Jestr

Journal of Engineering Science and Technology Review 11 (5) (2018) 76 - 85

Research Article

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

SNP Selection using Variable Ranking and Sequential Forward Floating Selection with Two Optimality Criteria

Dani Setiawan[1,2,3,*], Wisnu Ananta Kusuma[2] and Aji Hamim Wigena[3]

[1]*Akademi Teknik Soroako, Jl. Sumantri Brojonegoro No. 1, Kecamatan Nuha, Kabupaten Luwu Timur, Sulawesi Selatan, Indonesia, 92984*
[2]*Dept. of Computer Science, Institut Pertanian Bogor, Jl. Meranti Wing 20 Level V, Kampus Dramaga, Bogor, Jawa Barat, Indonesia, 16680*
[3]*Dept. of Statistics, Institut Pertanian Bogor, Jl. Meranti Wing 22 Level IV, Kampus Dramaga, Bogor, Jawa Barat, Indonesia, 16680*

Received 8 December 2017; Accepted 22 November 2018

Abstract

Bioinformatics is one of the many areas which apply feature selection techniques. In bioinformatics, genome wide association studies (GWAS) is an observational study aimed at determining whether a genetic variant is associated with a certain observed trait. Single nucleotide polymorphism (SNP) is the most popular genetic marker used to identify genetic polymorphisms. Here we propose the use of variable ranking methods to remove less important SNPs prior to SNP selection. We compared methods of SNP ranking by means of statistical approaches, i.e., correlation-adjusted marginal correlation score (CAR score) and influential score (I-score), and machine learning approach using random forest algorithm in an attempt to reduce the search space. The search in the reduced space was then conducted using sequential forward floating selection (SFFS) which wraps support vector regression (SVR), and the results obtained by two of multi-purpose kernels—radial basis function (RBF) kernel and Bessel kernel—were compared for this high-dimensional linear regression problem, i.e., the search for the most appropriate combination of SNPs which have association with the phenotypes of interest. We propose the use of two optimality or selection criteria, the adjusted R^2 and the mean squared error, in the hope that the selected SNPs are those with both high statistical significance and strong predictive power. Testing was conducted using two simulated data sets with and without epistasis. Our results show that the intersection of the two selected subsets obtained by the two selection criteria can reduce the number of, or even eliminate, false positives. Furthermore, they suggest that the removal of less important SNPs prior to SNP selection improves the selection results. They also suggest that the proposed SNP selection method is better than the methods proposed by De Oliveira et al. (2014) and Kusuma et al. (2016).

Keywords: association studies, bioinformatics, feature selection, single nucleotide polymorphism

1. Introduction

Genome wide association studies (GWAS) seek to determine whether a polymorphism is associated with a certain trait, commonly referred to as a phenotype, observed in individuals. Polymorphism is defined as a genetic variant at a single locus, i.e., location within a gene. A genetic variation must be present in at least 1% of a population to be considered a polymorphism. Such a variable site is commonly referred to as a single nucleotide polymorphism (SNP) [1]. SNP is the most popular markers used to identify genetic polymorphisms since it allows generation of abundant information on genetic variability at DNA level.

The observed genetic sequence information is called a genotype. In a genotype, one may observe the so-called epistasis by which the effect of one mutated gene (locus) is dependent on the presence of one or more genes. Thus, epistatic mutations have different effects in combination than individually. Epistasis arises due to interactions, either between genes or within them, that lead to non-linear effects [2]. Genotype-phenotype association based on single locus association is not suitable for complex phenotypes with epistatic interactions among genes, while multi-locus association, classified as a combinatorial problem, is capable of explaining complex genetic polymorphisms. Nevertheless, the search space grows exponentially with the size of the data.

Association study is a feature selection problem [3]. In population-based association studies, SNP is considered as the fundamental unit of analysis, i.e., the feature. A SNP describes a single base pair change that is variable across the general population at a frequency of at least 1%. The SNPs are treated as the predictor variables or the independent variables and the phenotype as the response or the dependent variable. Typically, one wants to find a subset of SNPs which is associated with the observed trait. One may also want to predict whether a new individual has the trait by analyzing the individual's selected SNPs [1].

Feature selection has been widely used in studies in which data sets with a very large number of variables are found. The three objectives of feature selection are: (1) to improve the prediction performance of the predictors, (2) to provide faster and more cost-effective predictors, and (3) to provide a better understanding of the underlying process that generated the data [3][4]. Feature selection techniques can be classified into three categories, depending on how they combine the feature selection search with the construction of

*E-mail address: mr.danisetiawan@gmail.com
ISSN: 1791-2377
doi:10.25103/jestr.115.09

the model: (1) filter methods, (2) wrapper methods, and (3) embedded methods [3].

Cover and Van Capenhout (1977) showed that only an exhaustive search, in which one evaluates all possible subsets, can guarantee the best feature subset from the full set of features [5]. However, for high-dimensional feature selection problems, i.e, problems with a large number of independent variables in their data sets, it is computationally intensive. In the case of SNP selection, the complete search space of all combinations of SNPs is given by 2^n, where n is the number of SNPs [6]. Many non-exhaustive search algorithms have been proposed for feature selection. Zongker and Jain (1996) evaluated the quality of the feature subsets generated by various algorithms and compared their computational requirements. They showed that sequential forward floating selection (SFFS) proposed by Pudil *et al*. (1994) dominates the other algorithms tested [7].

Some of the feature selection algorithms were proposed specifically for SNP selection problem. De Oliveira *et al*. (2014) proposed a combination of statistical approach, genetic algorithm (GA), and support vector regression (SVR) with Pearson universal kernel (PUK) introduced by Üstün *et al*. (2006) for quantitative phenotypes. In the first selection of markers, Spearman's rank correlation coefficient was used to construct the most significant groups of markers in order to reduce the search space. For each group defined after the first selection, a SVR model was constructed through the Pearson's correlation coefficient in 10-fold cross-validation. In the second selection of markers, a wrapper based on binary GA with cross-validation mean squared error (MSE) as the fitness function is applied [6]. Kusuma *et al*. (2016) proposed a combination of a novel heuristic search named gravitational search algorithm (GSA) which was introduced by Rashedi *et al*. (2009, 2010) and a wrapper based on sequential forward selection (SFS) with Spearman's rank correlation coefficient as selection criterion which was evaluated using SVR with Gaussian radial basis function (RBF) kernel. Their method consists of two steps: the first step is aimed at reducing the search space, i.e., removing redundant and irrelevant SNPs using GSA, and the second step is SNP selection using SFS (referred to as 'exhaustive search' in their paper) [8]. Both methods were tested on the same two simulated data sets generated by the function simulateSNPglm of the 'scrime' package in R [9]. The first data set only has main effects without interaction among SNPs, while the second one has epistasis among SNPs [6][8]. Tab. 1 summarizes the results of both methods. Note that none of these two methods were able to completely capture all SNPs which are actually associated with the phenotypes, and both methods produced false positives.

Table 1. SNPs selected from two simulated phenotypes using methods proposed by De Oliveira *et al*. (2014) and Kusuma *et al*. (2016).

Method	Selected SNPs[a]	Spearman's Correlation[b]
De Oliveira *et al.*	**1**, **10**, 15, **20**, **30**, **60**, 158, 177, 269, 274, 391, 446, 516, 673, 686, 693, 717, 725, 739, 825, 930	0.750
	3	0.950
Kusuma *et al.*	**1**, **10**, **20**, **30**, **50**, **60**, 72	0.830
	3, **4**, 8029	0.870

[a]Bold types denote SNPs which are actually associated with the corresponding phenotype. For each method, the first row is for the simulated phenotype without epistasis while the second row is for the simulated phenotype with epistasis.
[b]Source: Kusuma *et al*. (2016). In their paper, De Oliveira *et al*. (2014) used MSE as the selection criterion.

Here we propose the use of variable ranking methods to remove less important SNPs prior to SNP selection, a high-dimensional linear regression problem, in the hope that it can help to achieve better selection result and reduce the required computational time. We compared methods of SNP ranking by means of statistical approaches, i.e., correlation-adjusted marginal correlation score (CAR score) introduced by Zuber and Strimmer (2011) and influential score (I-score) described by Lo *et al*. (2015), and a machine learning approach using random forest algorithm introduced by Breiman (2001) to be used for search space reduction. The search in the reduced space was then conducted using SFFS described by Pudil *et al*. (1994) which wraps SVR, and the selection results obtained using two of multi-purpose kernels—the RBF kernel and the Bessel kernel—were compared. In this work, we only consider continuous or quantitative phenotypes. We propose the use of two selection criteria, the adjusted R^2 and the mean squared error, in the hope that the selected SNPs are those with both high statistical significance and strong predictive power as a way to reduce the number of, or even eliminate, false positives. Testing was conducted using the same two simulated data sets as in De Oliveira *et al*. (2014) and Kusuma *et al*. (2016).

2. Method

2.1 Genotype representation

SNPs are generally bi-allelic, i.e., there are two possible bases at the corresponding variable site within a gene. In other words, there are only two alleles in a single SNP: major allele and minor (or variant) allele. The minor allele frequency (MAF), also referred to as the variant allele frequency, refers to the frequency of the less common allele at a variable site. Here the term frequency refers to a population proportion. A variation in the nucleotide is considered a SNP if it has a MAF of at least 1%. MAF is widely used in population genetics studies since it can be used to distinguish between common and rare variants in the population [1].

A diploid organism has two non-identical copies of each chromosome, i.e., each individual carries two bases, corresponding to each of two homologous chromosomes, usually one from the mother and one from the father. Each of these copies is referred to as haplotype and the data composed of the combination of two haplotypes is referred to as genotype. Haplotype refers to the specific combination of alleles that are in alignment on a single homolog, defined as one of the two homologous chromosomes. As shown in Fig. 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 [10]. In its rawest form, the genotype is the pair of DNA bases adenine (A), thymine (T), guanine (G) and/or cytosine (C) which is observed at a location on the organism's genome. This pairing of homologous chromosomes that makes up an individual's genotype is different from that which makes up a DNA double helix. In the latter pairing, guanine always pairs with cytosine (G-C) and adenine always pairs with

thymine (A-T), while the former pairing is not thus restricted so that, for example, genotypes GT and AC can be observed [1].

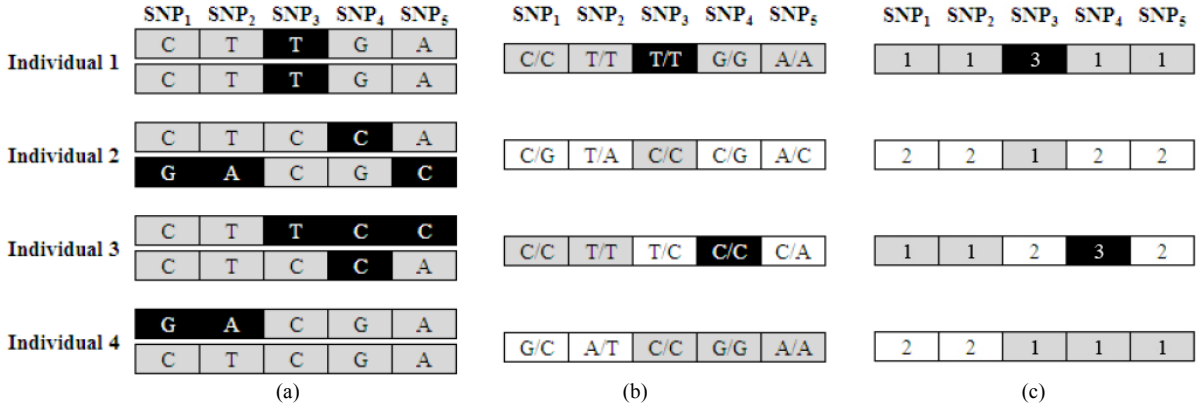

Fig. 1. (a) Haplotypes and (b) genotypes of four individuals constructed with five SNPs and (c) numerical representation of genotypes. (Adapted from Ilhan *et al*. 2013)

Within a given genotype, a SNP is called homozygous if both its alleles are the same and is called heterozygous if its two alleles are different. The simulated data sets generated by the function simulateSNPglm of the 'scrime' package in R use the following encoding of the states of a genotype: 1 for homozygous reference genotype (both alleles of SNP are major homozygous), 2 for heterozygous genotype (two alleles of SNP are heterozygous), and 3 for homozygous variant genotype (both alleles of SNP are minor homozygous) [9]. Thus, the genotype can be treated as a categorical variable. Subtracting 1 from these numbers gives the number of copies of the minor allele [11]. For this reason some researchers prefer to use the numbers 0, 1, and 2 for encoding the states of a genotype.

2.2 Correlation-adjusted marginal correlation score

In GWAS based on single locus analyses, each SNP is considered independently of all others and its association with the phenotype is computed using a univariate test statistic. While this approach is computationally inexpensive, it assumes complete independence of SNPs and thus ignores the correlation structure among SNPs, e.g., due to linkage or interaction among SNPs [12]. Zuber and Strimmer introduced two novel statistics, the correlation-adjusted *t*-score (CAT score) and the correlation-adjusted marginal correlation score (CAR score) in 2009 and 2011, respectively. CAT score is used for binary (qualitative) phenotype while CAR score is used for continuous (quantitative) phenotype. These two measures are multivariate generalizations of the standard univariate test statistics that take the correlation among SNPs explicitly into account and lead to improved rankings of SNPs [12]. The CAR scores are the correlations between the response (phenotype) and the Mahalanobis-decorrelated predictors (SNPs) [13].

Consider a linear regression model for a set of d SNPs, $X = \{X_1, \ldots, X_d\}$, and a continuous phenotype Y. The correlation matrix among SNPs is a $d \times d$ square matrix which is denoted here by $\mathbf{P}$. The vector of marginal correlations, $\mathbf{P}_{XY} = \left(\rho_{X_1 Y}, \ldots, \rho_{X_d Y}\right)^T$, contains the correlations between a phenotype and each individual SNP. If there is no correlation among SNPs ($\mathbf{P} = \mathbf{I}_d$) then the marginal correlations, $\mathbf{P}_{XY}$, provide an optimal ranking of SNP, and the sum of the squared marginal correlations equals to the squared multiple correlation coefficient or coefficient of determination, R^2. However, in the presence of correlation among SNPs, the squared marginal correlations do not sum up to R^2, i.e., $\mathbf{P}_{YX}\mathbf{P}_{XY} \neq R^2$. The CAR score is given by

$$\mathbf{P}_{XY}^{\text{adj}} = \mathbf{P}^{-1/2}\mathbf{P}_{XY} \tag{1}$$

The squared CAR scores sum up to the squared multiple correlation coefficient,

$$\left(\mathbf{P}_{XY}^{\text{adj}}\right)^T \mathbf{P}_{XY}^{\text{adj}} = \mathbf{P}_{YX}\mathbf{P}^{-1}\mathbf{P}_{XY} = R^2 \tag{2}$$

also known as the coefficient of determination or the proportion of variance explained, even in the presence of correlation among SNPs. This decomposition property allows CAR scores to assign importance to groups of SNPs, not only to individual SNPs. Moreover, CAR score has a grouping property which gives similar scores for highly correlated SNPs. This property protects against antagonistic SNPs, i.e., if two SNPs are highly correlated and one has a protective and the other a risk effect, then both SNPs are assigned low scores [12]. In Zuber and Strimmer (2011) it is argued that squared CAR scores are a natural measure for variable importance and it is shown that variable selection based on CAR scores is highly efficient compared to competing approaches such as elastic net lasso, or boosting. This method of assigning variable importance falls into the class of filter techniques [3].

2.3 Influential score

In GWAS, it has been observed that an increase in predictor found to be significantly correlated with a response does not necessarily lead to improvements in the predictive models. In other words, statistically significant predictor variables are not leading to good prediction [11]. Lo *et al.* (2015) suggest that higher statistical significance does not automatically imply stronger predictive power, while highly predictive variables do not necessarily appear as highly statistically significant. Motivated by this observation, Lo *et al.* (2015) further developed the so-called influential score, abbreviated to I-score, introduced by Chernoff *et al.* (2009). I-score is a measure that evaluates the amount of influence of a set of SNPs to quantify its association with a phenotype.

Consider n observations on a phenotype Y. In a so-called partition, which is a small subset of m SNPs, there are 3^m possible so-called cells since each SNP can be assigned a value of 1, 2, or 3. Each individual i in a partition Π_X is represented by a value Y_i of the dependent variable and one

of 3^m possible cells into which the m variables fall. The influential score for this partition is given by

$$I_{\Pi_X} = \sum_{j=1}^{3^m} \frac{n_j}{n} \frac{\left(\bar{Y}_j - \bar{Y}\right)^2}{s^2/n_j} = \frac{\sum_{j=1}^{3^m} n_j^2 \left(\bar{Y}_j - \bar{Y}\right)^2}{\sum_{i=1}^{n} (\bar{Y}_i - \bar{Y})^2} \quad (3)$$

where Y_i is the phenotype which corresponds to i-th individual, $\bar{Y}$ is the mean of all n phenotypes, s is the standard deviation of all n phenotypes, $\bar{Y}_j$ is the mean of the phenotype values in cell j, n_j is the number of individuals in cell j, and n is the total number of individuals. Thus, the measure I-score is a statistic which can be calculated from the observed data, and does not involve knowing the underlying distributions [11]. The values of I-score substantially greater than unity signify possible influence [14].

Lo *et al.* (2015) showed that I-score has the folowing desirable properties. First, it does not require specification of a model for the joint effect of the subset $\{\mathrm{SNP}_1, \ldots, \mathrm{SNP}_m\}$ on the phenotype Y since it is designed to capture the discrepancy between the conditional means of Y given the values of $\{\mathrm{SNP}_1, \ldots, \mathrm{SNP}_m\}$ and the overall mean of Y. Second, the expected value of I-score does not monotonically increase as more predictor variables are added to the variable subset. Rather, given a variable set of size m with $m-1$ truly influential variables, the I-score is typically higher under the influential $m-1$ variables than under all m variables. If $m-1$ variables are influential in the sense that any smaller subset of variables is less influential, then the removal of a variable to size $m-2$ will decrease the I-score. Thus, the I-score has a natural tendency to "peak" at variable set(s) that lead to high predictive power in the face of noisy variables under the current sample size [15]. For high-dimensional variable selection problem, one way to thin out the candidates, i.e., to reduce the search space is to apply the I-score to one explanatory variable at a time, and to focus on those which indicate strong marginal observable effects [14].

2.4 Variable ranking by random forest

Random forest algorithm which was developed by Breiman (2001) is an ensemble learning method for classification and regression that works by constructing a forest of random and uncorrelated decision trees at training time and outputting the class that is the mode of the classes (in classification problem) or the mean prediction of the individual trees (in regression problem). Random decision forests correct for decision trees' habit of overfitting to their training set [16]. The algorithm uses out-of-bag (OOB) error as an estimate of the generalization error and measures variable importance. The OOB error is the mean prediction error on each training sample x_i, using only the trees that did not have x_i in their bootstrap sample. The variable importance measures produced by random forests can be used for model reduction (e.g., use the "important" variables to build simpler, more readily interpretable models) [17]. Feature selection techniques using decision trees such as random forest algorithm are classified as embedded techniques [3].

The 'randomForest' package provides two variable importance measure, i.e., mean decrease in accuracy (MDA) and mean decrease in impurity (MDI). The first measure is computed from permuting out-of bag (OOB) data: For each tree, the prediction error (MSE for regression) on the out-of-bag portion of the data is recorded. Then the same is done after permuting each predictor variable. The difference between the two are then averaged over all trees, and normalized by the standard deviation of the differences. In other words, the random forest algorithm estimates the importance of a variable by looking at how much prediction error increases when OOB data for that variable is permuted while all others are left unchanged. The necessary calculations are carried out tree by tree as the random forest is constructed [18]. The rationale of the original random forest permutation importance is as follows: By randomly permuting the predictor variable X_j, its original association with the response Y is broken. When the permuted variable X_j, together with the remaining non-permuted predictor variables, is used to predict the response for the OOB observations, the prediction accuracy (i.e. the number of observations classified correctly) decreases substantially if the original variable X_j was associated with the response [19]. Strobl *et al.* (2008) formalize this idea as follows: Let $\overline{\mathcal{B}}^{(t)}$ be the OOB sample for a tree $t \in \{1, \ldots, N_t\}$ where N_t is the number of decision trees. The variable importance of variable X_j in tree t is

$$VI^{(t)}(X_j) = \frac{\sum_{i \in \overline{\mathcal{B}}^{(t)}} I\left(Y_i = \hat{Y}_i^{(t)}\right)}{|\overline{\mathcal{B}}^{(t)}|} - \frac{\sum_{i \in \overline{\mathcal{B}}^{(t)}} I\left(Y_i = \hat{Y}_{i,\pi_j}^{(t)}\right)}{|\overline{\mathcal{B}}^{(t)}|} \quad (4)$$

where $\hat{Y}_i^{(t)}$ and $\hat{Y}_{i,\pi_j}^{(t)}$ are the predicted value for observation i before and after permuting its value of variable X_j, I is the squared residual (for regression), and $\left|\overline{\mathcal{B}}^{(t)}\right|$ is the number of OOB samples for tree t. By definition, $VI^{(t)}(X_j) = 0$ if variable X_j is not in tree t. The raw variable importance score for variable X_j is then computed as the mean importance over all trees,

$$VI(X_j) = \frac{1}{N_t} \sum_{t=1}^{N_t} VI^{(t)}(X_j) \quad (5)$$

In the 'randomForest package, $VI(X_j)$ is normalized by dividing $VI(X_j)$ by the standard deviation of $VI(X_j)$ to obtain the so-called *z-score*,

$$z_j = VI(X_j) \Big/ \left(\frac{\hat{\sigma}}{\sqrt{N_t}}\right) \quad (6)$$

If $\hat{\sigma} = 0$ for a variable, the division is not done; nevertheless, $VI(X_j)$ is almost always equal to 0 in that case [18].

The second measure is the total decrease in node impurities (or total increase in node purities) from splitting on the variable X_j, averaged over all trees. Summarized from Louppe *et al.* (2013), internal nodes τ are labeled with a binary test (or split) s_τ dividing their subset in two subsets corresponding to their two children τ_L and τ_R, while the terminal nodes (or leaves) are labeled with a best guess value of the output variable. A tree is built from a learning sample of size N which identifies the split s_τ for which the partition of the N_τ node samples into τ_L and τ_R maximizes the decrease

$$\Delta i(s_\tau, \tau) = i(\tau) - p_L i(\tau_L) - p_R i(\tau_R) \quad (7)$$

of some impurity measure $i(\tau)$, and where $p_L = N_{\tau_L}/N_\tau$, and $p_R = N_{\tau_R}/N_\tau$. The importance of a variable X_j for predicting Y is evaluated by adding up the weighted impurity decreases $p(\tau)\Delta i(s_\tau, \tau)$ for all nodes τ where X_j is used in the split s_τ, averaged over all N_t trees in the forest T,

$$VI(X_j) = \frac{1}{N_t}\sum_{t=1}^{N_t}\sum_{\tau\in T:\, v(s_\tau)=X_j} p(\tau)\Delta i(s_\tau,\tau) \quad (8)$$

where $p(\tau) = N_\tau/N$ is the proportion of samples reaching τ and $v(s_\tau)$ is the variable used in split s_τ [20]. For regression, the node impurity is measured by residual sum of squares (RSS) [18].

2.5 Sequential forward floating selection

Sequential feature selection methods are classified as wrapper techniques [3]. These algorithms search for the best set of features by adding to and/or removing a small number of features at a time from the current feature set until the required value of an optimality criterion is obtained. The starting point of the search can be either an empty set which is then successively built up or the starting point can be the complete set of features in which unnecessary features are successively removed. The former approach is referred to as the 'bottom up' search while the latter is referred to as the 'top down' search. An example of the 'top down' search is the sequential backward selection (SBS) introduced by Marill and Green (1963) and the 'bottom up' search is the sequential forward selection (SFS) introduced by Whitney (1971). Both methods are generally suboptimal and suffer from the so-called 'nesting effect'. The term 'nesting effect' here is used to describe, in the case of the 'top down' search, that the discarded features cannot be reselected while in the case of the 'bottom up' search, the features once selected cannot later be discarded. The floating search methods proposed by Pudil *et al*. (2014) are intended to overcome the problem of 'nesting effect' among other things. They showed their performance to be very good compared with other search methods. The search in the forward direction is referred to as the sequential forward floating selection (SFFS), while in the opposite direction is referred to as the sequential backward floating selection (SBFS) [21].

Suppose that k features have already been selected from the complete set of features, $Y = \{y_j | j = 1,2,\dots,D\}$, where D is the total number of features, to form the set X_k with the corresponding criterion function $J(X_k)$. Suppose also that the values of $J(X_i)$ for all preceding subsets of size $i = 1,2,\dots,k-1$ have been computed and recorded.

1. *Step* 1 (*Inclusion*). Using the basic SFS method, select feature x_{k+1} from the set of available features, $Y - X_k$, to form the set X_{k+1}, i.e., the most significant feature x_{k+1} with respect to the set X_k is added to X_k. Therefore

$$X_{k+1} = X_k + x_{k+1}$$

2. *Step* 2 (*Conditional exclusion*). Find the least significant feature in the set X_{k+1}. If x_{k+1} is the least significant feature in the set X_{k+1}, i.e.

$$J(X_{k+1} - x_{k+1}) \geq J(X_{k+1} - x_j), \quad \forall j = 1,2,\dots,k$$

then set $k = k+1$ and return to Step 1, but if x_r, $1 \leq r \leq k$, is the least significant feature in the set X_{k+1}, i.e.

$$J(X_{k+1} - x_r) > J(X_k),$$

then exclude x_r from X_{k+1} to form a new set X'_k, i.e.

$$X'_k = X_{k+1} - x_r.$$

Note that $J(X'_k) > J(X_k)$ now. If $k = 2$ then set $X_k = X'_k$ and $J(X'_k) = J(X_k)$ and return to Step 1, else go to Step 3.

3. *Step* 3 *(Continuation of conditional exclusion).* Find the least significant feature x_s in the set X'_k. If $J(X'_k - x_s) \leq J(X_{k-1})$ then set $X_k = X'_k$ dan $J(X_k) = J(X'_k)$ and return to Step 1. If $J(X'_k - x_s) > J(X_{k-1})$ then exclude x_s from X'_k to form a reduced set X'_{k-1}, i.e.

$$X'_{k-1} = X'_k - x_s,$$

Set $k = k-1$. Now if $k = 2$ then set $X_k = X'_k$ and $J(X_k) = J(X'_k)$ and return to Step 1, else repeat Step 3.

The algorithm is initialized by setting $k = 0$ and $X_0 = \emptyset$, and the SFS method is applied until a feature set of cardinality 2 is obtained. Then the algorithm continues with Step 1.

2.6 Support vector regression

The first version of support vector machine (SVM) for regression, called support vector regression (SVR), was proposed by Drucker *et al*. (1997) [22]. Being a kernel-based learning method, it uses an implicit mapping of the input data into a high dimensional feature space defined by a kernel function, i.e., a function returning the inner product between the images of two data points (**x**, **y**) in the feature space. The learning then takes place in the feature space, provided the learning algorithm can be entirely be rewritten so that the data points only appear inside the dot products with other points. This is often referred to as the "kernel trick" [23]. When no further prior knowledge is available, the Gaussian radial basis function and the Bessel function of the first kind kernel are two of general purpose kernels typically used [24][25]. In the 'kernlab' package [26], these two kernels are given, respectively, by

$$k(\mathbf{x}, \mathbf{x}') = \exp(-\sigma\|\mathbf{x} - \mathbf{x}'\|) \quad (9)$$

$$k(\mathbf{x}, \mathbf{x}') = \frac{\text{Bessel}^n_{(v+1)}(\sigma\|\mathbf{x} - \mathbf{x}'\|)}{(\|\mathbf{x} - \mathbf{x}'\|)^{-n(v+1)}} \quad (10)$$

.

The scale, offset, degree, σ (sigma), n (order), and v (degree) are the the kernel parameters, while $\mathbf{x}$ and $\mathbf{x}'$ are two arbitrary vectors in the feature space. Xiang *et al*. (2013) showed that Bessel kernel function of the first kind has higher prediction accuracy and stronger generalization ability in SVR, which provides references for the kernel functions selection of SVR [27].

2.7 Proposed selection method

For the first simulated data set with 1000 SNPs, 2^{1000} possible combinations of SNPs constitute the complete search space, while for the second simulated data set with 10

000 SNPs, $2^{10\,000}$ possible combinations of SNPs constitute the complete search space. Thus, the search space has an exponential complexity $O(2^n)$ [6]. Prior to the SNP selection, the search space is reduced by assigning variable importance to all SNPs and remove less important SNPs.

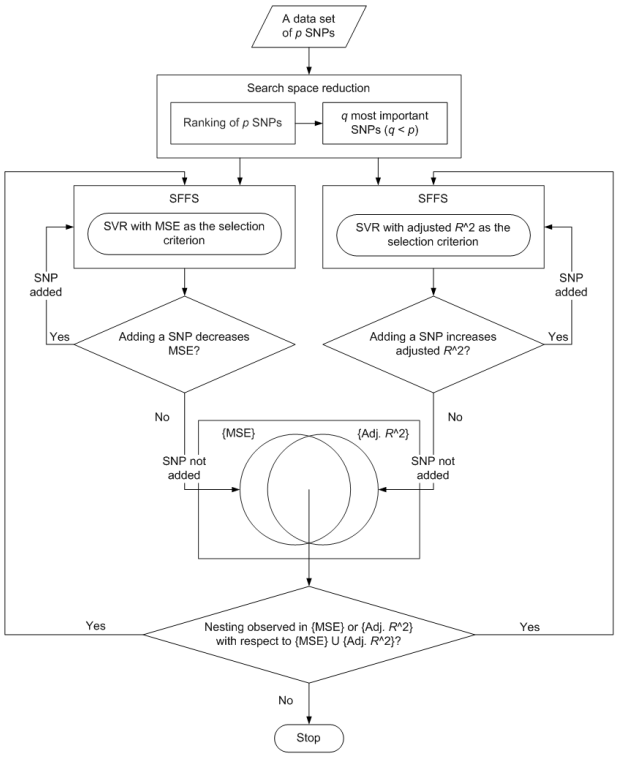

Fig. 2. Flowchart of the proposed selection method

After comparing the results of SNP ranking on two simulated data sets provided by two statistical approaches based on the CAR scores and the I-scores and one machine learning approach based on random forest algorithm, the best ranking method is selected. SNP ranking based on CAR scores is provided by the 'care' package [28], while SNP ranking based on I-scores is provided by our own code. SNP ranking by random forest is provided by the 'randomForest' package [18] utilizing parallel computing provided by the 'foreach' package to reduce the time needed to build the forest [29].

SNP selection is performed over the reduced search space using SFFS as described by Pudil *et al.* (1994). We wrote the code for SFFS by utilizing parallel computing provided by the 'foreach' package in order to reduce the search time. SNP subsets are evaluated using SVR provided by the 'kernlab' package and using, as selection or optimality criteria, the mean squared error,

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{Y}_i - Y_i\right)^2 \tag{11}$$

and the adjusted R^2

$$R^2_{adj} = 1 - (1 - R^2)\frac{n-1}{n-k-1} \tag{12}$$

where n is the number of samples, k is the number of current predictors, and R^2 is the coefficient of determination given by the square of Pearson's correlation coefficient of the actual response, Y, and the predicted response, $\hat{Y}$,

$$\rho = \frac{\mathrm{cov}\left(Y, \hat{Y}\right)}{\sigma_Y \sigma_{\hat{Y}}} \tag{13}$$

where $\mathrm{cov}\left(Y, \hat{Y}\right)$ is the covariance of Y and $\hat{Y}$, and σ_Y and $\sigma_{\hat{Y}}$ are standard deviations of Y and $\hat{Y}$, respectively. Adjusted R^2 is chosen over the coefficient of determination, R^2, which has the following drawback: R^2 increases with each addition of predictors to the model, and never decreases, so it is as if a better fit were obtained with the more terms added to the model, while in fact a model with too many terms will suffer from overfitting. But with the adjusted R^2, the addition of more and more useless variables to a model will decrease the adjusted R^2, and the addition of more useful variables will increase the adjusted R^2. The adjusted R^2 tells the percentage of variation explained only by the independent variables that actually affect the dependent variable [30].

We propose that the intersection of two variable subsets selected using the two selection criteria be considered the final selection result. This is motivated by our observation that by using only one selection criterion on simulated data sets results in a number of false positives while their intersection helps to reduce or even eliminate false positives. The adjusted R^2 and MSE are chosen as the two selection criteria in the hope that the selected SNPs are those with both high statistical significance and strong predictive power. Fig. 2 shows the flowchart of the proposed method.

In SVR training, all independent and dependent variables are scaled to zero mean and unit variance. This is done so that no variables would dominate others. Furthermore, 10-fold cross-validation is used on the training data to assess the quality of the models constructed by SVR and to avoid overfitting. The selection results obtained by the Gaussian RBF kernel and the Bessel kernel for simulated SNP data sets are compared. The 'kernlab' package provides linear, polynomial, Gaussian RBF, Laplace RBF, ANOVA RBF, Bessel function of the first kind, hyperbolic tangent, spline, and string kernels [26].

For this work we used a PC with Intel Core i5 M 480 @ 2.67 GHz, 4 GB RAM, and Windows 7 64-bit. We used R software version 3.4.1 using RStudio version 1.0.143 as the user interface [31].

3 Material

To assess the performance of our method, we chose two simulated data sets which were used by De Oliveira *et al*. (2014) and Kusuma *et al*. (2016). The first data set only has main effects without interaction among SNPs, while the second one has epistasis among SNPs. The two data sets were generated by the function simulateSNPglm of the 'scrime' package. The genotypes are categorical variables while the phenotypes are continuous variables. The states of a genotype is encoded as follows: 1 for homozygous reference genotype (both alleles of SNP are major homozygous), 2 for heterozygous genotype (two alleles of SNP are heterozygous), and 3 for homozygous variant genotype (both alleles of SNP are minor homozygous), where a minus sign before any of these numbers implies that the corresponding SNP does not affect the phenotype when it is of this genotype [9].

3.1 Simulated phenotype without epistasis

The linear regression model generated for simulated phenotype without epistasis is described by Eq. 14.

$$Y = \beta_0 + \sum_{i=1}^{7} \beta_i L_i + \text{error} \quad (14)$$

where $error$ is a normal random variable with mean 0 and standard deviation 5, $L_1 = (\text{SNP1} == 2)$, $L_2 = (\text{SNP10} == 1)$, $L_3 = (\text{SNP20} == 3)$, $L_4 = (\text{SNP30} == 3)$, $L_5 = (\text{SNP40} == 3)$, $L_6 = (\text{SNP50} == 2)$, $L_7 = (\text{SNP60} == 2)$, and Y is the simulated phenotype. The beta coefficients were set as $\beta_0 = 0$, $\beta_1 = \beta_2 = \beta_3 = 200$, $\beta_4 = 900$, $\beta_5 = \beta_6 = \beta_7 = 200$. The notation $L_1 = (\text{SNP1} == 2)$ means that if SNP1 is of genotype 2 then $L_1 = 1$, and $L_1 = 0$ otherwise. For this data set, 1000 markers were simulated for 250 subjects, with a minor allelic frequency (MAF), simulated for each SNP, based on a uniform distribution with minimum and maximum limits, respectively, 0.10 and 0.40 [6].

3.2 Simulated phenotype with epistasis

The linear regression model generated for simulated phenotype with epistasis is described by Eq. 15.

$$Y = \beta_0 + \sum_{i=1}^{3} \beta_i L_i + \text{error} \quad (15)$$

where $error$ is a normal random variable with mean 0 and standard deviation 1, $L_1 = (\text{SNP4} != 2)\ \&\ (\text{SNP3} != 1)$, $L_2 = (\text{SNP5} == 3)$, $L_3 = (\text{SNP12} != 1)\ \&\ (\text{SNP9} != 3)$ and Y is the simulated phenotype. The beta coefficients were set as $\beta_0 = 0$, $\beta_1 = \beta_2 = 150$, $\beta_3 = 40$. The notation $L_1 = (\text{SNP4} != 2)\ \&\ (\text{SNP3} != 1)$ means that if SNP4 is not of genotype 2 while at the same time SNP3 is not of genotype 1 then $L_1 = 1$. For this data set, 10 000 markers were simulated for 600 subjects, and MAF, simulated for each SNP, based on a uniform distribution with minimum and maximum limits as used on the simulated phenotype without epistasis [6].

4. Results and discussion

4.1 SNP selection over complete search space

First of all, we performed SNP selection over the complete search space of 1000 SNPs and 10 000 SNPs for simulated phenotype 1 and simulated phenotype 2, respectively, using a wrapper based on SFFS with MSE and adjusted R^2 as selection criteria which were evaluated using SVR. The intersection of the two SNP subsets obtained by using the two selection criteria was considered the final selection result. Based on our trial, we picked 10 as the value of the 'cost' parameter in SVR. The 'kernlab' package provides automatic tuning only for the parameter of the Gaussian and Laplace RBF kernel, sigma, so the Gaussian RBF kernel was the only kernel optimized in this work. The package uses heuristics in the function sigest to calculate a good sigma value for the Gaussian or Laplace RBF kernel, from the data. The parameters of the Bessel kernel—sigma, order, and degree—used their default unit values [26]. Furthermore, 10-fold cross-validation was used on the training data to assess the quality of the models constructed by SVR and to avoid overfitting.

A. Simulated phenotype without epistasis (simulated phenotype 1)

Tab. 2 shows the SNPs selected for simulated phenotype 1 using Gaussian RBF kernel. The selection criterion MSE obtained a variable subset of 7 true positives (TPs) and 1 false positive (FP), while the adjusted R^2 obtained a variable subset of 7 TPs and 2 FPs. The intersection of the two subsets was a subset of 7 TPs. Thus, all 3 FPs were discarded.

Table 2. Selection result for simulated phenotype 1 using Gaussian RBF kernel.

Selection Criterion	Selected SNPs	MSE	Adjusted R^2	Time (minutes)
MSE	**1**, **10**, **20**, **30**, **40**, **50**, 51, **60**	0.00564	-	-
Adjusted R^2	**1**, **10**, **20**, **30**, **40**, **50**, **60**, 91, 987	-	0.99578	-
Intersection	**1**, **10**, **20**, **30**, **40**, **50**, **60**	0.00589	0.99538	38.8

Tab. 3 shows the SNPs selected for simulated phenotype 1 using Bessel kernel. Both the selection criteria obtained 7 TPs. It is evident that the Bessel kernel outperforms the Gaussian RBF kernel for this data set. Note that it took less computational time with the Gaussian RBF kernel.

Table 3. Selection result for simulated phenotype 1 using Bessel kernel.

Selection Criterion	Selected SNPs	MSE	Adjusted R^2	Time (hour)
MSE	**1**, **10**, **20**, **30**, **40**, **50**, **60**	0.00633	-	-
Adjusted R^2	**1**, **10**, **20**, **30**, **40**, **50**, **60**	-	0.99350	-
Intersection	**1, 10, 20, 30, 40, 50, 60**	0.00633	0.99350	1

B. Simulated phenotype with epistasis (simulated phenotype 2)

Tab. 4 shows the SNPs selected for simulated phenotype 2 using Gaussian RBF kernel. The selection criterion MSE obtained a variable subset of 4 TPs and 2 FPs, while the adjusted R^2 obtained a variable subset of 3 TPs and 4 FPs. The intersection of the two subsets was a subset of 3 TPs. Thus, all 6 FPs were discarded and 1 TP was, unfortunately, also discarded.

Table 4. Selection result for simulated phenotype 2 using Gaussian RBF kernel.

Selection Criterion	Selected SNPs	MSE	Adjusted R^2	Time (hours)
MSE	**3**, **4**, **5**, **9**, 9955, 1667	0.00524	-	-
Adjusted R^2	**3**, **4**, **5**, 881, 7048, 7404, 2413	-	0.99426	-
Intersection	**3**, **4**, **5**	0.01039	0.99243	14.7

Tab. 5 shows the SNPs selected for simulated phenotype 2 using Bessel kernel. The selection criterion MSE obtained a variable subset of 5 TPs, while the selection criterion

adjusted R^2 obtained a variable subset of 3 TPs and 2 FPs. The intersection of the two subsets was a subset of 3 TPs. Thus, all 2 FPs were discarded and 2 TPs were, unfortunately, also discarded. Nevertheless, it is evident that the Bessel kernel performed better than the Gaussian RBF kernel.

Table 5. Selection result for simulated phenotype 2 using Bessel kernel.

Selection Criterion	Selected SNPs	MSE	Adjusted R^2	Time (hours)
MSE	**3**, **4**, **5**, **9**, **12**	0.00628	-	-
Adjusted R^2	**3**, **4**, **5**, 3115, 3719	-	0.99190	-
Intersection	**3**, **4**, **5**	0.01204	0.98833	19.7

The use of the intersection of the two variable subsets obtained by two selection criteria helps to reduce or even eliminate false positives. However, it may also discard true positives. Evidently the selection result for the simulated phenotype with epistasis was not yet satisfactory. Moreover, even though we have utilized parallel computing, the computational time for high-dimensional data sets were still pretty high. We then attempted to address these issues by considering the use of variable ranking methods to remove less important features prior to variable selection in the hope that it can help to achieve better selection result and reduce the computational time.

4.2 SNP ranking

SNP ranking based on squared CAR score for simulated phenotype 1 was computed in 1 to 2 seconds and the LHS of Fig. 3 shows 20 top-ranked SNPs. SNP40 was rather poorly ranked even though it is associated with the phenotype, while the other six SNPs were among the top 20 SNPs. For the simulated phenotype 2, the ranking was computed in 40 to 50 seconds and the RHS of Fig. 3 shows 20 top-ranked SNPs. SNP9 and SNP12 were poorly ranked even though both SNPs are associated with the phenotype, while the other three SNPs filled the top 4 positions.

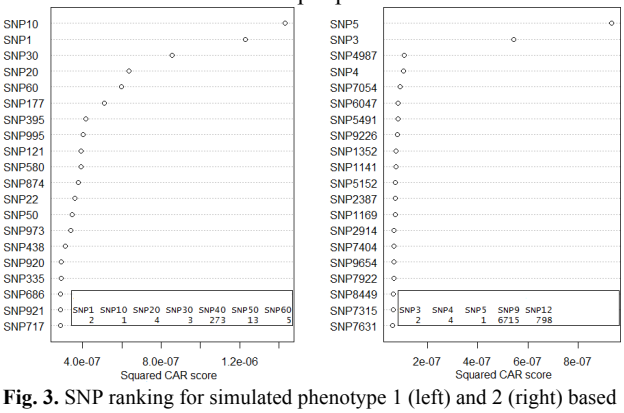

Fig. 3. SNP ranking for simulated phenotype 1 (left) and 2 (right) based on squared CAR score.

SNP ranking based on I-score for simulated phenotype 1 was also computed in 1 to 2 seconds and the LHS of Fig. 4 shows 20 top-ranked SNPs. SNP40 was poorly ranked and SNP30 was not among the top 20 SNPs, while the other five SNPs filled the top 5 positions. For the simulated phenotype 2, the ranking was computed in 18 to 19 seconds and the RHS of Fig. 4 shows 20 top-ranked SNPs. SNP9 and SNP12 were again poorly ranked even though both SNPs are associated with the phenotype, while the other three SNPs filled the top 3 positions.

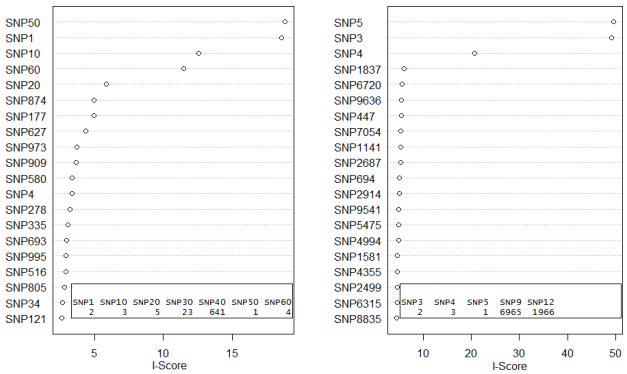

Fig. 4. SNP ranking for simulated phenotype 1 (left) and 2 (right) based on I- score

SNP ranking by random forest for simulated phenotype 1 was computed in approximately half a minute. We used the default number of trees in the 'randomForest' package, which is 500 [18], for all data sets. As for the 'mtry' parameter, if one has a very large number of variables but expects only very few to be "important", using larger 'mtry' may give better performance [17]. We observed that most of the time the 'mtry' parameter optimization gave 1000 as the optimum value which is exactly the number of SNPs in the data set. In the case of SNP40, ranking based on the percent increase in MSE (%IncMSE) resulted in it having a wide range of relatively poor rank in different computation, while ranking based on the decrease in RSS (IncNodePurity) placed it at the top 200 positions most of the time. Fig. 5 shows an instance of the top 20 SNPs for this data set.

SNP ranking by random forest for simulated phenotype 2 was computed in approximately 26 minutes. We observed that most of the time the 'mtry' parameter optimization gave 10 000 as the optimum value which again is exactly the number of SNPs in the data set. Using either of the two measures, the ranking is highly favorable for search space reduction (Fig. 6).

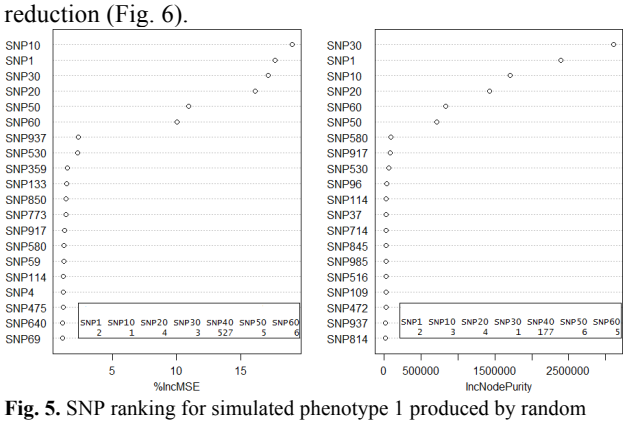

Fig. 5. SNP ranking for simulated phenotype 1 produced by random forest.

It is evident that random forest algorithm produced better SNP ranking for each of the two simulated phenotypes than either the CAR scores or I-scores, but with the longest computational time. Between the two measures of variable importance in random forest, IncNodePurity seems slightly more favorable for search space reduction.

4.3 SNP selection over reduced search space

Prior to SNP selection, all SNPs were ranked according to the total increase in node purities (IncNodePurity) from splitting on the variable, averaged over all trees. For

regression, the node impurity is measured by residual sum of squares (RSS). SNP selection over each of the two simulated data sets was then performed over the reduced search space of 200 highest-ranked SNPs. Note that we cannot yet offer a theoretical basis to determine the minimum number of top-ranked SNPs that should be included in the search. However, it is natural that the more top-ranked SNPs are included, the more likely it is to find all the relevant SNPs but with longer search times.

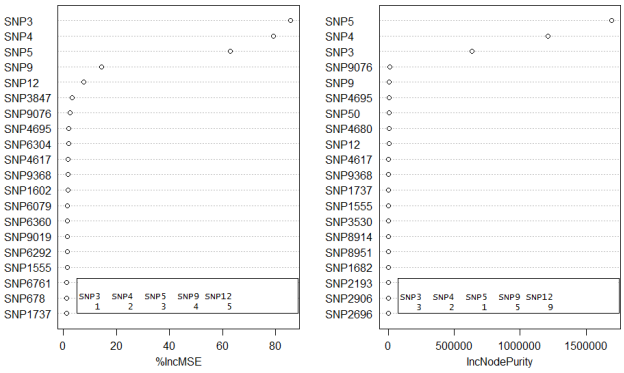

Fig. 6. SNP ranking for simulated phenotype 2 produced by random forest.

A. Simulated phenotype without epistasis (simulated phenotype 1)

Tab. 6 shows the final selection results for simulated phenotype 1. Both Gaussian RBF kernel and Bessel kernel gave perfect results with neither false positives nor false negatives. Note that it took less computational time with the search space reduction.

Table 6. Selection results for simulated phenotype 1 using Gaussian RBF and Bessel kernel.

Kernel	Selected SNPs	MSE	Adjusted R^2	Time (minutes)
RBF Gauss	**1**, **10**, **20**, **30**, **40**, **50**, **60**	0.00561	0.99539	6.8
Bessel	**1**, **10**, **20**, **30**, **40**, **50**, **60**	0.00633	0.99350	12.8

B. Simulated phenotype with epistasis (simulated phenotype 2)

Tab. 7 shows the final selection results for simulated phenotype 2 using Gaussian RBF kernel. We obtained three different selection results for different data sets generated at different times.

Table 7. Three instances of selection results for simulated phenotype 2 using Gaussian RBF kernel.

Result	Selected SNPs	MSE	Adjusted R^2	Time (minutes)
1	**3**, **4**, **5**	0.01026	0.99239	45.4
2	**3**, **4**, **5**, **9**, 9955	0.00775	0.99270	49.4
3	**3**, **4**, **5**, **9**, **12**, 4660	0.00770	0.99270	49.0

Tab. 8 shows the final selection results for simulated phenotype 2 using Bessel kernel. We obtained the same good selectin results for different data sets generated at different times. This observation supports the assertion of Xiang *et al*. (2013) that the Bessel kernel function of the first kind has higher prediction accuracy and stronger generalization ability in SVR [27].

Table 8. Selection result for simulated phenotype 2 using Bessel kernel.

Result	Selected SNPs	MSE	Adjusted R^2	Time (minutes)
1	**3**, **4**, **5**, **9**, **12**	0.00974	0.99103	56.6

During the search, we had an interesting observation in which the proposed method helped to overcome the 'nesting effect'. At first, the selection criterion MSE gave a subset of 5 TPs while the adjusted R^2 produced a nested FP SNP2256 which the SFFS was unable to remove through backward elimination. The intersection of the two subsets—a subset with 4 TPs SNP5, SNP3, SNP4, and SNP9 as members—was then used as a new starting point of the search. The final selection result with 5 TPs means that the corresponding 'nesting effect' was dealt with the successful removal of SNP2256.

5. Conclusions

Feature selection SFFS with two selection criteria, adjusted R^2 and mean squared error, was shown to give better results for the two simulated phenotypes than those given by the methods proposed by De Oliveira *et al*. (2014) and Kusuma *et al*. (2016). Using the two selection criteria in the hope that the selected SNPs are those with both high statistical significance and strong predictive power and considering the intersection of the two SNP subsets obtained by the two selection criteria as the selection result were shown to help in reducing or even eliminating false positives. However, it may also discard true positives and the selection result for the simulated phenotype with epistasis was not satisfactory. Moreover, even though we have utilized parallel computing, the computational time for the two high-dimensional data sets were still pretty high.

These issues were succesfully resolved by employing a variable ranking method to remove less important SNPs prior to SNP selection. Our comparison of variable ranking methods using the two simulated phenotypes suggested that random forest algorithm outperforms the two statistical approaches, i.e., the CAR score and the I-score. Nevertheless, we see the need to find or develop a better variable ranking method. The reduction of the search space with exponential complexity was shown to improve the selection result for the simulated phenotype with epistasis and greatly reduce the search time. Unfortunately, we cannot yet offer a theoretical basis to determine the minimum number of highest-ranked SNPs that should be included in the search.

Testing with the two simulated phenotypes suggests that SVR with the non-optimized Bessel kernel is more favorable for SNP selection than the optimized Gaussian RBF kernel although with longer search time. This observation supports the assertion of Xiang *et al*. (2013) that the Bessel kernel function of the first kind has higher prediction accuracy and stronger generalization ability in SVR.

The proposed method, however, does not tell the epistatic interaction, if any, between the selected SNPs. Further research may be devised to develop the proposed

selection method so as to be capable of inferring any epistatic interaction involved.

Availability of data and materials

R scripts for generating the genotype and phenotype from a linear model with epistasis (simulation 1) and without epistasis (simulation 2), SNP ranking using random forest, CAR score, and I-score, and R scripts for the feature selection algorithm are available upon request to the corresponding author.

Competing interests

The authors declare that they have no competing interests.

References

[1] A. S. Foulkes, Applied Statistical Genetics with R, Springer, New York (2009).

[2] R. Rieger, A. Michaelis, and M. M. Green, A glossary of genetics and cytogenetics: Classical and molecular, New York: Springer-Verlag (1968).

[3] Y. Saeys, I. Inza, and P. Larranaga, A review of feature selection techniques in bioinformatics, Bioinformatics vol. 23 no. 19 pages 2507-2517 (2007).

[4] I. Guyon and A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research 3: 1157-1182 (2003).

[5] T. M. Cover and J. M. Van Capenhout, On the possible orderings in the measurement selection problem, IEEE Transactions on Systems, Man, and Cybernetics vol. SMC-7 no. 9 (1977).

[6] F. C. De Oliveira, C. C. H. Borges, F. N. Almeida, F. F. e Silva, R. D. S. Verneque, M. V. G da Silva, and W. Arbex, SNPs selection using support vector regression and genetic algorithms in GWAS, BMC genomics 15 (2014).

[7] D. Zongker and A. Jain, Algorithms for feature selection: an evaluation, Proceedings of the 13th International Conference on Pattern Recognition (1996).

[8] W. A. Kusuma, L. S. Hasibuan, and M. A Istiadi, SNPs selection using gravitational search algorithm and exhaustive search for association mapping, IOP Conference Series Earth and Environmental Science, 31(1): 012015 (2016).

[9] H. Schwender and A. Fritsch, Package 'scrime', CRAN Repository (2015).

[10] I. Ilhan, Y. E. Goktepe, and S. Kahramanli, A genetic algorithm-support vector machine method for selecting tag single nucleotide polymorphisms, International Journal of Innovative Computing, Information and Control vol. 9 no. 2 (2013).

[11] A. Lo, H. Chernoff, T. Zheng, and S. H. Lo, Why significant variables aren't automatically good predictors, Proceedings of the National Academy of Sciences 112(45): 13892-13897 (2015).

[12] V. Zuber, A. P. D. Silva, and K. Strimmer, A novel algorithm for simultaneous SNP selection in high-dimensional genome-wide association studies, BMC Bioinformatics 13:284 (2012).

[13] V. Zuber and K. Strimmer K, High-dimensional regression and variable selection using CAR scores, Statistical Applications in Genetics and Molecular Biology 10: 34 (2011).

[14] H. Chernoff, S. H. Lo, and T. Zheng, Discovering influential variables: a method of partitions, The Annals of Applied Statistics 3(4): 1335-1369 (2009).

[15] A. Lo, H. Chernoff, T. Zheng, and S. H. Lo, Framework for making better predictions by directly estimating variables' predictivity, Proceedings of the National Academy of Sciences (113)50: 14277–14282 (2016).

[16] L. Breiman, Random forests, Machine Learning 45(1): 5-32 (2001).

[17] A. Liaw and M. Wiener, Classification and regression by randomForest, R News vol. 2/3 pp. 18-22 (2002).

[18] A. Liaw and M. Wiener, Package 'randomForest' , CRAN Repository (2015).

[19] C. Strobl, A. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis, Conditional variable importance for random forests, BMC Bioinformatics 9:307 (2008).

[20] G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts, Understanding variable importances in forests of randomized trees, Advances in Neural Information Processing Systems 26 vol. 1 page 431-439 (2013).

[21] P. Pudil, J. Novovičová, and J. Kittler, Floating search methods in feature selection, Pattern Recognition Letters 15: 1119-1125 (1994).

[22] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, Support vector regression machines, Advances in Neural Information Processing Systems 9:155-161 (1997).

[23] B. Schölkopf and A. J. Smola, Learning with Kernels, MIT Press, Cambridge (2002).

[24] A. Karatzoglou, A. Smola, and K. Hornik, Kernlab: an S4 package for kernel methods in R, Journal of Statistical Software 11(9): 1-20 (2004).

[25] A. Karatzoglou, D. Meyer, and K. Hornik, Support Vector Machines in R, Journal of Statistical Software vol. 15 issue 9 (2006).

[26] A. Karatzoglou, A. Smola, and K. Hornik, Package 'kernlab', CRAN Repository (2016).

[27] L. Xiang, Z. Quanyin, and W. Liuyang, Research of bessel kernel function of the first kind for support vector regression, Information Technology Journal 12(14): 2673-2682 (2013).

[28] V. Zuber and K. Strimmer, Package 'care', CRAN Repository (2015).

[29] S. Weston, Using The foreach Package, CRAN Repository (2015).

[30] J. Miles, R squared, adjusted R squared, Wiley StatsRef: Statistics Reference Online (2014).

[31] R Core Team. R: A language and environment for statistical computing, R Foundation for Statistical Computing, Vienna, Austria (2017).