Guo *et al. BMC Bioinformatics* (2019) 20:444
https://doi.org/10.1186/s12859-019-3022-z

BMC Bioinformatics

METHODOLOGY ARTICLE Open Access

Epi-GTBN: an approach of epistasis mining based on genetic Tabu algorithm and Bayesian network

Yang Guo[†], Zhiman Zhong[†], Chen Yang, Jiangfeng Hu, Yaling Jiang, Zizhen Liang, Hui Gao and Jianxiao Liu[*]

Abstract

Background: Mining epistatic loci which affects specific phenotypic traits is an important research issue in the field of biology. Bayesian network (BN) is a graphical model which can express the relationship between genetic loci and phenotype. Until now, it has been widely used into epistasis mining in many research work. However, this method has two disadvantages: low learning efficiency and easy to fall into local optimum. Genetic algorithm has the excellence of rapid global search and avoiding falling into local optimum. It is scalable and easy to integrate with other algorithms. This work proposes an epistasis mining approach based on genetic tabu algorithm and Bayesian network (*Epi-GTBN*). It uses genetic algorithm into the heuristic search strategy of Bayesian network. The individual structure can be evolved through the genetic operations of selection, crossover and mutation. It can help to find the optimal network structure, and then further to mine the epistasis loci effectively. In order to enhance the diversity of the population and obtain a more effective global optimal solution, we use the tabu search strategy into the operations of crossover and mutation in genetic algorithm. It can help to accelerate the convergence of the algorithm.

Results: We compared *Epi-GTBN* with other recent algorithms using both simulated and real datasets. The experimental results demonstrate that our method has much better epistasis detection accuracy in the case of not affecting the efficiency for different datasets.

Conclusions: The presented methodology (*Epi-GTBN*) is an effective method for epistasis detection, and it can be seen as an interesting addition to the arsenal used in complex traits analyses.

Keywords: Epistasis, Genetic algorithm, Tabu, Bayesian network

Background

With the rapid development of many high-throughput technologies, massive biological data has been produced in recent years, such as genome, transcription and phenotype data. It is possible to mine genetic loci affecting specific phenotypic traits (such as agronomic traits, human diseases, etc.) using the genome data. And it has also become a challenging research topic in today's biological field. Genome-Wide Association Study (GWAS) is a common method for detecting Single Nucleotide Polymorphism (SNP) associated with phenotypes in the whole genome. This method mainly focuses on the detection of major genes, but it cannot detect gene-gene interactions, or epistasis, mainly embodied in the interaction between SNPs. It needs to develop new approaches to mine the epistatic interactions for specific phenotypic traits.

At present, the following four kinds of methods are mainly used for the epistasis detection in case-control study: statistical method, information entropy method, multi-stage method, machine learning method.

(i). Statistical method. The logistic regression is a method used earlier to detect epistasis [1]. Later the improved logistic regression based on Group Lasso method is used into epistasis mining [2]. The result of logistic regression method is easy to explain, but it has the problems of overfitting, model decline

[*] Correspondence: liujianxiao321@163.com
[†]Yang Guo and Zhiman Zhong contributed equally to this work.
Hubei Key Laboratory of Agricultural Bioinformatics, College of Informatics, Huazhong Agricultural University, Wuhan 430070, People's Republic of China

and large amount of calculation. Other statistical methods mainly include functional regression model [3], statistical epistasis networks [4], variance analysis-based method named FastANOVA [5], etc. However, the efficiency of statistical methods is often not high and it needs to set more statistical factors and more complex parameters. Multifactor-Dimensionality Reduction (MDR) is a commonly used epistasis detection approach. But it adopts the exhaustive search strategy, leading to large amount of computation [6]. Subsequently, some researchers improved the MDR method, including MB-MDR, GMDR, FMDR, QMDR, UM-MDR [7], Crush-MDR, KNN-MDR, CMDR [8] and so on. In all, the MDR related methods are more complex when to deal with multiple locus, and the results obtained by these methods are difficult to explain, resulting in poor practicability. Bayesian theory is also used for epistasis mining. The representative method is Zhang's BEAM method [9], and subsequent improvement methods, including BEAM2, BEAM3 and JBASE [10]. However, the Bayesian theory related methods are complex, inefficient and with insufficient accuracy.

(ii). Information entropy method. In 2008, Dong et al. applied the information entropy theory to detect gene loci epistasis, and verified this method on simulated and real malaria dataset [11]. Hu et al. used the information gain approach to detect three-way epistatic interactions [12]. Kwon et al. dealt with the low-order and high-order epistatic interaction respectively based on the information entropy theory [13]. Li judged the epistasis using information entropy and Bayesian network K2 scoring method [14]. Besides, some research work use the information gain method, such as MBS-IGain [15], Exhaustive-IGain [16].

(iii) Multi-stage method. This kind of method firstly screens out a few important loci and then detects epistasis of the selected important locus. The representative methods include *SNPHarvester* [17], *SNPRuler* [18], *LEAP* [19], *EPIQ* [20], etc. On the basis of Boolean operation, *BOOST* detects epistasis using the stages of screening and testing [21]. The efficiency of this method is relatively high, but it is limited to the interaction between two SNPs, which leads to limited utility. The most important step in multi-stage method is to find the appropriate screening criteria, but it is easy to leave out some important epistasis loci.

(iv). Machine learning method. At present, machine learning methods are increasingly used for mining epistatic loci, such as random forest [22], support vector machine [23], association rules [24], neural network, etc. However, the biggest drawback of machine learning method is that the result is difficult to explain, and tends to overfitting. It often requires cross validation, resulting in high computation cost. In order to improve the detection efficiency and get the global optimal solution, some researchers use the evolutionary algorithm into epistasis mining, such as genetic algorithm (GA) [25], particle swarm optimization [26], etc. Ant colony algorithm is a frequently used heuristic search method for epistasis detection, including *AntEpiSeeker* [27], *AntMiner*, *MACOED* [28], *epiACO* [29], *FAACOSE* [30], etc. However, ant colony algorithm has some disadvantages, such as difficulty in determining the control parameters, premature stagnation and slow convergence in the early stage, which will affect the calculation accuracy.

Compared with other methods, Bayesian network has the advantages of constructing the causal relationship between objects, mining implicit knowledge, processing data with nonlinear relationship and noise, dealing with different data types, etc. In recent years, some research work use Bayesian network learning method to construct the network of gene loci and phenotype, and thus to detect the epistatic loci for specific phenotype [31, 32]. In this work, we firstly construct the network of gene loci for specific phenotype using Bayesian network, and then mine the epistasis for specific phenotype. However, due to Bayesian network usually uses the partial or random search strategy, it is easy to fall into local optimum and further to influence the learning accuracy. Genetic algorithm has the characteristic of rapid global search and avoiding falling into local optimization. In this approach, we use the genetic algorithm into the heuristic search strategy of Bayesian network. The evolution of individual structure is realized through three genetic operations (selection, crossover, mutation), and thus to find the optimal network structure. Inspired by the genetic tabu algorithm used in [33, 34], we use the tabu search strategy into the crossover and mutation operation of genetic algorithm. It can help to enhance the diversity of population and thus to obtain the global optimal solution. In the genetic algorithm, the quality of the initial population has an important effect on the result. We use mutual information entropy calculation method to calculate the relationship between gene loci and phenotype, and thus to construct the initial network. It can help to enhance the quality of the initial network. In order to speed up the calculation, we convert the genotypic data into binary Boolean data and then directly carry out the fast logic (bitwise) operation to calculate the mutual information. The simulated and real datasets are used to validate the proposed *Epi-GTBN*, and we compare it with other recent algorithms. Experiment results show *Epi-GTBN* has much better epistasis detection accuracy in the case of not affecting the efficiency.

Results

The *bnlearn* [35] is an *R* package for learning the graphical structure of Bayesian network, estimating their parameters and performing some useful inference (http://www.bnlearn.com/). The source code of this package is open, so we can modify and compile the source code conveniently. This package implements several kinds of constraint-based, score-based and *hybrid* structure learning algorithms, such as *IAMB*, *mmpc*, *hc*, *mmhc*, etc. On the basis of the source code of *bnlearn* package, we have implement the proposed *Epi-GTBN* using *R*, see http://122.205.95.139/Epi-GTBN/.

Experiments on simulated data

The experiment is carried out on the computer with the configuration of Intel(R) Core (TM) i7–4790 CPU@ 3.60GHz 4.00GHz, and 8G memory. GAMETES is a commonly used software for the epistasis data generation [36]. It can generate epistasis simulated data quickly and accurately, and generate two or multi-locus epistasis models by setting different parameters. The parameters that can be set in this software include: number of SNP loci, heritability, minimum allele frequency (MAF) and prevalence. Heritability is a measure of how well differences in genes account for differences in the traits. A heritability close to zero indicates that almost all of the variability in a trait is due to environmental factors, with very little influence from genetic differences. MAF refers to the frequency of unusual alleles in a given population. In the simulation file, the last column is phenotype *Class*, 1 represents case, 0 represents control. It uses 0, 1, 2 to express the genotype data, 0 denotes homozygote common genotype, 1 denotes heterozygous genotype and 2 denotes homozygote rare genotype.

Through setting different values of heritability h^2 and MAF, we use GAMETES to generate different simulated datasets. Each dataset includes 100 files. To evaluate the performance of the epistasis detection algorithm, we use Eq.(1) to calculate the detection accuracy.

$$Accuracy = \frac{Num_{edge}}{100} \tag{1}$$

In the equation, Num_{edge} refers to the number of datasets in which the disease associated SNPs are successfully identified among all 100 datasets generated by the same parameters.

Among existing approaches, *BEAM* [9], *AntEpiSeeker* [27], *SNPRuler* [18], *MDR* [6], *BOOST* [21] are powerful tools for the detection of epistatic interactions. In order to do the validation, we compare our *Epi-GTBN* with these tools on the simulated datasets. In addition, we use the frequently-used Bayesian network learning algorithm of *hill-climbing* (*hc*) to do comparison. In this section, we do the detection accuracy and efficiency comparison regarding 2-locus and 3-locus epistatic interaction with heritability h^2 set to 0.025, 0.05, 0.1, 0.2, 0.3, 0.4 and MAF set to 0.1, 0.2, 0.3, 0.4.

Generally, population size is set to 50–100 and crossover probability is set about 0.6 [37]. Mutation probability is generally set as 0.005–0.01 [38]. The tabu list length can’t be set too long or too short, such as more than or far less than the population size. The GA algorithm always converges within 60 iterations in our previous experiment. In our *Epi-GTBN*, we set the population size to 50, set crossover probability to 0.7, set mutation probability to 0.002, set the length of tabu list in the crossover operation to 30, set the maximum number of iterations to 60, set the generations of *k* to 3 when fitness value of the optimal individual and the population no longer increases.

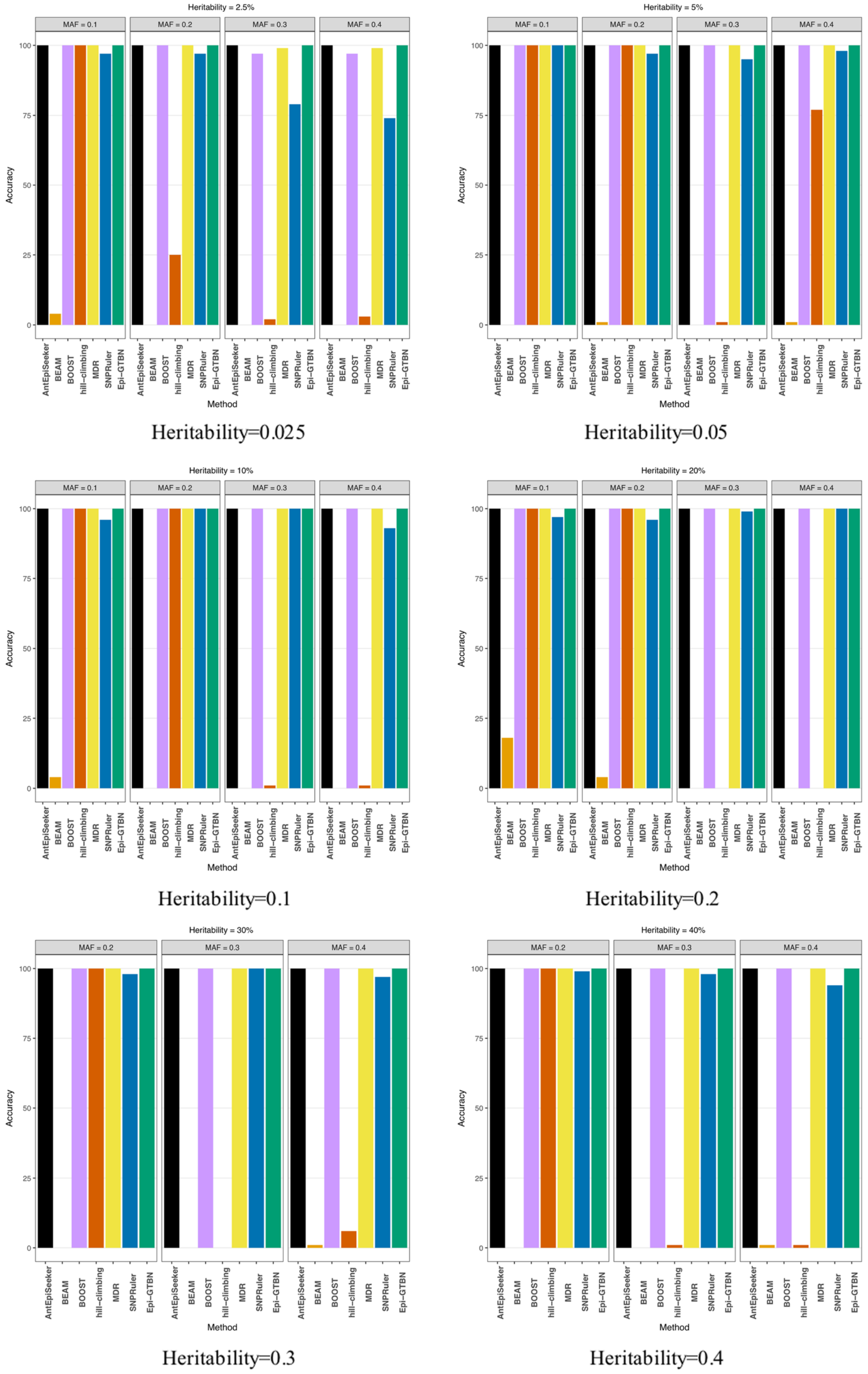
There are no parameters used in *BEAM*, *MDR* and *BOOST*. In *SNPRuler*, we set *listSize* to 2000, *depth* to 4 and set *updateRatio* to 0.5. These parameters have no effect on the result in our previous experiments. Population size is the only same parameter in both *AntEpiSeeker* and *Epi-GTBN*. So in *AntEpiSeeker*, we also set population size to 50 to ensure the fairness of the experiment. Similarly, we set the maximum number of iterations to 60 in *hill-climbing*.

Experiment of 2-locus epistasis detection

In this experiment, we compare the detection accuracy of 2-locus epistasis mining in the case of setting different heritability and MAF. Figure 1 and Fig. 2 show the detection accuracy and efficiency comparison of different methods (*AntEpiSeeker*, *BEAM*, *BOOST*, *hill-climbing*, *MDR*, *SNPRuler* and *Epi-GTBN*).

In Fig. 1, in the case of setting different heritability and MAF, we can see the 2-locus epistasis detection accuracy of *BEAM* and *hill-climbing* (*hc*) Bayesian network learning method is far less than other 4 kinds of methods. The accuracy of *Epi-GTBN*, *MDR*, *BOOST* and *AntEpiSeeker* is the largest of all, mostly hold steady at 100%. The accuracy of *SNPRuler* is slightly less than the above 4 kinds of approaches.

In Fig. 2, we can see the epistasis detection time of *hill-climbing* is the most of all, and it is far larger than the other 6 kinds of methods. The detection time of *BEAM*, *BOOST* and *SNPRuler* is the least of all, and the using time of *AntEpiSeeker*, *MDR* and *Epi-GTBN* is in



Heritability=0.025

Heritability=0.05

Heritability=0.1

Heritability=0.2

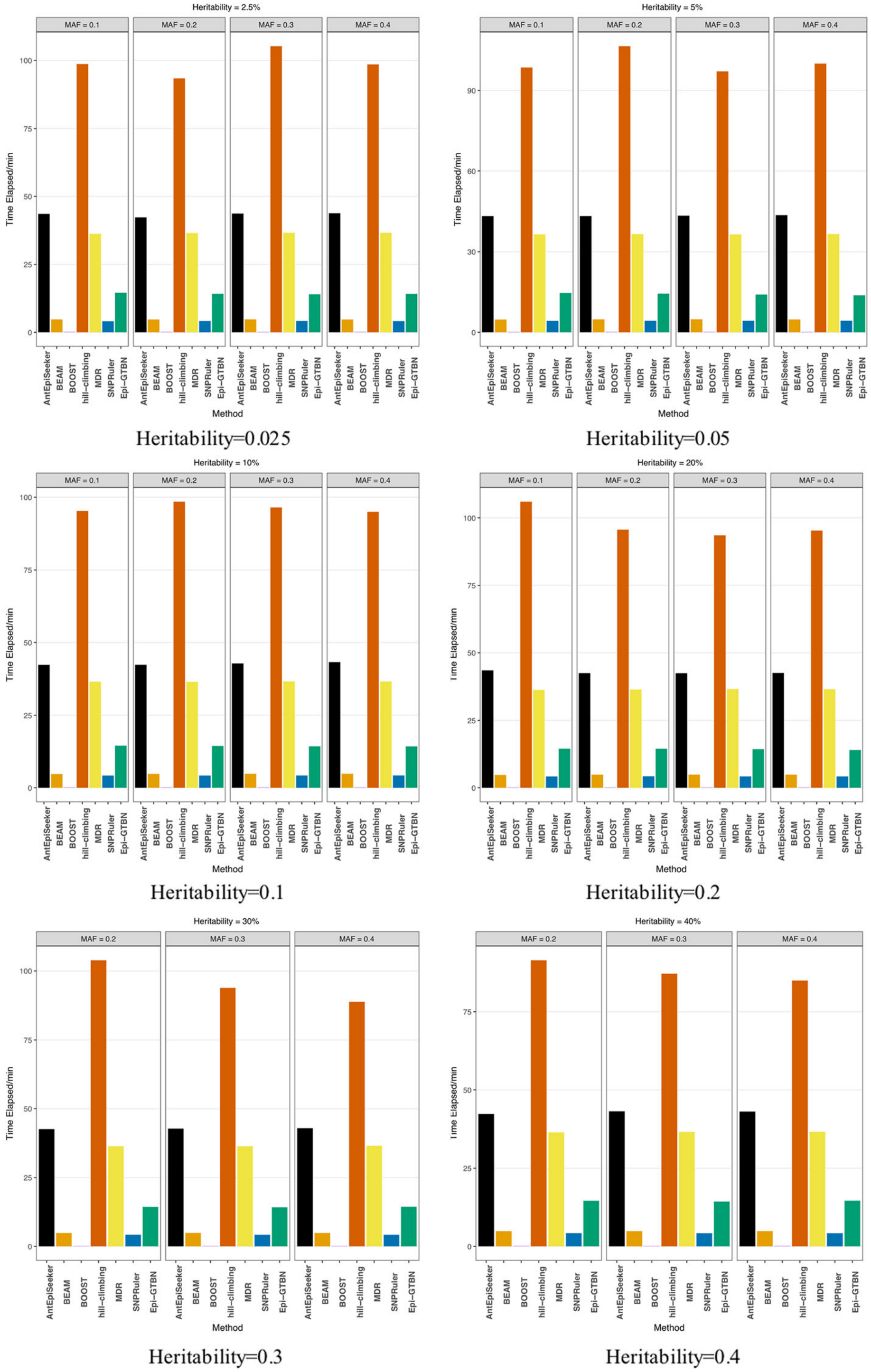
Heritability=0.3

Heritability=0.4

Fig. 1 2 locus epistasis detection accuracy comparison of different methods

the middle. The detection time of *Epi-GTBN* is less than *AntEpiSeeker* and *MDR*. In our *Epi-GTBN* approach, we convert the genotypic data into binary Boolean data, and use the fast logic (bitwise) operation directly to calculate the mutual information. This can save a lot of time of calculating the mutual information entropy between any two SNPs and *Class* when to construct the initial network.

In all, the detection accuracy of *MDR*, *BOOST* and *AntEpiSeeker* is same as our *Epi-GTBN* method, mostly



Heritability=0.025

Heritability=0.05

Heritability=0.1

Heritability=0.2

Heritability=0.3

Heritability=0.4

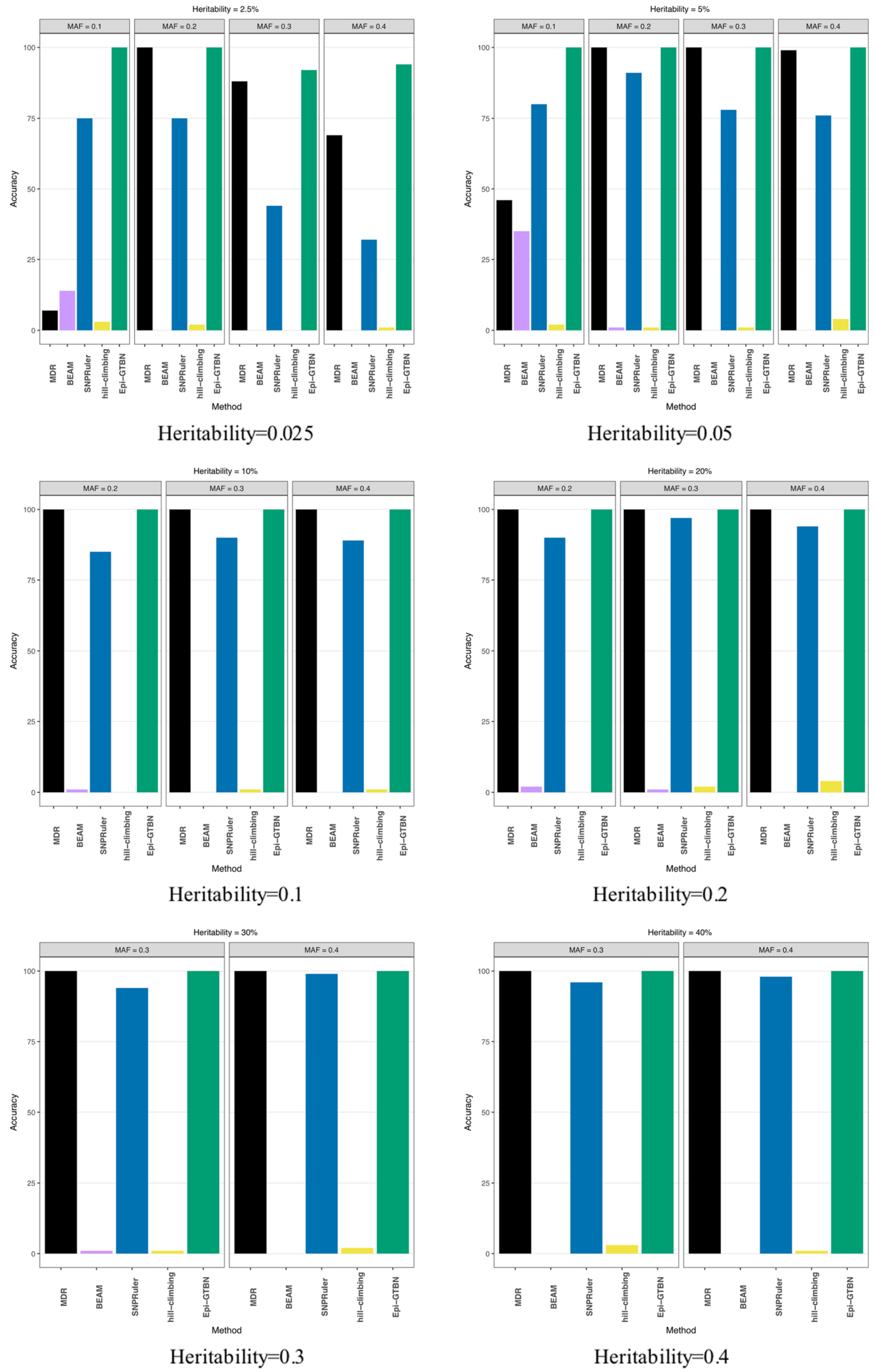
Fig. 2 2 locus epistasis detection efficiency comparison of different methods

hold steady at 100%. But the detection efficiency of *AntEpiSeeker* and *MDR* is lower than *Epi-GTBN* apparently. In addition, the parameter setting of *AntEpiSeeker* is complicated, and its result is related to parameters setting. *BOOST* can only detect the 2-locus epistasis, and it can't be used for the multi-locus epistasis detection. From the experiment results, we can see the epistatic detection approach based on genetic tabu algorithm and Bayesian network (*Epi-GTBN*) has much better detection accuracy in the case of not affecting the efficiency.

Experiment of 3-locus epistasis detection

In this experiment, we compare the accuracy of 3-locus epistasis mining in the case of setting different heritability and MAF. Figure 3 illustrates the detection accuracy comparison of different methods (*MDR*, *BEAM*, *SNPRuler*, *hill-climbing* and *Epi-GTBN*).

The 3-locus epistasis detection accuracy shown in Fig. 3 is almost the same as the case of 2-locus epistasis detection illustrated in Fig. 1. The detection accuracy of



Heritability=0.025

Heritability=0.05

Heritability=0.1

Heritability=0.2

Heritability=0.3

Heritability=0.4

Fig. 3 3 locus epistasis learning accuracy comparison of different methods

BEAM and *hill-climbing* (*hc*) method is the least of all. The accuracy of *Epi-GTBN* and *MDR* is the largest of all, at around 100%, which is slightly more than *SNPRuler*.

Experiments on real AMD data

In order to prove the effectiveness of *Epi-GTBN*, we also use the real age-related macular degeneration (AMD) dataset, which contains 103,611 SNPs genotyped with 96 cases and 50 controls [39]. AMD refers to pathological changes in the central area of the retina, and it is the most important cause of irreversible visual loss in elderly populations. AMD is considered as a complex disease whereby multiple SNP-SNP interactions interact with environmental factors to it. The AMD dataset has been widely used as a benchmark dataset to validate the epistasis mining approaches [13, 26, 29, 40–44].

After the filtration, there are 102,926 autosomal SNPs remained. Firstly, SNP loci with *p*-values from Chi square test less than 0.01 are removed from AMD dataset. Subsequently, 1039 SNP loci remain in the AMD dataset. Here, we explore 2-SNP AMD-associated genetic factors to validate the effectiveness of *Epi-GTBN*. We calculate the conditional mutual information between any two SNPs and phenotype firstly. Then we sort the SNP pairs and extract the SNP pairs whose mutual information is larger than 0.16. These SNP pairs are used to construct the initial network, and then *Epi-GTBN* is used to learn the Bayesian network of SNP loci and phenotype. There are 171 SNP-pairs in the final output set (see epistatic interactions of AMD.xlsx in Additional file 1). Table 1 shows the Top-10 epistatic interactions that *Epi-GTBN* have detected, which we compared with the results of other four methods that showed promising results when experimenting with simulated datasets (*AntEpiSeeker*, *MDR*, *BOOST*, *SNPRuler*). In the table, the column of *MI* means the calculated mutual information I ($Class$ | SNP_1, SNP_2) of SNP_1 and SNP_2 in the initial network. If a SNP-pair detected by *Epi-GTBN* was also detected by other method, it will be marked in the table. And if that SNP-pair also happens to be one of the top gene-gene interactions detected by other methods, additional information will be added into related cell in parentheses. We have put the full result of Top-10 epistatic interactions captured by *AntEpiSeeker*, *SNPRuler*, *BOOST*, *MDR* in Table 2 for reference.

As we know, how to evaluate the detection accuracy of the algorithms with real data is more difficult compared with the simulated data. This is due to the precise identification of all epistasis for the real data is not known. Therefore, we validate our method by searching for literature support in this work. In Table 1, we can see the Top-10 epistatic interactions detected using *Epi-GTBN* have strong literature support. The interaction (rs380390, rs1363688) and (rs380390, rs2402053) are the most statistically significant two among all detected SNP-SNP interactions, and it has also been reported by many literatures [26, 29, 40, 41, 44]. Similarly, the SNP-SNP interactions (rs380390, rs10512174), (rs380390, rs718263), (rs1329428, rs9328536), (rs1329428, rs7467596), (rs380390, rs724972) have been reported by many literatures. Additionally, besides Top-10 SNP-pairs, seven other SNP-pairs detected by *Epi-GTBN*: (rs380390, rs10512937), (rs380390, rs10483314), (rs380390, rs10507949), (rs1394608, rs3743175), (rs1394608, rs2828155), (rs1329428, rs3775652), (rs3775652, rs725518) received literature supports, as shown in Table 3. All these 171 SNP-pairs are displayed in Fig. 4.

In Fig. 4, a node denotes a SNP locus. Two linked nodes represent one SNP-pair of final 171 SNP-pairs. The larger of

Table 1 Top-10 epistatic interactions associated with AMD captured by Epi-GTBN compare with other methods

ID	SNP 1	SNP 2	MI	References	AntEpiSeeker	MDR	BOOST	SNPRuler
1	rs380390	rs1363688	0.205025859	Sun et al. 2017, Shang et al. 2014, Tuo et al. 2016, Shang et al. 2015	–	✓ (11)	–	–
2	rs380390	rs2402053	0.204420493	Sun et al. 2017, Tuo et al. 2016, Shang et al. 2015 Han et al. 2012	–	–	–	–
3	rs380390	rs10512174	0.192477486	Sun et al. 2017, Shang et al. 2015	–	–	–	–
4	rs380390	rs718263	0.192360092	Sun et al. 2017, Shang et al. 2015	–	–	–	–
5	rs1329428	rs9328536	0.190001652	Sun et al. 2017, Kwon et al. 2014, Tuo et al. 2016	✓ (top-10)	–	–	–
6	rs1329428	rs7467596	0.190001652	Tuo et al. 2016	–	–	–	–
7	rs10503216	rs9316435	0.188192429	–	✓ (top-10)	✓	✓	–
8	rs380390	rs335368	0.184951682	–	–	–	–	–
9	rs380390	rs555174	0.184735375	–	–	✓ (top-10)	–	–
10	rs380390	rs724972	0.183950563	Tuo et al. 2016	–	✓ (top-10)	–	–

Table 2 Top-10 epistatic interactions associated with AMD captured by AntEpiSeeker, SNPRuler, BOOST, MDR

ID/ Methods	AntEpiSeeker		SNPRuler		BOOST		MDR	
	SNP1	SNP2	SNP1	SNP2	SNP1	SNP2	SNP1	SNP2
1	rs1329428	rs9328536	rs10503790	rs6928748	rs9316435	rs10503216	rs555174	rs380390
References	Sun et al. 2017, Kwon et al. 2014, Tuo et al. 2016		–		–		–	
2	rs4880042	rs718309	rs657618	rs7908635	–	–	rs10507949	rs10511467
References	–		–		–		–	
3	rs9316435	rs10503216	rs10512781	rs10510099	–	–	rs1293449	rs380390
References	–		–		–		–	
4	rs10505112	rs10512174	rs215389	rs903645	–	–	rs961360	rs380390
References	–		–		–		–	
5	rs1359634	rs1740752	rs4526387	rs2105250	–	–	rs10511467	rs1394608
References	–		–		–		–	
6	rs1535891	rs6598991	rs485412	rs10497257	–	–	rs724972	rs380390
References	–		–		–		Tuo et al. 2016	
7	rs9294603	rs6540592	rs1677189	rs4947673	–	–	rs261796	rs380390
References	–		–		–		–	
8	rs943653	rs4128956	rs3829918	rs727200	–	–	rs1510134	rs380390
References	–		–		–		–	
9	rs1233255	rs860309	rs7533063	rs10484087	–	–	rs1742923	rs380390
References	–		–		–		–	
10	rs404199	rs10510895	rs1489402	rs10484087	–	–	rs1146382	rs380390
References	–		–		–		–	

the node, the more nodes linked with it. It can be seen evidently from Fig. 4 that three SNPs ‘rs380390’, ‘rs1329428’ and ‘rs10254116’ are associated with more other SNPs. This finding is consistent with the work that firstly introduced about the AMD dataset [39], in which the authors reported that two SNPs, ‘rs380390’ and ‘rs1329428’, were believed to be particularly associated with AMD. We hope that, from these results, some clues could be provided for the exploration of causative factors of AMD.

Conclusion

GWAS focuses on single locus and main effect gene locus detection. Although a lot of phenotype related loci can be found using this method, these loci can only explain very few genetic variations. One of the important reasons is that it does not consider the gene-gene interactions, namely epistasis. Therefore, the detection of epistasis is of great significance to the discovery, diagnosis, treatment and prevention of complex diseases. In this study, we propose an epistasis detection approach called *Epi-GTBN*. The genetic tabu approach is used into the search strategy of Bayesian network, and BIC scoring is used to calculate the fitness function value. *Epi-GTBN* can learn the optimal network structure quickly and accurately, and then used to mine epistatic loci. *Epi-GTBN* has 3 major features: *i*) The mutual information entropy is used in the generation of initial

Table 3 Other epistatic interactions associated with AMD captured by Epi-GTBN with literature support

ID	SNP 1	SNP 2	MI	References
1	rs380390	rs10507949	0.183066189	Shang et al. 2015
2	rs380390	rs10512937	0.176409436	Tuo et al. 2016
3	rs380390	rs10483314	0.172425422	Tuo et al. 2016
4	rs3775652	rs725518	0.170306079	Tuo et al. 2016
5	rs1329428	rs3775652	0.168639751	Tuo et al. 2016
6	rs1394608	rs3743175	0.162643832	Tang et al. 2009, Jiang et al. 2009
7	rs1394608	rs2828155	0.162643832	Tang et al. 2009, Jiang et al. 2009

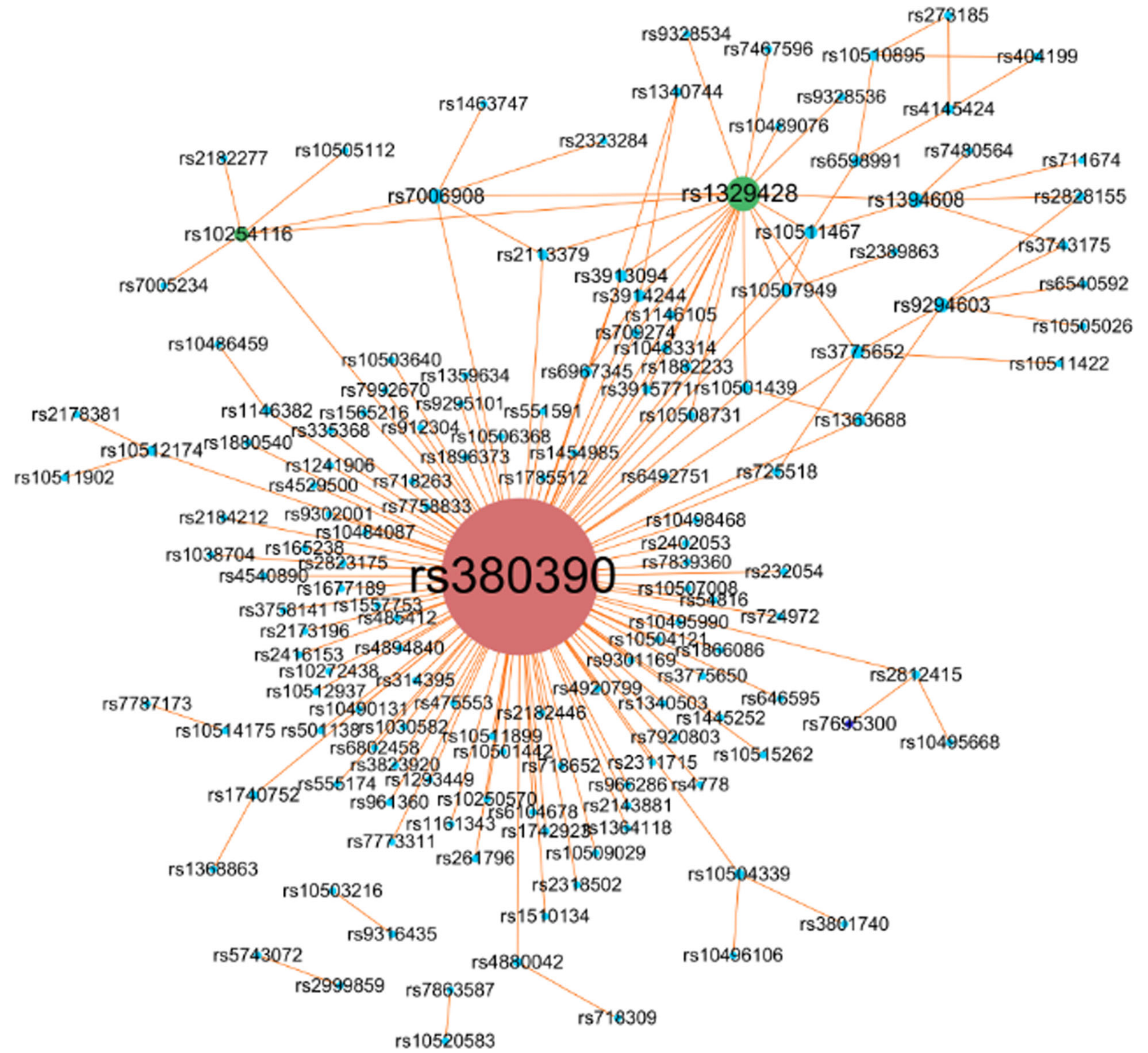


Fig. 4 SNP-SNP network of AMD

individuals to improve the quality of the initial population, which helps to improve the accuracy of epistasis detection. In order to speed up the calculation, it converts the genotypic data into binary Boolean data, and carries out the fast logic (bitwise) operation directly to calculate the mutual information. *ii*) It combines genetic algorithm with Bayesian network to construct the network of gene loci for specific phenotype. The operations of selection, crossover, mutation are used into the search strategy of Bayesian network, thus to evolve the individual structure to achieve the global optimization. *iii*) The tabu search strategy is applied to the operations of crossover and mutation of genetic algorithm, which enhances the diversity of population, and helps to get the global optimal solution and accelerate the convergence of the algorithm. Experimental results in simulated and real datasets elucidate that *Epi-GTBN* is an effective method for epistasis detection. Compared with other existing approaches, *Epi-GTBN* can detect epistatic loci accurately in the case of guaranteeing efficiency.

The next research work mainly includes the following aspects: the time used in the algorithm increases exponentially when to detect multi-locus epistasis. It needs to combine with other algorithms to perform optimization.

In addition, prior knowledge should be borrowed to accelerate the process of search.

Methods

Bayesian network

Bayesian network (BN) is a graphical model used to represent the probability distribution among variables. A Bayesian network consists of a directed acyclic graph and a series of conditional probability tables. The directed acyclic graph is used to express the conditional dependence relationships. The conditional probability distribution is used to parameterize the nodes. BN provides a way of expressing causality between variables.

Supposing $X = \{X_1, X_2, \ldots, X_n\}$, $BN = \{G, P\}$. $G = \{V, E\}$, G is a directed acyclic graph of X. V represents the node set in G and each node represents a variable in X. E represents the edge set in G. Each directed edge represents the conditional dependence relationship between the corresponding nodes. If there exists a directed edge from X_j to X_i, we call X_j the parent node of X_i, and X_i the sub node of X_j. P represents the conditional probability set of BN. $P = \{P(X_i \mid pa(X_i))\}$, $P(X_i \mid pa(X_i))$ denotes the conditional probability of X_i and pa (X_i) denotes the parent node set of X_i. The full probability distribution of Bayesian network is shown in Eq. (2)

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | pa(X_i)) \tag{2}$$

Bayesian network structure learning intends to find the optimal network to match the specific dataset. There are mainly two kinds of Bayesian network structure learning methods: the score-based structure learning methods and constraint-based structure learning methods. The score-based structure learning method firstly defines the scoring function, then it uses specific search strategy to find the network structure with the highest score. This method aims to find a graph with the highest fitting degree of the sample data. Due to the huge search space, it often needs a good search strategy to speed up the search process.

Genetic Tabu algorithm

According to the principle of natural selection, the genetic algorithm (GA) selects the chromosomes that are more suitable for the environment to reproduce. Then it produces a new generation of chromosomes that are more suitable for the environment through the process of crossover and mutation. In this way, it evolves generation by generation, and finally converges to one of the most adaptable chromosomes, so as to find the optimal solution. The genetic algorithm mainly uses three kinds of operations (selection, crossover, mutation) to evolve the population structure, and thus to search for the optimal solution. However, in the searching process of genetic algorithm, it is easy to generate same individual, which affects the diversity of the population. The genetic algorithm selects the better individual and inherits it directly, which is easy to produce local optimal solution, and not conducive to global search.

Tabu search is a famous heuristic search algorithm, which uses the memory function of tabu list to avoid generating some identical individuals, thereby increasing the diversity of population. In addition, tabu search method can accept the inferior solution in the search process, and thus has a stronger climbing ability. This enables tabu search to jump out of the local optimal solution and search for other regions in the search process, thus greatly increase the probability of obtaining better or global optimal solutions. In order to solve the above problems of genetic algorithm, we apply the unique memory function of tabu search into the operations in genetic algorithm inspired from [33, 34]. In [34], solution attributes in tabu list are used for the adjustment of mutation probability in genetic algorithm. In our work, we use the tabu search strategy into the improvement of the crossover operator and mutation operator. This method can be used to improve the performance of the algorithm and find the optimal network structure quickly and accurately.

Epi-GTBN approach

In this work, we construct the network of gene locus for specific phenotype using Bayesian network, and thus to mine the epistasis interactions. The genetic tabu algorithm is used into search strategy of Bayesian network. This approach mainly includes the following steps: initial network construction, initial network population generation, genetic manipulations (selection, crossover, mutation) of the network, getting epistasis interactions.

Network coding

In *Epi-GTBN*, each individual in the genetic algorithm corresponds to a Bayesian network structure, and it does the search in the space of Bayesian network structures. We use the adjacency matrix to represent the Bayesian network structure. Supposing the number of variables is n, and each individual can be represented as an adjacency matrix C of $n \times n$. We use 0/1 coding approach to represent the matrix. If node i is the parent node of node j, then $C_{ij} = 1$, otherwise, $C_{ij} = 0$, as depicted in Fig. 5.

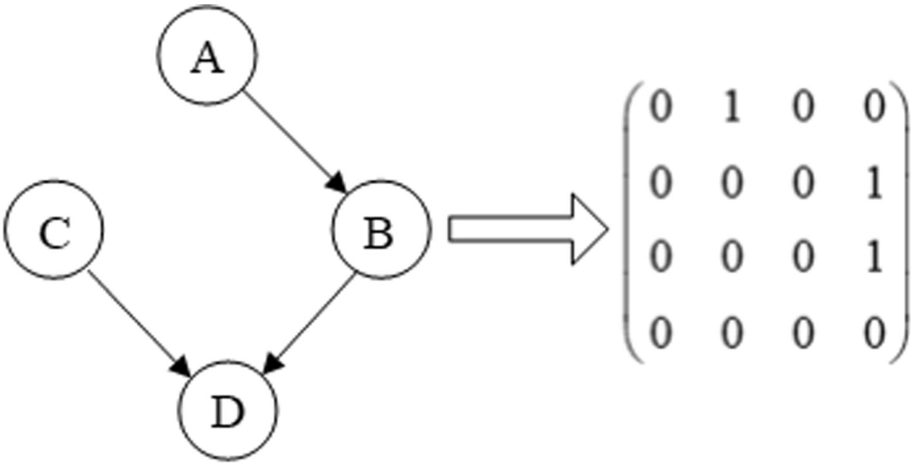


Fig. 5 Matrix coding of Bayesian network

Initial network population generation

In this work, the initial population refers to a set of different Bayesian network individual. In BN, each node denotes the SNP locus or phenotype, and each edge denotes the association between the nodes of SNP locus or phenotype. The quality of initial population has an important impact on the subsequent network structure learning. We calculate mutual information to express the association between multiple gene locus and phenotype [14], and thus to construct the initial network. In further to enhance the calculation efficiency, we convert the genotypic data into binary Boolean data firstly. Then we can use fast logic (bitwise) operation to calculate the mutual information directly. The concrete process of generating initial population is illustrated in Fig. 6.

In Fig. 6, it firstly converts the genotype data into binary Boolean data, and calculates the mutual information between multiple nodes and phenotype for all the possible combinations. Then it sorts the node pairs according to the value of mutual information and extracts the top-N node pairs. The top-N is determined according to the experiment results. For different datasets, we can extract different number of top-N node pairs. The top-N node pairs may not cover all the nodes, and it means there are nodes that are not included in the top-N node pairs. Then we select the first appearance of these nodes in the remaining node pairs, and also extract the corresponding node pairs. Finally, it constructs the initial network according to the node pairs as the initial individual. It generates next individual through adding an edge, dropping an edge or reversing an edge on the premise of not generating a ring. A new individual is generated on the basis of the next individual, until the number of individuals reaches the population size.

In the first step, we convert the genotype data into binary Boolean format. For example, the genotype data depicted in Fig. 7 is converted into the data format shown in Fig. 8.

In Fig. 7, each column denotes the genotype data of each SNP. In the last column *Class*, 1 denotes the case phenotype and 0 denotes the control phenotype. We can see there are four samples in Fig. 7. In Fig. 8, the first/middle/last four columns denote the binary Boolean expression when the genotype data is 0/1/2 respectively.

We use Eq.(3) to calculate the mutual information between the k epistatic SNP loci and *Class* [14]. In Eq.(3), we use Eq.(4) to calculate the information entropy of *Class*, and use Eq.(5) to calculate the joint entropy of k SNP locus.

$$\begin{aligned} I(Class|SNP_1,\ldots SNP_k) = {} & H(Class) + H(SNP_1,\ldots SNP_k) \\ & - H(Class, SNP_1,\ldots SNP_k) \end{aligned} \tag{3}$$

$$H(Class) = -\sum_{ct_i=0}^{1} p\left(class^i\right)\,\log p\left(class^i\right) \tag{4}$$

$$\begin{aligned} H(SNP_1,\ldots,SNP_k) = -\sum_{snp_1=0}^{2}\ldots\sum_{snp_k=0}^{2} p(snp_1,\ldots,snp_k) \\ \log p(snp_1,\ldots,snp_k) \end{aligned} \tag{5}$$

On the basis of binary expression of genotype data, we can conduct the logic **AND** operation to calculate the mutual information efficiently. For example, we use Eq.(6) to calculate $I(Class\ |SNP_B, SNP_C)$ shown in M_{bit}.

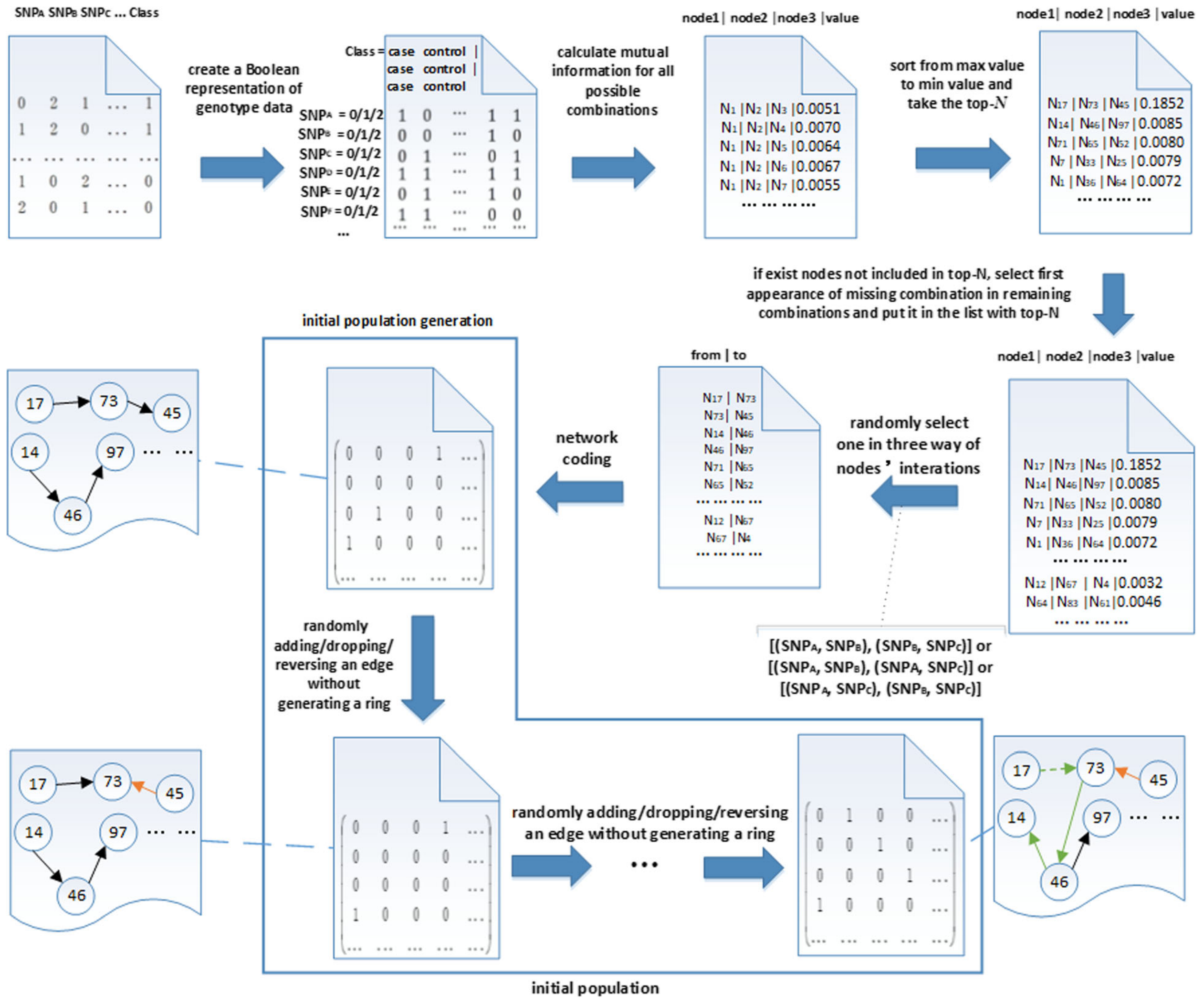

SNPA SNPB SNPC ... Class
0 2 1 ... 1
1 2 0 ... 1
... ...
1 0 2 ... 0
2 0 1 ... 0
create a Boolean representation of genotype data
Class = case control | case control | case control
SNPA = 0/1/2
SNPB = 0/1/2
SNPC = 0/1/2
SNPD = 0/1/2
SNPE = 0/1/2
SNPF = 0/1/2
...
calculate mutual information for all possible combinations
node1 | node2 | node3 | value
N1 | N2 | N3 | 0.0051
N1 | N2 | N4 | 0.0070
N1 | N2 | N5 | 0.0064
N1 | N2 | N6 | 0.0067
N1 | N2 | N7 | 0.0055
sort from max value to min value and take the top-N
node1 | node2 | node3 | value
N17 | N73 | N45 | 0.1852
N14 | N46 | N97 | 0.0085
N71 | N65 | N52 | 0.0080
N7 | N33 | N25 | 0.0079
N1 | N36 | N64 | 0.0072
if exist nodes not included in top-N, select first appearance of missing combination in remaining combinations and put it in the list with top-N
node1 | node2 | node3 | value
N17 | N73 | N45 | 0.1852
N14 | N46 | N97 | 0.0085
N71 | N65 | N52 | 0.0080
N7 | N33 | N25 | 0.0079
N1 | N36 | N64 | 0.0072
N12 | N67 | N4 | 0.0032
N64 | N83 | N61 | 0.0046
randomly select one in three way of nodes' interations
[(SNPA, SNPB), (SNPB, SNPC)] or
[(SNPA, SNPB), (SNPA, SNPC)] or
[(SNPA, SNPC), (SNPB, SNPC)]
from | to
N17 | N73
N73 | N45
N14 | N46
N46 | N97
N71 | N65
N65 | N52
N12 | N67
N67 | N4
network coding
initial population generation
17
73
45
14
97
46
randomly adding/dropping/reversing an edge without generating a ring
randomly adding/dropping/reversing an edge without generating a ring
initial population

Fig. 6 The process of initial population generation

$$
\begin{array}{cc}
 & \begin{array}{cccccccccccc} SNP_A & SNP_B & SNP_C & SNP_D & SNP_E & SNP_F & SNP_G & SNP_H & SNP_I & SNP_J & SNP_K & Class \end{array} \\
M = & \begin{bmatrix} \boxed{0} & 1 & 1 & 1 & 1 & 0 & 1 & 2 & 2 & 1 & 0 & 1 \\ \boxed{0} & 1 & 0 & 2 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 2 & 2 & 0 & 0 & 0 & 1 & 1 & 2 & 1 & 1 & 0 \\ 2 & 1 & 1 & 1 & 1 & 2 & 2 & 1 & 0 & 0 & 2 & 0 \end{bmatrix}
\end{array}
$$

Fig. 7 The genotype data

$$M_{bit} = \begin{array}{r|cccccccccccc} Class\ = & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ SNP_A = 0/1/2 & \boxed{1} & \boxed{1} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ SNP_B = 0/1/2 & 0 & 0 & 0 & 0 & \underline{1} & \underline{1} & \underline{0} & \underline{1} & 0 & 0 & 1 & 0 \\ SNP_C = 0/1/2 & 0 & 1 & 0 & 0 & \underline{1} & \underline{0} & \underline{0} & \underline{1} & 0 & 0 & 1 & 0 \\ SNP_D = 0/1/2 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{array}$$

Fig. 8 The binary Boolean expression of genotype data

$$I(Class|SNP_B, SNP_C) = H(Class) + H(SNP_B, SNP_C) - H(Class, SNP_B, SNP_C) \tag{6}$$

We use Eq.(7) to calculate $H(SNP_B, SNP_C)$ in Eq.(6).

$$\begin{aligned} H(SNP_B, SNP_C) &= -\sum_{snp_B=0}^{2}\sum_{snp_C=0}^{2} p(snp_B, snp_C)\ \log p(snp_B, snp_C) \\ &= -p(0,0)\ \log p(0,0) - p(0,1)\ \log p(0,1) \\ &\quad -p(0,2)\ \log p(0,2) - p(1,0)\ \log p(1,0) \\ &\quad -p(1,1)\ \log p(1,1) - p(1,2)\ \log p(1,2) \\ &\quad -p(2,0)\ \log p(2,0) - p(2,1)\ \log p(2,1) \\ &\quad -p(2,2)\ \log p(2,2) \end{aligned} \tag{7}$$

Using the underlined binary data in M_{bit}, we can calculate p (1, 1) using Eq.(8) through the **AND** operation of binary. It can be accomplished by the counting of "1" bits in a bit string (also called hamming weight).

$$\begin{aligned} p(1,1) &= \frac{n_{snp_B=snp_C=1}}{n_{sample}} = \frac{\text{hamming weight}\left[(1101)_2 AND_{bit}(1001)_2\right]}{4} \\ &= \frac{\text{hamming weight}\left[(1001)_2\right]}{4} = \frac{2}{4} = 0.5 \end{aligned} \tag{8}$$

Selection

The purpose of selection is to select a good individual from the current population so that they have a chance to be the offspring of the next generation. The principle of selection is that the individual with greater adaptability will be selected with larger probability, which embodies the survival of the fittest principle. This work mainly uses the roulette selection method. Supposing the fitness value is f_i about chromosome i, then the probability P_i of chromosome i being selected is calculated using Eq.(9). In the equation, num represents the size of population.

$$P_i = f_i / \sum_{i=1}^{num} f_i \tag{9}$$

Crossover

Crossover is the most important operation in the genetic algorithm. It can get better individuals in the new generation through the crossover operation, and the new individuals inherit the characteristics of their parents. The commonly used crossover operations include single column crossover, double or multi column crossover, uniform crossover, etc. The multi column crossover refers to a variation of several columns. In order to speed up the convergence rate, we intend to use the multi column crossover method.

Supposing two individuals of $Individual_1$ and $Individual_2$ in the population, it randomly selects two columns f_1, f_2 of $Individual_1$ and s_1, s_2 of $Individual_2$. The column f_1 of $Individual_1$ is exchanged with column s_1 of $Individual_2$, and column f_2 of $Individual_1$ is exchanged with column s_2 of $Individual_2$. $Individual_1[\ldots f_1 \ldots f_2 \ldots] \overset{\text{crossover}}{\leftrightarrow} Individual_2[\ldots s_1 \ldots s_2 \ldots]$. Then we can get $Individual_1[\ldots s_1 \ldots s_2 \ldots]$ and $Individual_2[\ldots f_1 \ldots f_2 \ldots]$. It will also judge whether the crossover operation will generate a ring or not. When there is no ring structure in both $Individual_1$ and $Individual_2$, they will be considered as new offspring.

Avoid ring generation If it will generate a ring structure when to exchange a row in the randomly chosen column in the process of crossover operation, the algorithm will skip that row. Then it judges the next row until all the two columns are exchanged. In Fig. 9, the crossover operation randomly selects two columns of $Individual_1$ and of $Individual_2$. Take a column pair for example, the second column is chosen in $Individual_1$, and the first column is chosen in $Individual_2$, as illustrated in the red mark of Fig. 9. The crossover operation is executed in the following two steps:

1). Exchange the first row in the chosen columns of two individuals. If the first row with the value of 1 in the second column of $Individual_1$ is exchanged with the first row with the value of 0 in the first column of $Individual_2$, we can see a ring structure in generated in $Individual_2$, as illustrated in the red mark of Fig. 9. Then the

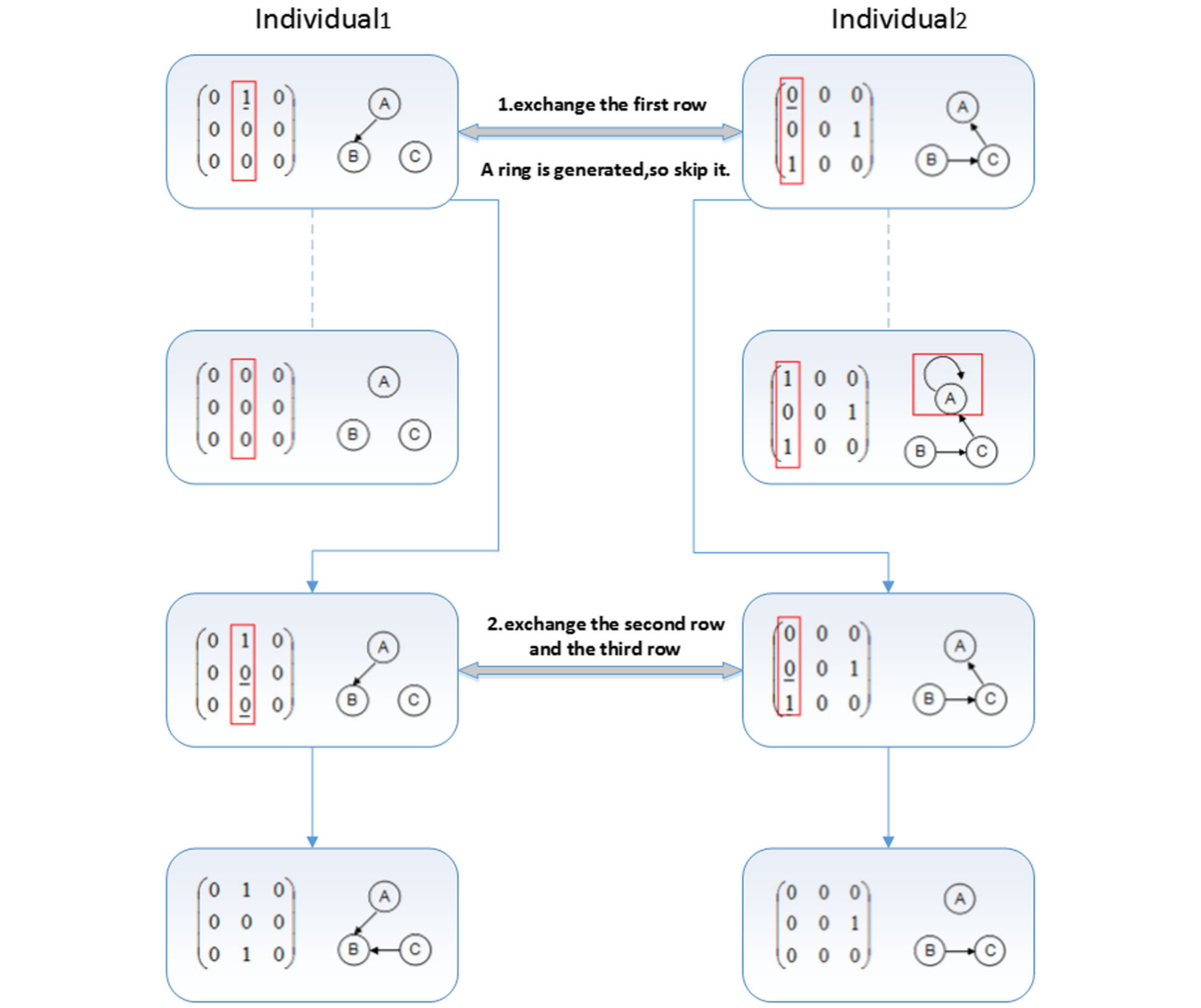


Fig. 9 Process of avoid ring structure generation

crossover operation will skip the first row and do not exchange that row, then do the exchanging from the second row.

2). Exchange the second row and the third row, we can see no ring structure will be generated. The crossover operation will not skip these rows, and the final two individuals can be obtained after the crossover operation.

The general crossover operator In the different offspring of particular population, crossover operation may produce the same offspring. This will cause the partial similarity of chromosomes in the population, and lead to search stagnant and prone to premature phenomenon. In Fig. 10, $Individual_1$ and $Individual_2$ are randomly chosen to do the crossover operation in iteration 1. Then we get two new offspring shown in iteration 2. In iteration 2, $Individual_2$ and $Individual_n$ are randomly chosen to do the crossover operation. In this way, two offspring are identical with the parents, as illustrated in the red mark. This crossover operation does not produce new offspring.

The crossover tabu operator In order to solve the search stagnant and premature phenomenon that generated by general crossover operators, we use the memory function of tabu search method into the crossover operation of genetic algorithm. We compare the generated new offspring individual with the individuals in the tabu list one by one after each crossover operation, as illustrated in Fig. 6. If the new offspring individual does not belong to the tabu list, the algorithm will enter into the next generation and the new individual will be stored into the tabu list. The crossover operation will be carried out repeatly until the offspring are not belonging to the tabu list,

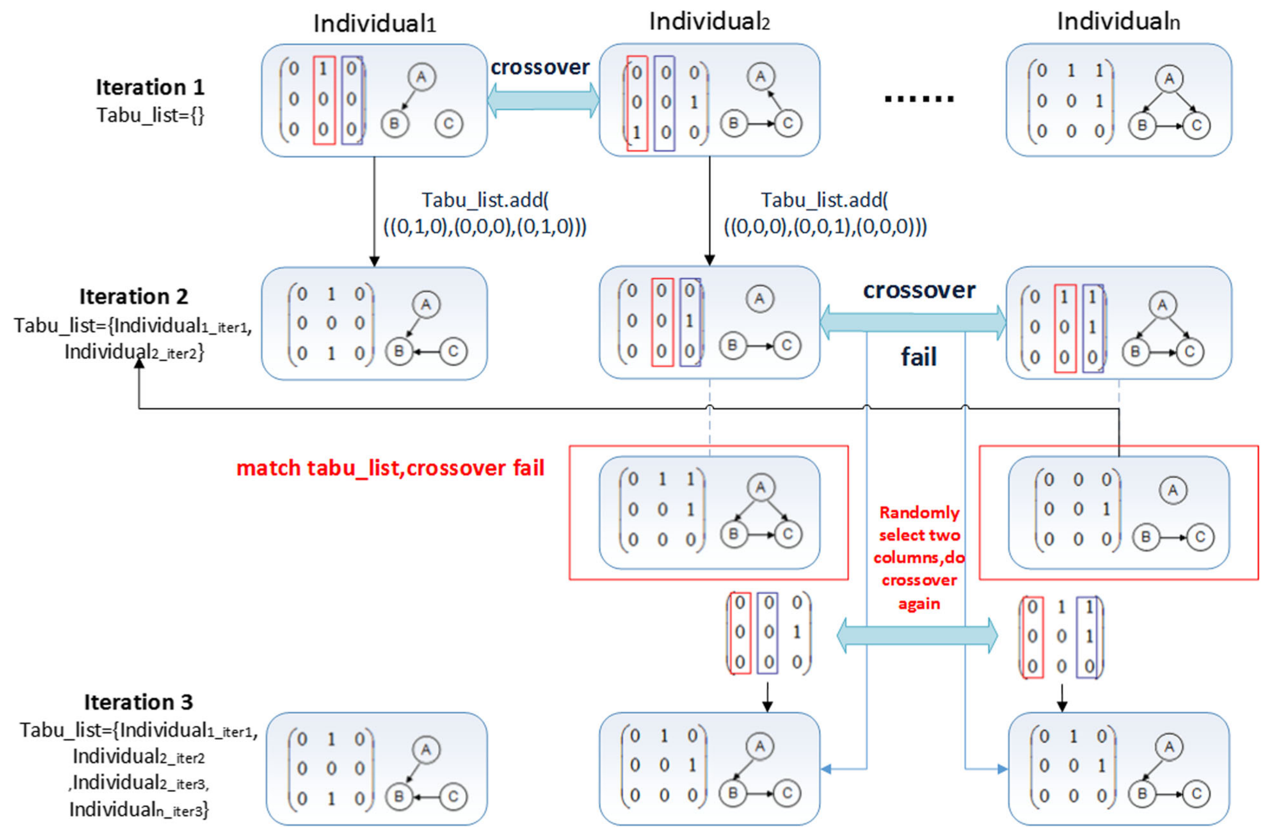


Fig. 10 Process of crossover tabu operator

as illustrated in Fig. 10. The detailed procedure is elucidated in Algorithm 1.

```
Algorithm 1. Algorithm of crossover tabu operation
Input: $Par\_Individual_1$, $Par\_Individual_2$, tabu_list, nnodes
Output: The coding of two new individuals $amat_1$, $amat_2$
1: Randomly select two columns and do crossover operation of $Par\_Individual_1$,
   $Par\_Individual_2$, get the coding $amat_1$, $amat_2$ respectively
2: for $Individual_i$ ($i$ =0, 1) do
3:     Edge number cnt=0
4:     Initialing the hash array
5:     for (k=0; k <nnodes*nnodes; k++)
6:        if ($amat_i[k] > 0$)
7:            hash[cnt++]=k
8:     end for
9:     Invoke tabu_match and check whether the new individual is in tabu_list or not
10:    if($Individual_i \in tabu\_list$)
11:        go to 1
12:    else
13:        Store hash array into tabu_list
14: end for
15: return $amat_1$, $amat_2$
```

In Algorithm 1, $Par\_Individual_1$, $Par\_Individual_2$ are two parent individuals. *tabu_list* denotes the tabu table, and *nnodes* denotes the node number. In the output, $amat_1$ and $amat_2$ represent the coding of two new individuals.

Mutation

The mutation operator first selects an individual in the population randomly. The selected individual randomly changes the structure with a certain mutation probability P_m. It is beneficial to increase the diversity of the population. The algorithm uses tabu mutation operator to select the mutation that has better variation fitness value, and it also ensures that a ring is not generated.

The general mutation operator If the random probability is less than the mutation probability, it randomly selects a locus to perform mutation, as shown in Fig. 11. If the individual fitness is improved after the mutation and no ring structure is generated, then accept this individual and store it into the population.

The tabu mutation operator The general mutation operator has strong randomness and may damage individuals with high fitness value. In order to solve this problem, we use the memory function of tabu list and propose a tabu mutation operator. This operator invokes evaluation function to determine the operation strategy. The new generated individual will be stored into the tabu list when the variation produces an inferior solution and improves the fitness value. The tabu mutation operator can avoid roundabout searches, and its climbing ability is better than the

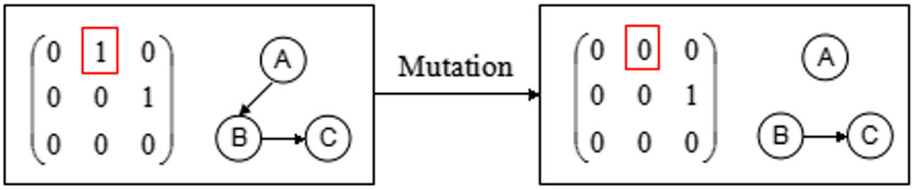


Fig. 11 The general mutation operator

general mutation operator. The concrete process is illustrated in Algorithm 2.

```
Algorithm 2. Algorithm of tabu mutation operation
Input: Individual, num, tabu_list
Output: $Individual_{mu}$
for $Individual_i$ ($i$=0,1,...,num) do
    store $Individual_i$ into tabu_list
end for
for $Individual_i$ ($i$=0,1,...,num) do
    for $edge_j$ in $Individual_i$
        if operation of $edge_j$ to $Individual_i$ can improve the fitness
            change $Individual_i$ to $Individual_{i2}$
            if $Individual_{i2}$ is acycilc
                Invoke tabu_match and judge whether $Individual_{i2}$ is in tabu_list or not
                    if($Individual_{i2} \in tabu\_list$)
                        continue
                else
                        Store $Individual_{i2}$ into tabu_list
                        $Individual_{mu\_i}$= $Individual_{i2}$
                        break
            else
                continue
        else
            continue
    end if
    $Individual_{mu\_i}$=$Individual_i$
  end for
end for
$Individual_{mu}$=$Individual_{mu\_1} \cup Individual_{mu\_2}, \ldots, \cup Individual_{mu\_num}$
return $Individual_{mu}$
```

In Algorithm 2, *Individual* represents the current population, *num* represents the size of current population and *tabu_list* denotes the tabu table. $Individual_{mu}$ represents the new individual generated by tabu mutation operation.

Fitness value evaluation

The fitness function is a standard to judge the quality of an individual or a solution. It determines which outstanding individuals are retained and which poorer individuals are eliminated. The genetic algorithm is an evolutionary search mechanism based on the fitness function. In this work, the fitness value evaluates the quality of the Bayesian network, and thus to guide the search strategy. In our method, the fitness value can be calculated using the *BN* scoring function. There are several kinds of BN scoring function, such as Bayesian Information Criterion (*BIC*), Akaike Information Criterion (*AIC*), Equivalent Dirichlet Posterior Density (*BDe*), *K*2, Log-likelihood, etc. To prevent over-fitting, we use the *BIC* scoring function to control the model complexity [45]. In addition, our experiment results demonstrate that the *BIC* scoring function has better learning effect.

In the case of given prior knowledge and sample data, Bayesian Information Criterion (*BIC*) selects the Bayesian network structure with the largest posterior probability. Supposing D represents the sample data, G represents the Bayesian network structure, we can get Eq.(10) using Bayesian formula. In the equation, $P(G)$ represents the priori knowledge of network structure.

$$P(G|D) = P(D|G)P(G)/P(D) \tag{10}$$

Using θ_G to denote the parameters of the network structure, we can get Eq.(11) through the marginal integration scheme.

$$P(D|G) = \int P(D|G,\theta_G)P(\theta_G|G)d\theta_G \tag{11}$$

The *BIC* scoring function is shown in Eq.(12).

$$BIC(G|D) = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} m_{ijk}\ \lg\frac{m_{ijk}}{m_{ij*}} - \sum_{i=1}^{n}\frac{q_i(r_i-1)}{2}\ \lg m \tag{12}$$

In the equation, m represents the total number of samples. n represents the number of variables. r_i represents the number of values for the ith variable. q_i represents the combinations number of the parent about the ith variable. m_{ijk} represents the sample number of ith variable takes the kth value, and its parent nodes take the jth combination.

The end judgement

When the fitness value of an optimal individual reaches a given threshold, achieving the maximum number of iterations, or the fitness value of the optimal individual and the population no longer increases after k generations, then end up the algorithm.

Additional file

Additional file 1: Epistatic interactions of AMD. (XLSX 35 kb)

Abbreviations

AIC: Akaike Information Criterion; AMD: Real age-related macular degeneration; BDe: Equivalent Dirichlet Posterior Density; BEAM: Bayesian epistasis association mapping; BIC: Bayesian Information Criterion; BN: Bayesian network; BOOST: Boolean operation-based screening; Epi-GTBN: Epistasis mining based on genetic tabu algorithm and Bayesian network; GWAS: Genome-wide association study; MAF: Minimum allele frequency; MDR: Multifactor-dimensionality reduction; SNP: Single nucleotide polymorphism

Acknowledgements

Not applicable.

Authors' contributions

YG, ZMZ designed the algorithms and carried out the experiments. CY generated the simulated experiment dataset using GAMETES. YLJ developed the source code of mutual information. JFH, ZZL and HG participated in the source code development of the algorithms. JXL developed the method and drafted the manuscript. All authors have read and approved this manuscript.

Funding

This research is supported by the National Natural Science Foundation of China under Grant No.31601078, the National Training Programs of Innovation and Entrepreneurship for Undergraduates under grant No. 201710504124, the Fundamental Research Funds for the Central Universities under grant No.2662018JC030. The funding bodies did not play any roles in the design of the study and collection, analysis, and interpretation of data and in writing the manuscript.

Availability of data and materials

All data generated or analyzed during this study are included in this published article (and the additional information files). The code for this work is available at http://122.205.95.139/Epi-GTBN/.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 4 October 2018 Accepted: 7 August 2019
Published online: 28 August 2019

References

1. Park MY, Hastie T. Penalized logistic regression for detecting gene interactions. Biostatistics. 2008;9(1):30–50.
2. Stanislas V, Dalmasso C, Ambroise C. Eigen-epistasis for detecting gene-gene interactions. BMC Bioinformatics. 2017;18(1):54.
3. Zhang F, Xie D, Liang M, et al. Functional regression models for epistasis analysis of multiple quantitative traits. PLoS Genet. 2016;12(4):e1005965.
4. De R, Hu T, Moore JH, et al. Characterizing gene-gene interactions in a statistical epistasis network of twelve candidate genes for obesity. Biodata Mining. 2015;8(1):45.
5. Zhang X, Zou F, Wang W. FastANOVA: an efficient algorithm for genome-wide association study. Int Confer Knowl Discov Data Mining. 2008;821.
6. 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(1):138–47.
7. Yu W, Lee S, Park T. A unified model based multifactor dimensionality reduction framework for detecting gene-gene interactions. Bioinformatics. 2016;32(17):i605.
8. Yang CH, Chuang LY, Lin YD. CMDR based differential evolution identify the epistatic interaction in genome-wide association studies. Bioinformatics. 2017;33(15):2354.
9. Zhang Y, Liu JS. Bayesian inference of epistatic interactions in case-control studies. Nat Genet. 2007;39(9):1167–73.
10. Colak R, Kim TH, Kazan H, et al. JBASE: joint Bayesian analysis of subphenotypes and epistasis. Bioinformatics. 2016;32(2):203.
11. Dong CZ, Chu X, Wang Y, et al. Exploration of gene-gene interaction effects using entropy-based methods. Eur J Hum Genet. 2008;16:229–35.
12. Hu T, Chen Y, Kiralis JW, et al. An information-gain approach to detecting three-way epistatic interactions in genetic association studies. J Am Med Inform Assoc Jamia. 2013;20(4):630.
13. Kwon MS, Park M, Park T. IGENT: efficient entropy based algorithm for genome-wide gene-gene interaction analysis. BMC Med Genet. 2014;7(Suppl 1):S6.
14. Li X. A fast and exhaustive method for heterogeneity and epistasis analysis based on multi-objective optimization. Bioinformatics. 2017;33(18):2829–36.
15. Jiang X, Neapolitan RE, Barmada MM, et al. Learning genetic epistasis using Bayesian network scoring criteria. BMC Bioinformatics. 2011;12(1):89.
16. Zeng Z, Jiang X, Richard N. Discovering causal interactions using Bayesian network scoring and information gain. BMC Bioinformatics. 2016;17(1):1–14.
17. Yang C, He ZX, Yang Q, et al. SNPHarvester: a filtering-based approach for detecting epistatic interactions in genome-wide association studies. Bioinformatics. 2009;25(4):504–11.
18. Wan X, Yang C, Yang Q, et al. Predictive rule inference for epistatic interaction detection in genome-wide association studies. Bioinformatics. 2010;26(1):30–7.
19. Jiang X, Neapolitan RE. LEAP: biomarker inference through learning and evaluating association patterns. Genet Epidemiol. 2015;39(3):173.
20. Arkin Y, Rahmani E, Kleber ME, et al. EPIQ-efficient detection of SNP-SNP epistatic interactions for quantitative traits. Bioinformatics. 2014;30(12):i19.
21. Wan 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(3):325.
22. Li J, Malley JD, Andrew AS, et al. Detecting gene-gene interactions using a permutation-based random forest method. Biodata Mining. 2016;9(1):14.
23. Chen SH, Sun J, Dimitrov L, et al. A support vector machine approach for detecting gene-gene interaction. Genet Epidemiol. 2008;32(2):152.
24. Zhang Q, Long Q, Ott J, et al. AprioriGWAS, a new pattern mining strategy for detecting genetic variants associated with disease through interaction effects. PLoS Comput Biol. 2014;10(6):e1003627.
25. Yang CH, Lin YD, Chuang LY, et al. Evaluation of breast cancer susceptibility using improved genetic algorithms to generate genotype SNP barcodes. IEEE/ACM Trans Comput Biol Bioinform. 2013;10(2):361.
26. Shang J, Sun Y, Li S, et al. An improved opposition-based learning particle swarm optimization for the detection of SNP-SNP interactions. Biomed Res Int. 2015;2015:524821.
27. 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(1):117.
28. Jing PJ, Shen HB. MACOED: a multi-objective ant colony optimization algorithm for SNP epistasis detection in genome-wide association studies. Bioinformatics. 2015;31(5):634–41.
29. Sun Y, Shang J, Liu JX, et al. epiACO-a method for identifying epistasis based on ant Colony optimization algorithm. Biodata Mining. 2017;10(1):23.
30. Yuan L, Yuan CA, Huang DS. FAACOSE: a fast adaptive ant colony optimization algorithm for detecting SNP epistasis. Complexity. 2017;2017(1):1–10.
31. Sherif FF, Zayed N, Fakhr M. Discovering Alzheimer genetic biomarkers using Bayesian networks. Adv Bioinforma. 2015;2015:8.
32. Jin Y, Su Y, Zhou XH, et al. Heterogeneous multimodal biomarkers analysis for Alzheimer's disease via Bayesian network. Eurasip J Bioinform Syst Biol. 2016;2016(1):12.
33. Glover F, Kelly JP, Laguna M. Genetic algorithms and tabu search: hybrids for optimization. Comput Oper Res. 1995;22(1):111–34.
34. Shin DJ, Kim JO, Kim TK, et al. Optimal service restoration and reconfiguration of network using genetic-Tabu algorithm. Electr Pow Syst Res. 2004;71(2):145–52.
35. Scutari M. Learning Bayesian networks with the bnlearn R package. J Stat Softw. 2010;35(3):1–22.
36. Urbanowicz RJ, Kiralis J, Sinnott-Armstrong NA, et al. GAMETES: a fast, direct algorithm for generating pure, strict, epistatic models with random architectures. Biodata Mining. 2012;5(1):16.

37. Dejong K. An analysis of the behavior of a class of genetic adaptive systems. Ann Arbor: Ph. D. Thesis, University of Michigan; 1975.
38. Schaffer JD, Caruana R, Eshelman LJ, et al. A study of control parameters affecting online performance of genetic algorithms for function optimization. International Conference on Genetic Algorithms. San Francisco: Morgan Kaufmann Publishers Inc; 1989. p. 51–60.
39. Klein RJ, Zeiss C, Chew EY, et al. Complement factor H polymorphism in age-related macular degeneration. Science. 2005;308(5720):385–9.
40. Shang J, Zhang J, Sun Y, et al. EpiMiner: a three-stage co-information based method for detecting and visualizing epistatic interactions. Digital Signal Process. 2014;24(1):1–13.
41. Tuo SH, Zhang J, Yuan XG, et al. FHSA-SED: two-locus model detection for genome-wide association study with harmony search algorithm. PLoS One. 2016;11(3):e0150669.
42. Jiang R, Tang W, Wu X, et al. A random forest approach to the detection of epistatic interactions in case-control studies. BMC Bioinformatics. 2009;10(s1):0.
43. Tang W, Wu X, Jiang R, et al. Epistatic module detection for case-control studies: a Bayesian model with a Gibbs sampling strategy. PLoS Genet. 2009;5(5):e1000464.
44. Han B, Chen X, Talebizadeh Z, et al. Genetic studies of complex human diseases: characterizing SNP-disease associations using Bayesian networks. BMC Syst Biol. 2012;6(3):1–12.
45. Li R, Dudek SM, Kim D, et al. Identification of genetic interaction networks via an evolutionary algorithm evolved Bayesian network. BioData Mining. 2016;9(1):18.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.