active interval. If four gametes occur, the active interval is closed, and a new interval begins from the candidate, otherwise the SNP is added to the active interval. This continues until the last SNP is reached, thus closing the final interval (see Fig. 4(a)).

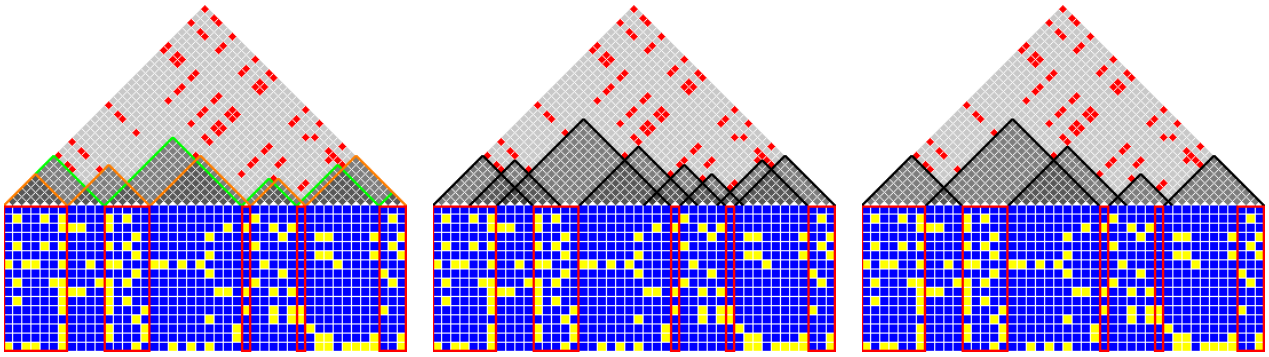
The run-time of LRScan depends on the number of SNPs, m, and the number of the FGTs performed for each SNP. Since the maximum number of distinct compatible SNP patterns (SDPs) that can be mutually compatible among n haplotypes is $2n-3$ [38], the FGT requires only $O(n)$ operations per SNP, assuming a constant-time overhead for each FGT. Therefore, the complexity for LRScan is $O(mn)$, and thus is linear in the number of genotypes.

A similar greedy Right-to-Left scan algorithm (RLScan) generating $C(h)_{RL}$ can be defined via straightforward modifications to (LRScan). Likewise $C(h)_{RL}$ can be generated by merely reversing the input sequence, applying LRScan, and adjusting the indices of the resulting intervals, including their starting and ending positions. Note that a cover's interval indices are assigned according to the sequence order, regardless of the scanning direction.

We define a similar notion over a set of genotypes. As described in Section 3, the only difference is the manner in which the FGT is performed. We find an *optimistic* left-to-right ($C(g)_{LR}$) and right-to-left ($C(g)_{RL}$) cover by closing an interval only when the subsequent SNP will be definitely and unambiguously incompatible with a SNP in the interval (regardless of the phasing chosen). Likewise, a *pessimistic* interval ($C(p)_{LR}$ and $C(p)_{RL}$) is closed off if there exists a phasing of the genotype set for which the next SNP will be incompatible with SNPs already in the interval.

The run-time of the pessimistic genotype scan is also $O(mn)$. Like the haplotype case, there is a limit on the number of distinct SNPs that can be compatible among n genotypes which is linear in n. Similarly, the adjusted FGT requires only $O(n)$ operations per SNP to determine if there exists any possible incompatibility.

The run-time of the optimistic genotype scan is more complex. Eskin et al ([14]) propose an alogorithm with $O(nm^2)$ complexity to determine if a single region admits a perfect phylogeny. We use a



(a) C_{LR} *and* C_{RL} (b) C_{Uber} *with cores* (c) C_{Max}

Figure 4: Shown above is the progression of covers. (a) depicts C_{LR} (green) and C_{RL} (orange) and their overlap (cores) outlined in red. (b) shows C_{Uber} and the cores from (a). In (c), each C_{Max} interval (a subset of C_{Uber}) encloses exactly one core.

similar algorithm, adding SNPs incrementally. Since each interval is bounded and we scan linearly across the genome, this allows for an $O(nm^2)$ algorithm to partition the entire genome.

4.2 Properties of C_{LR} and C_{RL}

Our first theorem states, "*The covers C_{LR} and C_{RL} have the same number of intervals k, and k is the minimal number of intervals possible for any complete cover.*" A proof of this and an associated Lemma 1 are provided in Appendix A. Moreover, certain core subintervals are common to all complete k-covers of a given SNP set. We define the intersections of corresponding intervals from C_{LR} and C_{RL} as *cores* (Fig. 4). According to Lemma 1, $e_i^R \leq e_i^L$ and $b_i^L \geq b_i^R$ (Fig. 10(a), Appendix), therefore $Core_i = [b_i^L, e_i^R]$. Lemma 2 and a detailed proof of this claim are included in Appendix A. Theorem 2 states, "*For any complete k-cover $C_{1,m}=\{I_1, \ldots, I_k\}$, the ith interval contains the entire ith core: $Core_i \cap I_i = Core_i$, and, it does not contain any part of another core $Core_j \cap I_i = \phi$, $1 \leq j \leq k, j \neq i$.* This is due to the interleaving of the non-overlapping intervals of C_{LR} and C_{RL}. $Core_i$ is necessarily compatible with both C_{LR_i} and C_{RL_i} and cannot be extended beyond the outside boundary of either. Therefore, no part of any two cores may be a part of the same interval. *A k-cover must contain k intervals each containing one core exclusively*. This leads to two corollaries. The first is that any interval that does not contain an entire core is not part of any complete k-cover and the second is that the ith core is only contained within the ith interval of any k-cover. Cores have several interesting properties worth noting. *All SNPs in a core are compatible, since each core is an intersection of two compatible regions* and *adjacent cores must contain at least one pair of incompatible SNPs*. Proofs of these properties are given in Appendix A.

5. MAXIMAL-K-COVER ALGORITHM

First we introduce UberScan, which generates the set of all the maximal compatible intervals, C_{Uber}. UberScan (Appendix B), is similar to the LRScan. Whenever a compatible interval ends at SNP s_i, instead of starting the next interval from s_{i+1} as LRScan does, UberScan finds the nearest SNP s_j $(j < i+1)$ that is incompatible with s_{i+1}, and the following SNP, s_{j+1}, begins the next interval. Note that s_{i+1} is a flagging SNP of the previous maximal compatible interval and s_j is a flagging SNP of the next maximal compatible interval. UberScan is a simple modification of LRScan with added bookkeeping to track of the index of the last occurrence of each unique SNP pattern[1]. A similar analysis of LRScan shows that UberScan also takes $O(mn)$ time. UberScan generates C_{Uber}, containing all maximal intervals of S, and generally, $\|C_{Uber}\| \gg k$. C_{Uber} contains all candidates for the Maximal-k-cover, C_{Max}, since a cover with maximal overlap must be composed of maximal intervals.

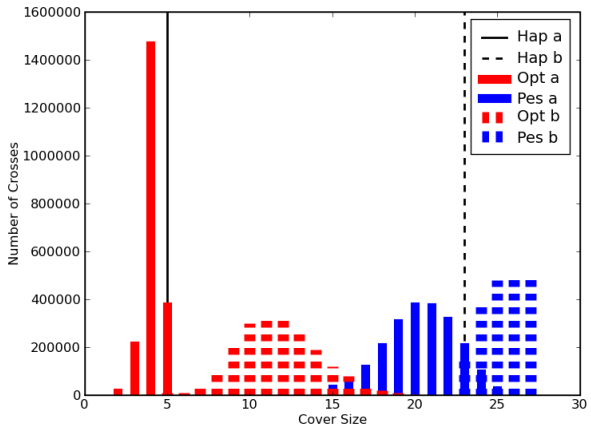
5.1 Finding the Maximal-k-cover by finding the longest path in a k-partite graph

A Maximal-k-cover, C_{Max}, is of particular interest as it covers the entire SNP set using the fewest, k, maximal intervals. While, C_{Max} is not necessarily unique, alternate solutions are generally similar. Next we provide a fast graph algorithm to compute all Maximal-k-covers.

We consider only those maximal intervals in C_{Uber} that entirely enclose a single core and no part of a second core, as defined by C_{LR} and C_{RL} (4.2). According to Theorem 2, these intervals are the candidates for Maximal-k-covers. Next, we organize the candidates into k groups according to the core it contains. Each core is contained within at least one maximal interval, thus, no group is empty. We then examine the overlap between groups. A candidate interval in group i can only overlap candidates from groups $i-1$ or $i+1$, since any two candidates enclosing non-adjacent cores (say $Core_x$ and $Core_y$) imply that at least one of them contains part of the core between $Core_x$ and $Core_y$, contradicting its qualification as a candidate.

The Maximal-k-cover problem is solved by recasting it as finding the longest path in a directed k-partite graph. Specifically, each candidate maximal interval is mapped to a node and each group as a part, with part i containing all the candidates covering $Core_i$. An edge connects nodes corresponding to overlapping candidates. Each edge's weight is the amount of overlap between the two intervals. The edge is directed towards the candidate that contains the next core in the sequence. As shown previously edges only exist between adjacent parts. Finally, we add a source with edges to all nodes in part 1 and a sink with edges from all the nodes in part k, both types of edges have weight 0. Finding a C_{Max} solution corresponds to finding the longest path in this directed graph with $k+1$ edges from source to sink. Note that the greedy approach of taking the largest interval that encloses each core does not always yield a

[1]Recall that the maximum number of distinct SNP patterns within a compatible interval is $2n-3$, or $O(n)$.



(a)

Figure 5: The distribution of cover sizes for genotypes resulting from all pairings of a set of haplotypes. For 'a' distributions, data was simulated using the infinite sites model and recombination. The source haplotypes cover size (the "ground truth") is represented by the solid black line. For 'b' distributions, a contrived haplotype set was made by phasing a pessimistic genotype result of 'a'. The source haplotypes cover size is the dashed black line.

correct answer, as shown in the fourth core of our running example (compare Fig. 2 (a) and Fig. 4 (c)).

The problem is a single-source shortest path problem for a weighted DAG, except that we search for longest path (maximizing instead of minimizing the sum of weight on the path) with a constraint on the number of steps. The constraint can be ignored since all edges lead from one part to the next, thus any path from source to sink will have $k+1$ steps. The problem can be solved using dynamic programming and requires only $\Theta(|V|+|E|)$ time, where V is the set of nodes, and E is the set of edges [8].

5.2 Critical SNPs

A *critical SNP* is any SNP whose removal reduces k, the minimum number of intervals required to cover the given SNP set. To check whether a SNP is critical, one could simply remove each SNP and recalculate k by either a LRScan or an RLScan. This naive approach requires m scans, and takes $O(nm^2)$ time. However, it is unnecessary to test every SNP. In fact, only flagging SNPs of maximal compatible intervals need to be considered. A flagging SNP bounding an interval on one side prevents the interval from growing toward an adjacent interval on that side, therefore a flagging SNP must be removed to allow any interval to grow, a necessary condition of reducing k. A complete proof is given in Appendix A. Since each maximal compatible interval has two flagging SNPs, one on each side of the interval, the total number of flagging SNPs is $O(\|C_{Uber}\|)$. The running-time for computing critical SNPs is $O(nm\|C_{Uber}\|)$.

6. PROPERTIES OF GENOTYPE INTERVAL COVERS

Determining four-gamete compatibility and compatible intervals over unphased genotype data is ambiguous in that there are many possible interpretations (phasings) of a set of genotypes as haplotypes. We define two approaches for determining compatibility among genotypes without explicitly phasing. The *optimistic* method determines intervals for which there might exist a phasing such that the interval *is* four-gamete compatible. The *pessimistic* method determines intervals by choosing phasings such that the current SNP is incompatible with the current interval wherever possible.

We compared the haplotype and genotype intervals on three data sets. First, we created simulated genotype data using a simple infinite-sites model of mutation with cross-over recombination - this served as a data source devoid of confounding factors such as experimental error and homoplasy. In the second, we used real data from F1 crosses between isogenic mouse strains. Lastly, we used two populations of HapMap data.

6.1 Relating Genotype and Haplotype Covers.

In simulated data, one can explore relationships between the compatible intervals of genotypes and the compatible intervals of their "source" haplotypes. Such simulated data can be used as ground truth data for a set of genotypes. The number of intervals in a pessimistic genotype cover is greater than or equal to the number of ground truth intervals. Thus, the number of intervals in an optimistic and pessimistic genotype scan are lower and upper bounds on the true number of intervals. Proof of these claims are provided by Theorems 4 and 5 of Appendix A.

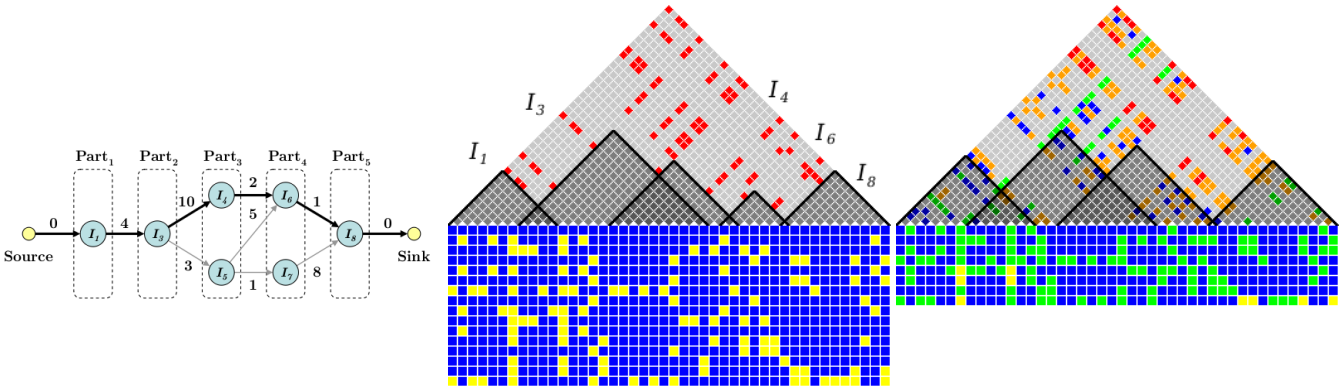
Fig. 5(a) represents the distribution of the set of all possible "pairings" of a fixed haplotype set into genotypes. For 'a' distributions, the "ground truth", or the number of intervals required to form a cover using the source haplotypes, is 5. Notice that the covers resulting from every *optimistic* genotype scans fall closer to the "ground truth" than those from every *pessimistic* scans. From our experimental results, we observe this is a common case. In contrast, 'b' distributions represent the same plot for a contrived, non-biology based, haplotype data set. These represent the distribution of the cover size of all genotypes that can be formed by pairings of the haplotype set that acheives one of the *pessimistic* covers from 'a' (this can always be acheived, as discussed in 6.2). Specifically, the "true" cover size for this contrived set is 23. Notice that the distribution is different from the biology-based model. In particular, the *optimistic* and *pessimistic* distributions are closer together and the "ground truth" is nearer the *pessimistic* estimations.

6.2 Acheiving Genotype Covers by Phasing

In many circumstances, it is useful to determine if a particular genotype cover or interval set is acheivable by phasing. Pessimistic genotype covers can always be acheived (see Algorithm 2 given in Appendix B). However, there does not always exist a phasing to accomplish a given optimistic genotype interval, in practice many candidate covers can be proposed, and it is trivial to verify candidate intervals using existing PPH methods ([19, 14, 1, 4]).

7. EXPERIMENTS AND RESULTS

We tested the performance of our algorithms on real datasets. The first is based on 8 inbred mouse strains and selected F1 crosses between these strains using a newly developed genome-wide genotyping platform ([44]). The second and third are human datasets from the International HapMap Project [7]. The first HapMap dataset describes a population of Utah residents from northern and western Europe (CEU). The second is a Yoruba population from Ibadan, Nigeria (YRI). The mouse genome contains 20 chromosomes (chromosome 1-19 and chromosome X). The number of SNPs per chro-



(a) (b) $C(h)_{Max}$ (c) $C(g)_{Max}$

Figure 6: (a) shows the k-partite graph used to find $C(h)_{Max}$ for the running example. The node represents the interval, and edges connect intervals that overlap, with weight representing the number of shared SNPs. The longest path (bold) computed from source to sink corresponds to C_{Max}. It contains intervals I_1, I_3, I_4, I_6, and I_8 from C_{Uber}, with a total overlap of 17. (b) is the Maximum-k-cover of the set of haplotypes, $C(h)_{Max}$, containing I_1, I_3, I_4, I_6, and I_8. (c) is the optimistic Maximum-k-cover of the set of genotypes, $C(g)_{Max}$.

mosome varies from 15K to 50K and includes strains: 129SvlmJ, A/J, C57BL/6J, CAST/EiJ, NOD/LtJ, NZO/HILtJ, PWK/PhJ, and WSB/EiJ along with 37 F1 crosses. The inbred founder mouse strains were used in the haplotype analysis, while the F1 crosses were used in the genotype analyses. With this mouse data set, we are able to ascertain an approximate ground truth, sans genotyping errors, with real-world rather than simulated data. Only fully informative SNPs among the 8 strains were used in our analysis, reducing the 600K total SNPs to 340K.

We found compatible intervals for 23 human chromosomes from the phased HapMap data (Chromosome 1-22 and Chromosome X). The CEU dataset has 348 haplotypes (174 individuals) with 34K - 222K SNPs per chromosome and the YRI dataset, has 348 haplotypes (174 individuals) with 38K - 252K SNPs per chromosome. Since the phasing method used by HapMap also imputes missing data, all data was used to evaluate the haplotype methods. The phased CEU data set has 2.6M SNPs and YRI has 2.9M SNPs.

Our algorithms were implemented in Python 2.6 and experiments were performed on a 2.67GHz Intel Core i7 processor with 8.0GB of RAM.

7.1 Run-times

The performance of interval scanning algorithms (LRScan, RLScan and UberScan) is linear in the number of genotypes, which enables us to compute C_{LR}, C_{RL}, and C_{Uber} efficiently. The run-times of all three scans and the Max-k-cover algorithm was recorded for all the data sets. Fig. 7(a) gives the times for calculating the haplotype interval covers on the mouse data set (inbred founders) and genotype covers for the F1 crosses. Fig. 7(b) shows run times for all covers over the HapMap CEU data set (YRI is similar). As shown, the run-times for haplotype and pessimistic genotype scans are linear in the number of genotypes. Optimistic genotype scans are quadratic in the number of SNPs in the largest interval, which is not bounded by the number of genotypes as in the other scans, but is far smaller than the size of the genome. Since the LRScan and RLScan are symmetric procedures, they have similar run-times. UberScan involves more bookkeeping and, thus, has higher times than the other scans. The optimistic genotype scan times are much higher than the haplotype and pessimistic scans due to the computationally intensive graph algorithm which must be performed at each step. Run-times for all scans vary additionally across chromosomes depending on properties of the data such as interval size.

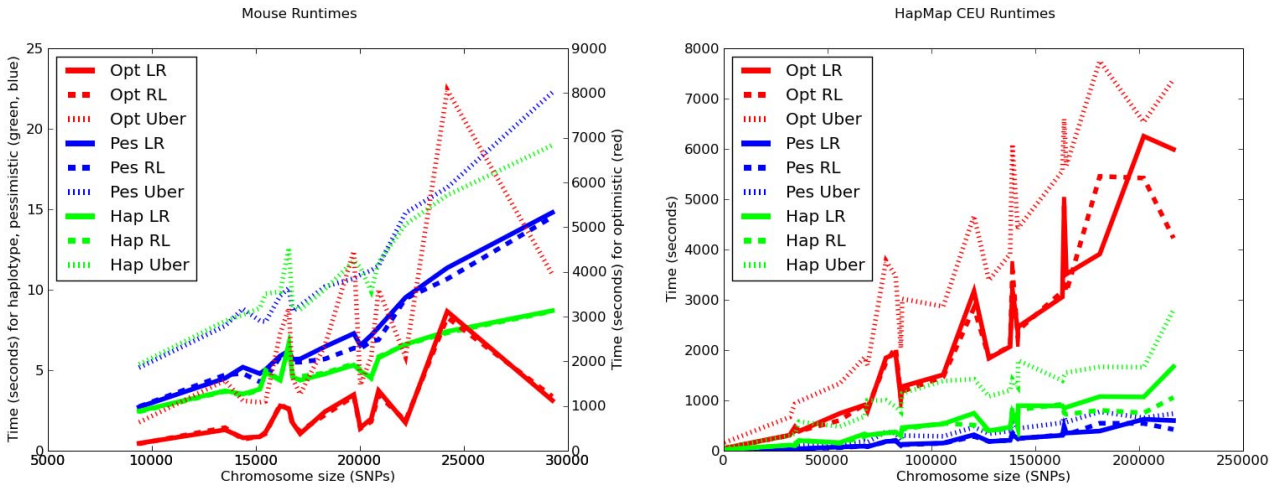
The k-partite graph component of the Max-k-cover algorithm takes the intervals of C_{Uber} as input. Fig. 8(b) shows the run-time of the Max-k-cover algorithm as a function of the number of intervals $\|C_{Uber}\|$. The run-time of the Max-k-cover algorithm has a linear relationship with the cardinality of C_{Uber}.

7.2 Interval and Core Statistics

We also collected various statistics over the C_{Max} intervals and cores, including interval lengths in terms of SNPs and genomic position, and the number of distinct haplotypes. The interval lengths shown in Figure 9 illustrate the prevalence of many long blocks (suggestive of conserved regions) punctuated by smaller intervals indicative of hot spots. Numbers of distinct haplotypes (Fig. 9(b)) indicate the relative diversity within each block. Note that the maximum number of distinct haplotypes is $\texttt{Min}(n, s+1)$, where s is the number of unique SNPs in the interval.

A large percentage of Max-k intervals in the optimistic genotype cover of the human CEU data set contain only a single SNP. By definition, these are Critical SNPs - removing one reduces the total number of intervals k by at least one. Such one-SNP intervals, which are incompatible with SNPs immediately adjacent, are not likely to be informative regarding recombinations and probably represent other biological or experimental artifacts such as genotype errors, homoplasy, or gene conversions. This explains the prevalence of intervals with two unique haplotypes as shown in Fig. 9(b), and the large number of single-SNP intervals, as shown in Fig. 9(a). Fig. 9(a) also shows the distribution of interval sizes in SNPs of the Max-k cover after the SNPs making up these single-SNP intervals have been removed. Notice that the average interval size is greater for all three cover types after removing these data.

The relationship between the haplotype covers and the optimistic and pessimistic genotype covers is illustrated in Figures 8 and 9. Fig. 8(a) shows the number of intervals in a k-cover of the CEU human data set for each chromosome. These covers demonstrate Theorems 4 and 5 (Appendix A) that $\|C(g)\| \leq \|C(h)\| \leq \|C(p)\|$. Thus, the genotype scans serve as effective upper and lower bounds on the "ground truth".



(a) *Mouse cover run-times*

(b) *HapMap CEU cover run-times*

Figure 7: Run-times to calculate covers over real data sets. (a) shows C_{LR}, C_{RL}, and C_{Uber} run-times versus the number of size of chromosome in SNPs for haplotype, optimistic genotype, and pessimistic genotype covers over the mouse data set. (b) similarly dipicts run-times of the HapMap CEU population.

8. DISCUSSION AND FUTURE WORK

By providing an effective means of partitioning haplotypes and genotypes into meaningful blocks on a genome-wide scale, we have enabled several new areas for exploration. An obvious application of FGT compatible intervals is to construct local perfect phylogeny trees, in an effort to find sets of frequently recurring and compatible trees [43]. This question is of particular importance in model organisms such as laboratory mice that are thought to derive from small set of founders (i.e., fancy mice) [44], and in communities where there are ongoing efforts to generate new model populations for systems biology [5]. Our method is effectively used in Yang *et al* (2010, in preparation) to identify meaningful blocks over which the historical subspecific origin of laboratory mice can be analyzed. Moreover, one can precisely define the *core* of such trees and quantify their variability. Both local phylogenetic trees derived from compatible SNP intervals and the limited haplotype diversity of compatible SNP intervals can be incorporated into disease association studies, as has been recently demonstrated [39, 26, 30]. With our introduction of a method for finding compatible intervals over outbred populations, it may be possible to gain the same benefits working with less controlled populations, including the human genome.

The prevalence of single-SNP cores in our results also suggests new methods for cleaning data. There are several possible sources for these small local features including genotyping errors, gene conversions, and homoplasy. In addition to the obvious benefits of eliminating putative errors from a given data set, the other two sources for single-SNP cores are of great interest to biologists, but are not well-characterized. One would expect that a systematic greedy reduction of an interval set from k to $k-1$ or $k-2$ intervals would expose larger scale structure and phylogenetic trees with improved support.

9. REFERENCES

[1] Vineet Bafna, Dan Gusfield, Giuseppe Lancia, and Shibu Yooseph. Haplotyping as perfect phylogeny: A direct approach. *Journal of Computational Biology*, 10(3-4), 2003.

[2] S. Besenbacher, T. Mailund, and M.H. Schierup. Local phylogeny mapping of quantitative traits: Higher accuracy and better ranking than single-marker association in genomewide scans. *Genetics*, 181:747–753, 2009.

[3] E. Buckler and M. Gore. An arabidopsis haplotype map takes root. *Nature Genetics*, 39(9):1056–1057, 2007.

[4] Ren Hua Chung and Dan Gusfield. Perfect phylogeny haplotyper: haplotype inferral using a tree model. *Bioinformatics Application Note*, 19(6), 2003.

[5] G. A. et al. Churchill. The collaborative cross, a community resource for the genetic analysis of complex traits. *Nature Genetics*, 36:1133–1137, 2004.

[6] R.M. Clark, G. Schweikert, C. Toomajian, S. Ossowski, G. Zeller, P. Shinn, N. Warthmann, T.T. Hu, G. Fu, D.A. Hinds, H. Chen, K.A. Frazer, D.H. Huson, B. Scholkopf, M. Nordborg, G. Ratsch, J.R. Ecker, and D. Weigel. Common sequence polymorphisms shaping genetic diversity in arabidopsis thaliana. *Science*, 317:338–342, 2007.

[7] The International Hapmap Consortium. A haplotype map of the human genome. *Nature*, 437:1299–1320, 2005.

[8] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms.* MIT Press, 2004.

[9] E. Cuppen. Haplotype-based genetics in mice and rats. *TRENDS in Genetics*, 21(6), 2005.

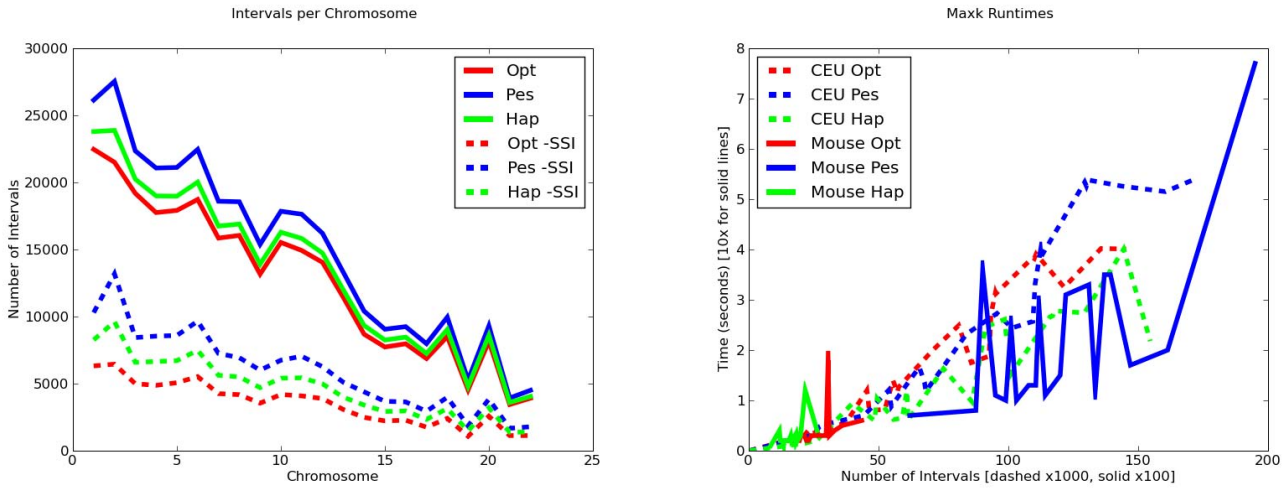
[10] M.J. Daly, J.D. Rioux, S.F. Schaffner, T.J. Hudson, and E.S. Lander. High-resolution haplotype structure in the human genome. *Nature Genetics*, 29:229–232, 2001.

[11] Z. Ding, T. Mailund, and Y.S. Song. Efficient whole-genome association mapping using local phylogenies for unphased genotype data. *Bioinformatics*, 24(19):2215–2221, 2008.

[12] Zhihong Ding, Vladimir Filkov, and Dan Gusfield. A linear-time algorithm for the perfect phylogeny haplotyping (pph) problem. *RECOMB*, 2005.

[13] E. Eskin, E. Halperin, and R.M. Karp. Large scale reconstruction of haplotypes from genotype data. *RECOMB*, 2003.

[14] Eleazar Eskin, Eran Halperin, and Richard M Karp. Efficient reconstruction of haplotype structure via perfect phylogeny. *Journal of bioinformatics and computational biology.*, 2003.



(a) *Max-k cover sizes per chromosome*

(b) *Max-k graph solution run times*

Figure 8: (a) shows the number of intervals required to form a k-cover over each chromosome before and after removing the SNPs from single-SNP intervals in the optimistic genotype cover. (b) shows the run times of the k-partite graph component of the Maximum-k-cover algorithm for the haplotype, optimistic genotype, and pessimistic genotype intervals over human and mouse data sets. The algorithm is linear in the number of Uber intervals $\|C_{Uber}\|$. All of these figures plot Max-k intervals over the CEU human data set.

[15] K.A. Frazer, E. Eskin, H.M. Kang, M.A. Bogue, D.A. Hinds, E.J. Beilharz, R.V. Gupta, J. Montgomery, M.M. Morenzoni, G.B. Nilsen, C.L. Pethiyagoda, L.L. Stuve, F.M. Johnson, M.J. Daly, C.M. Wade, and D.R. Cox. A sequence-based variation map of 8.27 million snps in inbred mouse strains. *Nature*, 2007.

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

[17] J. Gramm, T. Hartman, T. Nierhoff, R. Sharan, and T. Tantau. On the complexity of snp block partitioning under the perfect phylogeny model. *Discrete Mathematics*, 4175:92–102, 2008.

[18] V. Guryev, B.M.G. Smits, J. van de Belt, M. Verheul, N. Hubner, and E. Cuppen. Haplotype block structure is conserved across mammals. *PLoS Genetics*, 2(7):e121, July 2006.

[19] Dan Gusfield. Haplotyping as perfect phylogeny: Conceptual framework and efficient solutions. *RECOMB*, 2002.

[20] Dan Gusfield. The multi-state perfect phylogeny problem with missing and removable data: Solutions via integer-programming and chordal graph theory. *RECOMB*, 2009.

[21] E. Halperin and R.M. Karp. Perfect phylogeny and haplotype assignment. *Proceedings of RECOMB*, pages 10–19, 2004.

[22] R.R. Hudson and N.L. Kaplan. Statistical properties of the number of recombination events in the history of a sample of dna sequences. *Genetics*, 111:147–164, 1985.

[23] J. Kececioglu and D. Gusfield. Reconstructing a history of recombinations from a set of sequences. *Proceedings of SODA*, pages 471–480, 1998.

[24] S. Kim, V. Plagnol, T.T. Hu, C. Toomajian, R.M. Clark, S. Ossowski, J.R. Ecker, D. Weigel, and M. Nordborg. Recombination and linkage disequilibrium in arabidopsis thaliana. *Nature Genetics*, 39(9):1151–1155, 2007.

[25] Gad Kimmel and Ron Shamir. Gerbil: Genotype resolution and block identification using likelihood. *PNAS*, 102(1):158–162, 2005.

[26] T. Mailund, S. Besenbacher, and M. Schierup. Whole genome association mapping by incompatibilities and local perfect phylogenies. *BMC Bioinformatics*, 7:254, 2006.

[27] P. McClurg, M.T. Pletcher, T. Wiltshire, and A.I. Su. Comparative analysis of haplotype association mapping algorithms. *BMC Bioinformatics*, 7(1):61–76, 2006.

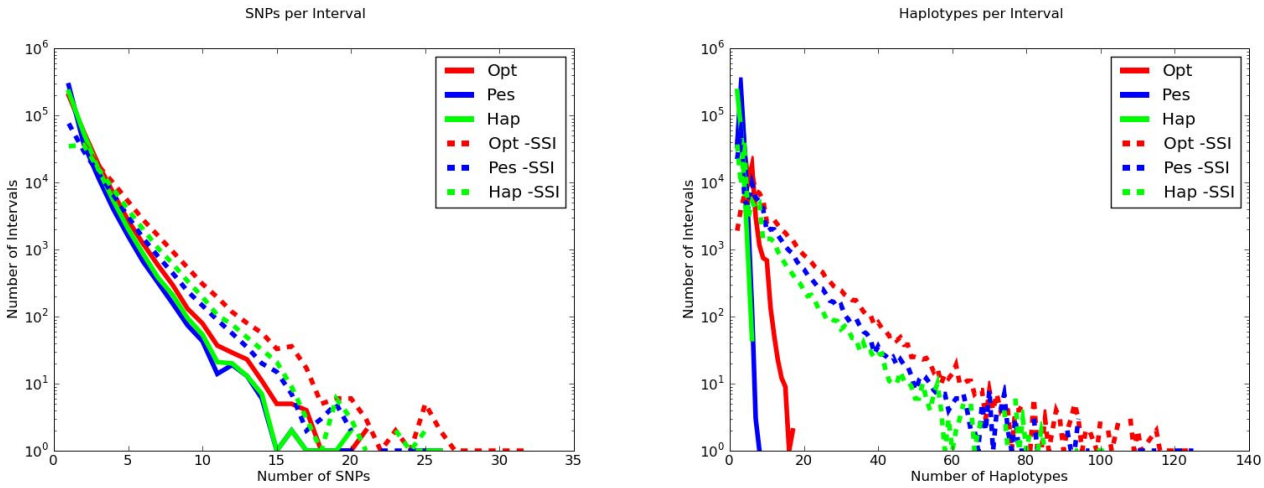
[28] Michael D. McMullen, Stephen Kresovich, Hector Sanchez Villeda, Peter Bradbury, Huihui Li, Qi Sun, Sherry Flint-Garcia, Jeffry Thornsberry, Charlotte Acharya, Christopher Bottoms, Patrick Brown, Chris Browne, Magen Eller, Kate Guill, Carlos Harjes, Dallas Kroon, Nick Lepak, Sharon E. Mitchell, Brooke Peterson, Gael Pressoir, Susan Romero, Marco Oropeza Rosas, Stella Salvo, Heather Yates, Mark Hanson, Elizabeth Jones, Stephen Smith, Jeffrey C. Glaubitz, Major Goodman, Doreen Ware, James B. Holland, and Edward S. Buckler. Genetic properties of the maize nested association mapping population. *Science*, 325(5941):737–740, 2009.

[29] R. Mott and J. Flint. Simultaneous detection and fine mapping of quantitative trait loci in mice using heterogeneous stocks. *Genetics*, 160:1609–1618, 2002.

[30] F. Pan, L. McMillan, F. Pardo-Manuel de Villena, D. Threadgill, and W. Wang. Treeqa: Quantitative genome wide association mapping using local perfect phylogeny trees. *PSB*, January 2009.

[31] P. Paschou, M.W. Mahoney, A. Javed, J.R. Kidd, A.J. Pakstis, S. Gu, K.K. Kidd, and P. Drineas. Intra- and interpopulation genotype reconstruction from tagging snps. *Genome Research*, 17(1):96–107, 2006.

[32] N. Patil, A.J. Berno, D.A. Hinds, W.A. Barrett, J.M. Doshi, C.R. Hacker, C.R. Kautzer, D.H. Lee, C. Marjoribanks, D.P. McDonough, B.T.N. Nguyen, M.C. Norris, J.B. Sheehan, N. Shen, D. Stern, R.P. Stokowski, D.J. Thomas, M.O.



(a) *Interval Size in SNPs*

(b) *Unique Haplotypes per Interval*

Figure 9: Interval size statistics for the CEU human population. A distribution of the number of SNPs per interval over this data set before and after removal of single-SNP intervals is represented by (a). (b) represents the distribution of the number of unique haplotypes per interval.

Trulson, K.R. Vyas, K.A. Frazer, S.P.A. Fodor, and D.R. Cox. Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. *Science*, 294(23):1719–1723, 2001.

[33] D.E. Reich, M. Cargill, S. Bolk, J. Ireland, P.C. Sabeti, D.J. Richter, T. Lavery, R. Kouyoumjian, S.F. Farhadlan, R. Ward, and E.K. Lander. Linkage disequilibrium in the human genome. *Nature*, 411:199–204, 2001.

[34] R. Schwartz, B.V. Halldorsson, V. Bafna, A.G. Clark, and S. Istrail. Robustness of inference of haplotype block structure. *Journal of Computational Biology*, 10:13–19, 2003.

[35] S. Shifman, J.T. Bell, R.R. Copley, M.S. Taylor, R. Williams, R. Mott, and J. Flint. A high-resolution single nucleotide polymorphism genetic map of the mouse genome. *PLoS Biology*, 4(12):e395, 2006.

[36] Y.S. Song, Z. Ding, D. Gusfield, C.H. Langley, and Y. Wu. Algorithms to distinguish the role of gene-conversion from single-crossover recombination in the derivation of snp sequences in populations. *Journal of Computational Biology*, 14(10):1273–1286, 2007.

[37] Y.S. Song, Y. Wu, and D. Gusfield. Efficient computation of close lower and upper bounds on the minimum number of recombinations in biological sequence evolution. *Bioinformatics*, 21:413–422, 2005.

[38] S. Sridhar, F. Lam, G.E. Blelloch, R. Ravi, and R. Schwartz. Efficiently finding the most parsimonious phylogenetic tree via linear programming. *LNCS:Proceedings of ISBRA*, 4463:37–48, 2007.

[39] A.R. Templeton, T. Maxwell, D. Posada, J.J. Stengard, E. Boerwinkle, and C.F. Sing. Tree scanning: A method for using haplotype trees in phenotype/genotype association studies. *Genetics*, 169:441–453, 2005.

[40] N. Wang, J.M. Akey, K. Zhang, R. Chakraborty, and L. Jin. Distribution of recombination of crossovers and the origin of haplotype blocks: The interplay of population history, recombination, and mutation. *American Journal of Human Genetics*, 71:1227–1237, 2002.

[41] C. Wiuf. Inference on recombination and block structure using unphased data. *Genetics*, 166:537–545, 2004.

[42] A. Woerner, M. Cox, and M. Hammer. Recombination-filtered genomic datasets by information maximization. *Bioinformatics*, 23(14):1851–1853, 2007.

[43] Y. Wu and D. Gusfield. Improved algorithms for inferring the minimum mosaic of a set of recombinations. *LNCS:Proceedings of Combinatorial Pattern Matching*, 4580:150–161, 2007.

[44] Y. Yang, T.A. Bell, G.A. Churchill, and F. Pardo-Manuel de Villena. On the subspecific origin of the laboratory mouse. *Nature Genetics*, 39(9):1100–1107, 2007.

[45] K. Zhang, M. Deng, P. Calabrese, M. Nordborg, and F. Sun. Haplotype block structure and its applications to association studies: Power and study designs. *American Journal of Human Genetics*, 71:1386–1394, 2002.

[46] K. Zhang, M. Deng, T. Chen, M.S. Waterman, and F. Sun. A dynamic programming algorithm for haplotype block partitioning. *PNAS*, 99(11):7335–7339, 2002.

[47] Kui Zhang, Zhauhui S. Qin, Jun S. Liu, Ting Chen, Michael S. Waterman, and Fengzhu Sun. Haplotype block partitioning and tag snp selection using genotype data and their applications to association studies. *Genome Research*, (14):908–916, 2004.

[48] Q. Zhang, W. Wang, L. McMillan, F. Pardo-Manuel de Villena, and D. Threadgill. Inferring genome-wide mosaic structure. *PSB*, January 2009.

[49] Q. Zhang, W. Wang, L. McMillan, J. Prins, F. Pardo-Manuel de Villena, and D. Threadgill. Genotype sequence segmentation: Handling constraints and noise. *Proceedings of 8th Workshop on Algorithms in Bioinformatics*, 2008.

See http://compgen.unc.edu/papers/ACM_BCB_2010/compat for additional figures and appendices.