BIOINFORMATICS

Vol. 22 no. 14 2006, pages e211–e219
doi:10.1093/bioinformatics/btl233

BNTagger: improved tagging SNP selection using Bayesian networks

Phil Hyoun Lee* and Hagit Shatkay*
School of Computing, Queen's University, Kingston, ON, Canada

ABSTRACT

Genetic variation analysis holds much promise as a basis for disease-gene association. However, due to the tremendous number of candidate single nucleotide polymorphisms (SNPs), there is a clear need to expedite genotyping by selecting and considering only a subset of all SNPs. This process is known as *tagging SNP selection*. Several methods for tagging SNP selection have been proposed, and have shown promising results. However, most of them rely on strong assumptions such as prior block-partitioning, bi-allelic SNPs, or a fixed number or location of tagging SNPs.

We introduce BNTagger, a new method for tagging SNP selection, based on conditional independence among SNPs. Using the formalism of Bayesian networks (BNs), our system aims to select a subset of independent and highly predictive SNPs. Similar to previous prediction-based methods, we aim to maximize the prediction accuracy of tagging SNPs, but unlike them, we neither fix the number nor the location of predictive tagging SNPs, nor require SNPs to be bi-allelic. In addition, for newly-genotyped samples, BNTagger directly uses genotype data as input, while producing as output haplotype data of all SNPs.

Using three public data sets, we compare the prediction performance of our method to that of three state-of-the-art tagging SNP selection methods. The results demonstrate that our method consistently improves upon previous methods in terms of prediction accuracy. Moreover, our method retains its good performance even when a very small number of tagging SNPs are used.

Contact: lee@cs.queensu.ca, shatkay@cs.queensu.ca

1 INTRODUCTION

A major interest of current genomics research is *disease-gene association*, that is, identifying which DNA variations are highly associated with a specific disease. In particular, single nucleotide polymorphisms (SNPs), which are the most common form of DNA variation, as well as sets of SNPs localized on one chromosome—referred to as *haplotypes*—are at the forefront of disease-gene association studies (Halldórsson *et al*., 2004b; Crawford and Nickerson, 2005). However, in most large-scale association studies, genotyping all SNPs in a candidate region for a large number of individuals is still costly and time-consuming. Thus, selecting a subset of SNPs that is sufficiently informative but still small enough to reduce the genotyping overhead is an important step toward disease-gene association. This process is known as *haplotype tagging SNP (htSNP) selection*, and it poses a current major challenge (Crawford and Nickerson, 2005; Johnson *et al*., 2001).

Several computational methods for htSNP selection have been proposed in the past few years. One widely-used approach is based on *the block structure of the human genome* (Daly *et al*., 2001; Gabriel *et al*., 2002). That is, the human genome can be viewed as a set of discrete blocks such that within each block, there is a very small set of common haplotypes shared by most of the population (i.e., 80–90%). Based on this idea, these methods aim to identify a subset of SNPs that can distinguish all the common haplotypes (Gabriel *et al*., 2002), or at least explain a certain percentage of them (Johnson *et al*., 2001; Avi-Itzhak *et al*., 2003). Another popular htSNP selection approach (Ao *et al*., 2005; Carlson *et al*., 2004), rooted in linkage disequilibrium (LD), is based on *pairwise association* of SNPs. This approach tries to select a set of htSNPs such that each of the SNPs on a haplotype is *highly associated* with one of the htSNPs. This way, although the SNP that is directly responsible for the disease may not be selected as an htSNP, the association of the target disease with that SNP can be indirectly deduced from its associated htSNP.

Bafna *et al*. (2003) and Halldórsson *et al*. (2004) proposed a somewhat different approach. They consider htSNPs to be a subset of all SNPs, from which the remaining SNPs can be reconstructed. Thus, they aim to select htSNPs based on how well they *predict* the remaining set of the unselected SNPs, referred to as *tagged* SNPs, and *reconstruct* the complete haplotypes using htSNPs. To quantify the confidence with which one group of SNPs can predict another, they suggested a new measure called *informativeness*. With the same predictive aim, Halperin *et al*. (2005) also proposed a new measure, directly evaluating the prediction accuracy of a set of SNPs. By limiting the number of predictive SNPs or restricting them to a *w*-bounded neighborhood (where *w* is a fixed window size $\leq$ 30), both methods can identify the optimal (under these restrictions) set of htSNPs satisfying their respective figure of merit.

These last two methods are not based on the block structure of the human genome. Thus, they do not assume prior block partitioning or limited diversity of haplotypes. Furthermore, they can use a combination of several SNPs to predict the others. Therefore, predictive methods typically select a smaller number of htSNPs than pairwise association methods (De Bakker *et al*., 2006). However, despite their advantages, these predictive methods still suffer from several limitations. All of them can only be applied to bi-allelic SNPs (i.e., ones

*To whom correspondence should be addressed.

having only two different alleles[1]), and their performance is limited by restrictions such as the small-bounded location or the fixed number of htSNPs for each prediction. In addition, most of them require haplotype information of htSNPs to reconstruct newly-genotyped samples.

In this paper, we present a new method, BNTagger, for selecting htSNPs based on their accuracy in predicting tagged SNPs, that is not limited by previous restrictions. In addition, we provide a haplotype-reconstruction framework for newly-genotyped samples. To identify a predictor-predictee relationship among SNPs, we utilize conditional independencies among SNPs in the framework of Bayesian networks. Bayesian networks (BNs) have been previously used for haplotype block partitioning (Greenspan and Geiger, 2003) and haplotype phasing (Xing *et al*., 2004), but to our knowledge, this is the first time that they are applied to htSNP selection. BNTagger uses three main steps:

(1) Identifying the conditional independence relations among SNPs.

(2) Selecting htSNPs using two heuristics.

(3) Reconstructing the complete haplotypes for newly-genotyped samples.

Similar to other predictive methods, our system aims to select htSNPs maximizing the prediction accuracy for the remaining tagged SNPs. However, it has several unique aspects. First, unlike all previous work (Bafna *et al*., 2003; Halldórsson *et al*., 2004; Halperin *et al*., 2005), we do not fix the neighborhood nor the number of predictive htSNPs for each tagged SNP. Although SNPs within close physical proximity are assumed to be in a state of high linkage disequilibrium (LD), recent studies have reported that the levels of LD vary across chromosomal regions (Reich *et al*., 2001; Daly *et al*., 2001). Therefore, as noted by Bafna *et al*. (2003), *''... it is neither efficient nor desirable to fix the neighborhood in which htSNPs are selected''*. Moreover, it is realistic to assume that a different number of htSNPs may be needed for predicting each tagged SNP.

Second, our system is not restricted to the case of bi-allelic SNPs. While most SNPs are indeed bi-allelic, there are SNPs that can take on more than two nucleotides. While these cases may be rare, it is still unknown whether disease variants are rare or common haplotypes (Crawford and Nickerson, 2005). Thus, it is desirable to impose as few restrictions as possible on htSNP selection (Palmer and Cardon, 2005).

Third, for newly-geneotyped samples, we directly construct *haplotype* data of *all* SNPs using *genotype* data of htSNPs. As pointed by Halperin *et al*. (2005), the accuracy of haplotype phasing based only on htSNPs is limited due to the reduced LD among htSNPs. Therefore, it is reasonable to assume that reliable haplotype data are not available in the case of newly-genotyped samples. However, we note that, unlike Halperin's method, which uses genotype data as input and as output as well, we directly output the *haplotype* data of all SNPs for new samples. Thus, subsequent haplotype phasing for the reconstructed samples is unnecessary.

We applied our method to three public data sets (Daly *et al*., 2001; Rieder *et al*., 1999; Nickerson *et al*., 2000). Based on leave-one-out and on 10-fold cross validation, our results demonstrate that using our selection method, about 2.9%–11.5% of the total SNPs are sufficient to predict the others with 90% accuracy. We also compare our prediction performance to that of recently published htSNP selection methods (Bafna *et al*., 2003; Halldörsson *et al*., 2004; Lin and Altman, 2004; Halperin *et al*., 2005). The results show that our method extracts *fewer htSNPs* while achieving the same level of prediction accuracy. Moreover, our method retains its good performance even when a very small number of htSNPs is used.

In section 2, we formulate the problem of htSNP selection in the context of prediction accuracy, and introduce the basic notations that are used throughout the paper. Section 3 briefly provides the necessary background on Bayesian networks, focusing on the concepts most relevant to our algorithm. Our selection and haplotype reconstruction algorithms are described in section 4. Section 5 reports our evaluation results. Section 6 summarizes our findings and outlines future directions.

2 PROBLEM FORMULATION

A haplotype represents the allele information of contiguous SNPs on *one* chromosome, while a genotype represents the *combined* allele information of the SNPs on a *pair* of chromosomes. Thus, the allele information of haplotypes takes on values from $\{a, g, c, t\}$, while that of genotypes takes on values from $\{a/a, a/g, a/c, a/t, \ldots, t/c, t/t\}$. When the combined allele information of a pair of haplotypes, h_j and h_k, comprises the genotype g_i, we say that h_j and h_k *resolve* g_i. For example, the two haplotypes $h_j = (a, g, a, c)$ and $h_k = (a, c, c, a)$ resolve the genotype $g_i = (a/a, c/g, a/c, a/c)$. We also refer to haplotypes h_j and h_k as the *complementary mates* of each other to resolve g_i, and consider them to be *compatible* with g_i.

Let D be a data set consisting of n haplotypes, $h_1, \ldots, h_n$, each with p different SNPs, $s_1, \ldots, s_p$. The set D can be viewed as an n by p matrix. Each row, D_{i-}, in D corresponds to haplotype h_i, while each column, D_{-j}, corresponds to a SNP s_j. D_{ij} denotes the j^{th} SNP in the i^{th} haplotype. We view each SNP as a discrete random variable, X_j, that takes on values from a finite domain $\{a, g, c, t\}$. Thus, we define the finite set $V = \{X_1, \ldots, X_p\}$, in which each random variable X_j corresponds to the j^{th} SNP on a haplotype in the data set D.

Given the set V of random variables corresponding to the p SNPs, our goal is to find a subset $T \subset V$, such that the size of T, $|T|$, is smaller than some pre-specified constant k, and SNPs in T can best predict the remaining unselected ones, $V - T$. As defined earlier, the selected SNPs are referred to as haplotype *tagging* SNPs (htSNPs), and the unselected ones are referred to as *tagged* SNPs. Suppose that our htSNP set T consists of q SNPs, $T = \{X_{t_1}, \ldots, X_{t_q}\}$. To predict the allele of a tagged SNP X_j given the alleles of the htSNPs, T, we use the posterior probability of X_j conditioned on the set T, $Pr(X_j \mid X_{t_1}, \ldots X_{t_q})$. That is, the allele whose conditional probability is the highest given the alleles of the predictive htSNPs is taken to be the allele of the tagged SNP. When multiple maximum probability solutions exist, the most common allele of X_j is selected. To capture the idea that this prediction can be either correct or incorrect, we introduce the following indicator function P_f.

[1] The nucleotide $\in \{a, g, c, t\}$ at a position in which a SNP occurred is called an *allele*.

Definition 1. *Prediction Indicator Function: Given a predictive htSNP set,* $T = \{X_{t_1}, \ldots, X_{t_q}\}$*, a predicted tagged SNP,* $X_j \in V - T$*, and a haplotype,* D_{i-}*, a prediction indicator function* $P_f(X_j, T, D_{i-})$ *is defined*[2] *as*

$$P_f(X_j, T, D_{i-}) = \\ = \begin{cases} 1 : if\ D_{ij} == \\ \qquad \underset{x \in \{a,g,c,t\}}{arg\ max}\ Pr(X_j = x \mid X_{t_1} = D_{it_1}, \ldots, X_{t_q} = D_{it_q}); \\ 0 : otherwise. \end{cases}$$

We note that the prediction of each tagged SNP is assumed to depend on the values of the htSNPs, but not on the other predicted tagged SNPs. Hence, prediction can be applied in any order. Using this prediction indicator function, we formally define our objective as follows:

Definition 2. *Maximally Predictive htSNP Set: Given a set of p SNPs,* $V = \{X_1, \ldots, X_p\}$*, a constant k, and a prediction indicator function* P_f*, a maximally predictive htSNP set,* $T = \{X_{t_1}, \ldots, X_{t_q}\}$*, for a set of haplotypes D is defined as a subset T of V,* $(T \subset V)$*, satisfying two criteria*:

$$1)\quad |T| < k, \quad and$$
$$2)\quad T = \underset{T' \subset V}{argmax} \sum_{j=1}^{p} \sum_{i=1}^{n} p_f(X_j, T', D_{i-}).$$

That is, T is the subset of SNPs that is likely to predict correctly the largest number of SNPs in $V - T$. BNTagger utilizes the framework of Bayesian networks to effectively compute the posterior probability in P_f and to select a set of htSNPs. In the next section, we briefly introduce the necessary background on Bayesian networks.

3 BAYESIAN NETWORKS

A Bayesian network (BN) is a graphical model of joint probability distributions that captures conditional independencies among its variables (Jensen, 2002). Given a finite set $V = \{X_1, \ldots, X_p\}$ of random variables, a Bayesian network has two components: a directed acyclic graph, G, and a set of conditional probability parameters, $\Theta = \{\theta_1, \ldots, \theta_p\}$. Each node of the graph G corresponds to a random variable X_j. An edge between two nodes represents a direct dependence between the two random variables, and the lack of an edge represents their *conditional independence*. Using the conditional independence encoded in the structure of the BN (Jensen, 2002), the joint probability distribution of the random variables in V can be computed as the product of their conditional probability parameters:

$$Pr(V) = \prod_{j=1}^{p} \theta_j = \prod_{j=1}^{p} Pr(X_j \mid pa(X_j)),$$

where $pa(X_j)$ denotes the *parent* nodes of X_j. The BN formalism enables the computation of the posterior probability of a target variable when the values of some of the other variables are observed. This computation process is typically referred to as *BN inference*. Suppose that we have observed the values of q variables, $X_{t_1} = e_1, \ldots, X_{tq} = e_q$, in a BN. Based on this information, the conditional distribution of X_j can be computed from the joint probability of V by marginalizing out all unobserved variables except X_j, denoted as $M = V - \{X_j, X_{t_1}, \ldots, X_{t_q}\}$ (Jensen, 2002). Let m denote any of the possible instantiation of the random variables in M. The posterior probability of X_j can thus be calculated as:

$$\begin{aligned} & Pr(X_j \mid X_{t_1} = e_1, \ldots, X_{t_q} = e_q) \\ & = \frac{\sum_m Pr(M = m, X_j, X_{t_1} = e_1, \ldots, X_{t_q} = e_q)}{Pr(X_{t_1} = e_1, \ldots, X_{t_q} = e_q)} \\ & = \frac{\sum_m \prod_{X_k \in V} Pr(X_k \mid pa(X_k))^*}{Pr(X_{t_1} = e_1, \ldots, X_{t_q} = e_q)}, \end{aligned} \tag{1}$$

where the summation is over all possible combinations of values m assigned to all the unobserved variables in M, and the value of every observed variable, X_{t_i}, is set to e_i in $Pr(X_k \mid pa(X_k))^*$.

The *Markov blanket* is another central concept in Bayesian networks. The Markov blanket of X_j includes the parents of X_j, the children of X_j, and the other parents of X_j's children (Jensen, 2002). In a BN, X_j is conditionally independent of all other variables given its Markov blanket. This typically speeds up the calculation of the posterior $Pr(X_j \mid X_{t_1} = e_1, \ldots, X_{t_q} = e_q)$ since when the Markov blanket of X_j is observed, only this information needs to be taken into account for computing the distribution of X_j.

Numerous BN inference algorithms have been developed to compute this posterior probability exactly or approximately. We use the *Generalized Variable Elimination* algorithm implemented in JavaBayes (Cozman, 2000) to compute the posterior probability used in our prediction indicator function P_f.

To use the BN inference algorithm, we must first identify the structure (G) and parameters (Θ) of the BN representing the haplotype data D. This process is referred to as *BN learning*. *Structure* learning aims to find the graph structure G which maximizes the conditional probability of G given the data D, as follows:

$$\begin{aligned} G &= \underset{G'}{argmax}\ Pr(G' \mid D) = \underset{G'}{argmax}\ \frac{Pr(D \mid G') \cdot Pr(G')}{Pr(D)} \\ &= \underset{G'}{argmax}\ Pr(D \mid G') \cdot Pr(G'). \end{aligned}$$

We use the Minimum Description Length (MDL) score (Lam and Bacchus, 1994) to reflect the above probabilistic scoring. In the same vein, *parameter* learning in a BN aims to find Θ which maximizes the conditional probability of Θ given the data D, $Pr(\Theta \mid D)$. We use a maximum-likelihood approach to estimate Θ.

4 METHODS

BNTagger aims to select a set of htSNPs that predicts the tagged SNPs with the highest accuracy. However, finding this set of htSNPs in the general case has been proven to be NP-hard (Bafna *et al*., 2003). To effectively identify the set of highly predictive SNPs, T, we use several heuristics, utilizing the framework of a Bayesian network (BN) and the conditional independence captured in it.

Figure 1 provides a simple example for how BNTagger utilizes the conditional independencies among SNPs to select htSNPs. The sample here consists of ten haplotypes with four SNPs each (Figure 1(a)); the BN structure that represents conditional independencies among the four SNPs along with the probability parameters is found via BN learning, and shown in Figure 1(b). For simplicity, the conditional probabilities are

[2] For any SNP $X_{t_i} \in T$, $P_f(X_{t_i}, T, D_{i-})$ is taken to be 1 always.

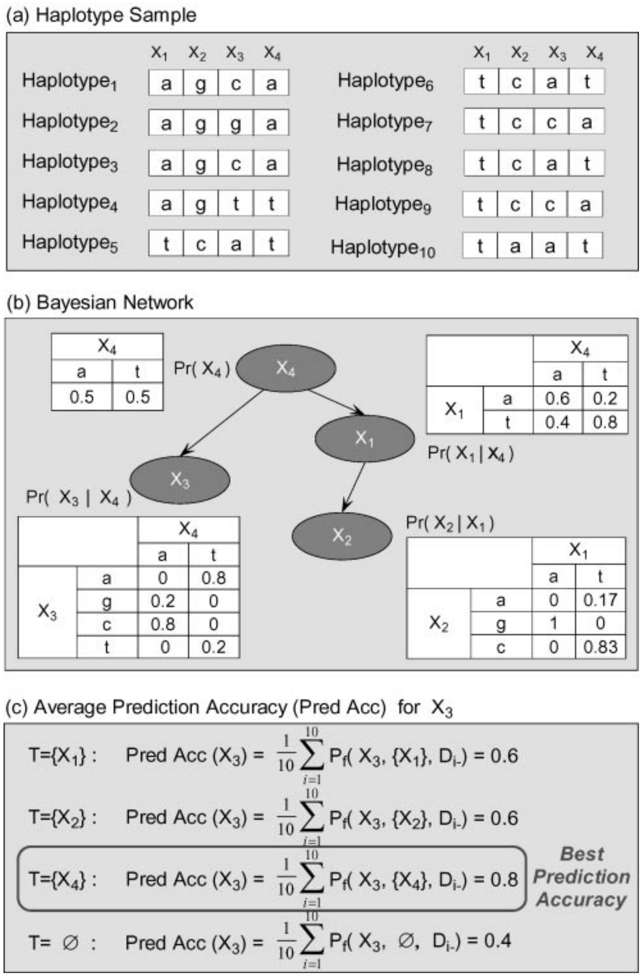

(a) Haplotype Sample
X1 X2 X3 X4
Haplotype1 a g c a
Haplotype2 a g g a
Haplotype3 a g c a
Haplotype4 a g t t
Haplotype5 t c a t
Haplotype6 t c a t
Haplotype7 t c c a
Haplotype8 t c a t
Haplotype9 t c c a
Haplotype10 t a a t
(b) Bayesian Network
Pr(X4)
X4: a 0.5, t 0.5
X4
X1
X3
X2
Pr(X1 | X4)
X1 = a: X4 = a 0.6, X4 = t 0.2
X1 = t: X4 = a 0.4, X4 = t 0.8
Pr(X3 | X4)
X3 = a: X4 = a 0, X4 = t 0.8
X3 = g: X4 = a 0.2, X4 = t 0
X3 = c: X4 = a 0.8, X4 = t 0
X3 = t: X4 = a 0, X4 = t 0.2
Pr(X2 | X1)
X2 = a: X1 = a 0, X1 = t 0.17
X2 = g: X1 = a 1, X1 = t 0
X2 = c: X1 = a 0, X1 = t 0.83
(c) Average Prediction Accuracy (Pred Acc) for X3
T={X1} : Pred Acc (X3) = 1/10 Σ Pf(X3, {X1}, Di-) = 0.6
T={X2} : Pred Acc (X3) = 1/10 Σ Pf(X3, {X2}, Di-) = 0.6
T={X4} : Pred Acc (X3) = 1/10 Σ Pf(X3, {X4}, Di-) = 0.8
Best Prediction Accuracy
T= ∅ : Pred Acc (X3) = 1/10 Σ Pf(X3, ∅, Di-) = 0.4

Fig. 1. A Bayesian network of SNPs and examples of prediction accuracy values.

shown only for alleles occurring in the sample. The other probabilities are considered here to be zero.

To select htSNPs given a Bayesian network, BNTagger starts with an empty htSNP set T, and sequentially examines the average prediction accuracy for each SNP (node) based on the current set, T. If the prediction accuracy for a SNP, X_j, is smaller than a pre-specified threshold, BNTagger adds X_j into T as a new htSNP, because X_j is not well-predicted by the current htSNPs in T. Clearly, the order in which SNPs are evaluated is very important, since it can directly affect the selected set of htSNPs and their prediction performance. Unlike other methods that sequentially examine SNPs in the order of their *chromosomal location*, BNTagger examines the SNPs in the *topological* order (from parents to children) in the BN. For example, in Figure 1(b), BNTagger first examines the root X_4, then its children X_3, X_1, and so on. Thus, when the prediction accuracy for each SNP X_j is evaluated, given T, the htSNPs in the current set T are all ancestors of X_j. This has two advantages:

First, the parent-child relation in the BN encodes the direct dependence between these nodes, that is, the state of child nodes depends primarily on the information of their parents. For example, Figure 1(c) shows the prediction accuracy[3] for SNP X_3 assuming each of the other SNPs, X_1, X_2, or X_4 as an htSNP, as well as when assuming no htSNP is used. All the prediction accuracies are higher when htSNP information is given than when it is not. Moreover, the best prediction accuracy is achieved when the parent of X_3, that is X_4, is used as a predictor.

Second, as shown in Definition 1, BNTagger calculates the prediction accuracy for each SNP X_j using the posterior probability of X_j *given* the allele information of the htSNPs. To calculate this posterior, the product of the conditional probabilities in the BN must be computed as was shown in Equation (1). However, if the set of htSNPs contains no descendants of X_j and the parents of X_j are already in the set of htSNPs, the posterior probability is the same as the conditional probability parameter of X_j, due to the conditional independence encoded in the BN. For instance, in Figure 1(c), the best prediction accuracy for the SNP X_3 is simply the maximum of its conditional probability parameters, $Pr(X_3 \mid X_4)$, shown in Figure 1(b).

As a result, the conditional independence structure and the conditional probability parameters in the BN guide BNTagger to find a set of highly predictive htSNPs, and expedite the evaluation procedure. We note though that in order to use the BN components, BNTagger must first build them. Once the BN is constructed and the htSNPs are selected, we also provide a reconstruction framework for newly-genotyped samples; as mentioned earlier, the main purpose of prediction-based htSNP selection is to *reconstruct* the original set of SNP information based on the selected htSNPs.

To summarize, BNTagger consists of three stages: I. Identification of the conditional independence relations among SNPs; II. htSNP selection; and III. Reconstruction of haplotype information for newly-genotyped samples. In the first stage, BN learning is used to identify a graph structure, G, and a set of conditional probability parameters, Θ, that best explain the given haplotype data, D. In the second stage, a heuristic search is applied to the identified BN model to find a set of htSNPs. The third stage provides the haplotype reconstruction framework for subsequent association studies. These three stages are depicted in Figure 2, and are further described in the following subsections.

4.1 Identification of conditional independence relations among SNPs

To use a Bayesian network as described above, its structure and parameters must first be *learned*. We implemented the *Sparse Candidate* algorithm (Friedman *et al*., 1999), which accelerates BN learning by restricting the parents of each node to a small subset of candidates. To select candidate parents for each node, we use the non-random association among SNPs, known as linkage disequilibrium (LD). Disease-gene association studies are typically based on the assumption that LD exists between a disease allele and adjacent SNPs (Crawford and Nickerson, 2005), thus it is widely used for quantifying relationships between SNPs in population genetics. Numerous LD measures have been used. Among them, we use the multi-allelic[4] extension of Lewontin's linkage disequilibrium (LD) measure, D' (Hedrick, 1987), which is one of the most commonly used measures for multi-allelic SNPs (Aulchenko *et al*., 2003).

We explain it here in detail. Let X_1 be an m-allelic SNP, and X_2 be an n--allelic SNP. Let f_i^1 be the relative frequency of the i^{th} allele for SNP X_1, while f_j^2 be the relative frequency of the j^{th} allele for SNP X_2. Let f_{ij} be the relative joint frequency of the i^{th} allele occurring for SNP X_1 and the j^{th} allele occurring for SNP X_2 (where $i = 1, \ldots, m$ and $j = 1, \ldots, n$). Formally, the multi-allelic extension of Lewontin's LD, D', is defined as:

$$D' = \sum_{i=1}^{m} \sum_{j=1}^{n} f_i^1 \cdot f_j^2 \left| \frac{f_{ij} - f_i^1 \cdot f_j^2}{D_{max}} \right|,$$

where D_{max} is the maximum value of LD between the i^{th} and the j^{th} alleles. In principle, D' measures the difference between the observed (f_{ij}) and the

[3] The prediction indicator function P_f (Definition 1) is used in the equations in Figure 1(c).

[4] Most LD measures assume SNPs to have only two different alleles. Multi-allelic LD measures extend these bi-allelic LD measures, by allowing SNPs to have more than two different alleles.

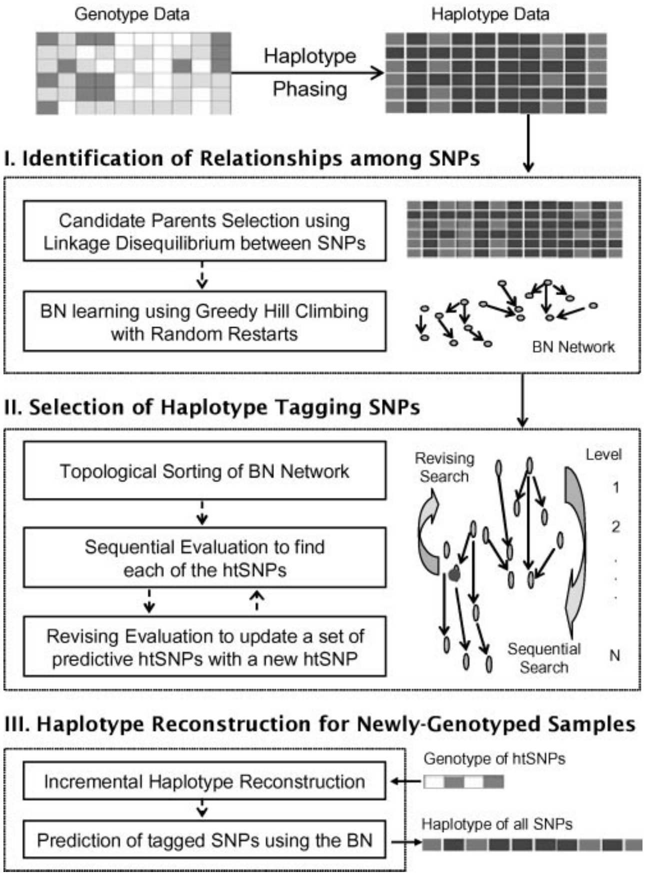

Genotype Data
Haplotype Data
Haplotype
Phasing
I. Identification of Relationships among SNPs
Candidate Parents Selection using Linkage Disequilibrium between SNPs
BN learning using Greedy Hill Climbing with Random Restarts
BN Network
II. Selection of Haplotype Tagging SNPs
Topological Sorting of BN Network
Sequential Evaluation to find each of the htSNPs
Revising Evaluation to update a set of predictive htSNPs with a new htSNP
Revising Search
Level
1
2
.
.
.
N
Sequential Search
III. Haplotype Reconstruction for Newly-Genotyped Samples
Incremental Haplotype Reconstruction
Prediction of tagged SNPs using the BN
Genotype of htSNPs
Haplotype of all SNPs

Fig. 2. Outline of haplotype tagging SNP selection and reconstruction in BNTagger.

expected frequency of haplotypes under independence ($f_i^1 \cdot f_j^2$), normalized by the maximum LD (D_{max}), and weighted by the expected joint frequency under independence ($f_i^1 \cdot f_j^2$).

Using the measure D', BNTagger first considers candidate parents for SNP X_j from the set $V - \{X_j\}$, whose pairwise disequilibrium with X_j, as measured by D', is in the top γ percent (here, $\gamma = 10$). The search for the optimal graph structure is performed using greedy hill climbing with random restarts. After N iterations ($N = 25{,}000$), we select the graph structure with the best MDL score (Lam and Bacchus, 1994). The conditional probability parameters $\Theta = \{\theta_1, \ldots, \theta_p\}$ are computed using maximum-likelihood estimation given the identified structure and the data.

4.2 Haplotype tagging SNP selection

Given the SNP-independence structure and the parameters constructed in the previous stage, we now identify a set of htSNPs, T, for the haplotype data, D. Since a different combination of htSNPs can be used to predict each tagged SNP, we also identify a set of predictive htSNPs, $T_{X_j} \subset T$, for each tagged SNP X_j.

As was demonstrated earlier, given the haplotype data, D, and the current set of htSNPs, T, we sequentially examine the average prediction accuracy for each SNP, X_j. If the prediction accuracy for the SNP X_j is smaller than a pre-specified threshold, α, X_j is added to the set of htSNPs, T. Otherwise, X_j is considered a tagged SNP, and the current htSNP set, T, is kept as its *candidate* set of predictive htSNPs, T_{X_j}. We call this procedure *sequential search*. When a new htSNP is added to T during the sequential search, we re-evaluate the prediction accuracy for previously examined tagged SNPs using the updated T. If the prediction accuracy for the re-examined tagged SNP is increased by using the new set T, its previously assigned candidate set of predictive htSNPs is updated to the new T. We call this procedure *revising search*.

In brief, BNTagger sequentially identifies a global set of htSNPs, T, based on their prediction accuracy, and iteratively updates the predictive set of htSNPs, T_{X_j}, for each tagged SNP, X_j. To efficiently conduct these procedures, BNTagger uses two heuristics. First, we topologically sort the nodes in the BN, which yields the *levels* of nodes as defined below, and conduct sequential search in this topological order.

Definition 3. *A level of node X_j in a Bayesian network is defined as*:

$$level(X_j) = \begin{cases} 1 & : if\ pa(X_j) = \phi; \\ \max\limits_{X_k \in pa(X_j)} (level(X_k)) + 1 & : otherwise. \end{cases}$$

The sequential search is conducted in the order of the levels from low to high. This way, the level of htSNPs in T is never greater than that of the currently examined node. As mentioned before, there are two advantages to this ordering: the value of child nodes depends primarily on the information of their parents, and when parents are htSNPs, the child's posterior probability is obtained directly from the network's parameters.

The second heuristic is for expediting the identification of predictive htSNPs for each tagged SNP. That is, if the current set of htSNPs, T, shows a prediction accuracy greater than a pre-specified threshold, β, for SNP X_j, we do not re-evaluate it any more. We formally define the current htSNP set T as the *prediction blanket* of X_j, and use it as the final set of predictive htSNPs for X_j. This second heuristic stems from an empirical observation that when the prediction accuracy for tagged SNP, X_j, given the current set T, is sufficiently high, new htSNPs often do not significantly improve the accuracy. This phenomenon was also observed by others (Ackerman *et al.*, 2003). Thus, it is typically unnecessary to examine the effect of every new htSNP on the tagged SNPs that are already well-predicted. The loss in accuracy is typically negligible. Moreover, the potential overfitting of predictive htSNP selection to the training data D is also reduced. Formally, we define the *prediction blanket* as follows:

Definition 4. *Given a prediction indicator function, P_f, and a constant β, the current set of htSNPs, $T = \{X_{t_1}, \ldots, X_{t_q}\}$, is defined as the prediction blanket of X_j if the average prediction accuracy for X_j, over all haplotypes D_{i-} given T is greater than β, that is*:

$$\left[\frac{1}{n}\sum_{i=1}^{n} P_f(X_j, T, D_{i-})\right] > \beta.$$

As a matter of fact, in a Bayesian network, re-evaluation can be avoided whenever T_{X_j} is the Markov blanket of X_j, as information about newly-added htSNPs does not affect the posterior probability of X_j given its Markov blanket. However, it is unlikely that *all* parents, *all* children, and *all* spouses of X_j (i.e., the complete Markov Blanket of X_j) will be included in the current htSNP set T, unless T is very large. Thus, our prediction blanket can be viewed as a relaxed version of the Markov blanket in the context of prediction. The selection algorithm is summarized in Table 1.

4.3 Reconstruction of newly-genotyped samples

The ultimate purpose of prediction-based htSNP selection is to reconstruct the information for all SNPs on a haplotype, using only the selected htSNPs in newly-genotyped samples (for instance, in new association studies). We propose a practical framework for this reconstruction. Our reconstruction algorithm takes *genotype* data of htSNPs as input, infers their resolving haplotypes[5] based on the previously used haplotype data set D, predicts

[5] As defined in the first paragraph of Section 2.

Table 1. BNTagger: Haplotype tagging SNP selection algorithm

```
$D$: training data ($n$ haplotypes with $p$ SNPs)
$P_f$: a prediction indicator function
$V$: a set of $p$ SNPs $\{X_1, X_2, \ldots, X_p\}$
$T$: a set of htSNPs $\{T_{t_1}, \ldots, T_{t_q}\}$

// predefined constants
$\alpha$: accuracy threshold for htSNPs
$\beta$: accuracy threshold for prediction blanket

level[$X_j$]: the level of $X_j$ in the BN
status[$X_j$]: the status of $X_j$
accuracy[$X_j$]: the prediction accuracy for $X_j$

Function SequentialSearch ($D$, $P_f$){/* Main function */
  $T = \phi$;
  $\forall_j$ status[$X_j$] = 'unchecked';
  $\forall_j$ accuracy[$X_j$] = 0;

  $L = \max_j$ level[$X_j$];
  for (each level $1 \leq l \leq L$)
    for (each node $X_j$ whose level is $l$)
      accuracy $= \frac{1}{n}\sum_{i=1}^{n} P_f(X_j, T, D_{i-})$;
      if (accuracy < $\alpha$)
        // add this node as an htSNP
        status[$X_j$] = 'htSNP';
        $T = T \cup \{X_j\}$;
        call RevisingSearch(level[$X_j$]);
      else if (accuracy > $\beta$)
        // the prediction blanket of $X_j$ is found
        status[$X_j$] = 'blanket_found';
        prediction_blanket[$X_j$] = $T$;
      else
        // store a candidate predictive htSNPs
        status[$X_j$] = 'tagged';
        prediction_blanket[$X_j$] = $T$;
        accuracy[$X_j$] = accuracy;
}

 Function RevisingSearch ($L$) {
   for (each node $X_k$
              whose level $\leq L$ and status = 'tagged')
      accuracy $= \frac{1}{n}\sum_{i=1}^{n} P_f(X_k, T, D_{i-})$;
      if(accuracy > $\beta$)
        status[$X_j$] = 'blanket_found';
        prediction_blanket[$X_k$] = $T$;
      else if (accuracy > accuracy[$X_k$])
        prediction_blanket[$X_k$] = $T$;
         accuracy[$X_k$] = accuracy;
}
```

the alleles of tagged SNPs using the Bayesian network model built in stage I, and outputs the *haplotype* information of *all* SNPs.

Suppose that our htSNP set T, as identified in stage II, consists of q SNPs, that is, $T = \{X_{t_1}, \ldots, X_{t_q}\}$. Let $g = (x_{t_{11}}/x_{t_{12}}, \ldots, x_{t_{q1}}/x_{t_{q2}})$ be a new *genotype*, consisting of the combined allele information of the q htSNPs. To deduce the haplotype information of g, we first select the most common haplotype in D, whose htSNP information is *compatible* with g. The *complementary mate* of the haplotype can then be automatically constructed. If we cannot find any haplotype compatible with g in D, we create a new haplotype whose alleles are assigned as the major allele for each heterozygous htSNP. Let h'_n be the new haplotype, and h'_{n_i} be its i^{th} element (where $i = 1, \ldots, q$). Given $g = (x_{t_{11}}/x_{t_{12}}, \ldots, x_{t_{q1}}/x_{t_{q2}})$ h_{n_i} can then be defined as:

$$h'_{n_i} = \begin{cases} x_{t_{i1}} & : \textit{if } x_{t_{i1}} = x_{t_{i2}}; \\ \underset{x \in \{x_{t_{i1}},\, x_{t_{i2}}\}}{argmax}\ Pr(X_{t_i} = x) & : \textit{otherwise}. \end{cases}$$

The prior probability, $Pr(X_{t_i})$, can be computed using our Bayesian network model. Again, its complementary mate can then be automatically constructed. In either case, the inferred two haplotypes for g are separately used for predicting the alleles of each tagged SNP. We call this procedure *incremental* haplotype reconstruction.

The principle of incremental haplotype reconstruction is based on Clark's parsimony approach (Clark, 1990). That is, it tries to resolve an ambiguous genotype using one of the *already identified* haplotypes. Moreover, rather than picking any compatible haplotype, it selects the most common one, since common haplotypes are the most likely candidates under the random mating assumption. Our haplotype reconstruction for the htSNP genotype thus follows the widely-used maximum parsimony approach. However, it differs from conventional algorithms in utilizing the *existing* haplotype information of *all* previously known SNPs, rather than directly phasing those in the genotype. We believe that utilizing this *prior* haplotype information is necessary. As noted earlier, haplotype phasing based on the set of htSNPs might not be as reliable as haplotype phasing based on the original set of SNPs due to the reduced linkage disequilibrium among htSNPs (Halperin *et al*., 2005).

Once the haplotype information of htSNPs is deduced, we use the same prediction rule introduced in Section 2 to predict the tagged SNPs. That is, the allele whose conditional probability is the highest given the alleles of the htSNPs is taken to be the allele for each tagged SNP. When multiple solutions exist, the most common allele of the tagged SNP is selected.

5 RESULTS

5.1 Evaluation methods

We compare the performance of our method with that of three state-of-the-art htSNP selection methods: 1) the Eigen2htSNP method based on principal component analysis (PCA) (Lin and Altman, 2004); 2) the Block-free method based on dynamic programming (Bafna *et al*., 2003; Halldörsson *et al*., 2004); and 3) the STAMPA method based on dynamic programming (Halperin *et al*., 2005). Lin and Altman (2004) tested Eigen2htSNP with two options: *varimax* and *greedy*, and predicted each tagged SNP using the *one* htSNP whose correlation coefficient with the tagged one is the highest. Bafna *et al*. (2003) and Halldörsson *et al*. (2004) tested the Block-free method with two window sizes: 21 and 13, and used the majority vote of htSNPs to predict each tagged SNP. Halperin *et al*. (2005) also relied on the majority vote of htSNPs for prediction, but unlike the previous two methods, they used the *genotype* data of htSNPs rather than haplotype data.

All these methods aim to select a set of highly predictive htSNPs for the unselected, tagged SNPs. Therefore, they have all been evaluated using prediction accuracy. Accordingly, this is the measure we use here for a fair comparison. We note that the published results (Bafna *et al*., 2003; Halldörsson *et al*., 2004; Lin and Altman, 2004; Halperin *et al*., 2005) were all based on different data sets. To compare BNTagger with each of these methods, we obtained the data set used to test each method, preprocessed it as described in the respective publication, and applied our algorithm to it. For evaluation, we use the same evaluation procedure used by each of the compared methods utilizing *leave-one-out* for the Block-free and the STAMPA methods (Bafna *et al*., 2003; Halldörsson *et al*., 2004; Halperin *et al*., 2005) and 10-*fold* cross

Table 2. Summary of test data sets

Data	Data Source	SNP No	Haplotype No	Phasing	Gene Diversity	LD (Std)	Recombination
ACE	Lin and Altman (2004)	52	22	PHASE	0.876	0.78 (0.34)	19.38%
LPL	Nickerson *et al.* (2000)	87	142	known	0.991	0.55 (0.35)	55.95%
IBD5-1	Lin and Altman (2004)	103	774	PHASE	0.981	0.53 (0.27)	94.3%
IBD5-2	Daly *et al.* (2001)	103	258	GERBIL	0.724	0.41 (0.23)	99.6%

validation for Eigen2htSNP (Lin and Altman, 2004), as described in the respective publications. As Lin and Altman (2004) did not provide their 10-fold split, we ran the 10-fold cross validation procedure 10 times, each using a randomized 10-way split, to ensure robustness. In all cases, the average prediction accuracy is used as the ultimate evaluation measure. The prediction performance of the compared methods for each data set was directly taken from their respective publications (Bafna *et al*., 2003; Halldórsson *et al*., 2004; Lin and Altman, 2004; Halperin *et al*., 2005).

5.2 Test data

Three public data sets, ACE (angiotensin converting enzyme) (Rieder *et al*., 1999; Lin and Altman, 2004), LPL (human lipoprotein lipase) (Nickerson *et al*., 2000; Bafna *et al*., 2003; Halldórsson *et al*., 2004), and IBD5 (inflammatory bowel disease 5) (Daly *et al*., 2001; Lin and Altman, 2004; Halperin *et al*., 2005) were used for evaluation. These data sets were previously used to test the three compared methods, as reported in their respective publications. We first analyzed the genetic characteristics of each data set based on: gene diversity, linkage disequilibrium, and recombination rate. The gene diversity, (i.e., the probability that two haplotypes chosen at random from the sample are different (Nei, 1987)), is measured by $(n/(n-1)) \cdot (1 - \sum_{i=1}^{k} p_i^2)$, where n is the total number of haplotypes, k is the number of distinct haplotypes, and p_i is the relative frequency of the i^{th} distinct haplotype. Linkage disequilibrium (LD) between SNPs is estimated by the multi-allelic extension of Lewontin's LD, D' as defined earlier (Hedrick, 1987), where the statistical significance of the standardized LD parameter is calculated using the χ^2 test with one degree of freedom. The recombination rate of each data set is measured by the four-gamete test (Hudson and Kaplan, 1985).

The first data set ACE (Rieder *et al*., 1999) contains 78 SNPs within a genomic region of $24Kb$ on chromosome $17q23$. Genotyping was done from 11 individuals. This data set was used by Lin and Altman to test Eigen2htSNP (Lin and Altman, 2004). Following their procedure, among the 78 original SNPs only 52 bi-allelic nonsingletons are analyzed. Partially due to the small number of SNPs and small sample size, this data set shows high average LD (0.78) and relatively low gene diversity (0.876). The recombination rate is also relatively low (19.38%).

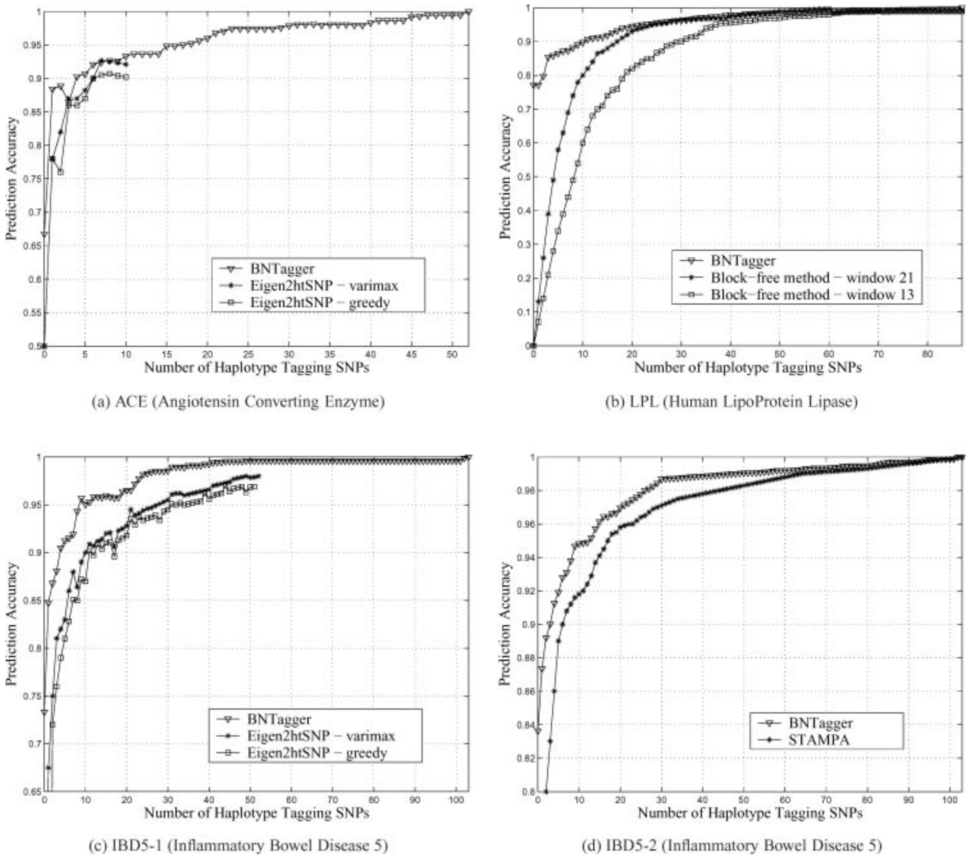
The second data set LPL (Nickerson *et al*., 2000), which was used by Bafna *et al*. (2003) and Halldórsson *et al*. (2004) to test the Block-free method, contains 88 SNPs spanning $5.5Kb$ on chromosome $19q13.22$. Genotyping was performed over 71 individuals. Following the analysis performed by Bafna *et al*. (2003), we analyze only 87 bi-allelic SNPs. Despite the small size of the LPL gene, this data set has high gene diversity (0.99) and low average LD (0.55), because it consists of haplotypes from three different populations. The four-gamete test shows 55.95% recombination or recurrent mutation.

The third data set, IBD5 (Daly *et al*., 2001) contains 103 SNPs on chromosome $5q31$, spanning $500Kb$. Genotyping was performed over 129 father-mother-child trios from a European population. This data set was used by Halperin *et al*. and by Lin and Altman to test the STAMPA (Halperin *et al*., 2005) and the Eigen2htSNP (Lin and Altman, 2004) methods, respectively. Lin and Altman (2004) analyzed data from all 387 individuals using PHASE (Stephens *et al*., 2001) for haplotype phasing. Halperin *et al*. (2005) analyzed data of only 129 individuals using GERBIL (Kimmel and Shamir, 2005) for haplotype phasing. Thus, following both of these two procedures, we created two separate data sets from IBD5, denoted as IBD5-1 (for Lin and Altman's) and IBD5-2 (for Halperin's). Both these sets have low linkage disequilibrium and high recombination rates. The summary of all data sets is given in Table 2.

5.3 Test results

We summarize the performance of BNTagger compared with the three state-of-the-art htSNP selection methods in Figure 3. We also compute the p-value of the difference in performance, using the Wilcoxon-ranksum test with 5% significance level. Overall, BNTagger consistently outperforms other methods on all data sets. Most importantly, improvement in prediction performance is most notable when the number of selected htSNPs is small, the average linkage disequilibrium in a data set is relatively low, and the gene diversity is high. This is a major advantage of BNTagger, since most htSNP selection methods have been known to suffer in those cases (Crawford and Nickerson, 2005; Johnson *et al*., 2001; Avi-Itzhak *et al*., 2003; Ao *et al*., 2005; Carlson *et al*., 2004). In other words, BNTagger retains its good performance even in what are considered to be hard cases.

The prediction performance of Eigen2htSNP (Lin and Altman, 2004) is compared with ours using two data sets: ACE and IBD5-1. For the first data set, ACE, Eigen2htSNP-varimax shows performance comparable to ours (see Figure 3(a); p-values are 0.2933 for varimax and 4.88×10^{-2} for greedy), but in the case of IBD5-1, its performance is considerably lower than ours, as shown in Figure 3(c) (p-values are 1.9489×10^{-6} for varimax and 1.5707×10^{-8} for greedy). The prediction performance of the Block-free method (Bafna *et al*., 2003; Halldórsson *et al*., 2004) is compared with ours using the LPL data set. Their performance increases substantially with the number of selected htSNPs, as shown in Figure 3(b), but the performance difference between ours and the Block-free method is significant when the number of htSNPs is smaller than 30 (p-values are 4.2×10^{-3} for window 21 and 1.2552×10^{-9} for window 13). The prediction performance of STAMPA (Halperin *et al*., 2005) is compared



(a) ACE (Angiotensin Converting Enzyme)

(b) LPL (Human LipoProtein Lipase)

(c) IBD5-1 (Inflammatory Bowel Disease 5)

(d) IBD5-2 (Inflammatory Bowel Disease 5)

Fig. 3. Prediction performance of BNTagger and the compared methods for test data sets.

Table 3. Prediction accuracy (in %) of BNTagger

Data Set	Percentage of Selected htSNPs				
	0%	5%	10%	25%	50%
ACE	66.7	86.5	92.1	93.7	97.4
LPL	77.2	86.6	89.0	95.0	98.3
IBD5-1	73.3	91.2	95.3	98.4	99.6
IBD5-2	83.6	91.9	94.9	98.0	99.0

with ours using the data set that Halperin *et al*. used, IBD5-2, as shown in Figure 3(d). Again, BNTagger outperforms STAMPA (p-value $= 0.7 \times 10^{-2}$), and the difference is significant as the number of htSNPs gets smaller (below 60).

Overall, as shown in Figure 3, our method uses a small fraction of SNPs as htSNPs (2.9%–11.5%) to achieve 90% prediction accuracy for all data sets: 4 htSNPs among 52 SNPs (7.7%) for data set ACE, 10 among 87 (11.5%) for LPL, 4 among 103 (3.9%) for IBD5-1, and 3 among 103 (2.9%) for IBD5-2. To achieve 95% prediction accuracy, we need 8.7%–32.7% of the target SNPs: 17 htSNPs among 52 SNPs (32.7%) for data set ACE, 22 among 87 (25.2%) for LPL, 9 among 103 (8.7%) for IBD5-1, and 13 among 103 (12.6%) for data set IBD5-2. Table 3 summarizes the prediction performance of BNTagger with respect to the percentage of the selected htSNPs.

As can be seen in Table 3, BNTagger can be reliably used even when the maximum number of htSNPs is very small. This is a major advantage of BNTagger. The explicit goal of htSNP selection is to save genotyping overhead, typically aiming at a 10–50 *fold* reduction in the number of target SNPs in the case of European samples (Palmer and Cardon, 2005). Thus, it is especially important to guarantee good prediction performance when the number of htSNPs is a small fraction of the total number of SNPs. We note that, unlike other methods, BNTagger can predict the allele information of all SNPs even without any htSNPs. In this case, the posterior probability of the predicted SNP X_j is the same as the prior probability of X_j. Thus, the prediction used by the function P_f, as shown in Definition 1, is still applicable even without selecting any htSNPs.

6 DISCUSSION

We presented BNTagger, a heuristic algorithm that uses the probabilistic framework of Bayesian networks to effectively identify a set of predictive htSNPs. BNTagger outperforms other state-of-the-art predictive methods when compared over their own data sets and prediction measure. Moreover, its improved performance is especially notable when a small number of htSNPs are selected. We believe that two main factors contribute to this improved performance:

(1) We do not restrict the htSNPs to any bounded location.

(2) We do not fix the number of htSNPs.

In addition, heuristics based on the conditional independencies among SNPs guide BNTagger to effectively find an improved set of htSNPs in terms of prediction accuracy.

Another major advantage of BNTagger is that, after the htSNPs are selected, it can directly reconstruct the *haplotype* information of newly-*genotyped* samples. BNTagger does not require prior haplotype phasing of htSNPs, which might not be reliable (Halperin *et al*., 2005). Instead, it deduces the haplotype information of the new sample based on the haplotype training data that was originally used for htSNP selection. In addition, BNTagger does not require SNPs to be bi-allelic nor does it assume prior block-partitioning. Nevertheless, it shows significant improvement in prediction performance for data sets with high gene diversity and relatively low linkage disequilibrium. Thus, we believe that BNTagger provides the most practical and comprehensive framework for htSNP selection, and can form a reliable basis for subsequent disease-gene association studies.

The improved performance of BNTagger comes at the cost of compromised running time. Currently, its running time varies from several minutes (when the number of SNPs is 52) to 2–4 hours (when the number is 103). Most of this time is spent on stage I, namely, learning the Bayesian network, rather than on htSNP selection or on haplotype reconstruction. As BNTagger does not partition the haplotype data (neither through blocks nor through a sliding-window[6]), it considers all SNPs at once. That is, the conditional independence structure among all SNPs is learned simultaneously, which substantially increases its running time as the number of SNPs increases. In practice, we argue that based on the clinical importance of disease-gene association studies (Crawford and Nickerson, 2005), improved prediction performance takes priority over running time—when the time is not prohibitively long. Nevertheless, our future research will focus on improving the speed of BNTagger, while minimizing loss in prediction performance. This will most likely involve the evaluation of alternative heuristics and optimization criteria. We also plan to provide BNTagger as an online service.

Currently, BNTagger does not directly set the number of selected htSNPs. Rather, it selects htSNPs based on their prediction accuracy compared to a predefined threshold (α). Thus, by adjusting this threshold, the number of selected htSNPs can be changed. We intend to revise our selection algorithm so that the number of htSNPs can be explicitly set, if needed. Finally, we used the multi-allelic extension of Lewontin's linkage disequilibrium (LD), D' (Hedrick, 1987), to expedite the learning procedure in stage I. We plan to apply other multi-allelic LD measures, and examine whether different measures affect the learned networks, the selected set of htSNPs, and their prediction performance.

ACKNOWLEDGEMENT

This work is supported by HS's NSERC Discovery grant 298292-04 and CFI New Opportunities Award 10437.

[6] Sliding-window-based algorithms confine the predictive htSNPs for each tagged SNP to the ones in the pre-defined neighborhood (i.e., sliding-window) of the tagged SNP (Meng *et al*., 2003).

REFERENCES

Ackerman,H. *et al*. (2003) Haplotype analysis of the TNF locus by association efficiency and entropy. *Genome Biol*., **4**, R24.1–13.

Ao,S.I. *et al*. (2005) CLUSTAG: hierarchical clustering and graph methods for selecting tag SNPs. *Bioinformatics*, **21**, 1735–1736.

Aulchenko,Y. *et al*. (2003) miLD and booLD programs for calculation and analysis of corrected linkage disequilibrium. *Ann Hum Genet*., **67**, 372–375.

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

Bafna,V. *et al*. (2003) Haplotypes and informative SNP selection algorithms: don't block out information. *In Proc. of Intl Conf Res Comp Mol Biol*., 19–27.

Carlson,C.S. *et al*. (2004) Selecting a maximally informative set of single-nucleotide polymorphisms for association analyses using linkage disequilibrium. *Am J Human Genet*., **74**, 106–120.

Clark,A.G. (1990) Inference of haplotypes from PCR-amplified samples of diploid populations. *Mol Biol Evo*., **7**, 111–122.

Cozman,F. (2000) Generalizing variable elimination in Bayesian networks. *In Proc. of the Workshop on Probabilistic Reasoning in Artificial Intelligence*, 27–32.

Crawford,D. and Nickerson,D. (2005) Definition and clinical importance of haplotypes. *Annu Rev Med*., **56**, 303–320.

Daly,M. *et al*. (2001) High-resolution haplotype structure in the human genome. *Nat Genet*., **29**, 229–232.

De Bakker,P.I.W. *et al*. (2006) Transferability of tag SNPs to capture common genetic variation in DNA repair genes across multiple populations. *In Proc. of Pac Symp Biocomput*., 478–486.

Friedman,N., Nachman,I. and Peér,D. (1999) Learning bayesian network structure from massive datasets: the ''sparse candidate'' algorithm. In *Proc. of the 15th Conference on Uncertainty in Artificial Intelligence (UAI),* 206–215.

Gabriel,S. *et al*. (2002) The structure of haplotype blocks in the human genome. *Science*, **296**, 2225–2229.

Greenspan,G. and Geiger,D. (2003) Model-based inference of haplotype block variation. *In Proc. of Intl Conf Res Comp Mol Biol*., 131–137.

Halldórsson,B.V. *et al*. (2004) Optimal haplotype block-free selection of tagging SNPs for genome-wide association studies. *Genome Res*., **14**, 1633–1640.

Halldörsson,B.V. *et al*. (2004b) A survey of computational methods for determining haplotyes. *Lecture Notes in Computer Science* **2983**, 26–47.

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.

Hedrick,P. (1987) Gametic disequilibrium measures: proceed with caution. *Genetics*, **117**, 331–341.

Hudson,R. and Kaplan,N. (1985) Statistical properties of the number of recombination events in the history of a sample of DNA sequences. *Genetics*, **111**, 147–164.

Jensen,F. (2002) *Bayesian networks and decision graphs*. In M. Jordan, S.L. Lauritzen, J.F. Lawless and V. Nair (eds), Springer-Verlag, New York.

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

Kimmel,G. and Shamir,R. (2005) GERBIL: genotype resolution and block identification using likelihood. *Proc. Natl Acad Sci*., **102**, 158–162.

Lam,W. and Bacchus,F. (1994) Learning bayesian belief networks: an approach based on the MDL principle. *Comp Intel*., **10**, 269–293.

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

Meng,Z. *et al*. (2003) Selection of genetic markers for association analyses using linkage disequilibrium and haplotypes. *Am J Human Genet*., **73**, 115–130.

Nei,M. (1987) *Molecular evolutionary genetics*. Columbia University Press, New York.

Nickerson,D. *et al*. (2000) Sequence Diversity and Large-Scale Typing of SNPs in the Human Apolipoprotein E Gene. *Genome Res*., **10**, 1532–1545.

Palmer,L. and Cardon,L. (2005) Shaking the tree: mapping complex disease genes with linkage disequilibrium. *Lancet*, **366**, 1223–1234.

Reich,D. *et al*. (2001) Linkage disequilibrium in the human genome. *Nature*, **411**, 199–204.

Rieder,M. *et al*. (1999) Sequence variation in the human angiotensin converting enzyme. *Nat Genet*., **22**, 59–62.

Stephens,M., Smith,N. and Donnelly,P. (2001) A new statistical method for haplotype reconstruction from population data. *Am J Human Genet*., **68**, 978–989.

Xing,E.P., Sharan,R. and Jordan,M.I. (2004) Bayesian haplotype inference via the Dirichlet process. In *Proc. of the 21st International Conference on Machine Learning,* 879–886.