Learning Predictive Interactions
Using
Information Gain and Bayesian Network Scoring

Xia Jiang [1]*, Jeremy Jao[1], Richard Neapolitan[2]

[1]Department of Biomedical Informatics, University of Pittsburgh, Pittsburgh, PA, 15213 USA

[2]Department of Preventive Medicine, Northwestern University Feinberg School of Medicine, Chicago, Il 60611USA

*Corresponding Author: xij6@pitt.edu

Keywords: Information gain, Bayesian network, epistasis, interaction, GWAS, SNP, Alzheimer's Disease, prediction, classification

Abstract

Background

The problems of correlation and classification are long-standing in the fields of statistics and machine learning, and techniques have been developed to address these problems. We are now in the era of high-dimensional data, which is data that can concern billions of variables. These data present new challenges. In particular, it is difficult to discover predictive variables, when each variable has little marginal effect. An example concerns *Genome-wide Association Studies* (*GWAS*) datasets, which involve millions of *single nucleotide polymorphism* (*SNPs*), where some of the SNPs interact epistatically to affect disease status. Towards determining these interacting SNPs, researchers developed techniques that addressed this specific problem. However, the problem is more general, and so these techniques are applicable to other problems concerning interactions. A difficulty with many of these techniques is that they do not distinguish whether a learned interaction is actually an interaction or whether it involves several variables with strong marginal effects.

Methodology/Findings

We address this problem using information gain and Bayesian network scoring. First, we identify candidate interactions by determining whether together variables provide more information than they do separately. Then we use Bayesian network scoring to see if a candidate interaction really is a likely model. Our strategy is called MBS-IGain. Using 100 simulated datasets and a real GWAS Alzheimer's dataset, we investigated the performance of MBS-IGain.

Conclusions/Significance

When analyzing the simulated datasets, MBS-IGain substantially out-performed nine previous methods at locating interacting predictors, and at identifying interactions exactly. When analyzing the real Alzheimer's dataset, we obtained new results and results that substantiated previous findings. We conclude that MBS-IGain is highly effective at finding interactions in high-dimensional datasets. This result is significant because we have increasingly abundant high-dimensional data in many domains, and to learn causes and perform prediction/classification using these data, we often must first identify interactions.

Introduction

We are in the era of high-dimensional or “big” data. We have data on the gene expression levels of thousands of gene which can be exploited to help provide personalized medical treatments [1]; we have data on millions of single nucleotide polymorphisms which can help us determine the genetic basis of disease [2]; we have abundant passive internet data which can be used for many purposes including learning an individual’s preferences [3] and detecting outbreaks [4]; we have abundant hospital data concerning workflow which can be used to determine good personnel combinations and sequencing [5].

There are several learning tasks involving data. The most straightforward task is simply to look for correlation. For example, we may test a drug versus a placebo, and perform a chi-square test to see if the desired health outcome is correlated with use of the drug. A related task is prediction. For example, we can analyze data on gene expression levels in breast cancer patients not only to see if certain genes are correlated with survival, but also to predict whether a given patient will survive [1]. Going a step further, we can learn a detailed model of the relationships among many variables to develop a rule-based expert systems [6] or a Bayesian network/influence diagram [7]. The model learned can then be used to perform numerous predictions and make decisions. Finally, we can use data to discover causal relationships among variables [8].

Numerous techniques have been developed to perform these learning tasks using low-dimensional data including linear regression and logistic regression, the perceptron [9], support vector machines [10], neural networks [11], and strategies for learning Bayesian network [7]. These techniques do not automatically handle high-dimensional data. However, some have been modified to do so. Techniques include regularized regression and lasso [12], which perform shrinkage. Furthermore, strategies such as ReliefF [13], have been developed to identify possible good predictors in high-dimensional datasets, which can then be provided to another method.

{Insert Table 1 about here}

These techniques require that a predictor of a given target has a relatively strong correlation with that target. So, if two or more predictors interact with little marginal effects, the predictors would not be discovered. Table 1 shows an interaction with little marginal effect. Variables *X* and *Y* are both trinary predictors of a binary target *Z*. The number next to each variable value shows the fraction of occurrence of that value in the population, and the entries in the table show the probability *Z* equals z_1 given each combination of the predictors. For example,

$$P(z_1 \mid \mathrm{x}_1, y_2) = 0.11.$$

We have

$$P(z_1 \mid y_1) = 0.0\times0.25 + 0.1\times0.5 + 0.0\times0.25 = 0.05$$

$$P(z_1 \mid y_2) = 0.11\times0.25 + 0.0\times0.5 + 0.11\times0.25 = 0.055$$

$$P(z_1 \mid y_3) = 0.0\times0.25 + 0.1\times0.5 + 0.0\times0.25 = 0.05.$$

So although *X* and *Y* together have a strong predictive strength for *Z*, *Y* exhibits little marginal effect. Similarly, *X* also exhibits little marginal effect. We say that *X* and *Y* *interact* to affect *Z*, where the interaction may or may not be causal.

Epistasis is the interaction of two more genes to affect phenotype. Biologically, epistasis is believed to occur when the effect of one gene is modified by one or more other genes. It is thought that much of genetic risk for disease is due to epistatic interactions with little or no marginal effects [14-17]. The advancement of high-throughput technologies has enabled *Genome Wide Association Studies* (*GWAS*). A *single nucleotide polymorphism* (*SNP*) results when a nucleotide that is typically present at a specific location on the genomic sequence is replaced by another nucleotide. These high dimensional GWAS datasets can concern millions of SNPs, and provide researchers unprecedented opportunities to investigate the complex genetic basis of diseases. By looking at single-locus associations using standard analyses such as chi-squared tests, researchers have identified over 150 risk loci associated with 60 common diseases and traits [18-21]. However, such analyses will miss SNPs that are interacting epistatically with little marginal effect. So, researchers endeavored to develop methods for learning epistasis from high-dimensional GWAS datasets. Traditional techniques such as *logistic regression* (*LR*) [22], *logistic regression with an interaction term* (*LRIT*) [23], penalized logistic regression [24] and Lasso [25] were applied to the task. Other techniques include *multifactor dimensionality reduction* (*MDR*) [26], *full interaction modeling* (*FIM*) [27], using *information gain* (*IG*) alone to investigate only 2-SNP interactions [28], *SNP Harvester* (*SH*) [29], the use of ReliefF [30], random forests [31], predictive rule inference [32], Bayesian *epistasis association mapping* (*BEAM*) [33], *maximum entropy conditional probability modeling* (*MECPM*) [34], and Bayesian network learning [35-37].

The problem of identifying interactions is not limited to SNPs. Predictors could be interacting in all of the situations discussed earlier. So this research on SNP-SNP interactions is applicable to all the problems involving high-dimensional datasets discussed above.

Next, we briefly discuss the *multiple beam search algorithm* (*MBS*), which was developed in [35], to illustrate a difficulty inherent in its methodology. When there are many possible predictors, MBS first uses a Bayesian network scoring criterion (discussed in the Methods Section) to identify the best set of possible predictors. It then initiates a beam from each member of this set. On this beam, it does greedy forward search, adding the predictor that increases the score the most, until no addition increases the score. It then greedily deletes the predictor such that the deletion increases the score the most, until no deletion increases the score. The final set of predictors is a candidate interaction.

There are two problems with this strategy. First, if two predictors each have a strong individual effect, then the model that includes both of them will have a high score and will end up being identified as an interaction, even if they do not interact at all. Second, if several predictors have very strong effects by themselves, they will be chosen on every beam, thereby blocking the beam from finding a lower scoring interacting SNP. The recently developed interaction discovery algorithm REGAL [37] repeatedly runs MBS, each time deleting the SNPs in the highest scoring model. This strategy addresses the second problem, but not the first one.

In this paper we again using MBS to investigate beams, and we use Bayesian network scoring to decide when to stop our search. However, we use information gain to decide which predictor to add instead of adding the predictor that most increases the score. We call the method *MBS-IGain*. We present the results of experiments using 100 simulated datasets comparing MBS-IGain to MBS, REGAL, and 7 other methods.

We note that no method, including ours, can overcome the “curse of dimensionality.” That is, if we have many possible predictors, it is not computationally possible to even investigate every two-predictor combination. We apply MBS-IGain to a real GWAS *late onset Alzheimer's disease*

(*LOAD*) data set that concerns data on 312,260 SNPs. In order to obtain computational feasibility, we first determine the 10,000 SNPs with the greatest marginal effect. So, if two SNPs interact with no marginal effect they would likely be missed. Regardless, we obtain interesting results in this real application. Another strategy for pre-processing SNP data is to use to restrict the search space by using knowledge about molecular pathways [38].

Methods

Our method utilizes Bayesian networks and information gain. So, first we review these two.

Bayesian Networks

A *Bayesian network* (*BN*) consists of a *directed acyclic graph* (*DAG*) $G = (V,E)$, whose nodeset V contains random variables and whose edges E represent relationships among the random variables, and a conditional probability distribution of each node X in V given each combination of values of its parents. Often the DAG is a causal DAG, which is a DAG containing an edge from X to Y only if X is a direct cause of Y [7].

Figure 1 shows a causal BN modeling the relationships among a small subset of variables related to respiratory diseases (developed using Netica). The values shown at each node are the alternatives and the prior probabilities of the alternatives (times 100). The value $h1$ means the patient has a smoking history and the value $h2$ means the patient does not. The other values have similar meaning.

Using a BN, we can determine conditional probabilities of interest with a BN inference algorithm [7]. For example, using the BN in Figure 1, if a patient has a smoking history (h_1), a positive chest X-ray (x_1), and fatigue (f_1), we can determine the probability of the individual having lung cancer. That is, we can compute $P(l_1 \mid h_1, x_1, f_1)$. Algorithms for exact inference in BNs have been developed [7]. However, the problem of doing inference in BNs is NP-hard [39]. So, approximation algorithms are often employed [7].

{Insert Figure 1 about here}

The task of learning a BN from data concerns learning both the parameters in a BN and the structure (called a DAG model). Specifically, a *DAG model* consists of a DAG $G = (V,E)$ where V is a set of random variables, and a parameter set θ whose members determine conditional probability distributions for G, but without specific numerical assignments to the parameters. The task of learning a unique DAG model from data is called *model selection*. As an example, if we had data on a large number of individuals and the values of the variables in Figure 1, we might be able to learn the DAG in Figure 1 from these data.

In the score-based structure learning approach, we assign a score to a DAG based on how well the DAG fits the data. Cooper and Herskovits [40] developed the Bayesian score, which is the probability of the data given the DAG. This score uses a Dirichlet distribution to represent prior belief for each conditional probability distribution in and contains hyperparameters representing these beliefs. The score is as follows:

$$score_{Bayes}(G : Data) = P(Data \mid G) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\sum_{k=1}^{r_i} a_{ijk})}{\Gamma(\sum_{k=1}^{r_i} a_{ijk} + \sum_{k=1}^{r_i} s_{ijk})} \prod_{k=1}^{r_i} \frac{\Gamma(a_{ijk} + s_{ijk})}{\Gamma(a_{ijk})}, \qquad (1)$$

where r_i is the number of states of X_i, q_i is the number of different instantiations of the parents of X_i, a_{ijk} is the ascertained prior belief concerning the number of times X_i took its k th value when the parents of X_i had their j th instantiation, and s_{ijk} is the number of times in the data that X_i took its k th value when the parents of X_i had their j th instantiation. The parameters a_{ijk} are known as hyperparameters.

When using the *Bayesian score* we often determine the values of the hyperparameters a_{ijk} from a single parameter α called the *prior equivalent sample size* [41]. If we want to use a prior equivalent sample size α and represent a prior uniform distribution for each variable in the network, for all i, j, and k we set $a_{ijk} = \alpha / r_i q_i$. In this case Equation 1 is as follows:

$$score_{\alpha}(G : Data) = P(Data \mid G) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha / q_i)}{\Gamma(\alpha / q_i + \sum_{k=1}^{r_i} s_{ijk})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha / r_i q_i + s_{ijk})}{\Gamma(\alpha / r_i q_i)}. \quad (2)$$

This version of the score is called the *Bayesian Dirichlet equivalent uniform* (*BDeu*) score.

The *Minimum Description Length (MDL)* score is based on information theory and tries to determine the model that minimizes the number of bits necessary to encode both the model and the data. Suzuki [42] developed the following version of this score:

$$score_{MDL}(G : Data) = d \frac{\log_2 m}{2} - m \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} P(x_{ik}, pa_{ij}) \log_2 \frac{P(x_{ik}, pa_{ij})}{P(x_{ik}) P(pa_{ij})}, \quad (3)$$

where d is the number of parameters necessary to store the probability distributions, m is the number of data items, r_i is the number of states of X_i, x_{ik} is the k th state of X_i, q_i is the number of instantiations of the parents of X_i, pa_{ij} is the j th instantiation of the parents PA_i of X_i, and the probabilities are computed using the data.

Note that when we are searching for the best model, we maximize the BDeu score but minimize the MDL score.

To learn a DAG from data we can score all DAGs using the Bayesian or MDL score and then choose the best scoring DAG. However, if the number of variables is not small, the number of candidate DAGs is forbiddingly large. Furthermore, the BN model selection problem has been shown to be NP-hard [43]. So heuristic algorithms have been developed to search over the space of DAGs during learning [7]. These heuristic algorithms assume each parent has a significant marginal effect on the child, and so they cannot learn BNs in which some variables interact with little marginal effect.

Information Gain

Information theory [44] is the discipline that deals with the quantification and communication of information. If Z is a discrete random variable with m alternatives, we define the *entropy*

$H(Z)$ as follows:

$$H(Z) = -\sum_{i=1}^{m} P(z_i)\log_2 P(z_i).$$

Shannon [44] showed that if we repeat n trials of the experiment having outcome Z, then the entropy $H(Z)$ is the limit as $n \to \infty$ of the expected value of the number of bits needed to report the outcome of each trial of the experiment. Entropy is a measure of our uncertainty in the value of Z since, as entropy increases, on the average it takes more bits to resolve our uncertainty. Entropy is maximized when $P(z_i) = 1/m$ for all i, and is minimized with value 0 when $P(z_i) = 1$ for some i.

The conditional entropy of Z given X is the expected value of the entropy of Z conditional on X. It is defined as follows (where X has k alternatives):

$$H(Z \mid X) = \sum_{j=1}^{k} H(Z \mid x_j)P(x_j),$$

By learning the value of X, we can reduce our uncertainty in Z. We define the *information gain* of Z relative to X as the expected reduction in the entropy of Z conditional on X:

$$IG(Z;X) = H(Z) - H(Z \mid X).$$

The conditional information gain of Z relative to X conditional on Y is the expected value of the information gain conditional on Y. It is as follows (where Y has l alternatives):

$$IG(Z;X \mid Y) = \sum_{i=1}^{l} IG(Z;X \mid y_i)P(y_i)$$

The following are some important properties of information gain:

1) $IG(Z;X) \geq 0$ with equality holding if and only if Z and X are independent.
2) $IG(Z;X) = IG(X;Z).$
3) $IG(Z;X,Y) = IG(Z;X \mid Y) + IG(Z;Y).$ (chain rule for information gain).
4) $IG(Z;X,Y) \geq IG(Z;Y).$

The 4th property follows easily from the 1st and 3rd ones.

Using Information Gain to Measure Interaction Strength

If we have three variables X, Y, and Z, we define the interaction strength of X and Y

relative to Z as follows:

$$IS(Z;X,Y) = IG(Z;X,Y) - IG(Z;X) - IG(Z;Y) .$$

We can generalize this definition to measure interaction strength of *X* and a set of variables A .

$$IS(Z;X,\mathrm{A}) = IG(Z;X,\mathrm{A}) - IG(Z;X) - IG(Z;\mathrm{A}) .$$

Technically since A is set, we should write $\{X\} \cup \mathrm{A}$ in the *IG* expression. However, we use the more succinct notation. Finally, we can define the interaction strength of the set A as follows:

$$IS(Z;\mathrm{A}) = IG(Z;\mathrm{A}) - \sum_{X \in \mathrm{A}} IG(Z;X) . \tag{4}$$

Interaction strength measures the increase in information gain obtained by considering X and A together relative to considering them separately. We can use the interaction strength to investigate interactions. The greater its value, the more indication we have that X and the variables in A are interacting to affect Z . In this way, we can discover variables that have little marginal effect while have a large effect together. Furthermore, we should be less likely to conclude that two non-interacting variables with strong individual effects are interacting because their individual information gains will be high, while their joint gain should not add much. In this next section we develop an algorithm for learning interactions that uses the interaction strength.

{Insert Figure 2 about here}

In general, the interaction strength can be positive or negative. Figure 2 shows three causal BN DAG models illustrating some of the possibilities. In Figure 2 (a), X and Z are independent conditional on Y . We then have by Properties (1) and (3) above

$$\begin{aligned} IG(Z;X,Y) &= IG(Z;X \mid Y) + IG(Z;Y) \\ &= 0 + IG(Z;Y) = IG(Z;Y). \end{aligned}$$

We therefore have by Property (1) above

$$\begin{aligned} IS(Z;X,Y) &= IG(Z;X,Y) - IG(Z;X) - IG(Z;Y) \\ &= IG(Z;Y) - IG(Z;X) - IG(Z;Y) \\ &= -IG(Z;X) \leq 0. \end{aligned}$$

On the other hand, in Figure 2(b) X and Y are independent causes of Z . If X and Y are independent, we have the following by Properties (1) and (3) above

$$\begin{aligned} IG(X;Z \mid Y) &= IG(X;Z,Y) - IG(X;Y) \\ &= IG(X;Z,Y). \end{aligned}$$

So, by Properties (2), (3), and (4)

$$\begin{aligned} IS(Z;X,Y) &= IG(Z;X,Y) - IG(Z;X) - IG(Z;Y) \\ &= IG(Z;X\,|\,Y) + IG(Z;Y) - IG(Z;X) - IG(Z;Y) \\ &= IG(X;Z,Y) - IG(Z;X) \\ &= IG(X;Z,Y) - IG(X;Z) \\ &\geq 0. \end{aligned}$$

In many applications the variables that we are investigating for interactions are known to be independent possible causes. For example, in a GWAS data set, the causal SNPs are often independent possible causes of the disease. In these cases we can be assured that the interaction strength is non-negative.

We stress that, in general, a high value of the interaction strength does not imply that the investigated variables are causes of the target. Consider the causal DAG model in Figure 2 (c). Since this complete DAG can represent any joint probability distribution of three variables, we could have the same interaction strength with this underlying causal mechanism as we would have when X and Y have a strong interactive causal effect on Z. So, in an agnostic search for interactions, we cannot assume that a discovered interaction is causal. For example, if we are searching for interactions when developing a BN DAG model and make every variable a target, we cannot assume that what appears to be a discovered interaction is a causal interaction.

Information Gain and the MDL Score

Suppose we have the simple DAG model in which there is a target *Z* with parent set *PA*. Then the local MDL score at node *Z* is as follows:

$$score_{MDL}(Z;PA\ :\ Data) = d\,\frac{\log_2 m}{2} - m\sum_{j=1}^{q}\sum_{k=1}^{r} P(z_k, pa_j)\log_2 \frac{P(z_k, pa_j)}{P(z_k)P(pa_j)},$$

where d is the number of parameters necessary to store the conditional distributions for *Z*, m is the number of data items, r is the number of states of *Z*, and q is the number of values the parents of *Z* can take. By $score_{MDL}(Z;PA\ :\ Data)$ we mean the MDL score of the model that has the variables in *PA* as the parents of *Z* and no other edges.

It is possible to show that

$$\sum_{j=1}^{q}\sum_{k=1}^{r} P(z_k, pa_j)\log_2 \frac{P(z_k, pa_j)}{P(z_k)P(pa_j)} = IG(Z;PA).$$

Note that in this context the *IG* also depends on *Data* since we use the data to compute the probabilities. However, we do not show that dependency. So, the MDL score is given by

$$score_{MDL}(Z;PA\ :\ Data) = d\,\frac{\log_2 m}{2} - m \times IG(Z;PA),$$

Recall that smaller scores are better. So, larger information gain improves the score.

Algorithm for Discovering Interactions

We assume that we have possible predictors X_i and a target variable Z. We want to find predictors that interact to affect Z. The algorithm that follows does this.

```
Algorithm *MBS-IGain*
Determine the set $Best$ of $n$ highest scoring predictors $X_i$ using $score(Z;X_i)$;
for each predictor $X_i \in Best$
    $G_i = \{X_i\}$;
    $flag = 0$;
    while $flag = 0$
        Determine predictor $X$ that maximizes $IS(Z;G_i,X)$;
        if
            $\frac{IS(Z;G_i,X)}{IG(Z;G_i)+IG(Z;X)} \le T$   or   $score(Z;G_i,X) < score(Z;G_i)$

            $flag = 1$;
        else
            add $X$ to $G_i$;
        endelse
    endwhile
endfor
Sort the $n$ models by $score(Z;G_i)$;
Output the sorted list;
```

The *score*(Z;G_i) in Algorithm *MBS-IGain* is either the MDL score or the BDeu score of the BN DAG model that has the variables in G_i as parents of our target variable Z. By *score*(Z:X_i) we mean only X_i is the parent of Z. This algorithm synergistically use Bayesian network scoring criteria and the *IS* and *IG* functions. First, if there are too many predictors to investigate all of them, the most promising predictors are selected using the scoring criterion. A beam is then started from each of these predictors. On each beam, we greedily select the predictor that has the highest *IS* with the set of predictors chosen so far. We end the search when either the *IS* is small relative to the individual *IG*s (as determined by a threshold *T*), or when adding the predictor decreases the score of the model. This latter exit criterion is important because we not only want to discover predictors that appear to be interacting, but we also want to discover likely models. Figure 3 illustrates the situation. Suppose our current model G_i is the one in Figure 3 (a), and we find that X_3 has the greatest interaction strength with variables already in the model, and the interaction is sufficiently strong that we exceed the threshold. If the model in Figure 3 (b) has a lower score than the one in Figure 3 (a), then it is less likely that all three nodes are parents of *Z*, and so we do not add X_3 in spite of the interaction strength being relatively large. On the other hand, the check for a sufficiently large *IS* is important because two or more SNPs could score very high as parents of *Z* when there is no evidence of an interaction. An example would be when *X* and *Y* each have strong causal strengths for *Z* but affect *Z* independently, as specified in the Noisy-OR model [3,7]. In this case the model in which *X* and *Y* both have edges directed at *Z* has a high score, even though there is no interaction.

Let *n* be the number of predictors investigated. There are *n* passes through the for-loop in MBS-Gain. In the worst case, there are $n(n\text{-}1)/2$ passes through the while-loop for each pass through the for-loop. So, technically the algorithm is $O(n^3)$. However, we stop developing each model after *M* predictors are added, where *M* is a parameter. In our experiments, M=4. So in practice the algorithm is $O(n^2)$.

{Insert Figure 3 about here}

We noted previously that the MDL score can be expressed in terms of *IG* as follows:

$$score_{MDL}(Z;PA\ :\ Data) = d\,\frac{\log_2 m}{2} - m \times IG(Z;PA),$$

Using this equality and some algebraic manipulation, it is possible to show the following (We do not show the dependence on *Data* for the sake of brevity):

$$IS(Z;X,\mathrm{A}) = \left(score_{MDL}(Z;X)\ + score_{MDL}\ (Z;\mathrm{A}) - score_{MDL}(Z;X,\mathrm{A})\right)/\,m + C$$

where *C* is a constant. Since *C* and *m* are the same for all predictors *X*, when doing our search in Algorithm *MBS-IGain*, we could just find the predictor that maximizes the following expression:

$$score_{MDL}(Z;X) + score_{MDL}(Z;\mathrm{A}) - score_{MDL}(Z;X,\mathrm{A})\,.$$

This result motivates investigating the use of the BDeu score in our search instead of the MDL score. That is, we find the predictor *X* that maximizes this expression:

$$\ln score_{\alpha}(Z;X,A) - \ln score_{\alpha}(Z;X) - \ln score_{\alpha}(Z;A)\,.$$

Jiang et al. [45] had better results discovering interactions using the BDeu score than using the MDL score. So, we might obtain improved results by using the BDeu score in our interaction strength formula.

Experiments

Chen et al. [46] generated datasets based on two 2-SNP interactions, two 3-SNP interactions, and one 5-SNP interaction, making a total of 15 causative SNPs. The effects of the interactions were combined using a Noisy-OR model [3,7]. The specific interacting SNPs were as follows:

1. {S1, S2, S3, S4, S5}

2. {S6, S7, S8}
3. {S9, S10, S11}
4. {S12, S13}
5. {S14, S15}

Three parameters were varied to create the interactions: 1) θ, which determined the penetrance; 2) β, which determined the minor allele frequency; and l, which determined the linkage disequilibrium of the true causative SNPs with the observed SNPs. See [46] for details concerning these parameters. For various combinations of these parameters, Chen et al. [46] developed datasets containing 1000 cases and 1000 controls. In our evaluation, we used the 100 1000-SNP datasets they developed with $\theta = 1$, $\beta = 1$, and $l = null$.

We compared the performance of MBS-IGain, MBS [35], and REGAL [37] using these 100 datasets. MBS is similar to MBS-IGain except that in the forward search it adds the predictor that increases the score rather than the interaction strength the most. It also does backward search deleting the predictor that increases the score the most. REGAL repeatedly runs MBS, each time deleting the predictors in the highest scoring interaction. By eliminates the predictors in the highest scoring interaction, we guarantee that in the next iteration they won't be chosen on any of the beams. In this way, lower scoring interacting predictors can be found in the subsequent iteration.

We ran all three methods with the MDL score and the BDeu score with α = 1, 4, 9, 54, and 128. For MBS-IGain we used threshold values of *T* = 0.01, 0.05. 0.1, 0.15, 0.2, 0.3, 0.4, 0.5. We configured REGAL to repeatedly run MBS 5 times. For all three methods, we limited the number of SNPs in a model to 5, which means at most *M*=4 SNPs were added on each beam.

The experiments were run on three computers. JLJ69-dt contains an Intel i7 4790k with 32GB of memory which runs on Linux Mint 17.1 and Oracle Java 8 64-bit. The two others, DBGAP and DBGAP2, contain identical two AMD Opteron 4280s per computer with 128GB of memory, 64GB per NUMA node, which run on Windows Server 2008 R2 and Oracle Java 8 64-bit. Each experiment involving the 100 datasets was run using a shell script to run a java-runtime file which iterated on the datasets in question, with each parameter on its own thread spread among the three computers. The script was run the same for each of the configurations mentioned above.

We compared the methods using the following two criteria:

Criterion 1: This criterion determines how well the interacting predictors (SNPs), are discovered without regard to whether the actual interactions themselves are discovered. It measures the frequency with which the interacting predictors are ranked among the first K predictors. First, the learned interactions are ordered by their scores. Then the predictors are ordered according to the first interaction in which each appears. Finally, the power according to criterion 1 is computed as follows:

$$Power_1(K) = \frac{1}{R \times M} \sum_{i=1}^{R} N_K(i)$$

where $N_K(i)$ is the number of interacting predictors appearing in the first K predictors for the *i*th dataset, *M* is the total number of interacting predictors, and R is the number of datasets. In our comparison experiments using the 100 1000 SNP datasets *M* = 15 and *R* = 100.

Criterion 2: This criterion measures how well each of the given interactions is discovered. Each interaction is investigated separately. The power is computed using the Jaccard index which is as follows:

$$Jaccard(A,B) = \frac{\#(A \cap B)}{\#(A \cup B)}.$$

This index is 1 if the two sets are identical and is 0 if they have no items in common. First, the learned interactions are ordered by their scores for each dataset *i*. Let M_j (*i*) denote the *j*th learned interaction in the *i*th dataset, and *C* denote the true interaction we are investigating. For each *i* and *j* we compute $Jaccard(M_j(i),C)$. Then we set

$$J_K(i,C) = \max_{1 \le j \le K} Jaccard(M_j(i),C)$$

Then the power according to criterion 2 is computed as follows:

$$Power_2(K,C) = \frac{1}{R \times M} \sum_{i=1}^{R} J_K(i,C)$$

where *M* is the total number of interacting predictors, and R is the number of datasets.

Reiman et al. [47] developed a GWAS *late onset Alzheimer's disease* (*LOAD*) data set that concerns data on 312,260 SNPs and contained records on 859 cases and 552 controls. We applied MBS-IGain to this dataset to investigate how well it can learn interactions in a real setting. In order to apply MBS-IGain to this dataset, it was necessary to first filter the SNPs because 312,260 is too many SNPs to handle. One of the 312,260 loci is the APOE gene; however, we still use the terminology “SNP” to refer to the loci. We filtered by choosing the 1000 top-scoring 1-SNP models, the 5000 top-scoring 1-SNP models, and the 10,000 top-scoring 1-SNP models, and ran MBS-IGain with each of these sets of SNPs. We also ran the top-scoring 100 SNP models with all the remaining SNPs. We used the BDeu Score with α = 4, the MDL score, and thresholds of 0.05 and 0.1. We limited the size of the models to 5, which means at most M=4 SNPs were added on each beam.

Ethics Statement

This research uses only simulated datasets and a real de-identified dataset. So it does not require IRB approval.

Results

Simulated Datasets

First, we compare the result of using MBS-IGain with the MDL score to its results using the BDeu score. For values of the threshold that are not extreme (0.05 to 2), the results were the best and were very similar. Furthermore for smaller values of α in the BDeu score (1, 4, 9), the results were the best and were very similar. We show the results that were obtained with T = 0.1 for both methods and with α = 4 for the BDeu score. Supplementary S1 Fig shows $Power_1(K)$ for $K \leq 100$ for the two scores. Supplementary S2 Fig shows $Power_2(K,C)$ for $K \leq 100$ for each of the 5 interactions for the two scores; Supplementary S2 Fig (f) shows the average of $Power_2(K,C)$ over all 5 interactions. We see that the BDeu score sometimes performed slightly better and the MDL score sometimes performed slightly better. Overall, the two scores yield very comparable results.

Next, we compare the results for MBS-IGain, REGAL, and MBS. Like MBS-IGain, the performances of REGAL and MBS were best when using the MDL score or the BDeu score with smaller values of α. The results we show are for when all three methods use the MDL score and MBS-IGain uses a threshold of $T = 0.1$. Figure 4 shows $Power_1(K)$ for $K \leq 100$ for the three methods. Figure 2 shows $Power_2(K,C)$ for $K \leq 100$ for each of the 5 interactions C for the three methods; Figure 5(f) shows the average of $Power_2(K,C)$ over all 5 interactions. Looking at Figure 4 and Figure 5 (f) we see that REGAL performs notably better than MBS both at locating the interacting SNPs and at identifying the interactions exactly. So, the strategy of removing interactions and repeatedly running MBS seems to pay off. However, MBS-IGain performs notably better than REGAL at both these tasks. So, it appears that a much better strategy than repeatedly running MBS is to use information gain to identify interactions and then using BN scoring to decide whether the learned interaction is a likely model. MBS-IGain locates over half of the interacting SNPs within the first 10 predictors identified, and locates almost 2/3 of them in the first 100 predictors.

{Insert Figures 4 and 5 about here}

We now discuss the results for each interaction individually to illustrate the advantage of using information gain and BN scoring instead of using BN scoring alone. We look at the interactions in the order in which MBS-IGain is capable of identifying them. Figure 5 (d) shows that MBS-IGain always locates interaction {S12,S13} exactly and it is the highest scoring interaction. That is, the average Jaccard Index for this interaction jumps to 1 at $K = 1$. Both REGAL and MBS are fairly good at locating this interaction early, but their Jaccard Indices are smaller (MBS actually performs better than REGAL for this interaction). The problem is that they include extra SNPs (ordinarily S6) in the model because the extra SNP increases the score. MBS-IGain does not add this extra SNP because it has weak interaction strength with {S12,S13}. Figure 5 (b) shows that MBS-IGain is good at locating interaction {S6,S7,S8} early, but REGAL and MBS are not. The problem is that they usually group S6 with {S12,S13}, and then often do not find S7 and S8 at all. We see from Figure 5 (c) that both MBS-IGain and REGAL are good at locating interaction {S9,S10,S11} early but MBS is not. According to that figure REGAL outperforms MBS-IGain slightly. However, this is a little misleading because MBS-IGain always ranks {S12,S13} first and often ranks {S6,S7,S8} second. REGAL does not do this as often. So it identifies {S9,S10,S11} slightly earlier. Figure 5 (a) reveals none of the methods are very good at identifying {S1,S2,S3,S4,S5}; however, overall MBS-IGain does best. Again, REGAL does slightly better early. We see from Figure 5 (e) that both REGAL and MBS cannot identify {S14,S15} at all, whereas MBS-IGain identifies it to a limited extent. This interaction is particularly difficult to learn because the penetrance for the interacting SNPs is only 0.07. By way of contrast, the penetrances for interaction {S12,S13} are 0.5 and 0.7. See [46] for the details of the 5 models.

Supplementary S1 Tab shows the average running times for the three methods when analyzing the 100 1000 SNP datasets. Not only did MBS-IGain perform substantially better than the other methods, but it also was the fastest. This result is apparent based on a simple analysis of the algorithms. MBS-IGain only does a forward search, whereas MBS does forward and backward search, and REGAL repeatedly runs MBS (in our experiments 5 times). Since the algorithms are quadratic-time, an extrapolation of the results for the MDL score in Supplementary S1 Tab indicates that it would take MBS-IGain about 12 days to handle 20,000 predictors, while it would take REGAL around two months. This time difference could be significant in a real search. The computation of the BDeu score appears to be faster than that of the MDL score based on Supplementary S1 Tab.

Figure 6 shows a comparison of MBS-IGain to 7 other methods according to performance Criterion 1. The 7 methods are as follows: *logistic regression* (*LR*) [22], *multifactor dimensionality reduction* (*MDR*) [26], *full interaction modeling* (*FIM*) [27], *information gain* (*IG*) [28], *SNP Harvester* (*SH*) [29], Bayesian *epistasis association mapping* (*BEAM*) [33], and *maximum entropy conditional probability modeling* (*MECPM*) [34]. The same 100 1000 SNP datasets were used in the comparison. That is, they are the datasets developed by Chen et al. [46] with $\theta = 1$, $\beta = 1$, and $l = null$. The results for the other 7 methods were obtained from Chen et al. [46]. As we see from Figure 6, MBS-IGain exhibited substantially better performance than the other 7 methods according to performance Criterion 1.

{Insert Figure 6 about here}

Six of the 8 methods produce a total ranking of all the SNPs while learning interactions. For these methods we can develop ROC curves. To obtain one point on the ROC curve, we determine the number *x* of true SNPs (ones involved in interactions) and the number *y* of false SNPs in the top *m* SNPs. Based on this value of *m*, the true positive rate (sensitivity) is *x*/15 and the false positive rate (1-specificity) is *y*/985. We repeat this calculation for many values of *m* spaced between 0 and 1000. Figure 7 shows the ROC curves for the 6 methods. MBS-Gain has a true positive rate of about 0.5 with a negligible false positive rate. SH has only around a 0.4 true positive rate with a negligible false positive rate. However, by the time we get to a false positive rate of 0.1, SH overtakes MBS-IGain, and SH is in turn overtaken by BEAM at a false positive rate of 0.4. So, if we can tolerate a 10% false positive rate, LH might be a better choice. However, ordinarily we want to keep the false positive rate much smaller than 10%.

{Insert Figure 7 about here}

Real Datasets

When analyzing the real LOAD GWAS dataset with MBS-IGain, the BDeu score discovered a total of 14,818 unique models and the MDL score discovered a total of 10,141 unique models. Figure 8a shows a histogram of the BDeu scores and Figure 8b shows the percentile distribution. We see that the vast majority of the BDeu score approximately follow a normal distribution; however, there are 14 outliers with much higher scores. Figure 8c shows a histogram of the MDL scores and Figure 8d shows the percentile distribution. Although the distribution of the MDL scores is not a close to being normal as that of the BDeu score, most of the scores are clumped together and there are 21 outliers. The outliers for both scores constitute notable findings.

{Insert Figure 8 about here}

Table 2 shows the notable interactions discovered by the BDeu score, and Table 3 shows the notable interactions discovered by the MDL score. For both scores the notable finding distribute into two groups. The first group contains higher-scoring interactions involving the APOE gene, and the second group contains lower-scoring interactions involving the APOC1 gene. APOE is the strongest genetic predictor of LOAD. Furthermore APOE and APOC1 are in linkage disequilibrium, and APOC1 predicts LOAD almost as well as APOE [48]. Our results indicate that every notable interaction includes one of them.

{Insert Tables 2 and 3 about here}

All the notable findings can provide LOAD researchers with candidate interactions which they can investigate further. We discuss some of the more interesting ones. The MDL score, with its larger DAG penalty, discovers two 2-locus models involving the GAB2 gene. A good deal of previous research has indicated that GAB2 and APOE interact to affect LOAD [47]. The MDL score also discovers several other 2-locus models containing APOE. The BDeu score, with its smaller DAG penalty, discovers as its highest scoring interaction, a 4-locus interaction containing GAB2, rs10510511, and rs197899; and discovers as its second highest scoring model, a 3-locus interaction containing GAB2, SPAG16, and APOE. The MDL score also discovers this latter interaction. The interaction strengths (*IS*) of these two models is much greater than the *IS* of the 2-SNP models. So, our results support that there might be other loci involved in the GAB2/APOE interaction. We know of no previous research indicating this. Furthermore, there is previous research indicating APOE and the SPAG16 gene interact to affect LOAD, but not with GAB2 [49].

{Insert Tables 4 and 5 about here}

Table 4 shows all the loci that appeared in learned interactions and their individual information gains. The third locus shown is the locus providing the most information gain other than APOE and APOC1. This locus does not appear in an interaction. We include it to show that, other than APOE and APOC1, no locus provides substantial information gain by itself. On the other hand several of the interactions provide substantially more information gain that APOE or APOC1 do individually. For example, the fifth model in Table 2 includes APOE and has an *IS* of 0.049. Recall that the IS is the information gain provided by the all the loci in the model taken together minus the sum of their individual information gain. Since the information gain provided by APOE by itself is 0.121 (see Table 4), this fifth model provides better than a 0.049/0.121 = 0.40 increase in information gain over APOE by itself. The seventh model in Table 3 includes APOC1 an also has an IS of 0.49. Since the information gain provided by APOC1 by itself is 0.080 (see Table 4), this seventh model provides better than a 0.049/0.080 = 0.61 increase in information gain over APOC1 by itself.

Table 5 lists the genes that appeared in learned interactions, and shows whether previous research indicated associations of those genes with LOAD. We see that many of our findings are new; however, we have also substantiated previous research.

Supplementary S2 Tab shows the running times in the various analyses performed in the LOAD study. These results seem to contradict those in Supplementary S1 Tab. That is, in the real analysis the MDL score seems more efficient that the BDeu score. The explanation would be that in the real study the MDL score usual ends its search much sooner due to its larger DAG penalty. Tables 2 and 3 show that the MDL score does learn smaller models.

Discussion

We presented MBS-IGain, a method for identifying interactive effects in high-dimensional datasets. Based on our experiments, MBS-IGain is highly effective at doing this, substantially exceeding other methods. The effectiveness of MBS-IGain is owing to its three components. First, if possible MBS-IGain initiates a beam from every predictor because we have no way of determining if it is involved in an interaction without investigating its effect in combination with other predictors. We noted that MBS-IGain should be able to handle 20,000 predictors in about 12 days. Nevertheless, in studies such as GWAS we often have millions of predictors. In these cases, we must prune the number down to a manageable size. In our investigation we did this by scoring all one-predictor models, and choosing the ones with the best score. Other strategies, such as the use of ReliefF [13] could be employed. Second, MBS-IGain uses information gain to determine whether to add a predictor on a given beam rather than using the score. In this way, we identify predictors that are interacting rather than merely identifying high scoring models. Third, MBS-IGain uses the score to end its search on each beam. In this, we only identify interactions that are likely models. These three components together are essential to the performance of MBS-IGain.

We applied MBS-IGain to a real LOAD dataset. We obtained new results, most notable are results indicating that there are more loci involved in the APOE/GAB2 interaction; and we also substantiated previous findings. These results not only substantiate the efficacy of MBS-IGain, but also provide LOAD researchers with avenues for further investigation.

In related research, Hu et al. [50] used information gain to link interacting SNPs in statistical epistasis networks. However, their technique was limited to investigating two-SNP interactions for the purpose of constructing a network. It did not search multiple beams, investigate higher-order interactions, or use the Bayesian score to judge whether an apparent interaction was a likely model.

Author Contributions

Conceived and designed the experiments: XJ. Performed the experiments: JJ. Analyzed the data: RN, XJ. Wrote the manuscript: RN, XJ.

Dataset Access

The simulated datasets used in the experiments were provided by the authors of [46]. Yue Wang (yuewang@vt.edu) has granted us permission to distribute them. They are available as Supplementary S1 Dataset. The real dataset was developed by the authors of [49]. Please contact Dietrich A. Stephan (dstephan@pitt.edu) concerning its availability.

References

1. Neapolitan RE, Jiang X (2015) Study of integrated heterogeneous data reveals prognostic power of gene expression for breast cancer survival. PLOS ONE.
2. Manolio TA, Collins FS (2009) The HapMap and genome-wide association studies in diagnosis and therapy. Annual Review of Medicine 60: 443-456.
3. Neapolitan RE, Jiang X (2007) Probabilistic methods for financial and marketing informatics, Burlington, MA; Morgan Kaufmann.
4. Mandel B, Culotta A, Boulahanis J, Stark D, Lewis, B, Rodrigue J (2012) A demographic analysis of online sentiment during hurricane Irene. Proceedings of the Second Workshop on Language in Social Media: 27-36.
5. Soulakis ND, Carson MB, Lee YJ, Schneider DH, Skeehan CT, Scholtens DM (2015) Visualizing collaborative electronic health record usage for hospitalized patients with heart failure. JAMIA. DOI: http://dx.doi.org/ 10.1093/jamia/ocu017.
6. Neapolitan RE (1989) Probabilistic reasoning in expert systems. NY, NY: Wiley.
7. Neapolitan RE (2004) Learning Bayesian networks. Upper Saddle River, NJ; Prentice Hall.
8. Spirtes P, Glymour C, Scheines R (2000) Causation, prediction, and search. Boston, MA; MIT Press.
9. Freund Y, Schapire RE (1999) Large margin classification using the perceptron algorithm. Machine Learning 37 (3): 277–296.
10. Cortes C, Vapnik VN (1995) Support-vector networks. Machine Learning 20(3): 273-297.
11. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70:489-501.
12. Chen SS, Donoho DL, Saunders MA (1998) Atomic decomposition by basis pursuit. SIAM Journal on Scientific Computing 20:33-61.
13. Marko RS, Igor K (2003) Theoretical and empirical analysis of Relief and ReliefF. Machine Learning Journal 53:23-69.
14. Galvin A, Ioannidis JPA, Dragani TA (2010) Beyond genome-wide association studies: genetic heterogeneity and individual predisposition to cancer. Trends in Genetics 26(3):132-141.
15. Manolio TA, Collins FS, Cox NJ, Goldstein DB, Hindorff LA, Hunter DJ, et al. (2009) Finding the missing heritability of complex diseases and complex traits. Nature 461:747-753.
16. Mahr B (2008) Personal genomics: The case of missing heritability. Nature 456:18-21.
17. Moore JH, Asselbergs FW, Williams SM (2010) Bioinformatics challenges for genome-wide association studies. Bioinformatics 26:445-455.
18. Manolio TA, Collins FS (2009) The HapMap and genome-wide association studies in diagnosis and therapy. Annual Review of Medicine 60:443-456.
19. Herbert A, Gerry NP, McQueen MB (2006) A common genetic variant is associated with adult and childhood obesity. Journal of Computational Biology 312:279-384.
20. Spinola M, Meyer P, Kammerer S, Falvella FS, Boettger MB, Hoyal CR, et al. (2001). Association of the PDCD5 locus with long cancer risk and prognosis in smokers. American Journal of Human Genetics 55: 27-46.
21. Lambert JC, Heath S, Even G, Campion D, Sleegers K, Hiltunen M. et al. (2009) Genome-wide association study identifies variants at CLU and CR1 associated with Alzheimer's disease. Nature Genetics 41:1094-1099.
22. Kooperberg C, Ruczinski I. 2005. Identifying interacting SNPs using Monte Carlo logic regression. Genet Epidemiol 28:157-170.

23. Agresti A. 2007. Categorical data analysis. 2nd edition. New York: Wiley.
24. Park MY, Hastie T. 2008. Penalized logistic regression for detecting gene interactions. Biostatistics 9:30-50.
25. Wu TT, Chen YF, Hastie T, Sobel E, Lange K. 2009. Genome-wide association analysis by lasso penalized logistic regression. Genome Analysis 25:714-721.
26. Hahn LW, Ritchie MD, Moore JH. 2003. Multifactor dimensionality reduction software for detecting gene-gene and gene-environment interactions. Bioinformatics 19:376-382.
27. Marchini J, Donnelly P, Cardon LR 2005. Genome-wide strategies for detecting multiple loci that influence complex diseases. Nature Genetics 37:413-417.
28. Moore JH, Gilbert JC, Tsai CT, Chiang FT, Holden T, Barney N, et al. 2006. A flexible computational framework for detecting characterizing and interpreting statistical patterns of epistasis in genetic studies of human disease susceptibility. J Theor Biol 241:252-261.
29. Yang C, He Z, Wan X, Yang Q, Xue H, Yu W 2009. SNPHarvester: a filtering-based approach for detecting epistatic interactions in genome-wide association studies. Bioinformatics 25:504-511.
30. Moore JH, White BC. 2007. Tuning ReliefF for genome-wide genetic analysis. In: Marchiori E, Moore JH, Rajapakee JC (Eds). Proceedings of EvoBIO 2007. Berlin: Springer-Verlag.
31. Meng Y, Yang Q, Cuenco KT, Cupples LA, Destefano AL, Lunetta KL 2007. Two-stage approach for identifying single-nucleotide polymorphisms associated with rheumatoid arthritis using random forests and Bayesian networks. BMC Proc 2007: 1 Suppl 1:S56.
32. Wan X, Yang C, Yang Q, Xue H, Tang NL, Yu W 2007. Predictive rule inference for epistatic interaction detection in genome-wide association studies. Bioinformatics 26(1):30-37.
33. Zhang Y, Liu JS. 2007. Bayesian inference of epistatic interactions in case control studies. Nature Genetics 39:1167-1173.
34. Miller DJ, Zhang Y, Yu G, Liu Y, Chen L, Langefeld CD, et al. 2009. An algorithm for learning maximum entropy probability models of disease risk that efficiently searches and sparingly encodes multilocus genomic interactions. Bioinformatics 25(19):2478-2485.
35. Jiang X, Barmada MM, Neapolitan RE, Visweswaran S, Cooper GF (2010) A fast algorithm for learning epistatic genomic relationships. AMIA 2010 Symposium Proceedings: 341-345.
36. Jiang X, Barmada MM, Cooper GF, Becich MJ (2011). A Bayesian method for evaluating and discovering disease loci associations. PLOS ONE 6(8): e22075.
37. Jiang X, Neapolitan RE (2015) LEAP: biomarker inference through learning and evaluating association patterns. Genetic Epidemiology 39(3):173-184.
38. Iossifov I, Zheng T, Baron M, Gilliam TC, Rzhetsky A (2008) Genetic-linkage mapping of complex hereditary disorders to a whole-genome molecular-interaction network. Genome Res. 18(7):1150-1162.
39. Cooper GF (1990) The computational complexity of probabilistic inference using Bayesian belief networks. Journal of Artificial Intelligence 42(2-3):393-405.
40. Cooper GF, Herskovits E. 1992. A Bayesian method for the induction of probabilistic networks from data. Machine Learning 9:309-347.
41. Heckerman D, Geiger D, Chickering D (1995) Learning Bayesian networks: the combination of knowledge and statistical data. Technical report MSR-TR-94-09. Microsoft Research.
42. Suzuki J (1999) Learning Bayesian belief networks based on the minimum description length principle: basic properties. IEICE Transactions on Fundamentals. E82-A:2237-2245.

43. Chickering M (1996) Learning Bayesian networks is NP-complete. In Fisher D, Lenz H (Eds). Learning from Data: Artificial Intelligence and Statistics V. NY, NY: Springer-Verlag.
44. Shannon CE (1948) A Mathematical theory of communication. The Bell System Technical Journal 27(3): 379-423.
45. Jiang X, Neapolitan RE, Barmada MM, Visweswaran S (2011) Learning genetic epistasis using Bayesian network scoring criteria. BMC Bioinformatics 12(89):1471-2105.
46. Chen L, Yu G, Langefeld CD, Miller DJ, Guy RT, Raghuram J, et al. (2011) Comparative analysis of methods for detecting interacting loci. BMC Genomics 12:344.
47. Rieman EM, Webster JA, Myers AJ, Hardy J, Dunckley T, Zismann VL, et al. (2007) GAB2 alleles modify Alzheimer's risk in APOE carriers. Neuron 54: 713-720.
48. Tycko B, Lee JH, Ciappa A, Saxena A, Li CM, Feng L, et al, (2004) APOE and APOC1 promoter polymorphisms and the risk of Alzheimer disease in African American and Caribbean Hispanic individuals. Arch Neurol 61(9): 1434–1439.
49. Turner SD, Martin ER, Beecham GW, Gilbert JR, Haines JL, Pericak-Vance MA, et al, (2008). Genome-wide Analysis of Gene-Gene Interaction in Alzheimer Disease. Abstract in ASHG 2008 Annual Meeting.
50. Hu T, Sinnott-Armstrong NA, Kiralis JW, Andrew AS, Karagas MR, Moore JH (2011) Characterizing genetic interactions in human disease association studies using statistical epistasis networks. BMC Bioinformatics 12: 364.

Supplementary Information

S1 Figure 1. $Power_1(K)$ for MBS-IGain using the MDL score with T = 0.1 and the BDeu score with α = 4 and T = 0.1.

S2 Figure 2. $Power_2(K,C)$ for MBS-IGain using the MDL score with T = 0.1 and the BDeu score with α = 4 and T = 0.1.

S3 Table 1. Average running times to process the 100 1000 SNP simulated datasets.

S4 Table 2. Running times for the LOAD study.

S5 Datasets. Simulated datasets used in the experiments.

Figure Captions

Figure 1 Bayesian network modeling relationships among respiratory diseases.

Figure 2 Causal BN DAG models.

Figure 3 If the DAG model in (b) has a lower score than the one in (a), we do not add node X_3.

Figure 4 Comparison of MBS-IGain, REGAL, and MBS all using the MDL score (with $T = 0.1$ for MBS-IGain) according to performance Criterion 1.

Figure 5 Comparison of MBS-IGain, REGAL, and MBS all using the MDL score (with $T = 0.1$ for MBS-IGain) according to performance Criterion 2.

Figure 6 Comparison of MBS-IGain to 7 other methods according to performance Criterion 1. SNP Harvester (SH), maximum entropy conditional probability modeling (MECPM), Bayesian epistasis association mapping (BEAM), logistic regression (LR), full interaction modeling (FIM), information gain (IG), multifactor dimensionality reduction (MDR).

Figure 7 ROC curves for MBS-IGain and 5 other methods. SNP Harvester (SH), Bayesian epistasis association mapping (BEAM), full interaction modeling (FIM), information gain (IG), multifactor dimensionality reduction (MDR).

Figure 8 Histograms and percentile distributions when determining interactive models from the LOAD dataset.

Tables

Table 1 An interaction with little marginal effect. Variables *X* and *Y* are both trinary predictors of a binary target *Z*. The number next to each variable value shows the fraction of occurrence of that value in the population, and the entries in the table show the probability *Z* equals z_1 given each combination of the predictors.

	x_1 (.25)	x_2 (.5)	x_3 (.25)
y_1 (.25)	0.0	0.1	0.0
y_2 (.5)	0.11	0.0	0.11
y_3 (.25)	0.0	0.1	0.0

Table 2 Interactions learned from the LOAD dataset using the BDeu score with α = 4. The third column shows gene on which the SNP resides if it is located in a gene; otherwise it shows the chromosome. The fourth column shows the BDeu score of the interaction, and the fifth column shows the interaction strength of the interaction (See Equation 4).

Rank	**Interaction**	**Genes**	**BDeu**	**IS**
1	rs10510511, rs197899, rs7115850, APOE	Chrome 3, Chrome 6, GAB2, APOE	-824.6	0.042
2	rs11895074, rs7115850, APOE	SPAG16, GAB2, APOE	-827.1	0.035
3	rs536128, rs7115850, APOE	CALN1, GAB2, APOE	-827.3	0.020
4	rs7101429, rs10510511, rs197899, APOE	GAB2, Chrome 3, Chrome 6, APOE	-828.7	0.039
5	rs11122116, rs16856748, rs734600, rs16992170, APOE	NPHP4, LRP2, EYA2, EYA2, APOE	-828.9	0.049
6	APOE	APOE	-836.3	0
7	rs41369150, rs2265264, rs11217838, rs4420638	FNDC3B, Chrome 10, ARHGEF12, APOC1	-861.8	0.049
8	rs7355646 , rs41369150, rs4420638	Chrome 2, FNDC3B, APOC1	-863.4	0.023
9	rs7355646, rs41528844, rs4420638	Chrome 2, ADAMTS16, APOC1	-865.6	0.018
10	rs2265264, rs4420638	Chrome 10, APOC1	-865.9	0.026
11	rs41369150 , rs4420638, rs6121360	FNDC3B, APOC1, TM9SF4	-866.1	0.019
12	rs10922885, rs7355646, rs4420638	Chrome 1, Chrome 2, APOC1	-867.3	0.023
13	rs41369150, rs4420638	FNDC3B, APOC1	-867.4	0.010
14	rs7355646, rs4420638	Chrome 2, APOC1	-868.6	0.012

Table 3 Interactions learned from the LOAD dataset using the MDL score. The third column shows the gene on which the SNP resides if it is located in a gene; otherwise it shows the chromosome. The fourth column shows the negative MDL score of the interaction, and the fifth column shows the interaction strength of the interaction (See Equation 4).

Rank	Interaction	Genes	MDL	IS
1	rs7115850, APOE	GAB2, APOE	160.5	0.013
2	rs197899, APOE	Chrome 6, APOE	158.3	0.009
3	rs1785928, APOE	ELP2, APOE	157.6	0.008
4	rs10793294, APOE	GAB2, APOE	156.0	0.012
5	rs41491045, APOE	Chrome 2, APOE	155.3	0.009
6	rs2057537, APOE	TCP11, APOE	155.2	0.007
7	APOE	APOE	154.7	0
8	rs12421071, APOE	Chrome 11, APOE	154.2	0.007
9	rs891159, APOE	ATG10, APOE	154.2	0.009
10	rs12674799, APOE	Chrome 8, APOE	153.4	0.010
11	rs1957731, APOE	Chrome 4, APOE	153.3	0.009
12	rs1389421, APOE	Chrome 11, APOE	153.0	0.007
13	rs986647, APOE	Chrome 4, APOE	153.0	0.009
14	rs17095891, rs8108841, APOE	Chrome 10, Chrome 19, APOE	152.8	0.016
15	rs2717389, rs10130967, APOE	Chrome 3, Chrome 14, APOE	140.7	0.022
16	rs898717, rs16975605, APOE	FRMD4A, Chrome 15, APOE	140.5	0.020
17	rs11895074, rs7115850, APOE	SPAG16, GAB 2, APOE	138.2	0.035
18	rs11676052, rs6719419, APOE	Chrome 2, Chrome 2, APOE	138.1	0.020
19	rs2265264, rs4420638	Chrome 10, APOC1	104.9	0.026
20	rs41369150, rs4420638	FNDC3B, APOC1	99.7	0.010
21	rs4420638	APOC1	97.1	0

Table 4 The loci involved in the 14 interactions learned using the BDeu score or the 21 interactions learned using the MDL Scored. The second column shows their rank when we score all 312,260 1-SNP models using the MDL score. The third column shows the information gain provided by the SNP by itself. The SNP in the third row is not in a learned interaction. It is included to show the highest scoring SNP other than APOE or APOC1.

Locus	**Rank**	**Info Gain**
APOE	1	0.121
rs4420638 (APOC1)	2	0.080
rs6784615	3	0.016
rs1785928	111	0.010
rs41528844	449	0.008
rs7355646	968	0.007
rs1389421	966	0.007
rs8108841	994	0.007
rs11895074	1085	0.007
rs7115850	1384	0.006
rs891159	1599	0.006
rs41369150	1781	0.006
rs898717	2731	0.006
rs17095891	2817	0.006
rs10510511	2901	0.006
rs16975605	3404	0.005
rs197899	3915	0.005
rs6719419	4219	0.005
rs536128	4841	0.005
rs2057537	4813	0.005
rs1957731	5781	0.005
rs6121360	6887	0.005
rs10130967	6901	0.005
rs986647	7820	0.005

rs10793294	7983	0.004
rs10922885	8045	0.004
rs2717389	8314	0.004
rs41491045	8527	0.004
rs12421071	8736	0.004
rs12674799	8741	0.004
rs11676052	8804	0.004
rs7101429	11208	0.004
rs11122116	13898	0.004
rs16992170	18747	0.004
rs11217838	20371	0.004
rs2265264	99260	0.001
rs16856748	108717	0.001
rs734600	138078	0.001

Table 5 The genes involved in the 14 interactions learned using the BDeu score or the 21 interactions learned using the MDL Scored. The second column whether previous research indicated whether they were involved in an interaction concerning LOAD, and the third column shows whether previous research indicated that they had an effect on LOAD when interactions were not considered.

Gene	**Prev. Int. Effect**	**Previous No Int. Effect**
APOE	Yes	Yes
APOC1	Yes	Yes
GAB2	Yes	No
SPAG16	Yes	No
CALN1	No	No
NPHP4	No	No
LRP2	No	Yes
EYA2	No	No
FNDC3B	No	No
ARHGEF12	No	No
ADAMTS16	No	Yes
TM9SF4	No	No
ELP2	No	Yes
TCP11	No	No
ATG10	Yes	No
ZNF77	No	No
FRMD4A	No	Yes