Indian J. Genet., 80(4) 395-401 (2020)
DOI: 10.31742/IJGPB.80.4.4

Comparative study of different non-parametric genomic selection methods under diverse genetic architecture

Neeraj Budhlakoti*, Anil Rai, D. C. Mishra, Seema Jaggi, Mukesh Kumar and A. R. Rao

ICAR-Indian Agricultural Statistics Research Institute, Pusa, New Delhi 110 012

(Received: August 2020; Revised: October 2020; Accepted: November 2020)

Abstract

Genomic Selection (GS) is the most prevalent method in today's scenario to access the genetic merit of individual under study. It selects the candidates for next breeding cycle on the basis of its genetic merit. GS has successfully been used in various plant and animal studies in last decade. Several parametric statistical models have been proposed and being used successfully in various GS studies. However, performance of parametric methods becomes very poor when we have non additive kind of genetic architecture. In such cases, generally performance of non-parametric methods are quite satisfactory as these methods do not require strict statistical assumptions. This article presents comparative performance of few most commonly used non-parametric methods for complex genetic architecture i.e. non-additive, using simulated dataset generated at different level of heritability and varying combination of population size. Among several non-parametric methods, SVM outperformed across a range of genetic architecture.

Key words: Genomic selection, epistasis, nonparametric, SVM and ANN.

Introduction

Genomic Selection (GS) is most popular method in plant science where dense genomic markers information is used to access merit of an individual. It is also a most popular technique for improving genetic gain of individuals now a days. The technique was first introduced and implemented by Meuwissen et al. (2001). They estimated individual effect of each marker by using appropriate statistical model, further sum of these markers effect is used for calculation of genetic merit of an individual. GS has successfully been used in various plant and animal studies during last decade. Several parametric statistical models have been proposed and are being used in various GS studies. These methods require some set of assumptions need to be hold on data in order to fit the model. But that may not always be the case; in such scenario performance of parametric methods is not quite encouraging. Non- parametric methods may perform better in this case as they do not assume any specific distribution of response and predictors.

As genetic architecture of plant is very complex in nature, so performance of GS methods become very poor in such cases, as they could not be able to model marker variance. As technology advances, it become cheap to generate genomic data on large scale resulting in availability of huge marker data. Further, due to huge number of epistatic interaction it becomes challenging to practice parametric methods (Moore and Williams 2009). In epistatic interaction, a number of loci are involved and also the possibility of interaction cannot be ignored. Epistatic interaction may play a crucial role for explaining genetic variation for quantitative traits, as ignoring these kind of interaction in the model may end up with lower genomic prediction accuracy (Cooper et al. 2002). Gianola et al. (2006) first used non-parametric and semi-parametric methods for modeling complex genetic architecture, as they also include such type of higher order interaction in these models. Subsequently, several statistical methods were implemented to model both main and epistasis effects for genomic selection (Cai et al. 2011, Xu 2007). Recently, some semi-parametric (Legarra et al. 2018) and other robust approaches (Tanaka 2018; Guha et al. 2019; Budhlakoti et al. 2020a; Budhlakoti et al. 2020b; Sehgal et al. 2020) have also been proposed and implemented in genomic selection.

*Corresponding author's e-mail: neeraj35669@gmail.com

Published by the Indian Society of Genetics & Plant Breeding, A-Block, F2, First Floor, NASC Complex, IARI P.O., Pusa Campus, New Delhi 110 012; Online management by www.isgpb.org; indianjournals.com

In this article, existing non-parametric models are reviewed, evaluated and their performance on simulated dataset has been studied. We have simulated genomic markers data and phenotype data for epistatic genetic architecture at different level of heritability (i.e. small, medium and high) with different combination of population size. Performance evaluation of different non-parametric methods i.e., RKHS, SVM, ANN and RF) has been done and appropriate model for different genetic architecture has been advocated.

Materials and methods

Several non-parametric methods have been studied in relation to genomic selection e.g., Reproducing Kernel Hilbert Space (RKHS), Support Vector Machine (SVM), Neural Network (NN) and Random Forest (RF). Each of these has been discussed one by one. In these non-parametric methods, SVM, NN and Random Forest are based on supervised machine learning based approach i.e. model is trained when labelled dataset is available (where input and output both are available). Here training dataset may consist of huge number of predictors (e.g., SNPs, Xi, where x refers to a vector containing genotypes of all SNPs for ith plants or animal) to predict the value of target phenotype (yi, that may be grain yield, thousand grain weight).

Reproducing Kernel Hilbert Space (RKHS)

This is based on what we called semi-parametric approach. It combines the merits of both non-parametric model with a mixed model frame work by Gianola et al. (2006). RKHS model can be expressed as

$$Y_i = w_i'\beta + z_i'u + g\left(X_i\right) + e_i$$

where $i = 1, 2, ..., n$ and β is a vector of fixed unknown effects (e.g. may be physical location of an individual or herd effect), u is a q x 1 vector which represent additive genetic effects, w_i and z_i are known vectors to be estimated, $g(X_i)$ function of the SNP data which is unknown and residuals e assumed to be normally $N(0, I\sigma^2)$ distributed.

Support Vector Machine (SVM)

The SVM is a machine learning based method, proposed by Vapnik et al. (1995). It is based on principle of maximum separating hyperplane. It constructs a hyperplane with the objective of separating data into different class. In case a problem is based on regression instead of classification i.e., when output data is continuous then the Support Vector Regression can be used. Support Vector Regression is an important application of SVM technique. In order to understand this, consider a mapping function $f(X)$: $R^p \rightarrow R$, given the set of training data

$$(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n), X_i \in R^p, Y_i \in R$$

Let us assume a simple linear function of following form:

$$f(X) = w' X + b$$

where, w is vector of weight to be estimated (i.e., regression coefficients) and b denotes bias. $f(X)$ is minimized by the following problem formulation:

$$\min_{w,b} \varnothing(w,b) = \frac{1}{2} \| w \|^2 + c\sum_{i=1}^{n} e_i^k$$

where, $e_i = Y_i - f(X_i)$ is error of i^{th} data point from training set or also known as loss function L(.) which measures quality of estimation and C represents regularization parameter which handles trade-off maximizing margin and minimizing of error term. For time being there are several choice of loss functions are available for SVM regression, among them most frequently used one are absolute loss, squared loss and ε-insensitive loss

Absolute loss: $L(Y - f(X) = |Y - f(X)|$

Squared loss: $L(Y - f(X) = |Y - f(X)|^2$

ε-insensitive loss: $L(Y - f(X)) = \begin{cases} 0 \quad \text{if} \quad |Y - f(X)| < \varepsilon \\ Y - f(X) \text{ otherwise} \end{cases}$

Where ε determines number of support vector in regression function. Here our focus is on ε-insensitive loss which minimizes

$$\frac{1}{2} \| w \|^2 + c\sum_{i=1}^{n} (\xi_{1i} + \xi_{2i})$$

Where ξ_{1i} and ξ_{2i} are slack variables subject to condition $\xi_{1i} > Y_i - f(X_i)$ and $\xi_{2i} > f(X_i) - Y_i - \varepsilon$. Solution to this minimization has following form (Nocedal and Wright 1999)

$$\hat{f}(X) = \sum_{i=1}^{n} \alpha_1 X_1 X + b$$

This solution solely depends on training data in form of inner product (X_i, X). So to takes advantage of higher dimensional space and one can use kernel trick to replace inner product:

$$k(X_i, X_j) = (\phi((X_i), \phi((X_i))$$

Kernel trick is usually a method of solving non-linear problem via linear classifier. It transforms linearly inseparable data to a linearly separable ones. The kernel function maps the original non-linear observations into a higher-dimensional space where these observations are linearly separable. There are several choice of kernel function are available e.g. linear kernel, Radial Basis Function kernel, Gaussian kernel etc. Choice of particular kernel function is data dependent and appropriate choice of underlying parameter requires extensive tuning.

Polynomial Kernel	$k(X_i, X_j) = (X_i, X_j + 1)^d$
RBF Kernel	$k(X_i, X_j) = \exp(-\sigma\|(X_i - X_j\|)$
Gaussian RBF Kernel	$k(X_i, X_j) = \exp(-\sigma\|(X_i - X_j\|^2)$
Sigmoid Kernel	$k(X_i, X_j) = \tanh(aX_i\ X_j + c)$

Neural network (NN)

A neural network is a circuit of neurons just like our brain. In every neural network it has an input and output inputs are connected to output through a connections and every connections have its own weight. During training phase neural network adjust the weights and learn from its previous error so that it can correctly classify or predict the desired output. Structure of neurons, basically consist of a) neurons b) node and c) bias.

The basic layout of the NN is a two-stage network with three types of layers: an input layer; a hidden layer; and an output layer. This model is called the feed-forward NN. In the hidden layer (not directly observed), one data-derived predictor (or basis function) is inferred at each of M neurons. These data derived predictors are formed by first inferring a score (U_{mi}) which is a linear combination of the input variables (marker genotypes, in our case), and then transforming this score using a non-linear activation function ϕ(.), that is $z_{mi} = \varphi_m(u_{mi}) = \varphi_m\left(w_{m0} + \sum_{j=1}^{p} x_{ij} w_{mj}\right)$, where U_{m0} is an intercept (also called as 'bias' term), and $w_m = \left\{w_{mj}\right\}_{m=1;j=1}^{m=M;j=p}$ is a vector of regression coefficients (i.e. weights). Likewise, in output layer phenotypes are regressed on the calculated parameters, $\left\{z_{mi}\right\}_{i=1;m=1}^{i=n;m=M}$, according to $y_i = \varphi\left(w_0 + \sum_{m=1}^{M} z_{mi} w_m\right) + e_i$, where ϕ(.) is usually a linear activation function and e_i is a model residual. Given the net input to a unit, the output of that unit is computed as:

$$z_{mi} = \varphi_m(u_{mi}, \theta) = \frac{1}{1 + \exp(-\theta u_{mi})}$$

where θ is a parameter controlling the shape of the activation function. This function try to move from 0 to 1 when the input x is greater than a certain value.

Random forest (RF)

Random forest is a group of binary tress constructed using recursive partitioning (RPART). The basic unit of random forest is known as a binary tree. A RF tree is built using CART (classification and regression tree) procedure. CART is recursive method which partition the tree into homogeneous or if not possible to a near-homogeneous terminal nodes. RF is often a collection of few thousands tree, where every single tree is developed using a bootstrap sample from original data. RF trees usually differ from CART as they are grown non deterministically using a two-stage randomization procedure. In order to find the best split for the node, RF selects at each and every node of each tree, a random subset of variables and only those variables are used further as a selection candidates. The purpose of this two-step randomization is to decorrelate trees so that the forest ensemble will have low variance, i.e., bagging phenomenon

Random forest algorithm usually encompasses the following phases:

1. Create bootstrap samples (i.e. *ntree*) from the original data.
2. Develop a tree from each bootstrap data set. At each node of the tree, randomly select variables (i.e., mtry generally *mtry*=p/3) for splitting. Develop the tree in such a manner that each terminal node has no fewer than *nodesize* cases.
3. Combine information from the trees (i.e., *ntree*) for prediction of test data set.
4. Estimate out-of-bag (OOB) error rate by using the test set i.e., data not used as a bootstrap sample.

Simulation study

For the purpose of illustration, simulated data were generated using QTL Bayesian interval mapping "qtlbim" (Yandell et al. 2012), a R based (R Development Core Team 2019) package. This package uses the Cockerham's model as the fundamental genetic model and has been trailed in many studies in the context to genomic selection (Yi et al. 2007; Yi and Shriner 2008; Piao et al. 2009; Yandell et al. 2012, Howard et al. 2014).

In present study, a total of twelve data sets have been simulated for genotypic and phenotype information. The data sets were simulated at four different heritability (0.1, 0.3, 0.5 and 0.7) and three different levels of population size (n=200, 300 and 500). Thus a range of diversified genetic architecture i.e. with very low heritability 0.10, 0.3 to medium 0.5 and high heritability 0.7 was created. For each stage the data has been simulated for 1000 SNPs for different combinations of population size (i.e. n=200, 300 and 500). Simulated data have 10 chromosomes with 100 SNPs in each with specified length. A total of 1000 markers are distributed over all the 10 chromosomes in such a way that each marker is equi-spaced over the chromosome. For epistatic architecture also the data has been simulated for epistatic effects of size 5 i.e., 5 two-way epistatic interactions among 10 pairs of loci with either positive or negative effects. In order to compare the performance of methods under study, cross validation techniques were used. Data is divided into two parts i.e., training and validation sets such that training set comprises of 70% data and validation set of 30%. Former one is used for model building and later one for model evaluation. The performance of methods was evaluated by calculating prediction accuracy and prediction error. Whole procedures is repeated 100 times and prediction accuracy and prediction error is calculated. In order to fit statistical model used in the study, R package STGS (Budhlakoti et al. 2019) is used for this purpose.

Evaluation measure

As an evaluation measure, prediction accuracy and prediction error were used. Prediction accuracy can be defined as pearson correlation coefficient between observed phenotypic value and predicted phenotypic value. Same can be expressed as

$$r = \frac{S_{Y,\hat{Y}}}{S_Y S\hat{Y}}$$

where $S_{Y,\hat{Y}}$ denotes the covariance between observed and predicted phenotypic value, S_Y is standard deviation of observed phenotype and $S_{\hat{Y}}$ denotes standard deviation of predicted phenotype. Prediction Error (PE) can be simply defined as mean sum of square error (MSE) between observed phenotypic value and predicted phenotypic value. Same can be expressed using following formula

$$PE / MSE = \frac{1}{2}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Where Y_i is observed response; $\hat{Y}_i$ is predicted phenotype value of *i*th individual and *n* denotes total no. of individuals' in the test set.

Results and discussion

The performance of different non-parametric methods under study i.e., RKHS, SVM, ANN and RF at diverse level of heritability and population size has been discussed here. Prediction accuracy and MSE were used as an evaluation measure for different models. Results of the same have been presented in Tables 1 and 2.

Table 1. Mean of genomic prediction accuracy for different non-parametric methods under study using simulated dataset for different combination of population size (n) and various levels of heritability (h^2)

h^2/methods	Population size	RKHS	ANN	SVM	RF
0.1	n=200	0.54	0.57	0.59	0.46
	n=300	0.55	0.56	0.57	0.48
	n=500	0.57	0.62	0.62	0.54
0.3	n=200	0.53	0.72	0.75	0.63
	n=300	0.55	0.73	0.77	0.67
	n=500	0.58	0.75	0.79	0.68
0.5	n=200	0.55	0.82	0.85	0.70
	n=300	0.45	0.82	0.85	0.71
	n=500	0.53	0.86	0.89	0.74
0.7	n=200	0.62	0.84	0.88	0.72
	n=300	0.60	0.88	0.89	0.73
	n=500	0.59	0.90	0.91	0.77

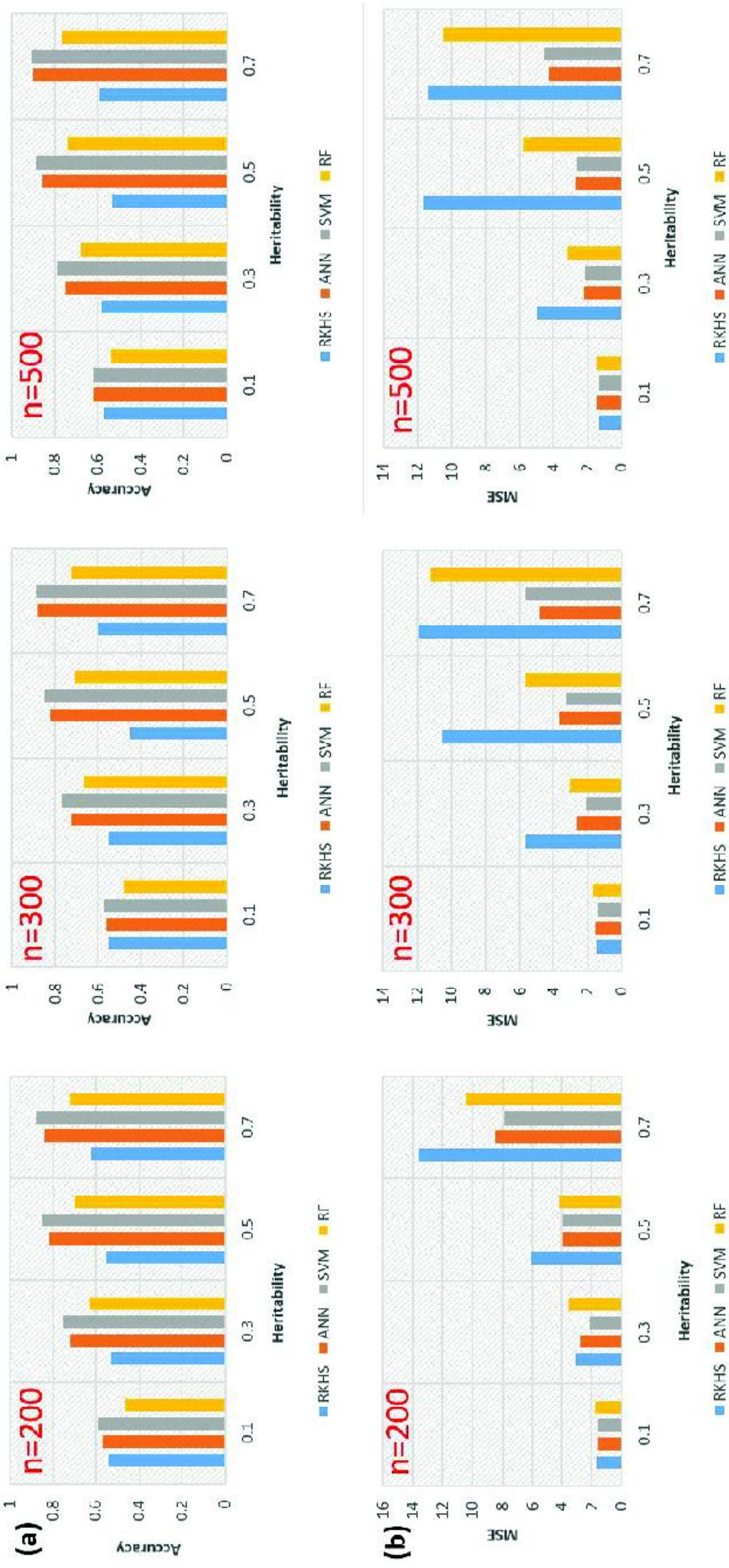

Fig. 1. Graphical representation results of different population size (n=200, 300 and 500) at diverse genomic architecture using different evaluation measure for methods under study i.e., a) Accuracy and b) MSE

Table 1 shows the prediction accuracy of different non-parametric methods under study. This table presents the average of prediction accuracy generated over 100 replications. Likewise, Table 2 shows the MSE of different non-parametric methods under study. These results are generated using simulated dataset at different levels of heritability (i.e., 0.1, 0.3, 0.5 and 0.7) for varying population size (i.e. 200, 300 and 500). Graphical representations of same is given in Fig. 1. Detailed discussion of results obtained at different heritability are discussed below.

At heritability 0.1

Table 1 clearly demonstrates the accuracy of ANN and SVM which is at par followed by RKHS, whereas performance for RF methods seems to be lowest among all methods. On the basis of MSE, a similar performance has been observed i.e., ANN and SVM at par followed by RKHS, whereas RF has highest MSE indicating poorest performance (Table 2).

At heritability 0.3

For this heritability, SVM is performing quite well as compared to other methods followed by ANN, RF and RKHS (Table 1). MSE also depicts the same picture, For SVM, MSE is lowest among all followed by ANN, RKHS and RF (Table 2).

At heritability 0.5

Here, the performance of SVM and ANN is at par followed by RF and RKHS with respect to

Table 2. Mean of genomic prediction error i.e., MSE for different non-parametric methods under study using simulated dataset for different combination of population size (n) and various levels of heritability (h^2)

h^2/methods	Population size	RKHS	ANN	SVM	RF
0.1	n=200	1.62	1.60	1.56	1.78
	n=300	1.42	1.52	1.38	1.63
	n=500	1.34	1.43	1.32	1.52
0.3	n=200	3.05	2.72	2.12	3.54
	n=300	5.62	2.64	2.10	3.02
	n=500	4.98	2.17	2.12	3.15
0.5	n=200	6.06	3.89	3.90	4.19
	n=300	10.5	3.64	3.20	5.69
	n=500	11.7	2.69	2.6	5.81
0.7	n=200	13.6	8.44	7.82	10.4
	n=300	11.9	4.81	5.64	11.2
	n=500	11.4	4.27	4.55	10.51

prediction accuracy (Table 1). Likewise, same pattern can be observed for MSE, SVM and ANN which has lowest MSE among all followed by RF and RKHS (Table 2).

At heritability 0.7

At this heritability level, SVM showed quite a good genomic prediction accuracy followed by ANN, RF and RKHS (Table 1). Also, at this heritability level, around 0.90 prediction accuracy was observed for SVM and ANN. Prediction error is also lowest for SVM followed by ANN, RF and RKHS (Table 2).

In general, it was also observed that with increase in population size, as a general trend, prediction accuracy is slightly increased and MSE decreased. This may be due to the reason that increased data size results in proper model fitting and training.

It can be concluded from the above results that performance of SVM is consistent throughout different levels of heritability with respect to prediction accuracy and MSE (Tables 1 and 2). However, ANN is also performing quite well, almost at par with SVM with increased population size. Performance of random forest is a bit poor at low heritability (h^2 0.1) however, it improves gradually with increase in heritability. Here, performance of RKHS is not found to be encouraging in comparison to their counterparts throughout the study.

Impact of genetic architecture on genomic prediction accuracy has been explored. In this study, comparative evaluation of most commonly used non-parametric methods for genomic selection has been done. We have presented the results of the same using most commonly used evaluation measures like prediction accuracy and MSE. Overall performance of SVM has been remarkable across the range of genetic architecture.

Authors' contribution

Conceptualization of research (NB, AR, ARR, SJ, MK); Designing of the experiments (NB, DCM, AR); Contribution of experimental materials (NB, DCM); Execution of field/lab experiments and data collection (NB, DCM); Analysis of data and interpretation (NB, DCM, AR); Preparation of manuscript (NB, DCM, AR).

Declaration

The authors declare no conflict of interest.

References

Budhlakoti N., Mishra D. C., Rai A. and Chaturvedi K. K. 2019. STGS: Genomic Selection using Single Trait. R package version 0.1.0. https://CRAN.R project.org/package=STGS.

Budhlakoti N., Rai A. and Mishra D. C. 2020a. Statistical Approach for improving Genomic prediction Accuracy through Efficient Diagnostic Measure of Influential Observation. Scientific Reports,**10**(1): 1-11.

Budhlakoti N., Rai A. and Mishra D. C. 2020b. Effect of influential observation in genomic prediction using LASSO diagnostic. Indian J. agric. Sci., **90**(6): 1155-1159.

Cai X., Huang A. and Xu S. 2011. Fast empirical bayesian lasso for multiple quantitative trait locus mapping. BMC Bioinformatics,**12**: 211. doi: 10.1186/1471-2105-12-211.

Cooper M, Podlich D W, Micallef K P, Smith O S, Jensen N M. 2002. Complexity, quantitative traits and plant breeding: a role for simulation modelling in the genetic improvement of crops. Quantitative Genetics, Genomics Plant Breed., 143-166.

Gianola D., de los Campos G., Hill W. G., Manfredi E. and Fernando R. 2009. Additive genetic variability and the bayesian alphabet. Genetics, **183**: 347-363.

Guha Majumdar S., Rai A. and Mishra D. C. 2019. Integrated Framework for Selection of Additive and

Nonadditive Genetic Markers for Genomic Selection. J. Comput. Biol., **26**: 1-11.

Howard R., Carriquiry A. L. and Beavis W. D. 2014. Parametric and nonparametric statistical methods for genomic selection of traits with additive and epistatic genetic architectures. G3, **4**: 1027-1046.

Hwang C. and Yoon K. 1981. Methods for Multiple Attribute Decision Making. *In*: Multiple Attribute Decision Making. Lecture Notes in Economics and Mathematical Systems, **186:** Springer, Berlin, Heidelberg.

Legarra A. and Reverter A. 2018. Semi-parametric estimates of population accuracy and bias of predictions of breeding values and future phenotypes using the LR method. Genet. Sel. Evol.,**50:** 53.

Meuwissen T. H. E., Hayes B. J. and Goddard M. E. 2001. Prediction of total genetic value using genome-wide dense marker maps. Genetics, **157**: 1819-1829.

Moore J. H. and Williams S. M. 2009. Epistasis and its implications for personal genetics. Am. J. Hum. Genet., **85**: 309-320.

Nocedal J. and Wright S. J. 1999. Numerical Optimization. Springer, New York.

Piao Z., Li M., Li P., Zhang J., Zhu C. et al. 2009. Bayesian dissection for genetic architecture of traits associated with nitrogen utilization efficiency in rice. African J. Biotechnol., **8**(24): 6834-6839.

R Core Team. 2019. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria, URL https://www.R-project.org/.

Sehgal D., Rosyara U., Mondal S., Singh R., Poland J. and Dreisigacker S. 2020. Incorporating Genome-Wide Association Mapping Results Into Genomic Prediction Models for Grain Yield and Yield Stability in CIMMYT Spring Bread Wheat. Front. Pl. Sci.,**11:** 197.

Tanaka E. 2018. Simple robust genomic prediction and outlier detection for a multi-environmental field trial. arXiv preprint arXiv:1807.07268.

Vapnik V. 1995. The Nature of Statistical Learning Theory, Ed. 2. Springer, New York.

Xu S. 2007. An empirical bayes method for estimating epistatic effects of quantitative trait loci. Biometrics, **63**: 513-521. doi. 10.1111/j.1541-0420. 2006.00711.

Yandell B. S. and Nengjun Y. 2007. R/qtlbim: QTL with Bayesian interval mapping in experimental crosses. Bioinformatics, **23:** 641-643.

Yi N., Shriner D., Banerjee S., Mehta T., Pomp D. et al. 2007. An efficient Bayesian model selection approach for interacting quantitative trait loci models with many effects. Genetics, **176:** 1865-1877.

Yi N. and Shriner D. 2008. Advances in Bayesian multiple QTL mapping in experimental designs. Heredity, **100:** 240-252.