Informative SNP Selection Methods Based on SNP Prediction

Jingwu He*, *Member, IEEE*, and Alexander Zelikovsky

Abstract*—The search for the association between complex diseases and single nucleotide polymorphisms (SNPs) or haplotypes has recently received great attention. For these studies, it is essential to use a small subset of informative SNPs, i.e., tag SNPs, accurately representing the rest of the SNPs. Tag SNP selection can achieve: 1) considerable budget savings by genotyping only a limited number of SNPs and computationally inferring all other SNPs or 2) necessary reduction of the huge SNP sets (obtained, e.g., from Affymetrix) for further fine haplotype analysis. In this paper, we show that the tag SNP selection strongly depends on how the chosen tags will be used—advantage of one tag set over another can only be considered with respect to a certain prediction method. We show how to separate tag selection from SNP prediction and propose greedy and local-minimization algorithms for tag SNP selection. We give two novel approaches to SNP prediction based on multiple linear regression (MLR) and support vector machines (SVMs). An extensive experimental study on various datasets including ten regions from hapMap project shows that the MLR prediction combined with stepwise tag selection uses fewer tags than the state-of-the-art method of Halperin *et al.* The MLR-based method also uses on average 30% fewer tags than IdSelect for statistical covering all SNPs. The tag selection based on SVM SNP prediction uses fewer tags to achieve the same prediction accuracy as the methods of Halldorsson *et al.

***Index Terms*—Genotypes, haplotypes, informative single nucleotide polymorphism (SNP), single nucleotide polymorphism (SNP), tag selection.**

I. Introduction

THE SEARCH FOR the association between complex diseases and single nucleotide polymorphisms (SNPs) has recently received great attention. For these studies, it is essential to use a small subset of informative SNPs accurately representing the rest of the SNPs.

First, informative SNPs can be used for selective SNP typing (i.e., only physically genotyping certain SNPs instead of scanning all genome) and computationally inferring all nontyped SNPs thus achieving considerable budget savings. Traditionally, such SNPs are called tags and the selection procedure is referred as *haplotype tagging* since SNPs are parts of existing chromosomes [30]. For example, Carlson *et al.* [6] select a maximally informative SNP for association analysis and Chapman, *et al.* [7] use haplotype tags to detect disease associations.

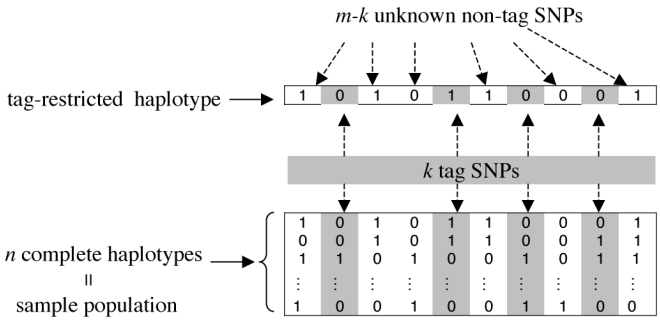

Fig. 1. Informative SNP selection problem. The shaded columns correspond to k tag SNPs and the clear columns correspond to nontag SNPs. The unknown $m - k$ nontag SNP values in tag-restricted individual (top) are predicted based on the known k tag values and complete sample population.

Second, informative SNPs can be used for data compaction of unphased genotypes. Indeed, recent success in high throughput genotyping technologies (e.g., Affimetrix Map Arrays) drastically increase the length of available SNP sequences and they should be compacted to be feasible for fine genotype analysis. In this application, the selected informative SNPs are referred as *index SNPs* [31]. Brinza *et al.* [4], [5] show that the indexing SNP approach combined with association search-based susceptibility prediction algorithms are very promising in disease association studies.

In diploid organisms each chromosome has two "copies" which are not completely identical. Each of two single copies is called a haplotype, while a description of the data consisting of mixture of the two haplotypes is called a genotype. Most experimental techniques for determining SNPs generate for each site an unordered pair of allele readings, one from each haplotype, which is called a genotype. Phasing, or splitting a genotype into two haplotypes, is usually inferred computationally [22].

The methods of informative SNP selection have been previously explored in statistical and pattern recognition communities as well as optimization community. In statistics, tags are required to *statistically cover* individual (nontagged) SNPs or haplotypes (sets of SNPs), where the quality of statistical covering is usually measured by correlation, e.g., find minimum number of tags such that for any nontag SNP there exists a highly correlated (squared correlation $R^2 > .8$) tag SNP [6], [7]. In the optimization community, the number of tags is usually minimized subject to upper bounds on *prediction error* measured in leave-one-out cross-validation experiments [12], [13].

The generic tagging problem can be formulated as the following problem (see Fig. 1).

Manuscript received July 18, 2006; revised October 13, 2006. This work was supported in part by the National Science Foundation under Grant CCF-0429735 and CRDF Award MOM2-3049-CS-03. Preliminary results of this paper were published in [17]–[19]. *Asterisk indicates corresponding author.*

J. He is with Department of Computer Science, Georgia State University, Atlanta, GA 30303 USA (e-mail: jingwu@cs.gsu.edu;,alexz@cs.gsu.edu).

A. Zelikovsky is with Department of Computer Science, Georgia State University, Atlanta, GA 30303 USA (e-mail: jingwu@cs.gsu.edu;,alexz@cs.gsu.edu).

Digital Object Identifier 10.1109/TNB.2007.891901

Informative SNP Selection Problem (ISSP). Given a sample S of a population P of *individuals* (either haplotypes or genotypes) on m SNPs, select positions of k ($k < m$) SNPs such that for any individual, one can predict nonselected SNPs from these k selected SNPs.

This tagging problem formulation implicitly relies on a certain prediction method. The corresponding problem is addressed in the following optimization formulation.

SNP Prediction Problem. Given the values of k tags of the individual x with unknown SNP s and n individuals with k tag SNP and known value of SNP s, find the value of s in x.

Similarly, following Stram *et al.* [26], a *statistical covering criteria* A_k for selected tags requires that the correlation R^2 between the tag(s) and nontag SNP s is higher than a certain threshold (e.g., 0.8).

Statistical Covering Problem. Given a statistical covering criteria A_k and a sample S, find k tags such that the number of statistically covered SNPs (including tags) according to A_k is maximized.

To solve the informative SNP selection problem, one needs to select the tag SNPs based on the population sample and then apply a prediction method to reconstruct unknown individual from its tag SNPs. In this paper, we propose to select tags based on SNP prediction method and give greedy and local-minimization algorithms for tag SNP selection. We suggest two novel approaches to SNP prediction based on multiple linear regression and support vector machines. An extensive experimental study on various datasets including ten regions from [14] shows that the MLR prediction combined with stepwise tag selection uses fewer tags than the state-of-the-art method of [13] for maximizing the prediction accuracy. The above method also uses on average 30% fewer tags than IdSelect [6] for statistical covering all SNPs. The tag selection based on SVM SNP prediction can use fewer tags to reach the same prediction accuracy as the methods of Halldorsson *et al.* [12].

II. Previous Work

Originally, haplotype tags have been selected based on the squared correlation R^2 between true and predicted SNPs in [7] and true and predicted halotype dosage in [26]. Since linkage disequilibrium is usually higher for closer SNPs, the entire SNP sequence is partitioned into blocks ([3], [6], [20], [24], [29], [30]) based on limited haplotype variability and then select tags in each block separately, thus ensuring high correlation between tags and predicted SNPs.

Reconstructing an individual from its typed tag SNPs has received much less attention. Zhang *et al.* [30] presents a method for selecting tag SNPs based on haplotype data, then reconstructing haplotypes with the partition-ligation-expectation-maximization algorithm. Halldorsson *et al.* [12] describes a block-free approach for tag selection. Their method considers a graph with vertices representing SNPs and edges if one SNP can be used to predict the other. The vertices (SNPs) with high degree are chosen as tags. To predict a nontag SNP, that SNP's neighbor's values are inspected and a majority vote is taken. The method is tested with leave-one-out cross-validation and can recover 90% of the haplotype data using only 20% of SNPs as tags.

Halperin *et al.* [13] describes a new method STAMPA for SNP prediction and tag selection. A SNP is predicted by inspecting the two closest tag SNPs from both left and right neighbor; the value of the unknown SNP is given by a majority vote over the two tag SNPs. They use dynamic programming to select tags to reach best prediction score. Their methods are compared with ldSelect and HapBlock on a variety of data sets, and could predict with 80% accuracy the SNPs in the daly dataset [10] using only 2 SNPs as tags.

Lee and Shatkay [21] have recently introduced BNTagger, a new method for tagging SNP selection, based on conditional independence among SNPs. Using the formalism of Bayesian networks (BNs), their system aims to select a subset of independent and highly predictive SNPs. For example, BNTagger uses 10% tags to reach 90% prediction accuracy. However, training of BNTagger requires a significant time—for dataset [10], selecting 52 tag SNPs out of 103 SNPs needs 2–4 h [21]. In this paper we do not compare the MLR with BNTTagger because the former has not been available.

Our preliminary results on the informative SNP selection and SNP prediction problems have been published in [17]–[19]. The results on the statistical covering problem have not been published before.

III. Tag SNP Selection

In this section we show how to separate tag selection from SNP prediction, formulate the corresponding optimization problem, and describe the general approach and two heuristics for tag selection based on prediction.

A *SNP prediction algorithm* A_k accepts as its input the values of k tags $(t_1, \cdots, t_k)$ of an individual x along with the known sample S, in which all of the SNP values for each individual in S is known. The output of A_k is the reconstruction of x, that is, A_k predicts the values of each of the nontag SNPs in x.

Assuming self-similarity of data (i.e., the correlation betweens SNPs in the entire population is similar as in the sample), one can expect that an algorithm predicting with high accuracy SNPs of an unknown individual will also predict with high accuracy SNPs of the sampled individual. Then, we expect that the better prediction algorithm will have fewer errors when predicting SNPs in the sample S. This expectation allows us to find tags using prediction algorithm as follows—check each k-tuple of tags and choose the k-tuple with the minimal number of errors in predicting the nontag SNPs in the sampled individuals. Note that if number of tags k is too small, then we will not be able to distinguish certain individuals and will be forced to have prediction errors. Thus, tag SNP selection based on prediction is reduced to the following problem:

Tag SNP Selection for Prediction. *Given a prediction algorithm A_k and a sample S, find k tags such that the prediction error e of A_k averaged over all SNPs in S (including tags) is minimized.*

Halperin *et al.* [13] proposed a prediction algorithm for solving the TSSP problem with a dynamic programming algorithm STAMPA. Unfortunately, his dynamic programming algorithm for small number of tags sometimes finds tags worse than randomly chosen tags.

Similarly, a *statistical covering criteria* has as an input a set of k column-tags $(t_1, \cdots, t_k)$ and a single SNP s on the sample S. A_k checks if the set of tags has a statistically significant correlation with s. For example, in Carlson *et al.* [6] A_k checks if the R^2 between s and t_i's is higher than a certain threshold (e.g., 0.8). A similar criteria is defined in Stram *et al.* [26]. In this paper, we suggest to compute A_k as a correlation between s and s' predicted using MLR algorithm. The statistical covering version of the tag SNP selection can be formulated as follows.

Tag SNP Selection for Statistical Covering. *Given a statistical covering criteria A_k and a sample S, find k tags such that the number of statistically covered SNPs (including tags) according to A_k is maximized.*

In general, these problems are computationally difficult and the runtime of an exact algorithm may become prohibitively slow. Below we propose two universal heuristics which can be applied to an arbitrary prediction algorithm or statistical covering criteria A_k.

The *Stepwise Tag Selection Algorithm* (STSA) starts with the best tag t_0, i.e., tag that minimizes error when predicting with A_k all other tags. Then STSA finds such tag t_1 which would be the best extension of $\{t_0\}$ and continue adding best tags until reaching the set of tags of the given size k. STSA produces *hereditary* set of tags, i.e., the chosen k tags contain the chosen $k-1$ tags. This hereditary property may be useful in case if the set of tags can be extended. The runtime of STSA is $O(knmT)$, where T is the runtime of the prediction algorithm. Note that for statistical covering, STSA is equivalent to the greedy algorithm used by Carlson *et al.* [6].

The *Local-Minimization Tag Selection algorithm* (LMT) is more accurately searching for a better set of tags among much larger possibilities. LMT starts with the k tags produced by STSA and then iteratively replaces each single tag with the best possible choice while not changing other tags. Such replacements will be continued until no significant improvement in the prediction quality (i.e., by more than given amount of $\epsilon\%$) can be achieved. The runtime of LMT is $O(knmT\epsilon^{-1})$ since the number of iterations cannot exceed $100/\epsilon$.

IV. SNP Prediction Algorithms

A. SNP Prediction

Usually, a genotype is represented by a vector with coordinates 0, 1, or 2, where 0 represents the homozygous site with major allele, 1 represents the homozygous site with minor allele, and 2 represents the heterozygous site. Respectively, each haplotype's coordinate is 0 or 1, where 0 represents the major allele and 1 represents the minor allele. The sample population S together with the tag-restricted individual x are represented as a matrix M. The matrix M has $n+1$ rows corresponding to n sample individuals and the individual x and $k+1$ columns corresponding to k tag SNPs and a single nontag SNP s. All values in M are known except the value of s in x. In case of haplotypes, there are only two possible resolutions of s, namely, s_0 and s_1 with the unknown SNP value equal to 0 or 1, respectively. For genotypes, there are 3 possible resolutions s_0, s_1, and s_2 corresponding to SNP values 0, 1, or 2, respectively. The SNP prediction method should chose correct resolution of s.

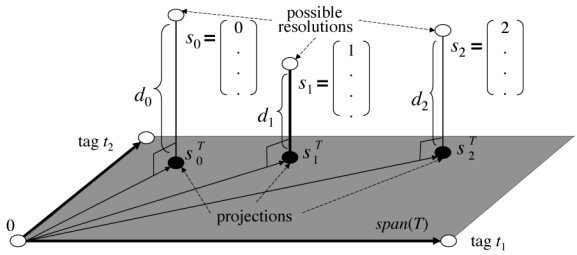

Fig. 2. MLR SNP prediction algorithm. Three possible resolutions s_0, s_1, and s_2 of s are projected on the span of tag SNPs (a dark plane). The unknown SNP value is predicted 1 since the distance between s_1 and its projection s_1^T is the shorter than for s_0 and s_2.

Given the values of k tags of an unknown individual x and the known full sample S, a SNP prediction algorithm A_k predicts the value of a single nontag SNP s in x (if there is more than one nontag SNP to predict, then we handle each one separately). Therefore, without loss of generality, we assume each individual has exactly $k+1$ SNPs.

In SNP prediction, y is a nontag SNP and x_i, $i = 1, \ldots, k$ are tags. Given the known tag values x^* in an individual, we should predict the nontag SNP value y^*. There are three possible values for each SNP (-1, 0, 1) in genotype corresponding to homozygous major allele, heterozygous allele, and homozygous minor allele; there are two possible values for each SNP(-1,1). Note that rather than encode SNP with more common notations (0, 2, 1), we use (-1, 0, 1)-notation.

Formally, let T be the $(n+1) \times k$ matrix consisting of $n+1$ rows corresponding to a tag-restricted genotype $x = (x_1^*, \ldots, x_k^*)$ and n sample genotypes x_i, $i = \overline{1,n}$, from X, $g_i = \{x_{i,1}, \ldots, x_{i,k}\}$, whose k coordinates correspond to k tag SNPs. The SNP s, a nontag SNP, is represented by a $(n+1)$-column with known values y_i, $i = \overline{1,n}$, for genotypes from X and the unknown value y^* for the genotype g which should be predicted

$$T = \begin{bmatrix} x_1^* & \ldots & x_k^* \\ x_{1,1} & \ldots & x_{1,k} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \ldots & x_{n,k} \end{bmatrix} \quad s = \begin{bmatrix} y^* \\ y_{1,k+1} \\ \vdots \\ y_{n,k+1} \end{bmatrix}$$

B. SNP Prediction Based on Multiple Linear Regression

The tags are selected based on the sample population with intention to derive conclusions about the entire population. Statistical analysis may ensure that a high prediction quality of nontag SNPs is not a coincidence. If certain SNPs are highly correlated (i.e., in linkage disequilibrium) in the sample, then we would expect that this correlation will be observed in the entire population. Therefore, it would be highly desirable that the tags contributing to nontag SNP prediction will correlate with the predicted SNP.

The general purpose of multiple linear regression is to learn the relationship between several independent variables and a response variable. The multiple linear regression model is given by

$$\mathbf{y} = \beta_0 + \beta_1 \mathbf{x_1} + \beta_2 \mathbf{x_2} + \ldots + \beta_k \mathbf{x_k} + \epsilon = \mathbf{X}\beta + \epsilon \quad (1)$$

where $\mathbf{y}$ is the response variable (represented by a column with n coordinates ($k \leq n-1$)), $\mathbf{x_i}$, $i = 1, \ldots, k$ are indepen-

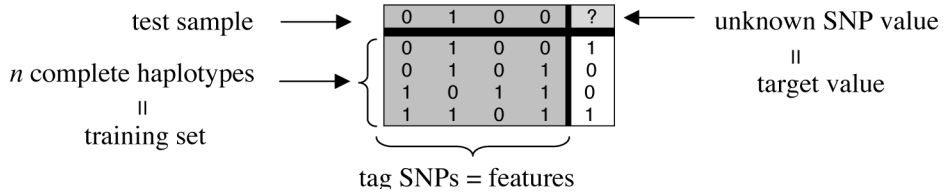

Fig. 3. The SNP prediction problem. Each haplotype with k tags in the training set belongs to a binary class, 0 or 1. These binary class values are represented in the last SNP column. Given a test sample of k tag-restricted haplotype, the unknown nontag SNP in the right corner should be classified based on the known tag SNP values and training set.

dent variables (columns), β_i, $i = 1, \ldots, k$ are regression coefficients, and ϵ (a column) is the model error. The regression coefficient β_i represents the independent contribution of the independent variable x_i to the prediction of $\mathbf{y}$. The multiple linear regression (MLR) method computes b_i, $i = 1, \ldots, k$ to estimate unknown *true coefficients* β_i, $i = 1, \ldots, k$ to minimize the error $||\epsilon||$ using the least squares method. Geometrically speaking, in the *estimation space span*(X), which is the linear closure of vectors x_i, $i = 1, \ldots, k$, we find the vector $\hat{y} = b_0 + b_1x_1 + b_2x_2 + \ldots + b_kx_k = Xb$ estimating y. The vector $\hat{y}$ minimizing distance (error) $||\epsilon|| = ||\hat{y} - y||$ is the projection of y on $span(X)$ and equals $\hat{y} = X(X^tX)^{-1}X^ty$. Given the values of independent variables $x^* = (x_1^*, \ldots, x_k^*)$, the MLR method can predict (estimate) the corresponding response variable y^* with $\hat{y}^* = x^*(X^tX)^{-1}X^ty$.

The proposed MLR SNP prediction method considers all possible resolutions of s together with the set of tag SNPs T as the vectors in $(n+1)$-dimensional Euclidean space. It assumes that the most probable resolution of s should be the "closest" to T. The distance between resolution of s and T is measured between s and its projection on the vector space span(T), the span of the set of tag SNPs T (see Fig. 2).

We calculate $d = ||\epsilon||$, the least square distance between s and T, i.e., $d = |T \cdot (T^t \cdot T)^{-1} \cdot T^t \cdot s - s|$. Our algorithm finds the value (-1, 0 or 1) for y^* and selects one minimizing d.

Running Time. Computing of $T^t \cdot T$ is $O(nk^2)$ since T is a $n \times k$ matrix and T^t is a $k \times n$ matrix. For inverting the $k \times k$ matrix $T^t \cdot T$, we need $O(k^3)$ steps. Let $k < n$, then the running time for computing $T' = T \cdot (T^t \cdot T)^{-1} \cdot T^t$ is $O(n^2k)$. The matrix of T' is the same for all these $(m-k)$ nontag SNPs, thus, the total running time for predicting a complete individual is $O(kn^2 + n^2(m-k)) = O(n^2m)$. If $k \geq n$, then only $(n-1)/2$ closest tags to the right and to the left of the predicted SNP are used. There are only $k - n + 1$ different matrices T' to compute and the total running time is $O(n^3m)$.

C. SNP Prediction Based on Support Vector Machine

In this section we propose to use support vector machine (SVM) for SNP prediction. SVM has recently attracted a lot of attention in bioinformatics research since it produces very accurate results and highly competitive with other data mining approaches such as neural networks. The SVM method is a learning system which is developed by Vapnik and Cortes [28] and is applied for solving problems in nonlinear classification, function estimation and density estimation. The basic principle behind SVM is to find an optimal maximal margin separating hyperplane between two classes. The goal is to maximize the margin between the solid planes separating the two classes and at the same time permit the least amount of errors as possible. SVM can also be used in the case when the data is not linearly separable. In this case, the data is mapped to a high dimensional future space using a nonlinear function. When using SVM, the dot products (x, y) in the future space must be fed to the SVM, which can be computed through a positive definite kernel in the input space.

After given a training set (a set of pairs, input vector: features and target), SVM builds a model. This model is later applied to unknown test set where the model maps an input vector to $+1$ (positive class) or -1 (negative class) output target value. In the SNP prediction problem, SVM builds a model after given n complete haplotypes as training set. Then when an unknown haplotype is given to SVM as a test sample, SVM is asked to predict the unknown SNP value (see Fig. 3). The training set consists of n complete haplotypes each containing 0-1-values of k tag SNPs and the 0-1-value of the single nontag SNP. The value of the the nontag SNP is used to classify haplotypes into two classes corresponding to the major (0) and minor (1) alleles. The class of the testing haplotype is predicted using SVM.

SVMlight is an implementation of Vapnik's SVM [27]. In this project, we have used SVM^{light} software as a black box to do the prediction. The SVM^{light} software has many features such as changing the kernel function and other parameters. We have used the radial basis function (RBF) kernel in our project; it is the default and recommended kernel function

$$\exp\left(-\gamma * |u - v|^2\right)$$

For the tradeoff between training error and margin, 0.05 is chosen (c value). Parameter gamma in RBF kernel was chosen as 0.1. These parameters were found by using try and error in our experiments and once the optimal parameters were found, we used the same for all the tests.

V. Test Datasets

The following datasets are used to measure the quality of our SNP prediction and informative SNP selection algorithms as well as comparison with the results of [13]. We use GERBIL algorithms [11] for resolving missing data. The SNPs with only one allele are removed from the original data.

Seven ENCODE Regions From HapMap. Regions ENr123 and ENm010 from two populations: 45 Han Chinese from Beijing (HCB) and 44 Japanese from Tokyo (JPT) for three regions (ENm013, ENr112, ENr113) from 30 CEPH family trios obtained from HapMap ENCODE Project [14].

Two Gene Regions From HapMap. Two gene regions STEAP and TRPM8 from 30 CEPH family trios were obtained from HapMap.

TABLE I
THE QUALITY OF SNP PREDICTION FROM THE GIVEN NUMBER OF TAGS (5% TO 15% OF THE TOTAL NUMBER OF SNPS (IN PARENTHESES)

Tags	Measure	ENm013 (360)	ENr112 (411)	ENr113 (514)	STEAP (22)	TRPM8 (101)	5q31 (103)
5%	accuracy	0.988	0.949	0.999	0.910	0.893	0.885
	avg R^2	0.639	0.832	0.832	0.425	0.596	0.688
	min R^2	0.486	0.433	0.630	0.010	0.001	0.096
10%	accuracy	0.999	0.994	1	0.969	0.947	0.935
	avg R^2	0.959	0.972	0.972	0.638	0.736	0.794
	min R^2	0.632	0.825	0.789	0.015	0.005	0.181
15%	accuracy	1	1	1	0.983	0.963	0.952
	avg R^2	1	1	1	0.748	0.829	0.845
	min R^2	1	1	1	0.017	0.009	0.248

We report the prediction accuracy (i.e., the percent of correctly predicted SNP values among all predicted nontag SNP values) and the average and minimum R^2 total number of SNPs in each dataset is in the parenthesis.

TABLE II
THE NUMBER OF TAG SNPS SUFFICIENT FOR MLR/STSA TO REACH PREDICTION ACCURACY BETWEEN 80% AND 99%

	prediction accuracy, %											
Datasets	80	85	90	91	92	93	94	95	96	97	98	99
ENm013	2	3	6	6	7	8	9	9	11	15	22	254
ENr112	6	9	14	16	18	20	24	33	63	95	126	187
ENr113	4	5	10	11	13	15	18	40	55	80	104	200
STEAP	1	1	1	2	2	2	2	2	3*	3	4*	4
TRPM8	1	2	4	5	5	6	7	8	10	15	15	24
5q31	1	2	5	7	7	9	13	16	21	31	41	55

The asterisks indicate cases when MLR/LMT needs one tag less than MLR/STSA.

Chromosome 5q31. The data set collected by [10] was derived from the 616 kilobase region of human chromosome *5q31* from 129 family trios.

LPL & Chromosome 21. The Clark *et al.* [8] data set consists of the haplotypes of 71 individuals typed over 88 SNPs in the human lipoprotein lipase (LPL) gene. The chromosome 21 data set consists the first 1000 of 24 047 SNPs typed over 20 haploid copies of human chromosome 21 [24].

VI. EXPERIMENTAL RESULTS AND DISCUSSION

We first report the prediction accuracy and the squared correlation R^2 between predicted and original nontag SNP values. The prediction accuracy is measured as the percentage of correctly predicted SNP values on nontag SNPs. Table I reports the prediction accuracy and the average and the minimum correlation R^2 for all nontag SNPs.

Alternatively, we apply leave-one-out cross validation to evaluate the quality of the MLR-tagging solution for the genotype tagging problem as follows: 1) one by one, each genotype vector is removed from the sample; 2) tag SNPs are selected using only the remaining genotypes; and 3) the "left out" genotype is reconstructed based on its tag SNPs and the values of tag and nontag SNPs in the remaining genotypes.

Table II presents the tagging results of leave-one-out experiments on the six genotype datasets. STSA combined with MLR (MLR/STSA) has almost the same quality as LMT combined with MLR (MLR/LMT) while being much faster. For example, when we perform the test on TRPM8 dataset, MLR/STSA needs 4 s to reach 95% prediction accuracy while MLR/LMT needs 51 s.[1]

[1]All experiments are performed on a computer with Intel Pentium 4, 3.06 GHz processor, and 2 GB of RAM.

TABLE III
NUMBER OF TAGS USED BY MLR-TAGGING, STAMPA, AND RLRP TO ACHIEVE 80% AND 90% PREDICTION ACCURACY IN LEAVE-ONE-OUT TESTS

Acc.	Algorithm	ENm013 (360)	ENr112 (411)	ENr113 (514)	STEAP (22)	TRPM8 (101)	5q31 (103)
80%	MLR	2	6	4	1	1	1
	STAMPA	5	9	11	2	3	2
	RLRP	11	17	35	4	9	10
90%	MLR	6	14	10	1	4	5
	STAMPA	12	17	18	2	6	6
	RLRP	48	52	58	8	22	35

In Table III, we compare MLR with STAMPA [13] and RLRP [16]. One may be wondering that for a prediction accuracy of 80%, the MLR only needs one SNP for "STEAP," "TRPM8," and "5q31" regions. In fact, if one predicts each SNP as 0 (i.e., homozygous with major allele), then the prediction accuracy on STEAP, TRPM8, and 5q31 data will be 79.36%, 72.53%, and 63.57%, respectively. MLR first predicts each SNP as 0 and then gets even higher prediction accuracy when it uses a single tag while STAMPA requires at least two tags for prediction.

Reported advantage of MLR for small k reflects the fact that STAMPA requires computing of $O(m3n)$ table of scores regardless of required number of tags k. This at once makes STAMPA slower in the beginning. Further, for larger k's, STSA slows down MLR while STAMPA does not asymptotically increase runtime. Asymptotically, runtime of STAMPA does not depend on number of tags k and in practice there is no visible dependency either. STAMPA is asymptotically faster but MLR is more accurate compared on four HapMap datasets (see Table IV).

According to the regression model (1), the tags which are more correlated with the predicted SNP have larger regression coefficients and, therefore, will contribute more to predicting the SNP. For example, for seven ENCODE regions [14] and $k = 10$ tags, the tag with the largest regression coefficient ($\approx$0.82 on average) has an average correlation 0.61 with the predicted SNP, the tag with the second largest regression coefficient has average correlation 0.28 and so on. Averaged over all considered real datasets, the correlation between regression coefficients and tag-to-nontag correlations is 0.96. One may believe if this method is applied to an interval that contains regions of high LD with regions of low LD, it may select more SNPs than necessary in the high LD region, because the lack of correlation in the low LD region will overwhelm SNPs in the region with high LD. In fact, when MLR-tagging is applied to data containing both high- and low-LD regions, the high-LD region always have small number of tags since tags in the low-LD region do not correlate with SNPs in the high-LD region and, therefore, do not contribute to high-LD SNP prediction. For ENm010JRT dataset containing 11 SNPs in the high-LD region and 94 SNPs in the low-LD region, only two tags are chosen in the high-LD region out of total $k = 59$ tags.

For maximizing statistical covering, each (nontag) SNP-column s is predicted with the MLR prediction algorithm. We say that the SNP-column s is counted as *statistically covered* if squared correlation R^2 between the predicted SNP-column s' and given SNP-column s is at least 0.8. In Table V, the first two rows show the correlation of prediction accuracy and number of statistically covered SNPs. The third row shows that it is slightly better to use the correct objective

TABLE IV
THE COMPARISON OF MLR'S AND STAMPA'S PREDICTION ACCURACY AND RUNNING TIME BY USING THE NUMBER OF TAGS (2, 5, 10, 15, 20, 25) ON REGION ENR123 (A) AND ENM010 (B) FROM TWO POPULATIONS: CHB AND JPT

Datasets	Han Chinese						Japanese					
	ENr123 (63 SNPs)				ENm010 (105 SNPs)		ENr123 (63 SNPs)				ENm010 (105 SNPs)	
	MLR		STAMPA		MLR	STAMPA	MLR		STAMPA		MLR	STAMPA
Num of Tags	Accuracy	Run Time	Accuracy	Run Time	Accuracy	Accuracy	Accuracy	Run Time	Accuracy	Run Time	Accuracy	Accuracy
2	0.803	0.247	0.744	4.109	0.814	0.792	0.935	0.763	0.895	3.451	0.814	0.792
5	0.928	0.633	0.903	4.136	0.938	0.909	0.955	1.388	0.938	3.462	0.938	0.909
10	0.981	1.893	0.937	4.180	0.980	0.953	0.968	3.978	0.956	3.652	0.980	0.953
15	0.992	3.798	0.952	4.267	0.994	0.968	0.979	10.308	0.960	3.655	0.994	0.968
20	0.998	6.345	0.960	4.385	0.998	0.981	0.989	13.345	0.966	3.852	0.998	0.981
25	0.999	9.357	0.969	4.425	1	0.986	0.995	15.357	0.969	4.425	1	0.986

Total number of SNPs in each dataset is in the parenthesis.

TABLE V
THE QUALITY OF MLR/STSA ON DALY *et al.* [10] DATA WITH TWO DIFFERENT TAGGING OBJECTIVES OVER DIFFERENT NUMBER OF TAG SNPS

objective of tagging		number of tag SNPs						
		0	1	2	4	6	8	10
SNP prediction	prediction accuracy, %	61.54	81.35	83.94	88.65	91.11	92.96	93.89
SNP prediction	# of SNPs covered	0	10	16	36	47	53	59
statistical covering	# of SNPs covered	0	11	24	38	50	54	61

TABLE VI
THE NUMBER OF TAG SNPS FOR STATISTICAL COVERING OF ALL SNPS REQUIRED BY THREE METHODS: MLR/STSA WITH PREDICTION OBJECTIVE, MLR/STSA WITH STATISTICAL COVERING OBJECTIVE, AND IDSELECT [6]

Algorithm	ENm013	ENr112	ENr113	STEAP	TRPM8	5q31
MLR (prediction)	56	82	106	13	46	44
MLR (statistical covering)	51	71	85	11	41	41
IdSelect	71	122	132	16	53	51

TABLE VII
LEAVE-ONE-OUT TESTS ARE PERFORMED ON THREE REAL HAPLOTYPE DATASETS

datasets (num of SNPs)	prediction accuracy %											
	80	85	90	91	92	93	94	95	96	97	98	99
5q31 (103)	1	**1**	**3**	**3**	**4**	**5**	**6**	**8**	**10**	**22**	**42**	**51**
TRPM8 (101)	1	**1**	**2**	5	5	6	7	8	10	15	15	24
STEAP (22)	1	1	1	1	1	1	1	2	**2**	**2**	**2**	2

The minimum number of tag SNPs needed to reach from 80% to 99% prediction accuracy is listed. The bolt numbers indicate cases when the SVM/STSA needs fewer tags than the MLR method of He *et al.* [16] for reaching same prediction accuracy.

TABLE VIII
THE COMPARISON OF OUR PROPOSED SVM/STSA METHOD AND THE MLR METHOD OF HE *et al.* [16] OVER DIFFERENT NUMBER OF TAG SNPS

datasets (num of SNPs)		methods	number of tag SNPs					
			1	2	4	6	8	10
5q31 (103)	prediction accuracy %	SVM/STSA	86.81	89.32	92.24	94.09	95.28	96.09
		MLR	81.15	83.84	88.15	90.91	92.66	93.49
	running time	SVM/STSA	3 hour	5 hour	11 hour	16 hour	18 hour	1 day
		MLR	0.77 sec	1.16 sec	4.07 sec	7.27 sec	11.26 sec	15.92 sec
TRPM8 (101)	prediction accuracy %	SVM/STSA	88.89	90.50	90.67	93.67	95.56	96.74
		MLR	80.68	85.32	90.75	93.74	95.16	96.38
	running time	SVM/STSA	1 hour	2 hour	5 hour	9 hour	16 hour	23 hour
		MLR	0.357 sec	0.787 sec	1.895 sec	3.376 sec	5.181 sec	7.373 sec
STEAP (22)	prediction accuracy %	SVM/STSA	94.02	98.18	99.68	99.73	99.79	99.80
		MLR	90.79	96.16	99.13	99.71	99.78	99.78
	running time	SVM/STSA	14 min	27 min	1 hour	2 hour	3 hour	4 hour
		MLR	0.034 sec	0.052 sec	0.118 sec	0.203 sec	0.304 sec	0.413 sec

(i.e., statistical covering) rather than prediction accuracy in order to maximize the number of statistically covered SNPs. Table VI shows that MLR/STSA uses on average 30% fewer tags than IdSelect [6] for statistical covering all SNPs.

Table VII presents the results of STSA combined with SVM (SVM/STSA) on leave-one-out experiments on the three haplotype datasets. Table VIII compares SVM/STSA with multiple linear regression method (MLR) on the three haplotype datasets. The proposed tagging method is more accurate than multiple linear regression method. For example, for small number of tag SNPs, SVM/STSA can obtain (up to 8%) better prediction accuracy than MLR with same number of tag SNPs. But SVM/STSA is considerably slower. Indeed, for 5q31 dataset, SVM/STSA needs 3 h to select 1 tag SNPs while MLR only needs 0.77 s.

We also compare SVM/STSA with the methods of Halldorson *et al.* [12] and the RLR method [16] in leave-one-out

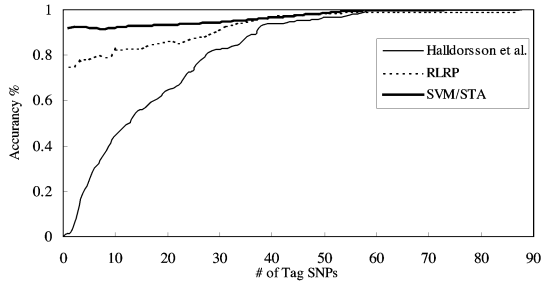

Fig. 4. Comparison among three haplotype tagging method on LPL data: SVM/STSA, Halldorson *et al.* [12], and He *et al.* [16] in a leave-one-out experiment. The x axis shows the number of SNPs typed, and the y axis shows the fraction of SNPs correctly imputed.

tests on the LPL data set (see Fig. 4). Note that the method of Halldorson *et al.* imputes a SNP based on the tag SNPs in the same neighborhood and in fact can be classified as a method for statistical coverage. If there is no tag SNPs in the neighborhood, then their method does not make any prediction. It is not surprising that it performs poorly for SNP prediction. The SVM/STSA method reconstructs each SNP based on the values of *all* tag SNPs which may potentially be far away. On the LPL dataset, SVM/STSA reaches, e.g., 90% accuracy using only one tag.

VII. Conclusion

In this paper, we show that the tag SNP selection strongly depends on how the chosen will be used—advantage of one tag set over another can only be considered with respect to a certain prediction method. We show how to separate tag selection from SNP prediction and propose greedy and local minimization algorithms for tag SNP selection. We suggest two novel approach to SNP prediction based on multiple linear regression and SVMs. An extensive experimental study on various datasets including 10 regions from [14] shows that the MLR prediction combined with stepwise tag selection uses fewer tags than the state-of-the-art method of [13]. The above method also uses on average 30% fewer tags than IdSelect [6] for statistical covering all SNPs. The tag selection based on SVM SNP prediction can use fewer tags to reach the same prediction accuracy as the methods of Halldorsson *et al.* [12]

References

[1] Affymetrix, 2005 [Online]. Available: http://www.affymetrix.com/products/arrays/

[2] H. Ackerman, S. Usen, R. Mott, A. Richardson, F. Sisay-Joof, P. Katundu, T. Taylor, R. Ward, M. Molyneux, M. Pinder, and D. P. Kwiatkowski, "Haplotypic analysis of the TNF locus by association efficiency and entropy," *Genome Biol.*, vol. 4, p. 24, 2003.

[3] H. I. Avi-Itzhak, X. Su, and F. M. de la Vega, "Selection of minimum subsets of single nucleotide polymorphism to capture haplotype block diversity," in *Proc. Pacific Symp. Biocomputing*, 2003, vol. 8, pp. 466–477.

[4] D. Brinza, J. He, and A. Zelikovsky, "Combinatorial search methods for multi-SNP disease association," in *Proc. Int. Conf. IEEE Engineering in Medicine and Biology (EMBC'06)*, pp. 5802–5805.

[5] D. Brinza and A. Zelikovsky, "Combinatorial methods for disease association search and susceptibility prediction," in *Proc. Workshop Algorithms in Bioinformatics (WABI 2006)*, vol. 4175, LNBI, pp. 286–297.

[6] C. S. Carlson, M. A. Eberle, M. J. Rieder, Q. Yi, L. Kruglyak, and D. A. Nickerson, "Selecting a maximally informative set of single-nucleotide polymorphisms for association analyses using linkage disequilibrium," *Amer. J. Hum. Genet.*, vol. 74, no. 1, pp. 106–120, 2004.

[7] J. M. Chapman, J. D. Cooper, J. A. Todd, and D. G. Clayton, "Detecting disease associations due to linkage disequilibrium using haplotype tags: A class of tests and the determinants of statistical power," *Hum. Heredity*, vol. 56, pp. 18–31, 2003.

[8] A. Clark, K. Weiss, D. Nickerson, S. Taylor, A. Buchanan, J. Stengard, V. Salomaa, E. Vartiainen, M. Perola, and E. Boerwinkle *et al.*, "Haplotype structure and population genetic inferences from nucleotide-sequence variation in human lipoprotein lipase," *Amer. J. Hum. Genet.*, vol. 63, pp. 595–612, 1998.

[9] A. Clark, "Finding genes underlying risk of complex disease by linkage disequilibrium mapping," *Curr. Opin. Genet. Develop.*, vol. 13, no. 3, pp. 296–302, 2003.

[10] M. Daly, J. Rioux, S. Schaffner, T. Hudson, and E. Lander, "High resolution haplotype structure in the human genome," *Nature Genet.*, vol. 29, pp. 229–232, 2001.

[11] G. Kimmel and R. Shamir, "GERBIL: genotype resolution and block identification using likelihood," *PNAS*, vol. 102, pp. 158–162, 2004.

[12] B. V. Halldorsson, V. Bafna, R. Lippert, R. Schwartz, F. M. de la Vega, A. G. Clark, and S. Istrail, "Optimal haplotype block-free selection of tagging SNPs for genome-wide association studies," *Genome Res.*, vol. 14, pp. 1633–1640, 2004.

[13] E. Halperin, G. Kimmel, and R. Shamir, "Tag SNP selection in genotype data for maximizing SNP prediciton accuracy," *Bioinformatics*, vol. 21, pp. 195–203, 2005.

[14] International HapMap Consortium, "The international HapMap project," *Nature* vol. 426, pp. 789–796, 2003 [Online]. Available: http://www.hapmap.org

[15] Y. H. Huang, K. Zhang, T. Chen, and K. M. Chao, "Approximation algorithms for the selection of robust tag SNPs," in *Proc. Workshop on Algorithms in Bioinformatics (WABI'04)*, vol. 3240, LNCS, pp. 278–289.

[16] J. He, K. Westbrooks, and A. Zelikovsky, "Linear reduction method for predictive and informative tag SNP selection," *Int. J. Bioinf. Res. Appl.*, vol. 3, pp. 249–260, 2005.

[17] J. He and A. Zelikovsky, "Tag SNP selection based on multivariate linear regression," in *Proc. Int. Conf. Computational Science (ICCS 2006)*, vol. 3992, LNCS, pp. 750–757.

[18] J. He, J. Zhang, G. Altun, A. Zelikovsky, and Y. Zhang, "Haplotype tagging using support vector machines," in *Proc. IEEE Int. Conf. Granular Computing (GRC 2006)*, pp. 758–761.

[19] J. He and A. Zelikovsky, "MLR-tagging: Informative SNP selection for unphased genotypes based on multiple linear regression," *Bioinformatics*, vol. 22, pp. 2558–2561, 2006.

[20] R. Judson, B. Salisbury, J. Schneider, A. Windemuth, and J. C. Stephens, "How many SNPs does a genome-wide haplotype map require?," *Pharmacogenomics*, vol. 3, pp. 379–391, 2002.

[21] P. H. Lee and H. Shatkay, "BNTagger: improved tagging SNP selection using Bayesian networks," *Bioinformatics*, vol. 22, pp. e211–e219, 2006.

[22] T. Niu, "Algorithms for inferring haplotypes," *Genet. Epidemiol.*, vol. 4, pp. 334–347, 2004.

[23] *Electronic Statistics Textbook*. Tulsa, OK: StatSoft, 1999 [Online]. Available: http://www.statsoft.com/textbook/stathome.html, StatSoft, Inc.

[24] N. Patil, A. Berno, D. Hinds, W. Barrett, J. Doshi, C. Hacker, C. Kautzer, D. Lee, C. Marjoribanks, D. McDonough, B. Nguyen, M. Norris, J. Sheehan, N. Shen, D. Stern, R. Stokowski, D. Thomas, M. Trulson, K. Vyas, K. Frazer, S. Fodor, and D. Cox, "Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome," *Science*, vol. 294, pp. 1719–1723, 2001.

[25] P. Sebastiani, R. Lazarus, S. Weiss, L. Kunkel, I. Kohane, and M. Ramoni, "Minimal haplotype tagging," *Proc. Nat. Acad. Sci.*, vol. 100, pp. 9900–9905, 2003.

[26] D. Stram, C. Haiman, J. Hirschhorn, D. Altshuler, L. Kolonel, B. Henderson, and M. Pike, "Choosing haplotype-tagging SNPs based on unphased genotype data using as preliminary sample of unrelated subjects with an example from the multiethnic cohort study," *Hum. Heredity*, vol. 55, pp. 27–36, 2003.

[27] T. Joachims, "Making large-scale SVM learning practical," in *Advances in Kernel Methods—Support Vector Learning*, B. Scholkopf, C. Burges, and A. Smola, Eds. Cambridge, MA: MIT Press, 1999.

[28] V. Vapnik and C. Cortes, "Support vector networks," *Mach. Learn.*, vol. 20, pp. 273–293, 1995.

[29] E. Weale, C. Depondt, S. J. Macdonald, A. Smith, P. S. Lai, S. D. Shorvon, N. W. Wood, and D. B. Goldstein, "Selection and evaluation of tagging SNPs in the neuronal-sodium-channel gene SCN1A: Implications for linkage-disequilibrium gene mapping," *Amer. J. Hum. Genet.*, vol. 73, pp. 551–565, 2003.

[30] K. Zhang, Z. Qin, J. Liu, T. Chen, M. Waterman, and F. Sun, "Haplotype block partitioning and tag SNP selection using genotype data and their applications to association studies," *Genome Res.*, vol. 14, pp. 908–916, 2004.

[31] P. Zhang, H. Sheng, and R. Uehara, "A double classification tree search algorithm for index SNP selection," *BMC Bioinformatics*, vol. 5, pp. 89–95, 2004.

Jingwu He received the M.S. and Ph.D. degrees in computer science from Georgia State University, Atlanta, in 2002 and 2006, respectively.

He is the author of more than 20 refereed publications. His research interests include bioinformatics, discrete and approximation algorithms, combinatorial optimization, distributed and mobile computing systems, software engineering, and databases.

Alexander Zelikovsky received the Ph.D. degree in computer science from the Institute of Mathematics of the Belorussian Academy of Sciences, Minsk, Belarus, in 1989.

He worked at the Institute of Mathematics in Kishinev, Moldova, from 1989 to 1995. Between 1992 and 1995 he visited Bonn University and the Institut fur Informatik, Saarbrueken, Germany. He was a Research Scientist at University of Virginia (1995–1997) and a Postdoctoral Scholar at the University of California, Los Angeles (1997–1998). He is currently an Associate Professor at Computer Science Department of Georgia State University, Atlanta, which he joined in 1999. He also serves on the editorial board of the *International Journal of Bioinformatics Research and Applications*, and is a guest editor for *LNCS Transactions on Computational Systems Biology*, *International Journal of Wireless and Mobile Computing*, and the *Journal of Universal Computer Science*. He is the author of more than 130 refereed publications. His research interests include bioinformatics, discrete and approximation algorithms, combinatorial optimization, VLSI physical layout design, and ad hoc wireless networks.

Dr. Zelikovsky is founding cochair of the ACIS International Workshop on Self-Assembling Wireless Networks (SAWN) and International Workshop on Bioinformatics Research and Applications (IWBRA). He is Program Committee cochair of the 2007 International Symposium on Bioinformatics Research and Applications (ISBRA). He is a guest editor for IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS. Dr. Zelikovsky received the best paper award at the joint Asia-South Pacific Design Automation/VLSI Design Conferences in 2003 and the best poster awards at the Annual BACUS Symposium on Photomask Technology in 2005 and the Fifth Georgia Tech International Conference on Bioinformatics in 2005.