JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 17, Number 1, 2010

Pp. 21–37
DOI: 10.1089/cmb.2007.0228

# Efficient Genome-Wide TagSNP Selection Across Populations via the Linkage Disequilibrium Criterion

LAN LIU,[1,2] YONGHUI WU,[1,2] STEFANO LONARDI,[1] and TAO JIANG[1]

## ABSTRACT

**In this article, we studied the tag single-nucleotide polymorphism (tagSNP) selection problem on multiple populations using the pairwise $r^2$ linkage disequilibrium criterion. We proposed a novel combinatorial optimization model for the tagSNP selection problem, called the *minimum common tagSNP selection* (MCTS) problem, and presented efficient solutions for MCTS. Our approach consists of the following three main steps: (i) partitioning the SNP markers into small disjoint components, (ii) applying some data reduction rules to simplify the problem, and (iii) applying either a fast greedy algorithm or a Lagrangian relaxation algorithm to solve the remaining (general) MCTS. These algorithms also provide lower bounds on tagging (*i.e.*, the minimum number of tagSNPs needed). The lower bounds allow us to evaluate how far our solution is from the optimum. To the best of our knowledge, it is the first time the tagging lower bounds are discussed in the literature. We assessed the performance of our algorithms on real HapMap data for genome-wide tagging. The experiments demonstrated that our algorithms run 3–4 orders of magnitude faster than the existing single-population tagging programs such as FESTA, LD-Select, and the multiple-population tagging method MultiPop-TagSelect. Our method also greatly reduced the required tagSNPs compared with LD-Select on a single population and MultiPop-TagSelect on multiple populations. Moreover, the numbers of tagSNPs selected by our algorithms are almost optimal because they are very close to the corresponding lower bounds obtained by our method.**

**Key words:** genome-wide tagSNP selection, greedy algorithm, HapMap, Lagrangian relaxation, linkage disequilibrium, multiple populations.

## 1. INTRODUCTION

THE RAPID DEVELOPMENT OF HIGH-THROUGHPUT GENOTYPING technologies has recently enabled genome-wide association studies to detect connections between genetic variants and human diseases. *Single-nucleotide polymorphism* (SNP) is the most frequent form of polymorphism in the human genome. Common SNPs with *minor-allele frequency* (MAF) of 5% have been estimated to occur once every ~600 bp (Kruglyak and Nickerson, 2001), and there are more than 10 million verified SNPs in dbSNP (The International SNP Working Group, 2001). Given these numbers, it is currently infeasible to take into account

[1]Department of Computer Science and Engineering, University of California, Riverside, California.
[2]Google, Inc., Mountain View, California.

all the available SNPs to carry out association studies. This motivates the selection of a *subset* of informative SNPs, called *tagSNPs*.

The selection of tagSNPs *in silico* is a well-studied research topic. Existing computational methods for tagSNP selection can be classified into the following two categories: *haplotype-based* methods (Johnson et al., 2001; Patil et al., 2001; Gabriel et al., 2002; Wang et al., 2002; Zhang et al., 2002, 2005; Avi-Itzhak et al., 2003; Ke and Cardon, 2003; Sebastiani et al., 2003) and *haplotype-independent* methods (Carlson et al., 2004; Halperin et al., 2005; Hampe et al., 2003; Lin et al., 2004; Liu et al., 2006; Magi et al., 2006; Phuong et al., 2005; Stram et al., 2003; Qin et al., 2006). The haplotype-based methods require phased multilocus haplotypes, whereas the haplotype-independent methods do not require haplotype information. The main shortcoming of haplotype-based methods is that the preprocessing step (*i.e.*, the inference of haplotypes from genotypes) is computationally demanding. In addition, as there is not an authoritative inference method, the haplotypes generated by the existing haplotype inference methods are often quite different (Zhang et al., 2002; Ding et al., 2005; Zeggini et al., 2005). Consequently, the tagSNPs selected by the haplotype-based methods would be quite different. Recently, Carlson *et al.* (2004) proposed a haplotype-independent method that employs the $r^2$ *linkage disequilibrium* (LD) statistical criterion to measure the association between SNPs. The tagSNPs selected by this method are shown to be effective in disease association mapping studies, because the measure $r^2$ is directly related to the statistical power of association mapping. Because this method has comparable performance at a lower computational cost than many other methods (Stram et al., 2003; Zhang and Jin, 2003), tagging approaches based on $r^2$ LD statistics have gained popularity among researchers in the SNP community (Zhang and Jin, 2003; Carlson et al., 2004; Hinds et al., 2005; Bakker et al, 2006; Magi et al., 2006; Qin et al., 2006).

Most approaches using the $r^2$ criterion require that tagSNPs be defined within a single population, because LD patterns (see the caption of Fig. 1A for a definition) are quite susceptible to population stratification (Carlson et al., 2004). In two populations with different evolutionary histories, a pair of SNPs having remarkably different allele frequencies and very weak LD may show strong LD in the admixed population (see such an example in Table 1). Recent study (Conrad et al., 2006) showed that the LD patterns and allele frequencies across populations are very different (Sawyer et al., 2005; Conrad et al., 2006) in fact. For example, among the populations collected in the HapMap project (*i.e.*, YRI: Yoruba in Ibadan, Nigeria; CEU: Utah residents with Northern and Western European ancestry; CHB: Han Chinese in Beijing, China; and JPT: Japanese in Tokyo, Japan), 81% of the SNPs in YRI population have a near perfect proxy (*i.e.*, SNPs that have $r^2 \geq 0.8$ with other SNPs), whereas in the other three populations, 91% of the SNPs have a near perfect proxy (International HapMap Consortium, 2005). Therefore, tagSNPs picked from the combined populations or one of the populations might not be sufficient to capture the variations in all populations. To maintain the power of association mapping, we need to generate a common (or universal) tagSNP set to type all the populations with sufficient accuracy.

A simple approach to select a universal tagSNP set is to tag one population first and then select a supplementary set for each of the other populations one by one (Bakker et al., 2006; Magi et al., 2006; Need and Goldstein, 2006). For instance, we can select a tagSNP set for non-African populations and a supplement for populations with significant African ancestry (Need and Goldstein, 2006). However, this sequential approach might not give a satisfactory solution, as the tagSNP set selected for one population

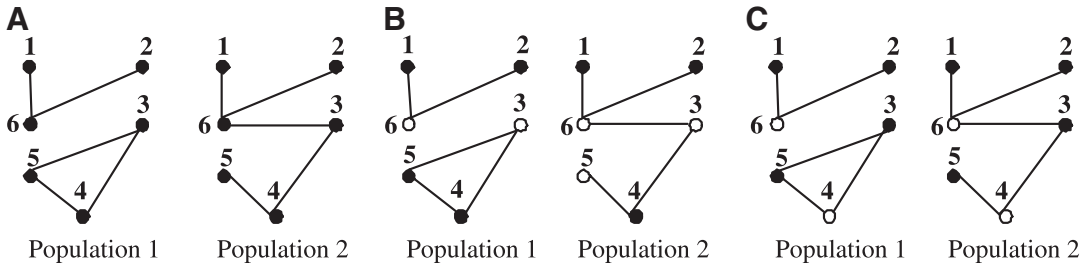


**FIG. 1.** (**A**) Linkage disequilibrium (LD) patterns in two populations. The vertices denote the SNP markers and the edges denote pairs of markers with strong LD (*i.e.,* the $r^2$ measure between the markers is greater than a given threshold). (**B**) Tagging results of the above simple sequential approach. We first chose markers 3 and 6 to tag population 1 and then chose an additional marker 5 to tag population 2. Three markers were selected in total to tag both populations. (**C**) Tagging results of an improved approach. We selected markers 4 and 6, considering both populations simultaneously. Only two markers were selected in total to tag both populations. SNP, single-nucleotide polymorphism.

TABLE 1. $r^2$ STATISTICS FOR A PAIR OF SNP MARKERS IN A SINGLE AND ADMIXED POPULATIONS

| | *Population 1* | | | | *Population 2* | | | | *Population 3* | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | *B* | *b* | | | *B* | *b* | | | *B* | *b* | |
| *A* | 0.9025 | 0.0475 | 0.95 | *A* | 0.0025 | 0.0475 | 0.05 | *A* | 0.4525 | 0.0475 | 0.5 |
| *a* | 0.0475 | 0.0025 | 0.05 | *a* | 0.0475 | 0.9025 | 0.95 | *a* | 0.0475 | 0.4525 | 0.5 |
| | 0.95 | 0.05 | $r^2 = 0$ | | 0.05 | 0.95 | $r^2 = 0$ | | 0.5 | 0.5 | $r^2 = 0.6561$ |

One SNP has alleles denoted as *A* and *a* while the other SNP has alleles denoted as *B* and *b*. Population 3 is an even mixture of populations 1 and 2.

might be far from being adequate to type the SNPs of the remaining populations. As a result, the supplementary tagSNP sets are large and the total number of tagSNPs chosen is far from the optimum. Moreover, the performance of the approach is sensitive to the specific order of the input populations. To generate the smallest set of tagSNPs on $K$ populations, one would have to execute the tagging procedure $K!$ times considering all possible orderings, which would be extremely inefficient for genome-wide tagging. We can improve the performance of the tagging approach by evaluating multiple populations at the same time. When choosing tagSNPs, we prefer those with "good properties" with respect to the collection of populations as a whole. An example of our tagging strategy is given in Figure 2.

### *1.1 Previous work on tagSNP selection based on the LD criterion*

There is a large body of scientific literature on the problem of selecting tagSNPs based on the $r^2$ LD criterion. Carlson *et al.* (2004) suggested a greedy procedure called LD-Select, which works as follows: (i) select the SNP with the maximum number of proxies, (ii) remove the SNP and its proxies from consideration, and (iii) repeat the above two steps until all SNPs have been tagged. This algorithm is very simple; however, it may miss solutions with the smallest number of tagSNPs in general, as pointed out by Qin et al. (2006). More recently, Qin *et al.* (2006) implemented a comprehensive search algorithm called FESTA, which first breaks down a large set of markers into disjoint pieces (called *precincts*) and then performs an exhaustive search on each piece if the estimated computational cost is below a certain threshold. FESTA usually gives a better solution than LD-Select, but because of the fact that it employs exhaustive search, it is too slow to be practical for genome-wide tagSNP selection.

The above two methods are only applicable to single-population tagSNP selection. Recently, Howie *et al.* presented an algorithm for multiple populations by extending LD-Select, called MultiPop-TagSelect. MultiPop-TagSelect combines the tagSNPs selected for each population by LD-Select to produce a universal tagSNP set for a collection of populations (Howie et al., 2006). The algorithm works reliably, and it could in principle be used with any tagSNP selection method for single populations. However, its accuracy highly depends on the performance of the single-population tagSNP selection method. Magi *et al.* (2006) also designed a software tool called REAPER which is rather similar to LD-Select if applied to a single

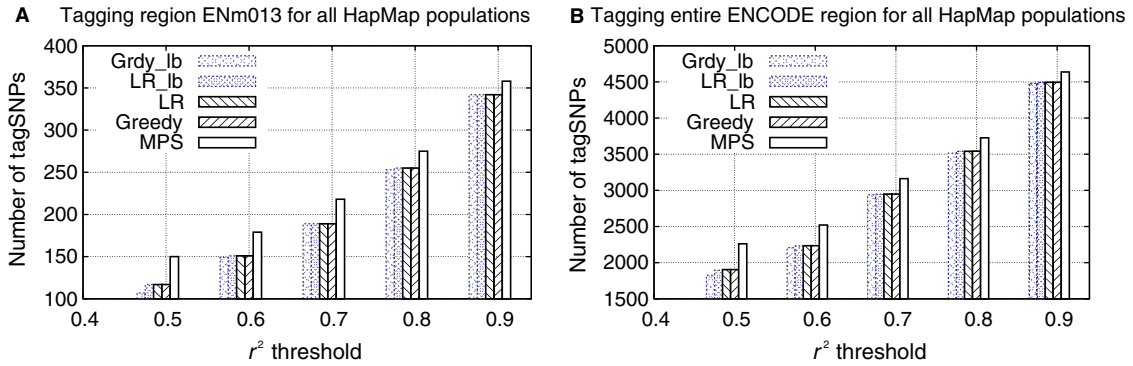


**FIG. 2.** (**A**) Tagging for HapMap populations on region ENm013 with 783 markers. (**B**) Tagging for HapMap populations on all ENCODE regions with 10,859 markers. Here, "Grdy_lb" stands for "Greedy-Tag_lb," "LRTag_lb" stands for "LRTag_lb," "LR" stands for "LRTag," "Greedy" stands for "GreedyTag," and "MPS" stands for "MultiPopTagSelect."

population. To select a universal tagSNP set for several populations, it first selects a tagSNP set for one population, and then it selects a supplement for the remaining populations one by one. As above mentioned, the performance of the method crucially depends on the choice of the initial tagSNP set and the ordering of the populations. It is not clear, moreover, how one should select tagSNPs for the first population so as to minimize the size of the final solution.

### *1.2 Our contribution on tagSNP section based on the LD criterion*

In this article, we took a different approach to the multipopulation tagSNP selection problem. Contrary to the previous methods, we do not generate a tagSNP set for each individual population separately, but rather we evaluate all the populations at the same time. The method that we proposed could be used to generate a universal or cosmopolitan tagSNP set for multiethnic, ethic-unknown, or admixed populations (Howie et al., 2006).

The main idea of our approach is to transform a multipopulation tagSNP selection problem, called the *minimum common tagSNP selection* (MCTS) problem (will be defined more precisely later in the article), into a minimum common dominating vertex set problem on multiple graphs. Each graph corresponds to one of the populations under consideration. The vertices in a graph correspond to the SNP markers of the population, and there is an edge between two markers when they are in strong LD (according to some given threshold). To find an optimal solution MCTS, we first decomposed it into disjoint subproblems, each of which is essentially a connected component of the union graph[1] and represents a precinct as defined by Qin et al. (2006). Then, for each precinct, we apply three data reduction rules repeatedly to further reduce the size of the subproblem, until none of the rules can be applied anymore. Finally, the reduced subproblems are solved by either a simple greedy approach (similar to *cosmopolitan tagging*) (Bakker et al., 2006) or a more sophisticated Lagrangian relaxation heuristic. Both algorithms will be explained in detail later in the article. Along with the solution produced by our algorithm, we also obtained lower bounds on the minimum number of tagSNPs required, which allows us to quantitatively assess how close our solution is from the true optimum.

We evaluated the performance of our method on real HapMap data for genome-wide tagging. The experimental results demonstrate that our algorithms run 3–4 orders of magnitude faster than the existing single-population tagging programs such as FESTA, LD-Select, and the multiple-population tagging method MultiPop-TagSelect. Our method also greatly reduced the required tagSNPs compared with LD-Select on a single population and MultiPop-TagSelect on multiple populations. Moreover, the numbers of tagSNPs selected by our algorithms are almost optimal because they are very close to the corresponding lower bounds provided by our method. For example, the gap between our solution and the lower bound is 1061 SNPs with $r^2$ threshold being 0.5 and 142 SNPs with the $r^2$ threshold being 0.8, given the entire human genome with 2,862,454 SNPs (MAF being 5%).

The rest of the article is organized as follows: In Section 2, we first propose a combinatorial optimization model for the MCTS problem and then present a computational complexity result. In Section 3, we introduce three rules to reduce the size of the problem and devise a greedy tagging algorithm, called GreedyTag, and a Lagrangian relaxation heuristic, called LRTag. After showing the experimental results in Section 4, we conclude the article with some remarks about the performance of our tagging method in Section 5.

## 2. FORMULATION OF THE MCTS PROBLEM

Consider $K$ distinct populations and a set $V$ of biallelic SNP markers $v_1, v_2, \ldots, v_n$. For convenience, we label the markers in all populations uniformly and uniquely. As the $r^2$ coefficient is unreliable for rare SNPs when the sample size is small (Carlson et al., 2004), we will consider only SNPs with MAF $\geq 5\%$. The set of SNPs might be different from population to population. We use $V_i \subseteq V$ to denote the SNP set in population *i.* Clearly, we have $V = V_1 \cup V_2 \cup \cdots \cup V_k$.

For a pair of SNP markers $v_{j_1}$ and $v_{j_2}$ in a population $i$ (for any $1 \leq i \leq K$), the $r^2$ coefficient between them is denoted by $r_i^2(v_{j_1}, v_{j_2})$. Markers $v_{j_1}$ and $v_{j_2}$ are said to be in *high LD* in population *i,* if $r_i^2(v_{j_1}, v_{j_2}) \geq \gamma_0$, where $\gamma_0$ is a predefined threshold ($\gamma_0$ will be set to 0.5 or higher in our study). Moreover, $v_{j_1}$ (or $v_{j_2}$) is considered being the *tagSNP* or *proxy* for $v_{j_2}$ (or $v_{j_1}$, respectively) in population *i.* For convenience, we define $E_i$ to be the set containing all the high-LD marker pairs in population $i$, *i.e.*, $E_i = \{(v_{j_1}, v_{j_2}) | r_i^2(v_{j_1}, v_{j_2}) \geq \gamma_0, v_{j_1}, v_{j_2} \in E_i\}$. Now we can formally define the MCTS problem.

[1]Given graphs $G_i = (V_i, E_i)(1 \leq i \leq k)$, the union graph is defined as $G = (V, E)$, where $V = \cup_i V_i$ and $E = \cup_i E_i$.

MINIMUM COMMON TAGSNP SELECTION

*Instance:* A collection of $K$ populations and a set $V$ of biallelic SNP markers. Each population $i$ ($1 \le i \le K$) has its marker set $V_i \subseteq V$ and LD patterns $E_i = \{(v_{j_1}, v_{j_2}) | r_i^2(v_{j_1}, v_{j_2}) \ge \gamma_0,\ v_{j_1}, v_{j_2} \in V_i\}$ where $\gamma_0$ is a predefined threshold.

*Feasible solution:* A subset $T \subseteq V$ such that for any marker $v \in V_i$, $v \notin T$ from some population $i$, there exists a marker $v'$ in $T \cap V_i$ with $(v, v') \in E_i$ (that is, $r_i^2(v, v') \ge \gamma_0$).

*Objective:* Minimize $|T|$.

A natural generalization of the above definiton is to allow the predefined threshold $\gamma_0$ to vary across populations. As the techniques discussed for a fixed threshold in our article can be trivially extended to solve the general problem, to simplify the presentation, we will assume that the threshold is fixed for all populations below.

It is easy to observe that any feasible solution to the MCTS problem is a common dominating vertex set in the graphs $\{Gi | 1 \le i \le K\}$, where $G_i = (V_i, E_i)$. In particular, the smallest set of tagSNPs for a single population is a minimum dominating vertex set of the corresponding graph. Obviously, the MCTS problem is NP-hard, because it is a generalization of the minimum dominating vertex set problem, which is known to be NP-hard (Bar-Yehuda and Moran, 1984).

**Theorem 1.** *The MCTS problem is NP-hard.*

We introduced some additional notations to be used later. To differentiate the occurrences of a marker in different populations, we used $v_j^i$ to represent the occurrence of marker $j$ in the $i^{th}$ population (if the marker occurs). Given a marker $v_j \in V$, we defined the following two sets:

$$\begin{aligned} N^i(v_j) &= \{v_{j'}^i | (v_j, v_{j'}) \in E_i, v_j, v_{j'} \in V_i\} \cup \{v_j^i | v_j \in V_i\} \\ N^*(v_j) &= \bigcup_{1 \le i \le K} N^i(v_j) \end{aligned} \tag{1}$$

The set $N^i(v_j)$ represents the subset of markers (actually, their occurrences) in population $i$ that are in strong-LD with $v_j$, and the set $N^*(v_j)$ represents the union of such subsets for all the populations. Note that $N^i(v_j)$ is empty if $v_j \notin V_i$. Given a marker $v_j \in V_i$ from population $i$, we defined the following set:

$$C(v_j^i) = \{v_{j'} | (v_j, v_{j'}) \in E_i, v_j, v_{j'} \in V_i\} \cup \{v_j\} \tag{2}$$

The set $C(v_j^i)$ is the subset of markers each of which can tag the occurrence $v_j^i$, whereas $N^*(v_j)$ is the subset of occurrences that the marker $v_j$ can tag.

Based on the above definitions, the MCTS problem can also be viewed as the following set cover problem. Given the universe $\mathcal{U} = \bigcup_{1 \le i \le K} \{v_j^i | v_j \in V_i\}$ and the collection $\mathcal{C} = \{N^*(v_j) | v_j \in V\}$, we found a subcollection of sets from $\mathcal{C}$ to cover $\mathcal{U}$. Clearly, the number of sets in a minimum set cover is equal to the number of markers in a minimum tagSNP set.

As a consequence, approximation algorithms that solve set cover can be applied to the MCTS problem. In practice, greedy algorithms are commonly used for set cover because of their simplicity and effectiveness. The simplest greedy algorithm for set cover, which picks the set that covers the most number of uncovered elements each time, achieves an approximation ratio of $\log(m)$, where $m$ is the number of elements to be covered (Vazirani, 2003). This implies a $\log(Kn)$ approximation algorithm for MCTS, $|\mathcal{U}| \le Kn$. However, the approximation ratio of $\log(Kn)$ could be too large in practice, because of the fact that $V$ may contain millions of markers and $n = |V|$. We thus hoped to design efficient heuristics to provide better solutions in this article. In fact, one of our results showed that a greedy algorithm augmented with some carefully designed heuristics can achieve a nearly optimal approximation ratio in practice.

## 3. OPTIMIZATION TECHNIQUES TO SOLVE THE MCTS PROBLEM

In principle, a minimum common tagSNP set can be found by exhaustive search. In reality, there are millions of SNP markers, and it is infeasible to conduct the exhaustive search. As human chromosomes consist of high-LD regions (*i.e.,* haplotype blocks) interspersed with *recombination hotspots*, we partition the markers into precincts such that markers in strong LD will belong in the same precinct. In this way, we could narrow down the search space and thus improve the efficiency of our algorithm.

To deal with multiple populations, we extend the concept of precinct defined originally by Qin et al. (2006). We say that two markers are in the same *precinct* if and only if they are in strong LD in some

population. Based on the simple observation that no marker in a precinct can tag a marker in another precinct, we can obtain a minimum tagSNP set for the combining the minimum tagSNP sets for each precinct. The precincts can be easily identified by running a breath first search in the union graph $G$. By partitioning the markers into precincts, we decompose the original problem into a set of disjoint subproblems of much smaller sizes. We then select tagSNPs for each precinct independently, which could save a lot of running time.

### 3.1. Data reduction rules

To further reduce the subproblem sizes and improve efficiency, we introduced three simple data reduction rules.

- Rule 1: *Pick all irreplaceable markers*. If a marker $v_j$ has no proxy from population $i$ (that is, $v_j$ is a singleton in $G_i = (V_i, E_i)$), then marker $v_j$ must be in the minimum tagSNP set.
- Rule 2: *Remove less informative markers*. Given two markers $v_{j'}$ and $v_j$, if $N^*(v_{j'}) \subseteq N^*(v_j)$, we say that $v_j$ is more *informative* than $v_{j'}$. Similarly, given a set of markers $v_{j_1}, v_{j_1}, \ldots, v_{j_k}$ if $N^*(v_{j_1}) \subseteq N^*(v_{j_2}) \subseteq \ldots \subseteq N^*(v_{j_k})$, $v_{j_k}$ is called the *maximally informative* SNP marker in the set. It is clear that we can discard less informative SNPs and only keep those maximally informative ones without degrading the quality of the solution.
- Rule 3: *Remove less stringent occurrences*. Given two occurrences $v_{j'}^{i'}$ and $v_j^i$, if $C_j^i \subseteq C_{j'}^{i'}$, we say that $v_{j'}^{i'}$ is *less stringent* than $v_j^i$. Similarly, given a set of occurrences $v_{j_1}^{i_1}, v_{j_2}^{i_2}, \ldots, v_{j_k}^{i_k}$ if $C_{j_k}^{i_k} \subseteq \ldots \subseteq C_{j_2}^{i_2} \subseteq C_{j_1}^{i_1}$, the occurrence $v_{j_k}^{i_k}$ is called the *most stringent* occurrence in the set. We observed that the markers selected to tag the most stringent occurrences will also tag the less stringent occurrences. Therefore, we considered only the most stringent occurrences and discard the others.

The above rules can also be viewed as data reduction rules applied to a $0/1$ matrix obtained as follows: Given the notations of the occurrence set $\mathcal{U}$, the marker set $V$ and the neighborhood collections $C$ introduced in the previous section, the rows in the matrix represent $\mathcal{U}$, the columns denote $V$, and each cell $(i, j)$ indicates whether the marker corresponding to column $j$ can tag the occurrence corresponding to row $i$ (*i.e.,* the value of a cell is set to 1 if the marker can tag the occurrence, and 0 otherwise). Thus, rule 2 is equivalent to redundant column deletion, and rule 3 is equivalent to redundant row deletion.

The above rules can be applied repeatedly and in any combination whenever applicable. The reduced problem obtained after the application of the above data reduction rules will be subject to our greedy algorithm or Lagrangian relaxation algorithm, as explained next.

### 3.2. A greedy algorithm for MCTS

Greedy algorithms are often desirable because of their simplicity and efficiency. The greedy algorithm, *GreedyTag,* below is adapted from the greedy algorithm for the set cover problem as presented by Vazirani (2003). By first applying the above data reduction rules, we will show later in the article that GreedyTag greatly outperforms the other greedy algorithms such as LD-Select and MultiPop-TagSelect. Moreover, a lower bound, called *GreedyTag_lb,* is produced by GreedyTag, which is equal to the number of tagSNPs selected by data reduction rule 1. Even though the lower bound is is somewhat loose because we consider only rule 1, it turned out to be pretty tight in our experiments on real data (see Section 4 for more details). We present the pseudo-code of GreedyTag in the following Algorithm 1.

### 3.3. A Lagrangian relaxation algorithm for MCTS

A subset $T$ of SNPs can be denoted by its characteristic vector $\boldsymbol{t} = t_1 t_2 \ldots t_n$, where $t_i = 1$ if $v_i \in T$, and $t_i = 0$ otherwise. It is thus easy to formulate the following integer linear program for MCTS.

$$\begin{aligned} \text{Minimize} \quad & |T| = \sum\nolimits_{1 \le j \le n} t_j \\ \text{subject to} \quad & \sum\nolimits_{v_{j'} \in C(v_j^i)} t_{j'} \ge 1 \qquad 1 \le i \le K \text{ and } 1 \le j \le n \\ & t_j \in \{0, 1\}, 1 \le j \le n \end{aligned} \tag{3}$$

Our second algorithm for MCTS is based on the Lagrangian relaxation framework. We assigned a non negative vector $\boldsymbol{\lambda} = \lambda_{11} \lambda_{12} \ldots \lambda_{K,n}$ of Lagrangian multipliers to the inequalities and obtained the following relaxed integer program:

**Algorithm 1** (GreedyTag: Greedy Algorithm for TagSNP Selection in Multiple Populations)

**Input**: A set $V$ of biallelic single-nucleotide polymorphism (SNP) markers and their pairwise $r^2$ linkage disequilibrium (LD) statistics in $K$ distinct populations. A predefined threshold $\gamma_0$ for $r^2$ LD statistics.
**Output**: A feasible tagSNP set $T \subseteq V$, and a lower bound $LB$.
**Begin**
Partition markers into precincts. Let the set of precincts be $\mathcal{P}$.
**For each** precinct $p \in \mathcal{P}$ {*the following will be executed in parallel on a multi-processor machine*}
Let $U$ be the set of SNPs and $W$ the set of marker occurrences in $p$.

Step 1: *Apply the three data reduction rules.*
$T_p \Leftarrow \emptyset$; $LB_p \Leftarrow 0$; UPDATED $\Leftarrow$ true;
**While** UPDATED {*execute the optimal rules iteratively*}
UPDATED $\Leftarrow$ false;
If there exists an irreplaceable marker $v_j \in U$ {Rule 1}
$U \Leftarrow U - \{v_j\}$; $W \Leftarrow W - N^*(v_j)$;
$T_p \Leftarrow T_p \cup \{v_j\}$; $LB_p \Leftarrow LB_p + 1$; UPDATED $\Leftarrow$ true;
If there exists a less informative marker $v_j \in U$ {Rule 2}
$U \Leftarrow U - \{v_j\}$; UPDATED $\Leftarrow$ true;
If there exists a less stringent occurrence $v_j^i \in W$ {Rule 3}
$W \Leftarrow W - \{v_j^i\}$; UPDATED $\Leftarrow$ true;
**For each** $v_j \in U$
$D(v_j) \Leftarrow N^*(v_j) \cap W$;

Step 2: *Select tagSNPs greedily.*
**While** $W$ is non-empty {*there are markers to be tagged*}
Let $v_{j_0} \Leftarrow argmax_{v_j \in U} |D(v_j)|$;
$T_p \Leftarrow T_p \cup \{v_{j_0}\}$; $U \Leftarrow U - \{v_{j_0}\}$; $W \Leftarrow W - N^*(v_{j_0})$;
**For each** $v_j \in U$
$D(v_j) \Leftarrow D(v_j) \cap W$;
$T \Leftarrow \bigcup_{p \in \mathcal{P}} T_p$; $LB \Leftarrow \sum_{p \in \mathcal{P}} LB_p$
**Output** $T$, $LB$ {*output the solution and lower bound*}
**End**

$$\begin{aligned} &\text{Minimize} \quad L(\mathbf{t}, \boldsymbol{\lambda}) = \sum\nolimits_{1 \le j \le n} t_j + \sum\nolimits_{1 \le i \le K,\, 1 \le j \le n} \lambda_{i,j} \left(1 - \sum\nolimits_{v_{j'} \in C(v_j^i)} t_{j'}\right) \\ &\text{subject to} \quad t_j \in \{0, 1\}, \lambda_{i,j} \ge 0, 1 \le i \le K, 1 \le j \le n \end{aligned} \tag{4}$$

For a given $\boldsymbol{\lambda}$, define $L(\boldsymbol{\lambda}) = min\ L(\mathrm{t}, \boldsymbol{\lambda})$. We observed that the size of any feasible tagSNP set $T$ would be an upper bound for $L(\boldsymbol{\lambda})$ in (4), and any $L(\boldsymbol{\lambda})$ would be a lower bound for $|T|$. Hence, we looked for $max\ L(\boldsymbol{\lambda})$, which gives the best lower bound for $min\ |T|$.

For any given $\boldsymbol{\lambda}$, we can easily obtain $L(\boldsymbol{\lambda})$ in (4) as follows: For convenience, we define $s(t_j)$ $(1 \le j \le n)$ as

$$s(t_j) = 1 - \sum_{1 \le i \le K,\ v_{j'} \in C(v_j^i)} \lambda_{i,j'} = 1 - \sum_{v_{j'}^i \in N^*(v_j)} \lambda_{i,j'}, \tag{5}$$

which are the *Lagrangian costs* associated with $t_j$ in Equation 4. Rearranging the terms in Equation 4, we have the objective function $L(\mathbf{t}, \boldsymbol{\lambda}) = \sum_{1 \le i \le K,\, 1 \le j \le n} \lambda_{i,j} + \sum_{1 \le j \le n} s(t_j) \cdot t_j$. To minimize the objective function, we have to set $t_j = 0$ if $s(t_j) > 0$, $t_j = 1$ if $s(t_j) < 0$, and $t_j$ an arbitrary value if $s(t_j) = 0$.

The vector $t$ obtained above may not be a feasible solution to Equation 3. In other words, some occurrence might not be tagged by any marker in $T = \{v_j | t_j = 1, 1 \le j \le n\}$ induced by the characteristic vector $t$. We will adopt a strategy called the *reduced cost heuristic* introduced by Balas and Carrera (1996) to deal with this issue.

Next we need to find a good multiplier vector $\boldsymbol{\lambda}$, *i.e.,* one that gives a near optimal lower bound. We utilized a standard optimization technique called *subgradient optimization* (Balas and Carrera, 1996), which iteratively updates the solution toward the subgradient direction to reach the optimum. We can define

$$S(\lambda_{i,j}) = 1 - \sum_{v_{j'} \in C(v_j^i)} t_{j'} \tag{6}$$

which simplifies $L(\mathbf{t}, \boldsymbol{\lambda}) = \sum_{1 \le j \le n} t_j + \sum_{1 \le i \le K, 1 \le j \le n} S(\lambda_{i,j}) \cdot \lambda_{i,j}$. Obviously, $\nabla \boldsymbol{\lambda} = (\nabla \lambda_{11}, \nabla \lambda_{12}, \ldots, \nabla \lambda_{K,n})$, where $\nabla \lambda_{i,j} = S(\lambda_{i,j})$. Starting from a initial setting $\boldsymbol{\lambda}^0$, we sequentially generated $\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, \boldsymbol{\lambda}^3, \ldots$, based on the following formula:

$$\boldsymbol{\lambda}^{k+1} = max\{\mathbf{0}, \boldsymbol{\lambda}^k + \alpha_k \frac{|T^*| - L^*}{||\nabla \boldsymbol{\lambda}^k||^2} \nabla \boldsymbol{\lambda}^k\}, \tag{7}$$

where $T^*$ is the smallest feasible tagSNP set found so far (*i.e.,* the best upper bound for *max* $L(t)$), $L^*$ is the largest of *max* $L(t)$ found so far (*i.e.*, the best lower bound for *max* $L(t)$), and $\{\alpha_0, \alpha_1, \alpha_1, \ldots\}$ is a decreasing sequence of predefined scalars.

The pseudo-code for the Lagrangian relaxation algorithm, *LRTag*, is given in Algorithm 2. In the algorithm, we started from an initial setting of $\boldsymbol{\lambda}^0$, generated a solution to $t^0$, and extended it to a valid tagSNP set as above mentioned. Then we updated $\boldsymbol{\lambda}^0$ into $\boldsymbol{\lambda}^1$ according to formula 7. We repeated the process until we were not able to improve $\boldsymbol{\lambda}$ or a predefined number of maximum iterations was reached. Over the entire iterative process, the smallest feasible set of tagSNPs found by LRTag would be output as a solution to the MCTS problem, and the largest $L(t)$ would be a lower bound for tagSNP selection, called *LRTag_lb*.

Both of our algorithms calculated lower bounds on the minimum number of required tagSNPs, one of which was found by GreedyTag (*i.e.,* GreedyTag_lb) and the other by LRTag (*i.e.,* LRTag_lb). We define *"gap"* as the difference between the highest lower bound and the cardinality of the smallest tagSNP set found by our algorithms, which will be used to measure the quality of the solutions.

## 4. EXPERIMENTAL RESULTS

In our experiments, we tested the algorithms GreedyTag and LRTag on the HapMap populations and compared their performance and efficiency with single-population tagging programs LD-Select and FESTA, and a multiple-population tagging program MultiPop-TagSelect. For convenience, we also denoted by GreedyTag the cardinality of a feasible tagSNP set obtained by the GreedyTag algorithm. We used similar notations for LRTag, LD-Select, FESTA, and MultiPop-TagSelect.

Both of our algorithms calculated lower bounds on the minimum number of required tagSNPs, one of which was found by GreedyTag (*i.e.,* GreedyTag_lb) and the other by LRTag (*i.e.,* LRTag_lb). We define *"gap"* as the difference between the highest lower bound and the cardinality of the smallest tagSNP set found by our algorithms, which will be used to measure the quality of the solutions.

We applied all the methods on the entire human genome data involving chromosomes 1 through 22 and on all ENCODE regions (ENm010, ENm013, ENm014, ENr112, ENr113, ENr123, ENr131, ENr213, ENr232, and ENr321) genotyped by the HapMap project (release no. 19, NCBI build 34, October 2005). For the ENCODE data, we estimated the $r^2$ statistics by using a two-marker expectation maximization (EM) algorithm to compute the maximum-likelihood values of the four gamete frequencies, which is also commonly adopted by LD-Select and HaploView (Bakker et al, 2006). For the entire human genome data, we directly download the $r^2$ statistics from the HapMap website (Hapmap LD Data, 2005), generated by HaploView to save computational cost. Note that HaploView only calculates LD for markers up to 250 kbps apart, which is reasonable because the LD for markers that are farther than 250 kbp would normally be very weak anyway, and high LD in such a case can happen only purely by chance. To save running time for dealing with the entire human genome data, we pruned the LD pattern data downloaded from the HapMap website by keeping only entries with $r^2$ greater than or equal to 0.5.

We ran all the programs on a 32-processor SGI Altix 4700 supercomputer system with 1.6 GHz CPU and 64 GB shared memory in the Computer Science Department, University of California, Riverside, CA. Our GreedyTag and LRTag algorithms used up to 15 threads in parallel, while each of the other programs is single threaded.[2]

[2]Note that if a program runs in time of $t$ with 15 threads, then its running time with one thread would be $15t$. This transformation can be used to compare the running times of our programs and those of the other programs on a single-thread mode.

**Algorithm 2** (LRTag: Lagrangian Relaxation Algorithm for TagSNP Selection in Multiple Populations)

**Input**: A set $V$ of $n$ biallelic SNP markers and their pairwise $r^2$ LD statistics in $K$ distinct populations. A pre-defined threshold $\gamma_0$ for $r^2$ LD statistics. A pre-defined initial scalar $\alpha_0$ and threshold $\alpha_{min}$ for subgradient optimization. A pre-defined maximum number $Iter_{max}$ of iterations and a pre-defined threshold $K_{max}$ of maximum trials.
**Output**: A feasible tagSNP set $T \subseteq V$, and a lower bound $LB$.
**Begin**
Partition markers into precincts. Let the set of precincts be $\mathcal{P}$
**For each** precinct $p \in \mathcal{P}$ {*the following will be executed in parallel on a multi-processor machine*}
Let $U$ be the set of SNPs and $W$ be the set of marker occurrences in $p$.

Step 1: *Apply the three data reduction rules and obtain a temporary tagSNP set $T_p$ and a lower bound $LB_p$.*
{The same as the rules in GreedyTag algorithm (see Algorithm 1 for details)}.

Step 2: *Select tagSNPs under a Lagrangian relaxation framework.*
Generate Lagrangian relaxation formula.
$k \Leftarrow 0$; $\alpha \Leftarrow \alpha_0$; $Iter \Leftarrow 0$; Initialize $\boldsymbol{\lambda}$ being an arbitrary non-negative vector; $LB_{p1} \Leftarrow 0$; $T_{p1} \Leftarrow U$;
**While** $(\alpha > \alpha_{min})$ and $(Iter < Iter_{max})$
$Iter \Leftarrow Iter + 1$; $new\_LB \Leftarrow \sum_{1 \le i \le K, 1 \le j \le n} \lambda_{i,j}$; $new\_T \Leftarrow \emptyset$;
{*Calculate a new lower bound new_LB*}
**For each** $v_j \in U$
$s_j \Leftarrow 1 - \sum_{v_{j'}^i \in N^*(v_j)} \lambda_{i,j'}$;
**If** $s_j \le 0$ $t_j \Leftarrow 1$; **Else** $t_j \Leftarrow 0$;
$new\_LB \Leftarrow new\_LB + s_j \cdot t_j$;
{*Obtain a feasible tagSNP set new_T by the reduced cost heuristic (RCH) method*}
**For each** $v_j \in U$ $RCH\_s_j \Leftarrow s_j$;
**For each** $v_j^i \in W$ $RCH\_\lambda_{i,j} \Leftarrow \lambda_{i,j}$;
**For each** $v_j^i \in W$
**If** $\sum_{v_{j'} \in C(v_j^i)} t_{j'} < 1$
$RCH\_s_m \Leftarrow min\{RCH\_s_{j'} : v_{j'} \in C(v_j^i)\}$;
$RCH\_\lambda_{i,j} \Leftarrow RCH\_\lambda_{i,j} + RCH\_s_m$;
**For each** $v_{j'} \in C(v_j^i)$
$RCH\_s_{j'} \Leftarrow RCH\_s_{j'} - RCH\_s_m$;
**If** $RCH\_s_{j'} \le 0$ $t_{j'} \Leftarrow 1$;
**For each** $v_j \in U$
**If** $t_j = 1$ $new\_T \Leftarrow new\_T \cup \{v_j\}$;
{*Update the lower bound $LB_{p1}$ and the tagSNP set $T_{p1}$*}
**If** $new\_LB \le LB_{p1}$ $k \Leftarrow k + 1$;
**If** $(k \ge K_{max})$ $\alpha \Leftarrow \alpha/2$; $k \Leftarrow 0$;
**Else** $LB_{p1} \Leftarrow new\_LB$; $k \Leftarrow 0$;
**If** $|new\_T| < |T_{p1}|$ $T_{p1} \Leftarrow new\_T$;
{*Update the Lagrangian multipliers $\boldsymbol{\lambda}$ by the subgradient optimization method*}
**For each** $v_j^i \in W$ $\nabla\lambda_{i,j} \Leftarrow 1 - \sum_{v_{j'} \in C(v_j^i)} t_{j'}$;
$\boldsymbol{\lambda} \Leftarrow max\{\mathbf{0}, \boldsymbol{\lambda} + \alpha \frac{|T_{p1}| - LB_{p1}}{||\nabla\boldsymbol{\lambda}||^2} \nabla\boldsymbol{\lambda}\}$;
{*Combine the solutions from step 1 and step 2*}
$T_p \Leftarrow T_p \cup T_{p1}$; $LB_p \Leftarrow LB_p + LB_{p1}$;
$T \Leftarrow \bigcup_{p \in \mathcal{P}} T_p$; $LB \Leftarrow \sum_{p \in \mathcal{P}} LB_p$
**Output** $T$, $LB$ {*output the solution and the lower bound*}
**End**

### *4.1. Tagging the ENCODE regions*

A dense set of SNPs across 10 large genomic regions have been produced by the HapMap ENCODE project. These regions serve as the foundation to evaluate the development of methodologies and technologies for detecting functional elements in human DNA. Each region is about 500 kb in length and has an SNP density about 1 SNP per 600 bp.

*4.1.1. Tagging ENCODE regions for a single population.* We tag each HapMap population separately by LD-Select, FESTA, and our new algorithms GreedyTag and LRTag. For illustration purposes,

Table 2. Summary of TagSNPs Identified by FESTA, LD-Select, GreedyTag, and LRTag for a Single Population, CEU, on All ENCODE Regions

| *Region* | *ENm010* | *ENm013* | *ENm014* | *ENr112* | *ENr113* | *ENr123* | *ENr131* | *ENr213* | *ENr232* | *ENr321* |
|---|---|---|---|---|---|---|---|---|---|---|
| No. of SNP | 525 | 692 | 904 | 947 | 1080 | 864 | 990 | 612 | 457 | 544 |
| $r^2 \geq 0.5$ | | | | | | | | | | |
| No. of precinct | 39 | 27 | 47 | 52 | 40 | 30 | 83 | 42 | 64 | 52 |
| No. of tagSNP (upper bound) | | | | | | | | | | |
| LD-Select | 62 | 38 | 65 | 84 | 77 | 69 | 112 | 62 | 72 | 68 |
| FESTA | 57 | 35 | 63 | 76 | 73 | 65 | 107 | 61 | 70 | 65 |
| GreedyTag | 56 | 35 | 63 | 76 | 73 | 62 | 107 | 61 | 70 | 64 |
| LRTag | 56 | 35 | 63 | 76 | 73 | 62 | 107 | 61 | 70 | 64 |
| No. of tagSNP (lower bound) | | | | | | | | | | |
| LRTag_lb | 55 | 35 | 63 | 76 | 73 | 62 | 107 | 60 | 70 | 64 |
| GreedyTag_lb | 50 | 33 | 63 | 72 | 69 | 54 | 101 | 55 | 70 | 62 |
| Gap | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| $r^2 \geq 0.8$ | | | | | | | | | | |
| No. of precinct | 116 | 69 | 121 | 139 | 131 | 129 | 175 | 105 | 106 | 107 |
| No. of tagSNP (upper bound) | | | | | | | | | | |
| LD-Select | 123 | 82 | 129 | 152 | 146 | 139 | 189 | 110 | 115 | 109 |
| FESTA | 122 | 79 | 129 | 152 | 143 | 139 | 186 | 110 | 114 | 109 |
| GreedyTag | 122 | 79 | 129 | 152 | 143 | 139 | 186 | 110 | 114 | 109 |
| LRTag | 122 | 79 | 129 | 152 | 143 | 139 | 186 | 110 | 114 | 109 |
| No. of tagSNP (lower bound) | | | | | | | | | | |
| GreedyTag_lb | 122 | 79 | 129 | 152 | 143 | 139 | 186 | 110 | 114 | 109 |
| LRTag_lb | 122 | 79 | 129 | 150 | 143 | 139 | 186 | 107 | 110 | 109 |
| Gap | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

we only showed the results for tagging the CEU population and compared the performance of the above algorithms in Table 2.

When the $r^2$ threshold is set as 0.5, the number of tagSNPs selected by our algorithm is on the average 9.3% of the total number of markers (the actual percentage number ranges from 5.1% to 15.3%). With a more stringent $r^2$ threshold of 0.8, the average number of tagSNPs rises to 17.6% of the total number of markers (ranging from 11.4% to 24.9%). The same trend was observed when applying our algorithms on the other populations (results are not shown because of space constraint).

On each ENCODE region, we observed that the gap between LRTag_lb and LRTag is at most one with the $r^2$ threshold being 0.5, and there is no gap when the $r^2$ threshold is set as 0.8. This demonstrates that our algorithm LRTag found near-optimal solutions in all test cases. In general, LRTag and GreedyTag always generated the smallest sets of tagSNPs, FESTA selected at most three more tagSNP, and LD-Select might select up to eight more tagSNPs.

As our algorithms and FESTA are all near-optimal, we compared the time efficiency of these programs in Table 3. Because LD-Select takes genotype data as input and the other programs take pairwise LD data as input, we do not compare LD-Select's running times directly with those of the others here (generally speaking, it takes LD-Select from 30 m in to 2 h on an ENCODE region). From Table 3, we can see that the running time of FESTA varied widely from 1 s to 64 h on different regions, while our algorithms GreedyTag and LRTag consistently took 1–2 s on all regions. In conclusion, our algorithms were 3–4 orders of magnitude faster than FESTA in most of the cases, and our algorithms found slightly smaller sets of tagSNPs

*4.1.2. Tagging ENCODE regions for multiple populations.* We tagged each and the entire ENCODE regions for all four HapMap populations by MultiPop-TagSelect, GreedyTag, and LRTag. The tagging results of these methods on each ENCODE region are summarized in Table 4. We also highlighted the results for region ENm013 and for the entire ENCODE region in Figure 1.

With the $r^2$ threshold set as 0.5, the number of tagSNPs selected by our algorithms is on the average 18.3% of the total number of markers (the actual percentage number ranges from 11.0% to 34.5%). With a more stringent $r^2$ threshold of 0.8, the average number of tagSNPs increases to 33.7% (ranging from 24.0%

TABLE 3. THE TIME EFFICIENCY OF FESTA, GREEDYTAG, AND LRTAG FOR SELECTING TAGSNPS FROM A SINGLE POPULATION, CEU, ON ALL ENCODE REGIONS

| *Region* | *ENm010* | *ENm013* | *ENm014* | *ENr112* | *ENr113* | *ENr123* | *ENr131* | *ENr213* | *ENr232* | *ENr321* |
|---|---|---|---|---|---|---|---|---|---|---|
| $r^2 \geq 0.5$ | | | | | | | | | | |
| FESTA | 3 h, 14 min | 3 h, 16 min | 4 h, 51 min | 3 h, 18 min | 14 h, 24 min | 64 h, 4 min | 5 h, 13 min | 2 h, 24 min | 1 h, 38 min | 45 min, 19 s |
| GreedyTag | 1 s | 1 s | 1 s | 1 s | 1 s | 1 s | 1 s | 1 s | 1 s | 1 s |
| LRTag | 1 s | 1 s | 1 s | 2 s | 1 s | 2 s | 1 s | 1 s | 1 s | 1 s |
| $r^2 \geq 0.8$ | | | | | | | | | | |
| FESTA | 1 s | 3 s | 28 min, 20 s | 44 min, 6 s | 12 min, 8 s | 50 min, 16 s | 3 s | 1 s | 2 s | 1 s |
| GreedyTag | <1 s | <1 s | <1 s | <1 s | <1 s | <1 s | <1 s | <1 s | <1 s | <1 s |
| LRTag | <1 s | <1 s | <1 s | <1 s | <1 s | <1 s | <1 s | <1 s | <1 s | <1 s |

The running times are obtained on a 32-processor SGI Altix 4700 supercomputer system.

to 50.5%). We observed that LRTag always performs the best in these tests, followed by the GreedyTag algorithm, and MultiPop-TagSelect always performs worst. When $r^2$ threshold is set as 0.5, LRTag requires 16.4% fewer markers on the average than MultiPop-TagSelect. When the $r^2$ threshold is 0.8, LRTag usually requires 5.1% fewer markers on the average than MultiPop-TagSelect.

The gap between LRTag_lb and LRTag is at most two for each ENCODE region and totally six for all ENCODE regions with the $r^2$ threshold being 0.5. There is no gap with the $r^2$ threshold being 0.8.

### 4.2. Genome-wide tagging

Because both LD-Select and MultiPop-TagSelect (written in Perl) took more than 20 h to tag a single chromosome, we reimplemented their algorithms in C++, called *LD-Select** and *MultiPop-TagSelect**, respectively, to give a fair comparison of the programs. As FESTA's "greedy-exhaustive hybrid search" is very computational demanding and hard to emulate, we excluded FESTA from the following comparative study.

TABLE 4. SUMMARY OF TAGSNPS IDENTIFIED BY MULTIPOP-TAGSELECT, GREEDYTAG, AND LRTAG FOR ALL HAPMAP POPULATIONS ON ENCODE REGIONS

| *Region* | *ENm010* | *ENm013* | *ENm014* | *ENr112* | *ENr113* | *ENr123* | *ENr131* | *ENr213* | *ENr232* | *ENr321* |
|---|---|---|---|---|---|---|---|---|---|---|
| No. of SNP | 783 | 1063 | 1261 | 1158 | 1485 | 1221 | 1186 | 900 | 777 | 1025 |
| $r^2 \geq 0.5$ | | | | | | | | | | |
| No. of precinct | 65 | 38 | 48 | 44 | 67 | 38 | 73 | 56 | 126 | 53 |
| No. of tagSNP (upper bound) | | | | | | | | | | |
| MultiPop-TagSelect | 206 | 150 | 201 | 238 | 260 | 181 | 306 | 200 | 286 | 233 |
| GreedyTag | 179 | 117 | 164 | 184 | 228 | 141 | 257 | 173 | 268 | 193 |
| LRTag | 179 | 117 | 164 | 184 | 228 | 141 | 257 | 173 | 268 | 193 |
| No. of tagSNP (lower bound) | | | | | | | | | | |
| LRTag_lb | 178 | 117 | 162 | 182 | 226 | 141 | 256 | 173 | 268 | 193 |
| GreedyTag_lb | 169 | 107 | 149 | 168 | 218 | 139 | 250 | 173 | 264 | 189 |
| Gap | 1 | 0 | 2 | 2 | 0 | 0 | 1 | 0 | 0 | 0 |
| $r^2 \geq 0.8$ | | | | | | | | | | |
| No. of precinct | 156 | 111 | 146 | 101 | 209 | 106 | 194 | 170 | 210 | 191 |
| No. of tagSNP (upper bound) | | | | | | | | | | |
| MultiPop-TagSelect | 338 | 275 | 321 | 425 | 454 | 329 | 462 | 324 | 402 | 396 |
| GreedyTag | 322 | 255 | 305 | 391 | 437 | 300 | 445 | 318 | 392 | 377 |
| LRTag | 322 | 255 | 305 | 389 | 437 | 300 | 445 | 318 | 392 | 377 |
| No. of tagSNP (lower bound) | | | | | | | | | | |
| LRTag_lb | 322 | 255 | 305 | 389 | 437 | 300 | 445 | 318 | 392 | 377 |
| GreedyTag_lb | 319 | 253 | 303 | 374 | 435 | 300 | 443 | 318 | 392 | 377 |
| Gap | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

TABLE 5. SUMMARY OF THE TAG SNPS SELECTED BY LD-SELECT, GREEDYTAG, AND LRTAG FOR A SINGLE POPULATION, CEU, ON EACH HUMAN CHROMOSOME

| | *1*[a] | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* | *11* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No. of SNP | 151,195 | 181,499 | 143,472 | 130,823 | 138,817 | 149,514 | 113,037 | 122,646 | 100,352 | 110,942 | 104,661 |
| $r^2 \geq 0.5$ | | | | | | | | | | | |
| No. of precinct | 15,752 | 29,426 | 12,901 | 11,906 | 11,998 | 11,831 | 10,512 | 9,900 | 9,438 | 10,153 | 9,979 |
| No. of tagSNP (upper bound) | | | | | | | | | | | |
| LD-Select* | 21,865 | 36,238 | 19,063 | 17,212 | 17,765 | 17,921 | 15,418 | 15,140 | 13,800 | 14,882 | 14,307 |
| GreedyTag | 20,806 | 35,083 | 17,984 | 16,295 | 16,769 | 16,815 | 14,584 | 14,203 | 13,066 | 14,041 | 13,600 |
| LRTag | 20,800 | 35,065 | 17,977 | 16,286 | 16,756 | 16,798 | 14,577 | 14,196 | 13,058 | 14,038 | 13,589 |
| No. of tagSNP (lower bound) | | | | | | | | | | | |
| LRTag-lb | 20,793 | 35,059 | 17,958 | 16,279 | 16,736 | 16,784 | 14,569 | 14,182 | 13,049 | 14,031 | 13,578 |
| GreedyTag-lb | 20,123 | 34,202 | 17,155 | 15,675 | 15,965 | 16,086 | 14,021 | 13,568 | 12,530 | 13,477 | 13,089 |
| Gap | 7 | 6 | 19 | 7 | 20 | 14 | 8 | 14 | 9 | 7 | 11 |
| $r^2 \geq 0.8$ | | | | | | | | | | | |
| No.of precinct | 35,990 | 51,098 | 31,916 | 28,650 | 29,931 | 30,632 | 26,181 | 26,120 | 23,739 | 25,186 | 23,544 |
| No. of tagSNP (upper bound) | | | | | | | | | | | |
| LD-Select* | 38,944 | 54,612 | 35,092 | 31,590 | 32,978 | 33,723 | 28,754 | 28,822 | 26,008 | 27,698 | 25,826 |
| GreedyTag | 38,534 | 54,081 | 34,602 | 31,124 | 32,502 | 33,229 | 28,362 | 28,394 | 25,666 | 27,302 | 25,485 |
| LRTag | 38,534 | 54,080 | 34,601 | 31,123 | 32,501 | 33,227 | 28,362 | 28,393 | 25,665 | 27,302 | 25,484 |
| No. of tagSNP (lower bound) | | | | | | | | | | | |
| LRTag-lb | 38,534 | 54,080 | 34,600 | 31,123 | 32,501 | 33,225 | 28,361 | 28,391 | 25,664 | 27,301 | 25,483 |
| GreedyTag-lb | 38,269 | 53,687 | 34,310 | 30,824 | 32,189 | 32,962 | 28,083 | 28,110 | 25,396 | 27,077 | 25,276 |
| Gap | 0 | 0 | 1 | 0 | 0 | 2 | 1 | 2 | 1 | 1 | 1 |

| | *12*[a] | *13* | *14* | *15* | *16* | *17* | *18* | *19* | *20* | *21* | *22* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No. of SNP | 100,437 | 84,184 | 68,485 | 58,491 | 57,083 | 47,505 | 62,666 | 29,341 | 51,206 | 27,955 | 26,996 |
| $r^2 \geq 0.5$ | | | | | | | | | | | |
| No. of precinct | 9,960 | 7,476 | 6,751 | 6,740 | 7,184 | 6,764 | 6,534 | 5,291 | 5,874 | 3,270 | 3,829 |
| No. of tagSNP (upper bound) | | | | | | | | | | | |
| LD-Select* | 14,243 | 10,996 | 9,703 | 9,364 | 9,962 | 8,656 | 9,115 | 6,464 | 7,972 | 4,470 | 5,029 |
| GreedyTag | 13,554 | 10,374 | 9,215 | 8,930 | 9,503 | 8,355 | 8,652 | 6,286 | 7,637 | 4,258 | 4,831 |
| LRTag | 13,548 | 10,370 | 9,212 | 8,923 | 9,500 | 8,354 | 8,649 | 6,284 | 7,634 | 4,257 | 4,831 |
| No. of tagSNP (lower bound) | | | | | | | | | | | |
| LRTag-lb | 13,539 | 10,363 | 9,203 | 8,920 | 9,500 | 8,353 | 8,646 | 6,283 | 7,630 | 4,256 | 4,830 |
| GreedyTag-lb | 13,048 | 9,988 | 8,867 | 8,589 | 9,241 | 8,180 | 8,332 | 6,190 | 7,380 | 4,125 | 4,714 |
| Gap | 9 | 7 | 9 | 3 | 0 | 1 | 3 | 1 | 4 | 1 | 1 |
| $r^2 \geq 0.8$ | | | | | | | | | | | |
| No. of tagSNP (upper bound) | | | | | | | | | | | |
| No. of precinct | 23,809 | 18,509 | 16,391 | 15,629 | 16,869 | 13,942 | 15,262 | 10,019 | 13,177 | 7,390 | 8,240 |
| LD-Select* | 25,887 | 20,221 | 17,723 | 16,908 | 18,194 | 14,778 | 16,498 | 10,494 | 14,194 | 7,912 | 8,727 |
| GreedyTag | 25,579 | 19,967 | 17,546 | 16,722 | 18,012 | 14,670 | 16,299 | 10,420 | 14,052 | 7,844 | 8,652 |
| LRTag | 25,579 | 19,967 | 17,545 | 16,722 | 18,012 | 14,669 | 16,299 | 10,420 | 14,052 | 7,844 | 8,652 |
| No. of tagSNP (lower bound) | | | | | | | | | | | |
| LRTag-lb | 25,578 | 19,966 | 17,545 | 16,722 | 18,012 | 14,668 | 16,299 | 10,420 | 14,051 | 7,844 | 8,652 |
| GreedyTag-lb | 25,387 | 19,778 | 17,405 | 16,608 | 17,836 | 14,588 | 16,181 | 10,382 | 13,943 | 7,774 | 8,601 |
| Gap | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |

[a]The numbers 1–11 and 12–22 denote chromosome numbers.

*4.2.1. Tagging the human genome for a single population.* We applied LD-Select*, GreedyTag, and LRTag on each HapMap population separately. For illustration purposes, we only discussed the results for tagging the CEU population and compared the performance of the above three algorithms. The details can be found in Table 5.

With the $r^2$ threshold set as 0.5, the number of tagSNPs selected by our algorithms is 14.4% of the total number of markers on the average (the actual percentage ranges from 11.2% to 21.4%). With a more stringent $r^2$ threshold of 0.8, the average number of tagSNPs increases to 26.6% (ranging from 22.2% to

TABLE 6. SUMMARY OF THE TAG SNPS SELECTED BY MULTIPOP-TAGSELECT, GREEDYTAG, AND LRTAG FOR ALL HAPMAP POPULATIONS ON EACH HUMAN CHROMOSOME

| | *1*[a] | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* | *11* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No. of SNP | 216,357 | 249,136 | 196,535 | 182,273 | 187,924 | 205,496 | 155,224 | 170,136 | 138,047 | 156,089 | 144,083 |
| $r^2 \geq 0.5$ | | | | | | | | | | | |
| No. of precinct | 16,234 | 26,836 | 12,835 | 12,251 | 12,414 | 11,862 | 10,332 | 10,101 | 9,254 | 10,220 | 9,568 |
| No. of tagSNP (upper bound) | | | | | | | | | | | |
| MultiPop-TagSelect* | 64,892 | 126,408 | 56,978 | 52,828 | 54,087 | 54,454 | 45,943 | 46,927 | 41,341 | 45,226 | 41,556 |
| GreedyTag | 59,126 | 122,372 | 51,266 | 47,650 | 48,661 | 48,817 | 41,169 | 42,206 | 37,289 | 40,713 | 37,365 |
| LRTag | 55,016 | 117,537 | 47,450 | 44,223 | 45,186 | 44,987 | 38,150 | 39,149 | 34,439 | 37,554 | 34,590 |
| No. of tagSNP (lower bound) | | | | | | | | | | | |
| LRTag-lb | 54,942 | 117,511 | 47,362 | 44,141 | 45,102 | 44,878 | 38,090 | 39,076 | 34,381 | 37,486 | 34,537 |
| GreedyTag-lb | 53,937 | 117,155 | 46,330 | 43,239 | 44,145 | 43,845 | 37,280 | 38,161 | 33,534 | 36,713 | 33,778 |
| Gap | 74 | 26 | 88 | 82 | 84 | 109 | 60 | 73 | 58 | 68 | 53 |
| $r^2 \geq 0.8$ | | | | | | | | | | | |
| No. of precinct | 42,450 | 56,135 | 35,192 | 33,434 | 33,211 | 33,228 | 28,543 | 28,948 | 25,485 | 28,277 | 25,428 |
| No. of tagSNP (upper bound) | | | | | | | | | | | |
| MultiPop-TagSelect* | 100,062 | 155,505 | 89,195 | 82,835 | 84,998 | 86,313 | 72,024 | 74,934 | 65,442 | 70,817 | 64,679 |
| GreedyTag | 94,797 | 150,664 | 84,091 | 78,077 | 80,188 | 80,981 | 67,818 | 70,678 | 61,708 | 66,676 | 60,721 |
| LRTag | 94,797 | 150,664 | 84,090 | 78,076 | 80,186 | 80,980 | 67,817 | 70,677 | 61,706 | 66,674 | 60,718 |
| No. of tagSNP (lower bound) | | | | | | | | | | | |
| LRTag-lb | 94,788 | 150,660 | 84,079 | 78,072 | 80,174 | 80,964 | 67,808 | 70,667 | 61,699 | 66,663 | 60,705 |
| GreedyTag-lb | 94,362 | 150,393 | 83,585 | 77,674 | 79,663 | 80,507 | 67,461 | 70,285 | 61,321 | 66,291 | 60,294 |
| Gap | 9 | 4 | 11 | 4 | 12 | 16 | 9 | 10 | 7 | 11 | 13 |

| | *12*[a] | *13* | *14* | *15* | *16* | *17* | *18* | *19* | *20* | *21* | *22* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| No. of SNP | 141,943 | 119,080 | 94,528 | 81,687 | 79,898 | 64,645 | 89,024 | 40,549 | 70,877 | 39,400 | 39,523 |
| $r^2 \geq 0.5$ | | | | | | | | | | | |
| No. of precinct | 10,086 | 7,810 | 6,532 | 6,667 | 7,328 | 6,952 | 6,875 | 5,127 | 6,139 | 3,290 | 3,884 |
| No. of tagSNP (upper bound) | | | | | | | | | | | |
| MultiPop-TagSelect* | 42,362 | 33,477 | 28,465 | 27,847 | 28,987 | 23,601 | 27,109 | 15,768 | 23,243 | 13,100 | 13,895 |
| GreedyTag | 38,563 | 30,183 | 25,706 | 25,408 | 26,432 | 21,931 | 24,789 | 14,785 | 21,319 | 12,010 | 12,980 |
| LRTag | 35,493 | 27,927 | 23,932 | 23,721 | 24,791 | 20,647 | 23,174 | 14,007 | 19,994 | 11,253 | 12,174 |
| No. of tagSNP (lower bound) | | | | | | | | | | | |
| LRTag-lb | 35,449 | 27,881 | 23,903 | 23,686 | 24,761 | 20,636 | 23,141 | 14,001 | 19,971 | 11,238 | 12,160 |
| GreedyTag lb | 34,833 | 27,306 | 23,440 | 23,229 | 24,294 | 20,366 | 22,697 | 13,814 | 19,601 | 11,035 | 12,017 |
| Gap | 44 | 46 | 29 | 35 | 30 | 11 | 33 | 6 | 23 | 15 | 14 |
| $r^2 \geq 0.8$ | | | | | | | | | | | |
| No. of precinct | 27,027 | 21,084 | 17,723 | 17,526 | 18,943 | 16,278 | 17,866 | 11,289 | 15,438 | 8,366 | 9,480 |
| No. of tagSNP (upper bound) | | | | | | | | | | | |
| MultiPop-TagSelect* | 65,521 | 52,863 | 44,226 | 42,380 | 43,913 | 34,289 | 41,893 | 22,274 | 35,251 | 19,990 | 20,624 |
| GreedyTag-lb | 61,828 | 49,797 | 41,867 | 40,250 | 41,726 | 32,862 | 39,833 | 21,465 | 33,676 | 19,060 | 19,742 |
| LRTag-lb | 61,826 | 49,796 | 41,867 | 40,250 | 41,724 | 32,862 | 39,833 | 21,464 | 33,675 | 19,060 | 19,741 |
| No. of tagSNP (lower bound) | | | | | | | | | | | |
| LRTag-lb | 61,816 | 49,791 | 41,860 | 40,247 | 41,721 | 32,860 | 39,832 | 21,464 | 33,673 | 19,059 | 19,739 |
| GreedyTag lb | 61,450 | 49,498 | 41,642 | 40,029 | 41,497 | 32,740 | 39,625 | 21,377 | 33,525 | 18,996 | 19,660 |
| Gap | 10 | 5 | 7 | 3 | 3 | 2 | 1 | 0 | 2 | 1 | 2 |

[a]The numbers 1–11 and 12–22 denote chromosome numbers.

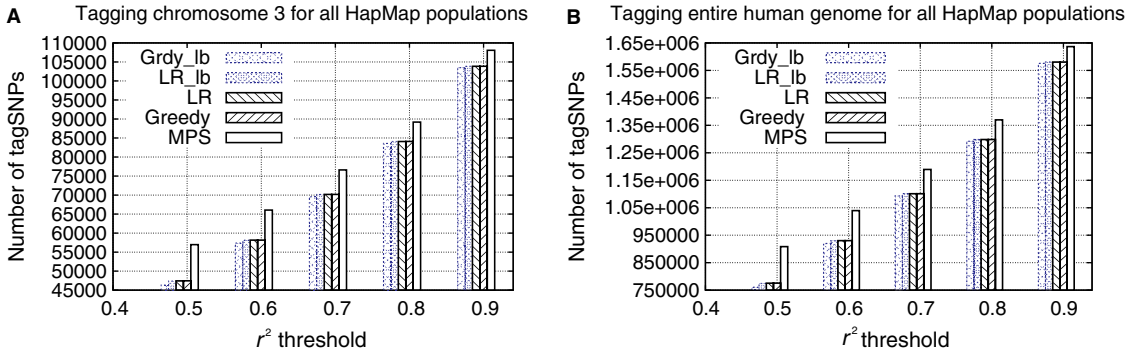


**FIG. 3.** (**A**) Tagging chromosome 3 for all HapMap populations with 196,535 markers. (**B**) Tagging the entire human genome for all HapMap populations with 2,862,454 markers. See the caption of Figure 1 for the definitions of the algorithm names.

35.5%). Similar trends were observed when applying our algorithms to the other populations (the results are not shown because of space limitation).

We observed that LRTag always performs the best, followed by the GreedyTag algorithm, and LD-Select* always performs the worst. With the $r^2$ threshold set as 0.5, LRTag usually requires 4.9% fewer tagSNPs (the actual percentage number ranges from 2.8% to 6.3%) on average than LD-Select* on each chromosome. When the $r^2$ threshold is increased to 0.8, LRTag usually requires 1.2% fewer tagSNPs (ranging from 0.07% to 1.5%) on average than LD-Select*.

We can see that, on each chromosome, the gap between the lower bound from LRTag_lb and the upper bound obtained by LRTag is on the average 7 (the actual number ranges from 1 to 20) with the $r^2$ threshold set as 0.5, and less than 1 (ranging from 0 to 2) with the $r^2$ threshold set being 0.8. This demonstrates that LRTag finds near-optimal solutions in all test cases even for genome-wide tagging on a single population. In fact, the performance of GreedyTag is not bad either.

*4.2.2. Tagging the human genome for multiple populations.* Finally, we tagged the entire human genome for all four HapMap populations by MultiPop-TagSelect, GreedyTag, and LRTag. We summarized the tagging results of these methods on each chromosome in Table 6, and then highlighted the results for chromosome 3 and all chromosomes in Figure 3.

With the $r^2$ threshold set as 0.5, the number of tagSNPs selected by our methods is on the average 27.3% of the total number of markers (the actual percentage ranges from 21.9% to 47.2%). With a more stringent $r^2$ threshold of 0.8, the average number of tagSNPs increases to 46.0% (ranging from 29.4% to 60.4%). Based on Table 6, we observed that LRTag always performs slightly better than GreedyTag and significantly better than MultiPop-TagSelect*. With the $r^2$ threshold set as 0.5, LRTag requires 6.8% fewer tagSNPs on average (the actual number ranges from 4.0% to 8.0%) than MultiPop-TagSelect* on each chromosome. With the $r^2$ threshold set as 0.8, LRTag requires 3.6% fewer markers on average (ranging from 2.7% to 4.3%) than MultiPop-TagSelect*.

The gap between the lower bound from LRTag_lb and upper bound of LRTag is on the average 48 for each chromosome (the actual number ranges from 6 to 109) with the $r^2$ threshold set as 0.5, and 6.5 (ranging from 0 to 16) with the $r^2$ threshold set being 0.8, as shown in Table 6.

TABLE 7. THE SPEEDS OF GREEDYTAG AND LRTAG FOR TAGGING THE ENTIRE ENCODE REGION FOR ALL HAPMAP POPULATIONS WITH THE $r^2$ THRESHOLD BEING 0.5

| *Region* | *ENm010* | *ENm013* | *ENm014* | *ENr112* | *ENr113* | *ENr123* | *ENr131* | *ENr213* | *ENr232* | *ENr321* |
|---|---|---|---|---|---|---|---|---|---|---|
| LRTag | 1 s | 4 s | 3 s | 5 s | 7 s | 5 s | 1 s | 1 s | 1 s | 2 s |
| GreedyTag | 1 s | 4 s | 4 s | 5 s | 7 s | 6 s | 1 s | 1 s | 1 s | 2 s |

The running time was evaluated on a 32-processor SGI Altix 4700 supercomputer system.

Table 8. The Speeds of GreedyTag and LRTag for Tagging the Human Genome for a Single Population, CEU, with the $r^2$ Threshold Being 0.5

| | *1*[a] | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* | *11* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LRTag | 1 min, 18 s | 1 min, 44 s | 1 min, 28 s | 1 min, 12 s | 1 min, 27 s | 3 min, 7 s | 1 min, 3 s | 1 min, 15 s | 57 s | 1 min, 6 s | 1 min, 8 s |
| GreedyTag | 1 min, 17 s | 1 min, 41 s | 1 min, 16 s | 1 min, 15 s | 1 min, 24 s | 3 min, 11 s | 58 s | 1 min, 16 s | 57 s | 1 min, 6 s | 1 min, 10 s |
| | *12*[a] | *13* | *14* | *15* | *16* | *17* | *18* | *19* | *20* | *21* | *22* |
| LRTag | 56 s | 50 s | 34 s | 28 s | 23 s | 46 s | 31 s | 9 s | 23 s | 11 s | 10 s |
| GreedyTag | 56 s | 50 s | 37 s | 27 s | 20 s | 47 s | 31 s | 10 s | 22 s | 11 s | 9 s |

The running time was evaluated on a 32-processor SGI Altix 4700 supercomputer system.
[a]The numbers 1–11 and 12–22 denote chromosome numbers.

## 5. CONCLUSION

Our LRTag and GreedyTag algorithms run quickly on ENCODE regions and the entire human genome for both single and multiple populations. On an ENCODE region with the $r^2$ threshold being 0.5, it takes our algorithms no more than 2 s to tag a single population (as shown in Table 3) and less than 7 s to tag multiple populations (as displayed in Table 7). On a human chromosome, it takes no more than 4 min to tag a single population (as shown in Table 8) and less than 12 min to tag on multiple populations (as displayed in Table 9). For $r^2$ thresholds greater than 0.5, our algorithms run faster. Hence, for any given $r^2$ threshold, it takes our algorithms less than a minute to tag the entire ENCODE region and less than an hour to tag the entire human genome.

If the number of populations of interest increases, the genotyping density increases or the $r^2$ threshold increases, the number of required tagSNPs also increases. For example, on multiple HapMap populations with the $r^2$ threshold being 0.5, we need to tag one SNP for about every six SNPs on the densely genotyped ENCODE regions. We need to tag one SNP for about every four SNPs on sparsely genotyped HapMap chromosomes.

All the lower and upper bounds produced by the discussed methods are shown in Figures 2 and 3. In the figures, we tagged the ENCODE regions and human genome on the HapMap populations with the $r^2$ thresholds being 0.5, 0.6, 0.7, and 0.8 separately. From all these test cases, we observed that LRTag always chooses the smallest set of tagSNPs, closely followed by GreedyTag, whereas MultiPop-TagSelect chooses the largest set of tagSNPs.

LRTag_lb always provides the best lower bound and LRTag the best upper bound among all methods considered. The simple greedy algorithm, GreedyTag, chooses slightly more tagSNPs than LRTag and the

Table 9. The Speeds of GreedyTag and LRTag for Tagging the Entire Human Genome for All HapMap Populations with the $r^2$ Threshold being 0.5

| | *1*[a] | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* | *11* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LRTag | 3 min, 4 s | 2 min, 2 s | 3 min, 9 s | 2 min, 51 s | 3 min, 37 s | 11 min, 4 s | 2 min, 12 s | 3 min, 45 s | 2 min, 24 s | 2 min, 49 s | 2 min, 20 s |
| GreedyTag | 3 min, 11 s | 1 min, 13 s | 3 min, 43 s | 2 min, 46 s | 3 min, 20 s | 10 min, 45 s | 2 min, 25 s | 2 min, 52 s | 2 min, 18 s | 2 min, 55 s | 2 min, 16 s |
| | *12*[a] | *13* | *14* | *15* | *16* | *17* | *18* | *19* | *20* | *21* | *22* |
| LRTag | 2 min, 55 s | 2 min, 11 s | 1 min, 28 s | 1 min | 48 s | 1 min, 10 s | 1 min, 17 s | 27 s | 56 s | 25 s | 30 s |
| GreedyTag | 3 min | 2 min, 27 s | 1 min, 16 s | 52 s | 23 s | 1 min, 9 s | 1 min, 16 s | 27 s | 50 s | 25 s | 30 s |

The running time was evaluated on a 32-processor SGI Altix 4700 supercomputer system.
[a]The numbers 1–11 and 12–22 denote chromosome numbers.

lower bound GreedyTag_lb is slightly lower than LRTag_lb, which indicates that the data reduction rules in Section 3 are very powerful.

When the $r^2$ threshold increases, the size of the precincts decreases. Consequently, the gap between the lower bound and the upper bound decreases. For the entire human genome with 2,862,454 markers, the gap between LRTag and LRTag_lb is 1061 when the $r^2$ threshold is 0.5, and 142 when the $r^2$ threshold increases to 0.8. The small gap shows that LRTag finds near-optimal solutions for genome-wide tagging.

## ACKNOWLEDGMENTS

We are grateful to the anonymous referee for several constructive comments. This research was supported in part by NSF grants IIS-0711129 and DBI-0321756, NIH grant 2R01LM008991, NSFC grant 60528001, NSF CAREER Award IIS-0447773, and a Changjiang Visiting Professorship at Tsinghua University. The programs (GreedyTag and LRTag) are for free to the public and are available upon request.*

## DISCLOSURE STATEMENT

No competing financial interests exist.

## REFERENCES

Avi-Itzhak, H., Su, X., and De La Vega, F.M. 2003. Selection of minimum subsets of single nucleotide polymorphisms to capture haplotype block diversity. *Proc. Pac. Symp. Biocomput.* 466–477.

Bakker, P., Burtt, N.P., Graham, R.R., *et al.* 2006. Transferability of tag SNPs in genetic association studies in multiple populations. *Nat. Genet.* 38, 1298–1303.

Balas, E., and Carrera, M.C. 1996. A dynamic subgradient-based branch-and-bound procedure for set covering. *Oper. Res.* 44, 875–890.

Bar-Yehuda, R., and Moran, S. 1984. On approximation problems related to the independent set and vertex cover problems. *Discov. Appl. Math.* 9, 1–10.

Carlson, C.S., Eberle, M.A., Rieder, M.J., *et al.* 2004. Selecting a maximally informative set of singlenucleotide polymorphisms for association analyses using linkage disequilibrium. *Am. J. Hum. Genet.* 74, 106–120.

Conrad, D.F., Jakobsson, M., Coop, G., *et al.* 2006. A worldwide survey of haplotype variation and linkage disequilibrium in the human genome *Nat. Genet.* 38, 1251–1260.

Ding, K., Zhou, K., Zhang, J., *et al.* 2005. The effect of haplotype-block definitions on inference of haplotype-block structure and htSNPs selection. *Mol. Biol. Evol.* 22, 148–159.

Gabriel, S.B., Schaffner, S.F., Nguyen, H., *et al.* 2002. The structure of haplotype blocks in the human genome. *Science* 296, 2225–2229.

Halperin, E., Kimmel, G., and Shamir, R. 2005. Tag SNP selection in genotype data for maximizing SNP prediction accuracy. *Bioinformatics* 21(Suppl 1), i195–i203.

Hampe, J., Schreiber, S., and Krawczak, M. 2003. Entropy-based SNP selection for genetic association studies. *Hum. Genet.* 114, 36–43.

HapMap LD data. 2005. Available at: www.hapmap.ncbi.nlm.nih.gov/downloads/ld_data/2005–10/.

Hinds, D.A., Stuve, L.L., Nilsen, G.B., *et al.* 2005. Whole-genome patterns of common DNA variation in three human populations. *Science* 307, 1072–1079.

Howie, B.N., Carlson, C.S., Rieder, M.J. *et al.* 2006. Efficient selection of tagging single-nucleotide polymorphisms in multiple populations. *Hum. Genet.* 120, 58–68.

International HapMap Consortium. 2005. A haplotype map of the human genome. *Nature* 437, 1299–1320.

Johnson, G.C., Esposito, L., Barratt, B.J., *et al.* 2001. Haplotype tagging for the identification of common disease genes. *Nat. Genet.* 29, 233–237.

Ke, X., and Cardon, L.R. 2003. Efficient selective screening of haplotype tag SNPs. *Bioinformatics* 19, 287–288.

---

*An extended abstract of this article, entitled "Efficient Algorithms for Genome-Wide TagSNP Selection Across Populations via the Linkage Disequilibrium Criterion," has appeared in the Proceedings of Computational Systems Bioinformatics (CSB'2007), August 2007, San Diego, CA.

Kruglyak, L., and Nickerson, D. 2001. Variation is the spice of life. *Nat. Genet.* 27, 234–236.

Lin, Z., and Altman, R.B. 2004. Finding haplotype tagging SNPs by use of principal components analysis. *Am. J. Hum. Genet.* 75, 850–861.

Liu, Z., Lin, S., and Tan, M. 2006. Genome-wide tagging SNPs with entropy-based Monte Carlo method. *J. Comput. Biol.* 13, 1606–1614.

Magi, R., Kaplinski, L., and Remm, M. 2006. The whole genome tagSNP selection and transferability among HapMap populations. *Pac. Symp. Biocomput.* 11, 535–543.

Need, A.C., and Goldstein, D.B. 2006. Genome-wide tagging for everyone. *Nat. Genet.* 38, 1227–1228.

Patil, N., Berno, A.J., Hinds, D.A., *et al.* 2001. Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. *Science* 294, 1719–1723.

Phuong, T.M., Lin, Z., and Altman, R.B. 2005. Choosing SNPs using feature selection. *Proc. IEEE Comput. Syst. Bioinform. Conf. (CSB)* 301–309.

Qin, Z.S., Gopalakrishnan, S., and Abecasis, G.R. 2006. An efficient comprehensive search algorithm for tagSNP selection using linkage disequilibrium criteria. *Bioinformatics* 22, 220–225.

Sawyer, S.L., Mukherjee, N., Pakstis, A.J., *et al.* 2005. Linkage disequilibrium patterns vary substantially among populations. *Eur. J. Hum. Genet*. 13, 677–686.

Sebastiani, P., Lazarus, R., Weiss, S.T., *et al.* 2003. Minimal haplotype tagging. *Proc. Natl. Acad. Sci. USA* 100, 9900–9905.

Stram, D.O., Haiman, C.A., Hirschhorn, J.N., *et al.* 2003. Choosing haplotype tagging SNPs based on unphased genotype data using a preliminary sample of unrelated subjects with an example from the multiethnic cohort study. *Hum. Hered.* 55, 27–36.

The International SNP Working Group. 2001. A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. *Nature* 409, 928–933.

Vazirani, V.V. 2003. *Approximation Algorithms*. Springer-Verlag, New York, NY.

Wang, N., Akey, J.M., Zhang, K., *et al.* 2002. Distribution of recombination crossovers and the origin of haplotype blocks: the interplay of population history, recombination, and mutation. *Am. J. Hum. Genet.* 71, 1227–1234.

Zhang, K., Deng, M., Chen, T., *et al.* 2002. A dynamic programming algorithm for haplotype partitioning. *Proc. Natl. Acad. Sci. USA* 99, 7335–7339.

Zhang, K., Qin, Z., Chen, T., *et al.* 2005. HapBlock: haplo-type block partitioning and tag SNP selection software using a set of dynamic programming algorithms. *Bioinformatics* 21, 131–134.

Zhang, K., and Jin, L. 2003. HaploBlockFinder: haplotype block analyses. *Bioinformatics* 19, 1300–1301.

Zeggini, E., Barton, A., Eyre, S., *et al.* 2005. Characterisation of the genomic architecture of human chromosome 17q and evaluation of different methods for haplotype block definition. *BMC Genet.* 6, 21.

Address correspondence to:
*Dr. Lan Liu*
*Department of Computer Science and Engineering*
*University of California*
*Riverside, CA 92507*

*E-mail:* l.liu@cs.ucr.edu