

# Algorithms for Association Study Design Using a Generalized Model of Haplotype Conservation

Russell Schwartz  
Department of Biological Sciences and  
School of Computer Science  
Carnegie Mellon University  
4400 Fifth Avenue  
Pittsburgh, PA 15213 USA  
russells@andrew.cmu.edu

## Abstract

*There is considerable interest in computational methods to assist in the use of genetic polymorphism data for locating disease-related genes. Haplotypes, contiguous sets of correlated variants, may provide a means of reducing the difficulty of the data analysis problems involved. The field to date has been dominated by methods based on the “haplotype block” hypothesis, which assumes discrete population-wide boundaries between conserved genetic segments, but there is strong reason to believe that haplotype blocks do not fully capture true haplotype conservation patterns. In this paper, we address the computational challenges of using a more flexible, block-free representation of haplotype structure called the “haplotype motif” model for downstream analysis problems. We develop algorithms for htSNP selection and missing data inference using this more generalized model of sequence conservation. Application to a dataset from the literature demonstrates the practical value of these block-free methods.*

## 1. Introduction

Recent advances in human genetics and high-throughput sequencing technologies have given new hope for uncovering genetic variations that underlie risk for common, complex diseases. Such disease-related variants may be found through case-control association studies, in which one compares genetic variations found in healthy (control) and diseased (case) individuals to pinpoint

those variants associated with the disease. Unfortunately, the prohibitive cost of typing dense marker sets in many people and the statistical challenges such large data sets would present have created a need for methods to reduce the amount of genotyping required and the size of the resulting statistical analysis problems. One leading strategy involves identifying conserved haplotype patterns in order to exploit correlations between distinct variant sites (loci) within the genome. Haplotypes can, for example, be used to identify small sets of single nucleotide polymorphisms (SNPs) that collectively characterize all or most of the variation in a region [5]. These small sets, known as “haplotype tagging SNPs” (htSNPs), could then be sequenced in place of the full set of variants in future studies. Developing methods to exploit haplotype patterns in such ways poses two key challenges. How can we best represent the patterns of correlation in sets of haploid sequences? And how can we best solve key analysis problems, such as htSNP selection and missing data reconstruction, using any given model of this haplotype structure?

An influential study by Daly et al. [3] suggested that conserved haplotype patterns take the form of “haplotype blocks,” discrete regions of low diversity whose boundaries are conserved across distinct haplotypes. Significant theoretical [7, 14] and empirical [14, 12, 11, 9] evidence has since emerged that considerable conserved substructure is lost when the data is fit to a block structure. Nonetheless, one strong argument for assuming block structure has been algorithmic convenience. We can efficiently locate optimal block partitions and htSNP sets for a broad range of block definitions [15] and apply them trivially to missing data inference and to association-based sta-

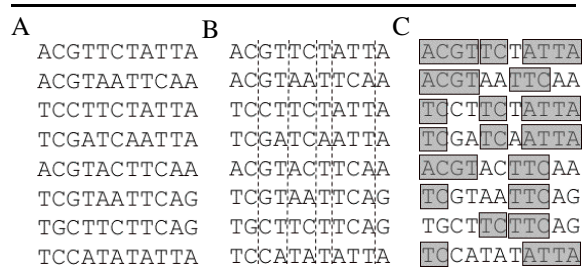
tistical testing.

Block-free models of haplotype structure have to date proven less amenable to downstream analysis. Approaches have been developed to infer haplotype structure without a prior block assumption [13, 10] and to solve related problems, such as locating frequent sub-sequences [16]. Only heuristic methods have been developed, though, for htSNP selection and missing data inference based on such block-free models [11]. We can solve such problems without an explicit haplotype structure model [1], which may have advantages in avoiding preconceptions about a subject that is only imperfectly understood. We nonetheless argue here for the value of developing methods using an explicit model of conserved haplotype structure, primarily because of the independent value of the identified structure to other problems, especially association testing. Furthermore, the primary hope for future advances on these problems would appear to be improving our understanding of the nature of haplotype conservation; methods incorporating a model of haplotype structure will be better able to leverage future discoveries in this area.

The purpose of this paper is to demonstrate that important downstream analysis problems can be solved with a block-free model of haplotype structure. It provides the first tractable method for optimal htSNP selection from identified block-free haplotype structure. It further presents two related algorithms for motif-based missing data inference. All of these methods are based on a recent technique for identifying block-free haplotype structure, the “haplotype motif” model [9], which decomposes sequences into disjoint segments that are statistically overrepresented in a population (haplotype motifs) and isolated polymorphic alleles. A motif decomposition, like a block decomposition, is a hypothesis about which segments of chromosome in a population were inherited intact from common ancestors. Figure 1 compares haplotype motifs to haplotype blocks. Application to a dataset from the literature demonstrates that the methods presented here provide high accuracy in SNP prediction for both rare and common variants and yield improved performance over a leading reference block method.

## 2. Methods

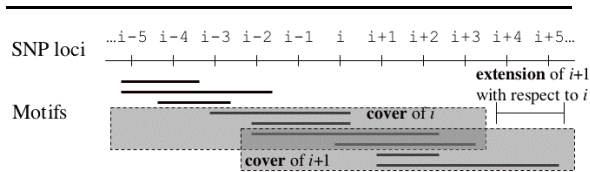
In this section, we provide detailed descriptions of algorithms for missing data inference and htSNP selection. These methods all make use of the “haplotype motif” model [9] and the associated methods for



**Figure 1. Illustration of haplotype block and haplotype motif models. This figure is not based on any specific block method or actual run of the motif algorithm. A: Eleven SNPs in eight individuals; B: block decomposition with vertical dashed lines showing block boundaries; C: motif decomposition with boxed subsequences in greyscale showing identified over-represented motifs.**

finding motifs, estimating their population frequencies, and interpreting sequences in terms of them. The motif model assumes sequences are generated from concatenations of conserved motifs or isolated bases. The methods all assume that we have defined a set of motifs  $M = \{m_1, \dots, m_t\}$  for the population under study. We formally define each motif as a tuple  $m_i = (s_i, p_i, f_i)$  where  $s_i$  is the sequence of SNP alleles of motif  $i$ ,  $p_i$  is the position in the full sequence of its starting site, and  $f_i$  is its frequency in the parsed training data. We also define  $q_i \equiv p_i + |s_i| - 1$ , the position in the full sequence of the motif’s ending site. We now define some additional terminology that will help in formally describing the methods:

- A *resolution* of a region of SNPs  $[i, j]$  is a tuple  $(i, j, R)$  where  $R \subseteq \{i, \dots, j\}$ . A resolution represents a possible solution to the htSNP problem in a local region.
- Two resolutions,  $\rho_1$  and  $\rho_2$ , are *consistent* (abbreviated  $\rho_1 \star \rho_2$ ) if they contain the same SNP set over the intersection of the regions they cover. That is, given  $\rho_1 = (i_1, j_1, R_1), R_1 = r_{11}, \dots, r_{1k_1}$  and  $\rho_2 = (i_2, j_2, R_2), R_2 = r_{21}, \dots, r_{2k_2}$  then  $\rho_1$  and  $\rho_2$  are consistent if and only if for all  $k$  such that  $\max\{r_{11}, r_{21}\} \leq k \leq \min\{r_{1k_1}, r_{2k_2}\}, k \in R_1 \Leftrightarrow k \in R_2$ .
- A motif  $m_i$  *touches* a site  $j$  if  $p_i \leq j \leq q_i$ .
- The *cover* of a SNP site  $i$ , called  $C_i$ , is the set



**Figure 2. Illustration of cover and extension.** The figure shows a hypothetical genomic axis (top) with positions of motifs along it shown by horizontal lines (bottom). Overlapping grey boxes show the covers of SNP sites  $i$  and  $i + 1$ . The extension of  $i + 1$  with respect to  $i$  is marked.

of all motifs that touch that SNP site. We define  $a_i$  and  $b_i$  to be the sites of lowest and highest index, respectively, touched by any motif in  $C_i$ .

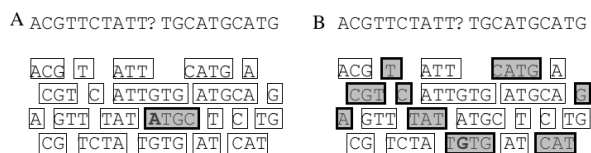
- The *extension* of cover  $C_i$  with respect to cover  $C_{i-1}$  is the set of SNP sites touched by some motif in  $C_i$  but no motif in  $C_{i-1}$ .
- The *span* of a set of motifs  $M$ , denoted  $\Sigma(M)$ , is the union of SNP sites touched by all motifs in the set.

Figure 2 illustrates *covers* and *extensions*.

## 2.1. Local Algorithm for Missing Data Inference

We first describe a method for inferring missing values in haploid sequences. We can formalize missing data inference by assuming we are given as input a sequence with some missing site to infer and a set of motifs on which to base the inference. In this first method, we will find the one best motif covering the missing site according to a probability model and infer the missing value from the corresponding value in that best-fit motif. This concept is illustrated in figure 3A.

The method is based on optimizing for a simple probability model also used in the motif discovery process for determining the probability of a given motif  $m_i$  matching to a known sequence  $S$ . Assume we are given a motif  $m_i = (s_i, p_i, f_i)$ , where  $s_i = s_{i,1}, \dots, s_{i,k_i}$ ; a prior mismatch probability  $\mu$ , and a target sequence  $S = S_1, \dots, S_n$ , in which a missing site  $j$  is denoted by a special symbol  $S_j = '?'$ . We wish to establish a probability that sequence  $S$  was generated from motif  $m_i$ . We formally model the probability of observing the known sites of  $S$  in the



**Figure 3. Illustration of the missing data inference methods.** Each assumes we have a sequence (above) with a missing site to be inferred (the '?') and a pool of motifs (below) that might match different parts of a sequence. **A:** In the local method, the single strongest motif match covering the missing site provides the value inferred for that site; **B:** In the global method, the optimal tiling of the full sequence in terms of motifs is constructed and the missing value filled in from the corresponding position of the tiling.

span of  $m_i$  given that the sequence is generated from  $m_i$  as follows:

$$P(S|m_i) = \prod_{j=1}^{k_i} M(s_{i,j}, S_{j+p_i-1}), \text{ where}$$

$$M(a, b) = \begin{cases} 1 & b = '?' \\ (1 - \mu) & a = b \wedge b \neq '?' \\ \mu & a \neq b \wedge b \neq '?' \end{cases}$$

We further define the prior probability of  $m_i$ ,  $P(m_i)$ , to be its population frequency  $f_i$ .

Given these definitions, the inference algorithm is straightforward. For each motif  $m_i$  covering missing site  $j$  in  $S$ , we compute  $P(S|m_i)P(m_i)$ . We select the motif  $m_{i^*}$  maximizing  $P(S|m_{i^*})P(m_{i^*})$  and set the missing value  $S_j$  to be  $s_{i^*,j-p_{i^*}+1}$ ,  $m_{i^*}$ 's value at that site. This algorithm can also be used to correct for likely sequencing errors or to screen out likely recent mutations by allowing inference of both observed and missing sites.

The run time of this algorithm depends on the maximum motif length  $n_{max}$ , the sequence length  $n$ , and the number of motifs per site, which is a function of  $n_{max}$  and the number of sequences  $m$  in the reference population. Testing each motif for compatibility requires  $O(n_{max})$  time per site, for  $n$  sites and at most  $n_{max}m$  motifs per site, yielding a bound of  $O(nn_{max}^2m)$  beyond the time required to determine motifs.

## 2.2. Global Algorithm for Missing Data Inference

We can construct an alternative missing data inference algorithm by finding an optimal global motif explanation for the target sequence and filling in each missing site based on the motif touching it in that explanation. In other words, we tile the target sequence with motifs in the most probable way and fill in a missing site based on the motif of the tiling that covers that missing site. This global strategy is illustrated in Figure 3B. We first extend the probability model from the prior section to cover multiple motifs explaining different parts of a sequence. We define the probability of a sequence given two or more non-overlapping motifs to be the product of the probabilities of the sequence given the motifs individually, thereby assuming that non-overlapping motifs are sampled independently from one another. More formally, the probability of a given explanation, or parse, of a sequence  $S$  for a sequence of contiguous motifs  $m_{i_1}, \dots, m_{i_k}$  is defined as

$$P(S|m_{i_1}, \dots, m_{i_k}) \equiv \prod_{l=1}^k \prod_{j=1}^{k_{i_l}} M(s_{i_l, j}, S_{j+p_{i_l}-1})$$

with all parameters defined as before. The probability is defined to be zero for non-contiguous motif sets (i.e. the motif  $i+1$  does not start at the next polymorphic site after the end of motif  $i$  for some  $i$ ).  $P(S|\emptyset)$  is defined to be 1. The prior probability of the parse,  $P(m_{i_1}, \dots, m_{i_k})$ , is defined, on the assumption that non-overlapping motifs appear independently, to be  $\prod_{l=1}^k f_{i_l}$ .

Our main algorithmic problem is to find the motif set  $\{m_{i_1}, \dots, m_{i_k}\} \subseteq M$  covering the full sequence that maximizes  $P(S|m_1, \dots, m_t)P(m_1, \dots, m_t)$ . Let  $M_k \equiv \{m_i = (s_i, p_i, f_i) \mid p_i + |s_i| - 1 = k\}$ , the set of motifs ending at site  $k$ . Then define  $MAX(k)$  to be the probability of the optimal explanation of the  $k^{th}$  prefix of  $S$ ,  $(S_1, \dots, S_k)$ . We can find  $MAX(k)$  via the recurrence

$$MAX(k) = \max_{m_i \in M_k} \{P(S|MAX(p_i - 1)) \times$$

$$P(MAX(p_i - 1)) \times P(S|m_i) \times P(m_i)\}$$

with the base case  $MAX(0) = 1$  by dynamic programming. Given the optimal global motif parse corresponding to  $MAX(n)$ , we can fill in each missing site based on the assigned motif touching it. Like the local algorithm, this method can be used to correct for sequencing errors or recent mutations by also altering observed sites that disagree with their assigned motifs.

The run time is dominated by the dynamic programming parsing, which requires worst case time  $O(nn_{max}^2 m)$  where  $m$  is the number of sequences in the reference population from which motifs were defined, as it must examine up to  $m$  motifs for each of  $n_{max}$  possible lengths for each of  $n$  prefixes.

## 2.3. Finding htSNP Sets

**2.3.1. Optimization Metric** An optimization metric for motif-based htSNP selection is usually defined, perhaps implicitly, with respect to some method for using the htSNPs to reconstruct the information left out of the htSNP set. Our metric is informally defined so as to choose htSNPs optimally to facilitate reconstruction of the full sequence by the local inference method with the additional assumption of independence between non-overlapping motifs.

To formalize the metric, we first define the “expected error” of a SNP set  $R$  at a given hidden site  $j$ .  $ExpectedError(R, j)$  estimates the probability over the distribution of sequences generated by the motif model that the local inference algorithm would incorrectly infer the value of site  $j$  by observing SNP set  $R$ . To calculate  $ExpectedError(R, j)$  we must find for each motif  $m_i$  touching  $j$  the motif  $m_k$  we would expect to be inferred to have generated  $m_i$ 's sequence. We first replace all sites not in  $R$  in all motifs with the special symbol ‘?’’. Let  $p = \max\{p_i, p_k\}$  and  $q = \min\{q_i, q_k\}$  and define  $P(m_i|m_k) \equiv \prod_{l=p}^q M(s_{i, l-p_k+1}, s_{i, l-p_i+1})$ . We then find the  $\hat{m}_i$  maximizing  $P(\hat{m}_i)P(m_i|\hat{m}_i)$ . (Note  $\hat{m}_i$  may be  $m_i$ .) We then define a function on motifs,  $e_j$ , where  $e_j(m_i) = f_i$  if  $m_i$  and  $\hat{m}_i$  differ at site  $j$  and  $e_j(m_i) = 0$  otherwise. Then  $ExpectedError(\rho, j) = \sum_{m_i \in C_j} e_j(m_i)$ . That is,  $ExpectedError(\rho, j)$  is the sum of the frequencies of all motifs touching  $j$  from which we would expect the local algorithm to infer the wrong allele at  $j$  by examining only sites in  $\rho$ .

While we can formulate various objectives based on  $ExpectedError$ , we focus here on finding the SNP set of a given size minimizing the total expected number of errors over all sites under the above probability model. More formally, we assume we are given a motif set  $M$  and a target number of htSNPs  $k_{max}$  and seek the resolution  $\rho = (1, n, R)$  where  $|R| = k_{max}$  minimizing  $\sum_{j=1}^n ExpectedError(\rho, j)$ . We call this the MINERR problem. It is a block-free analog to the partition with fixed number of tagging SNPs (FTS) problem of Zhang et al. [16, 17].

**2.3.2. Algorithm** The principle behind the algorithm is to exhaustively enumerate possible solutions in a local region around each SNP locus and use dynamic programming to build a global optimum out of these local solutions. The algorithm is thus similar in structure to that of the bounded-width algorithm developed in Bafna et al. [1], although using a two-dimensional dynamic programming method like that in Zhang et al. [16, 17].

Define  $MinError(j, R_j, k)$  to be the minimum of  $\sum_{i=1}^j ExpectedError(R_i, i)$  over SNP sets  $R_1, \dots, R_{j-1}$  for which  $|\bigcup_{i=1}^j R_i| = k$ .  $MinError$  is thus the cost of a partial solution to the MINERR problem for a prefix of the SNP loci and for a particular number of SNPs  $k$  and suffix  $R_j$ . The global solution for a specified target number of SNPs,  $k_{max}$ , will be that corresponding to  $\min_{R_n} \{MinError(n, R_n, k_{max})\}$ . Further define  $Extension(R, j)$ , the “extension cost” of  $R$ , to be the number of SNPs in the intersection of  $R$  and the extension of  $C_j$  with respect to  $C_{j-1}$ . We can find  $MinError(j, R_j, k)$  by dynamic programming with the recurrence

$$MinError(j, R_j, k) = \min_{R_{j-1} \in \Sigma(C_{j-1})^*, \rho_{j-1} * \rho_j} \{$$

$$MinError(j-1, R_{j-1}, k - Extension(R_j, j)) + ExpectedError(R_j, j)\}$$

where  $\rho_j = (a_j, b_j, R_j) \forall j$ .

Run time is dominated by the cost of calculating  $ExpectedError(R, j)$ , which requires time  $O(mn_{max}^2)$  for each of  $O(2^{2n_{max}-1})$  resolutions at each of  $n$  sites and  $O(k_{max})$  SNP counts, giving a total bound of  $O(mnn_{max}^2 k_{max} 2^{2n_{max}-1})$ . Although run-time is exponential in maximum motif length, we can generally treat that as a constant, analogous to the treatment of maximum block length [15] and maximum width of SNP association [1] as constants in the prior literature. We do find some long conserved segments in practice and must artificially limit the maximum motif length. Breaking up very long motifs like this seems to have little practical effect on the methods’ accuracies. Memory usage is dominated by the  $O(nk_{max} 2^{2n_{max}-1})$  cost of storing scores for all possible resolutions and SNP counts at each site. To reduce memory usage in practice, we implemented a heuristic garbage-collection method that dynamically prunes dead-end paths in the dynamic programming matrix.

**2.3.3. Alternative Metrics** We also briefly discuss two variants on the MINERR method. One, called

MINSNP, takes as input the motif set  $M$  and a target error bound  $\epsilon$  and returns a SNP set  $R$  of minimum size such that  $ExpectedError(R, j) < \epsilon$  for all  $j$ . MINSNP is a block-free analog to the common block-based problem of choosing a minimal SNP set characterizing the haplotype in each block of an observed sequence with at least some minimum probability. MINSNP can be solved by a simpler version of the MINERR algorithm that iterates over  $j$  from 1 to  $n$  finding the minimum size SNP set guaranteeing  $ExpectedError(R, i) < \epsilon \forall i \leq j$  on the  $j^{th}$  prefix of the SNPs for each possible local context of site  $j$ . This algorithm has runtime and memory usage a factor of  $k_{max}$  lower than does the MINERR algorithm, often making MINSNP the most practical variant of the method. We have focused on MINERR, though, because it better lends itself to a thorough empirical validation. We might also seek the SNP set of minimum size to achieve a given average per-site expected error, a variant we call MINAVG. We can solve MINAVG given the MINERR algorithm through binary search of sizes of SNP set, which adds an additional factor of  $O(\log n)$  to the run time. Although we go into depth only on the MINERR algorithm in this paper, all three have been implemented in the Hapmotif package that has been made available for download.

## 2.4. Block-based analysis

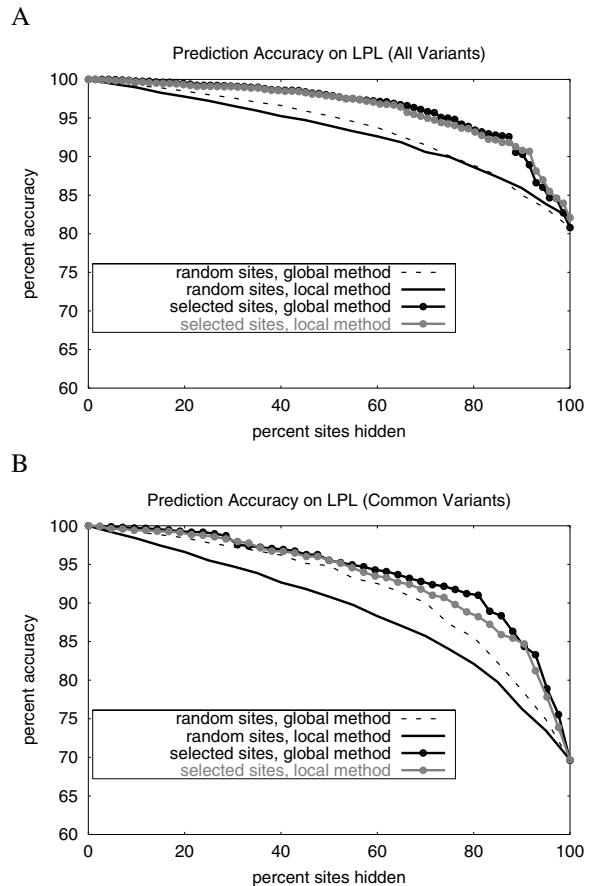
In order to provide a baseline for evaluating the proposed methods, we also implemented a variant of a leading block-based method from the literature. It would be both impractical and beyond the scope of this paper to compare to all block-based methods in the literature. We therefore used one representative block definition, that of Patil et al. [8], because of its prior use in seminal work on haplotype block algorithms [15, 16, 17]. This method defines a block as a region in which a fraction  $\alpha$  of the sequences can be characterized by “common” haplotypes within that block region. The prior studies, which specifically used Patil et al.’s twenty-chromosome sample of chromosome 21, defined a common haplotype as one occurring at least twice. We generalize that definition to larger datasets by defining a common haplotype as one occurring in at least 10% of the sequences in the data set. We apply a variant of Zhang et al.’s [15] dynamic programming algorithm to find a decomposition minimizing the number of SNPs needed to characterize unambiguously a fraction  $\alpha$  of the population. We infer missing sites by assigning to each sequence the haplotype in each block region that matches the most observed sites, using pop-

ulation frequency to break ties, and filling in missing data from these assigned haplotypes.

### 3. Results

We base our validation on a data set of computationally inferred, PCR validated haplotypes from 71 SNPs in 142 chromosomes for the lipoprotein lipase (LPL) gene [6, 2]. This data set was chosen because it has experimentally verified haplotypes, a sufficient number of SNP sites to allow for a meaningful test of prediction accuracy, and a sufficient number of chromosomes to allow for adequate cross-validation. Furthermore, unlike most available haplotype data sets, it includes all polymorphisms detected in the population sample rather than only common ones, allowing us to test whether common variants can be used to predict rarer variants accurately. This question is of great practical importance because the genetic variants causal of disease that we wish to find via inference studies are likely to be rare, while the SNPs being gathered for population-wide screens are almost always only those common in the population. We created two versions of the data set, one containing all variants and one containing only those with minor allele frequency at least 10%. For purposes of cross-validation, we randomly split each data set into two equal-sized subsets, with one subset used for defining motifs and the other used for validating the results of our missing site inference and htSNP selection methods.

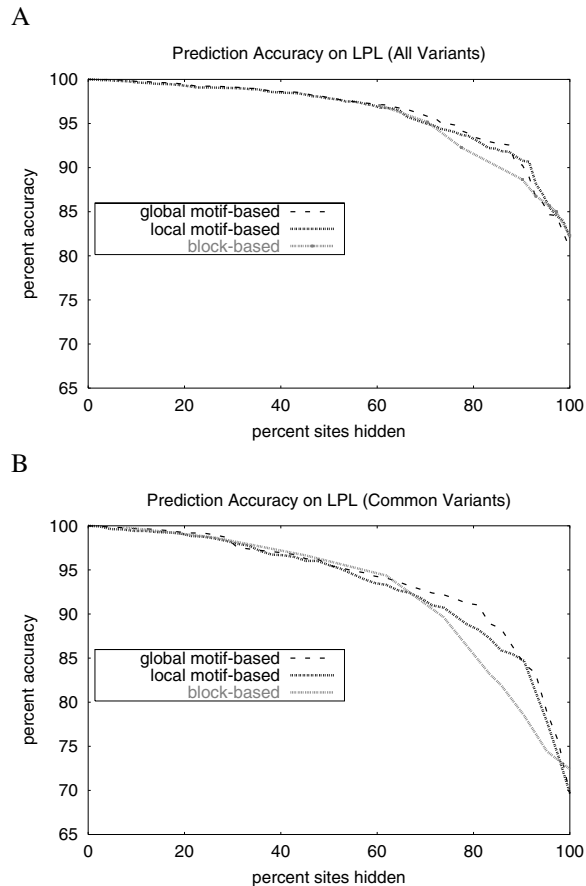
We defined haplotype motifs using p-value 0.025, prior mismatch probability zero, and maximum motif length 10. We allowed a prior mismatch probability of 0.1% during missing data inference to account for variants missing from the training data. We found optimal MINERR solutions for each possible number of SNPs (0 to 71 for the full data sets, 0 to 42 for common variants only). For each solution, we tested our ability to predict missing sites in the testing data by both local and global inference algorithms, recording the percent accuracy on hidden sites and the fraction of sites hidden. For tests on randomly chosen SNP sets, we used ten copies of the testing data with randomly hidden sites for each fraction hidden from 0% to 100% in increments of 5%. To apply the block method, we varied the value of  $\alpha$  between 0.5 and 1.0 in increments of 0.05, calculating block decompositions and minimal SNP sets on the training data for each  $\alpha$ , and tested prediction accuracy on the testing data by the block-based missing site prediction algorithm, recording the percent accuracy and percent of sites hidden for each  $\alpha$ .



**Figure 4. Comparison of local and global inference using randomly hidden SNPs and htSNPs. A: all variants; B: common variants**

Figure 4 compares prediction accuracy as a function of fraction of sites hidden for block-free MINERR htSNPs and randomly hidden SNPs. For both the full data (Fig. 4A) and the common variants only (Fig. 4B), accuracy using htSNPs is noticeably higher than that using randomly chosen SNPs, although the difference is more pronounced when all variants are present. While the global method is clearly superior with random SNPs, the results are more ambiguous for htSNPs. The two prediction methods yield very similar results when less than 50% of sites are hidden, the global method yields superior results when about 50% to 90% of sites are hidden, and the local method yields generally superior results when more than 90% of sites are hidden.

Figure 5 compares motif-based and block-based htSNP selection methods. All three methods (lo-



**Figure 5. Comparison of prediction accuracy from htSNPs with motif and block-based methods. A: all variants; B: common variants**

cal motif-based, global motif-based, and block-based) produce nearly identical results until about 50% of sites are hidden, at which point all begin to drop noticeably in accuracy. The block-based method, though, drops in accuracy significantly faster than the motif-based methods. This effect is more pronounced for common variants than it is for the full data set. Note that even small differences in the curves can correspond to large practical differences in the value of the methods for association study design. The motif method often needs fewer than half as many htSNPs as the block method to achieve a given level of prediction accuracy.

## 4. Discussion

We have demonstrated computational methods for missing data inference and htSNP selection using generalized patterns of haplotype sequence conservation that we call “haplotype motifs.” While dropping the assumption of shared block boundaries between conserved segments can be expected to yield better matches to true genetic conservation patterns, it also leads to more complicated optimization problems. We nonetheless show that under reasonable probability models for generalized haplotype structure, key optimization problems in the design of haplotype-based association studies can be solved efficiently. Furthermore, the resulting block-free methods appear to perform very well in practice on real haplotype data. The motif method yields prediction accuracy as functions of htSNP set size nearly identical to that of a leading block method when more than half of sites are used and can yield reductions in minimum htSNP set sizes of more than 50% over much of the useful range of prediction accuracy.

It is an open question whether a block-free or block-based method will ultimately prove more useful for htSNP selection. The motif method is in principle limited to shorter total lengths of individual conserved segments than the block method because of the former’s exponential dependence on twice the maximum motif length. Thus, we trade off greater flexibility in modeling short-range correlations for the loss of occasional long-range correlation information. We argue that this is a sound trade-off, as work with haplotype motif methods and large-scale block studies [8, 4] both suggest that the great majority of conserved regions are only a few SNPs long. Definitely answering this question will, however, require more extensive empirical studies.

This work could be advanced in many ways. Our current work uses haplotype data as input, rather than the more easily available unphased genotype data. We could use a separate phasing program to convert genotype data to predicted haplotype data as input to our program. The motif model, though, has obvious applications to haplotype phasing, which could be merged with the motif discovery and parsing procedure. The algorithms in the present work are not tied to our specific construction of haplotype motifs and might yield better practical performance for more sophisticated generalized models of haplotype structure. The work might also benefit from a more sophisticated model of mismatches between observed sequences and predicted motif structures. Finally, we must recognize that the value of our methods and

all others for these problems will depend on properties of the specific genetic regions examined, such as their diversity in the population, their historical recombination rates, and the degrees to which they exhibit evidence of population substructure. As the data to examine these issues becomes available, empirical studies will prove extremely valuable in evaluating the prospects of these computational approaches for helping us achieve our true goal: finding the genetic bases common of human diseases.

**Code availability:** All of the algorithms described in this paper have been implemented and are available for download from <http://www-2.cs.cmu.edu/~russells/software/hapmotif.html>.

**Acknowledgments:** I thank Bjarni Halldórsson for helpful discussions. I am also grateful to anonymous referees for their comments on earlier drafts of this paper. This work was supported in part by a grant from the Samuel and Emma Winters Foundation.

## References

- [1] V. Bafna, B. Halldórsson, R. Schwartz, A. G. Clark, and S. Istrail. Haplotypes and informative SNP selection algorithms. In *Proc. 7th Intl. Conf. on Computational Molecular Biology (RECOMB-02)*, pages 19–27, 2003.
- [2] A. G. Clark, K. M. Weiss, D. A. Nickerson, S. L. Taylor, A. Buchanan, J. Stengard, V. Salomaa, E. Vartiainen, M. Perola, E. Boerwinkle, , and C. F. Sing. Haplotype structure and population genetic inference from nucleotide-sequence variation in human lipoprotein lipase. *Am J Hum Genet*, 63:595–612, 1998.
- [3] M. J. Daly, J. D. Rioux, S. F. Schaffner, and T. J. Hudson. High-resolution haplotype structure in the human genome. *Nat Genet*, 29:229–232, 2001.
- [4] 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.
- [5] G. C. Johnson, L. Esposito, B. J. Barret, A. N. Smith, J. Heward, G. Di Genova, H. Ueda, H. J. Cordell, I. A. Eaves, F. Dudbridge, R. C. Twells, F. Payne, W. Hughes, S. Nutland, H. Stevens, P. Carr, E. Tuomilehto-Wolf, J. Tuomilehto, S. C. Gough, D. G. Clayton, and J. A. Todd. Haplotype tagging for the identification of common disease genes. *Nat Genet*, 29:233–237, 2001.
- [6] D. A. Nickerson, S. L. Taylor, K. M. Weiss, A. G. Clark, R. G. Hutchinson, J. H. Stengard, V. Salomaa, E. Vartiainen, E. Boerwinkle, and C. F. Sing. DNA sequence diversity in a 9.7-kb region of the human lipoprotein lipase gene. *Nat Genet*, 19:233–240, 1998.
- [7] M. Nordborg and S. Tavare. Linkage disequilibrium: what history has to tell us. *Trends Genet*, 18:83–90, 2002.
- [8] 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. Nguyen, M. C. Norris, J. B. Sheehan, N. Shen, D. Stern, R. P. Stokowski, D. J. Thomas, M. O. Trulson, K. R. Vyas, K. A. Frazer, S. P. Fodor, and D. R. Cox. Blocks of limited haplotype diversity revealed by high resolution scanning of human chromosome 21. *Science*, 294:1719–1722, 2001.
- [9] R. Schwartz. Haplotype motifs: an algorithmic approach to finding evolutionarily conserved patterns in haploid sequences. In *Proc. 2nd IEEE Computational Systems Biology Bioinformatics Conference (CSB2003)*, pages 306–314, 2003.
- [10] R. Schwartz, A. G. Clark, and S. Istrail. Methods for inferring block-wise ancestral history from haploid sequences: The haplotype coloring problem. *Lecture Notes in Computer Science*, 2452:44–59, 2002. (Proc. 2nd Intl. Workshop on Algorithms in Bioinformatics (WABI2002)).
- [11] R. Schwartz, A. G. Clark, and S. Istrail. Inferring piecewise ancestral history from haploid sequences. *Lecture Notes in Bioinformatics*, 2983:62–73, 2004.
- [12] R. Schwartz, B. Halldórsson, V. Bafna, A. G. Clark, and S. Istrail. Robustness of inference of haplotype block structure. *J Comp Biol*, 10:13–21, 2003.
- [13] E. Ukkonen. Finding founder sequences from a set of recombinants. *Lecture Notes in Computer Science*, 2452, 2002. (Proc. 2nd Intl. Workshop on Algorithms in Bioinformatics (WABI2002)).
- [14] J. D. Wall and J. K. Pritchard. Assessing the performance of the haplotype block model of linkage disequilibrium. *Am J Hum Genet*, 73:502–515, 2003.
- [15] K. Zhang, M. Deng, T. Chen, M. S. Waterman, and F. Sun. A dynamic programming algorithm for haplotype block partitioning. *Proc Natl Acad Sci USA*, 99:7335–7339, 2002.
- [16] K. Zhang, F. Sun, M. S. Waterman, and T. Chen. Dynamic programming algorithms for haplotype block partitioning: applications to human chromosome 21 haplotype data. In *Proc. 8th Annual International Conference on Computational Molecular Biology (RECOMB03)*, pages 332–340, 2003.
- [17] K. Zhang, F. Sun, M. S. Waterman, and T. Chen. Haplotype block partition with limited resources and applications to human chromosome 21 haplotype data. *Am J Hum Genet*, 73:61–73, 2003.