International Journal of Innovative
Computing, Information and Control
Volume **9**, Number **2**, February **2013**

ICIC International ©2013 ISSN 1349-4198
pp. 525–541

A GENETIC ALGORITHM – SUPPORT VECTOR MACHINE METHOD FOR SELECTING TAG SINGLE NUCLEOTIDE POLYMORPHISMS

Ilhan Ilhan[1], Yunus Emre Goktepe[1] and Sirzat Kahramanli[2]

[1]Department of Computer Engineering
Selcuk University
Konya, Turkey
{ ilhan; yegoktepe }@selcuk.edu.tr

[2]Department of Computer Engineering
Mevlana University
Konya, Turkey
skahramanli@mevlana.edu.tr

Received November 2011; revised April 2012

Abstract. *Obtaining the association between complex diseases and single nucleotide polymorphisms (SNPs) is one of the most important medical problems. Although obtaining the full set of SNPs is a very challenging issue, there are subsets of tag SNPs, each of which allows predicting the rest of SNPs with enough accuracy. Here, the problem is to obtain such a subset of tag SNPs that makes it possible to predict the rest of SNPs with maximal possible accuracy. However, the methods developed for this aim cannot reach the accuracy level enough for practical applications. In this study, a new approach using the Genetic Algorithm for selecting the tag SNPs and Support Vector Machine for predicting the values of the rest of SNPs is proposed. The results of the experiments performed on a number of datasets demonstrate that the proposed method can predict the values of the rest of SNPs with a significantly better accuracy than other methods with the same purpose.*

Keywords: Single nucleotide polymorphisms, Tag SNPs, Genetic algorithm, Support vector machine

1. **Introduction.** Genetic variants associated with *complex diseases* are one of the issues in current studies on human genome. There are many *genome-wide association studies* in the related literature [1,2] to identify genetic variants that might be associated with complex diseases. *SNPs (Single Nucleotide Polymorphisms)* make up most of the genetic variants, and it is estimated that human genome contains approximately 10^6 SNPs [3]. Therefore, SNPs attract intense attention of researchers in the genome-wide association studies [4,5]. Statistical significance of a genome-wide association study is directly related to the number of individuals and SNPs [6]. However, it is still very time-consuming and costly for large-scale association studies to genotype all of the SNPs located within a candidate locus [7-9]. For this reason, a subset of SNPs that will predict *the rest of SNPs* with a small number of errors should be selected. An element of this subset is called a *tag SNP*. Consequently, it is very important to find out a minimal subset of tag SNPs with the prediction of the rest of SNPs with maximal accuracy [7-10].

Various methods have been developed for tag SNP selection so far [7-29]. These methods can be classified into three groups: *block-based*, *block-free* and *linkage disequilibrium (LD)-based* methods. Block-based methods are based on block structure of human genome [30,31]. The basic characteristic of block-based methods is that human genome can be

partitioned into discrete blocks, and that a particular population shares a very small set of common haplotypes within each block. The purpose of these methods is to determine the subset of SNPs that can distinguish all common haplotypes [11-15]. According to these methods, first, a human genome is partitioned into haplotype blocks, and then a subset of tag SNPs is chosen for each block. The main problem of the block-based methods is that blocks cannot always be defined accurately, and it is still unknown how these blocks should accurately be identified [16].

In block-free methods, subset of tag SNPs is used for the reconstruction of the rest of SNPs [7-10,16-22]. These methods do not utilize block partitioning or limited haplotype diversity used in block-based methods. They use the weaker correlations occurring in nearby blocks [18,19]. Lin and Altman [17] suggested a method referred to as *Eigen2htSNP*, which can predict a tagged SNP using a tag SNP having the highest correlation with the tagged SNP. Since SNPs with a low correlation are used in SNP prediction in the Eigen2htSNP method, its prediction accuracy is low. Halperin et al. [7] developed a method for selecting a subset of tag SNPs and named it as *STAMPA*. This method selects at least two tag SNPs that unfortunately may sometimes be worse than a randomly chosen subset of tag SNPs [10]. Lee and Shatkay [8] proposed a method for the selection of tag SNPs based on conditional independence among SNPs and called it *BNTagger*. This method aims to select independent and highly predictive SNPs using Bayesian networks. In this method, the number of tag SNPs to be selected is not given to the algorithm as input. Instead of this, tag SNPs are selected in accordance with the threshold value described before and given to the algorithm as input. Unfortunately, this method is rather time-consuming [10]. He and Zelikovsky [10] proposed two new approaches for SNP prediction that are based on Multiple Linear Regression *(MLR-Tagging)* and Support Vector Machine *(SVM/STSA)*. SVM/STSA is more efficient than MLR-Tagging. However, since it produces hereditary subset of tag SNPs, it is rather time-consuming as well [26]. Moreover, this hereditary property is not always useful. Yang et al. [9] developed a method called as *BPSO*, which is the binary version of particle swarm optimization (PSO) algorithm. Since BPSO method uses the same prediction algorithm as STAMPA, it has the same disadvantages as this method. Recently, a hybrid model, named as Particle Swarm Optimization-Support Vector Machine *(PSO-SVM)* method, has been developed by Lin and Leu [21]. In this model, PSO and SVM are combined using parameter optimization and property choice. In PSO-SVM method, PSO and SVM are used for selection of tag SNPs and for prediction of the rest of SNPs. But the prediction accuracy of the PSO-SVM rapidly diminishes with decreasing the number of tag SNPs. Mahdevar et al. [22] proposed a genetic algorithm based on heuristic method named as *GTagger*. In this method, to calculate the fitness function, correlation and Shannon entropy are used. This method provides low prediction accuracy as well.

Among tag SNP selection methods, LD-based methods use the linkage disequilibrium relationship present between SNP pairs. These methods try to select a set of tag SNPs that is highly associated with each of the SNPs on a given haplotype [23-25,27-29]. However, it is hard to reduce the number of tag SNPs on loci with low LD.

Every method using *support vector machine* (SVM) [32,33] as an SNP prediction model should be able to produce a wide variety of subsets of tag SNPs during random selection of tag SNPs in the search space [6]. One of the best ways of such a selection of tag SNPs is to use a *genetic algorithm* (GA) [34-36]. Unfortunately, a genetic algorithm alone cannot select the tag SNPs with enough accuracy level at the minimum cost of genotyping [6,9]. This is because a GA randomly changes the number of tag SNPs in the individuals, and hence generally the number of selected tag SNPs may be changed in a wide range. If this number is significantly smaller than the optimal one, then the prediction accuracy is

expected to be less than enough. In contrast, if this number is significantly bigger than optimal one, then the genotyping is expected to be more expensive costly. Therefore, there is a need for adjusting the number and the positions of tag SNPs. In this study, the optimal number of tag SNPs for a given dataset is obtained by the exhaustive search method. It is given to the GA as an input. Let us denote the optimal number of tag SNPs by N, the set of tag SNPs generated for an individual in a generation of the GA by GS, and the cardinality by $|GS|$. In this study, the adjusting of the content of the GS set is done as follows:

If $|GS| < N$, then GS is expanded with the new tag SNPs so that $|GS|$ becomes equivalent to N.

If $|GS| > N$, then GS is reduced by removing its elements so that $|GS|$ becomes equivalent to N.

According to this approach, the total number of SNPs is n; in the jth iteration of the adjusting $n - |GS| - j + 1$ testing will be needed. The total number of such testing is to be

$$T = \sum_{j=1}^{k} (n - |GS| - j + 1),$$

where $k = |N - |GS||$. Since always $|GS| \geq 0$ and $k < n$,

$$T = \sum_{j=1}^{k} (n - |GS| - j + 1) < n^2$$

That is, the worst time complexity of selecting a subset of tag SNPs with the best prediction accuracy is to be quadratic (polynomial) in the number of SNPs. We predict the values of the rest of SNPs for a given dataset based on the subset of tag SNPs obtained for this dataset. Our experimental studies showed that the best approach for this aim is the well known *support vector machine* (SVM). Therefore, we developed an approach called *Genetic Algorithm – Support Vector Machine (GA-SVM) Method*, working in the following two steps:

1. GA produces a certain subset of tag SNPs for a certain individual at each of its own iteration and adjusts the number and positions of the elements of this subset.
2. Based on the subsets generated by GA, SVM predicts the values of the rest of SNPs for each individual.

For the evaluation of the prediction accuracy of the algorithm implementing this approach, we use the "*Leave-one-out cross-validation*" (LOOCV) method [8-10,21,37]. The experiments performed on a lot of datasets show that the proposed method provides averagely 6.87% better accuracy than other methods of the same purpose at the range of 2 to 10 tag SNPs.

The rest of the paper is organized as follows. In Section 2, the selection problem of tag SNPs is explained. In Section 3, the method used for selection of tag SNPs is presented. The experimental datasets and results are given in Section 4, and Section 5 presents the conclusions.

2. **The Tag SNPs Selection Problem.**

A *diploid organism* has two non-identical copies of each chromosome. Each of these copies is named as *haplotype* and the data composed of the combination of two haplotypes is named as *genotype* [1]. As it is shown in Figure 1, while each haplotype represents allele information about certain adjacent SNPs on a given chromosome, each genotype represents combined allele information of SNPs on a certain pair of homologous chromosomes [6].

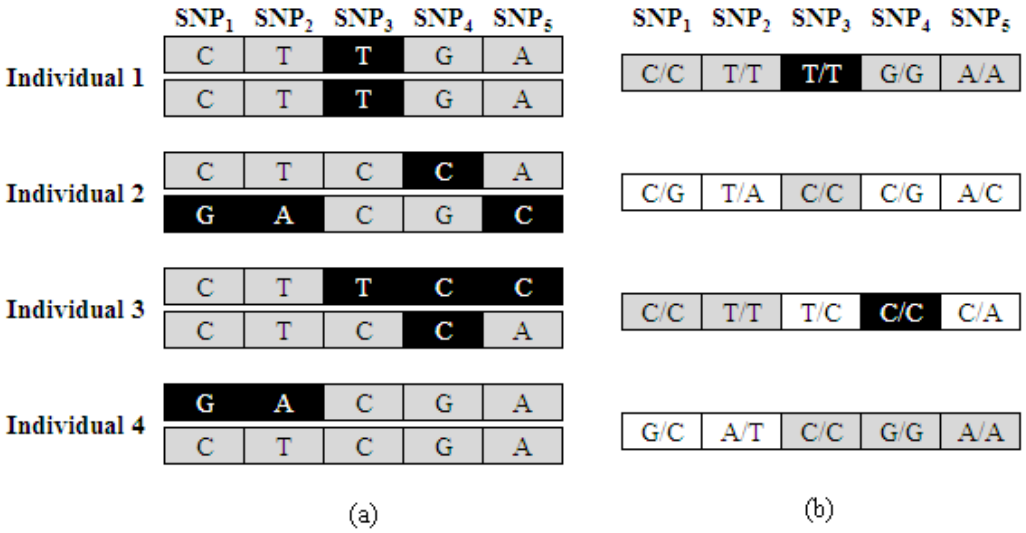

FIGURE 1. (a) Haplotypes and (b) genotypes of four individuals constructed with five SNPs

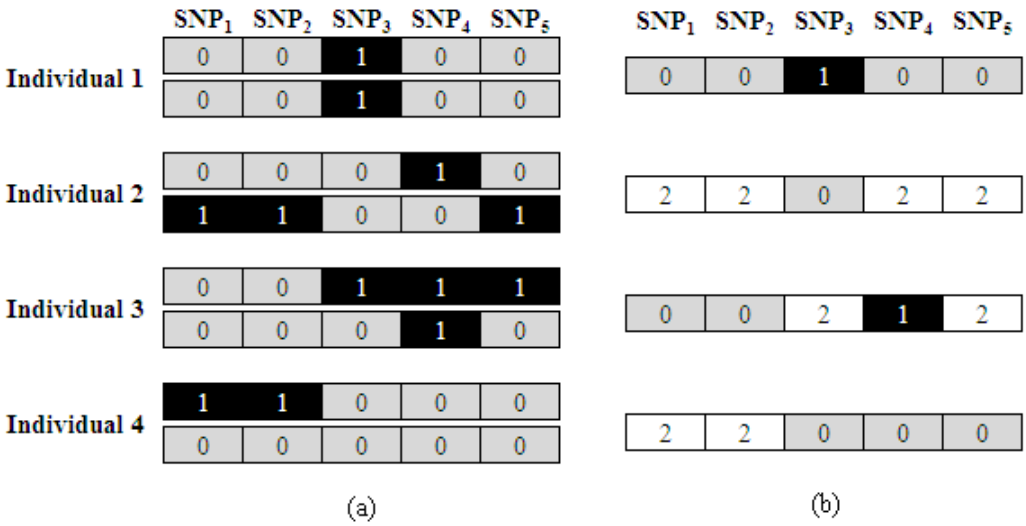

FIGURE 2. Numerical representation of (a) haplotypes and (b) genotypes of four individuals

In a haplotype sequence, SNPs are generally *bi-allelic*, i.e., there are only two alleles in a single SNP: a *major allele* and *minor allele* [26]. Each haplotype for bi-allelic SNPs can be represented by a binary string, where 0 and 1 correspond to major and minor alleles, respectively. Within a given genotype, an SNP is *homozygous* if the alleles are the same and it is *heterozygous* if the alleles are different. That is, each genotype has four states; namely, 00, 01, 10, and 11, from which 01 and 10 are recognized as the same states and are marked by the single number 2. That is, in this encoding of the states of a genotype, a 0 shows that both alleles of SNP are *major homozygous*; 1 shows that both alleles of SNP are *minor homozygous*, and 2 $(01, 10)$ shows that two alleles of SNP are *heterozygous* (Figure 2) [6].

In order to find a subset of tag SNPs providing the prediction the rest of SNPs, a *haplotype matrix* H is used. In this matrix, every row represents a certain haplotype h_i, $i \in \{1, 2, \ldots, m\}$, and every column represents a certain SNP_j, $j \in \{1, 2, \ldots, n\}$. Here,

the problem is to find such a subset $TS = \{t_1, t_2, \ldots, t_N\}$ of tag SNPs that is the minimal among all possible subsets of tag SNPs and provides a prediction of rest of the SNPs with an accuracy higher than other subsets of tag SNPs provide.

3. **The GA-SVM Method.** As mentioned above, there are three categories of methods for selection of the tag SNPs and for the prediction of the allele (major or minor) of the rest of SNPs: *LD-based methods* [23-25,27-29], *block-based methods* [11-15] and *block-free methods* [7-10,16-22]. It should be noted that LD-based methods suffer from low prediction accuracy rates due to decrease in the linkage disequilibrium (*correlation coefficient*) between the tag SNPs and the rest of SNPs [26]. Main problem of the block-based methods is that the correct block identification algorithm is still unknown. In addition, selection of tag SNPs based on local correlations between the markers of each block ignores inter-block correlations [16]. Among block-free methods, a widely used one is STAMPA [8-10,20,21,26], which does not give good results for all datasets or all SNPs [10]. The second block-free based method is the BNTagger method which selects the tag SNPs based on some threshold value, but this method is very time-consuming, too [10]. The third block-free based method is SVM/STSA that uses hereditary property of the set of tag SNPs. However, as it is stated in [26], it is time-consuming as well as BNTagger method. The fourth block-free based method is the hybrid PSO-SVM. But the prediction accuracy of this method is rather poor when the number of tag SNPs is low.

In order to increase the prediction accuracy of SNPs classification problem we suggest a hybrid method, in which the tag SNPs are selected by the genetic algorithm, and the prediction of the rest of SNPs is performed by support vector machine. As mentioned above, we call this method as GA-SVM method.

3.1. **Selecting the tag SNPs by the genetic algorithm.**

3.1.1. *The initial population.* The dataset is represented by a binary matrix H with m rows and n columns, where each row and each column are marked by a certain haplotype (individual) and certain SNPs, respectively [10]. Each individual in such a matrix is represented by an n-bit binary vector. Based on this matrix, the genetic algorithm forms a *population matrix* P with n columns and q rows, where q is the number of individuals in the population chosen randomly from the range 10 to 200 [38,39]. The entry $p_{ij} \in \{0, 1\}$ of the matrix P represents the value of jth SNP for ith individual. A 1 in an individual shows that the associated SNP is a tag SNP and a 0 shows that the value of associated SNP has to be predicted. While all individuals have the same number of tag SNPs denoted by N, different individuals have different combinations of $N < n$ SNPs from the n SNPs. For example, in Figure 3 is a P matrix with $n = 15$ SNPs and 5 individuals where are shown 5 sets of the size 5 differing from each other by at least one tag SNPs.

	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}
Individual 1	0	0	1	0	0	1	0	1	0	0	0	1	0	0	1
Individual 2	1	0	0	1	0	0	0	1	0	1	0	0	1	0	0
Individual 3	1	0	1	0	0	0	1	0	0	0	1	0	0	1	0
Individual 4	0	1	1	0	0	1	0	0	1	0	0	1	0	0	0
Individual 5	0	0	0	1	0	0	1	1	0	0	0	0	1	0	1

FIGURE 3. Population matrix consisting of 5 individuals and 15 SNPs. Each individual consists of 5 tag SNPs and 10 tagged SNPs.

A genetic algorithm is an iterative procedure, which maintains a constant population size formed from the candidate solutions [40,41]. In each of the iterations of this algorithm, three genetic operators (selection, crossover, and mutation) are performed to generate a new population (offspring). The chromosomes of new populations are evaluated by using the fitness function given in the next subsection. Based on these evaluations, the newly generated populations which are better than earlier ones are fixed as the candidate solutions [40,41].

3.1.2. *The fitness evaluation.* In GA, for the *fitness evaluation* of populations, *Leave-one-out cross validation (LOOCV)*, *10-fold cross-validation* and *5-fold cross-validation* methods are used. In this study, we use LOOCV method because its *prediction accuracy* is significantly better than the others [8-10,21,37,42]. According to LOOCV method, in the jth iteration, (1) the jth haplotype is removed from the matrix H. (2) From the remaining haplotypes, the tag SNPs are selected by using the GA. (3) The selected tag SNPs are used for predicting 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 processed as the validation data. In this case, the prediction accuracy (fitness value) is obtained as the ratio of the number of SNPs predicted accurately to the total number of predicted SNPs.

3.1.3. *The natural selection.* The individuals with the highest fitness values in the population should survive, and the others should be removed [43]. *Natural selection* occurs at each iteration of the algorithm. GA includes various selection methods such as *roulette wheel selection*, *random selection*, *scaling selection, tournament selection, hierarchical selection*, and so on. In this study, we used the commonly used roulette wheel selection method [44] because it resembles to the rotation of a wheel on which each chromosome has an area proportional to its fitness. According to this method, a set A of ordered cumulative probabilities of q individuals and a set B of q numbers generated randomly in the range from 0 to 1 are formed. Then, for each number $b_j \in B$, a number $c_i = \min\{a_i \in A : a_i \geq b_j\}$ is selected. The new population is formed as the set $C = \{c_i\}_{i=1}^{q}$.

3.1.4. *The crossover operation.* In order to improve the new population generated as a result of natural selection, the *crossover operation* with a rate of C_R is applied to the individuals of the new population. Generally, the individuals to be subjected to the crossover operation are chosen randomly. In this study, in order to obtain offspring chromosomes from parent chromosomes, uniform crossover operator [45] is used. In order to apply this operator, a crossover mask with 0.5 mixing ratio should be created [46]. This mask is used for determining the particular bits of the parent chromosomes to be crossover. In a certain bit-position of the crossover mask, a 1 means that the SNPs associated with that bit must be crossed between its two parents, while a 0 indicates that the SNPs associated with that bit should remain unchanged. In this study, $C_R = 0.9$ is used as the crossover rate [26,47]. An example of uniform crossover operation performed on the individuals 1 and 3 (Figure 3) is given in Figure 4, where the 3th row is the crossover mask produced by mixing ratio of 0.5.

3.1.5. *The mutation operation.* In order to improve the population generated by crossover operation, the *mutation operator* with a mutation rate M_R is applied to it. For this aim, the mutation operator changes some bits of the population. In order to obtain which bits are to be mutated, a random number between 0 and 1 is generated for each bit-position in all chromosomes. If the mentioned number is smaller than M_R, then the corresponding bits of all chromosomes will be mutated by changing each 0 to 1 and each 1 to 0 [26]. In

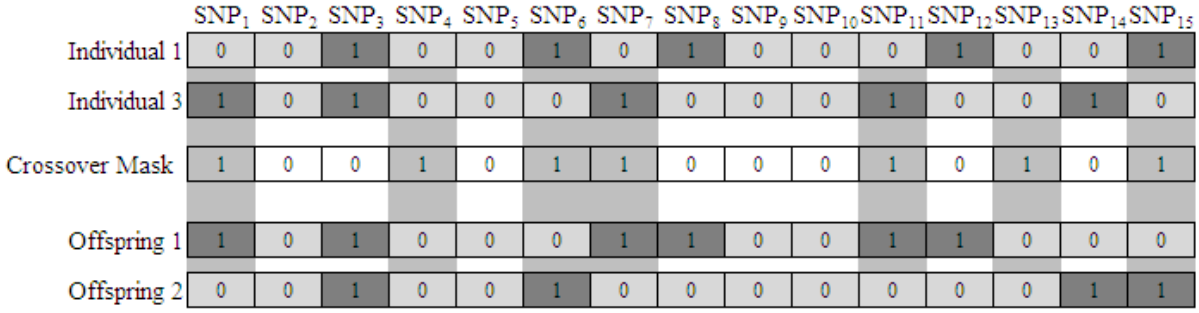

FIGURE 4. Uniform crossover operation applied on individuals 1 and 3

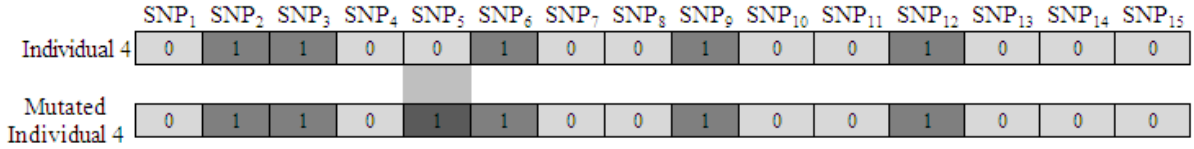

FIGURE 5. The mutation operation on SNP_5 for the individual 4

this study, $M_R = 0.01$ is used as the mutation rate [26,47]. For example, in Figure 5, the mutation operation on SNP_5 for the individual 4 (Figure 3) is shown.

3.1.6. *Adjusting the number of tag SNPs.* After the crossover and mutation operations, the number of 1's indicating the tag SNPs for chromosomes may have been changed [26]. Therefore, the number of tag SNPs for each chromosome should be adjusted so that the number M of tag SNPs for each chromosome can be equivalent to the number N given as input for the GA-SVM algorithm. Two approaches for the solution of this problem have been proposed [9,22,26]. In [22,26], the authors explain an approach, referred to as *random search method*, according to which if $M > N$ then $M - N$ tag SNPs selected randomly are neglected. If $M < N$, then in order to achieve the requested number of tag SNPs, $N - M$ SNPs are not selected yet are added to the group of the selected tag SNPs. Unfortunately, such an adjustment of the numbers of tag SNPs for the chromosomes causes different prediction accuracy rates in different iterations [9]. In [9], the authors propose another approach, named as *local search algorithm*, according to which, in the process of adjusting the number of selected tag SNPs for chromosomes, the new chromosome with better prediction accuracy is fixed. For each candidate chromosome, the process of calculation of prediction accuracy is done by the LOOCV method mentioned above. Unfortunately, in the random search method, the significant change in the prediction accuracy from one iteration to another makes the selection of tag SNPs very hard. In order to minimize the fluctuation of the prediction accuracy, the excessively increasing the number of the iteration is needed. But in this case, the algorithm takes so much time that the solution of many practical applications becomes impossible [35]. In the local search algorithm, to find a new chromosome with the best prediction accuracy, the LOOCV method, which is rather time-consuming, is used [9]. Therefore, it is also impractical for many applications [37].

Our experiments with the LOOCV and 10-fold cross-validation methods show that while 10-fold cross-validation method fulfils the same task as LOOCV method does, it works approximately 10 times faster than LOOCV method [6]. Therefore, in the local search algorithm introduced above, we use the 10-fold cross-validation method, according to which, the dataset H containing up to thousands of haplotype strings and to be

processed by the 10-fold cross-validation method is divided into 10 equal parts that are extracted one by one. For example, in Figure 4, the tag SNPs contained in offspring 1 are marked by $M = 6$ 1's. In Figure 6, 6 candidate chromosomes are prepared one at a time replacing the values of tag SNPs from 1 to 0 in the offspring 1 mentioned. The new offspring 1 chromosome is determined as the candidate chromosome with the best prediction accuracy.

After adjusting the number of tag SNPs, fitness values (prediction accuracy) for all individuals in new population are calculated by LOOCV method, and the individual with the best fitness value is fixed. This operation is repeated $N_G \in \{20, 21, \ldots, 200\}$, where N_G is the number of required generations (iterations) given as an input to the algorithm. From the set of individuals (tag SNPs) fixed as a result of the generations, the one with the best fitness value is returned.

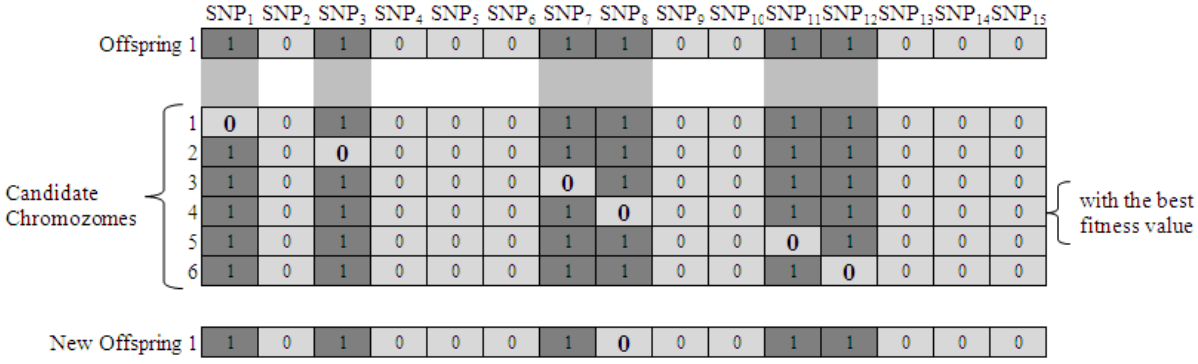

FIGURE 6. The preparation of candidate chromosomes one at a time replacing 1's to 0's in offspring 1

3.2. **Predicting the rest of SNPs by support vector machine.** In order to predict the values of the rest of SNPs, various methods such as *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] are used. Among these methods, in bioinformatics, SVM is preferred since it produces very accurate results and is highly competitive with other data mining approaches such as neural networks [1,10,48,49]. For example, in Figure 7, the process of prediction of rest of SNPs is given. As it is seen in this figure, SVM initially builds a model using the values of the SNPs in haplotypes given as the training set. Then, the values of the rest of SNPs (unknown SNPs values) belonging to the haplotype in the test set are predicted by using this model and the tag SNPs obtained by the GA method introduced above.

In SVM-based prediction method, every haplotype present in H matrix (Figure 7) is considered one test set, where each tag SNP represents one certain feature and each of the rest of SNPs represents one certain class. The ratio of the number of SNPs predicted accurately to the total number of predicted SNPs is referred to as the prediction accuracy.

For predicting the values of the rest of SNPs, we use the *radial basis function (RBF) kernel* of the integrated software *Libsvm* designed for support vector classification [50]. Libsvm is used with the parameters γ and C; γ is a parameter to dominate the generalization ability of SVM by regulating the amplitude of the RBF kernel function and C is a parameter controlling the tradeoff between maximizing the margin and minimizing the training error [32,33]. Various experiments were carried out for the values of γ parameter ranging from 0.01 to 10. While relatively low prediction accuracy rates for all tag SNPs were obtained for the values of this parameter smaller than 0.1, it was observed that

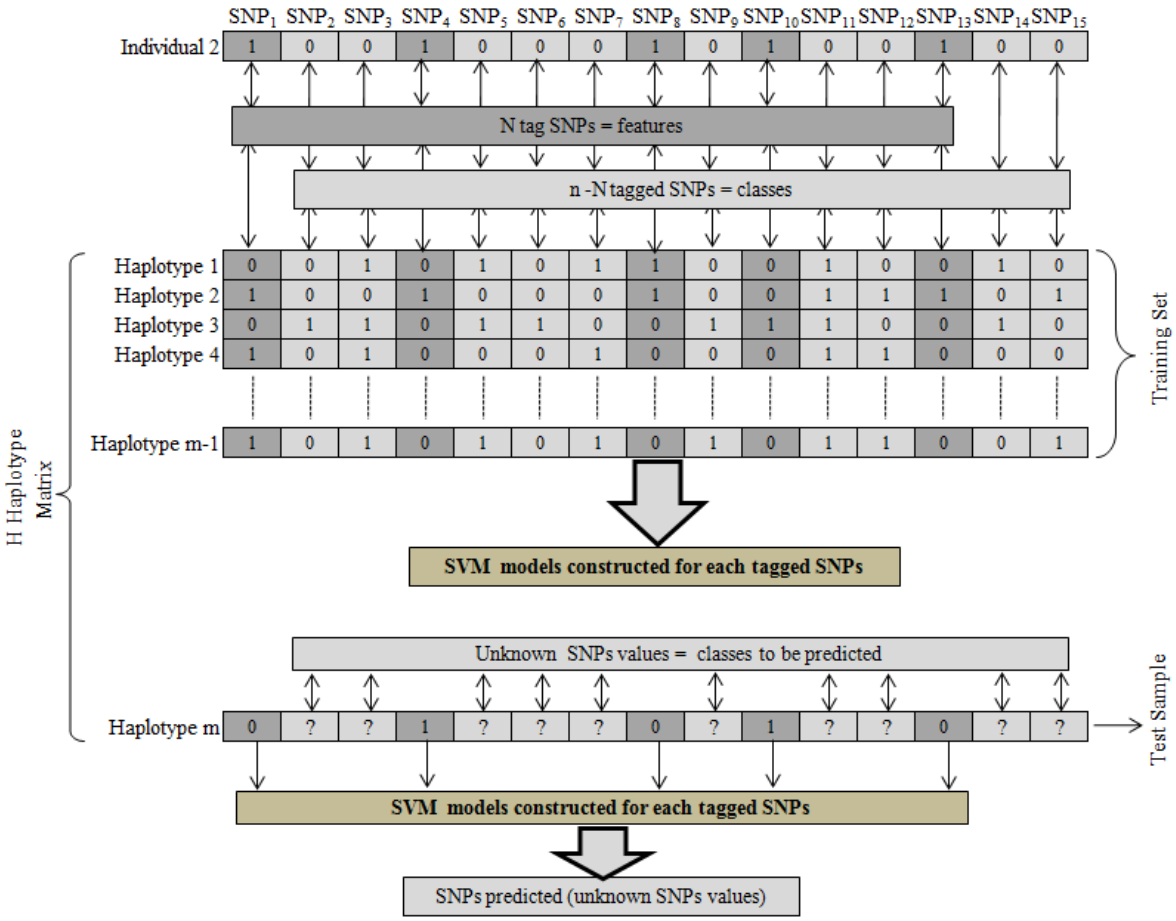

FIGURE 7. The process of prediction of the rest of SNPs belonging to the haplotype m

prediction accuracy rates remained approximately unchanged due to the increase in the number of tag SNPs for the values higher than 0.1. In addition, various experiments were carried out for the values of parameter C ranging from 0.01 to 10. While low prediction accuracy rates for all tag SNPs were obtained for the values of this parameter higher than 0.05, it was observed that prediction accuracy rates increased very little due to the increase in the number of tag SNPs for the values less than 0.05. However, for the values of $\gamma = 0.1$ and $C = 0.05$, prediction accuracy for the GA-SVM algorithm increased exponentially since the number of tag SNPs increased. Therefore, for the prediction of values of the rest of SNPs by the Libsvm, they are used in the experiments as it is recommended in the literature [10].

3.3. **The GA-SVM algorithm.** This algorithm (Figure 8) consists of modules implementing the procedures: *creation of initial population, fitness evaluation, selection, crossover, mutation*, and *adjusting* explained in Subsection 3.1.

4. **Experimental Results.** We processed the following datasets borrowed from the *HapMap* project [51] and from other related papers.

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

Input: *Haplotype matrix H; The number of generation N_G;*
The number of individuals in population q; Crossover Rate C_R;
Mutation Rate M_R; The number of tag SNPs N
Output: *The set of tag SNPs; Prediction accuracy of the set of tag SNPs*
Begin
1. *Based on matrix H and q, create an initial population*
2. *Evaluate the fitness of the population*
3. *Repeat the following steps N_G times*
3.1. *Let the better individuals survive using selection mechanism*
3.2. *Create new individuals using crossover with C_R*
3.3. *Mutate some individuals by using M_R*
3.4. *Adjust the number of selected tag SNPs*
3.5. *Evaluate the fitness of the population*
3.6. *Save the best individual in current population*
4. *Select the individual with the best fitness value*
5. *Return the set of tag SNPs and prediction accuracy of the selected individual*
End

FIGURE 8. The GA-SVM algorithm

ABCB1 (ATP-Binding Cassette, sub-family B) [53]: This dataset is a gene responsible for *P*-glycoprotein and extends over 74 kb of the genome sequence. It consists of 494 haplotypes belonging to 247 individuals and includes 27 bi-allelic SNPs.

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

The chromosome 5q31 dataset [30]: This dataset was derived from the 616 kb region of human chromosome *5q31* from 129 family trios. While this dataset includes 103 bi-allelic SNPs, we used only the children population in it.

The D9 dataset of population D [31]: This dataset consists of 180 haplotypes belonging to 30 family trios taken from Yoruba's population. It includes 49 bi-allelic SNPs.

Two gene regions STEAP and TRPM8: These datasets consist of 30 CEPH (Utah residents with ancestry from northern and western Europe) family trios obtained from HapMap [51]. While the numbers of bi-allelic SNPs in each region are 22 and 101, we used only the population of parents in these datasets.

In order to evaluate the performance of the GA-SVM approach proposed in this study, we have written a program in *MATLAB 7.4* where the abovementioned *Libsvm software* [50] is used as SVM. In the experiments, we used a target machine with an *Intel Core2Quad@2.83 GHz* processor and 4 GB memory, running on *Microsoft Windows 7 Professional Edition OS*.

The experiments showed that there was an increase in prediction accuracy for the number of generation and population size of a GA up to 20, and there were not any improvements in prediction accuracy for the number of generation and population size of a GA above 20. Hence, we chose a GA with a generation number of 20, and population size of 20. In addition, we performed many experiments to obtain the best crossover and mutation rates. The results of these experiments showed that the changes in mutation and crossover rates do not significantly affect the prediction accuracy of the GA-SVM

algorithm. Therefore, for crossover and mutation rates we used the values 0.9 and 0.01 as it is recommended in the related literature [47].

The proposed GA-SVM method was compared with *BNTagger* [8] and *Eigen2htSNP* [17] methods on the ACE dataset with 52 SNPs. The result of experiments for this dataset is given in Figure 9. As it is seen in this figure, GA-SVM method exhibited a performance (prediction accuracy) significantly better than BNTagger and Eigen2htSNP methods for all numbers of tag SNPs. While our method performs worse only for a single tag SNP, it performs significantly better in all other situations where the number of tag SNPs is higher than 1. Moreover, as it is seen in the figure, the prediction accuracy of GA-SVM method regularly increases with increasing the number of tag SNPs. Furthermore, for 2 to 10 tag SNPs, GA-SVM method has average prediction accuracy rates of 2.53% and 5.02%, which are higher than those of BNTagger and Eigen2htSNP methods, respectively.

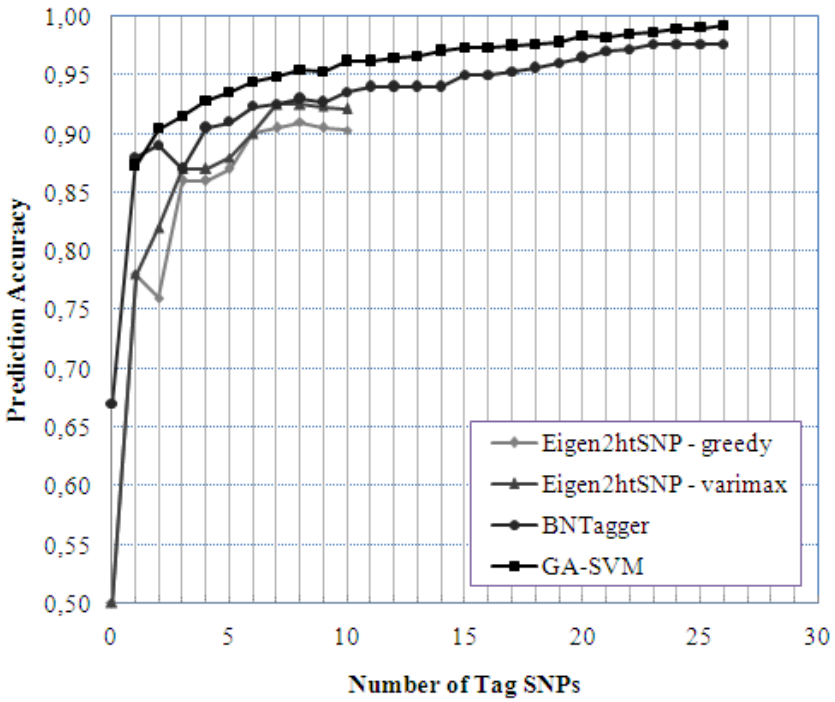

FIGURE 9. Increasing the prediction accuracy of different methods by increasing the number of tag SNPs

For ABCB1 dataset with 27 SNPs, the proposed GA-SVM method was compared with Eigen2htSNP and *STAMPA* [7] methods. For one tag SNP, our method reached 95.3% prediction accuracy in contrast to 55% achieved by Eigen2htSNP methods. For two tag SNPs, our method reached 97% prediction accuracy in contrast to 96.5% achieved by STAMPA method. As it is seen in Figure 10, at the range of 2 to 23 tag SNPs, GA-SVM method exhibits 16.6% and 0.65% more prediction accuracy than Eigen2htSNP and STAMPA methods, respectively, on the average.

The prediction accuracy of the proposed GA-SVM approach for LPL dataset involving 88 unique haplotypes was compared with those of STAMPA, BNTagger, *SVM/STSA* [10], *BPSO* [9] and *PSO-SVM* [21] methods. As it is seen in Figure 11, the GA-SVM method exhibited better performance than the other methods. In particular, at the range of 2 to 20 tag SNPs, the proposed GA-SVM method predicted the rest of SNPs with 3.64%, 2.12%, 3.92%, 5.89% and 1.03% more prediction accuracy than PSO-SVM, BPSO, SVM/STSA, BNTagger and STAMPA methods, respectively, on the average.

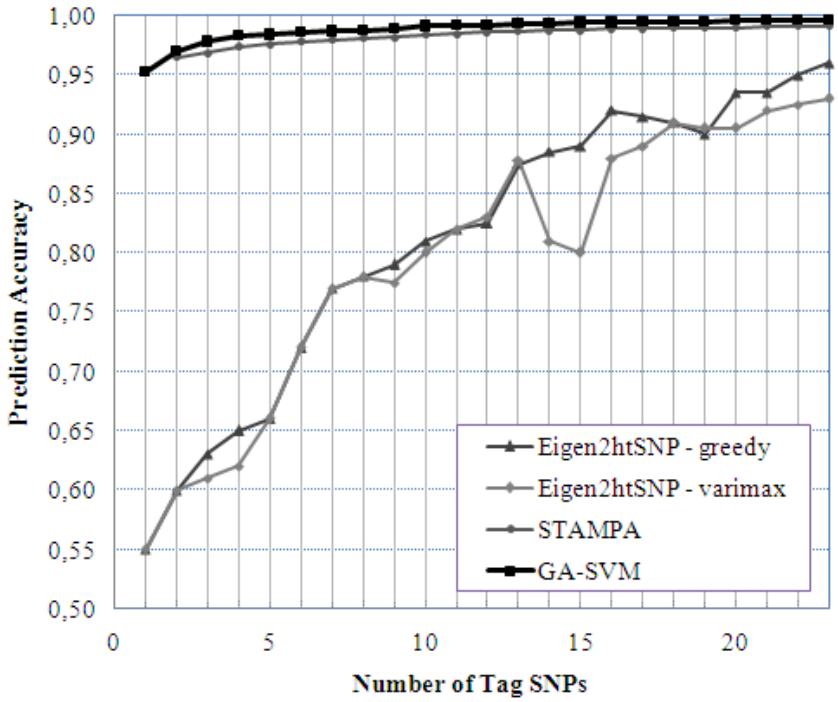

FIGURE 10. Comparison of prediction accuracy of GA-SVM method with STAMPA and Eigen2htSNP methods

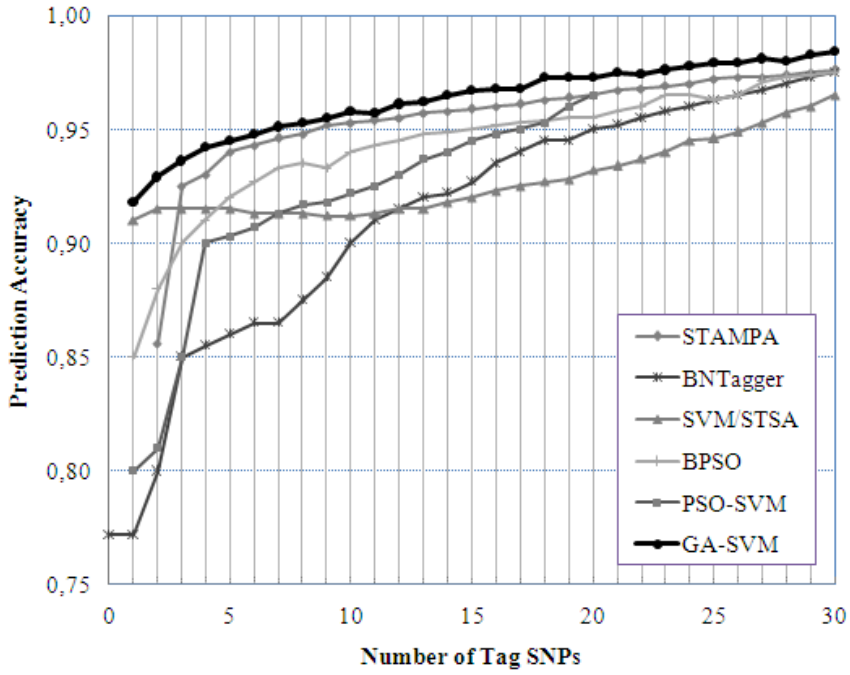

FIGURE 11. The prediction accuracy of GA-SVM and other recent methods for LPL dataset consisting of 88 unique haplotypes

In Table 1, the minimal numbers of tag SNPs needed for achieving the given prediction accuracy are given for the dataset LPL with 88 unique haplotypes.

In Table 1, it is seen that while the GA-SVM method requires more tag SNPs than PSO-SVM method in the narrow range of 98% to 99%, it requires significantly fewer number

TABLE 1. The minimal number of tag SNPs providing the given prediction accuracy for dataset LPL

Prediction Accuracy (%)	The Minimal Number of Tag SNPs Needed for Achieving the Prediction Accuracy Given in the First Column					
	STAMPA	BNTagger	SVM/STSA	BPSO	PSO-SVM	GA-SVM
90	3	10	1	3	4	1
91	3	11	1	4	7	1
92	3	13	13	5	10	2
93	4	16	20	7	12	3
94	5	17	23	10	13	4
95	9	20	27	15	17	7
96	16	24	29	22	20	12
97	24	28	39	27	22	18
98	34	34	42	30	26	27
99	44	43	47	37	31	37

TABLE 2. The prediction accuracies for the dataset 5q31 for different numbers of tag SNPs

The Number of Tag SNPs	The Prediction Accuracy (%) Provided					
	STAMPA	BNTagger	SVM/STSA	BPSO	PSO-SVM	GA-SVM
1	–	87,13	86,81	–	85,00	86,85
2	80,00	89,10	89,32	90,02	90,07	90,23
3	84,28	90,00	–	90,91	91,78	92,17
4	86,70	91,18	92,24	91,89	92,63	92,93
5	88,81	91,90	–	93,12	93,98	94,54
6	90,06	92,73	94,09	94,10	94,96	95,22
7	91,23	93,08	–	95,00	95,91	96,31
8	91,90	93,81	95,28	95,00	95,93	96,71
9	92,28	94,76	–	95,87	95,97	97,53
10	93,01	94,90	96,09	95,93	96,85	97,90

of tag SNPs than the other methods in the wide range of 90% to 97%. In other words, if the required prediction accuracy is more than 97%, the PSO-SVM method should be used, whereas in all other situations the GA-SVM method has to be used. This finding is supported by Tables 2-5, in which the prediction accuracy levels provided by the methods STAMPA, BNTagger, SVM/STSA, BPSO and PSO-SVM for the datasets 5q31, TRPM8, STEAP and D9 are given.

As it is seen in Tables 2-5, for all numbers of tag SNPs in the range of 1 to 10, the GA-SVM method provides a significantly higher accuracy rate than the other methods.

5. **Conclusions.** In this study, a new method for selecting the tag SNPs and predicting the rest of SNPs in a gene is proposed. This knowledge is basically used for identifying the genetic variants associated with complex diseases. In the proposed method, which is simply referred to as GA-SVM method, for the prediction of SNPs and selection of tag SNPs, SVM and GA are used, respectively. The prediction accuracy of the GA-SVM method was compared experimentally with other existing methods by using the datasets of different sizes. The results of experiments show that the proposed method

TABLE 3. The prediction accuracies for the dataset TRPM8 for different numbers of tag SNPs (this dataset has not been processed by the method BNTagger.)

The Number of Tag SNPs	The Prediction Accuracy (%) Provided				
	STAMPA	SVM/STSA	BPSO	PSO-SVM	GA-SVM
1	–	88,89	–	85,00	89,95
2	82,57	90,50	90,02	90,00	90,63
3	86,23	–	90,10	90,00	92,17
4	87,89	90,67	90,17	92,09	92,97
5	89,30	–	92,23	92,98	93,34
6	91,90	93,67	92,71	93,61	94,52
7	92,42	–	93,94	95,14	95,31
8	93,14	95,56	95,00	95,87	96,16
9	93,44	–	96,03	95,92	96,58
10	94,70	96,74	96,12	95,96	97,17

TABLE 4. The prediction accuracies for the dataset STEAP for different numbers of tag SNPs (this dataset has not been processed by the methods BNTagger, BPSO and PSO-SVM.)

The Number of Tag SNPs	The Prediction Accuracy (%) Provided		
	STAMPA	SVM/STSA	GA-SVM
1	–	94,02	94,36
2	95,00	98,18	98,27
3	95,10	–	99,45
4	94,67	99,68	99,70
5	96,22	–	99,72
6	96,43	99,73	99,76
7	96,81	–	99,81
8	97,14	99,79	99,85
9	97,66	–	99,89
10	97,98	99,80	99,93

TABLE 5. The prediction accuracies for the dataset D9 for different numbers of tag SNPs (this dataset has been processed only by the method BPSO.)

The Number of Tag SNPs	The Prediction Accuracy (%) Provided	
	BPSO	GA-SVM
1	–	81,91
2	74,52	84,43
3	75,21	86,20
4	77,34	87,23
5	78,51	88,40
6	79,22	89,14
7	80,53	89,91
8	81,22	91,00
9	83,03	91,92
10	84,71	91,83

has significantly higher accuracy rates than other methods for all possible numbers of tag SNPs.

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

REFERENCES

[1] Y. Q. Zhang and J. C. Rajapakse, *Machine Learning in Bioinformatics*, Wiley, New York, 2008.
[2] M. M. Iles, What can genome-wide association studies tell us about the genetics of common disease, *PLoS Genet.*, vol.4, no.2, 2008.
[3] L. Kruglyak and D. A. Nickerson, Variation is the spice of life, *Nature Genetics*, vol.27, no.3, pp.234-236, 2001.
[4] B. V. Halldorsson, V. Bafna, N. Edwards, R. Lippert, S. Yooseph and S. Istrail, A survey of computational methods for determining haplotyes, *Lecture Notes in Computer Science*, vol.2983, pp.26-47, 2004.
[5] D. Crawford and D. A. Nickerson, Definition and clinical importance of haplotypes, *Annual Review of Medicine*, vol.56, no.1, pp.303-320, 2005.
[6] İ. İlhan, Y. E. Göktepe and S̃. Kahramanlı, Tag SNP selection using GA-SVM approach, *IADIS European Conference on Data Mining*, Rome, Italy, pp.27-34, 2011.
[7] E. Halperin, G. Kimmel and R. Shamir, Tag SNP selection in genotype data for maximizing SNP prediction accuracy, *Bioinformatics*, vol.21, pp.195-203, 2005.
[8] P. H. Lee and H. Shatkay, BNTagger: Improved tagging SNP selection using Bayesian networks, *Bioinformatics*, vol.22, no.14, pp.211-219, 2006.
[9] C. Y. Yang, C. H. Hou and L. Y. Chuang, Improved tag SNP selection using binary particle swarm optimization, *IEEE Congress on Evolutionary Computation*, pp.854-860, 2008.
[10] J. He and A. Zelikovsky, Informative SNP selection methods based on SNP prediction, *IEEE Trans. Nanobioscience*, vol.6, no.1, pp.60-67, 2007.
[11] N. Patil, A. J. Berno, D. A. Hinds, W. A. Barrett, J. M. Doshi, C. R. Hacker et al., Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21, *Science*, vol.294, no.5547, pp.1719-1723, 2001.
[12] K. Zhang, M. Deng, T. Chen, M. S. Waterman and F. Sun, A dynamic programming algorithm for haplotype block partitioning, *Proc. of the National Academy of Sciences of the United States of America*, vol.99, no.11, pp.7335-7339, 2002.
[13] K. Zhang, F. Sun, M. S. Waterman and T. Chen, Haplotype block partition with limited resources and applications to human chromosome 21 haplotype data, *American Journal of Human Genetics*, vol.73, no.1, pp.63-73, 2003.
[14] X. Ke and L. R. Cardon, Efficient selective screening of haplotype tag SNPs, *Bioinformatics*, vol.19, no.2, pp.287-288, 2003.
[15] K. Zhang, Z. Qin, J. Liu, T. Chen, M. Waterman and F. Sun, Haplotype block partitioning and tag SNP selection using genotype data and their applications to association studies, *Genome Res.*, vol.14, no.5, pp.908-916, 2004.
[16] T. M. Phuong, Z. Lin and R. B. Altman, Choosing SNPs using feature selection, *Proc. of the IEEE Computational Systems Bioinformatics Conference*, Standford, CA, pp.301-309, 2005.
[17] Z. Lin and R. Altman, Finding haplotype tagging SNP by use of principle component analysis, *Am. J. Hum. Genet.*, vol.75, no.5, pp.850-861, 2004.
[18] V. Bafna, B. V. Halldorsson, R. Schwrtz, A. Clark and S. Istrail, Haplotypes and informative SNP selection algorithms: Don't block out information, *Proc. of the 7th Annual International Conference on Research in Computational Molecular Biology*, Berlin, Germany, pp.19-27, 2003.
[19] B. V. Halldorsson, V. Bafna, R. Lippert, R. Schwartz, F. M. De La Vega, A. G. Clark and S. Istrail, Optimal haplotype block-free selection of tagging SNPs for genome-wide association studies, *Genome Research*, vol.14, no.1, pp.1633-1640, 2004.
[20] P. Bertolazzi, G. Felici and P. Festa, Logic based methods for SNPs tagging and reconstruction, *Computer & Operations Research*, vol.37, no.8, pp.1419-1426, 2009.
[21] M. H. Lin and C. L. Leu, A hybrid PSO-SVM approach for haplotype tagging SNP selection problem, *International Journal of Computer Science and Information Security*, vol.8, no.6, pp.60-65, 2010.

[22] G. Mahdevar, J. Zahiri, M. Sadeghi, A. N. Dalini and H. Ahrabian, Tag SNP selection via a genetic algorithm, *Journal of Biomedical Informatics*, vol.43, no.5, pp.800-804, 2010.
[23] S. I. Ao, K. Yip, M. Ng, D. Cheung, P. Y. Fong, I. Melhado and P. C. Sham, CLUSTAG: Hierarchical clustering and graph methods for selecting tag SNPs, *Bioinformatics*, vol.21, no.8, pp.1735-1736, 2004.
[24] H. I. Avi-Itzhak, X. Su and F. M. De La Vega, Selection of minimum subsets of single nucleotide polymorphisms to capture haplotype block diversity, *Proc. of Pac. Symp. Biocomput.*, vol.8, pp.466-477, 2003.
[25] C. S. Carlson, A. E. Michael, J. R. Mark, Y. Qian, K. Leonid and A. N. Deborah, Selecting a maximally informative set of single nucleotide polymorphisms for association analyses using linkage disequilibrium, *Am. J. Human Genet.*, vol.74, no.1, pp.106-120, 2004.
[26] L. Y. Chuang, C. H. Hou and C. Y. Yang, A novel prediction method for tag SNP selection using genetic algorithm based on KNN, *International Journal of Chemical and Biological Engineering*, vol.3, no.1, pp.12-17, 2010.
[27] J. Hampe, S. Schreiber and M. Krawczak, Entropy-based SNP selection for genetic association studies, *Hum. Genet.*, vol.114, no.1, pp.36-43, 2003.
[28] K. Hao, Genome-wide selection of tag SNPs using multiple marker correlation, *Bioinformatics*, vol.23, no.23, pp.3178-3184, 2007.
[29] G. Liu, Y. Wang and L. Wong, FastTagger: An efficient algorithm for genome-wide tag SNP selection using multi-marker linkage disequilibrium, *BMC Bioinformatics*, vol.11, 2010.
[30] M. J. Daly, J. D. Rioux, S. F. Schaffner, T. J. Hudson and E. S. Lander, High-resolution haplotype structure in the human genome, *Nature Genetetic*, vol.29, no.2, pp.229-232, 2001.
[31] S. B. Gabriel, S. F. Schaffner, H. Nguyen, J. M. Moore, J. Roy, B. Blumenstiel et al., The structure of haplotype blocks in the human genome, *Science*, vol.296, no.5576, pp.2225-2229, 2002.
[32] V. N. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.
[33] C. Cores and V. N. Vapnik, Support vector networks, *Machine Learning*, vol.20, no.3, pp.273-297, 1995.
[34] J. H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, 1975.
[35] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, New York, 1989.
[36] A. Kumamoto, A. Utani and H. Yamamoto, Advanced particle swarm optimization for computing plural acceptable solutions, *International Journal of Innovative Computing, Information and Control*, vol.5, no.11(B), pp.4383-4392, 2009.
[37] S. Arlot and A. Celisse, A survey of cross-validation procedures for model selection, *Statistics Surveys*, vol.4, pp.40-79, 2010.
[38] T. Sağ and M. Cunkaş, A tool for multiobjective evoluationary algorithms, *Advances in Engineering Software*, vol.40, no.9, pp.902-912, 2009.
[39] Y. Guo, X. Cao, H. Yin and Z. Tang, Coevolutionary optimization algorithm with dynamic subpopulation size, *International Journal of Innovative Computing, Information and Control*, vol.3, no.2, pp.435-448, 2007.
[40] S. Zou, Y. Huang, Y. Wang, J. Wang and C. Zhou, SVM learning from imbalanced data by GA sampling for protein domain predicting, *The 9th International Conference for Young Computer Scientists*, pp.982-987, 2008.
[41] F. Xhafa, J. Carretero and A. Abraham, Genetic algorithm based schedulers for grid computing systems, *International Journal of Innovative Computing, Information and Control*, vol.3, no.5, pp.1053-1071, 2007.
[42] A. R. Ahad, T. Ogata, J. Tan, H. Kim and S. Ishikawa, A complex motion recognition technique employing directional motion templates, *International Journal of Innovative Computing, Information and Control*, vol.4, no.8, pp.1943-1954, 2008.
[43] P. J. Angeline, Evolution revolution: An introduction to the special track on genetic and evolutionary programming, *IEEE Expert Intelligent Systems and Their Applications*, vol.10, no.3, pp.6-10, 1995.
[44] D. E. Goldberg and K. Deb, A comparative analysis of selection schemes used in genetic algorithms, *Foundations of Genetic Algorithms*, vol.1, pp.69-93, 1991.
[45] G. Sywerda, Uniform crossover in genetic algorithms, *Proc. of the 3rd International Conference on Genetic Algorithms*, Los Altos, CA, pp.2-9, 1989.
[46] A. Prügel-Bennett, The mixing rate of different crossover operators, *Foundations of Genetic Algorithms*, vol.6, pp.261-274, 2001.

[47] W. Y. Lin, W. Y. Lee and T. P. Hong, Adapting crossover and mutation rates in genetic algorithms, *Journal of Information Science and Engineering*, vol.19, no.5, pp.889-903, 2003.

[48] X. Song, W. Chen and B. Jiang, Sample reducing method in support vector machine based on K-closest sub-clusters, *International Journal of Innovative Computing, Information and Control*, vol.4, no.7, pp.1751-1760, 2008.

[49] R.-C. Chen and S.-P. Chen, Intrusion detection using a hybrid support vector machine based on entropy and TF-IDF, *International Journal of Innovative Computing, Information and Control*, vol.4, no.2, pp.413-424, 2008.

[50] C. C. Chang and C. J. Lin, *LIBSVM: A Library for Support Vector Machines*, http://www.csie.ntu.edu.tw/∼cjlin/libsvm, 2001.

[51] International HapMap Consortium, The international HapMap project, *Nature*, vol.426, no.6968, pp.789-796, 2003.

[52] M. J. Rieder, S. L. Taylor, A. G. Clark and D. A. Nickerson, Sequence variation in the human angiotensin converting enzyme, *Nat. Genet.*, vol.22, no.1, pp.59-62, 1999.

[53] D. L. Kroetz, C. Pauli-Magnus, L. M. Hodges et al., Sequence diversity and haplotype structure in the human ABCB1 (MDR1, multi drug resistance transporter) gene, *Pharmacogenetics*, vol.13, no.8, pp.481-494, 2003.

[54] A. Clark, K. Weiss, D. Nickerson, S. Taylor, A. Buchanan, J. Stengard et al., Haplotype structure and population genetic inferences from nucleotide-sequence variation in human lipoprotein lipase, *Hum. Genet.*, vol.63, no.2, pp.595-612, 1998.