A gene-based permuted Xgboost method for detecting and ranking gene-gene interactions of qualitative trait

Yingjie GUO, Chenxi WU, Ao li, Junwei Zhang, Alon keinan, Maozu GUO

Abstract: Boosted tree is a popular and highly effective method in machine learning for modeling additive models with non-linear terms. In this paper, we propose a novel gene-based, permuted, extreme, gradient boosting method called gpXGB to detect interactions between genes in qualitative traits, which has advantage in both statistical power and biological interpretability. (The main idea is to permute the genotype within each class of the dataset in two ways, one keep the interaction between genes and another remove such interactions, then rank the AUC differences of the result of XGB after these two different types of permutation.)
The framework rank the interacting gene pairs by estimating the AUC difference of a XGB classification model on two test datasets through permutation that one keeping the pairwise interaction while the other removing the interaction.

1. Introduction

Genome-wide association studies (GWAS) have identified over six thousand single-nucleotide polymorphisms (SNPs) associated with complex diseases or traits. Earlier GWAS analysis strategies were largely based on single locus models, which test the association between individual markers and a given phenotype independently. Although this type of approaches have successfully identified many regions of disease susceptibility, most of these SNPs identified have small effect sizes which failed to fully account for the heritability of complex traits. Genetic interaction has been hypothesized to play an important role in the genetic basis of complex diseases and traits and to be one of the possible solutions to this problem of “missing heritability”. Even if genetic interaction explains only a tiny fraction of “missing heritability”, they can still provide some biological insight on the pathway level through by aiding the construction of novel gene pathway topologies.

The first investigations on genetic interactions have first been investigated at the SNP level, in which various statistical methods, including logic and logistic regression, odds-ratio, linkage disequilibrium(LD) and entropy-based statistic, are employed to detect SNP-SNP interactions (i.e. epistasis). Other techniques that have been used to study SNP-SNP interactions include multifactor dimensionality reduction, Tuning ReliefF, Random Jungle, BEAM, BOOST[1] and pRF[2]. These marker-based methods may encounter some common challenges, such as the complexity arising from the large number of pairwise or higher-order tests because all pairs or groups of SNPs have to be considered; the extensive burden of multiple-testing correction they entail.(what is this) In this paper, we aim to improve the power of gene-gene interaction detection by moving beyond SNP level, and instead considering all potential pairs of SNPs from each of a pair of genes in a single gene-based interaction detection.

Gene-based approaches have been successful for regular GWAS tests of main (marginal) associations, and there are several potential advantages in extending this methodology to gene-gene interaction detections. Firstly, a gene-based approach can substantially reduce the number of tests needed. For example, for 20,000 genes, there are ~ 2×10^{8} possible pairwise gene-based interactions to be tested, while for 3 million SNPs there are over ~ 5×10^{12} possible marker-based interactions to be tested. Secondly, a gene-based interaction test may have greater power, because when there are multiple interactions between features in the targeted genes (or other kind of regions), the effect of these interactions may be aggregated by the algorithm. Such aggregation has already been seen in gene-based GWAS tests for main association effect. Thirdly, a gene-based approach may be better at leveraging prior biological knowledge, which is often on the level of genes. For example, one may test pairs of genes that exhibit protein-protein interactions (PPI) or that participate in the same pathways.

In the work of Peng et al [3], canonical correlation analysis between two genes is done on both the case and the control group, and a U-statistic, called CCU, is used to measure the difference of the correlation between these two genes, which is used to indicate the presence of interaction. A limitation of this method is that in the correlation analysis only linear relations are considered. To overcome this limitation, [4, 5] extended CCU to KCCU, where the canonical correlation analysis is kernelized to account for possible non-linearity. Li et al. [6] introduced another method called GBIGM which is entropy-based and non-parametric, which was based on an entropy-based non-parametric. More recently, Emily [7] developed a new method called AGGrGATOr which combines the p-values in marker-level interaction tests to measure the interaction between two genes. Earlier
[8] this strategy was successfully used for the interaction detection for quantitative phenotypes.

In this paper, rather than designing a new dedicated statistic, we use a machine learning algorithm extreme gradient boost (Xgboost[9]) to propose a new approach, called gene-based permuted extreme gradient boost (gpXGB), to detect gene-gene interaction. The idea is to compare the performance of Xgboost on two different test datasets obtained from different permutation strategies, one keeping while another removing the interactions between selected gene pairs. Our method does not require explicit modeling of interacting terms and allow any kind of the functional form that interaction might take. An advantage of gpXGB is that it is nonparametric, hence may be more flexible for data-driven exploratory genome-wide association studies.

2. Materials and Methods

In this section we first detail the gpXGB approach. Then we describe the various simulation studies conducted to assess the type-I error rate as well as the statistical power of our approach in gene-gene interaction detection. Finally, we apply our approach to the NESDA () dataset to evaluate our approach in a real-lift situation.

2.1 Overview of gpXGB

Our method, gpXGB, is a machine learning based procedure for detecting the interaction between two genes in susceptibility with a binary phenotype, typically a case/control disease status. Let $y \in \{0, 1\}$ be the phenotype, where $y = 0$ stands for membership of the control group and $y = 1$ for membership of the case group. Let n be the number of instances in our sample, $\mathbf{Y} = \{y_1, \ldots, y_n\}$ be the vector consisting of their observed binary phenotypes. Let $\mathbf{X}_g$, where $g = 1, \ldots, G$ be the G genes in our gene list, each a collection of m_g SNP markers. The observed genotypes for gene $\mathbf{X}_g$ can be represented by an $n \times m_g$ matrix $\mathbf{X}_g = [x_{i,j}]_{1 \le i \le n, 1 \le j \le m_g}$ where $x_{i,j} \in \{0,1,2\}$ is the number of copies of the minor allele for SNP j carried by individual i . Let $\mathbf{X}_g^D$ and $\mathbf{X}_g^C$ be the matrices whose columns are those columns of $\mathbf{X}_g$ corresponding to samples in the case and control group, respectively. The genotype values in these matrices may also be adjusted to account for various covariates and population stratification.

We choose Xgboost as our classifier because gradient boosting decision tree (GBDT) is an effective and relatively model-agnostic way to approximate true target function which may have non-linear structure, and Xgboost is an algorithm which improves upon GBDT for its precision and computational efficiency.

Our approach consists of three steps: 1) training 2) permutation 3) testing and ranking. We start by training an Xgboost model with all the genes in the gene list and use cross-validation to choose a best model and save it. Then, for each selected pair of genes, we use our permutation strategies to generate two different test datasets, one keep while the other remove the interaction between the selected pair of genes. Lastly, we calculate the performance difference $DAUC$ for the two test datasets on the well-trained model. Which we use as a measurement for the strength of interaction between these two genes. The various steps of the gpXGB framework are illustrated in Figure1.

2.2 Overview of Xgboost

Xgboost[9] is a scalable supervised machine learning system based on tree boosting, and recently has been dominating applied machine learning as well as in Kaggle competitions. It is an algorithm which improves GBDT with speed and performance improvement.

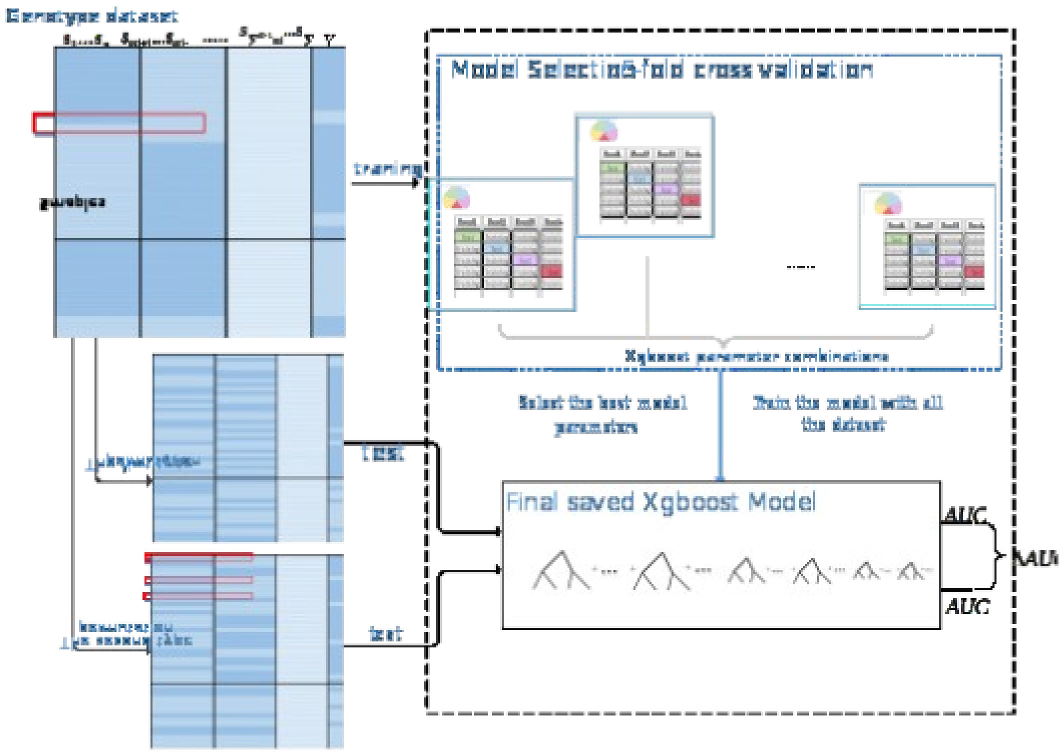
2.2.1 Ensemble of CARTs

In this ensemble model, the base classifier is CART (Classifying And Regression Tree), which is similar to decision trees, but on each leaf, instead of a classification, a real-valued score is assigned. This makes ensemble training easier and may also provide more information beyond classification.

Let F be the space of functions that can be represented by CARTs, the ensemble predictor is

$\hat{y} = \sum_{k} f_k,\ f_k \in F$. In our case we interpret it as in logistic regression, namely

$$p(y=1|x) = \frac{1}{1+e^{-\hat{y}(x)}} \qquad (1)$$



Genotype dataset
training
test
test
Model Selection 5-fold cross validation
Xgboost parameter combinations
Select the best model parameters
Train the model with all the dataset
Final saved Xgboost Model
AUC
AUC
AUC

Figure1 The framework of gpXGB

Hence, the learning objective is

$$obj = \sum_{i} l(y_i, \hat{y}(x_i)) + \sum_{k} W(f_k) \qquad (2)$$

Where $l(y, \hat{y}) = y\log(1+e^{-\hat{y}}) + (1-y)\log(1+e^{\hat{y}})$ is the logistic regression loss function, and $W(f_k)$ is the regularizer.

2.2.2 Gradient Boosting

It is not feasible to train all the trees in the ensemble together at once because it is hard to calculate the gradient as which is needed in traditional optimization methods. Instead, Xgboost use an additive training strategy: fix the trees have already learned, add new trees one at a time. Let $\hat{y}^{(t)}$ be the predictor at iteration t , then

$$\hat{y}^{(0)} = 0 \qquad (3)$$

$$\hat{y}^{(t)} = y^{(t-1)} + f_t \qquad (4)$$

Where $f_t \in F$ and optimizes the following target function, which is obtained by the Taylor expansion of the lost function for logistic regression to the second order.

$$obj^t = \sum_i (g_i(\hat{y}^{(t-1)}(x_i)) f_t(x_i) + \frac{h_i(\hat{y}^{(t-1)}(x_i))^2}{2} f_t^2(x_i)) + \Omega(f_i) \qquad (5)$$

Here $g_i(\hat{y}) = \frac{d}{d\hat{y}} l(y_i, y)$, $h_i(\hat{y}) = \frac{d}{d\hat{y}} l(y_i, y)$.

2.2.3 Regularizer and Training strategy for CARTs

For any $f \in F$, let T be the number of leaves in the tree representing f , $w_1, \ldots w_T$ be the scores on the leaves. Then the regularizer used in XGBoost is

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_j w_j^2 \qquad (6)$$

The purpose of the second term is that it can smoothen the leaf scores.

To optimize f_t , firstly note that given a tree structure, obj^t is a quadratic function of the scores w_j , and the minimum of obj^t as well as the w_j that minimizes obj^t can be easily calculated given the tree structure. Now the tree can be constructed by a greedy algorithm in which one starts with a tree with one single node, and repeatedly split its leaves in a way that maximizes the decrease in obj^t in each step.

2.3 Permutation

The idea of detecting SNP interaction through permutation has previously been used by Greene [10] and Jing [2].Greene et al. designed an explicit test of epistasis to reflect only nonlinear interaction or epistasis component of the model. They advanced the traditional permutation testing framework by shuffling each SNP column instead of randomizing the class label, which can generate permuted datasets for testing the null hypothesis that the only genetic associations in the data are linear or additive in nature and that any nonlinear interaction effects are only there by chance. This yields an explicit test of epistasis when combining with a method such as MDR, is capable of modeling nonlinear interactions. Jing et al. developed a permuted Random Forest (pRF) method. They generated two test dataset by permutating the genotype of a pair of SNPs. In one dataset the SNPs are shuffled independently, while in the other they are shuffled as a pair hence keeping the pairwise association. The difference of error rate between the two test dataset on a well-trained RF model is then used to measure the strength of the interaction of selected SNP pair. Other SNPs except for the selected pair kept their original form in both dataset, so the interactions among other non-selected SNPs were preserved in both of the permutation framework.

Both methods above are marker-based. Motivated by them, we designed a gene-based permutation strategy for our interaction detection. For each pair of genes, we carried out two permutation strategies to generate two test datasets. Firstly, we divide the samples by class label into case and control groups. Then, in the first permutation strategy, we shuffle the genotype of all the genes independently among the samples within each group, which removes all associations between genes within each group while keeping the association between SNPs within each gene. The independent margin effect of each gene is preserved due to the unchanged genotype frequencies of each gene within each group before and after permutation. In the second permutation strategy, within both the case and the control group, the genotypes of the two chosen genes are shuffled together as a group while the genotypes of all other genes are shuffled independently. Hence, the difference between the two test dataset is merely the presentation or deletion of the interaction between the pair of genes.

Comparing with the approach of Jing etc., our approach is gene-based as opposed to marker-based, hence is more suitable for detecting interaction between genes. Also, in selecting interaction terms we used a forward selection strategy instead of the backward selection strategy used by Jing etc.
(emphasize the difference between our permutation and Jing's)

2.4 Testing and Ranking

Our approach for gene-based gene-gene interaction detection is based on comparing the performance of a model trained with XGBoost (c.f. section 2.2) on two test datasets obtained through the different permutation strategies described in section 2.3. Our interaction estimation technique is based on the following observation. If X_{g1} and X_{g2} interact, then the trained model should have significantly better predictive performance on the test data obtained through the second permutation strategy than the test data obtained through the first permutation strategy, because in the second kind of permutation the relationship between these two genes are preserved while in the first kind of permutation such information is lost. On the other hand, if the performance on the two testing data sets are similar, we consider the interaction between the two genes to be absent.

In this paper, we use AUC (Area Under Curve), the area under the a receiver operating characteristic (ROC) curve, to measure the classifier performance. In machine learning, ROC curve is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied. Consider the ROC plot of the true positive rate vs the false positive rate as the threshold value for classifying an item as class 0 is increasing from 0 to 1: if the classifier is very good, the true positive rate will increase quickly and the area under the curve will be close to 1. If the classifier is no better than random guessing, the true positive rate will increase linearly with the false positive rate and the area under the curve will be around 0.5. Generally, the larger the AUC is, the better performance the classifier has. Especially, the AUC is independent of the fraction of the test dataset that belongs to class 0 or class 1, and is useful for evaluating the performance of classifiers even on unbalanced data sets.

In the testing step, for each pair of genes, the Xgboost model is tested on two permuted datasets obtained as described in 2.3, and the performances are measured by AUC scores. Both types of permutation are repeated 100 times, and the average AUC is calculated. We named the average AUC from the first permutation strategy, in which the test dataset only maintain the margin effect of genes in the gene list, as AUC1, and named the average AUC from the second permutation strategy, in which the test dataset kept both interaction between selected pair of gene and margin effect of all the genes, as AUC2. Therefore, the difference between AUC2 and AUC1, which we call the AUC difference $DAUC = AUC2 - AUC1$, would be a measure of the classifier performance caused by the interaction.

In the last step, after the $DAUC$ has been calculated for each pair of genes, we rank the gene pairs by their $DAUC$ s, and consider the pair of genes with the larges $DAUC$ to be the one with the strongest interaction among all the gene pairs. In practice, we can pick top 5 pairs as the candidate interaction pair or selected candidate pairs through the distribution of $DAUC$.

The ALGORITHM 2 is the framework of gpXGB.

ALGORITHM2: gpXGB

Input: genotype dataset $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}, x_i \in R^{(m_1 + \ldots + m_G)}, y_i \in \{0, 1\}$; gene list file with position information for G genes; buffer region size.

Output: The ranking list sorted by $DAUC$ for all the gene pairs in the gene list.

```
1. Train Xgboost model, using grid search to find the proper parameter combination of the Xgboost, using 5-fold cross validation for each parameter combination and select the best parameter combination that gives the best average predictive performance.
2. Divide dataset S into training and testing set under the scheme of 5-fold cross-validation.
2. XgboostModel = trainXgboost($S$ , parameter combination)
3. for $i = 1$ to $G - 1$ do
4.   for $j = i + 1$ to $G$ do
5.     for $k = 1$ to $100$ do
6.       testData1=TypeIPermu ()  // generate test data with the first permutation strategy
7.       testData2=TypeIIPermu ()  // generate test data with the second permutation strategy
8.       $AUC_{k1}$ =calcAUC(Predict(XgboostModel,testData1))
9.       $AUC_{k2}$ = calcAUC(Predict(XgboostModel,testData1))
10.      $DAUC_k = AUC_{k2} - AUC_{k1}$   //$DAUC$ for the $k$-th permutation
11.    end for
12.    $DAUC_{ij} = (\sum_{k=1}^{100} DAUC_k)/100$   //Average $DAUC$ for 100 times permutation
```

```
    end for
end for
Return $C_G^2$ pairs of gene interaction ranking list sorted by $DAUC_{ij}$ with decrease order
```

2.5 Simulation study

The goal of this simulation study is to evaluate the performance of gpXGB procedure for gene-gene interaction detection. All simulated datasets were set to have 50 SNPs. Among them 2 SNPs were functional and the remaining 48 SNPs were non-functional. The 50 SNPs formed 5 genes, each had 10 SNPs. The 2 functional SNPs were put into the first and second gene, and the performance is measured by how likely our algorithm can rank the two interacting genes as the most significant. We chose the publicly available tool GAMETES[11] to generate the simulated genotype data. This tool is designed to generate epistasis models that we refer to as pure and strict. Purely and strictly epistasis models constitute the most difficult type of disease association model to detect, as such associations may be observed only if all n-loci are included in the disease model. This requirement makes these types of models an attractive gold standard for simulation studies of complex multi-locus effects.

In this simulation study, to test the effects of heritability (which measures the strength of correlation between genotype and phenotype) and sample size, we performed experiments under two different scenarios. In the first scenario, we tested two locus epistasis models with five different heritability (0.01, 0.025, 0.05, 0.1 and 0.2) and two different minor allele frequencies (0.2 and 0.4) with prevalence set to be 0.2 and sample size to be 3000. Ten models for each of the 10 heritability-allele frequency combinations were generated, so that we had 100 models in total in accordance to Hardy-Weinberg proportions. The penetrance tables were generated for these 100 models in the absence of main effect. One hundred datasets were generated from each model with balanced cases and controls,resulting in 10000 datasets in total in this scenario. In the second scenario, we set heritability to be (XX, XX) and MAF to be 0.2, prevalence to be XX with sample size 10000. Then, 100 datasets were generated by random sampling from this large dataset for each of the 5 sample sizes 1000, 2000, 3000, 4000 and 5000. In this scenario, we have 500 datasets in total.

2.6 Real data analysis

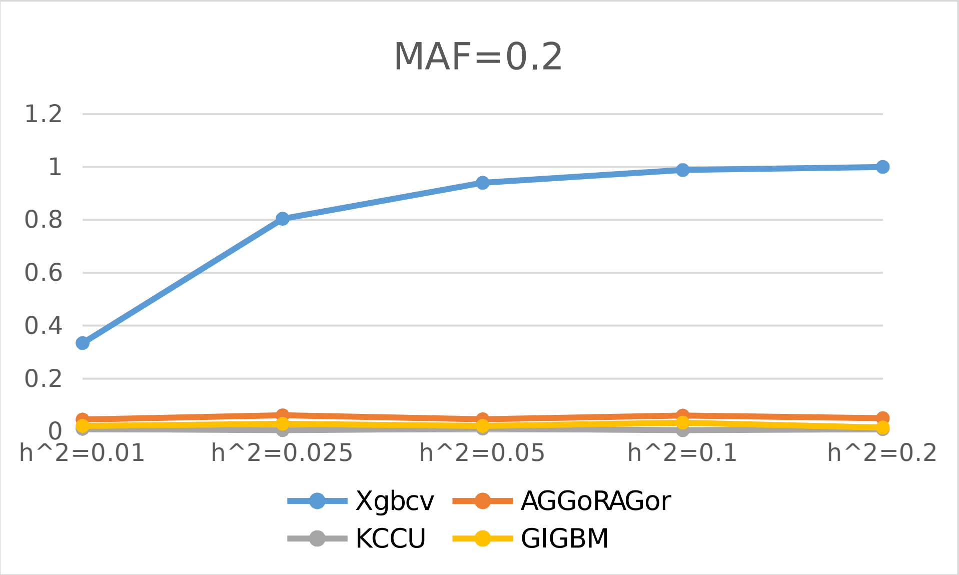
To assess the capacity of gpXGB to deal with real case-control phenotype, we first investigated the susceptibility of a set of pairs of genes to Rheumatoid Arthritis (RA), a chronic autoimmune joint disease where persistent inflammation affects bone remodeling leading to progressive bone destruction. We used the GSE39428 dataset for which genotyping is performed using a custom-designed Illumina 382-SNP VeraCode microarray (Illumina, San Diego, CA, USA) to determine possible associations of genes to RA. The dataset contains 266 cases and 163 controls. After preprocessing, we obtain 381 SNPs encoding 17 genes. Next, we used the WTCCC dataset as a replication cohort WTCCC (2007). The datasets were genotyped in the British population using the Affymetrix GeneChip 500k. Quality control was performed in PLINK with several steps. First we removed samples with reported sex that did not match the heterozygosity rates observed on chromosome X [12]. We additionally filtered out SNPs with >10% missingness, with a minor allele frequency (MAF) < 0.05, or for which missingness was significantly correlated with phenotype ($p<1$�10^{-4}). We further filter out SNPs that are not in Hardy-Weinberg equilibrium in controls, as well as filter out samples with >10% missing SNPs. After the QC steps, we have XX SNPs, XXX samples with XXX cases and XXX controls.

In a second analysis, we aim to verify some gene-gene interaction in the RA pathway hsa05323 in KEGG pathway dataset. Genotyping coordinates are given in NCBI Build36/UCSC hg18 (National Center for Biotechnology Information, Bethesda, MD). There are XX genes in the pathway, and we can mapping XX based on Build36 annotation. For each gene, we add 10k to both the upstream and downstream. Gender was included as covariate in the analysis. Principal component analysis was conducted using GCAT[], and top 10 PCs were also included as covariates to account for potential population stratification.

2.7 Competitive methods

The performance of our procedure gpXGB was compared to three previously published methods: Kernel Canonical Correlation-based U-statistic analysis (KCCU)[4, 5], the gene-based information gain method (GBIGM)[6] and A Gene-based Gene-Gene interaction test method (AGGrEGATOr)[7]. We adapted them to the task of ranking.

2. Results



3.1 methods to compare

4. Discussion

Conflict of interest

Acknowledgements

References

1. Wan, X., et al., *BOOST: A fast approach to detecting gene-gene interactions in genome-wide case-control studies.* Am J Hum Genet, 2010. **87**(3): p. 325-40.
2. Li, J., et al., *Detecting gene-gene interactions using a permutation-based random forest method.* BioData Min, 2016. **9**: p. 14.
3. Peng, Q., J. Zhao, and F. Xue, *A gene-based method for detecting gene-gene co-association in a case-control association study.* Eur J Hum Genet, 2010. **18**(5): p. 582-7.
4. Yuan, Z., et al., *Detection for gene-gene co-association via kernel canonical correlation analysis.* BMC Genet, 2012. **13**: p. 83.
5. Larson, N.B., et al., *Kernel canonical correlation analysis for assessing gene-gene interactions and application to ovarian cancer.* Eur J Hum Genet, 2014. **22**(1): p. 126-31.
6. Li, J., et al., *A gene-based information gain method for detecting gene-gene interactions in case-control studies.* Eur J Hum Genet, 2015. **23**(11): p. 1566-72.
7. Emily, M., *AGGrEGATOr: A Gene-based GEne-Gene interActTiOn test for case-control association studies.* Stat Appl Genet Mol Biol, 2016. **15**(2): p. 151-71.
8. Ma, L., A.G. Clark, and A. Keinan, *Gene-based testing of interactions in association studies of quantitative traits.* PLoS Genet, 2013. **9**(2): p. e1003321.
9. Chen, T. and C. Guestrin. *XGBoost: A Scalable Tree Boosting System*. in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016.
10. Greene, C.S., et al., *Enabling personal genomics with an explicit test of epistasis.* Pac Symp Biocomput, 2010: p. 327-36.
11. Urbanowicz, R.J., et al., *GAMETES: a fast, direct algorithm for generating*

pure, strict, epistatic models with random architectures. BioData Min, 2012. **5**(1): p. 16.

12. Laurie, C.C., et al., *Quality control and quality assurance in genotypic data for genome-wide association studies.* Genet Epidemiol, 2010. **34**(6): p. 591-602.

KCCU

The kernel canonical correlation-based U-statistic model (KCCU) is a gene-based gene-gene interaction detection method which can reflect nonlinear relationship between two genes in the case-control dataset. In KCCU, for given two genes $l, m \in \{1, \ldots, G\}$, such that $l \neq m$. Consider the genotype matrices $\mathbf{X}_l^D, \mathbf{X}_m^D, \mathbf{X}_m^C$ and $\mathbf{X}_l^C$, with corresponding reduced kernel representations $\mathbf{K}_l^D, \mathbf{K}_m^D, \mathbf{K}_l^C$ and $\mathbf{K}_m^C$. Define $r_{lm}^D = CCA(\mathbf{K}_l^D, \mathbf{K}_m^D)$ and $r_{lm}^C = CCA(\mathbf{K}_l^C, \mathbf{K}_m^C)$ to be the respective maximal kernel canonical correlations for case and control between gene l and m . After transform the r_{lm}^D and r_{lm}^C to an analog of the Fisher's simple correlation coefficient transformation, we can obtain the KCCU statistic. The details of KCCU method can be found in the paper [].