SEE: a novel multi-objective evolutionary algorithm for identifying SNP epistasis in genome-wide association studies

Liyan Sun\textsuperscript{a,b}, Guixia Liu\textsuperscript{a,b}, Lingtao Su\textsuperscript{a,b} and Rongquan Wang\textsuperscript{a,b}

\textsuperscript{a}Department of Computational Intelligence, College of Computer Science and Technology, Jilin University, Changchun, P.R. China; \textsuperscript{b}Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, P.R. China

ABSTRACT

Although genome-wide association studies play an increasingly important role in identifying causes of complex diseases, detecting SNP epistasis in these studies is a computational challenge. The existing methods are usually based on a single-correlation model between SNP combinations and phenotype and their performance is often unsatisfactory. The highest average power of the existing methods is 0.58 on DME models and 0.97 on DNME models. The highest average F-measure of the existing methods is 0.44 on DME models and 0.90 on DNME models. The lowest average computation time (second) of the existing methods is 2.12 on DME models and 2.09 on DNME models. In this work, a novel multi-objective evolutionary algorithm named SEE is presented for identifying SNP epistasis. SEE uses a novel evolutionary strategy based on sort, exploration and exploitation. SEE was compared with other existing methods using 72 simulated datasets. The average power of SEE is 0.71 with DME models and 0.99 with DNME models. The average F-measure of SEE is 0.68 with DME models and 0.99 with DNME models. The average computation time of SEE is 0.21 with DME models and 0.40 with DNME models. It is indicated that SEE outperforms other algorithms in both F-measure and computation time. It was then utilized to analyze real data obtained from the Wellcome Trust Case Control Consortium. Availability and Implementation: SEE is freely available at https://github.com/sunliyan0000/SEE.

ARTICLE HISTORY

Received 4 October 2018
Accepted 7 March 2019

KEYWORDS

Single-nucleotide polymorphism; epistasis; genome-wide association studies; evolutionary algorithm

Introduction

Identifying genes responsible for Mendelian diseases has met great success, but it is still very challenging for non-Mendelian (complex) diseases [1\textsuperscript{-}3]. As the regulatory mechanism in the human genome is very complicated, complex diseases may be caused by multiple genetic variations instead of a single genetic variation. These multiple genetic variations may be not significantly associated with the complex diseases individually, but they may have strong association jointly. This phenomenon is referred to as epistasis [4\textsuperscript{-}8].

With the development of genome-wide single nucleotide polymorphism (SNP) genotyping technology, genome-wide association studies (GWAS) in which hundreds of thousands of SNPs of case samples and control samples are genotyped play an important role in identifying the causes of complex diseases[9\textsuperscript{-}14].

Recently, a number of methods have been proposed to identify multi-locus models (epistasis or SNP combinations) in GWAS As shown in Figure 1, these approaches can generally be classified into five categories: exhaustive search, stochastic search, filter search, model search and evolution search.

Exhaustive search [15\textsuperscript{-}17] enumerates all possible SNP combinations to identify significantly associated ones. It calculates a score for each SNP combination and then determines which SNP combinations are associated based on the threshold specified by the user. It requires tremendous computing resources while analyzing real GWAS data. For example, to detect two-locus models, it has to calculate 250 billion scores for a set of real GWAS data containing 500,000 SNPs. In order to analyze real GWAS data within an acceptable time, the exhaustive search method is

CONTACT

Guixia Liu liugx@jlu.edu.cn Department of Computational Intelligence, College of Computer Science and Technology, Jilin University, Changchun 130012, P.R. China.

Supplemental data for this article can be accessed here.

© 2019 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
always implemented on GPU or other parallel computing platforms [18, 19].

Stochastic search [20–22] detects associated SNP combinations via random sampling. Although the computation time is much shorter than that of exhaustive search, it depends heavily on the random seed. To improve the probability of identifying the associated SNP combination, it needs a large number of iterations, especially when there are millions of SNPs.

Filter search [23–25] can be divided into two subcategories: exhaustive and not exhaustive. The non-exhaustive search calculates a score for each SNP and then performs an exhaustive search on the SNPs whose scores are greater than a specified threshold. It performs poorly if the real associated SNPs have no marginal effects. The exhaustive filter search enumerates all possible SNP combinations as the exhaustive search does. Unlike the exhaustive search, the exhaustive filter search calculates a sample fast but not with a very accurate score for each combination. The complex and accurate analysis is then performed on the SNP combinations greater than the specified threshold. As the fast score greatly reduces the number of SNPs or SNP combinations on which the complex and accurate analysis has to be conducted, the computation time is lower. However, it is very difficult to design a reasonably fast score to remove SNPs or SNP combinations without mistakes. Filter search always performs poorly when the real associated SNPs do not affect the phenotype as expected by the author.

Model search [26–32] builds a classification model based on GWAS data. The model utilizes SNPs to accurately classify samples to cases and controls. It assigns a score to each SNP according to its contribution in the classification. The SNPs which are greater than the specified threshold are returned as the significantly associated ones. However, the result of model search strongly depends on the construction process of building the model. While building the model, most of the classification algorithms tend to use variables with large marginal effects. Model search performs poorly in identifying SNPs without marginal effects.

Evolution search [33–40] employs evolutionary algorithms, such as differential evolution (DE), harmony search (HS) and ant colony optimization (ACO), to find SNP combinations with higher scores (more associated with the phenotype). It always contains two parts: the evolutionary algorithm and the evaluation objective of the algorithm. The evolutionary algorithm reduces the time complexity of searching, and the objective determines which SNP combinations are better ones. How to design the two parts is the key of evolution search.

Figure 1. Classification of algorithms for analyzing GWAS data.
In this work, a novel evolution search algorithm is presented. Compared with previous algorithms, SEE mainly has the following advantages:

1. In order to better adapt to GWAS data, a novel evolutionary algorithm based on sort, exploration and exploitation is designed and implemented.
2. Exploration rate is introduced into the evolutionary algorithm to control the ways of generating a new individual.
3. Adjust operation is introduced into the evolutionary algorithm to improve the ability.
4. Eight evolution objectives are successfully integrated via the sort operation. The four of them are measures of the association between an SNP combination and the phenotype and the others are used to measure the epistasis.

To show the performance of SEE, we performed experiments on 72 simulated datasets. SEE was compared with BEAM [20], DECMDR [37], MACOED [35], SNPHarvester [22], SNPRuler [41] and AntEpiSeeker [33]. The results on simulated datasets indicate that SEE outperforms other existing algorithms in both F-measure and computation time. SEE was then utilized to analyze a real GWAS dataset which contains about 2000 individuals for each of seven major diseases and a shared set of about 3000 controls [42]. Many genes and SNPs associated with the seven major diseases have been found.

Materials and methods

Hardware

All experiments were performed on a computation platform using a Windows system with 12G RAM and i3-3220 CPU.

Algorithm

In order to better adapt to detect epistatic interactions in GWAS, we designed a novel evolutionary algorithm called SEE. The algorithm is mainly composed of three parts: sort, exploration and exploitation. The pseudo code of the algorithm is shown below (Algorithm 1) and detailed descriptions are given in the subsequent subsections.++

Algorithm 1. SEE

1: initialize the tracing table.
2: //tracing is a hash table which records all calculated individuals.
3: initialize the population randomly.
4: calculate and sort the population.
5: insert individuals in the population into tracing.
6: calculate the exploration rate er.
7: the number of generation G = 0.
8: while termination criterion is not satisfied (e.g. G < maximum iterations) do
9: generate a random number ran ∈ [0, 1).
10: if ran < er then
11: generate a new individual using exploration.
12: else
13: generate a new individual using exploitation.
14: end if
15: adjust the new individual.
16: if adjust success then
17: calculate evaluation objectives for the adjusted individual.
18: insert the calculated individual into tracing
19: if the individual is useful then
20: replace the worst individual in the population with the adjusted individual.
21: sort the population.
22: update er.
23: end if
24: end if
25: G = G + 1
26: end while
27: generate the result.

Individual

In the SEE algorithm, an individual in the population is a vector. It is regarded as a feasible combination. Suppose n SNPs and l-way epistatic interaction detection in a case–control study. An individual in the population is represented by Equation (1).

\[
I_G = (f_{0,i,G}, f_{1,i,G}, \ldots, f_{l-1,i,G}) | f \in \text{SNPs} \wedge f_{j,i,G} < f_{j+1,i,G}, \forall i \in [0, \text{numPop}) \wedge f_{j,i,G} \in [0, n), \forall j \in [0, l)
\]

where \(I\) represents an individual; \(i\) represents the index of the individual in the population; \(G\) represents the generation number of the population; \(\text{numPop}\) represents the number of individuals in the population. The SEE algorithm starts with \(G = 0\) and ends with \(G = \text{maxIter} \) (a parameter specified by the user). \(f_{j,i,G}\) represents a selected variable in the \(i\)th individual in the \(G\)th generation. At initialization (i.e. \(G = 0\)), the elements in an individual are randomly selected from 0 to \(n\), where \(n\) is the number of SNPs in the GWAS data.
Population

A population is made up of many (\text{numPop}) individuals. At initialization (i.e. \(G=0\)), the population initializes all individuals randomly. As the algorithm runs and \(G\) increases, the individuals in the population are evolved towards better ones by exploration and exploitation.

Sort

In the SSE algorithm, sort operation is to sort the individuals in the population. Unlike other evolutionary algorithms, eight evaluation objectives are calculated simultaneously to measure an individual. They are \(ce\), \(gini\), \(k2\), \(g\), \(cec\), \(ginic\), \(k2c\) and \(gc\). The first four objectives are used to measure the association between an individual and the phenotype. As four evaluation objectives from different fields are used while deciding the informative individuals, the population is more informative and the SSE algorithm has better performance \([35, 36, 38]\). The latter four objectives are designed to measure the epistasis.

Conditional entropy (ce)

Mutual information (mi) is a measure of the mutual independence between two variables. It is widely used to measure the association between an SNP combination (\(l\)) and phenotype (\(Y\)). As shown in Equation (2), since \(H(Y)\) (entropy of \(Y\)) remains unchanged in a set of GWAS data, \(ce\) is also appropriate to measure the association between \(I\) and \(Y\). The smaller the value of \(ce\), the more significant the association between \(I\) and \(Y\) is

\[
mi(l, Y) = H(Y) - ce(Y|I) \\
H(Y) = - \sum_{s \in S} p(s) \times \log p(s) \\
\]

\[
ce(Y|I) = - \sum_{c \in C, s \in S} p(c, s) \times \log \left( \frac{p(c, s)}{p(c)} \right), \tag{2}
\]

where \(ce\) is one of the eight evaluation objectives used in SEE algorithm; \(mi\) is the mutual information of \(I\) and \(Y\); \(I\) represents an individual (SNP combination); \(Y\) represents the phenotype; \(H\) represents the information entropy; \(C\) is the set of combinations of \(I\) with different values (the length of \(C\) is \(3I\) as the possible value of an element in \(I\) is 0, 1 or 2 and \(I\) represents the length of \(I\)); \(S\) represents the set of phenotype values; \(p(s)\) represents the probability of samples where the SNP combination takes \(c\) and the phenotype takes \(s\); \(p(c)\) represents the probability of samples where the SNP combination takes \(c\); \(p(c,s)\) represents the probability of samples where the SNP combination takes \(c\) and the phenotype takes \(s\).

Gini impurity (gini)

Gini impurity \([43]\) is used by the CART (classification and regression tree). It is a measure of how often a randomly chosen element from the set would be incorrectly labelled if it is randomly labelled according to the distribution of labels in the subset. It is often used in feature selection. As shown in Equation (3), the difference between \(Gini(Y)\) and \(Gini(Y,I)\) is widely used to measure the association between \(Y\) and \(I\). As \(Gini(Y)\) remains unchanged in a set of GWAS data, \(gini(Y,I)\) is also appropriate to be a measure of the association. The smaller the value of \(gini\) is, the more significant the association between \(I\) and \(Y\) is

\[
Gini(Y) = \sum_{s \in S} p(s) \times (1 - p(s)) \\
Gini(Y,I) = \sum_{c \in C} p(c) \times \sum_{s \in S} p(s|c) \times (1 - p(s|c)), \tag{3}
\]

\[
gini(Y,I) = Gini(Y,I),
\]

where \(gini\) is one of the eight evaluation objectives used in SEE algorithm; \(I\) represents an individual (SNP combination); \(Y\) represents the phenotype; \(C\) is the set of combinations of \(I\) with different values; \(S\) represents the set of phenotype values; \(p(c)\) represents the probability of samples where the SNP combination takes \(c\); \(p(s|c)\) represents the probability of samples where the phenotype takes \(s\) in samples where the SNP combination takes \(c\).

Bayesian network scoring criterion (k2)

The Bayesian network scoring criterion is derived from the BN (the directed graphical model) \([44, 45]\). In the GWAS BN, the genotypes and phenotypes are of different fields and their conditional dependences are edges. The \(k2\) score is shown in Equation (4). The smaller the value of \(k2\), the more significant the association between \(I\) and \(Y\) is

\[
k2(Y,I) = \prod_{c \in C} \left( \frac{(SL-1)!}{(m_c + SL-1)!} \prod_{s \in S} m_c! \right), \tag{4}
\]

where \(k2\) is one of the eight evaluation objectives used in SEE algorithm; \(I\) represents an individual; \(Y\) represents the phenotype; \(C\) is the set of combinations of \(I\) with different values; \(S\) represents the set of phenotype values; \(SL\) represents the length of \(S\); \(m_c\) is the number of samples where the SNP combination takes \(c\); \(m_c|s\) is the number of samples where the SNP combination takes \(c\) and the phenotype takes \(s\).
G-test p-value (g)

In statistics, G-tests are likelihood-ratio or maximum likelihood statistical significance tests that are increasingly being used in situations where chi-squared tests were previously recommended \[1, 46, 47\]. The formula is shown in Equation (5). Given the null hypothesis that \( I \) (SNP combination or individual) and \( Y \) (phenotype) are independent, the distribution of G-statistic is approximately a chi-squared distribution. The length of \( C \) minus 1 is the degrees of freedom. The p-value of the G-statistic is defined as \( g \). The smaller the value of \( g \), the more significant the association between \( I \) and \( Y \) is.

\[
G\text{-statistic}(Y, I) = 2 \sum_{c \in C} \sum_{s \in S} m_{cs} \ln \frac{m_{cs}}{E_{cs}}.
\]

\[
g(Y, I) = pvalue_g(G\text{-statistic}(Y, I))
\]

where \( g \) is one of the eight evaluation objectives used in the SEE algorithm; \( I \) represents an individual (SNP combination); \( Y \) represents the phenotype; \( C \) is the set of combinations of \( I \) with different values; \( S \) represents the set of phenotype values; \( m_c \) is the number of samples where the SNP combination takes \( c \); \( m_s \) is the number of samples where the phenotype takes \( s \); \( m_{cs} \) is the number of samples where the SNP combination takes \( c \) and the phenotype takes \( s \); \( E_{cs} \) is the expected number of samples where the SNP combination takes \( c \) and the phenotype takes \( s \) if the SNP combination and the phenotype are independent; \( pvalue\_of \) represents the function to compute the p-value of the chi-square distribution.

Change of the evaluation objectives

In GWAS, an individual which has low \( ce \), gini, \( k2 \) and \( g \) may not be an epistatic interaction, because it may contain an SNP which has a strong marginal effect. The strong SNP makes all its supersets (the SNP combinations which contain the strong SNP) strong. The phenomenon makes it difficult to find epistatic interactions if SNPs with strong marginal effects are not removed. However, the epistatic interactions which contain a strong SNP will not be found, if the removal is performed.

To tackle this phenomenon, four objectives are designed. They are measures of the difference between an individual and its best element.

Change of \( ce \) (cec)

Change of \( ce \) is designed to measure the difference of \( ce \) between an individual and its best element which has the lowest \( ce \) in \( I \). The formula is shown in Equation (6). The smaller the value of cec, the more significant the epistatic interaction of \( I \) is

\[
cec(Y, I) = ce(Y|I) - ce(Y|E),
\]

where cec is one of the eight evaluation objectives used in SEE algorithm; \( I \) represents an individual; \( Y \) represents the phenotype; \( E \) is the SNP which has the lowest \( ce \) in \( I \).

Change of gini (ginic)

Change of \( g \) is designed to measure the difference of gini between an individual and its best element which has the lowest gini in \( I \). The formula is shown in Equation (7). The smaller the value of ginic, the more significant the epistatic interaction of \( I \) is

\[
ginic(Y, I) = gini(Y|I) - gini(Y|E),
\]

where ginic is one of the eight evaluation objectives used in the SEE algorithm; \( I \) represents an individual; \( Y \) represents the phenotype; \( E \) is the SNP which has the lowest gini in \( I \).

Change of \( k2 \) (k2c)

Change of \( k2 \) is designed to measure the difference of \( k2 \) between an individual and its best element which has the lowest \( k2 \) in \( I \). The formula is shown in Equation (8). The smaller the value of k2c, the more significant the epistatic interaction of \( I \) is

\[
k2c(Y, I) = k2(Y, I) / k2(Y, E),
\]

where k2c is one of the eight evaluation objectives used in the SEE algorithm; \( I \) represents an individual; \( Y \) represents the phenotype; \( E \) is the SNP which has the lowest \( k2 \) in \( I \).

Change of \( g \) (gc)

Change of \( g \) is designed to measure the difference of \( g \) between an individual and its best element which has the lowest \( g \) in \( I \). The formula is shown in Equation (9). The smaller the value of gc, the more significant the epistatic interaction of \( I \) is

\[
gc(Y, I) = \begin{cases} 
\frac{g(Y, I)}{g(Y, E)}, & g(Y, E) \neq 0 \\
1, & g(Y, E) = 0 \land g(Y, I) = 0 \\
4, & g(Y, E) = 0 \land g(Y, I) \neq 0
\end{cases}
\]

where gc is one of the eight evaluation objectives used in the SEE algorithm; \( I \) represents an individual; \( Y \) represents the phenotype; \( E \) is the SNP which has the lowest \( g \) in \( I \).
Sort the population based on the eight evaluation objectives

While multiple evaluation objectives are utilized, how to combine them is a common, difficult and interesting point. A number of methods have been tried [35, 36, 38]. In this work, a novel approach to combine the objectives is designed. Figure 2 shows a description and an example of the sort operation. First, the values of the eight objectives for each individual in the population are calculated. Second, for each objective, the rank values are assigned for each individual. Third, each rank value is multiplied by its corresponding weight. The weights are the parameters which the SEE algorithm needs. The higher the weight is, the more important its corresponding objective is. Fourth, for each individual, eight weighted rank values are added up and each individual has a summed rank value as a combined objective. Fifth, the individuals in the population are sorted based on the sum of the weighted rank values from small to large. Individuals in the front are more associated with the phenotype than those in the back.

Exploration rate (er)

Any evolutionary algorithm should have two abilities. One is the ability to make the population converge, which can make the algorithm find the optimal solution. The other is the ability to make the population jump out of the local minimum, which can make the algorithm find the global optimal solution. In SEE, exploitation which has the first ability and exploration which has the second ability are designed. Exploitation is used when the population contains many species (SNPs). It tries to make full use of the information within the population to generate a better individual. Exploration is used when the population contains few species. It tries to introduce new species into the population. The exploration rate, shown in Equation (10), is designed to determine the probability of calling the exploitation and exploration operations.

\[
\text{coverage} = \frac{\text{numSpecies}}{\text{numPop} \times \text{l}}, \\
\text{er} = 1 - \text{coverage}^{\text{pe}} 
\]  

(10)

where er is the exploration rate SEE used; numSpecies is the number of SNPs (no duplication) in the population; numPop is the number of individuals in the population; pe is a parameter specified by the user.

Exploitation

The exploitation operation is used to make full use of the information within the population to generate a better individual. The pseudo code of exploitation is shown in Algorithm 2. Exploitation randomly selects two better individuals and does crossover on them (as the genetic algorithm does) to generate a new individual.

\[\text{Algorithm 2. Exploitation}\]

1: \text{i1} = \text{min(rand(numPop), rand(numPop))} \\
2: \text{i2} = \text{min(rand(numPop), rand(numPop))} \\
3: //rand(num) is a function which returns a random integer in \([0, \text{num}]\) \\
4: //min(a, b) is a function which returns the minimal number between a and b. \\
5: \text{e} = \text{pop[i1]} \\
6: \text{i3} = \text{rand(l)} \\
7: \text{i4} = \text{rand(l)} \\
8: \text{e[i3]} = \text{pop[i2][i4]} \\
9: \text{return e as the new individual generated by exploitation.}

Exploration

The exploration operation is used to introduce new species into the population. When few species are contained in the population, the population may be trapped in local minima and the probability of calling exploration should be increased. The pseudo code of exploration is shown in Algorithm 3. Exploration randomly selects a worse individual and generates a probability based on its position in the population. According to the probability, SEE randomly selects the way of introducing new species.

\[\text{Algorithm 3. Exploration}\]

1: \text{i1} = \text{max(rand(numPop), rand(numPop))}. \\
2: //rand(num) is a function which returns a random integer in \([0, \text{num}]\). \\
3: //max(a, b) is a function which returns the maximum between a and b. \\
4: generate a random number \text{ran} \in [0, 1) \\
5: \text{if ran} < \text{i1/numPop then} \\
6: \text{randomly generate a new individual e.} \\
7: \text{else} \\
8: \text{e} = \text{pop[i1]}. \\
9: \text{i2} = \text{rand(l)}. \\
10: \text{e[i2]} = \text{rand(l)}. \\
11: //l is the number of SNPs in the GWAS data and \text{n} is the number of SNPs in an individual. \\
12: \text{return e as the new individual generated by exploration.}
Adjust

When exploitation or exploration is called, a new individual is generated. The elements in the individual are exchanged to satisfy the Inequality in Equation (1). If the Inequality cannot be satisfied, because the individual contains two or more identical elements, the adjust operation is failed. When the Inequality is satisfied, SEE adjusts the new individual using the tracing table. If the individual has already been calculated, an adjust operation is performed to generate a new individual which is the nearest to it and has not been calculated. The adjust operation is shown in Figure 3.

The description in Figure 3 is very well understood, but it needs too much memory. In order to run SEE on a common computation platform, the pseudo code shown in Algorithm 4 is the implementation of Figure 3. It is more complex but requires much less memory.

Algorithm 4. Adjust
1: the input individual is \( x \).
2: the length of \( x \) is \( l \).
3: \( n \) is the number of SNPS.
4: stepInTable is the parameter specified by users.
5: tracing is a set containing all calculated individuals.
6: waiting1 and waiting0 are empty queues.
7: if \( x \) contains two or more identical elements then
8: return adjust failed.
9: else
10: if \( x \) is not in tracing then
If $i == 0$ → stepInTable $- 1$

if waiting1 is not empty then
  get an individual from waiting1 as $x$.
  if $x$ is not in tracing then
    clear waiting1 and waiting0.
  return $x$ as the adjusted individual.
else
  for $i = 0$ → $l - 1$ do
    $xx = x$
    $xx[i] = xx[i] + 1$
    if $xx[i] \in [0, n]$ then
      if ($i == 0$ and $xx[i] > xx[i - 1]$) or ($i == 0$ and $xx[i] < xx[i + 1]$) or ($xx[i] > xx[i - 1]$ and $xx[i] < xx[i + 1]$) then
        insert $xx$ to waiting1.
    end if
  end for
if waiting0 is not empty then
  get an individual from waiting0 as $x$.
if $x$ is not in tracing then
  clear waiting1 and waiting0.
return $x$ as the adjusted individual.
else
  for $i = 0$ → $l - 1$ do
    $xx = x$
    $xx[i] = xx[i] - 1$
    if $xx[i] \in [0, n]$ then
      if ($i == 0$ and $xx[i] > xx[i - 1]$) or ($i == 0$ and $xx[i] < xx[i + 1]$) or ($xx[i] > xx[i - 1]$ and $xx[i] < xx[i + 1]$) then
        insert $xx$ to waiting0.
      end if
    end if
  end for
end if
end if

return $x$ as the adjusted individual.

**Generate result**

In the SEE algorithm, eight parameters are provided to control the result. They are $cCe$, $cCec$, $cGini$, $cGinic$, $cK2$, $cK2c$, $cG$ and $cGc$. They are the thresholds of $ce$, $cec$, $gini$, $ginic$, $k2$, $k2c$, $g$ and $gc$. When the termination criterion is satisfied, the individuals in the population are checked by the thresholds. The generated results only contain individuals whose evolution objectives are all less than or equal to the corresponding thresholds. In addition, $rn$ is the parameter that controls the number of individuals in the result. The pseudo code is shown in Algorithm 5.

**Algorithm 5.** Generate result

1: pop is the population when the termination criterion is satisfied.
2: numPop is the number of individuals in the population.
3: $cCe,cCec,cGini,cGinic,cK2,cK2c,cG,cGc$ are the thresholds of the eight evaluation objectives specified by user.
4: $rn$ is the parameter that controls the number of individuals in the result.
5: $j = 0$
6: initialize $r$ as an empty set.
7: for $i = 0$ → numPop $- 1$ do
8:   $x = pop[i]$
9:   if $x.ce <= cCe$ and $x.ce < cCec$ and $x.gini <= cGinic$ and $x.ginic <= cGinic$ and $x.k2 <= cK2$ and $x.k2c <= cK2c$ and $x.g <= cG$ and $x.gc <= cGc$ and ($rn <= 0$ or $j < rn$) then
10:      insert $x$ into $r$.
11:      $j = j + 1$
12:   end if
13: end for
14: return $r$ as the result.

**Results and discussion**

**Experiments on simulated data**

In the experiments on simulated data, the common goal was to detect the specific two-locus SNP combination (target) in each artefact epistasis model. The
Disease loci with and without marginal effects were used to test BEAM, DECMDR, MACOED, SNPHarvester, SNPRuler, AntEpiSeeker and SEE. The types of software used in this work are listed in Table 1.

**Table 1. Software used in this study.**

| Software | Language |
|----------|----------|
| BEAM | C++ |
| DECMDR | Java |
| MACOED | Matlab |
| SNPHarvester | Java |
| SNPRuler | Java |
| AntEpiSeeker | C++ |
| SEE | C++ |

Simulated datasets

In this work, two different types of epistasis models commonly used in generating simulated datasets are presented: the DME model (disease loci with marginal effects) and the DNME model (disease loci without marginal effects).

Twelve DME models were obtained from MACOED [35] and 60 DNME models from DECMDR [37]. To study the effects of the algorithm, 100 datasets (files) were generated for each model (penetrance table), with each dataset containing 100 SNPs and 1600 samples (800 cases and 800 controls). All datasets were

![Simulated datasets](image-url)
Table 2. Parameter settings in DME.

| Software          | Parameters                                      |
|-------------------|-------------------------------------------------|
| BEAM              | CHAIN = 1,BURNIN = 0;MCMC = 0;THIN = 0;PRIOR1 = 0.01;PRIOR2 = 0.01;SINGLE_ONLY = 0;MINDISTANCE = 1000000;INITIALTRYS = 20;TRY_LENGTH = 0;AUTORESTART = 5 |
| DECMDR            | s = 1;p = 80;g = 40;m = 0.5;r = 0.5,o = 2        |
| MACOED            | num_ant = 80;max_iter = 40;dim_epi = 2;value = 0.05/4950;tau0 = 1;10 = 0.8;rou = 0.9;lambda = 2 |
| SNPHarvester      | listSize = 80;depth = 2;updateRatio = 0.2      |
| AntEpiSeeker      | iAntCount = 80;IntCountLarge = 10;iTopModel = 80;iTopLoci = 16;rou = 0.05;pha = 100;largehapsize = 6;smallhapsize = 3;iEpiModel = 2;value = 0.01 |
| SEE               | wCe = 1;wGec = 1;wGinic = 1;wK2 = 1;wK2c = 1;wG = 1;wCc = 1;wCc = maxcCc = 0;cGini = maxcGinic = 0;ccK2 = maxccK2c = 1;G = 0.05;Cc = 1;r = 1;pe = 0.4;numPop = 80;order = 2;stepInTable = 4;maxiter = 80^4;where max means the maximum double value in C++ programming language |

Note: For each piece of software, the recommended parameters were used in the experiments. For fair comparison, all evolutionary algorithms (MACOED, DECMDR, AntEpiSeeker and SEE) had the same number of generated new individuals (SNP combinations). In AntEpiSeeker, it is recommended that iItCountLarge = 0.5 × iItCountSmall. iItCountLarge was set to 10 and iItCountLarge was set to 30 for fairness.

simulated by the software GAMETES_2.1. In each simulated dataset, the SNPs in the last two columns were the disease associated ones. In order to eliminate the effect of the location, two columns were randomly selected to swap places with the last two columns before a simulated dataset was analyzed. The detailed description of the simulated datasets is given in the Supplementary Appendix and the penetrance tables are provided in Supplemental Table S1.

Evaluation criteria

In order to evaluate the performance of the epistasis-learning algorithm, the power and F-measure were utilized as the evaluation criterion simultaneously. They are defined as Equation (11). Power is a traditional evaluation criterion in GWAS. F-measure, recall and precision are commonly used in the pattern recognition field. F-measure is the harmonic mean of precision and recall. It is chosen as the second evaluation criteria. In order to evaluate the performance of the epistasis-learning algorithm, the power and F-measure were utilized as the evaluation criterion simultaneously. They are defined as Equation (11).

\[
\begin{align*}
\text{power} & = \frac{\#(S)}{100} \\
\text{recall} & = \frac{TP}{TP + FN} \\
\text{precision} & = \frac{TP}{TP + FP} \\
F\text{-measure} & = 2 \cdot \frac{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} 
\end{align*}
\]

where power and F-measure are two criteria used in this work. \#(S) is the number of datasets in which the disease-specific two-locus SNP combination is successfully identified among all 100 datasets generated by the same parameters and penetrance table; TP represents the true positives; FP represents the false positives and FN represents the false negatives.

Comparison of SEE with existing methods on disease loci with marginal effects

In this experiment, we compared SEE with six other existing methods, including BEAM, DECMDR, MACOED, SNPRuler, SNPHarvester and AntEpiSeeker. The parameters were set as shown in Table 2. For each DME model and method, 100 data files were analyzed and the evaluation criteria were calculated as Equation (11).

Figure 4 presents the power performance comparisons on the DME models. SEE outperforms the other methods on most of the DME models. On DME_01-04, because the genetic heritability is too low (0.005), distinguishing between the associated SNP combination and unassociated ones is very difficult. SNPRuler and AntEpiSeeker return many unassociated ones as a result, so they perform well on DME01-04. With the increase of the genetic heritability (0.02 on DME05-12), SEE performs better on power. With the DME models, the highest average power of the other six methods is 0.58 (generated by AntEpiSeeker). The average power of SEE is 0.71. Figure 5 presents the F-measure performance comparisons on the DME models. SEE outperforms the others with all DME models, except for DME04. The highest average F-measure of the other six methods is 0.44 (generated by DECMDR). The average F-measure of SEE is 0.68. Figure 6 presents the computation time comparisons on the DME models. SEE outperforms the others with all DME models. The lowest average computation time of the other six methods is 2.12 (generated by SNPRuler). The average computation time of SEE is 0.21. The detailed
The experiment results are shown in Supplemental Tables S2–S4. It is indicated that with DME models, SEE outperforms other existing methods on power, F-measure and computation time.

**Comparison of SEE with existing methods on disease loci without marginal effects**

In this experiment, we compared SEE with six other existing methods, including BEAM, DECMDR, MACOED, SNPHarvester, SNPRuler and AntEpiSeeker. The parameters were set as shown in Table 3. As the DNME models belong to pure epistatic models and the population of any evolutionary algorithms has less information to utilize, the max iteration on these models was set higher than that on the DME models. For the same reason, the pe of SEE was set higher to increase the probability of calling exploration on the DNME models. For each DNME model and method, 100 data files were analyzed and the evaluation criteria were calculated as Equation (11).

Figure 4. Power performance comparisons between BEAM (B), DECMDR (D), MACOED (M), SNPHarvester (H), SNPRuler (R), AntEpiSeeker (S) and SEE (W) with the 12 DME models. For each DME model, the power is calculated as shown in Equation (11). The absence of bars indicates zero power.
Figure S1 in the Supplementary Appendix presents the power performance comparisons on the DNME models. SEE outperforms the other methods on all DNME models. The highest average power of the other six methods is 0.97 (generated by SNPRuler). The average power of SEE is 0.99. Figure 7 presents the F-measure performance comparisons on the DNME models. SEE outperforms the others on all DNME models. The highest average F-measure of the other six methods is 0.90 (generated by MACOED). The average F-measure of SEE is 0.99. Figure S2 in the Supplementary Appendix presents the computation time comparisons on the DNME models. SEE outperforms the others on all DNME models. The lowest average computation time of the other six methods is 2.09 (generated by SNPRuler). The average computation time of SEE is 0.40. The detailed experiment results are shown in Supplemental Tables S2–S4. It is indicated that on DNME models, SEE outperforms other existing methods on power, F-measure and computation time.
Experiments on real data

As shown in the results on simulated data, SEE has good performance. SEE was then utilized to conduct experiments on a real dataset obtained from the Wellcome Trust Case Control Consortium (WTCCC) [42] which had about 2000 individuals examined for each of seven major diseases and a shared set of about 3000 controls. After removing the SNPs and samples recommended by WTCCC and combining each dataset of diseases with the shared dataset of controls, seven GWAS datasets were generated. For each of them, SNPs which were unchanged in all samples were removed. Refer to Supplemental Tables S5–S7 for a more detailed description of the seven real GWAS datasets.

As shown in Table 4, the SNPs found by SEE were mapped to genes based on the NCBI database [48]. The genes were then searched in the CTD database (Comparative Toxicogenomics Database) [49]. A lot of genes which had DE (Direct Evidence) or NDE (Not
Table 3. Parameter settings in DNME.

| Software       | Parameters                                                                 |
|----------------|-----------------------------------------------------------------------------|
| BEAM           | CHAIN = 1;BURNIN = 0;MCMC = 0;THIN = 0;PRIOR1 = 0.01;PRIOR2 = 0.01;SINGLE_ONLY = 0;MINDISTANCE = 1000000;INITIALTRYS = 0;TRY_LENGTH = 0;AUTORESTART = 5 |
| DECMDR         | s = 1;p = 80;g = 140;m = 0.5;r = 0.5;o = 2                                  |
| MACOED         | num_ant = 80;max_iter = 140;dim_epi = 2;pvalue = 0.05/4950;tau0 = 1.70;tau = 0.8;rou = 0.9;lambda = 2   |
| SNPHarvester   | No parameter is set and all parameters take default values.                |
| SNPRuler       | listSize = 80;depth = 2;updateRatio = 0.2                                |
| AntEpiSeeker   | iAntCount = 80;iItCountLarge = 40;iItCountSmall = 10;alpha = 1;iTopModel = 80;iTopLoci = 16;rou = 0.05;pe = 1.0;large-hapsize = 6;smallhapsize = 3;EpiModel = 2;pvalue = 0.01 |
| SEE            | wCe = 1.0;wCec = 1.0;wGini = 1.0;wGinic = 1.0;wK2 = 1.0;wK2c = 1.0;wG = 1.0;wGc = 1.0;cCe = maxcCec = 0.0;Cgini = maxcCinic = 0.0;K2 = maxcK2c = 1.0;Gc = 0.05;GcG = 1.0;pe = 0.4;numPop = 80;order = 2;stepnTable = 4;maxIter = 80*140;where max means the maximum double value in C++ programming language |

Note: For each piece of software, the recommended parameters were used in the experiments. For fair comparison, all evolutionary algorithms had the same number of generating new individuals (SNP combinations). In AntEpiSeeker, it is recommended that iItCountLarge = 0.5 × iItCountSmall. iItCountLarge was set to 40 and iItCountLarge was set to 100 for fairness.

Direct Evidence) association with the corresponding disease were found.

As shown in Figure 8, in order to improve the readability of the results, after mapping SNPs to genes, CytoScape [50] was utilized to make networks based on the gene combinations found by SEE. Figure 8 indicates that CSGALNACT1, LOC105375925, LOC107984858, CENPN etc. play a very important role in bipolar disorder. More details are shown in Figures S3–S9 of the Supplementary Appendix. It is indicated that CACNG1, CELF4, PNLIPRP3, CSGALNACT1 etc. play a key role in coronary artery disease. LDB2, LOC107986262, SMG1P5 and RRP15 etc. play an important role in Crohn's disease. LOC101929163, C6orf110, CACNG1, FRMPD4 etc. play an important role in rheumatoid arthritis. LOC101929163, C6orf110, GPSM3, NOTCH4 etc. have a key role in type 1 diabetes. HTR3B, KIAA1671, CSGALNACT1 and C4orf50 etc. are important role in type 2 diabetes.

Comparison of the methods

BEAM and SNPHarvester both belong to the category of stochastic search. As shown in the experiments on simulated data, SNPHarvester is much better than BEAM.

SNPRuler is an exhaustive filter search method, it is claimed that SNPRuler cannot perform well on DNME models. However, in the experiments on DNME models, SNPRuler performs better than most of the others on power. On F-measure, SNPRuler performs poorly.

DECMDR, MACOED, AntEpiSeeker and SEE all belong to the category of evolution search. MACOED and AntEpiSeeker both utilize ACO (ant colony optimization) to search the SNP combinations. The optimization algorithm of DECMDR is DE (differential evolution). The optimization algorithm of SEE is designed by us for detecting SNP combinations. On real GWAS data, the table in ACO to record the pheromone of each SNP pair is too large to store in the memory of a common computation platform. DECMDR and SEE use less memory in optimization. In addition, SEE utilizes a boolean representation of genotype data to store the GWAS data as BOOST does, so the memory it needs is very small. While calculating the eight evaluation objectives, the contingency table is collected by the boolean operation. The computation time of SEE is the lowest among the algorithms. DECMDR utilizes only MDR as the evaluation objective. MDR is a traditional measure of the association of the SNP combination and the phenotype. However, MDR is highly dependent on the random seed (how to divide the samples into folds). The MDR value varies, sometimes very much, when the random seed is changed. AntEpiSeeker uses the chi-square test to measure the SNP combination. MACOED uses logistic regression and BN (similar to $k_2$) as the objective in ACO, and then the chi-square test is conducted. As logistic regression is very time consuming, the computation time of MACOED is the highest among the algorithms. Although SEE calculates eight evaluation objectives for each SNP combination, because of the boolean operation, the calculation is very fast. As eight evaluation objectives are utilized to measure an SNP combination, the population in SEE is more informative and less generation is needed to detect the right SNP combination. The latter four evaluation objectives are designed by us to distinguish real epistasis SNP combinations from the combinations which contain a very associated SNP.

The advantages of SEE

SEE has higher power and F-measure. As eight evaluation objectives are utilized simultaneously, the power
Figure 7. F-measure comparisons between BEAM (B), DECMDR (D), MACOED (M), SNPHarvester (H), SNPRuler (R), AntEpiSeeker (S), SEE (W) on the 60 DNME models. For each DNME model, the F-measure is calculated as shown in Equation (11). The absence of bars indicates zero F-measure.
of SEE is higher. When generating a new SNP combination by SEE, there is more information to utilize in the population. The adjust operation in SEE also increases the power for detecting SNP combinations. SEE can distinguish real epistasis SNP combinations from the combinations which contain a very associated SNP.

SEE needs much less memory. As the GWAS data are stored by boolean representation, they take up much less memory than most other existing methods. SEE needs only about 1 GB memory while analyzing the GWAS data which contains about 4000 samples and 500,000 SNPs. Unlike most other methods, it can perform data analysis steadily on most computing platforms.

SEE is very fast. As SEE is written in C++ and takes advantage of boolean operation greatly, the computation time of SEE is very low.

The limitations of SEE

Although the code of SEE has been optimized again and again, the sort operation is still very time consuming. Another method should be designed to combine multiple evaluation objectives.

Conclusions

In this paper, we propose a novel algorithm called SEE for identifying SNP epistasis in genome-wide association studies. It is successfully designed to combine eight evolution objectives and a newly designed evolutionary algorithm. It was assessed through comparison with six other existing approaches on 72 simulated datasets. It is indicated that SEE is powerful to handle GWAS data both in terms of F-measure and computation time. We then utilized it to analyze real data obtained from the Wellcome Trust Case Control Consortium. A lot of disease-associated SNPs and genes (reported or not on CTD database) has been found. The results may be helpful to further explain the seven diseases. We believe that SEE will be widely used in GWAS.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the National Nature Science Foundation of China [grant numbers 61772226, 61373051,
Table 4. Part of the SEE results for seven diseases in the WTCCC dataset.

| Chrom | SNP1 | Gene1 | CTD | SNP2 | Gene2 | CTD | p-value (G-test) |
|-------|------|-------|-----|------|-------|-----|-----------------|
| Bipolar disorder | | | | | | | |
| All | rs17480050 | CSGALNACT1 | NDE | rs7325200 | HTR2A | DE | 0 |
| chr 1 | rs4633297 | UNKNOWN | NDE | rs6588191 | PDE4B | DE | 5.31E-05 |
| chr 5 | rs17209251 | NRC3C1 | DE | rs10054583 | PDZD2 | NDE | 0.005268119 |
| chr 9 | rs11243677 | ANKG2 | DE | rs914715 | TLE4 | NDE | 0.0011108 |
| chr 10 | rs1938526 | ANK3 | DE | rs2488084 | UNKNOWN | UNKNOWN | 0.004175791 |
| chr 11 | rs567236 | TENM4 | DE | rs12422135 | UNKNOWN | UNKNOWN | 0.003828214 |
| chr 12 | rs17703136 | GRIP1 | NDE | rs1006737 | CACNA1C | DE | 0.002946767 |
| chr 22 | rs17435801 | UNKNOWN | UNKNOWN | rs16980239 | GRK3 | DE | 0.00625281 |
| Coronary artery disease | | | | | | | |
| All | rs3785579 | CACNG1 | NDE | rs3027898 | IRAK1 | DE | 0 |
| chr 1 | rs4846770 | MIA3 | DE | rs5027299 | KDM1A | NDE | 2.31E-07 |
| chr 2 | rs13139968 | CDDC148 | DE | rs13600247 | LOC102275159|ABCG8 | NDE|DE | 0.002844722 |
| chr 3 | rs1199332 | MRAS | DE | rs17042882 | PLCL2 | NDE | 0 |
| chr 6 | rs519758 | GFO1D | DE | rs6923943 | PHACTR1 | DE | 2.43E-06 |
| chr 7 | rs10270161 | UNKNOWN | UNKNOWN | rs11556924 | ZC3HC1 | DE | 0.00088387 |
| chr 10 | rs11226078 | PDZD2 | NDE | rs914715 | TLE4 | NDE | 0.0010118 |
| chr 11 | rs1938526 | ANK3 | DE | rs2488084 | UNKNOWN | UNKNOWN | 0.004175791 |
| chr 12 | rs17703136 | GRIP1 | NDE | rs1006737 | CACNA1C | DE | 0.002946767 |
| chr 22 | rs17435801 | UNKNOWN | UNKNOWN | rs16980239 | GRK3 | DE | 0.00625281 |
| Crohn’s disease | | | | | | | |
| All | rs11209026 | IL23R | DE | rs1933641 | RRP15 | NDE | 0 |
| chr 1 | rs12109026 | IL23R | DE | rs1933641 | RRP15 | NDE | 0 |
| chr 2 | rs10210302 | ATG16L1 | DE | rs4080478 | UNKNOWN | UNKNOWN | 9.08E-12 |
| chr 3 | rs4135229 | PPARG | DE | rs11712729 | ERC2 | NF | 0.00887644 |
| chr 5 | rs2416472 | UNKNOWN | UNKNOWN | rs1000113 | IRGM | DE | 4.31E-11 |
| chr 7 | rs9650985 | LHFPL3 | NF | rs17151914 | LEP | DE | 0.00036593 |
| chr 11 | rs10883371 | LOC101928387 | NF | rs7489213 | GRK3 | DE | 0.017E-07 |
| chr 16 | rs4471699 | SMG1PS | NF | rs2076756 | NOD2 | DE | 0 |
| chr 18 | rs7234029 | PTN2 | DE | rs4496495 | LOC103572022|CABLES1 | NF|NDE | 0.00016182 |
| Rheumatoid arthritis | | | | | | | |
| All | rs3785579 | CACNG1 | NDE | rs16992357 | RUNX1 | DE | 0 |
| chr 1 | rs2488457 | AP4B1-AS1 | NF|DE | rs7393166 | TIMEM51|C1orf195 | NF|NF | 0 |
| chr 2 | rs4578903 | AFF3 | DE | rs1440065 | UNKNOWN | UNKNOWN | 0 |
| chr 5 | rs6859219 | ANKRDS5 | DE | rs1689046 | LOC10378998 | NF | 0 |
| chr 7 | rs9276435 | HLA-DQA2 | DE | rs4959053 | PSORS1C1 | NF | 0 |
| chr 12 | rs4502063 | CLEC12A | DE | rs10843606 | UNKNOWN | UNKNOWN | 0 |
| chr 15 | rs16942398 | ACAN | DE | rs16942813 | AKAP13 | DE | 0 |
| Type 1 diabetes | | | | | | | |
| All | rs2488457 | AP4B1-AS1 | NF|DE | rs9273363 | HLA-DQ8B1-AS1 | NF | 0 |
| chr 1 | rs2488457 | AP4B1-AS1 | NF|DE | rs3790857 | PGM1 | DE | 2.93E-08 |
| chr 3 | rs805006 | ITTP1 | DE | rs16861590 | CPF | DE | 0 |
| chr 6 | rs9272723 | LOC107986589|HLA-DQA1 | NF|DE | rs9275523 | UNKNOWN | UNKNOWN | 0 |
| chr 9 | rs7020673 | LOC107986589|HLA-DQA1 | NF|DE | rs9275523 | UNKNOWN | UNKNOWN | 0 |
| chr 10 | rs7078717 | NRG3 | NF | rs942200 | IL2RA | DE | 0 |
| Type 2 diabetes | | | | | | | |
| All | rs17480050 | CSGALNACT1 | NDE | rs7070399 | TCF7L2 | DE | 0 |
| chr 1 | rs16850854 | LOC100506747 | NF | rs340835 | PROX1 | DE | 0.009939957 |
| chr 4 | rs16837871 | C4orf50 | NF | rs5341 | EDNRA | DE | 0 |
| chr 6 | rs2328549 | CDKAL1 | DE | rs4398751 | UNKNOWN | UNKNOWN | 0.00129422 |
| chr 7 | rs1477528 | LOC349160 | NF | rs1715637 | JAZF1 | DE | 0 |
| chr 8 | rs4669672 | SLC30A8 | DE | rs17480050 | CSGALNACT1 | NDE | 0 |
| chr 9 | rs16919975 | GLIS3 | DE | rs7039577 | BRINP1 | NDE | 0.008634383 |
| chr 10 | rs17094393 | GFRAD1 | NDE | rs7094195 | TCF7L2 | DE | 2.06E-08 |
| chr 11 | rs17161117 | HTRB3 | DE | rs2188966 | ABCB1 | DE | 0 |
| chr 16 | rs41356047 | UNKNOWN | UNKNOWN | rs8050136 | FTO | DE | 3.87E-06 |
| chr 20 | rs16999183 | PCSK2 | DE | rs6094514 | UNKNOWN | UNKNOWN | 1.20E-09 |

Note: The complete results are given in Supplemental Table S8. DE means the gene has Direct Evidence for association with the disease in the CTD database. NDE means the gene has Not Direct Evidence. NF means the gene is not found in the CTD database. Unknown means the SNP cannot be mapped to any gene based on the NCBI database.

61502343); Science and Technology Development Program of Jilin Province [grant number 20140204004G]; Science Research Funds for the Guangxi Universities [grant number KY2015ZD122]; Science Research Funds for the Wuzhou University [grant number 2014A002]; Project of Science and Technology Innovation Platform of Computing and Software Science (985 Engineering); Key Laboratory for Symbol Computation and Knowledge Engineering of the National Education Ministry of China; Fundamental Research Funds for the Central.
References

[1] Collins A, Lonjou C, Morton N. Genetic epidemiology of single-nucleotide polymorphisms. Proc Natl Acad Sci USA. 1999;96:15173–15177.
[2] Schork NJ, Fallin D, Lanchbury JS. Single nucleotide polymorphisms and the future of genetic epidemiology. Clin Genet. 2001;58:250–264.
[3] Schork NJ, Murray SS, Frazer KA, et al. Common vs. rare allele hypotheses for complex diseases. Curr Opin Genet Dev. 2009;19:212–219.
[4] Cordell HJ. Epistasis: what it means, what it doesn’t mean, and statistical methods to detect it in humans. Hum Mol Genet. 2002;11:2463–2468.
[5] Cordell HJ. Detecting gene–gene interactions that underlie human diseases. Nat Rev Genet. 2009;10:392–404.
[6] Shang J, Zhang J, Yan S, et al. Performance analysis of novel methods for detecting epistasis. BMC Bioinform. 2011;12:475.
[7] Wei WH, Hemani G, Haley CS. Detecting epistasis in human complex traits. Nat Rev Genet. 2014;15:722–733.
[8] Mackay TF, Moore JH. Why epistasis is important for tackling complex human disease genetics. Genome Med. 2014;6:42.
[9] Hirschhorn JN, Daly MJ. Genome-wide association studies for common diseases and complex traits. Nat Rev Genet. 2005;6:95–108.
[10] Pearson TA, Manolio TA. How to interpret a genome-wide association study. JAMA. 2008;299:1335–1344.
[11] Manolio TA. Genomewide association studies and assessment of the risk of disease. N Engl J Med. 2010;363:166–176.
[12] Li J, Humphreys K, Darabi H, et al. A genome-wide association scan on estrogen receptor-negative breast cancer. Breast Cancer Res. 2010;12:R93.
[13] Shan J, Mahfouwd W, Dsouza SP, et al. Genome-Wide Association Studies (GWAS) breast cancer susceptibility loci in Arabs: susceptibility and prognostic implications in Tunisians. Breast Cancer Res Treat. 2012;135:715–724.
[14] Chen F, Chen GK, Stram DO, et al. A genome-wide association study of breast cancer in women of African ancestry. Hum Genet. 2013;132:39–48.
[15] Ritchie MD, Hahn LW, Roodi N, et al. Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. Am J Hum Genet. 2001;69:138–147.
[16] Purcell S, Neale B, Todd-Brown K, et al. PLINK: a tool set for whole-genome association and population-based linkage analyses. Am J Hum Genet. 2007;81:559–575.
[17] Gynensie A, Moody J, Laio H, et al. BiForce Toolbox: powerful high-throughput computational analysis of gene–gene interactions in genome-wide association studies. Nucleic Acids Res. 2012;40:628–632.
[18] Zhu Z, Tong X, Zhu Z, et al. Development of GMDR-GPU for gene-gene interaction analysis and its application to WTCCC GWAS data for type 2 diabetes. PLoS One. 2013;8:e61943.
[19] Wang Q, Shi F, Kowalczyk A, et al., editors. GWISF: a universal GPU interface for exhaustive search of pairwise interactions in case-control GWAS in minutes. In: IEEE International Conference on Bioinformatics and Biomedicine; 2014 Nov 2–5; Belfast, UK; 2014. p. 403–409.
[20] Zhang Y, Liu JS. Bayesian inference of epistatic interactions in case-control studies. Nat Genet. 2007;39:1167–1173.
[21] Peng T, Du P, Li Y. PBEAM: a parallel implementation of BEAM for genome-wide inference of epistatic interactions. Bioinformation. 2009;3:349–351.
[22] Yang C, He Z, Wan X, et al. SNPHarvester: a filtering-based approach for detecting epistatic interactions in genome-wide association studies. Bioinformatics. 2009;25:504–511.
[23] Marchini J, Donnelly P, Cardon LR. Genome-wide strategies for detecting multiple loci that influence complex diseases. Nat Genet. 2005;37:413–417.
[24] Zhang X, Zou F, Wang W, et al. FastChi: an efficient algorithm for analyzing gene-gene interactions. In: Altman RB, Dunker AK, Hunter L, editors. Pacific Symposium on Biocomputing 2009; 2009 Jan 5–9; Kohala Coast, Hawaii, USA. Stanford University; 2009. p. 528–539. Available from: http://psb.stanford.edu/phs-online/proceedings/psb09/.
[25] Wang X, Yang C, Yang Q, et al. BOOST: a fast approach to detecting gene-gene interactions in genome-wide case-control studies. Am J Hum Genet. 2010;87:325–340.
[26] Jiang R, Tang W, Wu X, et al. A random forest approach to the detection of epistatic interactions in case-control studies. BMC Bioinform. 2009;10:1–12.
[27] Wu TT, Chen YF, Hastie T, et al. Genome-wide association analysis by lasso penalized logistic regression. Bioinformatics. 2009;25:714–721.
[28] Schwarz DF, König IR, Ziegler A. On safari to Random Jungle: a fast implementation of random forests for high-dimensional data. Bioinformatics. 2010;26:1752–1758.
[29] Yoshida M, Koike A. SNPInterForest: a new method for detecting epistatic interactions. BMC Bioinform. 2011;12:469.
[30] Li J, Horstman B, Chen Y. Detecting epistatic effects in association studies at a genomic level based on an ensemble approach. Bioinformatics. 2011;27:2222–2229.
[31] Wu Q, Ye Y, Liu Y, et al. SNP selection and classification of genome-wide SNP data using stratified sampling random forests. IEEE Trans on Nanobiosci. 2012;11:216–227.
[32] Nguyen TT, Huang JZ, Wu Q, et al. Genome-wide association data classification and SNPs selection using two-stage quality-based random forests. BMC Genomics. 2015;16:55.
[33] Wang Y, Liu X, Robbins K, et al. AntEpiSeeker: detecting epistatic interactions for case-control studies using a two-stage ant colony optimization algorithm. BMC Res Notes. 2010;3:117.
[34] Aflakparast M, Salimi H, Gerami A, et al. Cuckoo search epistasis: a new method for exploring significant genetic interactions. Heredity. 2014;112:666–674.
[35] Jing P-J, Shen H-B. MACOED: a multi-objective ant colony optimization algorithm for SNP epistasis detection in genome-wide association studies. Bioinformatics. 2014;31:634–641.

[36] Tuo S, Zhang J, Yuan X, et al. FHSA-SED: two-locus model detection for genome-wide association study with harmony search algorithm. PLoS One. 2016;11:e0150669.

[37] Yang CH, Chuang LY, Lin YD. CMDR based differential evolution identify the epistatic interaction in genome-wide association studies. Bioinformatics. 2017;33:2354–2362.

[38] Sun Y, Shang J, Liu JX, et al. epiACO – a method for identifying epistasis based on ant colony optimization algorithm. Biodata Min. 2017;10:23.

[39] Yuan L, Yuan CA, Huang DS. FAACOSE: a fast adaptive ant colony optimization algorithm for detecting SNP epistasis. Complexity. 2017;2017:1–10.

[40] Tuo S, Zhang J, Yuan X, et al. Niche harmony search algorithm for detecting complex disease associated high-order SNP combinations. Sci Rep. 2017;7:11529.

[41] Wan X, Yang C, Yang Q, et al. Predictive rule inference for epistatic interaction detection in genome-wide association studies. Bioinformatics. 2010;26:30–37.

[42] Burton PR, Clayton DG, Cardon LR, et al. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. Nature. 2007;447:661–678.

[43] Breiman L. Classification and regression trees. New York (NY): Routledge; 1984.

[44] Visweswaran S, Wong AKI, Barmada MM. A Bayesian method for identifying genetic interactions. AMIA Annu Symp Proc/AMIA Symp AMIA Symp. 2009;2009:673–677.

[45] Cooper GF, Herskovits E. A Bayesian method for the induction of probabilistic networks from data. Mach Learn. 1992;9:309–347.

[46] Hoey J. The two-way likelihood ratio (G) test and comparison to two-way chi squared test; 2012. (arXiv preprint arXiv:12064881).

[47] Stamatis DH. Essential statistical concepts for the quality professional. Boca Raton (FL): CRC Press; 2012.

[48] Sherry ST, Ward MH, Kholodov M, et al. dbSNP: the NCBI database of genetic variation. Nucleic Acids Res. 2001;29:308–311.

[49] Peter DA, Grondin MC, Robin J, et al. The Comparative Toxicogenomics Database: update 2013. Nucleic Acids Res. 2011;41:D1104–D1114.

[50] Kohl M, Wiese S, Warscheid B. Cytoscape: software for visualization and analysis of biological networks. In: Hamacher M, Eisenacher M, Stephan C, editors. Data mining in proteomics: from standards to applications. Totowa (NJ): Humana Press; 2011. p. 291–303.