Journal of Biomedical Informatics 46 (2013) 328–340

Contents lists available at SciVerse ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

A genetic algorithm–support vector machine method with parameter optimization for selecting the tag SNPs

İlhan İlhan [a,*], Gülay Tezel [b,1]

[a] Akören Vocational School, Selçuk University, 42460 Akören, Konya, Turkey
[b] Department of Computer Engineering, Fac. Eng. Arch., Selçuk University, 42003 Konya, Turkey

ARTICLE INFO

Article history:
Received 10 January 2012
Accepted 11 December 2012
Available online 20 December 2012

Keywords:
Single Nucleotide Polymorphisms (SNPs)
Tag SNPs
Genetic algorithm (GA)
Support vector machine (SVM)
Particle swarm optimization (PSO)

ABSTRACT

SNPs (Single Nucleotide Polymorphisms) include millions of changes in human genome, and therefore, are promising tools for disease-gene association studies. However, this kind of studies is constrained by the high expense of genotyping millions of SNPs. For this reason, it is required to obtain a suitable subset of SNPs to accurately represent the rest of SNPs. For this purpose, many methods have been developed to select a convenient subset of tag SNPs, but all of them only provide low prediction accuracy. In the present study, a brand new method is developed and introduced as GA–SVM with parameter optimization. This method benefits from support vector machine (SVM) and genetic algorithm (GA) to predict SNPs and to select tag SNPs, respectively. Furthermore, it also uses particle swarm optimization (PSO) algorithm to optimize C and γ parameters of support vector machine. It is experimentally tested on a wide range of datasets, and the obtained results demonstrate that this method can provide better prediction accuracy in identifying tag SNPs compared to other methods at present.

1. Introduction

One of the important study subjects about human genome is the investigation of genetic variants related to *complex diseases*. Most of these *genome-wide association* (*GWA*) studies [1,2] are aimed to determine genetic variants possibly related to complex diseases. Genetic variants mostly consist of *SNPs* (*Single Nucleotide Polymorphisms*), and human genome is estimated to include around 10 million SNPs [3]. In this regard, it is generally preferred to use SNPs in GWA studies [4,5]. The number of individuals and SNPs are quite effective on the statistical significance of a GWA study [6]. However, it is still very expensive and time-consuming to genotype all the SNPs in a large population found in the candidate area for large-scale GWA studies [7–9]. Therefore, a subset of SNPs should be selected to predict *the rest of SNPs* with an acceptable error limit. In this subset, a single SNP is called a *tag SNP*. In addition, it is quite important to determine a minimum subset of tag SNPs to predict the rest of SNPs with maximum accuracy [7–10].

There are different methods developed to select tag SNPs in recent years [7–29]. These can be categorized into three main groups as *block-based*, *linkage disequilibrium* (*LD*)*-based* and *block-free* methods. Among them, block-based methods are mainly established on the block structure of human genome [30,31]. These methods are based on the fact that human genome can be divided into discrete blocks, and small sets of common haplotypes in each block are shared by a specific population. In this respect, these methods aim to find a subset of SNPs in order to distinguish all of the shared haplotypes [11–15]. Accordingly, a human genome is divided into haplotype blocks in the first place, and later, a subset of tag SNPs is selected for each block in the second place. However, it is not always possible to determine the blocks accurately, which is the main problem of this method, and it is still under discussion in which way to define these blocks [16]. Furthermore, inter-block correlations are ignored during the selection of tag SNPs, which is performed only by the local correlation between the markers of each block [16].

Another type of tag SNP selection methods is LD-based methods which consider the relation between SNP pairs (linkage disequilibrium). The main aim of these methods is to choose a set of tag SNPs highly related to each SNP on a definite haplotype [23–25,27–29]. However, it is not easy to decrease the number of tag SNPs on loci with low LD.

On the other hand, many block-free methods were introduced recently by investigators [7–10,16–22]. These methods consider tag SNPs as a subset of all SNPs to re-build the rest of SNPs. Contrary to the block-based methods, block division and limited haplotype diversity are not used in these methods. Instead, weaker correlations between adjacent blocks are preferred [18,19]. Lin and Altman [17] developed a method (*Eigen2htSNP*) to predict a tagged SNP by using a tag SNP with the highest correlation with the tagged SNP. However, the prediction accuracy of Eigen2htSNP method is low due to the slightly correlated SNPs used in SNP

[*] Corresponding author. Fax: +90 332 461 28 92.
E-mail addresses: ilhan@selcuk.edu.tr (İ. İlhan), gtezel@selcuk.edu.tr (G. Tezel).
[1] Fax: +90 332 241 06 35.

1532-0464/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jbi.2012.12.002

prediction. Similarly, Halperin et al. [7] suggested another method (*STAMPA*) to select a subset of tag SNPs. This method selects a minimum of two tag SNPs which could sometimes give worse results than a randomly selected subset of tag SNPs [9]. Another method for selecting a tag SNPs was proposed by Lee and Shatkay [8], which was called *BNTagger* method considering conditional independency among SNPs. The aim of this method is to select independent but highly predictive SNPs through Bayesian networks. In this method, the algorithm does not use the number of tag SNPs to be selected as input variable, and instead, tag SNPs are selected according to previously determined threshold value used as input in algorithm. However, this is a very time-consuming method [9]. On the other hand, He and Zelikovsky [10] suggested two new methods for SNP prediction based on Multiple Linear Regression (*MLR-Tagging*) and Support Vector Machine (*SVM/STSA*), of which SVM/STSA is considered more effective, yet it is still very time-consuming due to its production of hereditary subset of tag SNPs [26] as this is not always useful. Yang et al. [9] proposed a new method (*BPSO*), which is a binary version of particle swarm optimization algorithm. However, this method has the same drawbacks of STAMPA as both methods share the same prediction algorithm. In the recent period, Lin and Leu [21] has developed a hybrid method known as Particle Swarm Optimization–Support Vector Machine (*PSO–SVM*). This method combines PSO and SVM through parameter optimization and property choice. PSO and SVM are used for selection of tag SNPs and for prediction of the rest of SNPs, respectively. However, prediction accuracy is quite poor while the number of tag SNPs is small. Mahdevar et al. [22] suggested a heuristic method (*GTagger*) using genetic algorithms. This method uses correlation and Shannon entropy to calculate fitness function, but its prediction accuracy is low.

Diverse subsets of tag SNPs could be obtained in random selection of tag SNPs by a method using *support vector machine* (*SVM*) [32,33] as SNPs prediction model in the search space [6]. It is well-known that *genetic algorithms* (*GAs*) are able to scan the points in the search space quite well through their genetic operators [34–36]. From this regard, this study suggests a new approach to select tag SNPs and to predict non-tag SNPs by using the genetic algorithm and the support vector machine, respectively. In addition, *particle swarm optimization* (*PSO*) algorithm is used in the study to optimize C and γ parameters of support vector machine. This approach is called as *the GA–SVM method with parameter optimization*. "*Leave-one-out cross-validation*" (*LOOCV*) method is used to evaluate the *prediction accuracy* of the algorithm in this approach. Experimental results on many datasets demonstrate that the suggested approach is able to identify tag SNPs with significantly higher prediction accuracy than other methods at present.

In the rest of the paper, Section 2 explains the selection problem of tag SNPs, Section 3 presents the method used for selection problem of tag SNPs, Section 4 gives the experimental datasets used in the study, Section 5 presents the experimental results, and Section 6 concludes the paper.

2. The selection problem of tag SNPs

A *diploid organism* contains two non-identical copies of each *chromosome* and a set of SNPs on each of these copies is named as *haplotype*, while the data including the combination of two haplotype is named as *genotype* [1]. Each haplotype gives allele information of adjacent SNPs on a given chromosome, and each genotype represents the combined allele information of SNPs on a certain pair of homologous chromosomes (Supplementary Fig. 1) [6].

The most frequently observed nucleotide of a SNP in a given population is referred to as *major allele*, while others are known as *minor allele*. For *bi-allelic* SNPs, haplotypes can be represented as a string of symbols {0, 1}, where 0 and 1 stand for major and minor alleles, respectively. Within a genotype, SNPs are accepted *homozygous* when both alleles are the same, and *heterozygous* when both alleles are different. Therefore, a genotype can also be represented with any of these numbers {0, 1, and 2}, where 0 indicates both alleles of SNP are *major homozygous*, while 1 stands for *minor homozygous*, and 2 stands for *heterozygous* (0/1, 1/0) (Supplementary Fig. 2) [6].

A *haplotype matrix H* is used to determine a suitable subset of tag SNPs to predict the rest of SNPs. Here, each row represents a definite haplotype h_i, $i\{1,2,\ldots,m\}$ and each column indicates a definite SNP$_j$, $j \in \{1,2,\ldots,n\}$. The main problem is to detect a convenient subset $T = \{t_1,t_2,\ldots,t_k\}$ of tag SNPs, which should be minimum of all choices, but enables the prediction of rest of SNPs by a higher accuracy.

3. The GA–SVM method with parameter optimization

In the present study, a new hybrid method is suggested increasing the prediction accuracy of SNPs classification. This method uses genetic algorithm to select tag SNPs, and support vector machine to predict the rest of SNPs. C and γ parameters of support vector machine are optimized by particle swarm optimization algorithm. As mentioned above, this method is called as GA–SVM method with parameter optimization.

Fig. 1 demonstrates this method which consists of modules to perform the procedures including the formation of the initial populations, calculation of fitness value (fitness evaluation), finding pbest values and gbest, calculating the particle speeds and updating their locations, selection, crossover, mutation, and adjusting operations.

3.1. Selecting the tag SNPs by the genetic algorithm

3.1.1. The initial population for SNPs

The initial population of SNPs is represented with a binary matrix where the chromosomes in the population are given in rows, and SNPs are in columns. The input of algorithm is matrix H with m rows and n columns where a certain haplotype and SNPs are represented in each row and column, respectively [10]. Each haplotype in such a matrix is represented by an n-bit binary vector. Through this matrix, genetic algorithm produces a *matrix population of* P_{GA} with n columns and N_p rows where N_p is the number of chromosomes in population randomly selected between 10 and 200 [37,38]. The $p_{ij} \in \{0,1\}$ of the matrix P_{GA} stands for the jth SNP for the ith chromosome. 1 indicates that the relevant SNP is a tag SNP, while 0 shows the necessity to predict the value of relevant SNP. All the chromosomes share the same number of tag SNPs represented by N_{tag}, while different chromosomes have different combinations of N_{tag} SNPs from the n SNPs. For instance, a P_{GA} matrix with $n = 15$ SNPs and 5 chromosomes given in Fig. 2, there are 5 sets of the size 5 that differ from each other by at least one tag SNPs.

Genetic algorithm is an iterative procedure to maintain a constant population size in candidate solutions [39]. In each iteration of this algorithm, three genetic operators (selection, crossover, and mutation) are performed to create a new population (offspring). Chromosomes of new population are assessed by using the fitness function (Eq. (1)) shown in the subsequent section. With this evaluation, better new populations are detected as candidate solutions [39].

3.1.2. The fitness evaluation

Leave-one-out cross validation (LOOCV) method is used for *fitness evaluation* in the study [7–10,17,21,40]. In this method, the jth haplotype is removed from the matrix H in the jth iteration, and the tag SNPs are selected from the remaining haplotypes by

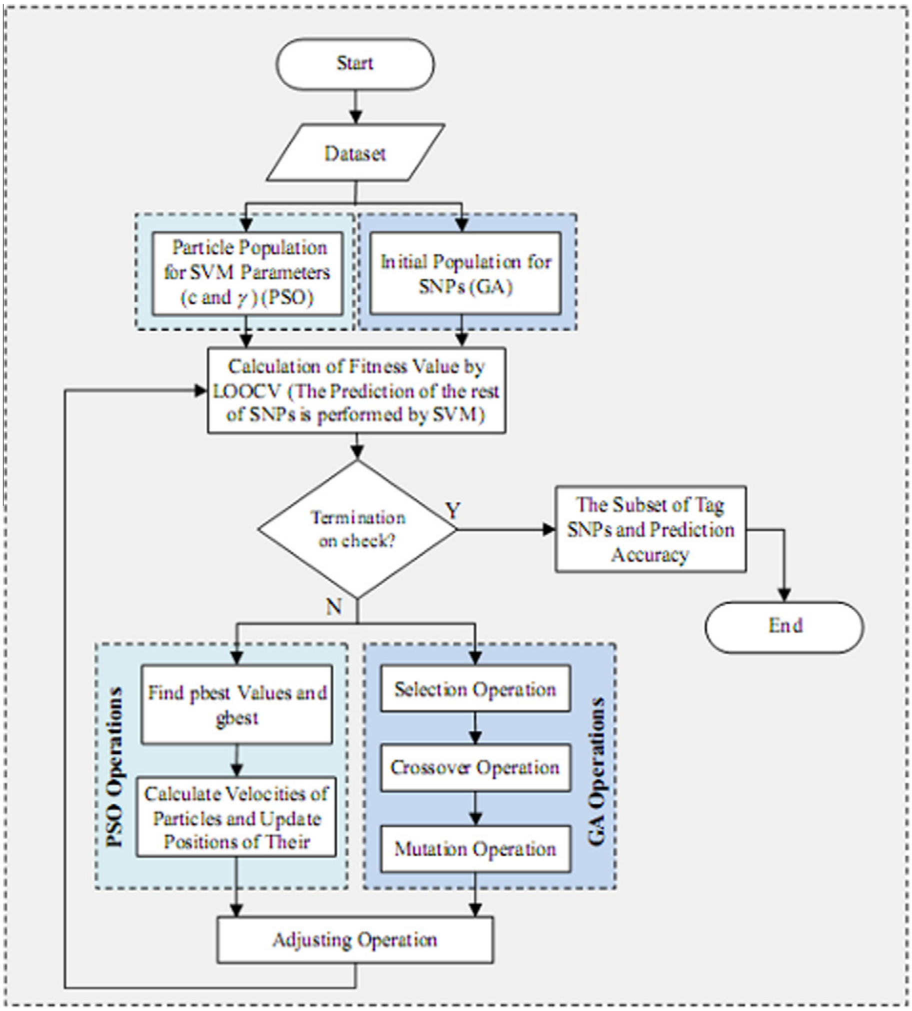
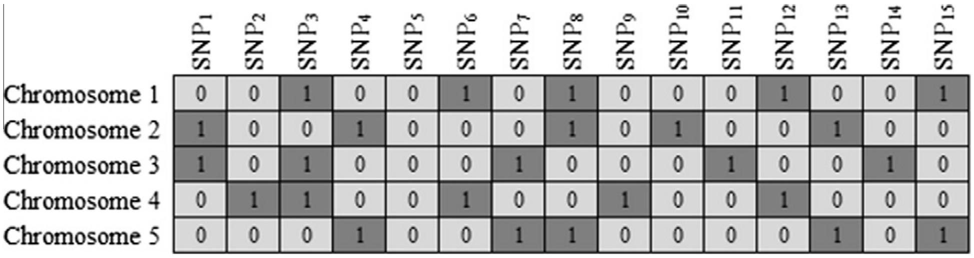

Start
Dataset
Particle Population for SVM Parameters (c and γ) (PSO)
Initial Population for SNPs (GA)
Calculation of Fitness Value by LOOCV (The Prediction of the rest of SNPs is performed by SVM)
Termination on check?
Y
N
The Subset of Tag SNPs and Prediction Accuracy
End
PSO Operations
Find pbest Values and gbest
Calculate Velocities of Particles and Update Positions of Their
GA Operations
Selection Operation
Crossover Operation
Mutation Operation
Adjusting Operation

Fig. 1. The flowchart of the GA–SVM algorithm with parameter optimization.



	SNP_1	SNP_2	SNP_3	SNP_4	SNP_5	SNP_6	SNP_7	SNP_8	SNP_9	SNP_{10}	SNP_{11}	SNP_{12}	SNP_{13}	SNP_{14}	SNP_{15}
Chromosome 1	0	0	1	0	0	1	0	1	0	0	0	1	0	0	1
Chromosome 2	1	0	0	1	0	0	0	1	0	1	0	0	1	0	0
Chromosome 3	1	0	1	0	0	0	1	0	0	0	1	0	0	1	0
Chromosome 4	0	1	1	0	0	1	0	0	1	0	0	1	0	0	0
Chromosome 5	0	0	0	1	0	0	1	1	0	0	0	0	1	0	1

Fig. 2. Population matrix including 5 chromosomes and 15 SNPs. Each chromosome consists of 5 tag SNPs and 10 tagged SNPs.

using the GA. Subsequently, these tag SNPs are used to predict *the tagged SNPs* (the rest of SNPs) present in the removed haplotype. This process is repeated for all $j = 1,2,\ldots,m$, i.e., until all haplotypes in H are used as the validation data. The ratio of the number of SNPs predicted accurately to the total number of predicted SNPs gives the prediction accuracy (fitness value), which is calculated with Eq. (1), where N_c is the number of accurately predicted SNPs, while N_a is the number of all predicted SNPs.

$$PA = N_c/N_a. \qquad (1)$$

3.1.3. The natural selection

Including every chromosome of a population in the selection would not be much effective on the next generation; therefore, chromosomes obtaining higher fitness values should survive, while other chromosomes should be eliminated [41]. Natural selection takes place in each iteration of algorithm. There are different selection methods in GA, including *roulette wheel selection*, *random selection*, *scaling selection*, *tournament selection*, *hierarchical selection*, *etc.* Of these, roulette wheel selection method is used in the present study as it is the most common method [42]. This method looks like a cycling wheel in which each chromosome obtains an area in parallel with its fitness. This method creates a set A of cumulative probabilities of N_p chromosomes in order and a set B of N_p numbers generated randomly in the range of 0 and 1. Subsequently, $c_i = min\{a_i \in A\colon a_i \geqslant b_j\}$ is selected for each number $b_j \in B$. Consequently, the new population set of $C = \{c_i\}_{i=1}^{q}$ is created.

3.1.4. The crossover operation

The *crossover operation* with C_R rate is performed on the new population to improve the new population resulting from natural selection. Chromosomes are randomly selected to apply crossover operation in general. Additionally, the uniform crossover operator is used in the present study to obtain offspring chromosomes from parent chromosomes [43]. For this purpose, a crossover mask has to be created with 0.5 mixing ratio [44], which is used to determine the special bits of the parent chromosomes to crossover. In crossover mask, a 1-bit indicates SNPs related to bit to be crossed between both parents, while a 0-bit demonstrates the SNPs related to bit that should not change. The crossover rate is set to $C_R = 0.9$ in the present study [26,45]. Fig. 3 demonstrates a sample for uniform crossover operation implemented on chromosomes 1 and 3 (Fig. 2) in which the 3rd row is crossover mask produced by mixing ratio of 0.5.

3.1.5. The mutation operation

The *mutation operator* is performed with a mutation rate M_R to improve the population produced by crossover operation. For this purpose, mutation operator modifies certain bits in the population. A random number between 0 and 1 is produced for each bit-position in chromosomes to determine which bits to mutate. When the number is lower than M_R, relevant bits should be mutated through change of every 0 bits to 1, and every 1 bits to 0 [26]. The mutation rate is set to $M_R = 0.01$ in the present study [26,45]. Fig. 4 represents mutation operation performed on SNP_5 for the chromosome 4 (Fig. 2).

3.1.6. Adjusting the number of tag SNPs

After crossover and mutation operations, the number of 1 bits demonstrating the tag SNPs for chromosomes could be changed [26]. Therefore, the number of tag SNPs for each chromosome should be adjusted so that the number M_{tag} of tag SNPs for each chromosome can be equivalent to the number N_{tag} given as input for the GA–SVM algorithm with parameter optimization. There are two methods in literature, suggested solving this problem [9,22,26]. One of them is *random search method*, in which randomly selected $M_{tag} - N_{tag}$ tag SNPs are transformed into 0 if $M_{tag} > N_{tag}$; on the other hand, [22,26]; $N_{tag} - M_{tag}$ SNPs not selected yet are transformed into 1 if $M_{tag} < N_{tag}$ in order to reach the desired number of tag SNPs. Unfortunately, random adjustment of the number of tag SNPs for chromosomes results in diverse prediction accuracy rates for different iterations [9]. The other method is *local search algorithm*, in which the new chromosomes with higher prediction accuracy are fixed in the adjustment process for the number of selected tag SNPs in chromosomes [9]. The prediction accuracy is calculated by the LOOCV method for each candidate chromosome. However, prediction accuracy considerably changes from one iteration to another in the random search method, which complicates the selection of tag SNPs. In order to minimize the fluctuation of the prediction accuracy, it is necessary to increase the number of the iteration excessively. But in this case, the algorithm takes so much time that the solution of many practical applications becomes impossible [35]. The LOOCV method, which is highly time-consuming, is used in local search algorithm to find a new chromosome with the highest prediction accuracy [9]. Therefore, it is also unpractical for many applications [40].

The comparison of experimental results of LOOCV and *10-fold cross-validation methods* reveals that the latter method works approximately 10 times faster than the former one despite the fact they produce the same results [6]. For this reason, 10-fold cross-validation method is used in the local search algorithm [6]. Fig. 3 represents the sample of tag SNPs in offspring 2, marked $M_{tag} = 4$ times with 1's. In Fig. 5, the values of tagged SNPs are replaced from 0 to 1 in the offspring 2 by 11 candidate chromosomes one at a time. The new offspring 2 chromosome is chosen as the candidate chromosome that has the best prediction accuracy.

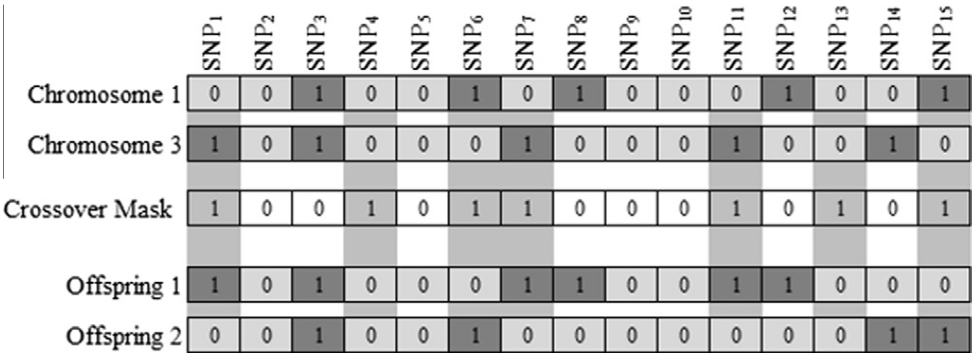
After the modification of tag SNPs number, LOOCV method is used to calculate the fitness value (prediction accuracy) for each chromosome of the new population, and the chromosome with the best fitness value is determined. This process is repeated by $N_G \in \{20, 21, \ldots, 200\}$, where N_G signifies the number of required generations (iterations) used by the algorithm as input. The chromosome with the best fitness value within the set is returned as a result of the generations.

3.2. Optimizing the SVM parameters (C and γ) by the particle swarm optimization

In this study, SVM is used to predict the values of the tagged SNPs using the values of tag SNPs. In SVM classifier, *radial basis function* (*RBF*) is used as the kernel function. The RBF kernel function requires that C and γ should be set. $\gamma$is an important parameter to dominate the generalization ability of SVM by regulating the amplitude of the RBF kernel function and C is a parameter controlling the trade off between maximizing the margin and minimizing the training error [32,33]. It is not known beforehand which C and γ values are best for a given problem; therefore, some kind of parameter researches should be performed [46]. For this purpose, PSO algorithm is used in the study.

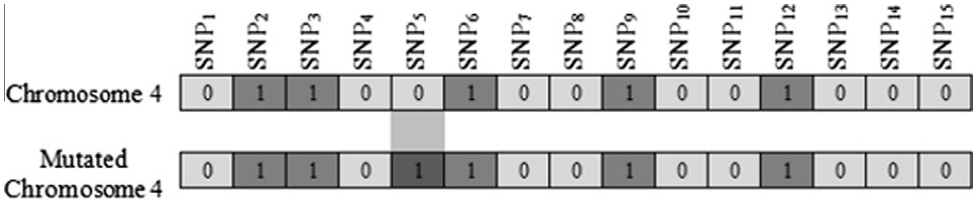
3.2.1. The initial population for SVM parameters

Initial population for SVM parameters is represented by a matrix where the rows represent particles in population and the columns represent SVM parameters, respectively. The PSO algorithm forms a *population matrix* P_{PSO} with two columns and N_p rows. The columns here represent C and γ parameters and the rows



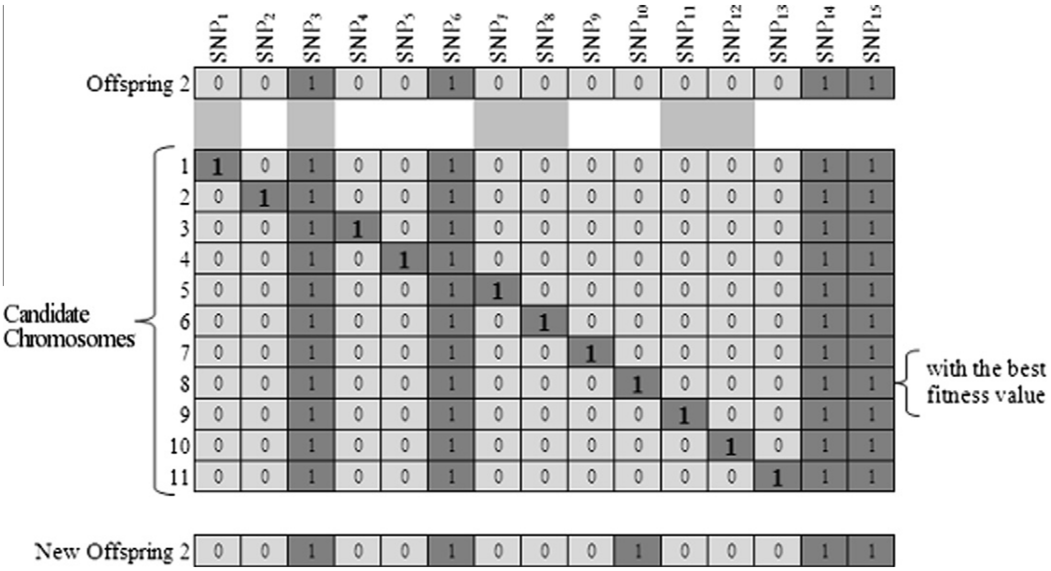
	SNP_1	SNP_2	SNP_3	SNP_4	SNP_5	SNP_6	SNP_7	SNP_8	SNP_9	SNP_{10}	SNP_{11}	SNP_{12}	SNP_{13}	SNP_{14}	SNP_{15}
Chromosome 1	0	0	1	0	0	1	0	1	0	0	0	1	0	0	1
Chromosome 3	1	0	1	0	0	0	1	0	0	0	1	0	0	1	0
Crossover Mask	1	0	0	1	0	1	1	0	0	0	1	0	1	0	1
Offspring 1	1	0	1	0	0	0	1	1	0	0	1	1	0	0	0
Offspring 2	0	0	1	0	0	1	0	0	0	0	0	0	0	1	1

Fig. 3. Uniform crossover operation on chromosomes 1 and 3.



	SNP_1	SNP_2	SNP_3	SNP_4	SNP_5	SNP_6	SNP_7	SNP_8	SNP_9	SNP_{10}	SNP_{11}	SNP_{12}	SNP_{13}	SNP_{14}	SNP_{15}
Chromosome 4	0	1	1	0	0	1	0	0	1	0	0	1	0	0	0
Mutated Chromosome 4	0	1	1	0	1	1	0	0	1	0	0	1	0	0	0

Fig. 4. The mutation operation on SNP_5 for the chromosome 4.



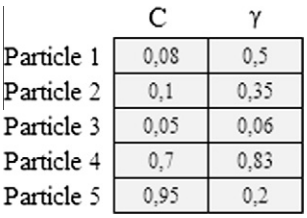
	SNP_1	SNP_2	SNP_3	SNP_4	SNP_5	SNP_6	SNP_7	SNP_8	SNP_9	SNP_{10}	SNP_{11}	SNP_{12}	SNP_{13}	SNP_{14}	SNP_{15}
Offspring 2	0	0	1	0	0	1	0	0	0	0	0	0	0	1	1
1	1	0	1	0	0	1	0	0	0	0	0	0	0	1	1
2	0	1	1	0	0	1	0	0	0	0	0	0	0	1	1
3	0	0	1	1	0	1	0	0	0	0	0	0	0	1	1
4	0	0	1	0	1	1	0	0	0	0	0	0	0	1	1
5	0	0	1	0	0	1	1	0	0	0	0	0	0	1	1
6	0	0	1	0	0	1	0	1	0	0	0	0	0	1	1
7	0	0	1	0	0	1	0	0	1	0	0	0	0	1	1
8	0	0	1	0	0	1	0	0	0	1	0	0	0	1	1
9	0	0	1	0	0	1	0	0	0	0	1	0	0	1	1
10	0	0	1	0	0	1	0	0	0	0	0	1	0	1	1
11	0	0	1	0	0	1	0	0	0	0	0	0	1	1	1
New Offspring 2	0	0	1	0	0	1	0	0	0	1	0	0	0	1	1

Fig. 5. The preparation of candidate chromosomes by replacing 0s to 1s in offspring 2 one at a time.

represent N_p number of particles. The number of rows in particle matrix P_{PSO} equals to that of the population matrix P_{GA}, formed by GA. The entries $p_{ij} \in [0,1]$ of the matrix P_{PSO} represent the values of C and γ parameters for the ith particle. A population matrix is given as an example in Fig. 6. As can be seen in this figure, population matrix consisted of five particles and two parameters (C and γ).

3.2.2. The fitness evaluation

In this study, the fitness function calculated for GA is used as the fitness function of PSO algorithm. The fitness value calculated for each chromosome in population matrix P_{GA} (the population matrix for GA) is also used as the fitness value of the particle that is the equivalent in population matrix P_{PSO} (the population matrix for PSO). For instance, if the calculated fitness value is 0.93 for the chromosome 2 in Fig. 2, this value is also the fitness value of particle 2 in Fig. 6. As can be seen in Eq. (1), the fitness value (prediction accuracy) is obtained as the ratio of the number of accurate predicted SNPs (N_c) to the total number of predicted SNPs (N_a).



	C	γ
Particle 1	0,08	0,5
Particle 2	0,1	0,35
Particle 3	0,05	0,06
Particle 4	0,7	0,83
Particle 5	0,95	0,2

Fig. 6. Particle population matrix included 5 particles. Each particle consists of C and γ parameters.

3.2.3. Finding pbest values and gbest

In each iteration of PSO algorithm, every particle is updated based on "the two best" values. The first one is the best fitness value that the particle finds until that time. Furthermore, this value is kept in the memory for later use and it is called "*pbest*"; that is, the best value of the particle. As for the other, it is the best fitness value acquired by any particle in the population until that time. This value is the global best value in the population and is called "*gbest*".

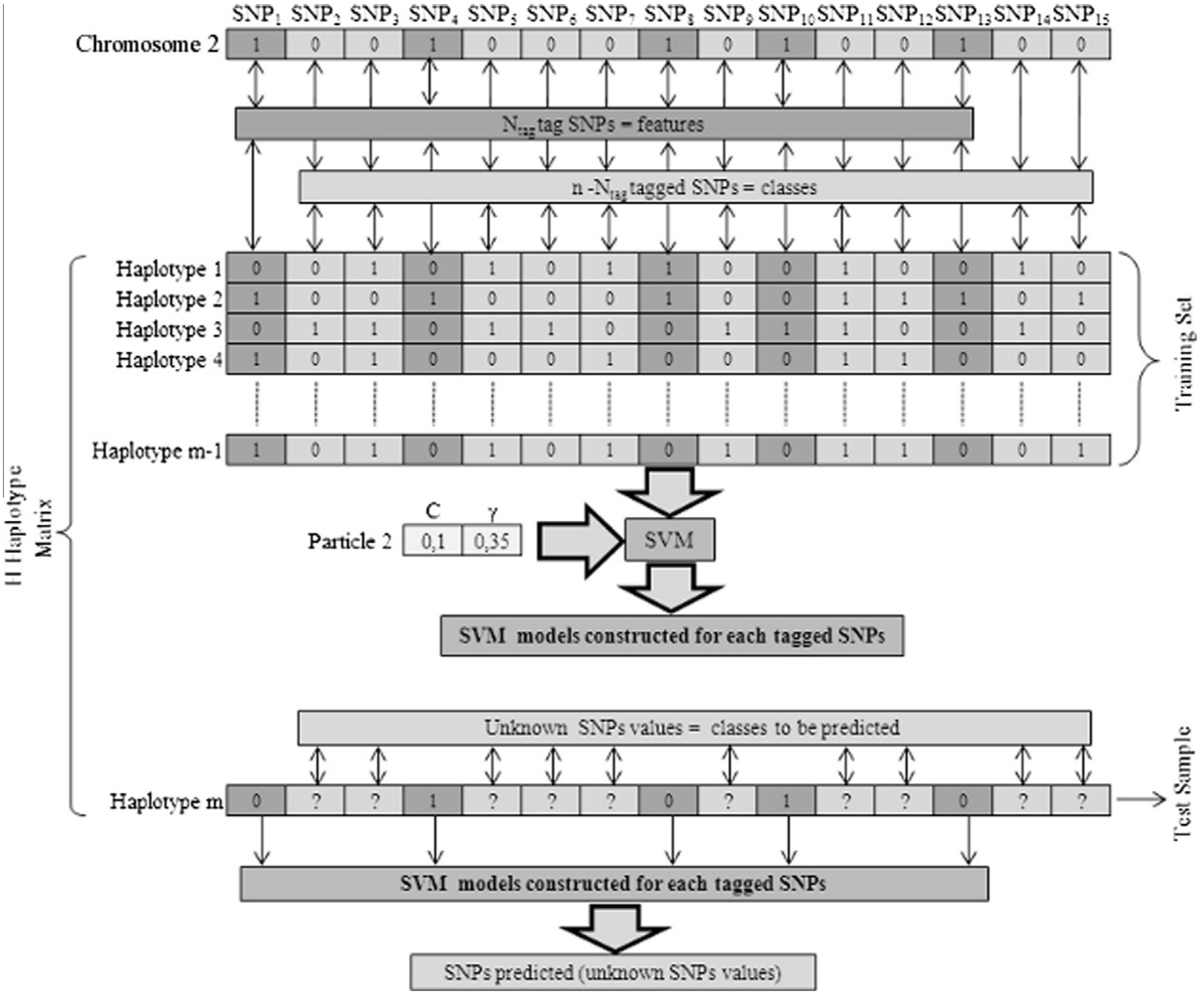
3.2.4. Calculating the particle speeds and updating their locations

The speed of each particle in the population is calculated and their locations are updated according to the best value (pbest) of the particles and the best global value (gbest) found in the previous step. Eqs. (2) and (3) are used respectively to calculate the speed of the particles and to update their locations.

$$v_{ij}^{k+1} = w \times v_{ij}k + c_1 \times r_1 \times (pbest_{ij}^k - x_{ij}^k) + c_2 \times r_2 \times (gbest_j^k - x_{ij}^k) \tag{2}$$

$$X_{ij}^{k+1} = X_{ij}^k + v_{ij}^{k+1} \tag{3}$$

Here $i = 1,2,\ldots,N_p$, $k = 1,2,\ldots,N_G$ and $j = 1,2$ and N_p shows the size of the population; N_G shows the number of iterations and 2 shows the dimension of the problem space (C and γ). v_{ij}^k and x_{ij}^k are the speed and solution (location) of ith particle, respectively. $pbest_{ij}^k$ is the best solution of the ith particle reached until that time and $gbest_j^k$ is the best global solution acquired by any particle in the population until that time. c_1 and c_2 are *acceleration* (*learning*) *factors* and direct movements based on particle's own experience and the experiences of other particles in the population, respectively. In this study, both learning factors c_1 and c_2 are taken as 2 [47]. r_1 and



	SNP_1	SNP_2	SNP_3	SNP_4	SNP_5	SNP_6	SNP_7	SNP_8	SNP_9	SNP_{10}	SNP_{11}	SNP_{12}	SNP_{13}	SNP_{14}	SNP_{15}
Chromosome 2	1	0	0	1	0	0	0	1	0	1	0	0	1	0	0
Haplotype 1	0	0	1	0	1	0	1	1	0	0	1	0	0	1	0
Haplotype 2	1	0	0	1	0	0	0	1	0	0	1	1	1	0	1
Haplotype 3	0	1	1	0	1	1	0	0	1	1	1	0	0	1	0
Haplotype 4	1	0	1	0	0	0	1	0	0	0	1	1	0	0	0
Haplotype m-1	1	0	1	0	1	0	1	0	1	0	1	1	0	0	1
Haplotype m	0	?	?	1	?	?	?	0	?	1	?	?	0	?	?

N_{tag} tag SNPs = features
n -N_{tag} tagged SNPs = classes
Training Set
H Haplotype Matrix
Particle 2
C
γ
0,1
0,35
SVM
SVM models constructed for each tagged SNPs
Unknown SNPs values = classes to be predicted
Test Sample
SVM models constructed for each tagged SNPs
SNPs predicted (unknown SNPs values)

Fig. 7. The process of prediction of the rest of SNPs of the haplotype *m*.

r_2 are random values within the range of [0,1]. *w* is *the inertia weight* and larger inertia weights allow global search, while smaller inertia weights make local search possible. The experiments performed with values 0.8, 1 and 1.2 of inertia weights indicated that the results (prediction accuracies) gained for inertia value 1 were better than those gained for others. Therefore, in this study, the w value is taken as 1 in order to reach as much better solution as possible [48].

3.3. Predicting the rest of SNPs by support vector machine

There are different methods including *correlation-based* [16,17], *entropy-based* [22,27], *k-nearest neighbors-based* [19,26], *STAMPA-based* (selection of tag SNPs to maximize prediction accuracy) [7,9], *Bayesian network-based* [8] and *SVM-based* [10,21] methods that are used to predict the values of the rest of SNPs. Of these methods, SVM is generally preferred in bioinformatics due to its accurate results and high competition with other data mining methods like neural networks [1,10,49]. SVM creates a model based on the SNP values in haplotype given as training set in the first place. Subsequently, it predicts the rest of SNPs values in the haplotype present in the test set through the developed model and tag SNPs resulting from the above-mentioned GA method.

In SVM-based prediction method, every haplotype present in *H* matrix is considered one test set where each tag SNP represents one certain feature and each of the rest SNPs represents one certain class. Fig. 7 gives an example of the prediction process for the rest of SNPs.

Integrated software *Libsvm*, which is designed for support vector classification, is used in the present study [50]. Optimized *C* and γ parameters in PSO algorithm are used with it.

When the fitness value is calculated for a chromosome in the population matrix formed by GA, *C* and γ parameters in the particle that is equivalent to the same row in the particle population matrix formed by PSO are used. For example, to calculate the fitness value of chromosome 2 in Fig. 7, particle 2 in the particle population matrix is used. This value is accepted as the fitness value of chromosome 2 for GA and particle 2 for PSO.

4. Experimental datasets

The following datasets of the *HapMap* project [51] and the related studies were processed in the present study.

ACE (Angiotensin Converting Enzyme) dataset [52]: It includes 22 haplotypes of 11 individuals and 52 bi-allelic SNPs at 24 Kb genomic region on chromosome *17q23*.

ABCB1 (ATP-Binding Cassette, sub-family B) [53]*:* This dataset is related to *P*-glycoprotein and includes 74 Kb of the genome sequence. It contains 494 haplotypes of 247 individuals and 27 bi-allelic SNPs.

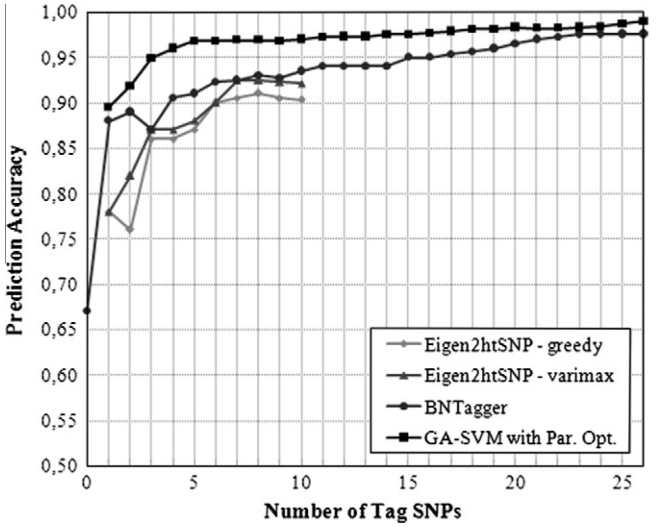


Fig. 8. The prediction accuracy rates of GA–SVM method with parameter optimization and other recent methods for ACE dataset.

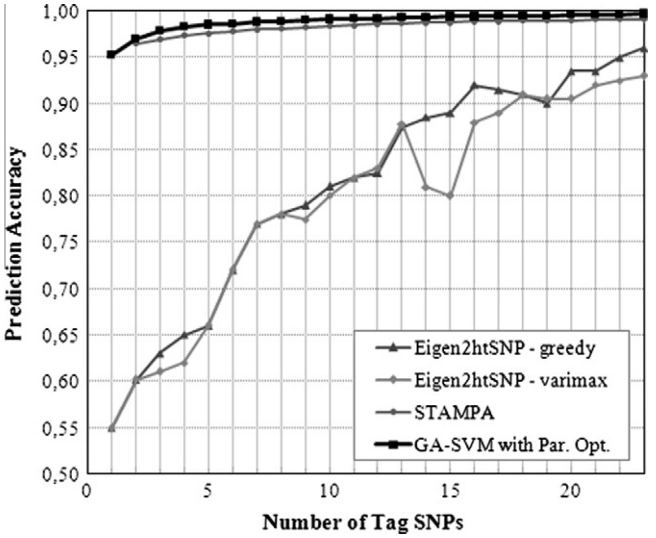


Fig. 9. The prediction accuracy rates of GA–SVM method with parameter optimization and other recent methods for ABCBI dataset.

LPL (The Human Lipoprotein Lipase) dataset [54]: This includes 5.5 Kb of region on chromosome *19q13.22*. It contains 88 SNPs and 142 haplotypes of 71 individuals.

The chromosome 5q31 dataset [30]*:* This is obtained from the 616 Kb region of human chromosome *5q31* from 129 family trios. It contains 103 bi-allelic SNPs; however, only the children population was used in the study.

Two gene regions STEAP and TRPM8. These datasets include 30 CEPH (Utah residents with ancestry from northern and western Europe) family trios from HapMap [51]. The number of bi-allelic SNPs in each region changed between 22 and 101; however, only the parent population was used in the study.

The D9 *dataset of population D* [31]: This dataset includes 180 haplotypes of 30 family trios in Yoruba's population. It contains 49 bi-allelic SNPs.

Three ENCODE regions from Hapmap Regions ENm013, ENr112 and ENr113 from 30 CEPH family trios obtained from HapMap ENCODE project [51] are 500 Kb regions of chromosomes *7q21.13*, *2p16.3* and *4q26*, respectively. The number of bi-allelic SNPs genotyped in each region is 361, 412 and 515. The genotypes corresponding to the parents from each dataset are used in the study.

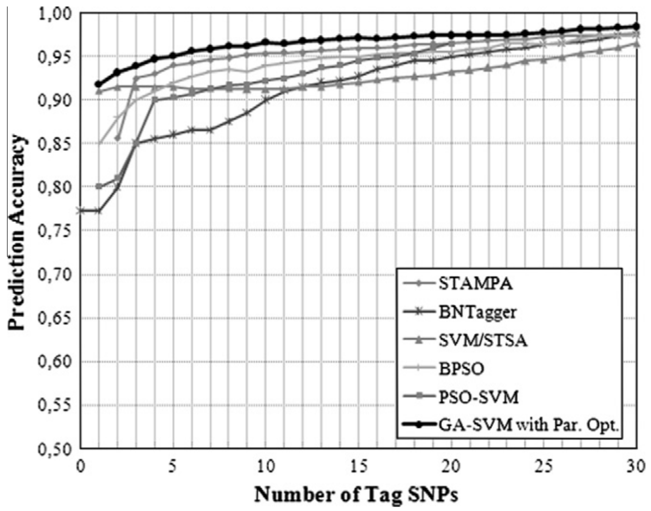


Fig. 10. The prediction accuracy rates of GA–SVM method with parameter optimization and other recent methods for LPL dataset including 88 unique haplotypes.

5. Experimental results

A program was developed in *MATLAB 7.4* software to evaluate the performance of GA–SVM method with parameter optimization in the study, and for this purpose, a SVM program designed by Chang and Lin and named as Libsvm software [50] was used. A target machine was used for the experiments, which had an *Intel Core2Quad@2.83 GHz* processor and 4 GB memory, and run on *Microsoft Windows 7 Professional Edition OS*. A GA was selected for the experiments, which had a generation number of 20, population size of 20, crossover rate of 0.9, and mutation rate of 0.01. In addition, a PSO was used, which had a generation number of 20, population size of 20, both of learning factors of 2 and inertia weight of 1. For both C and γ parameters the search range $[10^{-2}, 10^{2}]$ was used, and it was accepted that $[V_{min}, V_{max}] = [-$ $-10, 10]$. These parameters were determined by trial and error method in the experiments and they were concluded to provide the best results. Thus, they were used for all datasets in this study.

Various experiments for different range values of C and γ parameters were carried out to test the advantages of PSO over exhaustive grid search. In the first of these experiments, while the same fitness values were obtained with exhaustive grid search (the number of grid points is $50^2 = 2500$) and PSO within the range of [0.1, 5] for these parameters, it was seen that PSO worked 250 times faster than the other. In the second experiment, though the same fitness values were obtained with exhaustive grid search (the number of grid points is $100^2 = 10000$) and PSO in the range of [0.1, 10] for these parameters, PSO worked 1000 times faster than the other. These experiments indicated that the PSO used for the optimization of C and γ parameters worked much faster than exhaustive grid search algorithm. Moreover, both algorithms provided the same fitness values.

In addition, the LOOCV method was used at the haplotype level to evaluate the prediction accuracy of GA–SVM method with parameter optimization [7–10,17,21]. However, 10-fold cross validation was used in the local search algorithm used in the adjusting procedure in this method.

The GA–SVM method with parameter optimization suggested for ACE dataset with 52 SNPs was compared to BNTagger [8] and

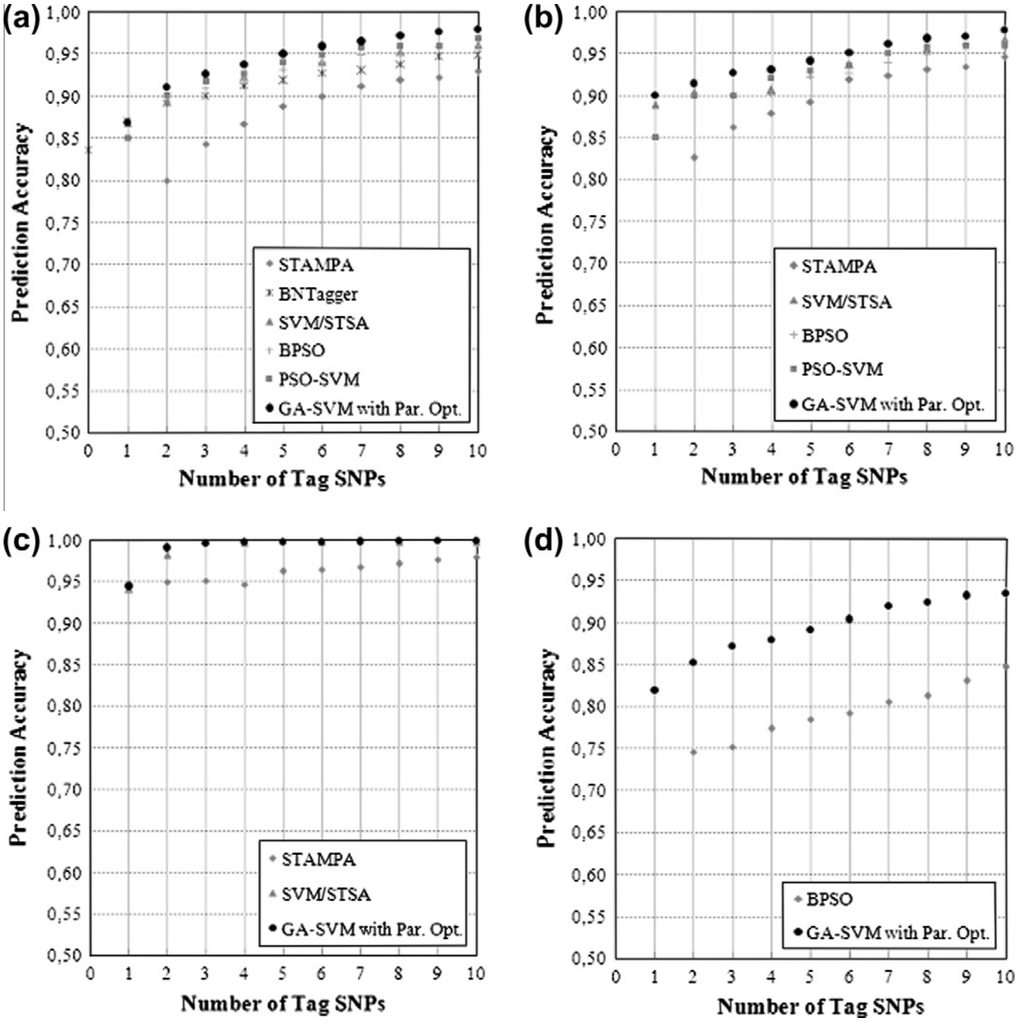


Fig. 11. The prediction accuracy rates for (a) 5q31 (b) TRPM8 (c) STEAP and (d) D9 datasets at different numbers of tag SNPs.

Eigen2htSNP [17] methods. Fig. 8 presents the experimental results for this dataset. Accordingly, the proposed method provided considerable higher performance than other two methods for all tag SNPs numbers and conditions. In addition, prediction accuracy of GA–SVM method with parameter optimization gradually increases in parallel with the tag SNPs number. And, it showed 4.39% and 7.2% more prediction accuracies at the range of 1–10 tag SNPs than BNTagger and Eigen2htSNP methods, respectively, on the average.

The GA–SVM method with parameter optimization suggested for ABCB1 dataset with 27 SNPs was compared to Eigen2htSNP and STAMPA [7] methods in the present study. Accordingly, it obtained 95.3% prediction accuracy for one tag SNP, while Eigen2htSNP methods achieved only 55% of prediction accuracy. In addition, it obtained 97% prediction accuracy for two tag SNPs, while STAMPA method achieved 96.5%. As seen in Fig. 9, the proposed method obtained 27.2% and 0.8% higher prediction accuracy at the range of 2–10 tag SNPs than other two methods, respectively.

The GA–SVM approach with parameter optimization was compared to STAMPA, BNTagger, SVM/STSA [10], BPSO [9] and PSO–SVM [21] methods in terms of prediction accuracy for LPL dataset involving 88 unique haplotypes. Accordingly, as seen in Fig. 10, the suggested method presented higher performance rates than other methods. Especially at the range of 2–20 tag SNPs, it obtained 4.11%, 2.59%, 4.40%, 6.37% and 1.50% higher prediction accuracy rates than PSO–SVM, BPSO, SVM/STSA, BNTagger and STAMPA methods, respectively.

Fig. 11a gives the comparison results on prediction accuracies of GA–SVM method with parameter optimization and PSO–SVM, BPSO, SVM/STSA, BNTagger and STAMPA methods for different tag SNPs numbers in the dataset 5q31 with 103 SNPs. In this figure, the proposed method obtained 1.15%, 0.92%, 2.15%, 2.6% and 5.76% more prediction accuracies at the range of 1–10 tag SNPs compared to PSO–SVM, BPSO, SVM/STSA, BNTagger and STAMPA methods, respectively.

The GA–SVM method with parameter optimization was compared to PSO–SVM, BPSO, SVM/STSA and STAMPA methods in terms of TRPM8 dataset. Accordingly, the suggested method provided higher accuracy than PSO–SVM, BPSO, SVM/STSA and STAMPA methods, as can be seen in Fig. 11b.

The GA–SVM method with parameter optimization obtained higher prediction accuracy rates for STEAP and D9 datasets compared to STAMPA, SVM/STSA and BPSO methods. The comparison results are shown in Fig. 11c and d.

Various experiments were carried out to evaluate the performance of the GA–SVM method with parameter optimization on larger datasets. When choosing the datasets used in these experiments, it was made sure that these datasets were commonly used by the methods compared. Therefore, the datasets ENm013, ENr112 and ENr113, which are also used by the STAMPA, were selected (The datasets with more than 100 SNPs were processed by

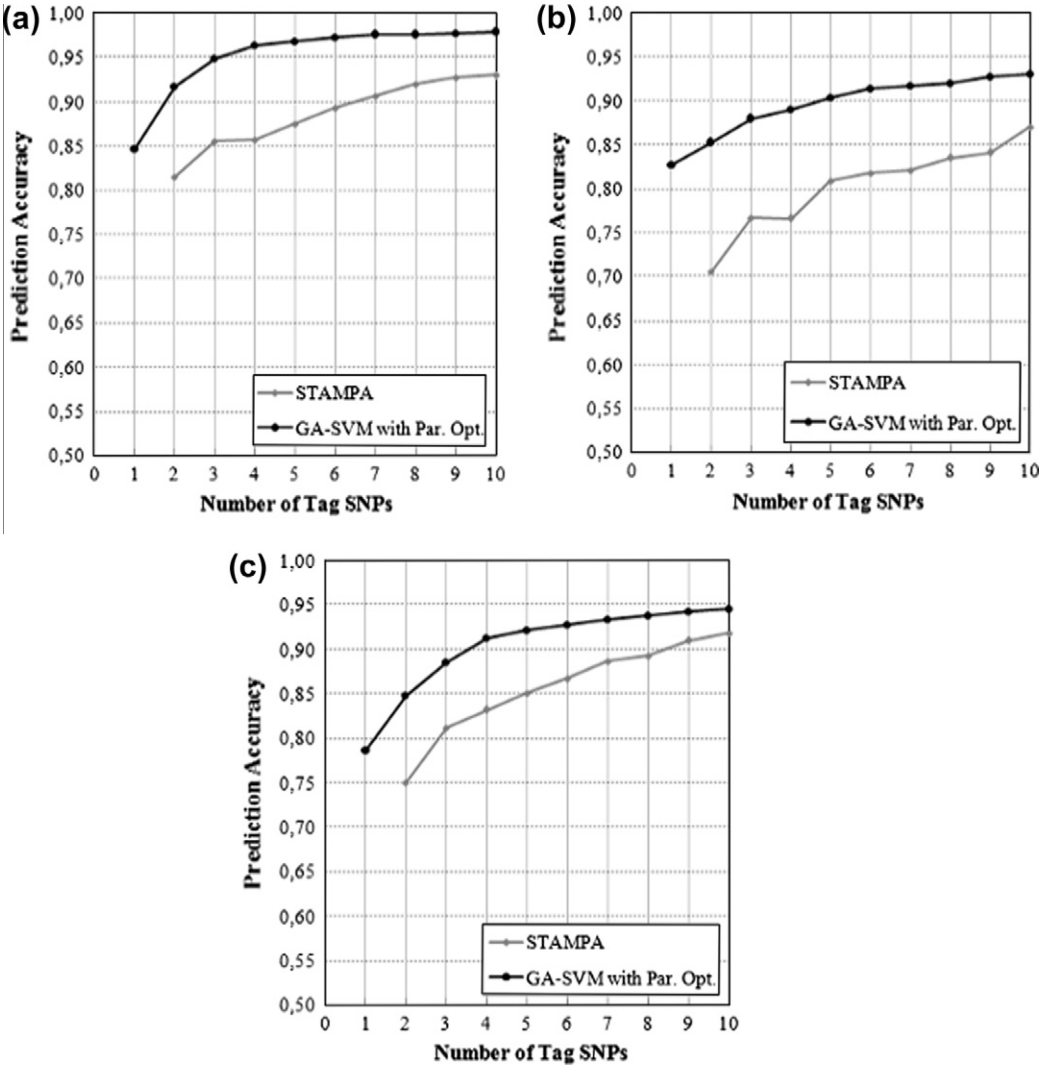


Fig. 12. The prediction accuracy rates for (a) ENm013 (b) ENr112 and (c) ENr113 datasets at different numbers of tag SNPs.

only the STAMPA method). As it is seen in Fig. 12, in the experiments carried out on ENm013, ENr112 and ENr113 datasets, at the range of 2–10 tag SNPs, the GA–SVM method with parameter optimization, in comparison with the STAMPA method, provided higher prediction accuracy of 7.8%, 10% and 5.9% on the average, respectively.

The running times of the proposed method for different numbers of tag SNPs are given in Table 1. The running times of the SVM/STSA method on 5q31, TRPM8 and STEAP datasets were taken from the original study. The running times of the STAMPA method in the range of 2–10 tag SNPs on all datasets were determined by carrying out experiments, but since the time change in this range is very small, it is not shown in the table. For instance, the STAMPA method is able to determine all tag SNPs in the range of 2–10 for ACE, the smallest dataset that we used, in 2 s. On the other hand, the proposed method determines 2 tag SNPs in 2 min and 16 s, while it determines 10 tag SNPs in 6 min and 41 s. The STAMPA method selects all tag SNPs from 2 to 10 for the dataset 5q31 in 7 s, while the SVM/STSA method selects 2 tag SNPs in 5 h and 10 tag SNPs in 24 h. As for the suggested method, it determines 2 tag SNPs in 2 h and 54 min and 10 tag SNPs in 8 h and 30 min. The experiments carried out on ENr113, the largest dataset, indicated that the STAMPA method determined all tag SNPs in the range of 2–10 in approximately 6 min. The suggested method, on the other hand, determines 2 tag SNPs in 7 h and 12 min and 10 tag SNPs in 29 h 19 min. As it is understood from the experiments carried out, the GA–SVM method with parameter optimization works faster than the SVM/STSA method, while it works slower than the STAMPA method. Since the LOOCV method was used to evaluate the prediction accuracy in all three methods, the number of haplotypes affects the running times as the number of SNPs does.

As can be seen in Table 1, for all of the datasets except for the ABCB1 and STEAP, the running time of the GA–SVM method with parameter optimization goes up as the number of tag SNPs increase as well. However, as the number of tag SNPs increases, the running time of the proposed method decreases for the ABCB1 dataset, while it remains almost the same for the STEAP dataset. In the experiments carried out, it was observed that while the number of tag SNPs increases within 1–10 tag SNP range, the running time of the proposed method increased for datasets with more than 40 SNPs, while the running time of the proposed system generally decreased for datasets with less than 40 SNPs.

In this study, two more experiments were carried out to better evaluate the running time and prediction accuracy of the GA–SVM method with parameter optimization proposed and to give an idea about real scenarios in which nothing is known about the patient studied on. In the first one, the PSO applied for parameter optimi-

Table 1
Running times of GA–SVM with parameter optimization and SVM/STSA methods for different datasets at different numbers of tag SNPs.

Datasets (NHap, NSNP)	Methods	The number of tag SNPs									
		1	2	3	4	5	6	7	8	9	10
ACE (22, 52)	GA–SVM with Par. Opt.	1 m 39 s	2 m 16 s	3 m 25 s	4 m 17 s	4 m 58 s	4 m 44 s	5 m 52 s	6 m	6 m 25 s	6 m 41 s
ABCB1 (494, 27)	GA–SVM with Par. Opt.	1 h 15 m	1 h 7 m	58 m 43 s	57 m 23 s	51 m 23 s	53 m 11 s	47 m 54 s	43 m 41 s	40 m 10 s	37 m 30 s
LPL (88, 88)	GA–SVM with Par. Opt.	7 m 35 s	11 m 13 s	15 m 41 s	19 m 23 s	22 m 4 s	26 m 19 s	30 m 58 s	35 m 58 s	38 m 55 s	44 m 24 s
D9 (180, 49)	GA–SVM with Par. Opt.	14 m 8 s	19 m 5 s	23 m 3 s	31 m 23 s	34 m 3 s	42 m 40 s	52 m 57 s	53 m 42 s	1 h 4 m	1 h 17 m
5q31 (258, 103)	SVM/STSA	3 h	5 h	–	11 h	–	16 h	–	18 h	–	24 h
	GA–SVM with Par. Opt.	2 h 32 m	2 h 54 m	3 h 45 m	4 h 5 m	4 h 11 m	5 h 21 m	6 h 10 m	6 h 55 m	7 h 21 m	8 h 30 m
TRPM8 (120, 101)	SVM/STSA	1 h	2 h	–	5 h	–	9 h	–	16 h	–	23 h
	GA–SVM with Par. Opt.	19 m 25 s	23 m 31 s	29 m 31 s	28 m	37 m 27 s	33 m 47 s	47 m 24 s	46 m 13 s	48 m 8 s	53 m 34 s
STEAP (120, 22)	SVM/STSA	14 m	27 m	–	1 h	–	2 h	–	3 h	–	4 h
	GA–SVM with Par. Opt.	2 m 35 s	2 m 30 s	2 m 44 s	2 m 37 s	2 m 58 s	2 m 58 s	2 m 57 s	2 m 52 s	2 m 43 s	2 m 46 s
ENm013 (120, 361)	GA–SVM with Par. Opt.	2 h 40 m	3 h 32 m	4 h 27 m	5 h 43 m	6 h 12 m	6 h 54 m	7 h 45 m	10 h 32 m	10 h 43 m	15 h 27 m
ENr112 (120, 412)	GA–SVM with Par. Opt.	3 h 4 m	4 h 12 m	5 h 35 m	6 h 47 m	10 h 32 m	11 h 20 m	13 h	16 h 55 m	20 h 26 m	24 h 53 m
ENr113 (120, 515)	GA–SVM with Par. Opt.	5 h 18 m	7 h 12 m	10 h 21 m	10 h 54 m	13 h 52 m	17 h 15 m	19 h 59 m	22 h 21 m	25 h 37 m	29 h 19 m

NHap = The number of haplotypes, NSNP = The number of SNPs, h = hour, m = minute, s = second.

Table 2
Prediction accuracies of the GA–SVM method with parameter optimization and the methods used in the two experiments for different datasets at different numbers of tag SNPs (to make evaluations at the individual level, the LPL dataset that consists of 142 haplotypes was used in the experiments).

Datasets (NHap, NSNP)	Methods	The number of tag SNPs									
		1	2	3	4	5	6	7	8	9	10
ACE (22, 52)	GA–SVM with LK. (H)	88.6	90.9	93.5	94.5	95.4	95.8	95.9	96.1	96.2	96.4
	GA–SVM with PO. (I)	89.0	91.5	92.9	93.3	93.5	93.6	93.5	93.7	93.7	93.4
	GA–SVM with PO. (H)	89.5	91.8	94.9	96.0	96.8	96.8	96.9	96.9	96.8	97.0
ABCB1 (494, 27)	GA–SVM with LK. (H)	95.3	97.0	97.8	98.3	98.5	98.6	98.7	98.8	99.0	99.1
	GA–SVM with PO. (I)	95.3	97.0	97.7	98.3	98.5	98.6	98.8	98.9	99.0	99.1
	GA–SVM with PO. (H)	95.3	97.0	97.8	98.3	98.6	98.6	98.8	98.9	99.0	99.1
LPL (142, 88)	GA–SVM with LK. (H)	93.1	94.5	95.5	95.7	96.2	96.6	97.1	97.2	97.4	97.5
	GA–SVM with PO. (I)	92.6	93.9	94.7	95.4	95.7	96.0	96.2	96.3	96.5	96.5
	GA–SVM with PO. (H)	93.1	94.7	95.8	96.2	96.8	97.2	97.6	97.8	97.8	97.9
D9 (180, 49)	GA–SVM with LK. (H)	81.5	84.7	86.3	87.5	88.7	89.5	90.4	91.2	92.4	92.8
	GA–SVM with PO. (I)	81.5	84.8	86.3	87.6	88.8	90.1	90.7	91.3	92.3	92.5
	GA–SVM with PO. (H)	81.9	85.2	87.1	87.9	89.1	90.4	92.0	92.4	93.2	93.4
5q31 (258, 103)	GA–SVM with LK. (H)	86.7	89.6	91.8	93.2	94.7	95.7	96.3	96.7	97.1	97.3
	GA–SVM with PO. (I)	86.6	89.5	92.3	93.7	94.8	95.6	96.1	96.5	96.6	96.8
	GA–SVM with PO. (H)	86.8	91.1	92.6	93.8	95.0	95.9	96.5	97.2	97.6	97.9
TRPM8 (120, 101)	GA–SVM with LK. (H)	90.0	90.6	92.2	93.0	93.3	94.5	95.3	96.2	96.6	97.2
	GA–SVM with PO. (I)	89.8	90.9	92.4	92.7	93.4	94.3	95.5	96.3	96.5	97.3
	GA–SVM with PO. (H)	90.0	91.4	92.8	93.1	94.1	95.1	96.1	96.9	97.1	97.8
STEAP (120, 22)	GA–SVM with LK. (H)	94.4	98.3	99.5	99.7	99.7	99.8	99.8	99.9	99.9	99.9
	GA–SVM with PO. (I)	94.5	98.6	99.5	99.7	99.8	99.8	99.8	99.8	99.9	99.9
	GA–SVM with PO. (H)	94.5	99.1	99.6	99.8	99.8	99.8	99.9	99.9	99.9	99.9
ENm013 (120, 361)	GA–SVM with LK. (H)	86.1	92.5	94.1	95.4	96.4	96.8	97.3	97.4	97.6	97.8
	GA–SVM with PO. (I)	84.1	91.5	94.2	95.1	96.3	96.7	97.2	97.5	97.6	97.8
	GA–SVM with PO. (H)	84.6	91.7	94.8	96.3	96.8	97.2	97.6	97.6	97.7	97.9
ENr112 (120, 412)	GA–SVM with LK. (H)	82.6	85.1	87.0	88.6	90.1	90.7	91.6	91.9	92.7	93.0
	GA–SVM with PO. (I)	82.4	85.2	87.4	88.5	90.0	90.9	91.4	91.7	92.2	92.5
	GA–SVM with PO. (H)	82.7	85.3	88.0	89.0	90.3	91.4	91.7	92.0	92.7	93.0
ENr113 (120, 515)	GA–SVM with LK. (H)	78.5	84.5	88.2	91.0	91.7	92.5	93.0	93.4	93.8	94.1
	GA–SVM with PO. (I)	78.2	84.4	88.3	90.7	91.5	92.3	92.9	93.4	93.9	94.0
	GA–SVM with PO. (H)	78.6	84.8	88.5	91.2	92.1	92.7	93.3	93.8	94.2	94.5

NHap = The number of haplotypes, NSNP = The number of SNPs, GA–SVM with LK. (H) = The GA–SVM method, in which the LOOCV method implemented at the haplotype level and linear kernel function are used, GA–SVM with PO. (I) = The GA–SVM method with parameter optimization, in which the LOOCV method implemented at the individual level is used, GA–SVM with PO. (H) = The GA–SVM method with parameter optimization, in which the LOOCV method implemented at the haplotype level is used.

zation in the proposed method was not used and Linear kernel function was used instead of RBF kernel function in the SVM classifier (GA–SVM with LK. (H)). In the second, to evaluate the prediction accuracy of the GA–SVM method with parameter optimization, the LOOCV method applied at individual level instead of the LOOCV method applied at the haplotype level was

Table 3
Running times of the GA–SVM method with parameter optimization and the methods used in the two experiments for different datasets at different numbers of tag SNPs (To make evaluations at the individual level, the LPL dataset that consists of 142 haplotypes was used in the experiments).

Datasets (NHap, NSNP)	Methods	The number of tag SNPs									
		1	2	3	4	5	6	7	8	9	10
ACE (22, 52)	GA–SVM with LK. (H)	1 m 35 s	2 m 36 s	2 m 45 s	3 m 3 s	3 m 47 s	4 m 30 s	4 m 31 s	5 m 44 s	5 m 25 s	5 m 35 s
	GA–SVM with PO. (I)	1 m 48 s	2 m 8 s	2 m 46 s	3 m 27 s	4 m 10 s	5 m 24 s	5 m 10 s	6 m 20 s	5 m 41 s	6 m 4 s
	GA–SVM with PO. (H)	1 m 39 s	2 m 16 s	3 m 25 s	4 m 17 s	4 m 58 s	4 m 44 s	5 m 52 s	6 m	6 m 25 s	6 m 41 s
ABCB1 (494, 27)	GA–SVM with LK. (H)	38 m 29 s	37 m 59 s	34 m 52 s	32 m 4 s	33 m 51 s	33 m 36 s	34 m 9 s	30 m 31 s	30 m 47 s	27 m 10 s
	GA–SVM with PO. (I)	21 m 33 s	20 m 48 s	20 m 38 s	21 m 27 s	20 m 35 s	22 m 14 s	19 m 5 s	18 m 8 s	20 m 47 s	15 m 39 s
	GA–SVM with PO. (H)	1 h 15 m	1 h 7 m	58 m 43 s	57 m 23 s	51 m 23 s	53 m 11 s	47 m 54 s	43 m 41 s	40 m 10 s	37 m 30 s
LPL (142, 88)	GA–SVM with LK. (H)	9 m 55 s	16 m 10 s	16 m 47 s	20 m 8 s	23 m 2 s	27 m 25 s	33 m 35 s	33 m 55 s	36 m 12 s	41 m 32 s
	GA–SVM with PO. (I)	8 m 13 s	16 m 28 s	20 m 8 s	27 m 44 s	30 m 59 s	34 m 49 s	35 m 48 s	38 m 43 s	48 m 24 s	50 m 50 s
	GA–SVM with PO. (H)	19 m 9 s	21 m 43 s	29 m 59 s	34 m 26 s	38 m 20 s	40 m 22 s	50 m 14 s	55 m 27 s	1 h 7 m	1 h 9 m
D9 (180, 49)	GA–SVM with LK. (H)	10 m 18 s	12 m	14 m 22 s	19 m 21 s	21 m 59 s	28 m 47 s	32 m 58 s	40 m 12 s	46 m 57 s	49 m 54 s
	GA–SVM with PO. (I)	6 m 52 s	9 m 54 s	13 m 57 s	16 m 51 s	24 m 9 s	31 m 17 s	35 m 13 s	42 m	41 m 13 s	59 m 24 s
	GA–SVM with PO. (H)	14 m 8 s	19 m 5 s	23 m 3 s	31 m 23 s	34 m 3 s	42 m 40 s	52 m 57 s	53 m 42 s	1 h 4 m	1 h 17 m
5q31 (258, 103)	GA–SVM with LK. (H)	1 h 21 m	1 h 38 m	2 h 5 m	2 h 16 m	2 h 32 m	3 h 1 m	3 h 30 m	4 h 7 m	4 h 41 m	5 h 8 m
	GA–SVM with PO. (I)	1 h 7 m	1 h 39 m	2 h 20 m	2 h 47 m	3 h 19 m	3 h 50 m	4 h 59 m	5 h 24 m	6 h 15 m	6 h 17 m
	GA–SVM with PO. (H)	2 h 32 m	2 h 54 m	3 h 45 m	4 h 5 m	4 h 11 m	5 h 21 m	6 h 10 m	6 h 55 m	7 h 21 m	8 h 30 m
TRPM8 (120, 101)	GA–SVM with LK. (H)	13 m 23 s	15 m 22 s	17 m 22 s	21 m 11 s	23 m 23 s	24 m 55 s	29 m 37 s	28 m 32 s	27 m	36 m 7 s
	GA–SVM with PO. (I)	10 m 47 s	17 m 11 s	19 m 9 s	26 m 36 s	33 m 50 s	32 m 38 s	36 m 55 s	39 m 42 s	44 m 8 s	48 m 33 s
	GA–SVM with PO. (H)	19 m 25 s	23 m 31 s	29 m 31 s	28 m	37 m 27 s	33 m 47 s	47 m 24 s	46 m 13 s	48 m 8 s	53 m 34 s
STEAP (120, 22)	GA–SVM with LK. (H)	1 m 46 s	1 m 51 s	1 m 49 s	1 m 55 s	1 m 54 s	1 m 42 s	1 m 42 s	1 m 45 s	1 m 28 s	1 m 35 s
	GA–SVM with PO. (I)	1 m 29 s	1 m 30 s	1 m 36 s	1 m 43 s	1 m 36 s	1 m 43 s	1 m 47 s	1 m 40 s	1 m 53 s	1 m 39 s
	GA–SVM with PO. (H)	2 m 35 s	2 m 30 s	2 m 44 s	2 m 37 s	2 m 58 s	2 m 58 s	2 m 57 s	2 m 52 s	2 m 43 s	2 m 46 s
ENm013 (120, 361)	GA–SVM with LK. (H)	2 h 3 m	2 h 52 m	3 h 34 m	4 h 34 m	5 h 59 m	5 h 33 m	5 h 44 m	7 h 28 m	8 h 34 m	13 h 26 m
	GA–SVM with PO. (I)	2 h 37 m	2 h 50 m	3 h 32 m	4 h	6 h 33 m	5 h 58 m	8 h 40 m	9 h 1 m	12 h 5 m	15 h 47 m
	GA–SVM with PO. (H)	2 h 40 m	3 h 32 m	4 h 27 m	5 h 43 m	6 h 12 m	6 h 54 m	7 h 45 m	10 h 32 m	10 h 43 m	15 h 27 m
ENr112 (120, 412)	GA–SVM with LK. (H)	2 h 30 m	3 h 9 m	4 h 29 m	5 h 12 m	7 h 26 m	9 h 21 m	10 h 29 m	11 h 33 m	14 h 51 m	20 h 35 m
	GA–SVM with PO. (I)	2 h 51 m	3 h 29 m	4 h 53 m	6 h 12 m	8 h 56 m	10 h 29 m	12 h 31 m	14 h 43 m	18 h 27 m	23 h 41 m
	GA–SVM with PO. (H)	3 h 4 m	4 h 12 m	5 h 35 m	6 h 47 m	10 h 32 m	11 h 20 m	13 h 2 m	16 h 55 m	20 h 26 m	24 h 53 m
ENr113 (120, 515)	GA–SVM with LK. (H)	4 h 16 m	5 h 1 m	7 h 40 m	8 h 53 m	9 h 50 m	13 h 53 m	16 h	17 h 50 m	22 h 16 m	28 h 16 m
	GA–SVM with PO. (I)	4 h 58 m	6 h 31 m	9 h 25 m	10 h 15 m	11 h 47 m	15 h 25 m	17 h 49 m	20 h 11 m	24 h 7 m	28 h 49 m
	GA–SVM with PO. (H)	5 h 18 m	7 h 12 m	10 h 21 m	10 h 54 m	13 h 52 m	17 h 15 m	19 h 59 m	22 h 21 m	25 h 37 m	29 h 19 m

NHap = The number of haplotypes, NSNP = The number of SNPs, h = hour, m = minute, s = second, GA–SVM with LK. (H) = The GA–SVM method, in which the LOOCV method implemented at the haplotype level and linear kernel function are used, GA–SVM with PO. (I) = The GA–SVM method with parameter optimization, in which the LOOCV method implemented at the individual level is used, GA–SVM with PO. (H) = The GA–SVM method with parameter optimization, in which the LOOCV method implemented at the haplotype level is used.

used (GA–SVM with PO. (I)). Prediction accuracies of the GA–SVM method with parameter optimization and the methods used in the two experiments are presented in Table 2 for different datasets at different numbers of tag SNPs. As it is clear from Table 2, the proposed GA–SVM method with parameter optimization provided better prediction accuracies for different datasets at the range of 1–10 tag SNPs in comparison with the other two methods. For example, in comparison with the methods used in the first and second experiments, the method suggested for ACE, which is the smallest dataset, provided 1.01% and 2.53% higher prediction accuracies the average, respectively. Similarly, for ENr113, the largest dataset, the proposed method provided 0.30% and 0.41% higher prediction accuracies on the average respectively in comparison with the other two methods used in the experiments.

In the second experiment, it was seen that the GA–SVM method with parameter optimization in which LOOCV method was implemented at individual level rather than at haplotype exhibited higher prediction accuracy compared to the methods in other publications. For example, it was demonstrated that the GA–SVM method with parameter optimization in which LOOCV method was implemented at individual level for ACE dataset at the range of 1–10 tag SNPs showed 1.81% and 4.71% higher prediction accuracy on average compared to BNTagger and Eigen2htSNP methods, respectively. Similarly, it was demonstrated that the method used in the second experiment for ENr113 dataset at the range of 1–10 tag SNPs exhibited 4.16% higher prediction accuracy on average compared to the STAMPA method.

Running times of the GA–SVM method with parameter optimization and the methods used in the two experiments are presented in Table 3 for different datasets at different numbers of tag SNPs. As can be seen in Table 3, the proposed GA–SVM method with parameter optimization for different datasets determines the tag SNPs in the range of 1–10 more slowly in comparison with the methods used in the first and second experiments.

6. Discussion and conclusion

A new method is developed and suggested in the present study to select the tag SNPs, and accordingly, predict the rest of SNPs in a gene. This information is commonly used to identify the genetic variants related to complicated disorders. The method suggested in the study is named as GA–SVM method with parameter optimization, and it benefits from SVM and GA to predict SNPs and select tag SNPs, respectively. In addition, PSO is used to optimize C and γ parameters of support vector machine. In the study, GA–SVM method with parameter optimization is experimentally compared to other common methods to test its prediction accuracy on datasets with different sizes.

In this study, the GA–SVM method with parameter optimization was applied to tag SNP selection problem, and two different search algorithms were used to select both tag SNPs and SVM parameters. Instead of the LOOCV method, 10-fold cross validation was used to determine the chromosome with the best fitness value among the candidate chromosomes in the local search algorithm used in the adjusting procedure of tag SNP selection. Thus, the speed of this search algorithm was increased ten times. Furthermore, the PSO was preferred to exhaustive grid search because it was observed that the former algorithm worked approximately 250 times faster than the latter even for the narrow range of [0.1,5] (the number of grid points is $50^2 = 2500$) for C and γ parameters.

As can be seen in the results of the experiments carried out on the datasets used, as the number of tag SNPs increases, the prediction accuracy of the suggested method regularly increases, too. Moreover, as the number of SNPs increases, the running time of the GA–SVM method with parameter optimization increases as it is the case with the STAMPA and SVM/STSA methods.

Consequently, experiment results prove that the method suggested in the study has considerably higher prediction accuracy for all possible number of tag SNPs than other methods.

Acknowledgments

This study is supported by Selçuk University Scientific Research Projects Coordinatorship, Konya, Turkey. The authors would like to thank the editors and anonymous reviewers of this manuscript for their very helpful suggestions.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.jbi.2012.12.002.

References

[1] Zhang YQ, Rajapakse JC. Machine learning in bioinformatics. New York: Wiley; 2008. p. 367–412.
[2] Iles MM. What can genome-wide association studies tell us about the genetics of common disease. PLoS Genet 2008;4(2):e33.
[3] Kruglyak L, Nickerson DA. Variation is the spice of life. Nat Genet 2001;27(3):234–6.
[4] Halldorsson BV, Bafna V, Edwards N, Lippert R, Yooseph S, Istrail S. A survey of computational methods for determining haplotyes. Lect Notes Comput Sci 2004;2983:26–47.
[5] Crawford D, Nickerson DA. Definition and clinical importance of haplotypes. Ann Rev Med 2005;56(1):303–20.
[6] İlhan İ, Göktepe YE, Kahramanlı Ş. Tag SNP selection using GA–SVM approach. In: Proceedings of the IADIS European conference on data mining 2011, Rome, Italy; 2011. p. 27–34.
[7] Halperin E, Kimmel G, Shamir R. Tag SNP selection in genotype data for maximizing SNP prediction accuracy. Bioinformatics 2005;21(suppl. 1):195–203.
[8] Lee PH, Shatkay H. BNTagger: improved tagging SNP selection using Bayesian networks. Bioinformatics 2006;22(14):211–9.
[9] Yang CY, Hou CH, Chuang LY. Improved tag SNP selection using binary particle swarm optimization. In: IEEE congress on evolutionary computation (CEC 2008); 2008. p. 854–60.
[10] He J, Zelikovsky A. Informative SNP selection methods based on SNP prediction. IEEE Trans Nanobioscience 2007;6(1):60–7.
[11] Patil N, Berno AJ, Hinds DA, Barrett WA, Doshi JM, Hacker CR, et al. Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. Science 2001;294(5547):1719–23.
[12] Zhang K, Deng M, Chen T, Waterman MS, Sun F. A dynamic programming algorithm for haplotype block partitioning. Proc Natl Acad Sci USA 2002;99(11):7335–9.
[13] Zhang K, Sun F, Waterman MS, Chen T. Haplotype block partition with limited resources and applications to human chromosome 21 haplotype data. Am J Hum Genet 2003;73(1):63–73.
[14] Ke X, Cardon LR. Efficient selective screening of haplotype tag SNPs. Bioinformatics 2003;19(2):287–8.
[15] Zhang K, Qin Z, Liu J, Chen T, Waterman M, Sun F. Haplotype block partitioning and tag SNP selection using genotype data and their applications to association studies. Genome Res 2004;14(5):908–16.
[16] Phuong TM, Lin Z, Altman RB. Choosing SNPs using feature selection. In: Proceedings of the 2005 IEEE computational systems bioinformatics conference (CSB'05), Standford, CA; 2005. p. 301–9.
[17] Lin Z, Altman R. Finding haplotype tagging SNP by use of principle component analysis. Am J Hum Genet 2004;75(5):850–61.
[18] Bafna V, Halldorsson BV, Schwrtz R, Clark A, Istrail S. Haplotypes and informative SNP selection algorithms: don't block out information. In: Proceedings of the seventh annual international conference on research in computational molecular biology (RECOMB 03), Berlin, Germany; 2003. p. 19–27.
[19] Halldorsson BV, Bafna V, Lippert R, Schwartz R, De La Vega FM, Clark AG, et al. Optimal haplotype block-free selection of tagging SNPs for genome-wide association studies. Genome Res 2004;14(1):1633–40.
[20] Bertolazzi P, Felici G, Festa P. Logic based methods for SNPs tagging and reconstruction. Comput Operat Res 2009;37(8):1419–26.
[21] Lin MH, Leu CL. A hybrid PSO–SVM approach for haplotype tagging SNP selection problem. Int J Comput Sci Info Secur 2010;8(6):60–5.
[22] Mahdevar G, Zahiri J, Sadeghi M, Dalini AN, Ahrabian H. Tag SNP selection via a genetic algorithm. J Biomed Inform 2010;43(5):800–4.
[23] Ao SI, Yip K, Ng M, Cheung D, Fong PY, Melhado I, et al. CLUSTAG: hierarchical clustering and graph methods for selecting tag SNPs. Bioinformatics 2004;21(8):1735–6.
[24] Avi-Itzhak HI, Su X, De La Vega FM. Selection of minimum subsets of single nucleotide polymorphisms to capture haplotype block diversity. Pac Symp Biocomput 2003;8:466–77.
[25] Carlson CS, Michael AE, Mark JR, Qian Y, Leonid K, Deborah AN. Selecting a maximally informative set of single nucleotide polymorphisms for association analyses using linkage disequilibrium. Am J Hum Genet 2004;74(1):106–20.

[26] Chuang LY, Hou CH, Yang CY. A novel prediction method for tag SNP selection using genetic algorithm based on KNN. Int J Chem Biol Eng 2010;3(1):12–7.

[27] Hampe J, Schreiber S, Krawczak M. Entropy-based SNP selection for genetic association studies. Hum Genet 2003;114(1):36–43.

[28] Hao K. Genome-wide selection of tag SNPs using multiple marker correlation. Bioinformatics 2007;23(23):3178–84.

[29] Liu G, Wang Y, Wong L. FastTagger: an efficient algorithm for genome-wide tag SNP selection using multi-marker linkage disequilibrium. BMC Bioinformatics 2010;11:66. 11.

[30] Daly MJ, Rioux JD, Schaffner SF, Hudson TJ, Lander ES. High-resolution haplotype structure in the human genome. Nat Genet 2001;29(2):229–32.

[31] Gabriel SB, Schaffner SF, Nguyen H, Moore JM, Roy J, Blumenstiel B, et al. The structure of haplotype blocks in the human genome. Science 2002;296(5576):2225–9.

[32] Vapnik VN. Statistical learning theory. New York: Wiley; 1998.

[33] Cores C, Vapnik VN. Support vector networks. Mach Learn 1995;20(3): 273–97.

[34] Holland JH. Adaptation in natural and artificial systems. Ann Arbor: The University of Michigan Press; 1975.

[35] Goldberg DE. Genetic algorithms in search, optimization, and machine learning. New York: Addison Wesley; 1989.

[36] Kumamoto A, Utani A, Yamamoto H. Advanced Particle Swarm Optimization For Computing Plural Acceptable Solutions. Int J Innovative Comput Info Control 2009;5(11B):4383–92.

[37] Sağ T, Cunkaş M. A tool for multiobjective evolutionary algorithms. Adv Eng Softw 2009;40(9):902–12.

[38] Guo Y, Cao X, Yin H, Tang Z. Coevolutionary optimization algorithm with dynamic sub-population size. Int J Innovative Comput Info Control 2007;3(2):435–48.

[39] Zou S, Huang Y, Wang Y, Wang J, Zhou C. SVM learning from imbalanced data by GA sampling for protein domain predicting. In: The 9th international conference for young computer scientists; 2008. p. 982–7.

[40] Arlot S, Celisse A. A survey of cross-validation procedures for model selection. Stat Surveys 2010;4:40–79.

[41] Angeline PJ. Evolution revolution: an introduction to the special track on genetic and evolutionary programming. IEEE Expert Intell Syst Appl 1995;10(3):6–10.

[42] Goldberg DE, Deb K. A comparative analysis of selection schemes used in genetic algorithms. Found Genet Algorithms 1991;1:69–93.

[43] Sywerda G. Uniform crossover in genetic algorithms. In: Proceedings of the 3rd international conference on genetic algorithms, Los Altos, CA; 1989. p. 2–9.

[44] Prügel-Bennett A. The mixing rate of different crossover operators. Found Genet Algorithms 2001;6:261–74.

[45] Lin WY, Lee WY, Hong TP. Adapting crossover and mutation rates in genetic algorithms. J Info Sci Eng 2003;19(5):889–903.

[46] Hsu CW, Chang CC, Lin CJ. A practical guide to support vector classification. Bioinformatics 2010;1(1):1–16.

[47] Kennedy J, Eberhart R. Particle swarm optimization. In: IEEE international conference on neural networks, Path, Australia; 1995. p. 1942–8.

[48] Song Y, Eberhart R. A modified particle swarm optimizer. IEEE World Congr Comput Intell 1998;1998:63–7.

[49] Song X, Chen W, Jiang B. Sample reducing method in support vector machine based on K-closest sub-clusters. Int J Innovative Comput Info Control 2008;4(7):1751–60.

[50] Chang CC, Lin CJ. LIBSVM: a library for support vector machines; 2001. Software: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

[51] International HapMap Consortium. The international HapMap project. Nature 2003;426(6968):789–96.

[52] Rieder MJ, Taylor SL, Clark AG, Nickerson DA. Sequence variation in the human angiotensin converting enzyme. Nat Genet 1999;22(1):59–62.

[53] Kroetz DL, Pauli-Magnus C, Hodges LM, Huang CC, Kawamoto M, Johns SJ, et al. Sequence diversity and haplotype structure in the human ABCB1 (MDR1, multi drug resistance transporter) gene. Pharmacogenetics 2003;13(8):481–94.

[54] Clark A, Weiss K, Nickerson D, Taylor S, Buchanan A, Stengard J, et al. Haplotype structure and population genetic inferences from nucleotide-sequence variation in human lipoprotein lipase. Hum Genet 1998;63(2): 595–612.