Article

Chinese Science Bulletin

Bioinformatics

August 2013 Vol.58 No.23: 2828–2835
doi: 10.1007/s11434-012-5475-6

Bayesian optimization algorithm-based methods searching for risk/protective factors

WEI Bin[1,2,3,4], PENG QinKe[2,3,4*], CHEN Xiao[2,3,4] & ZHAO Jing[2]

[1] *Engineering University of CAPF, Xi'an 710086, China;*
[2] *Electronic and Information School, Xi'an Jiaotong University, Xi'an 710049, China;*
[3] *Key Laboratory for Intelligent Networks and Network Security of Ministry of Education, Xi'an Jiaotong University, Xi'an 710049, China;*
[4] *State Key Laboratory for Manufacturing Systems Engineering and School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China*

Received January 16, 2012; accepted April 1, 2012; published online May 10, 2013

The risks of developing complex diseases are likely to be determined by single nucleotide polymorphisms (SNPs), which are the most common form of DNA variations. Rapidly developing genotyping technologies have made it possible to assess the influence of SNPs on a particular disease. The aim of this paper is to identify the risk/protective factors of a disease, which are modeled as a subset of SNPs (with specified alleles) with the maximum odds ratio. On the basis of risk/protective factor and the relationship between nucleotides and amino acids, two novel risk/protective factors (called *k*-relaxed risk/protective factors and weighted-relaxed risk/protective factors) are proposed to consider more complex disease-associated SNPs. However, the enormous amount of possible SNPs interactions presents a mathematical and computational challenge. In this paper, we use the Bayesian Optimization Algorithm (BOA) to search for the risk/protective factors of a particular disease. Determining the Bayesian network (BN) structure is *NP*-hard; therefore, the binary particle swarm optimization was used to determine the BN structure. The proposed algorithm was tested on four datasets. Experimental results showed that the algorithm proposed in this paper is a promising method for discovering SNPs interactions that cause/prevent diseases.

risk/protective factors, single nucleotide polymorphisms, Bayesian optimization algorithm, binary particle swarm optimization

Citation: Wei B, Peng Q K, Chen X, et al. Bayesian optimization algorithm-based methods searching for risk/protective factors. Chin Sci Bull, 2013, 58: 2828–2835, doi: 10.1007/s11434-012-5475-6

Searching for genetic factors that influence phenotype, such as a disease, is the major goal of modern geneticists. The risk of developing a complex disease are expected to be heavily influenced by the patterns of single nucleotide polymorphisms (SNPs), which are the most common form of DNA variations [1]. Increasing empirical evidence suggests that interactions among loci contribute widely to complex human diseases. The biological interest is how SNPs interact with each other to influence susceptibility to complex diseases [2–4]. However, most previous studies used the single-SNP analysis strategy, in which each SNP was tested individually for association with a specific disease [5,6]. The number of possible interaction combinations among genotyped markers is astronomical for a large scale case-control association study, making prohibitive to search for one or a very few disease-related interactions among all these combinations [7].

Therefore, the aim of this paper is to search for the most disease associated (risk) and the most disease resistant (protective) *k* SNP sets. Following the method in [8], we modeled the disease risk/protective factors as the multi-SNPs (with specified alleles) with the maximum odds ratio, which was defined as the ratio of the odds of the disease occurring in the exposed group compared with the unexposed group. When the frequency of multi-SNPs in the case group is greater than that in the controls, it is regarded

*Corresponding author (email: qkpeng@xjtu.edu.cn)

as a risk factor. In the reverse situation, it is regarded as a protective factor. On the basis of the relationship between nucleotides and amino acids, we extended the risk/protective factor to two novel versions: (1) *k*-relaxed risk/protective factors, in which exposed and unexposed individuals can deviate in at most *k* sites from the given set of multi-SNPs; (2) weighted-relaxed risk/protective factors, in which individuals can have a distance within a threshold from the given set of multi-SNPs. The sheer number of SNPs involved [9,10] mean that some current combinatorial approaches are too computationally intensive to detect higher order interactions in large datasets [11,12]. Thus, new methodologies for solving this problem are required.

This problem is *NP*-hard and can be viewed as a feature selection problem. Ref. [9] showed that evolutionary algorithms are particularly suited for *NP*-hard problems. In this paper, we use the Bayesian Optimization Algorithm (BOA) [13,14] to identify the various versions of risk/protective factor. The BOA, which is a promising approach in the estimation of distribution algorithms (EDAs) that models the probabilistic model of solutions on the basis of Bayesian networks (BN), is able to detect a SNP that has a weak main effect, but has significant interaction with other SNPs. However, the number of possible structures of the BN grows exponentially with the increasing number of variables, and learning the optimal structure of the network by considering all possible structures not feasible [15]. In this paper, binary particle swarm optimization (BPSO) is used to learn the structure of the BN. The hybrid algorithm BOA_BPSO was used to identify the risk/protective factor for four diseases. The experimental results demonstrated that our algorithm is a powerful tool for discovering the mapping relationship between a disease and SNPs.

1 Problem formulation

Assume that we have *m* samples (each one with *n* SNPs).

Let $\sum = \{0,1,2\}$ denote the value of each SNP, where 0 and 1 stand for homozygous sites with major and minor alleles, respectively, and 2 stands for heterozygous sites.

1.1 Risk/protective factors

SNPs in the coding region can alter the amino acid sequence and increase or decrease the risk developing a disease [16]. The risk/protective factor can be modeled as the multiple SNPs resulting in causation/prevention of a disease. Here, the odds ratio is used to measure the risk/protective factor, which is defined as follows

$$OR\_\text{risk} = \frac{d/(D-d)}{h/(H-h)}, \tag{1}$$

$$OR\_\text{protective} = \frac{h/(H-h)}{d/(D-d)}, \tag{2}$$

where d and h are the number of cases and controls with specified alleles, respectively. D and H are the number of cases and controls, respectively. The larger the OR_risk, the stronger the positive (risk) association between the combination of SNPs and the disease. Similarly, the larger the $OR\_\text{protective}$, the stronger the negative (protective) association between the multiple SNPs and the disease.

1.2 *k*-relaxed risk/protective factors

We propose a novel kind of risk/protective factor (termed *k*-relaxed risk/protective factor, hereafter) that incorporates into the model the codon, which comprises three consecutive bases that encode an amino acid. As mentioned in the previous section, SNPs in the coding region can alter the amino acid sequence. However, that could happen if one or two bases in the codon are varied rather than all of the three bases (Figure 1, only the first base is varied). Therefore, we propose a *k*-relaxed risk/protective factors in which exposed individuals can deviate in at most *k* sites from a given set of multiple SNPs. The *k_OR* for risk and protective factors are defined as follows

$$k\_OR\_\text{risk} = \frac{d\_k/(D-d\_k)}{h\_k/(H-h\_k)}, \tag{3}$$

$$k\_OR\_\text{protective} = \frac{h\_k/(H-h\_k)}{d\_k/(D-d\_k)}, \tag{4}$$

where $\bullet\_k$ is the number of individuals with at most *k* different SNPs from a specified set of multi-SNPs.

1.3 Weighted-relaxed risk/protective factors

In this subsection, we extend the *k*-relaxed *OR* to the weighted-relaxed risk/protective factor. There are $4^3 = 64$ possible different codon combinations with a triplet codon of four nucleotides; however, only 20 standard amino acids are involved in translation, thus several codon combinations encode one amino acid (Figure 2). If the variation occurs in the last base of a codon (C→G), there is no effect on encoded amino acid. However, if the variation occurs in the first or second base, the amino acid would be changed, i.e., SNPs have various weights. The *w_OR* for risk and protective factors are defined as follows

$$w\_OR\_\text{risk} = \frac{d\_w/(D-d\_w)}{h\_w/(H-h\_w)}, \tag{5}$$

$$w\_OR\_\text{protective} = \frac{h\_w/(H-h\_w)}{d\_w/(D-d\_w)}, \tag{6}$$

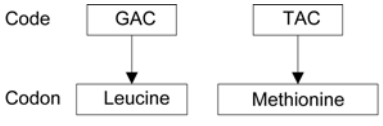

Figure 1 Example of a SNP that changes the amino acid.

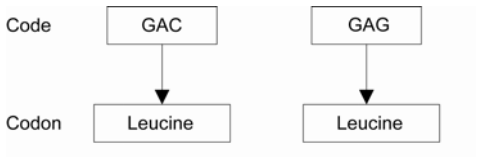

Figure 2 Example of a SNP that does change the encoded amino acid.

where $\bullet\_w$ is the number of individuals with $\sum_i w_i |x_i - s_i| \leqslant w$, w is a threshold, x is an individual (case or control), s_i is a specified multi-SNP, $w_i = OR\_\bullet_i$.

2 Methods

As mentioned in Section 1, evolutionary algorithms (EAs) are particularly suited for problems to which exhaustive enumeration cannot be applied. EDAs were selected to analyze the disease risk in this paper. EDAs could overcome certain drawbacks presented by classical EAs [17]. Unlike EAs, which rely on variation operators to produce offspring, EDAs create offspring through sampling a probabilistic model that has been learned during the optimization process [18,19]. BOA is a promising approach in EDAs, which employs Bayesian networks (BNs) as its probabilistic model to model dependencies among variables [13].

2.1 BOA

A BN is a directed acyclic graph where the nodes correspond to the variables in the data set and the edges correspond to the conditional dependencies that are represented as conditional probabilities [20,21]. The probabilistic model of BOA is shown as follows

$$P(x_1, x_2, ..., x_n) = \prod_{i=1}^{n} P(x_i \mid \pi_i), \tag{7}$$

where $X = (x_1, x_2, ..., x_n)$ is a vector of variables, and π_i is the list of parents of variable x_i.

The BOA generates the first population at random with uniform distribution [22]. Next, better solutions are selected. Then, a BN that fits the selected solutions is constructed. Finally, several new individuals are generated using the joint distribution encoded by the network.

2.2 Encoding

Each individual is represented as follows, where N is the number of SNPs involved, and is used to represent a particular combination of SNPs.

$$S_i = (s_{i1}, s_{i1'}, s_{i2}, s_{i2'}, ..., s_{iN}, s_{iN'}). \tag{8}$$

The individual of BOA is a binary vector, there are four kinds of combination (0, 0), (0, 1), (1, 0), and (1, 1) for $s_{ij}, s_{ij'}$. Thus, the following rule is used to select multi-SNPs with specified alleles

$$s_{ij}, s_{ij'} = \begin{cases} 0,0 & \text{select } jth \text{ SNP with genotype } 0 \\ 0,1 & \text{select } jth \text{ SNP with genotype } 1 \\ 1,0 & \text{select } jth \text{ SNP with genotype } 2 \\ 1,1 & \text{non-select } jth \text{ SNP} \end{cases}. \tag{9}$$

2.3 Learning the structure

The efficiency of BOA depends on how well the BN reflects the dependencies of the variables [23]. There are two major tasks of constructing a BN: (1) learning the structure (the topology of the network); and (2) learning the parameters (the conditional probabilities). Learning the parameters for a specified structure is easy; however, learning the structure is difficult [15]. There are two components for learning the structure: (1) a scoring metric (used to measure how good the network model is); and (2) a search procedure (used to explore the space of possible networks to find the one with the highest score) [23].

(i) Scoring metric. In this study, the Bayesian-Dirichlet (BD) metric, which is defined as follows, was used as the scoring metric of BN.

$$P(D \mid B_S) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}!, \tag{10}$$

where B_s is a network, D is the population of promising solutions, N_{ijk} is the number of cases in D, $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. For more detail see [13].

(ii) Search procedure. Particle swarm optimization (PSO) is an iterative optimization algorithm inspired by the observation of collective behaviors in animals (e.g., bird flocking) [24]. In PSO, each candidate solution to an optimization problem is represented by one particle. Each particle i is described by its position x_i and velocity v_i. The algorithm starts with random initialization of the particles. The particles then change their positions according to their velocities, which are updated in each iteration. Given that p_i is the best position found by particle i in all the preceding iterations and p_g is the best position found so far by the entire swarm, the velocity and position of particle i in bit j will be updated according to the following formulae:

$$\begin{aligned} v_{ij}(t+1) = v_{ij}(t) &+ c_1 r_1 (p_{ij}(t) - x_{ij}(t)) \\ &+ c_2 r_2 (p_{gj}(t) - x_{ij}(t)), \end{aligned} \tag{11}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), \tag{12}$$

where r_1 and r_2 are random numbers between 0 and 1, and c_1 and c_2 define the degree of influence of p_i and p_g on the particle's velocity. The velocity v_{ij} is bounded within a range of $[-V_{\max}, V_{\max}]$ to prevent the particle from flying out of the solution space.

Many optimization problems are set in a discrete space; therefore, Kennedy and Eberhart extended the PSO to the BPSO in 1997 [24]. In BPSO, a particle moves in a state

space restricted to zero or one in each bit, where v_{ij} represents the probability of the bit x_{ij} taking the value 1. Therefore, v_{ij} must be constrained to the interval [0.0, 1.0]. A logistic transformation $S(v_{ij})$ can be used to accomplish this modification, and the position update function is defined as follows

$$x_{ij}(t+1)=\begin{cases}1 & \text{rand}() < \dfrac{1}{1+e^{-v_{ij}}} \\ 0 & \text{otherwise}\end{cases}, \tag{13}$$

where $\dfrac{1}{1+e^{-v_{ij}}} \in (0,1)$ and rand() is a random number selected from a uniform distribution in [0.0, 1.0].

(iii) BPSO encoding. A BN structure can be represented by a particle of BPSO with $(2*N)^2$ dimensions. It can be viewed as a $2*N\times 2*N$ matrix X, in which the element x_{ij} is defined as follows

$$x_{ij}=\begin{cases}1 \text{ if } X_i \text{ is parent of } X_j \\ 0 \text{ otherwise}\end{cases}, \tag{14}$$

where N is the number of SNPs involved, i.e., the dimension of an individual from the BOA; X corresponds to s.

(iv) Avoiding the illegal individuals. It must be taken into account that illegal individuals (solutions with cycles) could be generated during the iteration. Therefore, we used a repairing operator to transform the illegal individuals to legal ones. If two nodes are each other's parent, one of the nodes is selected randomly and its parental relationship to the other node is removed. That is, the corresponding bit is altered from 1 to 0. To avoid the occurrence of reflexive edges, the diagonals of the candidate network matrix are set to 0.

The details of the BOA_BPSO are given as follows:

Step 1: Initialize the BOA, which generates M individuals randomly.

Step 2: Compute the fitness value (OR) of each individual.

Step 3: Select a set of promising individuals.

Step 4: Construct the BN based on the selected individuals.

Step 4.1: Generate N velocity and position vectors randomly for BPSO;

Step 4.2: Repair the illegal BNs.

Step 4.3: Compute the fitness value (eq. (10)) of each particle.

Step 4.4: Update the p_i and p_g.

Step 4.5: Update the BPSO velocity and position vectors.

Step 4.6: Output the best particle (the best BN) of BPSO.

Step 5: Generate a set of new individuals according to the above BN.

Step 6: If the termination criteria are met, stop, otherwise, go to (step 2).

3 Experimental results

3.1 Dataset

In this paper, four datasets (Autoimmune disorder (AD), Crohn's disease (CD), Lung cancer (LC), and Tick-borne encephalitis (TE)) were used to evaluate the effectiveness of our algorithm. All the datasets were supplied by Brinza et al. [8]. Their characteristics are summarized in Table 1.

3.2 Results

Firstly, the quality of our method was evaluated by the *OR* (risk or protective; Table 2). In the table, "D" indicates the dataset and "R/P" indicates risk/protective. The 5th and 6th columns show the frequencies of the best combination of SNPs, i.e., risk/protective factors, in case and control groups, respectively. The last column shows the baseline of RFs/PFs, which is defined as the *number of cases/number of controls* or its reciprocal. The experimental results in Table 2 show that the BOA_BPSO was able to effectively determine the difference between cases and controls. Therefore, we believe that the method used in this paper is a promising tool that can be used to discover SNPs interaction that cause/prevent diseases.

Tables 3 and 4 show the results of *k_OR*_risk/protective and *w_OR*_risk/protective on the four datasets, respectively, where k=5 and w=1.2. For all datasets, *OR* was outperformed by *k_OR*, which was inferior to *w_OR* on most of

Table 1 Description of the datasets

Dataset	No. of SNPs	No. of cases	No. of controls
CD	103	144	243
AD	108	384	652
LC	141	322	273
TE	41	21	54

Table 2 Searching for the risk/protective factors using BOA_BPSO

R/P	D	*OR*	*OR* with 95% CI	Case frequency	Control frequency	*P*-value	No. of SNPs in that combination	Baseline
R	CD	68.04	4.16–1.11×10^3	0.19	0.00	2.84×10^{-43}	10	0.59
	AD	58.62	3.60–954.23	0.07	0.00	1.03×10^{-62}	16	0.59
	LC	105.45	6.44–1.73×10^3	0.14	0.00	6.54×10^{-46}	12	1.18
	TE	57.64	3.31–1.00×10^3	0.57	0.02	2.09×10^{-12}	5	0.39
P	CD	94.19	5.67–1.57×10^3	0.00	0.16	1.04×10^{-28}	5	1.69
	AD	70.13	4.22–1.17×10^3	0.00	0.05	2.07×10^{-33}	10	1.70
	LC	111.03	6.82–1.81×10^3	0.00	0.17	8.74×10^{-57}	10	0.85
	TE	68.63	3.70–1.27×10^3	0.01	0.38	2.72×10^{-8}	6	2.57

Table 3 Searching for the *k*-relaxed risk/protective factors using BOA_BPSO

R/P	D	*OR*	*OR* with 95% CI	Case frequency	Control frequency	*P*-value	No. of SNPs in that combination
R	CD	134.52	$8.26–2.19\times10^{3}$	0.32	0.00	1.67×10^{-54}	22
	AD	94.05	$5.81–1.52\times10^{3}$	0.11	0.00	8.40×10^{-83}	25
	LC	219.21	$13.50–3.56\times10^{3}$	0.25	0.00	4.52×10^{-65}	23
	TE	272.33	$14.75–5.03\times10^{3}$	0.87	0.02	4.75×10^{-10}	13
P	CD	129.72	$7.86\text{-}2–14\times10^{3}$	0.00	0.21	7.90×10^{-34}	25
	AD	129.49	$7.91–2.12\times10^{3}$	0.00	0.09	8.62×10^{-51}	25
	LC	243.52	$15.01–3.94\times10^{3}$	0.00	0.31	1.28×10^{-77}	22
	TE	259.92	$13.72–4.92\times10^{3}$	0.01	0.71	5.36×10^{-11}	21

Table 4 Searching for the weighted-relaxed risk/protective factors using BOA_BPSO

R/P	D	*OR*	*OR* with 95% CI	Case frequency	Control frequency	*P*-value	No. of SNPs in that combination
R	CD	170.39	$10.48–2.77\times10^{3}$	0.37	0.00	1.60×10^{-57}	19
	AD	110.63	$6.84–1.79\times10^{3}$	0.13	0.00	2.02×10^{-90}	25
	LC	323.38	$19.95–5.24\times10^{3}$	0.33	0.00	1.02×10^{-74}	48
	TE	235.24	$12.89–4.29\times10^{3}$	0.85	0.02	2.82×10^{-10}	15
P	CD	252.42	$15.41–4.13\times10^{3}$	0.00	0.34	1.73×10^{-44}	16
	AD	97.20	$5.80–1.60\times10^{3}$	0.00	0.07	5.45×10^{-42}	16
	LC	304.47	$18.82–4.83\times10^{3}$	0.00	0.36	1.34×10^{-82}	54
	TE	423.89	$21.31–8.43\times10^{3}$	0.01	0.81	1.80×10^{-11}	21

the datasets. The *k_OR* found more significant RFs/PFs than the *OR* based method and *w_OR* found the most significant ones. However, for TE, the *k_OR* based method found more significant RFs than *w_OR*. Thus, we conclude that the *k_OR* and *w_OR* methods are significantly better than *OR*.

Figures 3 to 6 show the results for the three kinds of risk/protective factor searched by our method. The *x*-axis corresponds to the iteration of BOA_BPSO. It is clear that the BOA_BPSO had a fast convergence speed at the early stages of the optimization process during most of the test, and it can maintain the diversity of the population at the latter stages. In addition, we can see from the figures that the weighted-relaxed risk/protective factors and *k*-relaxed risk/protective factors have an obvious advantage over the traditional factor.

In this section, the BOA_BPSO was compared with a genetic algorithm (GA), (BPSO), univariate marginal distribution algorithm (UMDA), and randomized complimentary greedy search (RCGS) [8] (Table 5). The aim of this paper was to identify the most disease associated or the most disease resistant SNP sets, i.e., a SNP set with the maximum odds ratio. Thus, the higher the *OR*, the better the algorithm. Table 5 shows that our method outperformed the other algorithms on all datasets. That may be explained by the fact that the other algorithms do not take into account the interdependent relations of SNPs. For the *OR* of TE, our method was only slightly better than that of RCGS; however, our results were comparable. In addition, some methods obtained the same "number of SNPs in that combination"; however, the *OR* may be different, which can be explained by the fact that the actual selected SNPs may be different, even if the total number (the number of SNPs in that

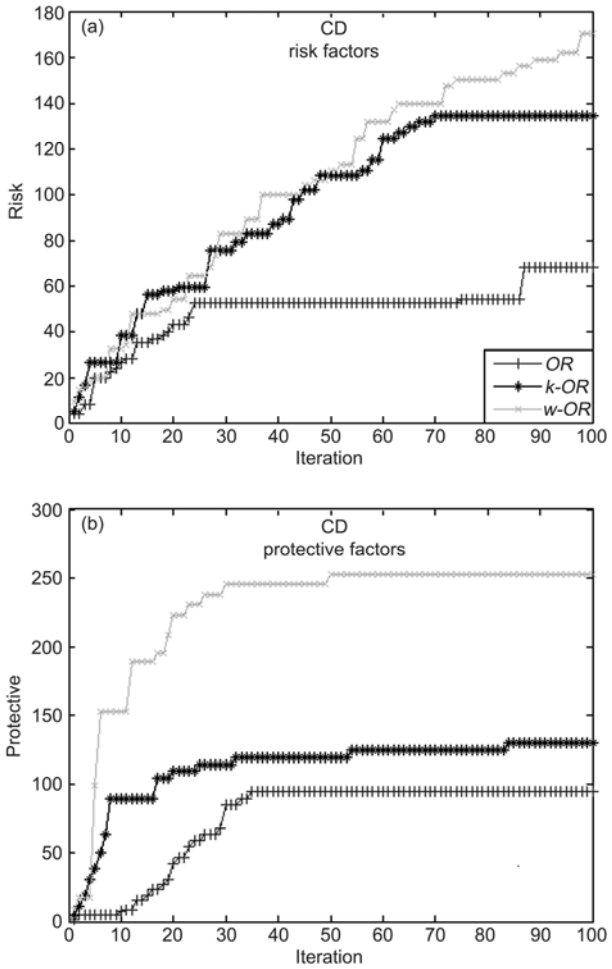

Figure 3 Comparison of three types of risk/protective factors versus iteration on the CD dataset.

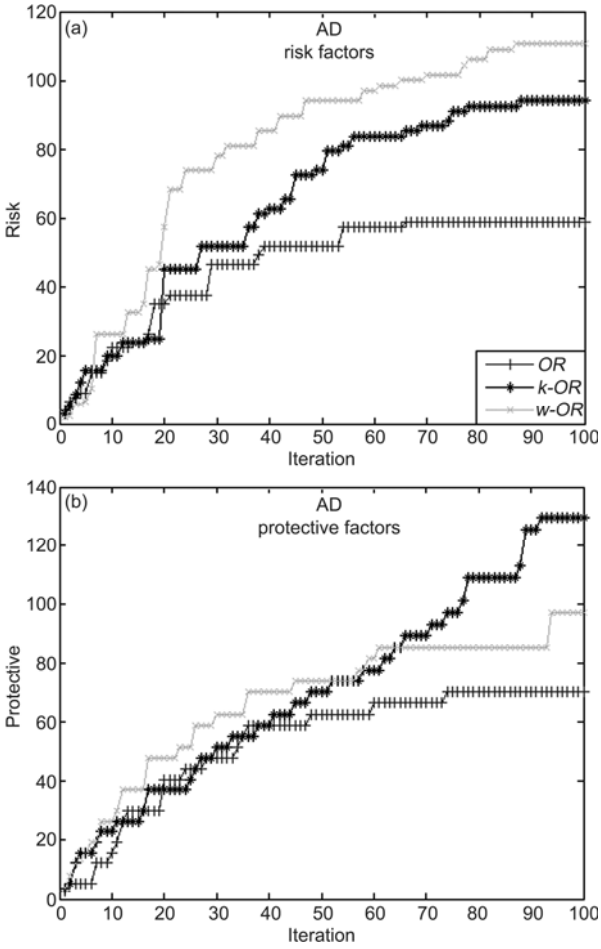

Figure 4 Comparison of three types of risk/protective factors versus iteration on the AD dataset.

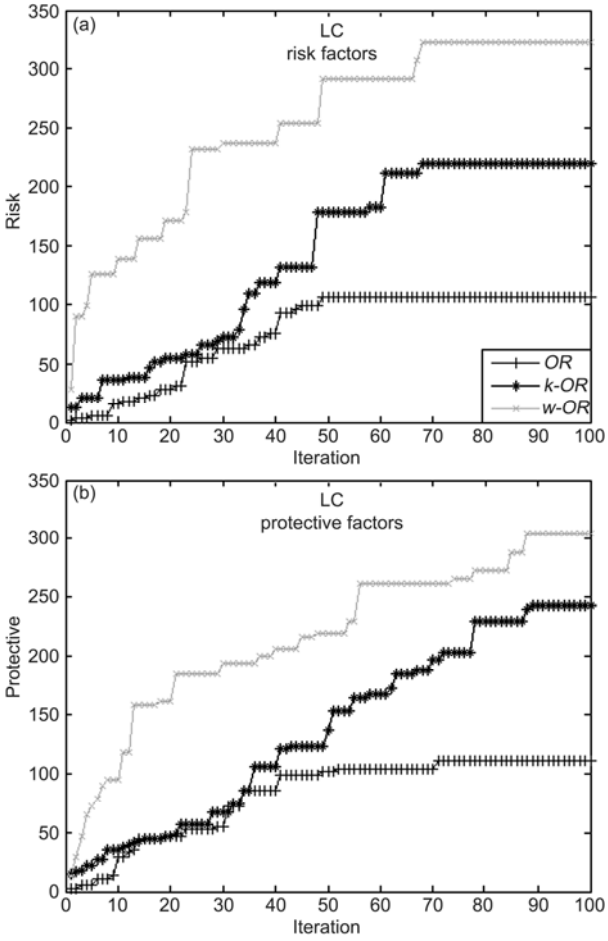

Figure 5 Comparison of three types of risk/protective factors versus iteration on the LC dataset.

Table 5 Comparison of five methods in searching for various risk factors

D	Methods	*OR*	Case frequency	Control frequency	Running time (s)	No. of SNPs in that combination
CD	GA	48.93	0.05	0.01	4400	38
	BPSO	58.68	0.06	0.00	3165	19
	UMDA	59.35	0.06	0.00	1018	17
	RCGS	52.23	0.11	0.00	955	13
	BOA_BPSO	68.04	0.19	0.00	1958	10
AD	GA	55.24	0.02	0.00	15304	17
	BPSO	52.43	0.01	0.01	781	78
	UMDA	54.08	0.03	0.00	4940	12
	RCGS	57.06	0.05	0.00	4025	17
	BOA_BPSO	58.62	0.07	0.00	6895	16
LC	GA	74.87	0.04	0.00	4552	7
	BPSO	86.80	0.05	0.00	3165	13
	UMDA	87.95	0.07	0.00	2018	12
	RCGS	97.82	0.15	0.00	1550	12
	BOA_BPSO	105.45	0.14	0.00	2716	12
TE	GA	35.23	0.19	0.01	2463	12
	BPSO	49.58	0.39	0.00	1846	12
	UMDA	52.14	0.38	0.00	1094	9
	RCGS	57.33	0.31	0.00	520	5
	BOA_BPSO	57.64	0.57	0.02	1208	5

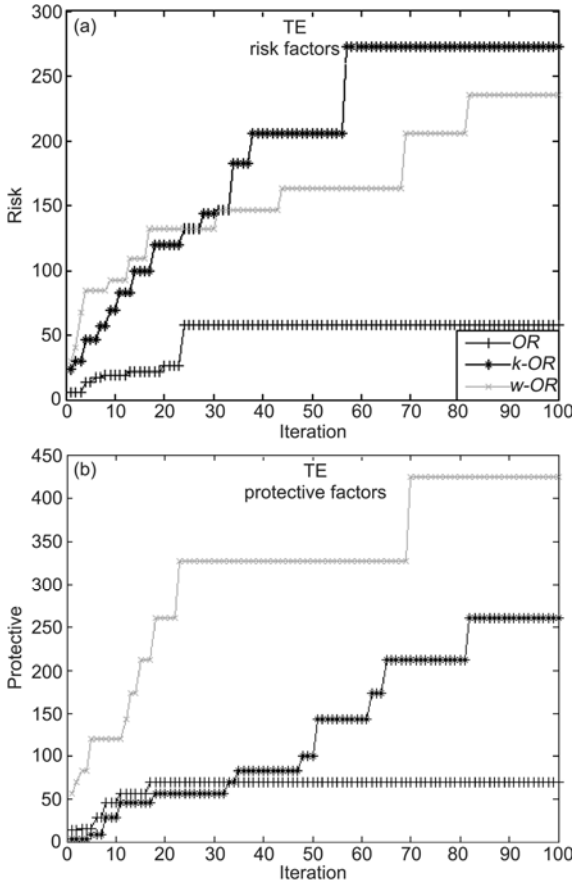

Figure 6 Comparison of three types of risk/protective factors versus iteration on the TE dataset.

combination) is the same. Our algorithm needs to learn the BN structure; therefore, it was slower (but acceptable) than some of others. However, it is faster than GA and BPSO. Thus, in the near future, our main task is to improve the time efficiency of the algorithm. Finally, we conclude that our algorithm better able to identify the most disease associated SNPs than the other methods.

To determine whether the differences between the BOA_BPSO and the other five algorithms were statistically significant, we used a *t*-test with a 0.05 level of significance (each experiment was conducted 30 times). Table 6 shows the value of the two-tailed *P*-values. Table 6 shows that the differences between the results obtained by BOA_BPSO and those of other four algorithms were statistically significant in almost all cases. The results signify that the null hypothesis is false and the methods differ significantly, i.e., the proposed method beats the competitors in a statistically meaningful way.

3.3 Sensitivity in relation to parameters

In this subsection, the effects of k and w on the solutions were investigated (Figures 7 and 8). In the figures, the x-axis corresponds to the value of k or w. k was increased from 3 to 23 in steps of 2, and w was increased from 0.8 to 5.6. Variations of solutions were observed with different k and w for most of the datasets. A too large or small value of k and w made the results of k-relaxed risk/protective factors and weighted-relaxed risk/protective factors come close to (or be worse than) the risk/protective factors. Figures 7 and 8 indicate that our algorithm achieves the best results when k=5 and w=1.2.

4 Conclusions

It is unlikely that a single SNP will efficiently discriminate between cases and controls; however, it is plausible that a set of SNPs could contribute to diseases risk. In this paper, three kinds of risk/protective factor were used to investigate the relationship between SNPs and disease. However, the large number of SNPs makes association studies difficult to conduct. Therefore, BOA_BPSO was proposed to overcome the shortcoming presented by traditional algorithms. The BOA was used to identify the multi-SNPs associated with diseases, and the BPSO was used to learn the structure of the BOA network. The experimental results showed that the algorithm used in this paper is a promising method for discovering SNP interactions that cause/prevent diseases.

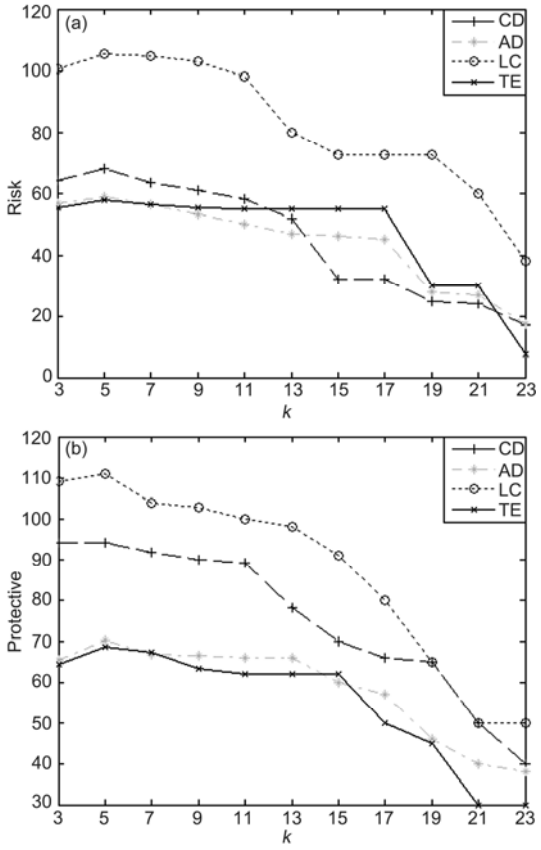

Figure 7 The results of k-relaxed risk/protective factors with various k. (a) For risk factors; (b) for protective factors.

Table 6 The *t*-test between BOA_BPSO and the other four algorithms on five datasets

D	Methods	*t*-test	Methods	*t*-test
CD	BOA_BPSO-GA	8.756×10^{-34}	BOA_BPSO-BPSO	9.661×10^{-25}
	BOA_BPSO-UMDA	4.861×10^{-19}	BOA_BPSO-RCGS	2.813×10^{-27}
AD	BOA_BPSO-GA	4.013×10^{-15}	BOA_BPSO-BPSO	6.310×10^{-19}
	BOA_BPSO-UMDA	1.316×10^{-9}	BOA_BPSO-RCGS	8.413×10^{-4}
LC	BOA_BPSO-GA	1.234×10^{-35}	BOA_BPSO-BPSO	6.126×10^{-30}
	BOA_BPSO-UMDA	4.156×10^{-29}	BOA_BPSO-RCGS	7.515×10^{-14}
TE	BOA_BPSO-GA	4.130×10^{-20}	BOA_BPSO-BPSO	8.133×10^{-17}
	BOA_BPSO-UMDA	1.034×10^{-6}	BOA_BPSO-RCGS	2.015×10^{-2}

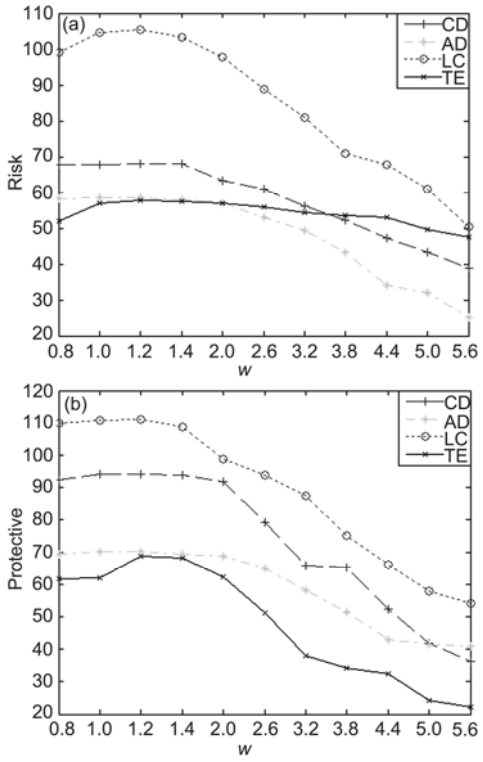

Figure 8 The results of weighted-relaxed risk/protective factors with various *w*. (a) For risk factors; (b) for protective factors.

This work was supported by the National Natural Science Foundation of China (*60774086 and 61173111*) *and Ph. D. Program Foundation of Ministry of Education of China* (*20090201110027*).

1 Wei B, Peng Q K, Zhang Q W, et al. Identification of a combination of SNPs associated with Graves' disease using swarm intelligence. Sci China Life Sci, 2011, 54: 139–145
2 Moore J H. The ubiquitous nature of epistasis in de-termining susceptibility to common human diseases. Hum Hered, 2003, 56: 73–82
3 Jasnos L, Korona R. Epistatic buffering of fitness loss in yeast double deletion strains. Nat Genet, 2007, 39: 550–554
4 Martin G, Elena S F, Lenormand T. Distributions of epistasis in microbes fit predictions from a fitness landscape model. Nat Genet, 2007, 39: 555–560
5 Hirschhorn J N, Daly M J. Genome-wide association studies for common diseases and complex traits. Nat Rev Genet, 2005, 6: 95–108
6 Mccarthy M I, Abecasis G R, Cardon L R, et al. Genome-wide association studies for complex traits: Consensus, uncertainty and challenges. Nat Rev Genet, 2008, 9: 356–369
7 Culverhouse R, Suarez B K, Lin J, et al. A perspective on epistasis: Limits of models displaying no main effect. Am J Hum Genet, 2002, 70: 461–471
8 Brinza D, Zelikovsky A. Design and validation of methods searching for risk factors in genotype case-control studies. J Comput Biol, 2008, 15: 81–90
9 Kelemen A, Vasilakos A V, Liang Y. Computational intelligence in bioinformatics: SNP/Haplotype data in ge-netic association study for common diseases. IEEE Trans Inf Technol Biomed, 2009, 13: 841–847
10 Thornton T A, Moore J H, Haines J L. Genetics, statistics and human disease: Analytical retooling for com-plexity. Trends Genet, 2004, 20: 640–647
11 Hirschhorn J N. Genomewide association studies illuminating biologic pathways. N Engl J Med, 2009, 360: 1699–1701
12 Goldstein D B. Common genetic variation and human traits. N Engl J Med, 2009, 360: 1696–1698
13 Pelikan M, Goldberg D E, CantuPaz E. BOA: The Bayesian optimization algorithm. P Genet Evol Comput Conf, 1999, 525–532
14 Pelikan M, Goldberg D E, Cantu P E. Linkage problem, distribution estimation, and Bayesian networks. Evol Comput, 2000, 8: 311–340
15 Heckerman D, Geiger D, Chickering D M. Learning Bayesian networks: The combination of knowledge and statistical data. Mach Learn, 1995, 20: 197–243
16 Ruczinski I, Kooperberg C, LeBlanc M L. Ex-ploring interactions in high-dimensional genomic data: An overview of Logic Regression, with applications. J Multi-variate Anal, 2004, 90: 178–195
17 Bashir S, Naeem M, Shah S I. A comparative study of heuristic algorithms: GA and UMDA in spatially multiplexed communication systems. Eng Appl Artif Intel, 2010, 23: 95–101
18 Chen T, Ke T, Chen G L, et al. Analysis of com-putational time of simple estimation of distribution algo-rithms. IEEE Trans Evol Comput, 2010, 14: 1–22
19 Shapiro J L. Drift and scaling in estimation of distribution algorithms. Evol Comput, 2005, 13: 99–123
20 Chrubasik B. Readings on the principles and applications of decision-analysis: Vol 1: General collection; vol 2: Professional collection-Howard, RA, Matheson, JE. Eur J Oper Res, 1986, 27: 383–384
21 Kyburg H E. Probabilistic reasoning in intelligent systems-networks of plausible inference-pearl. J Philos, 1991, 88: 434–437
22 Schwarz J, Ocenasek J. A problem-knowledge based evolutionary algorithm KBOA for hypergraph parti-tioning. In: Proceedings of the Fourth Joint Conference on Knowledge-Based Software Engineering, IO Press, Brno, Czech Republic, 2000. 51–58
23 Pelikan M, Sastry K, Goldberg D E. Scalability of the Bayesian optimization algorithm. Int J Approx Reason, 2002, 31: 221–258
24 Kennedy J, Eberhart R C. A discrete binary ver-sion of the particle swarm algorithm. Conf Proc—IEEE Int Conf Syst Man Cybern, 1997, 5: 4104–4108