10-26-2018

An Optimization Approach to Epistasis Detection

Lizhi Wang
Iowa State University, lzwang@iastate.edu

Maryam Nikouei Mehr
Iowa State University, mnmehr@iastate.edu

An Optimization Approach to Epistasis Detection

Abstract
Epistasis refers to the phenomenon where the interaction of multiple genes affects a certain phenotype in addition to their individual additive effects. Similar epistatic effects are also ubiquitous in other application areas, such as gene-environment interactions, where a certain effect is triggered only when a particular combination of genes and environmental components is present. Epistasis detection has been recognized as a major challenge in the field of genetics. Previously proposed methods either focused on finding two-gene interactions using brute force enumeration or resorted to heuristic algorithms to search only a subset of the solution space. Herein we present an optimization approach that can identify the number of explanatory variables responsible for the epistasis as well as the exact combination of these variables. Results from simulation experiments using a soybean data set suggested that the proposed approach had a 95.5% chance of correctly detecting second-order to fifth-order epistases, which was a significant improvement over two alternative approaches in the literature.

Keywords
Bioinformatics, Epistatic effect, Multiple linear regression, Mixed integer linear programming, Optimization

Disciplines
Bioinformatics | Genetics | Operational Research

Comments
This is a manuscript of an article published as Wang, Lizhi, and Maryam Nikouei Mehr. "An Optimization Approach to Epistasis Detection." *European Journal of Operational Research* (2018). DOI: 10.1016/j.ejor.2018.10.032. Posted with permission.

Highlights

- The epistatic effect refers to the interactions of genes
- We defined a model to represent both additive and epistatic effects
- We present a model and an algorithm that can detect epistatic effects
- The effectiveness of the new approach has been validated with simulation experiments

An Optimization Approach to Epistasis Detection

Lizhi Wang[a,*], Maryam Nikouei Mehr[a]

[a]*Department of Industrial and Manufacturing Systems Engineering, Iowa State University, Ames, Iowa 50011, USA*

Abstract

Epistasis refers to the phenomenon where the interaction of multiple genes affects a certain phenotype in addition to their individual additive effects. Similar epistatic effects are also ubiquitous in other application areas, such as gene-environment interactions, where a certain effect is triggered only when a particular combination of genes and environmental components is present. Epistasis detection has been recognized as a major challenge in the field of genetics. Previously proposed methods either focused on finding two-gene interactions using brute force enumeration or resorted to heuristic algorithms to search only a subset of the solution space. Herein we present an optimization approach that can identify the number of explanatory variables responsible for the epistasis as well as the exact combination of these variables. Results from simulation experiments using a soybean data set suggested that the proposed approach had a 95.5% chance of correctly detecting second-order to fifth-order epistases, which was a significant improvement over two alternative approaches in the literature.

Keywords: Bioinformatics, epistatic effect, multiple linear regression, mixed integer linear programming, optimization

1. Introduction

The epistatic effect refers to the interaction of genes [1, 2]. This term was first used by Bateson [3] in 1909 to describe the phenomenon where an allele at one locus prevents the allele at another locus from manifesting its effect. More recently, geneticists and biologists have discovered higher-order epistasis for complex traits that involve five or more genes. For example, Taylor and Ehrenreich [4] discovered a colony morphology trait in yeast strains that was caused by genetic interactions of five or more loci.

In statistical terms, the epistatic effect can be defined in a multiple linear regression context where each explanatory variable makes an additive contribution to the response variable. The epistatic effect is the additional (positive or negative) effect that is triggered when a certain combination of genes take certain allelic variations simultaneously. For example, the height of a certain plant species may be influenced by three alleles each with two variants (A or a, B or b, and C or c). An epistasis may be defined as the phenomenon that plants with the combination of (A, B, c) are one inch taller than the sum of the individual effects of alleles A, B, and c.

Epistasis holds the key to many scientific discoveries in genetics. Ritchie et al. [5] identified three genes whose interactions are responsible for sporadic breast cancer. Combarros et al. [6] found 36 examples of significant epistasis in sporadic Alzheimer's disease. Witnessing the increasing capability of discovering

*Corresponding author
Email address: lzwang@iastate.edu (Lizhi Wang)

epistatic genes in sickle-cell anemia, Nagel pointed out that [7] such techniques have the potential of advancing therapeutic strategies that target epistatic genes in concert with the risks involved in individual patients.

Detecting the specific epistasis that triggers the epistatic effect is a notoriously challenging task. Mackay and Moore [8] summarized three challenges for this task. *First*, commonly used parametric statistical methods were not designed to detect interactions and often struggle to provide precise parameter estimation. *Second*, it requires prohibitive computational resources to enumerate all possible combinatorial epistasis candidates. For the example with three alleles and two variants (A or a, B or b, and C or c), if we know the epistasis is triggered by two alleles, then there are 12 possible solutions: (A, B), (A, b), (A, C), (A, c), (a, B), (a, b), (a, C), (a, c), (B, C), (B, c), (b, C), and (b, c). Without the information about the complexity of the epistasis, i.e., the number of explanatory variables involved, which could be either two or three alleles, then the total number of solutions becomes 20. In general, if the total number of explanatory variables is p, and the epistasis could involve between two to p variables, then the number of possible solutions is $(3^p - 2p - 1)$. For 30 explanatory variables, this number is 206 trillion, so it would be computationally prohibitive to enumerate all possible epistases to find the true one. *Third*, it would be much more challenging to validate epistasis models through combinatorial experiments, since the process of validating even a small number of epistases candidates through field or lab experiments is usually prohibitively expensive, time-consuming, and labor intensive.

A related problem in machine learning and statistics is feature selection [9, 10], which has interesting similarities and differences with epistasis detection. The problem of feature selection is concerned with selecting a subset from a larger set of explanatory variables to explain the response variable, with a goal of obtaining a simplified model. Although both problems deal with variable selection, they have different assumptions and objectives. Feature selection assumes that the response variable can be sufficiently explained by a parsimonious subset of significant variables, and the objective is to make a selection that includes significant variables and excludes insignificant ones; the cost of excluding a significant variable is usually much larger than that of including an insignificant variable. In contrast, epistatic detection assumes that a set of explanatory variables are already known to have individual additive effects on the response variable, but an unknown combination of a few explanatory variables has an additional epistatic effect when they take certain values, and the objective is to find such combination in its exact form; the cost of selecting a variable that does not belong to the epistasis is as high as failing to select one that does. In fact, the study of epistatic effects requires efficient algorithms for both feature selection and epistasis detection. The genotype data sets usually contain many thousands or even millions of genes, with thousands of observed phenotype responses. Feature selection can be used first to select a small subset of genes, and then an epistasis detection algorithm can be applied to detect the epistatic interactions among these significant genes. This two-step approach was used in our simulation study to test the effectiveness of our epistasis detection algorithm.

Multiple methods have been proposed for detecting epistasis in genome-wide two- or multiple-locus interactions. Most of these methods focused on second-order epistasis through exhaustive search [11, 12, 13, 14, 15]. Evans et al. [16] pointed out that even an exhaustive search of two-locus pairs across the genome would identify loci that would not have been identified using a single-locus search. Since many complex diseases have been associated with the interactions of multiple genes [17, 18, 19], several studies have targeted higher order epistasis [5, 20, 21, 22]. Almost all these studies resorted to heuristic algorithms with non-exhaustive search, due to the complexity of the problem. For example, the EDCF method [22] is based on the clustering of relatively frequent items; the algorithm in [21] runs k-means clustering algorithms on all single nucleotide polymorphisms (SNPs) and then applies information theory to examine the candidates

from each cluster; MegaSNPHunter [23] and SNPRuler [11] used machine learning techniques; Ritchie et al. [5] proposed the model-free and non-parametric MDR method, which first reduces the dimension of genotype predictors from n to one and then evaluates the one-dimension predictor for its ability to classify disease status through cross-validation and permutation testing; and Yang et al. [24] used a local search heuristic that swapped one pair of genes at a time, which was shown to outperform the Bayesian Epistasis Association Mapping method [25]. A summary of machine learning approaches for epistasis detection can be found in [26].

The approach we propose in this paper was designed to detect multi-way epistasis, and it exhaustively searches all possible solutions in an efficient manner by taking advantage of combinatorial optimization modeling and solution techniques. This model can detect not only the complexity of the epistasis, but also their exact combination that triggers the epistatic effect. We also designed a heuristic algorithm for solving data sets with large numbers of genes and observations. Both the optimization model and the new algorithm were tested in a simulation study using a soybean data set. Results suggested that the new algorithm was able to find the global optimal solution for epistasis detection in 3,821 out of 4,000 independent simulation repetitions, each within 600 seconds, which was a significant improvement over two alternative algorithms compared in the simulation experiments.

2. Proposed Epistasis Detection Approach

In this section, we propose our approach for detecting epistatic effects from explanatory and response variables. We first formally give the problem statement in Section 2.1, and then present in Section 2.2 a local search heuristic that finds a local optimal solution by swapping one pair of genes at a time. In Section 2.3, we cast the epistasis detection problem as an MILP model, and in Section 2.4, we design an algorithm for data sets with large numbers of genes and observations by combining the MILP model with feature selection and the local search heuristic.

2.1. Problem Statement

We start by reviewing the multiple linear regression model, which can be used to capture additive effects of individual genes:

$$y_i = \beta_0 + \sum_{j=1}^{p} X_{ij}\beta_j + \epsilon_i, \forall i \in \{1, ..., n\}. \quad (1)$$

The notations used in the model include:

- p: the number of explanatory variables (genes)
- n: the number of observations (individuals)
- X_{ij}: the explanatory variable j for observation i, which is assumed to be binary
- y_i: the response variable for observation i (observed phenotype)
- β_0: the intercept coefficient
- β_j: the additive effect coefficient, which is the differential effect of $X_{ij} = 1$ over $X_{ij} = 0$ for any i
- ϵ_i: the residual effect for observation i.

To capture the epistatic effects between two genes, say j_1 and j_2, beyond the additive effects, several studies [1, 27] have introduced interactions terms to the multiple linear regression model:

$$\begin{aligned} y_i &= \beta_0 + \sum_{j=1}^{p} X_{ij}\beta_j + X_{ij_1}X_{ij_2}\gamma_{11} + X_{ij_1}(1-X_{ij_2})\gamma_{10} \\ &\quad + (1-X_{ij_1})X_{ij_2}\gamma_{01} + (1-X_{ij_1})(1-X_{ij_2})\gamma_{00} + \epsilon_i, \forall i \in \{1,...,n\}, \end{aligned} \tag{2}$$

where $\gamma_{k_1k_2}$ is the magnitude of the epistatic effect triggered by the combination of $X_{ij_1} = k_1$ and $X_{ij_2} = k_2$. Although Model (2) can quantify the effects of interactions between the two genes, it cannot determine by itself which two genes have such interactions, and it does not apply to multi-way epistatic interactions.

We now define a more generalized model to represent the epistatic effect:

$$y_i = \beta_0 + \sum_{j=1}^{p} X_{ij}\beta_j + b \cdot I\left(\sum_{j=1}^{p} X_{ij}(\lambda_j - \mu_j) = \sum_{j=1}^{p} \lambda_j\right) + \epsilon_i, \forall i \in \{1,...,n\}. \tag{3}$$

Here, b is the magnitude of the epistatic effect; $I(\cdot)$ is the indicator function, which is equal to 1 if the statement in the parentheses is true and 0 otherwise; and λ_j and μ_j are binary variables that define the epistasis, which is triggered if and only if $X_{ij} = 1, \forall j \in \{j : \lambda_j = 1\}$ and $X_{ij} = 0, \forall j \in \{j : \mu_j = 1\}$, for any i. In this model, for a given set of observations $X \in \mathbb{B}^{n,p}$ and $y \in \mathbb{R}^n$, we are trying to infer not only β_0 and β_j but also b, λ_j, and μ_j for all $j \in \{1,...,n\}$. If the epistasis is present as defined in Model (3), then correctly inferring these parameters will lead to a smaller prediction error than a basic multiple linear regression Model (1) would. Compared with Model (2), Model (3) is able to reveal not only the complexity of the epistasis (the number of genes that are involved) but also the exact combination of diallelic variants that triggers the effect.

The key to solving Model (3) is λ and μ that represent the epistasis. Once these two variables are revealed, then solving for all other variables becomes as straightforward as solving a basic multiple linear regression model. First, we calculate $z_i = \begin{cases} 1 & \text{if } \sum_{j=1}^{p} X_{ij}(\lambda_j - \mu_j) = \sum_{j=1}^{p} \lambda_j \\ 0 & \text{otherwise} \end{cases}, \forall i \in \{1,...,n\}$, then we define $\hat{X} = [1_{n\times 1}, X, z]$, which can be used to estimate the response variables as $\hat{y} = \hat{X}(\hat{X}^\top X)^{-1}\hat{X}^\top y$; the other variables β_0, β, and b can also be calculated accordingly. We denote the root mean square error (RMSE) as a function of λ and μ as $\zeta(\lambda, \mu) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y_i} - y_i)^2}$, which evaluates the quality of the solution λ and μ. The true epistasis is expected to result in a lower RMSE than a false one.

2.2. Local Search Heuristic

In this section, we present a heuristic algorithm that searches for a local optimal solution to Model (3), which is similar with the PathSeeker algorithm [24] in essence but is more flexible. We will refer to this algorithm as $(\lambda^*, \mu^*) = \mathcal{L}(X, y, \lambda^0, \mu^0)$, where the input parameters of the algorithm include the data set (X, y) and an optional initial solution (λ^0, μ^0) and the output is a local optimal solution (λ^*, μ^*) that represents the epistasis. The algorithm iteratively swaps genes in and out of the incumbent solution, one pair at a time, to achieve the minimal RMSE. A major difference between our local search heuristic and the PathSeeker algorithm is that the latter assumes that the complexity of the epistasis is known, whereas the former assumes that we only know the upper bound, denoted as K, rather than the actual complexity of

the epistasis. In the incumbent solution, the local search heuristic always keeps track of K indices for the K potential genes responsible for the epistasis, and uses a dummy index 0 as a place holder when fewer than K genes are found to be responsible. As such, our local search heuristic is able to take an initial solution with the wrong complexity and find its way towards a local optimal solution to the complexity and composition of the epistasis. Details of this heuristic algorithm are described as follows.

Local search heuristic

Input parameters: $X \in \mathbb{B}^{N\times p}$, $y \in \mathbb{R}^{N\times 1}$, and optionally (λ^0, μ^0).

Output decisions: $(\lambda^\star, \mu^\star)$.

Start: If the optional initial solution (λ^0, μ^0) is provided, then use it as the incumbent solution: $(\lambda^\star = \lambda^0, \mu^\star = \mu^0)$, otherwise initialize the incumbent solution $(\lambda^\star, \mu^\star)$ as two random binary vectors such that $\lambda_j^\star + \mu_j^\star \leq 1, \forall j \in \{1, ..., p\}$ and $\sum_{j=1}^{p}(\lambda_j^\star + \mu_j^\star) = K$. Evaluate its RMSE $\zeta(\lambda^\star, \mu^\star)$. Define $\mathcal{J} = \{j : \lambda_j^\star + \mu_j^\star = 1\}$ and, if necessary, extend its cardinality to K by adding $K - \sum_{j=1}^{p}(\lambda_j^\star + \mu_j^\star)$ elements of $\{0\}$.

while $\mathcal{J} \neq \emptyset$ **do**

  for $j \in \{1, ..., p\} \backslash \mathcal{J}$ **do**

    For all $k \in \{1, ..., p\}$, define $\hat{\lambda}_k^j = \begin{cases} 1 & \text{if } k = j \\ 0 & \text{if } k = \mathcal{J}_1 \\ \lambda_j^\star & \text{otherwise} \end{cases}$, $\hat{\mu}_k^j = \begin{cases} 0 & \text{if } k = j \\ 0 & \text{if } k = \mathcal{J}_1 \\ \mu_k^\star & \text{otherwise} \end{cases}$, $\bar{\lambda}_k^j = \begin{cases} 0 & \text{if } k = j \\ 0 & \text{if } k = \mathcal{J}_1 \\ \lambda_j^\star & \text{otherwise} \end{cases}$,

    and $\bar{\mu}_k^j = \begin{cases} 1 & \text{if } k = j \\ 0 & \text{if } k = \mathcal{J}_1 \\ \mu_k^\star & \text{otherwise} \end{cases}$, where $\mathcal{J}_1$ refers to the first element in the set $\mathcal{J}$. Evaluate $\zeta(\hat{\lambda}^j, \hat{\mu}^j)$ and $\zeta(\bar{\lambda}^j, \bar{\mu}^j)$.

  end for

  Evaluate $\zeta(\lambda^0, \mu^0)$, where $\lambda_k^0 = \begin{cases} 0 & \text{if } k = \mathcal{J}_1 \\ \lambda_k^\star & \text{otherwise} \end{cases}$, $\mu_k^0 = \begin{cases} 0 & \text{if } k = \mathcal{J}_1 \\ \mu_k^\star & \text{otherwise} \end{cases}$, $\forall k \in \{1, ..., p\}$.

  Evaluate $\zeta(\lambda^1, \mu^1)$, where $\lambda_k^1 = \begin{cases} \mu_{\mathcal{J}_1}^\star & \text{if } k = \mathcal{J}_1 \\ \lambda_k^\star & \text{otherwise} \end{cases}$, $\mu_k^1 = \begin{cases} \lambda_{\mathcal{J}_1}^\star & \text{if } k = \mathcal{J}_1 \\ \mu_k^\star & \text{otherwise} \end{cases}$, $\forall k \in \{1, ..., p\}$.

  if $\min_{j\in\{1,...,p\}} \{\zeta(\hat{\lambda}^j, \hat{\mu}^j), \zeta(*, \bar{\mu}^j), \zeta(\lambda^0, \mu^0), \zeta(\bar{\lambda}^1, \mu^1)\} < \zeta(\lambda^\star, \mu^\star)$ **then**

    Update the incumbent solution $(\lambda^\star, \mu^\star)$ with the one that achieved the smallest RMSE.

    Update $\mathcal{J} = \{j : \lambda_j^\star + \mu_j^\star = 1\}$ and, if necessary, extend its cardinality to K by adding $K - \sum_{j=1}^{p}(\lambda_j^\star + \mu_j^\star)$ elements of $\{0\}$.

  else

    Remove the first element in $\mathcal{J}$: $\mathcal{J} \leftarrow \mathcal{J} \backslash \mathcal{J}_1$.

  end if

end while

2.3. Optimization Model

We cast Model (3) as the following mixed integer optimization problem.

$$\min_{\beta_0,\beta,b,\lambda,\mu,z} \quad \sum_{i=1}^{n} \left| y_i - \left(\beta_0 + \sum_{j=1}^{p} X_{ij}\beta_j + b \cdot z_i \right) \right| \tag{4}$$

$$\text{s.t.} \quad \sum_{j=1}^{p} X_{ij}(\lambda_j - \mu_j) \geq -p(1 - z_i) + \sum_{j=1}^{p} \lambda_j \quad \forall i \in \{1, ..., n\} \tag{5}$$

$$\sum_{j=1}^{p} X_{ij}(\lambda_j - \mu_j) \leq \sum_{j=1}^{p} \lambda_j - 1 + p \cdot z_i \quad \forall i \in \{1, ..., n\} \tag{6}$$

$$\lambda_j + \mu_j \leq 1 \quad \forall j \in \{1, ..., p\} \tag{7}$$

$$\lambda_j, \mu_j, z_i \in \{0, 1\} \quad \forall i \in \{1, ..., n\}, \forall j \in \{1, ..., p\} \tag{8}$$

$$\beta_0, \beta_j, b \text{ free} \quad \forall j \in \{1, ..., p\}. \tag{9}$$

Here, the Objective (4) is to minimize the sum of positive and negative errors between predicted responses and actual observations. We use this objective function, as opposed to the commonly used RMSE in linear regression models [28, 29, 30], because it can be linearized and is computationally more tractable. Constraints (5) and (6) jointly define a binary variable z_i that checks the presence of the epistatic effect in each observation: $z_i = 1$ if and only if $\sum_{j=1}^{p} X_{ij}(\lambda_j - \mu_j) = \sum_{j=1}^{p} \lambda_j$, for all $i \in \{1, ..., n\}$. Constraint (5) ensures the "only if" direction: if $z_i = 1$, then $\sum_{j=1}^{p} X_{ij}(\lambda_j - \mu_j) \geq \sum_{j=1}^{p} \lambda_j$; since $\sum_{j=1}^{p} X_{ij}(\lambda_j - \mu_j) \leq \sum_{j=1}^{p} X_{ij}\lambda_j \leq \sum_{j=1}^{p} \lambda_j$ is always true, it leads to the equation $\sum_{j=1}^{p} X_{ij}(\lambda_j - \mu_j) = \sum_{j=1}^{p} \lambda_j$. Conversely, Constraint (6) enforces the "if" direction: if $\sum_{j=1}^{p} X_{ij}(\lambda_j - \mu_j) \geq \sum_{j=1}^{p} \lambda_j$, then $z_i = 1$. Constraint (7) means that the epistasis cannot logically require the jth explanatory variable to be one ($\lambda_j = 1$) and zero ($\mu_j = 1$) at the same time. Constraints (8) and (9) define the appropriate types of the decision variables.

Due to the absolute value function and the bilinear term $b \cdot z_i$ in the objective function, Model (4)-(9) is a nonlinear non-convex combinatorial optimization problem, generally hard to solve. In the following, we reformulate Model (4)-(9) into an equivalent mixed integer linear program (MILP) by introducing three sets of new variables. For all $i \in \{1, ..., n\}$, non-negative variables e_i^+ and e_i^- denote the positive and negative part of the prediction error for observation i, respectively, and the variable w_i is defined to be equal to $b \cdot z_i$. We also assume that we know the upper and lower bounds of the magnitude of the epistatic effect: $\underline{b} \leq b \leq \overline{b}$, which is necessary to linearize the bilinear term $b \cdot z_i$. We will refer to the following MILP (10)-(18)

as $\mathcal{M}(X, y)$, with X and y being observation parameters.

$$\min_{\beta_0,\beta,b,e_i^+,e_i^-,\lambda,\mu,z,w} \quad \sum_{i=1}^{n}(e_i^+ + e_i^-) \tag{10}$$

$$\text{s.t.} \quad y_i - \left(\beta_0 + \sum_{j=1}^{p} X_{ij}\beta_j + w_i\right) = e_i^+ - e_i^- \quad \forall i \in \{1, ..., n\} \tag{11}$$

$$w_i \leq \bar{b} z_i \qquad \forall i \in \{1, ..., n\} \tag{12}$$

$$w_i \geq \underline{b} z_i \qquad \forall i \in \{1, ..., n\} \tag{13}$$

$$w_i \leq b - \underline{b}(1 - z_i) \qquad \forall i \in \{1, ..., n\} \tag{14}$$

$$w_i \geq b - \bar{b}(1 - z_i) \qquad \forall i \in \{1, ..., n\} \tag{15}$$

$$\text{Constraints (5)-(8)} \tag{16}$$

$$e_i^+, e_i^- \geq 0 \qquad \forall i \in \{1, ..., n\} \tag{17}$$

$$\beta_0, \beta_j, b, w_i \text{ free} \qquad \forall i \in \{1, ..., n\}, \forall j \in \{1, ..., p\}. \tag{18}$$

Here, the Objective (10) is to minimize the positive and negative errors between predicted and actual observations, which is equivalent to Objective (4). These two error terms are defined in Constraint (11), at least one of which must be zero in the optimal solution (otherwise the objective value would have been improved by reducing the two error terms simultaneously). Constraints (12)-(15) are equivalent to the equation $w_i = b \cdot z_i$ when $\underline{b} \leq b \leq \bar{b}$, which is a commonly used reformulation linearization technique [31]. Constraints (17) and (18) define the appropriate types of the decision variables.

2.4. Algorithm for Large Data Sets

In theory, the MILP Model (10)-(18) can be used to detect epistasis for data sets of any dimensions. In practice, however, solving such model using the existing branch-and-bound algorithms [32] and solvers becomes increasingly time-consuming as the dimensions of the problem grow. This is a particularly relevant restriction in the big data era, when high throughput genotyping and phenotyping technologies are able to collect data at increasingly high efficiency. To address this challenge, a feature selection algorithm can be used to reduce the number of genes to a few dozen (assuming the validity of a parsimonious model). In the following, we introduce a new algorithm for solving Model (10)-(18) for a few dozen genes and a large number of observations, say N.

The idea of the algorithm is to iteratively solve Model (10)-(18) with small samples of observations and refine its solutions with the local search heuristic. The algorithm starts by initializing an incumbent solution (λ^*, μ^*) and goes through two iterative steps. In Step 1, Model (10)-(18) is solved on a small subset of observations to provide a candidate epistasis solution, which is refined in Step 2 using the local search heuristic. This candidate epistasis will replace the incumbent solution if it achieves a lower RMSE. The Step 1 and Step 2 loop continues until the stopping criteria are met, which could be defined based on the RMSE of the incumbent solution, the number of iterations, or computation time. Details of this algorithm are described as follows and diagrammed in Figure 1.

New Algorithm
Input parameters: $X \in \mathbb{B}^{N \times p}$ and $y \in \mathbb{R}^{N \times 1}$.
Output decisions: $\lambda^* \in \mathbb{B}^{p \times 1}$ and $\mu^* \in \mathbb{B}^{p \times 1}$.

Start: Determine the sample size n. Initialize the iteration counter $t = 0$ and the incumbent solution $(\lambda^* = 0^{p\times 1}, \mu^* = 0^{p\times 1})$. Evaluated the RMSE of the incumbent solution as $\zeta(\lambda^*, \mu^*)$. Go to Step 1.
Step 1: Detection. Increase t by 1: $t \leftarrow t + 1$. Randomly select a set $\mathcal{N}$ such that $\mathcal{N} \subseteq \{1, 2, ..., N\}$ and $|\mathcal{N}| = n$. Solve model $\mathcal{M}(X_{\mathcal{N}}, y_{\mathcal{N}})$, where $X_{\mathcal{N}}$ is a matrix with those rows from the input X whose indices belong to the set $\mathcal{N}$, and ditto for $y_{\mathcal{N}}$. Let (λ^t, μ^t) denote part of the optimal solution from $\mathcal{M}(X_{\mathcal{N}}, y_{\mathcal{N}})$. Go to Step 2.
Step 2: Refinement. Refine the solution using the local search heuristic $(\lambda^t, \mu^t) = \mathcal{L}(X, y, \lambda^t, \mu^t)$. If $\zeta(\lambda^t, \mu^t) < \zeta(\lambda^*, \mu^*)$, then update the incumbent (λ^*, μ^*) as (λ^t, μ^t).
Check point: If Stopping criteria are met then finish the algorithm; otherwise go back to Step 1.

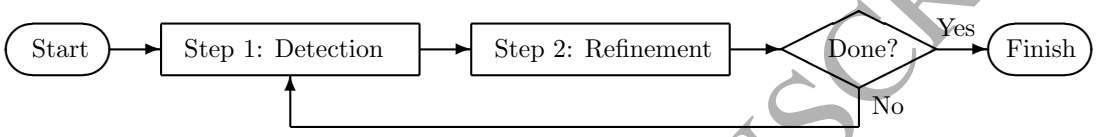

Figure 1: Diagram of the new algorithm for epistasis detection.

Remark 1: At the starting point, the sample size n should be small enough to make model $\mathcal{M}(X_{\mathcal{N}}, y_{\mathcal{N}})$ computationally tractable and large enough to include sufficient information about the epistasis. A sensitivity analysis of this parameter can be found in Section 3.2.
Remark 2: In this algorithm and in the simulation experiments, the RMSE function $\zeta(\lambda, \mu)$ was calculated for all individuals in the data set rather than a small sample.
Remark 3: In Step 1, the random selection of set $\mathcal{N}$ can be made more representative of the set $\{1, ..., N\}$ by first calculating the estimation error $|y_i - \hat{y}_i|$ for all $i \in \{1, ..., N\}$ from the previous iteration and then choosing $\mathcal{N}$ to include individuals whose estimation errors are representative of those of the set $\{1, ..., N\}$.
Remark 4: The stopping criteria could include the RMSE falling below a predetermined threshold or the number of iterations or computation time exceeding a predetermined value.

3. Computational Experiments

We conducted computational experiments to test the effectiveness of the proposed approach.

3.1. Data and Algorithm Implementation

We collected a soybean genotype data set from SoyBase [33], which consisted of 42,509 individuals of soybean cultivars each having 20,087 genes (represented by SNPs). In the simulation, we assumed that a parsimonious set of significant genes are responsible for a certain phenotype, and a feature selection algorithm is available to select a small set of genes that includes all significant genes and a number of insignificant ones. This assumption is reasonable because the purpose of the simulation is to test the effectiveness of the epistasis detection algorithms rather than that of feature selection algorithms.

We randomly generated ground truth values for β_0, β, b, λ, μ and ϵ and used these parameters to generate the phenotype response data y according to Model (3). The distributions of these random parameters are described as follows.

- β_0: normal distribution $\mathcal{N}(0, 30^2)$.
- β_i: uniform distribution $\mathcal{U}(15, 30)$, for all $j \in \{1, ..., p_0\}$.
- β_j: 0, for all $j \in \{p_0 + 1, ..., p\}$.
- b: uniform distribution $\mathcal{U}(15, 30)$.
- (λ, μ): first initialize λ and μ as zero vectors; then generate a random set $\mathcal{J} \subseteq \{1, ..., p\}$ that contains K unique indices; finally for all $j \in \mathcal{J}$, assign either $\lambda_j = 1$ or $\mu_j = 1$ with equal probability.
- ϵ_i: normal distribution $\mathcal{N}(0, \sigma^2)$, independent and identically distributed for all $i \in \{1, ..., n\}$; different values of σ were being used in the simulation.

The reason that we included a new parameter p_0 is two-fold. First, when a feature selection algorithm is used to reduce the number of explanatory variables from 20,087 to a small number, say $p = 40$, it may include some, say $p_0 = 30$, significant variables that have positive additive effects as well as some, say $p - p_0 = 10$, insignificant variables that have negligible additive effects. Second, certain genes may contribute to an epistatic effect without having noticeable individual additive effects themselves.

The new model and algorithm proposed in Sections 2.3 and 2.4 were implemented in Octave [34], using CPLEX [35] as the MILP solver for Model (10)-(18). Two alternative algorithms were also implemented in Octave for comparison: Enumeration and PathSeeker. The Enumeration algorithm evaluates $\zeta(\lambda, \mu)$ for all feasible values of (λ, μ) in a brute force manner. The PathSeeker algorithm was presented in [24] and is similar with our local search heuristic in Section 2.2. The implementation of these two algorithms was based on our understanding of the ideas reported in the literature and customized to fit the data format and simulation setting of our experiments. We made an honest effort to implement the algorithms in the most efficient manner that we were capable of, so that the performance differences reflected the differences in efficiency of algorithmic design more than the differences in efficiency of implementation.

3.2. Sensitivity Analysis of Model (10)-(18)

Table 1: Parameters for the sensitivity analysis

Parameter	Meaning	Values
K	Upper bound of the complexity of epistasis	{2, 3, 4, 5}
$p(p_0)$	Number of (significant) explanatory variables	{25(15), 50(40), 75(65), 100(90)}
n	Number of observations	{100, 200, 300, 400}
σ	Standard deviation of random error	{0, 2, 4, 6}

We conducted a sensitivity analysis of four parameters that could affect the effectiveness of the optimization Model (10)-(18), which are summarized in Table 1. The values for these parameters were chosen to explore the ranges of parameters within which the model is computationally tractable. A full-factorial experiment requires solving the model for $4^4 = 256$ different sets of parameters, and we ran 10 repetitions for each set. The optimization model used a small subset of the genotype and phenotype data, consisting of p genes and n individuals randomly selected from the full data set. We set 1,800 seconds as the time limit for CPLEX to solve the optimization model. We also applied the local search heuristic to refine the solution from the optimization model, which used a larger subset of the data, consisting of the p genes and all 42,509

individuals. Three values were recorded for each repetition: computation time, RMSE of the solution from Model (10)-(18), and RMSE of the solution from the local search heuristic (using the solution from Model (10)-(18) as the starting point). Almost half of the time, the 1,800-second time limit for Model (10)-(18) was reached. The total computation time for this sensitivity analysis was approximately 40 CPU days.

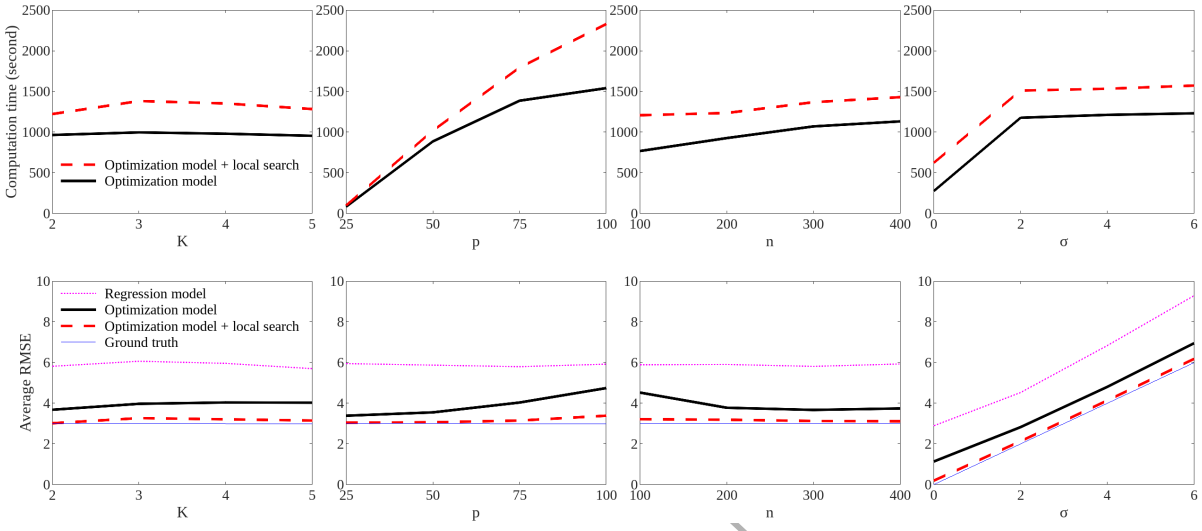

Figure 2: Results of sensitivity analysis. The four subfigures in the top row show the average computation times (capped at 1,800 seconds) of solving Model (10)-(18) using CPLEX and the local search heuristic for different parameter settings. The four subfigures in the bottom row show the average RMSEs using the basic multiple linear regression Model (1), the optimization Model (10)-(18), the optimization Model (10)-(18) and local search heuristic, and Model (3) with ground truth.

Results of the sensitivity analysis are summarized in Figure 2. We point out five noticeable observations. (1) The model is insensitive to the complexity of the epistasis, both in terms of computation time and accuracy of results. This is a desirable property, compared with exhaustive search algorithms whose computational complexity is an exponential function of the complexity of the epistasis. (2) The model is very sensitive to the number of candidate genes, p. When p exceeds 50, the computation time is likely to exceed 1,800 seconds and the accuracy of the result gets noticeably worse. (3) Computation time of the model is somewhat sensitive to the number of individuals, n. A sample size as small as $n = 100$ results in high RMSEs; when n exceeds 200 a larger sample size does not further reduce the RMSE. (4) Without random error, Model (10)-(18) can be solved efficiently to a reasonable accuracy. The computation time of the model is sensitive to the standard deviation of the random error. (5) Within 1,800 seconds, CPLEX is able to find a high quality solution that is either global optimal or in its neighborhood. In the latter case, the local search heuristic is effective and efficient in finding the global optimal solution using the solution from the optimization model as a starting point.

3.3. Comparison of Algorithms

We compared our new algorithm presented in Section 2.4 with the Enumeration and PathSeeker algorithms using a total of 4,000 independent repetitions of simulation, including 1,000 for each integer value of K between 2 and 5. In each repetition, a subset of genotype data was randomly selected from the original data set, consisting of $p = 40$ candidate genes (including $p_0 = 30$ significant ones) for all 42,509 individuals; the phenotype data was generated using the same ground truth assumptions of additive and epistatic effects described in Section 3.1. The same genotype and phenotype data set was used for all three algorithms for epistasis detection in each repetition.

For Model (10)-(18) in the new algorithm, we used $n = 6 \times 2^K$ as the sample size, which is small enough to keep the model tractable and large enough to make the epistasis detectable by including, on average, 6 instances of individuals that receive the epistatic effect. Rather than letting CPLEX solve the model to global optimality, we used very loose stopping critera: a 50% optimality gap or a 60-second time limit. The rationale is two-fold. First, as suggested by the sensitivity analysis results, the local search heuristic does not require a very high quality starting point to find the global optimal solution, whereas CPLEX would take much longer to arrive at global optimality. Second, since Model (10)-(18) is using a small sample of all individuals in each iteration, it may be a better use of the time to solve the model multiple iterations than to solve it once to a smaller optimality gap. For the alternative algorithms, we created a list of all possible solutions of epistases with the complexity ranging from 2 to 5. The Enumeration algorithm went through the list in a random order and checked the RMSEs of all the solutions. The PathSeeker algorithm was applied to all the solutions on the list in a random order as starting points. For all three algorithms, we used $(\lambda^* = 0^{p\times 1}, \mu^* = 0^{p\times 1})$ as the initial incumbent solution, which leads to the basic multiple linear regression Model (1) without consideration of the epistatic effect, and we recorded how the RMSE of the incumbent solution was improving over time. A time limit of 600 seconds was imposed for all 3 algorithms. The total computation time for the comparison experiment was approximately 63 CPU days.

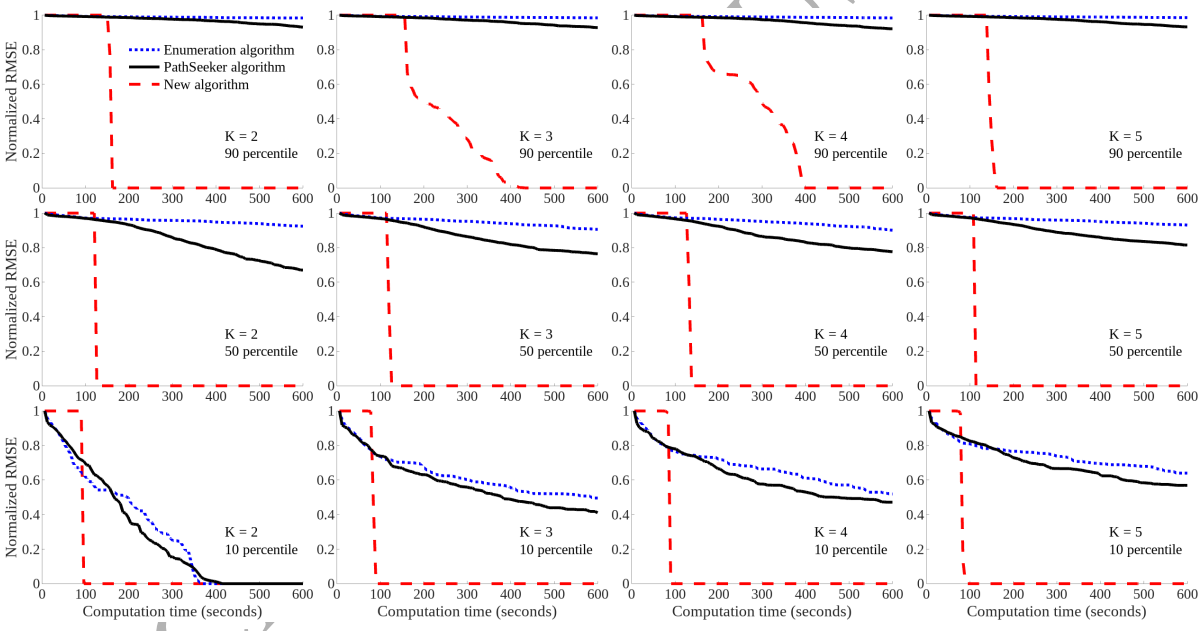

Figure 3: Comparison of the three algorithms' dynamic progress. The four columns of subfigures correspond to different complexity levels of the ground truth epistases, and the three rows correspond to different percentiles of results among the 1,000 repetitions. The RMSE values are normalized as $(\zeta - \zeta_0)/(\zeta_1 - \zeta_0)$, where ζ is the RMSE of the solution obtained by any of these algorithms, ζ_0 is the RMSE of the global optimal solution, and ζ_1 is the RMSE of Model (1) without consideration of the epistatic effect. Since the incumbent epistasis was empty at the beginning and improves over time, the normalized RMSE should start from 1 and gradually decrease to 0 or somewhere in between.

Figure 3 compares the three algorithms with respect to their ability to find improving incumbent solutions over time. We point out four observations. (1) In 10 out of 12 subfigures, the new algorithm reached global optimality within 200 seconds, and the other two subfigures took no more than 500 seconds; whereas the other two alternative algorithms were able to reach global optimality only at the 10th percentile of RMSE for the 1,000 repetitions when $K = 2$. (2) In 10 out of 12 subfigures, the new algorithm was able to reach global optimality in one iteration. The two exceptions were at the 90th percentile with $K = 3$ and $K = 4$,

which required 2 or 3 more iterations. (3) Solution quality of the two alternative algorithms deteriorated as the epistasis complexity increases, but the new algorithm did not show such sensitivity. (4) PathSeeker outperforms the Enumeration algorithm in almost all cases, due to its effectiveness in finding local optimal solutions rather than randomly exploring the solution space. The new algorithm outperforms PathSeeker by feeding the local search heuristic with high quality incumbent solutions from the optimization model as starting points.

Table 2 compares the solution quality of the three algorithms at the end of the 600-second time limit. The PathSeeker algorithm slightly outperformed the Enumeration algorithm, yet they both struggled to detect the true epistatic effect, especially for $K \geq 4$. The solution quality also deteriorated for more complex epistases, which is due to the exhaustive search nature of these algorithms. On the contrary, the new algorithm had an overall 95.5% (3,821 out of 4,000) chance of finding the global optimal solution, and this success rate got even higher for more complex epistases. This can also be attributed to the nature of the new algorithm, which detects the epistasis by looking for discrepancies between individuals that receive and not receive the epistatic effect. When the epistasis is more complex, a smaller subset of individuals receive the effect, and their discrepancies from other individuals are more outstanding and harder to be diluted and explained away by additive effects.

Table 2: Comparison of the three algorithms' solution quality at the end of the 600-second time limit. The values are the numbers of times that the normalized RMSEs fall into the labeled ranges.

Algorithm	K	Normalized RMSE			
		$(-\infty, 0]$	$(0, 0.01]$	$(0.01, 0.1]$	$(0.1, 1]$
Enumeration algorithm	2	143	5	4	848
	3	16	3	13	968
	4	0	3	7	990
	5	0	0	2	998
PathSeeker algorithm	2	124	41	56	779
	3	4	4	24	968
	4	2	3	13	982
	5	0	1	4	995
New algorithm	2	888	112	0	0
	3	958	5	14	23
	4	976	0	3	21
	5	999	0	1	0

These results suggested that the new algorithm demonstrated substantial improvements over the two alternatives, and the main reason is the combination of the optimization Model (10)-(18) and the local search heuristic. The former explores the solution space of all epistases by taking advantage of the efficient CPLEX solver for solving Model (10)-(18), and the latter takes an incumbent solution and refines it by finding its local optimal solution, which usually ends up being globally optimal.

4. Conclusion

The main contribution of this paper is a new approach for detecting the epistatic effect. At the core of this approach is a combinatorial optimization model that detects both the complexity of the epistasis and the exact combination of genes that triggers the effect. Our sensitivity analysis revealed that this model

is most effective and computationally efficient for a few dozen genes and a couple of hundred observed individuals. We also designed a new algorithm to detect the epistatic effect for a large data set with tens of thousands of genes and individuals. First, a feature selection algorithm can be used to reduce the number of genes to a few dozen, and then small samples of individuals are iteratively drawn to feed the optimization model, the solutions from which will be subsequently refined using a local search heuristic. We conducted computational experiments using a soybean data set, which consisted of 20,087 genes and 42,509 individuals. When compared with two popular algorithms from the literature, the Enumeration and PathSeeker algorithms, the new algorithm demonstrated significant improvement in the effectiveness and efficiency of detecting multi-way epistases.

As a caveat, the proposed approach has several limitations. For example, the validity of the new algorithm depends on three assumptions: (1) Model (3) is a reasonable approximation of the ground truth, (2) the phenotype of interest is determined by a small number of significant genes, and (3) an effective feature selection algorithm is available to select a small set of genes that includes all the significant ones and possibly some insignificant ones. The model also assumes that the X matrix is binary, meaning that it only applies to explanatory variables that are qualitatively categorized rather than quantitatively measured. A potentially fruitful direction of future research is to extend the proposed Model (3) and the new algorithm to include multiple epistatic effects simultaneously.

Acknowledgements

The authors are grateful to the Associate Editor and three anonymous reviewers for their insightful and constructive feedback. This research was partially supported by the U.S. Department of Agriculture NIFA Award 2017-67007-26175, the National Science Foundation Award 1830478, and by the Plant Sciences Institute at Iowa State University.

References

References

[1] H. J. Cordell, Epistasis: What it means, what it doesn't mean, and statistical methods to detect it in humans, Human Molecular Genetics 11 (2002) 2463–2468.

[2] S. Xu, Z. Jia, Genomewide analysis of epistatic effects for quantitative traits in barley, Genetics 175 (2007) 1955–1963.

[3] W. Bateson, Mendel's Principles of Heredity, Cambridge University Press, 1909.

[4] M. Taylor, I. Ehrenreich, Genetic interactions involving five or more genes contribute to a complex trait in yeast, PLoS genetics 10 (5) (2014) e1004324.

[5] M. D. Ritchie, L. W. Hahn, N. Roodi, L. R. Bailey, W. D. Dupont, F. F. Parl, J. H. Moore, Multifactor–dimensionality reduction reveals high–order interactions among estrogen–metabolism genes in sporadic breast cancer, The American Journal of Human Genetics 69 (2001) 138–147.

[6] O. Combarros, M. Cortina-Borja, A. D. Smith, D. J. Lehmann, Epistasis in sporadic Alzheimer's disease, Neurobiology of Aging 30 (2009) 1333–11349.

[7] R. L. Nagel, Epistasis and the genetics of human diseases, Comptes Rendus Biologies 328 (2005) 606–615.

[8] T. F. Mackay, J. H. Moore, Why epistasis is important for tackling complex human disease genetics, Genome Medicine 6 (2014) 1.

[9] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research 3 (Mar) (2003) 1157–1182.

[10] A. Unler, A. Murat, A discrete particle swarm optimization method for feature selection in binary classification problems, European Journal of Operational Research 206 (3) (2010) 528–539.

[11] X. Wan, C. Yang, Q. Y. et al., BOOST: A fast approach to detecting gene-gene interactions in genome-wide case-control studies, The American Journal of Human Genetics 87 (3) (2010) 325–340.

[12] X. Zhang, S. Huang, F. Zou, W. Wang., TEAM: efficient two-locus epistasis tests in human genome-wide association study, Bioinformatics 26 (2010) i217–i227.

[13] J. Piriyapongsa, C. Ngamphiw, A. I. et al., iLOCi: a SNP interaction prioritization technique for detecting epistasis in genome-wide association studies, BMC Genomics 13 (7) (2012) 1.

[14] D. Sluga, T. Curk, B. Zupan, U. Lotric, Heterogeneous computing architecture for fast detection of SNP-SNP interactions, BMC Bioinformatics 15 (2014) 216.

[15] J. Gonzàlez-Domìnguez, L. Wienbrandt, J. C. K. et al., Parallelizing epistasis detection in GWAS on FPGA and GPU-accelerated computing systems, IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) 12 (2015) 982–994.

[16] D. M. Evans, J. Marchini, A. P. Morris, L. R. Cardon, Two-stage two-locus models in genome-wide association, PLoS Genetics 2 (2006) e157.

[17] R. L. Collins, T. Hu, C. W. et al., Multifactor dimensionality reduction reveals a three-locus epistatic interaction associated with susceptibility to pulmonary tuberculosis, BioData Mining 6 (2013) 1.

[18] B. Maher, The case of the missing heritability, Nature 456 (2008) 18–21.

[19] J. H. Moore, F. W. Asselbergs, S. M. Williams, Bioinformatics challenges for genome-wide association studies, Bioinformatics 26 (2010) 445–455.

[20] J. C. Kässens, L. Wienbrandt, J. González-Domínguez, B. Schmidt, M. Schimmler, High-speed exhaustive 3-locus interaction epistasis analysis on FPGAs, Journal of Computational Science 9 (2002) 131–136.

[21] S. Leem, H. H. Jeong, J. Lee, K. Weea, K. Sohn, Fast detection of high–order epistatic interactions in genome–wide association studies using information theoretic measure, Computational Biology and Chemistry 50 (2014) 19–28.

[22] M. Xie, J. Li, T. Jiang, Detecting genome-wide epistases based on the clustering of relatively frequent items, Bioinformatics 28 (2012) 5–12.

[23] X. Wan, C. Yang, Q. Y. et al., MegaSNPHunter: a learning approach to detect disease predisposition SNPs and high level interactions in genome wide association study, BMC Bioinformatics 10 (2009) 1.

[24] C. Yang, Z. He, X. Wan, Q. Yang, H. Xue, W. Yu, SNPHarvester: a filtering-based approach for detecting epistatic interactions in genome-wide association studies, Bioinformatics 25 (4) (2008) 504–511.

[25] Y. Zhang, J. Liu, Bayesian inference of epistatic interactions in case-control studies, Nature Genetics 39 (9) (2007) 1167.

[26] R. Upstill-Goddard, D. Eccles, J. Fliege, A. Collins, Machine learning approaches for the discovery of gene–gene interactions in disease data, Briefings in Bioinformatics 14 (2) (2012) 251–260.

[27] Y. Zhang, S. Xu, A penalized maximum likelihood method for estimating epistatic effects of QTL, Heredity 95 (1) (2005) 96–104.

[28] J. S. Armstrong, F. Collopy, Error measures for generalizing about forecasting methods: Empirical comparisons, International Journal of Forecasting 8 (1992) 69–80.

[29] D. C. Montgomery, E. A. Peck, G. G. Vining, Introduction to Linear Regression Analysis, John Wiley & Sons, 2015.

[30] S. Chatterjee, A. S. Hadi, Regression Analysis by Example, John Wiley & Sons, 2015.

[31] H. D. Sherali, W. P. Adams, A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems, Vol. 31, Springer Science & Business Media, 2013.

[32] E. L. Lawler, D. E. Wood, Branch-and-bound methods: A survey, Operations Research 14 (4) (1966) 699–719.

[33] Soybase, `https://www.soybase.org` (2018).

[34] J. W. Eaton, D. Bateman, S. Hauberg, R. Wehbring, GNU Octave version 4.0.0 manual: a high-level interactive language for numerical computations, 2015.

[35] Cplex, `https://www.ibm.com/analytics/data-science/prescriptive-analytics/cplex-optimizer` (2018).