A NEW MODEL OF MULTI-MARKER CORRELATION FOR GENOME-WIDE TAG SNP SELECTION

WEI-BUNG WANG TAO JIANG
weiw@cs.ucr.edu jiang@cs.ucr.edu

Department of Computer Science, University of California - Riverside

Tag SNP selection is an important problem in computational biology and genetics because a small set of tag SNP markers may help reduce the cost of genotyping and thus genome-wide association studies. Several methods for selecting a smallest possible set of tag SNPs based on different formulations of tag SNP selection (block-based or genome-wide) and mathematical models of marker correlation have been investigated in the literature. In this paper, we propose a new model of multi-marker correlation for genome-wide tag SNP selection, and a simple greedy algorithm to select a smallest possible set of tag SNPs according to the model. Our experimental results on several real datasets from the HapMap project demonstrate that the new model yields more succinct tag SNP sets than the previous methods.

1. Introduction

Single nucleotide polymorphisms (SNPs) represent the most frequent form of genetic variations in the human genome. They play an important role in genome-wide association studies that intend to help us understand the correlation between genetic variations and human diseases. Assaying (or genotyping) all SNP markers in the involved genomes would be desirable, but it is expensive and unnecessary. Since SNPs are often not independent, a subset of SNPs may be sufficiently informative and allow us to infer all the other SNPs. The *tag SNP selection* problem is thus to find a smallest possible set of tag SNPs that would enable us to infer all the other SNPs with a certain level of confidence [9]. Clearly, the smaller the tag SNP set, the more genotyping cost it could help save.

Two frameworks for tag SNP selection have been studied in the literature: block-based and genome-wide. The block-based tag SNP selection framework focuses on *haplotype* patterns in a population.[a] The approach assumes that the chromosomes can be partitioned into blocks separated by recombination hotspots, so that there are few recombinations within a block. Then it attempts to identify a smallest possible set of tag SNPs for each block so that all the possible halpotype patterns formed by the SNPs in the block can be fully represented by the haplotype patterns formed by the tag SNPs [14]. The genome-wide framework does not partition a

[a] Recall that humans are diploids and our chromosomes form pairs, each of which consists of a paternal chromosome and a maternal chromosome. A haplotype refers to the set of SNPs from a single chromosome. A pair of corresponding paternal and maternal haplotypes form a genotype.

chromosome into blocks. Instead, it considers the correlation between SNP markers across the entire genome [1]. Typically, a SNP marker has two states in a population. The state with a higher frequency is called the *major allele* and the other is called the *minor allele*. In the other words, the SNP markers are usually *bi-allelic*. It is a common practice to consider only SNPs whose *minor allele frequency* (MAF) is at least 5%. Genome-wide tag SNP methods generally follow two approaches. Halldórsson *et al.* [3] define "informativeness" of SNPs and attempt to find the most informative set of SNPs. The other approach, such as the one adopted by Carlson *et al.* [1], usually evaluates the *linkage disequilibrium* (LD) between the states of two SNP markers using the correlation coefficient r^2, which indicates the dependency between the two markers, and aims at finding a smallest set of tag SNPs such that all the other SNPs are strongly linked to the selected tag SNPs in terms of the LD coefficient r^2 (more precisely, each of them is linked to some tag SNPs with an r^2 coefficient above a certain threshold). The tag SNPs selected by this approach are shown to be effective in disease association mapping studies, since the coefficient r^2 is directly related to the statistical power of association mapping. Genome-wide tag SNP selection based on the r^2 LD statistics has gained popularity among researchers in the SNP community [1, 2, 8, 12, 15, 18], because it has a comparable performance at a lower computational cost than many other methods [17, 18]. In this paper, we will be focused on genome-wide tag SNP selection using the r^2 LD statistics.

Most of the existing tag SNP selection methods in this framework consider the r^2 coefficient between a pair of SNP markers [1, 11, 12, 15]. Hence, each of the SNPs is guaranteed to be tagged by a single tag SNP selected. Recently, Hao *et al.* [4, 5] extended the r^2 statistics to describe the statistical correlation between a group of (*e.g.* two or three) markers and another marker. We will simply refer to this as the *multi-marker correlation model*. In this model, a SNP is tagged by a group of tag SNPs if it is correlated to the group with an r^2 coefficient above a certain threshold. Hao *et al.* [4, 5] presented a greedy algorithm for selecting tag SNPs to cover a certain (large) fraction of a given set of SNPs and showed that the multi-marker correlation model is more effective than the traditional pairwise correlation model in terms of reducing the number of required tag SNPs.

In this paper, we generalize the multi-marker correlation model in [4, 5] to further improve its effectiveness. Comparing with the model in [4, 5], our model is more natural and supports more succinct tag SNP sets. We will also present a simple greedy algorithm to select a smallest possible set of tag SNPs according to this multi-marker model, and compare its performance with those of the previous methods on real HapMap data.

Genome-wide tag SNP selection methods can also be classified as *haplotype-based* or *haplotype-independent*, depending on how the r^2 statistics is obtained. For genotype data, the r^2 statistics is usually estimated using a maximum likelihood approach [6, 10], which could be time consuming on a large set of SNPs. However, when phased haplotypes are available, the r^2 coefficients can be calculated very easily and efficiently. The haplotype-based methods require phased haplotype data while the haplotype-independent methods do not. In this work, we will consider both types of data.

The rest of the paper is organized as follows. In Section 2, we introduce a new multi-marker correlation model and discuss how to calculate the r^2 LD statistics under the model for both haplotype and genotype data. Section 3 presents the simple greedy algorithm for selecting tag SNPs. In Section 4, we discuss the implementation of the algorithm and test its performance on some real HapMap datasets. We also compare the performance of our algorithm with those of the most recent algorithms on genome-wide tag SNP selection given in [4, 5, 11]. Section 5 concludes the paper with a few remarks. For the ease of reading, we defer some illustrative figures and a detailed mathematical proof required in the calculation of the multi-marker correlation coefficient r^2 to Appendix A.

2. The New Multi-Marker Correlation Model

In this section, we propose a new multi-marker correlation model that generalizes the model introduced in [4, 5]. We also discuss how to calculate the r^2 statistics under the new model for both haplotype and genotype data.

2.1. *Multi-Marker Correlation on Haplotype Data*

The statistical correlation between a group of k markers and another marker will be referred to as k-marker correlation. For simplicity, we define below the 2-marker correlation model. The generalization of the model to 3 or more markers is straightforward. Consider three bi-allelic SNPs A, B and C. Each of them has possible alleles A/a, B/b and C/c, respectively. Here, the uppercase letters represent both the SNPs as well their major alleles and the lowercase ones represent the minor alleles. Given the states (*i.e.* alleles) of SNPs A and B, it might be possible for us to infer the state of SNP C, if SNP C is correlated with both SNPs A and B. Clearly, if $Pr(C \mid AB) > 0.5$, we would opt to predict the major allele C instead of the minor allele c when the haplotype AB is observed.

For a fixed population of haplotype data and any haplotype h, let n_h denote the number of times that the haplotype h is observed in the population. Consider three SNPs A, B and C again. For each haplotype $h \in \{AB, Ab, aB, ab\}$, if $n_{hC} > n_{hc}$, then we would opt to predict allele C when observing haplotype h (assuming that the SNP C is unassayed). We put all the haplotypes $h \in \{AB, Ab, aB, ab\}$ such that $n_{hC} > n_{hc}$ into a *major bucket* and the others into a *minor bucket.* For example, if $n_{ABC} > n_{ABc}, n_{abC} > n_{abc}$ and $n_{AbC} < n_{AbC}, n_{aBC} < n_{aBc}$, then the major bucket will contain haplotypes $\{AB, ab\}$ while the minor bucket contains haplotypes $\{Ab, aB\}$. This would suggest a prediction of the allele C when any of the haplotypes $\{AB, ab\}$ in the major bucket is observed.

To define the r^2 correlation coefficient, we introduce a new bi-allelic (compound) marker M that combines the SNPs A and B. The major and minor alleles of M are M/m. We say that the marker M is in state (allele) M if any of the haplotypes in the major bucket is observed, or otherwise it is in state m. Hence, the numbers of observations of alleles M and m are defined as $n_M = n_{AB} + n_{ab}$ and $n_m = n_{Ab} + n_{aB}$. We can define the r^2 statistics between the two markers {A,B} and the marker C as the usual r^2 statistics between the new marker M and the marker C.

Occasionally, we may have a tie between haplotype counts in the population, such as $n_{hC} = n_{hc}$. In this case, we would have to decide whether to put the haplotype h in the major bucket or the minor bucket. The following claim shows that it is usually advantageous to put the haplotype in the minor bucket.

Claim 2.1. *Consider three SNP markers with alleles A/a, B/b, and C/c, and the correlation coefficient r^2 between the markers $\{A, B\}$ and the marker C. If h is an observed haplotype on the markers A and B, and the numbers of observations satisfy $n_{hC} = n_{hc}$, then putting h in the minor bucket leads to a higher r^2 value most of the time.*

Proof. See Appendix A. □

Since there are 4 possible haplotypes on markers A and B, there are $2^4 = 16$ ways to fill the major bucket. After eliminating symmetric ways and the empty set, there are $2^4/2 - 1 = 7$ different ways to separate the 4 possible haplotypes into two buckets. Note that, a split of the four haplotypes like $\{AB, Ab\}/\{aB, ab\}$ really represents the single-marker correlation between markers A and C. Therefore, the seven different separations correspond to two single-marker and five 2-marker correlations.

In [4, 5], Hao *et al.* proposed a very similar 2-marker correlation model to define the correlation between markers {A,B} and marker C. However, they require that one of the buckets must contain exactly one haplotype (unless the split actually represents a single-marker correlation). For example, a split like $\{AB\}/\{Ab, aB, ab\}$ would be allowed but the split $\{AB, ab\}/\{Ab, aB\}$ is not. Therefore, the 2-marker correlation model in [4, 5] allows a total of $2 + 4 = 6$ different splits, two of which correspond to single-marker correlations. Clearly, our new model is more flexible and gives us the opportunity to cover more SNPs with the same set of tag SNPs. Therefore, it may help reduce the number of tag SNPs required. This flexibility is even more obvious when we consider the correlation between a group of three markers and another marker. To infer a fourth SNP D from three SNPs A, B and C, our model allows $2^{2^3}/2 - 1 = 127$ possible splits of the 8 haplotypes on the SNPs A, B, and C into the major and minor buckets (modulo symmetry). However, because the model of Hao *et al.* in [4, 5] requires that one of the buckets must contain exactly one haplotype, it only allows $3 + 3 \cdot 4 + 8 = 23$ different splits, including 3 splits corresponding to single-marker correlations and another 12 corresponding to two-marker correlations.

2.2. *Calculating r^2 Values on Genotype Data*

Obtaining r^2 values from haplotype data is trivial. However, if the SNP data is in the form of unphased genotypes, we cannot obtain r^2 values directly since the above definition is based on haplotype data. There are two ways to deal with genotype data. One is to use some haplotype inference program such as PHASE [13, 16] to convert the genotype data into a haplotype data. The other way is to estimate k-marker haplotype frequencies directly from the population without phasing. The former method is trivial. So, here we discuss the latter method.

Hill [6] proposed in 1974 a maximum likelihood method to estimate the degree of LD between two loci (*i.e.* markers) given the frequencies of diploid genotypes in a random-mating population. Then he generalized the method to estimate haplotype frequencies at several loci in 1975 [7]. This method has been used to estimate LD r^2 statistics for more than 30 years. For example, it was used in [10] to estimate the LD among multi-allelic markers.

Hill's method works as follows. For simplicity, let us only consider estimating the frequency of 3-marker haplotypes. Consider a sample of population data from N random-mating individuals. Let n_g be the number of times that genotype g is observed in the sample. Denote as f_h the frequency of haplotype h. Let $\hat{f}_h$ be the maximum likelihood estimation of f_h. For three SNPs A, B and C, the frequency of haplotype ABC satisfies the following equation (due to Hardy-Weinberg equilibrium):

$$\begin{aligned}\hat{f}_{ABC} = \frac{1}{2N}\Bigg(&2n_{AABBCC} + n_{AABBCc} + n_{AABbCC} + n_{AaBBCC}\\ &+n_{AABbCc}\frac{\hat{f}_{ABC}\hat{f}_{Abc}}{\hat{f}_{ABC}\hat{f}_{Abc} + \hat{f}_{ABc}\hat{f}_{AbC}}\\ &+n_{AaBBCc}\frac{\hat{f}_{ABC}\hat{f}_{aBc}}{\hat{f}_{ABC}\hat{f}_{aBc} + \hat{f}_{ABc}\hat{f}_{aBC}}\\ &+n_{AaBbCC}\frac{\hat{f}_{ABC}\hat{f}_{abC}}{\hat{f}_{ABC}\hat{f}_{abC} + \hat{f}_{AbC}\hat{f}_{aBC}}\\ &+n_{AaBbCc}\frac{\hat{f}_{ABC}\hat{f}_{abc}}{\hat{f}_{ABC}\hat{f}_{abc} + \hat{f}_{ABc}\hat{f}_{abC} + \hat{f}_{AbC}\hat{f}_{aBc} + \hat{f}_{Abc}\hat{f}_{aBC}}\Bigg).\end{aligned} \tag{1}$$

We can set up equations for the frequencies of the other seven haplotypes on SNPs A, B, and C similarly. Solving these equations can be done by a standard *expectation-maximization* (EM) algorithm [6, 10]. The EM algorithm is iterative. It begins with a random guess of the frequencies. The frequencies obtained at the left hand side in Equation (1) will be repeatedly inserted into the right hand side to improve the estimation. When the improvement is sufficiently small (*e.g.* smaller than a predetermined threshold, typically 10^{-15}), the algorithm terminates and starts a new round with another random guess. After a sufficient number of rounds, it outputs all feasible solutions. We merge the solutions with distances smaller than a threshold (*e.g.* $\epsilon = 10^{-4}$), and obtain the r^2 value using these estimated 3-marker haplotype frequencies.

There are two things that we have to be careful with when applying Hill's method. The first is that the method assumes the population was produced from random mating and Hardy-Weinberg equilibrium holds. Therefore, datasets consisting of related individuals (such as the CEU dataset in HapMap) would not be suitable. The CEU data consists of family trios, not random-mating individuals. The second is that errors caused by the EM algorithm may lead to wrong assignment of haplotypes into the major and minor buckets. For example, Claim 2.1 says that when $n_{hC} = n_{hc}$, it is advantageous to assign the haplotype h to the minor bucket instead of the major bucket. However, if $f_{hC} = f_{hc}$ but $\hat{f}_{hC}$ happens to be slightly higher than $\hat{f}_{hc}$ due to some error in the EM computation, we will assign h to the major bucket without caution. This could lead to a reduced r^2 value. To avoid this, we assign h to the minor bucket as long as $\hat{f}_{hC} < \hat{f}_{hc} + \epsilon$ for some small $\epsilon > 0$.

3. The Greedy Algorithm for Selecting Tag SNPs

In this section, we first define some notations that will be useful in the algorithm, and then describe the algorithm. For simplicity, we present the algorithm for the 2-marker correlation model first, and then generalize it to work for the multi-marker model. At the end of the section, we analyze the time complexity of the algorithm.

3.1. *Some Notations*

In the rest of the paper, we call a group of three SNPs, which includes two potential *tagging* SNPs s_i, s_j and one SNP s_k to be tagged, a *triplet* and denote it as $(s_i, s_j \rhd s_k)$. Similarly, a *quartet* is a group of four SNPs including three potential tagging SNPs and SNP to be tagged. The triplets are used in the 2-marker correlation model and the quartets in the 3-marker correlation model. Each such triplet or quartet has a correlation coefficient r^2 value. We will only be interested in triplets and quartets whose correlation coefficient values r^2 are above a certain threshold. It is convenient to think of the triplets or quartets as edges in a hypergraph. Let us regard SNPs as vertices in the hypergraph. The tagging SNPs in a triplet or a quartet have an *outgoing edge* to the SNP to be tagged. This edge can be also thought of as an *incoming edge* of the tagged SNP from the tagging SNPs. Figure A1 shows an example hypergraph with five triplets.

During a tag SNP selection process, a SNP has three possible states: *uncovered*, *covered* and *picked*. A SNP is picked if it has been selected as a tag SNP. A SNP s is covered if either s is picked or there is a triplet $(s_i, s_j \rhd s)$ where s_i, s_j are picked. In this case we say that SNPs s_i, s_j *cover* s. A SNP is uncovered if it is not picked nor covered. Sometimes, we may use the term *partially covered.* A SNP s is partially covered if it is uncovered and there is a triplet $(s_i, s_j \rhd s)$ such that either s_i or s_j is picked but not both.

3.2. *The Algorithm for the 2-Marker Correlation Model*

An outline of our algorithm is shown in Figure A2. To avoid considering SNPs that cannot possibly be linked, we set a window size of W bps (in terms of the physical distance on a chromosome). For every triplet of SNPs within the window size, we compute its r^2 value as previously described. Then we run an iterative greedy-based algorithm to select a set of tag SNPs as follows. We first initialize all SNPs as uncovered. In each iteration, we pick an appropriate SNP, put it in the tag SNP set, and then check if any uncovered SNPs are now covered due to the newly selected SNP. We repeat this process until all SNPs are covered.

So the main issue is how to pick an appropriate SNP in each iteration. Our first preference is an uncovered SNP that has no incoming edges. A SNP without incoming edges cannot be tagged by any other SNPs and has to be picked as a tag SNP sooner or later. Therefore, we always check if there is such a SNP. If all SNPs have incoming edges, we pick a SNP (covered or uncovered) that can cover the largest number of uncovered SNPs. If there is a tie, the SNP that partially covers the most uncovered SNPs is preferred. Note that, a covered SNP may also be picked in the above if it covers many other SNPs.

After picking each SNP, we need update and remove some triplets that are no longer useful. A triplet $t = (s_i, s_j \rhd s_k)$ should be removed if any one of the following conditions holds:

(1) s_k is covered, and therefore t is useless.
(2) s_i and s_j are both picked. In this case, s_i and s_j together tag s_k. After changing the state of s_k to covered, t is no longer useful.
(3) There is another triplet $t' = (s_i, s'_j \rhd s_k)$ where s'_j is picked. In this case, the triplet t is superseded by the triplet t' and thus redundant.

Note that, although the condition 3 seems optional and unnecessary, it is actually important since keeping useless triplets in the algorithm may actually affect the final result when useless triplets are involved in the partial coverage of SNPs (and ties have to be broken in the algorithm).

Algorithm 3.1 MMTagger(for 2-Marker Model)

```
Require: set of triplets
 1: while there are SNPs uncovered do
 2:     if there is a SNP $s$ with no incoming edges then
 3:         $s^* \leftarrow s$
 4:     else
 5:         $s^* \leftarrow$ a SNP that covers the most uncovered SNPs
 6:     Put $s^*$ in the tag SNP set                 /* $s^*$ is picked */
 7:     for each triplets $t$ of form $(s., s. \rhd s^*)$ do
 8:         remove $t$ and its corresponding edges
 9:     for each triplets $t$ of form $(s^*, s_i \rhd s_j)$ or $(s_i, s^* \rhd s_j)$ do
10:         if $s_i$ is picked then
11:             put $s_j$ into covered SNP set
12:             remove all triplets of form $(s., s. \rhd s_j)$ or $(s., s. \rhd s_j)$
13:         else
14:             remove all triplets of form $(s_i, s. \rhd s_j)$ or $(s., s_i \rhd s_j)$
```

Algorithm 3.1 illustrates the pseudocode of the algorithm. In the algorithm, lines 2–5 pick the next SNP. The subsequent lines update the states of the SNPs and remove useless/redundant triplets.

3.3. *Extension to the 3-Marker Correlation Model*

The extension is straightforward. The outline in Figure A2 still works except that we need now calculate r^2 values for quartets. The above greedy algorithm can also be kept the same, although we should modify the removal of useless/redundant quartets slightly. The third condition should be changed to: if there is another quartet $q' = (s_i, s'_j, s'_k \rhd s_l)$ where s'_j, s'_k are picked, then we remove the quartet q.

It is also straightforward to extend the algorithm to the k-marker correlation model, although calculating r^2 values for groups of k SNPs from haplotype data could be very demanding when k is larger than 4, not to mention doing the calculation for genotype data.

3.4. *Time Complexity*

Suppose that there are m SNPs $s_1, s_2, \ldots, s_m$ on a chromosome sorted by their positions. For simplicity, we assume that there are at most w SNPs within each window of W bps. We need compute the r^2 values of all possible triplets involving three SNPs from the same windows. If the first SNP with the smallest index is among $s_1, s_2, \ldots, s_{m-w}$, there will be $\binom{w-1}{2}$ combinations for the second and the third SNPs. If the first SNP is among $s_{m-w+1}, \ldots, s_m$, then there are totally $\binom{w}{3}$ combinations for all three SNPs. The time complexity of computing the r^2 values is therefore $(m-w)\binom{w-1}{2} + \binom{w}{3} = O(mw^2)$. Similarly, the time complexity to compute r^2 values of all possible quartets is $O(mw^3)$.

Assume that there are T triplets with sufficiently high r^2 values. During the selection of tag SNPs, we maintain a data structure where each SNP has two linked-lists to the triplets containing the SNP. One list contains all the triplets corresponding to the outgoing edges and the other contains all the triplets corresponding to the incoming edges. For each SNP, we also keep track of the number of triplets containing the SNP, and various other statistics on these triplets. Therefore, in each iteration of the selection algorithm, we need only scan all the SNPs and use these numbers to pick an appropriate one. To keep the data structure up-to-date, we need update a triplet $t = (s_i, s_j, \rhd s_k)$ when

(1) s_i or s_j is picked;
(2) s_k is covered and t needs to be removed; or
(3) t is superseded by another triplet and needs to be removed.

If it takes $O(1)$ time to retrieve each triplet that we need update, then the time complexity will be reasonably low. In cases 1 and 2, we can access each of the involved triplets in $O(1)$ time given the data structure. To achieve $O(1)$ access time in case 3, we sort all the triplets in each linked list corresponding to outgoing edges in preprocessing. As a result, if s_i is picked as a tag SNP, then $(s_i, s_j \rhd s_k)$ will supersede all triplets of the form $(s_h, s_j \rhd s_k)$ for some h. These triplets $(s_h, s_j \rhd s_k)$ must be neighbors of $(s_i, s_j \rhd s_k)$ on s_j's outgoing linked list. Therefore, we can access to each of these triplets in $O(1)$ time. Since a triplet may be updated at most 3 times, the time to select tag SNPs is $O(T)$. The preprocessing may take $O(T \log T)$ time.

In practice, the algorithm spends most of its time on evaluating r^2 values. Therefore, we say that the time complexity of the algorithm is $O(mw^2)$ (or $O(mw^3)$) for the 2-marker correlation (or 3-marker correlation) models, respectively.

4. Experimental Result

We have implemented the above algorithm as a C program, simply called MMTagger. In this section, we compare MMTagger with the program LRTag in [11] and the program MultiTag in [4] on real datasets from the HapMap project. The following is a brief summary of the features of the three programs to be compared.

- LRTag [11] uses the traditional single-marker correlation model and works for a single population as well as multiple populations. The algorithm is based on a powerful combinatorial optimization technique called *Lagrangian relaxation*. According to the extensive tests in [11], LRTag outperforms other state-of-the-art single-marker programs such as FESTA [15] and LD-Select [1] in terms of the number of selected tag SNPs. It requires the pairwise r^2 statistics as the input.
- MultiTag [4] uses a multi-marker correlation model which is more restricted than our model. It is a greedy algorithm. The input to MultiTag must be a population haplotype data.
- MMTagger is a greedy algorithm using a more general multi-marker correlation model. Its input is a population data, either in the form of haplotypes or genotypes.

In order to compare these three programs, we need phased haplotype data. We downloaded the CEU ENCODE region data from the HapMap project[b] and use the first 5 of the 10 sample datasets. For LRTag, we need a preprocessing step to calculate the pairwise r^2 values. For both MMTagger and MultiTag, we use a window size W of 100K bps so that SNPs farther than W bps apart are not considered as correlated. To make it fair, we also apply this restriction when calculating r^2 values for LRTag.

Table 1 shows the numbers of the tag SNPs selected by LRTag, MultiTag and MMTagger using different parameters. The reduction of tag SNPs by using the multi-marker correlation models is obvious. However, the running time of the programs based on the multi-marker correlation models (MultiTag and MMTagger) is much longer. LRTag requires only pairwise r^2 values, but MultiTag and MMTagger need r^2 values for each group of three or four SNPs. In general, MMTagger selected fewer tag SNPs than MultiTag. In fact, the improvement is quite significant when the threshold for r^2 is 0.9 or larger.

When comparing the performance of MultiTag and MMTagger, we should also take into account the running time and memory usage. We thus downloaded the entire chromosomal data of the Japanese and Chinese populations from HapMap[c] and used chromosomes 19, 21 and 22 as our test data.

[b] http://www.hapmap.org/downloads/phasing/2005-03_phaseI/ENCODE/

[c] http://www.hapmap.org/downloads/phasing/2006-07_phaseII/phased/

Table 1. Numbers of tag SNPs selected in CEU ENCODE region

Region	ENm010	ENm013	ENm014	ENr112	ENr113
# SNP	459	731	874	868	1035
$r^2 \geq 0.8$					
LRTag	119	88	134	148	133
2-marker MultiTag	75	57	80	87	75
2-marker MMTagger	72	52	78	85	73
3-marker MultiTag	68	53	75	78	64
3-marker MMTagger	62	48	75	68	59
$r^2 \geq 0.9$					
LRTag	148	121	172	204	190
2-marker MultiTag	100	76	111	118	122
2-marker MMTagger	92	73	100	109	115
3-marker MultiTag	91	66	102	101	100
3-marker MMTagger	79	58	85	81	81
$r^2 \geq 0.95$					
LRTag	192	148	196	268	247
2-marker MultiTag	127	96	131	157	156
2-marker MMTagger	117	92	122	141	149
3-marker MultiTag	120	83	119	138	145
3-marker MMTagger	97	66	102	107	112

Hao [4] mentioned two different methods to implement his greedy algorithm and handle a large number of input SNPs: (1) Preprocess and compute all r^2 values, and (2) Calculate r^2 values on the fly while selecting tag SNPs. The former method would lead to heavy memory load and/or file I/O load. The latter method may lead to redundant r^2 value computation. MultiTag employs the latter method. In our implementation of MMTagger, we choose the former method to speed up the computation.

Table 2. MMTagger vs. MultiTag

Chromosome	# SNP	mode	r^2	program	# SNPs Selected	Time (hours)	Memory (M bytes)
JPT+CHB chr19	28931	2-marker	0.9	MultiTag	9600	26hrs	30–35
				MMTagger	9145	2mins	125
		3-marker	0.95	MultiTag	N/A	>700hrs	30–35
				MMTagger	10032	<1hr	657
JPT+CHB chr21	28914	2-marker	0.9	MultiTag	7115	42hrs	30–35
				MMTagger	6766	2mins	187
		3-marker	0.95	MultiTag	N/A	>700hrs	30–35
				MMTagger	7404	<1hr	1210
JPT+CHB chr22	26595	2-marker	0.9	MultiTag	7557	93hrs	30–35
				MMTagger	7221	2mins	183
		3-marker	0.95	MultiTag	N/A	>700hrs	30–35
				MMTagger	7788	3hrs	1216

Note: Both programs were run on a desktop PC with dual AMD Athlon(tm) processors of 2.1 GHz.

Table 2 illustrates a head-to-head comparison between MultiTag and MMTagger. Note that, for the memory usage, we were able to insert some code into MMTagger to obtain the precise maximum memory used by the program. However, we were not able to get the precise memory usage numbers for MultiTag and could only provide a rough estimate. The following gives a detailed comparison between the two programs.

- MMTagger is able to achieve a smaller tag SNP set than MultiTag mostly because our multi-marker correlation model is more general and flexible.
- MMTagger's heuristic to always pick uncovered SNPs with no incoming edges first may also be a factor in its improved performance. This heuristic can be easily incorporated into MultiTag.
- MMTagger may pick a SNP that has been covered if it covers many other SNPs. However, MultiTag always picks an uncovered SNP. Modifying MultiTag to allow covered SNPs to be picked would cost its more time since it calculates r^2 values on the fly. However, this does not impact the running time of MMTagger much because it pre-calculates all r^2 values.
- MMTagger is much faster than MultiTag. Its running time mostly depends on the window size W, since it spends most time on calculating the r^2 values. The running time of MultiTag depends on both the window size W and the number of tag SNPs selected. Hence, it requires more time for higher r^2 thresholds since more tag SNPs would be required. Hao [4] reported that the program took about 300 hours to process the human chromosome 2 data on a typical workstation (Intel Xeon 2.80 GHz CPU and 512 MB memory).
- MMTagger requires much more memory. Its memory usage grows when the r^2 threshold decreases, as more triplets/quartets would be qualified. To run the program on a large chromosome such as human chromosome 2, it require about 4 GB of memory for the 3-marker correlation model when the r^2 threshold is 0.9. However, MultiTag's memory usage is pretty reasonable even for large chromosomes and low r^2 thresholds.
- MMTagger and MultiTag use the window size W in slightly different ways. MMTagger requires that all SNPs in a triplet/quartet should be in the same window, while MultiTag requires that a covered SNP and each of its tagging SNPs should not be farther than W. Therefore, the distance of the two tagging SNPs of a triplet may actually be as far as $2W$ in MultiTag.

As observed before, the 2-marker correlation model improves on the single-marker correlation model significantly. A similar significant improvement from the 2-marker model to the 3-marker model is also shown in Table 2. Although it is likely that the 4-marker model will show further improvements, we are not able to extend the results to the 4-marker model because MMTagger would require too much time and memory on any realistic datasets. For the same reason, MultiTag was only implemented for the 2-marker and 3-marker models in [4, 5]

5. Conclusion

We have introduced a new multi-marker correlation model that generalizes a previous result in the literature. A greedy algorithm is designed to select tag SNPs based on the model. Our experimental results on real datasets from the HapMap project demonstrate that the algorithm produces the most succinct tag SNP sets compared with the previous algorithms.

Acknowledgements

The research is supported in part by NSF grant IIS-0711129 and NIH grant LM008991.

References

[1] Carlson, C., *et al.* Selecting a maximally informative set of single-nucleotide polymorphisms for association analyses using linkage disequilibrium, *The American Journal of Human Genetics*, 74(1):106–120, 2004.

[2] De Bakker, P., *et al.* Transferability of tag SNPs in genetic association studies in multiple populations, *Nature Genetics*, 38(11):1298–1303, 2006.

[3] Halldórsson, B. V., *et al.* Optimal haplotype block-free selection of tagging SNPs for genome-wide association studies, *Genome Research*, 14:1633–1640, 2004.

[4] Hao, K., Genome-wide selection of tag SNPs using multiple-marker correlation, *Bioinformatics*, 23(23):3178–3184, 2007.

[5] Hao, K., Di, X., and Cawley, S., LdCompare: rapid computation of single- and multiple-marker r^2 and genetic coverage, *Bioinformatics*, 23(2):252–254, 2007.

[6] Hill, W., Estimation of linkage disequilibrium in randomly mating populations, *Heredity*, 33(2):229–239, 1974.

[7] Hill, W., Tests for association of gene frequencies at several loci in random mating diploid populations, *Biometrics*, 31(4):881–888, 1975.

[8] Hinds, D., *et al.* Whole-genome patterns of common DNA variation in three human populations, *Science*, 307(5712):1072–1079, 2005.

[9] Johnson, G., *et al.* Haplotype tagging for the identification of common disease genes, *Nature Genetics*, 29:233–237, 2001.

[10] Kalinowski, S. and Hedrick, P., Estimation of linkage disequilibrium for loci with multiple alleles: basic approach and an application using data from bighorn sheep, *Heredity*, 87:698–708, 2001.

[11] Liu, L., Wu, Y., Lonardi, S., and Jiang, T., Effcient algorithms for genome-wide tagSNP selection across populations via linkage disequilibrium criterion, *Proc. 6th Annual International Conference on Computational Systems Bioinformatics*, 67–78, 2007.

[12] Mägi, R., Kaplinski, L., and Remm, M., The whole genome tagSNP selection and transferability among HapMap populations, *Pacific Symposium on Biocomputing*, 11:535–543, 2006.

[13] Marchini, J., *et al.* A comparison of phasing algorithms for trios and unrelated individuals, *The American Journal of Human Genetics*, 78:437–450, 2006.

[14] Patil, N., *et al.* Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21, *Science*, 294(5547):1719–1723, 2001.

[15] Qin, Z., Gopalakrishnan, S., and Abecasis, G., An effient comprehensive search algorithm for tagSNP selection using linkage disequilibrium criteria, *Bioinformatics*, 22(2):220–225, 2006.

[16] Stephens, M., Smith, N., and Donnelly, P., A new statistical method for haplotype reconstruction from population data, *The American Journal of Human Genetics*, 68:978–989, 2001.

[17] Stram, D., *et al.* Choosing haplotype tagging SNPs based on unphased genotype data using a preliminary sample of unrelated subjects with an example from the multiethnic cohort study, *Human Heredity* 55(1):27–36, 2003.

[18] Zhang, Kun and Jin, Li, HaploBlockFinder: Haplotype block analyses, *Bioinformatics*, 19(10):1300–1301, 2003.

Appendix A. The Missing Proof and Figures

Proof of Claim 2.1: Let us consider the frequency table as shown in Table A1, where A is a SNP to be covered/tagged and M is a compound marker representing several (*e.g.* two or three) SNPs. Let n_{AM} denote the number of times that the haplotype AM is observed in the population, $n_A = n_{AM} + n_{Am}$, and n the total number of haplotypes.

Table A1. Number of observations of each haplotype

	A	a	
M	n_{AM}	n_{aM}	n_M
m	n_{Am}	n_{am}	n_m
	n_A	n_a	n

For any haplotype h on M, if $n_{Ah} > n_{ah}$, we would put h in the major bucket, otherwise we put it in the minor bucket. However, when $n_{Ah} = n_{ah}$, it seems that we could put h in either the major bucket or the minor bucket. We show in the following that putting h in the minor bucket leads to a bigger r^2 value between M and A. By definition of the r^2 statistics,

$$\begin{aligned} r^2 &= \frac{(p_{AM} - p_A p_M)^2}{p_A p_a p_M p_m} \\ &= \frac{(n_{AM} \cdot n - n_A n_M)^2}{n_A n_a n_M n_m} \\ &= \frac{(n_{AM} n_{am} - n_{Am} n_{aM})^2}{(n_{AM} + n_{Am})(n_{aM} + n_{am})(n_{AM} + n_{aM})(n_{Am} + n_{am})} \end{aligned}$$

We take the partial derivative of r^2 with respect to n_{AM} and obtain

$$\begin{aligned} \frac{\partial r^2}{\partial n_{AM}} = & \frac{(n_{AM} n_{am} - n_{Am} n_{aM})}{n_A n_a n_M n_m} \cdot \\ & \left(2n_{am} - \frac{(n_{AM} n_{am} - n_{Am} n_{aM})(2n_{AM} + n_{Am} + n_{aM})}{(n_{AM} + n_{Am})(n_{AM} + n_{aM})} \right) \end{aligned}$$

By simplifying the equation, we get

$$\begin{aligned} \frac{\partial r^2}{\partial n_{AM}} &= c \left(2n_{am} - \frac{X(n_A + n_M)}{n_A n_M} \right) \\ \frac{\partial r^2}{\partial n_{Am}} &= c \left(-2n_{aM} - \frac{X(n_A + n_m)}{n_A n_m} \right) \\ \frac{\partial r^2}{\partial n_{aM}} &= c \left(-2n_{Am} - \frac{X(n_a + n_M)}{n_a n_M} \right) \\ \frac{\partial r^2}{\partial n_{am}} &= c \left(2n_{AM} - \frac{X(n_a + n_m)}{n_a n_m} \right) \end{aligned}$$

where $c = \frac{(n_{AM} n_{am} - n_{Am} n_{aM})}{n_A n_a n_M n_m}$, $X = (n_{AM} n_{am} - n_{Am} n_{aM})$.

Suppose that $n_{Ah} = n_{ah}$. If we put haplotype h in the major bucket, then the r^2 value would change by approximately $n_{Ah} \cdot \frac{\partial r^2}{\partial n_{AM}} + n_{ah} \cdot \frac{\partial r^2}{\partial n_{aM}}$. If we put h in the minor bucket, then the r^2 value would change by approximately $n_{Ah} \cdot \frac{\partial r^2}{\partial n_{Am}} + n_{ah} \cdot \frac{\partial r^2}{\partial n_{am}}$. Let

$$\begin{aligned}\Delta_M &= \frac{\partial r^2}{\partial n_{AM}} + \frac{\partial r^2}{\partial n_{aM}} \\ &= c\left(2n_{am} - 2n_{Am} - X\left(\frac{1}{n_A} + \frac{1}{n_a} + \frac{2}{n_M}\right)\right) \\ \Delta_m &= \frac{\partial r^2}{\partial n_{Am}} + \frac{\partial r^2}{\partial n_{am}} \\ &= c\left(2n_{AM} - 2n_{aM} - X\left(\frac{1}{n_A} + \frac{1}{n_a} + \frac{2}{n_m}\right)\right)\end{aligned}$$

We have

$$\begin{aligned}\Delta_m - \Delta_M &= 2c(n_{AM} - n_{aM} + n_{Am} - n_{am}) + cX\left(\frac{2}{n_M} - \frac{2}{n_m}\right) \\ &= 2c(n_A - n_a) + 2cX\left(\frac{1}{n_M} - \frac{1}{n_m}\right)\end{aligned}$$

We need check if $\Delta_m - \Delta_M \geq 0$ holds. By multiplying both side with $\frac{n_M n_m}{2c}$ we get

$$\begin{aligned}&\frac{1}{2c} n_M n_m (\Delta_m - \Delta_M) \\ &= (n_A - n_a) n_M n_m + (n_{AM} n_{am} - n_{Am} n_{aM})(n_m - n_M) \\ &= (n_{AM} + n_{Am} - n_{aM} - n_{am})(n_{AM} + n_{aM})(n_{Am} + n_{am}) \\ &\qquad + (n_{AM} n_{am} - n_{Am} n_{aM})(n_{Am} + n_{am} - n_{AM} - n_{aM}) \\ &= n_{AM}(n_{AM} + n_{aM}) n_{Am} + n_{Am} n_{AM}(n_{Am} + n_{am}) \\ &\qquad - n_{aM}(n_{AM} + n_{aM}) n_{am} - n_{am} n_{aM}(n_{Am} + n_{am}) \\ &= n_{AM} n_{Am} \cdot n - n_{aM} n_{am} \\ &= n(n_{AM} n_{Am} - n_{aM} n_{am})\end{aligned}$$

where $n = n_{AM} + n_{Am} + n_{aM} + n_{am}$. Therefore, $\Delta_m \geq \Delta_M$ if and only if $n_{AM} n_{Am} \geq n_{aM} n_{am}$. When the latter inequality holds, putting the haplotype h in the minor bucket will result in a higher r^2 value.

Since $n_{AM} + n_{Am} = n_A > n_a = n_{aM} + n_{am}$, $n_{AM} n_{Am}$ tends to be greater than $n_{aM} n_{am}$ in practice. Moveover, even when $n_{AM} n_{Am} < n_{aM} n_{am}$, putting the haplotype h in the minor bucket would increase n_{Am} and n_{am} at the same time, and hence result in a greater increase in $n_{AM} n_{Am}$ than in $n_{aM} n_{am}$ since n_{AM} is usually larger than n_{aM}. This could help improve the r^2 value in the long run. Therefore, putting h in the minor bucket may still be better in this case. For example, suppose $n_{AM} = 100$, $n_{Am} = 0$, $n_{aM} = 5$, and $n_{am} = 20$ before haplotype h is considered. If $n_{Ah} = n_{ah} = 1$, then putting h in the major (or minor) bucket results in $r^2 = 0.7261$ (or $r^2 = 0.7235$, respectively). However, if $n_{Ah} = n_{ah} = 3$, then putting h in the major (or minor) bucket leads to $r^2 = 0.6628$ (or $r^2 = 0.6631$, respectively).

Note that, the tag SNP selection program MultiTag in [4, 5] considers all the possible splits of the haplotypes in question and picks the one that results in the highest r^2 value. So, ties between haplotype counts are not an issue. However, we cannot afford doing this in our tag SNP selection program MMTagger (to be introduced in Section 4) because our multi-marker correlation model allows for many more possible splits. Trying all such splits would be very inefficient. Since the above analysis shows that putting haplotype h in the

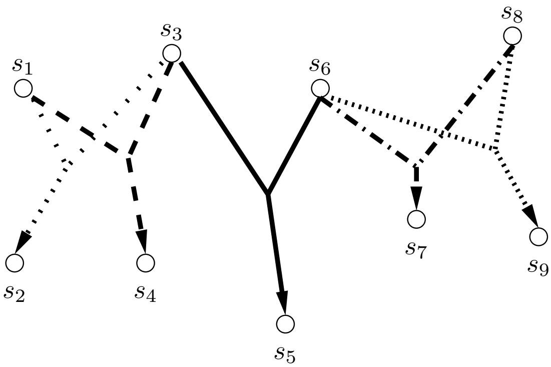

Fig. A1. An example with five triplets: $(s_1, s_3 \rhd s_2)$, $(s_1, s_3 \rhd s_4)$, $(s_3, s_6 \rhd s_5)$, $(s_6, s_8 \rhd s_7)$ and $(s_6, s_8 \rhd s_9)$.

minor bucket is generally better when we have a tie $n_{Ah} = n_{ah}$, MMTagger always puts h in the minor bucket when such a tie arises. □

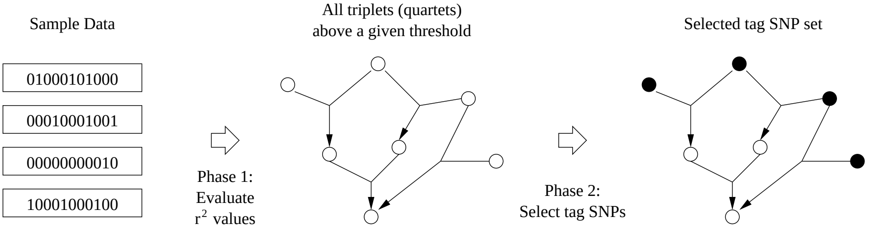

Fig. A2. An outline of our algorithm.