Published April 12, 2018

ORIGINAL RESEARCH

Applications of Machine Learning Methods to Genomic Selection in Breeding Wheat for Rust Resistance

Juan Manuel González-Camacho, Leonardo Ornella, Paulino Pérez-Rodríguez, Daniel Gianola, Susanne Dreisigacker, and José Crossa*

ABSTRACT

New methods and algorithms are being developed for predicting untested phenotypes in schemes commonly used in genomic selection (GS). The prediction of disease resistance in GS has its own peculiarities: a) there is consensus about the additive nature of quantitative adult plant resistance (APR) genes, although epistasis has been found in some populations; b) rust resistance requires effective combinations of major and minor genes; and c) disease resistance is commonly measured based on ordinal scales (e.g., scales from 1–5, 1–9, etc.). Machine learning (ML) is a field of computer science that uses algorithms and existing samples to capture characteristics of target patterns. In this paper we discuss several state-of-the-art ML methods that could be applied in GS. Many of them have already been used to predict rust resistance in wheat. Others are very appealing, given their performance for predicting other wheat traits with similar characteristics. We briefly describe the proposed methods in the Appendix.

Core Ideas

- Genomic-enabled prediction
- Machine learning
- Wheat breeding
- Rust resistance

THE DEVELOPMENT of low-cost genotyping strategies such as single nucleotide polymorphisms (SNP) and genotyping-by-sequencing (GBS) (Elshire et al., 2011; Kumar et al., 2012) has made it possible for genomic selection (GS) to offer new possibilities for improving the efficiency of plant breeding methods and programs (Bassi et al., 2016).

J.M. González-Camacho, P. Pérez-Rodríguez, Statistics and Computer Science Graduate Program, Colegio de Postgraduados, Montecillo, Texcoco, México. CP 56230; L. Ornella, NIDERA SA 2600 Venado Tuerto, Argentina; D. Gianola, Dep. of Animal Sciences, Dairy Science, and Biostatistics and Medical Informatics, Univ. of Wisconsin, Madison, WI; S. Dreisigacker, J. Crossa, Biometrics and Statistics Unit, International Maize and Wheat Improvement Center (CIMMYT), Apdo. México City. Received 23 May 2017. Accepted 29 Jan. 2018. *Corresponding author (j.crossa@cgiar.org).

Abbreviations: APR, adult plant resistance; ANN, artificial neural networks; AUC, area under the receiver operating characteristic (ROC) curve; BL, Bayesian LASSO; BRNN, Bayesian regularized neural network; BST, boosting; CIMMYT, International Maize and Wheat Improvement Center; DTH, days to heading; GBS, genotyping-by-sequencing; GLS, gray leaf spot; GS, genomic selection; GY, grain yield; LR, leaf rust; MAS, marker-assisted selection; ML, machine learning; MLP, multi-layer perceptron; PLR, parametric linear regression; PNN, probabilistic neural network; PS, phenotypic selection; QTLs, quantitative trait loci; RBFNN, radial basis function neural network; *RE*, relative efficiency; RF, random forests; RFC, random forest classifier; RFR, random forest regression; RKHS, reproducing kernel Hilbert space; RR, ridge regression; SNP, single nucleotide polymorphisms; SR, stem rust; SVC-l, support vector linear classifier; SVC-g, support vector Gaussian classifier; SVM, support vector machine; SVR-l, support vector linear regression; SVR-g, support vector Gaussian regression; YR, yellow rust.

Plant Genome 11:170104
doi: 10.3835/plantgenome2017.11.0104

Crossa et al. (2010), for example, observed in four wheat datasets that the inclusion of markers significantly outperformed (by 8 to 36%) a pedigree-based model in predictive ability. Based on computer simulation studies, Bernardo and Yu (2007) showed that using the whole set of markers available for genotyping achieved better prediction of breeding values than using subsets of markers found to be significantly associated with quantitative trait loci (QTLs), i.e., marker-assisted selection (MAS). This was empirically confirmed by Heffner et al. (2011), who compared phenotypic selection (PS), MAS, and GS prediction performance for 13 phenotypic traits in 374 winter wheat (*Triticum aestivum* L.) breeding lines. Using GS, prediction accuracy measured by the correlation between observed and predictive values was 28% higher than with MAS, and 95% as high as with PS. Moreover, the mean correlation across six genomic selection indices was 14% higher than for PS. Arruda et al. (2016) found correlations between observed and predicted values that ranged from 0.4 to 0.9 in GS models for Fusarium head blight resistance in wheat, and lower values ($r < 0.3$) in MAS models. Rutkoski et al. (2014) obtained $r \cong 0.57$ using only GBLUP and $r > 0.6$ using GBLUP after adding a candidate gene-linked marker (*Sr2*) as a fixed effect (i.e., a combination of GS and MAS).

Rust diseases are one of main causes of wheat production losses throughout the world. *Puccinia graminis* (stem rust) and *Puccinia striiformis* (yellow rust) cause major economic losses and, hence, receive attention in wheat breeding programs (Ellis et al., 2014). Stem rust (SR) resistance is generally categorized into two groups: (i) all stage resistance, and (ii) slow rusting, or quantitative adult plant resistance (APR). In the first case, resistance is conferred by race-specific genes and is related to a hypersensitive response; in the second case, slow rusting resistance is usually conferred by multiple loci and not related to a hypersensitive response (Singh et al., 2011). Furthermore, slow rusting quantitative resistance is considered more durable than resistance conferred by pathogen recognition genes and must be improved over multiple selection cycles using screening nurseries for evaluation (Singh et al., 2011).

Strategies such as MAS and GS are alternatives for developing high-yielding wheat with APR. However, the complexity of these traits makes MAS difficult to implement. For instance, the *Sr25* gene provides SR resistance only when the *Sr2* gene is present (Singh et al., 2011). Also, the number of molecular markers associated with SR resistance genes is not enough for conducting MAS (Singh et al., 2011). Thus, GS is an important option for accumulating favorable alleles for rust resistance (Rutkoski et al., 2011; Ornella et al., 2012). Nevertheless, using GS for predicting disease resistance has its own peculiarities:

1. Linear GS approaches are usually limited to modeling additive effects; however, machine learning (ML) methods are able to include epistasis. Although there is general consensus about the additive nature of APR to rust (Ornella et al., 2012; Rutkoski et al., 2014), some populations show epistasis (Rouse et al., 2014).
2. Most GS applications assume that phenotypic response is continuous and normally distributed, whereas disease resistance is commonly expressed in ordinal scales (e.g., scales from 1 to 9, from 1 to 5, etc.) (Roelfs et al., 1992). Even if the data are transformed, many of the aforementioned problems remain in the model (Gianola 1980, 1982; Kizilkaya et al., 2014; Montesinos-López et al., 2015a). A distinctive attribute of many supervised learning algorithms is that there is no restriction regarding the distribution of response variables. This characteristic makes them less sensitive to the problems that arise in parametric models when ordinal scores are used to quantify diseases. The disadvantages of dealing with count data are discussed by Montesinos-López et al. (2015a, b), who also present an appealing parametric solution to this difficulty.
3. Clearly most economically important traits are affected by large numbers of genes with small effects (e.g., de los Campos et al., 2013), and durable rust resistance in wheat is determined by effective combinations of minor and major genes (Bansal et al., 2014). In this situation, simulation studies suggest the good performance of methods that use variable selection and differential shrinkage of allelic effects (e.g., Bayes B, de los Campos et al., 2013). Hence, ML should provide more flexible methods for genomic-enabled prediction values for SR resistance of wheat lines (González-Recio et al., 2014).

The main objective of this paper is to discuss several state-of-the-art ML methods applied in GS, particularly for predicting wheat diseases. First, we present a brief introduction to ML. Second, we compare results from well-known linear methods (ridge regression, Bayesian LASSO) with those from ML: random forests (RF), support vector machine (SVM) and radial basis function neural network (RBFNN). Third, we examine classification models instead of the regression models commonly used in GS. There are only a few reports of GS being used on rust (and even fewer on ML); however, there is considerable information related to mapping experiments that provide information about the relationship between QTLs and phenotypes. Finally, in the Appendix we present a brief summary of ML methods discussed in this review.

MATERIALS AND METHODS

Machine Learning Methods: An Overview

With advances in GS, volumes of data have dramatically increased, and new research efforts aimed at integrating and unifying several fields of research, such as computer science, ML, bioinformatics, mathematics, statistics, and genetics, have emerged as new data-driven science. This new field of research focuses on estimating more accurate predictive values of unobserved individuals by using statistical learning or ML methods. For example, artificial neural networks (ANN) are common prediction tools in ML. In GS, when a feedforward ANN with a single hidden layer is applied, each marker of the input vector is connected to all neurons in the hidden layer,

and these are connected to the output layer with the predicted phenotypic responses. More recently, developments in ANN and faster computer processing have allowed increasing the number of layers to ANN (deep learning) and improving prediction accuracy to capture higher-order interactions between covariates.

The ML is concerned with developing and applying computer algorithms that improve with data (Gareth et al., 2013). Learning can be classified as either supervised or unsupervised. In supervised learning, the objective is to predict a desired output value (trait) inferred from input data. The prediction task is called classification if outputs are categorical (e.g., red-black-blue, or susceptible-moderate-resistant), and regression if outputs are continuous. In unsupervised learning, the objective is to discover groups and associations among input variables where there is no output variable (Hastie et al., 2009).

Many types of methods are used in supervised learning, such as nearest-neighbors methods, decision trees, naive Bayes, Bayes nets, and rule-based learning (Kotsiantis 2007). Methods that have been applied in GS include SVM, RF, and ANN (Gianola et al., 2011; González-Camacho et al., 2012; Pérez-Rodríguez et al., 2012). The reproducing kernel Hilbert space (RKHS), initially presented as a semi-parametric method (Gianola et al., 2006), is now also included in the ML group (González-Recio et al., 2014).

Many ML methods have been implemented in statistical and data-mining open-source software, e.g., Weka (Hall et al., 2009) and R (R Core Team 2016), which run on most modern operating systems (Windows, macOS, and Linux). Because this kind of software is open-source, users can freely modify source codes to fit their own specific needs (Sonnenburg et al., 2007). ML methods have been developed under different theoretical frameworks using classic and Bayesian statistical approaches; they have helped explain field results and focus on the development and improvement of learning algorithms. One example is the theory of probably approximately correct learning described by Valiant (1984) that facilitates the development of boosting algorithms (BST) (Freund and Schapire 1996), with the statistical learning theory being the backbone of SVM (Cortes and Vapnik 1995). The SVM were evaluated for rust resistance (Ornella et al., 2012) and will be discussed in the next section and in the Appendix. To the best of our knowledge, BST has not been tested for GS in rust; however, some authors such as Ogutu et al. (2011) and González-Recio and Forni (2011) reported an outstanding performance of BST on simulated and real data. Boosting is a method for improving the accuracy of regression and classification models (Hastie et al., 2009). Combining multiple learning algorithms helps to improve predictive performance better than any of the constituent learning algorithms alone; the most popular ensemble-based algorithms are bagging, boosting and AdaBoost (Polikar 2006). Sun et al. (2012) proposed a successful ensemble-based approach to imputation of moderate-density genotypes for GS. The RF is another type of ensemble algorithm where the nonparametric function is the average of regression decision trees or classification (Hastie et al., 2009).

Alternatively, BST combines different predictors with some shrinkage imposed on each iteration, given a training dataset $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, where $\mathbf{x}_i$ is a set of input vectors and y_i is the response variable. The goal is to obtain an $\hat{F}_x$ approximation of the function F_x^* that minimizes the expected value of some specified loss function (e.g., mean squared error or exponential loss) over the joint distribution of all $(\mathbf{x}, y)$ values. A common way to obtain F_x^* is using an expansion of the form:

$$F_M(\mathbf{x}) = \sum_{m=1}^{M} \beta_m h_m(\mathbf{x}, y)$$

where $h_m(\mathbf{x}, y)$, the base learner, is a function relating $\mathbf{x}$ to y (commonly used base learners are decision trees or ordinary least-squares regression) and β_m is a weight assigned to the learner model. This expansion is commonly obtained using gradient descent techniques, i.e., $\hat{F}_x$ is sequentially updated, such that $F_M(\mathbf{x}) = F_{M-1}(\mathbf{x}) + \beta_M h_M(\mathbf{x}, y)$, where $\beta_M h_M(\mathbf{x}, y)$ is the learner that tends to the gradient of the loss function at a gradient-boosting point; a more detailed description is found in Friedman (2001). González-Recio et al. (2014) applied a BST algorithm for GS. Another BST development is the AdaBoost classifier: once a new base learner is added, the data are reweighed, and then correctly classified cases lose weight, whereas cases that are misclassified gain weight. Thus, next learners focus more on the instances that previous weak learners misclassified. This may cause AdaBoost not to be robust when dealing with noisy data and the presence of outliers (Freund and Schapire, 1996).

Gradient-boosted methods can deal with interactions among several variables, and select the variables automatically. These methods are also robust in the presence of outliers, missing data, and numerous correlated and irrelevant variables, and they take into account the importance of the variables in exactly the same way as RF (Walters et al., 2012).

We also review what are called single learners, i.e., single functions that cannot be decomposed into smaller units. The RKHS and SVM are examples of single learners with good learning performance. The increase of different modeling approaches in GS has made ML converge with classic statistical methods commonly applied in animal and plant breeding (González-Recio et al., 2014).

Parametric Linear Regression (PLR) Versus ML Methods in Genomic Breeding for Rust Resistance

Since GS was originally introduced by Meuwissen et al. (2001), a large number of parametric linear regression (PLR) models have been developed for prediction. The first models proposed by Meuwissen et al. (2001)— e.g., Bayes A and Bayes B— were followed by a plethora of Bayesian regression models— e.g., Bayes C, the Bayesian LASSO and Bayes R (de los Campos et al., 2013; Gianola 2013).

Based on many experiments performed at CIMMYT and elsewhere (Table 1), we observed that several complex traits (e.g., yield and days to heading) are better represented by nonlinear approaches that are able capture small effect cryptic epistatic effects (González-Camacho et al., 2012; Pérez-Rodríguez et al., 2012; Crossa et al., 2014). Pérez-Rodríguez et al. (2012) evaluated 306 CIMMYT wheat lines genotyped with 1717 DArT markers, and days to heading (DTH) and grain yield (GY) traits were measured in 12 environments. The linear models applied were Bayesian LASSO (BL), Bayesian Ridge Regression, Bayes A and Bayes B and the nonlinear models were RKHS, Bayesian regularized neural network (BRNN) and radial basis function neural network (RBFNN). Results show that three nonlinear models had better prediction accuracy than PLR models and that nonlinear models RKHS and RBFNN were consistently better than the linear models. Furthermore, González-Camacho et al. (2012) evaluated BL, RKHS, and RBFNN on 21 maize datasets and also found a slight but consistent superiority of nonlinear models.

The additive and oligogenic nature of rust seems to defeat the potential supremacy of non-parametric models (Ornella et al., 2014, Poland and Rutkoski, 2016). Ornella et al. (2012) compared the performance of ridge regression as implemented in the R package rrBLUP (Endelman, 2011), with BL, support vector linear regression (SVR-l), and support vector Gaussian regression (SVR-g) for predicting SR and yellow rust (YR) resistance in five wheat populations. For SR, they analyzed the performance of the models on 10 wheat datasets, each including 90 to 180 individuals (Ornella et al., 2012). The BL and ridge regression (RR) models had similar prediction performance, with a small superiority over SVR models. The BL produced the best results in seven of the 10 datasets. This result supports reports about the additive nature of SR resistance (Singh et al., 2008). Although APR to stem rust is additive in nature, the measurement is semi-quantitative. Although the scale of measurement ranges from 0 to 100% of disease severity, it is comparable to the one through nine scale used to measure the disease. Meanwhile, rust heritability is usually high and major genes play a crucial role in enhancing resistance in CIMMYT wheat germplasm. Further discussion on this topic can be found in the section on classification models.

Regarding YR resistance, for the nine datasets evaluated, predictions were not as good as those for SR resistance, probably because the heritability of YR resistance is lower than the heritability of SR resistance; for example, for population PBW343 × Kingbird, YR resistance had $H^2 = 0.45$, while for SR resistance $H^2 = 0.90$ (Ornella et al., 2012). The superiority of BL was less evident; SVR-l and SVR-g produced the best results (with statistically significant differences) and BL showed the best correlations in three populations.

Rutkoski et al. (2014) evaluated the performance of multiple linear regression models: GBLUP implemented in the rrBLUP package (Endelman, 2011), BL, and Bayes Cπ for predicting adult plant SR resistance in a set of 365 wheat lines characterized by GBS. They also included eight markers (identified in each training dataset by genome-wide association analysis) as fixed effects in the model. The best results were obtained with GBLUP considering markers linked to *Sr2* as fixed effects. In Rutkoski et al. (2014), broad-sense heritability was 0.82 and the most strongly significant *Sr2* linked marker explained 27% of the genetic variation. Including QTLs with large effects as fixed factors seems promising for PLR models in GS. Arruda et al. (2016) also obtained the best predictive correlations in GS for traits associated with Fusarium head blight resistance in wheat using rrBLUP + "in-house" QTLs (identified by genome-wide association studies, in the same population) treated as fixed effects.

Using simulation, Bernardo (2014) found that major genes should be fitted as having fixed effects in GS, especially if a few major genes are present and if each gene contributes more than 10% of the genetic variance. Higher heritability values (> 0.5) also strengthen the impact of considering major genes as fixed effects. The ML offers flexibility regarding combinations of genes with major and minor effects. Ornella et al. (2014) evaluated six regression models (BL, RR, RKHS, random forest regression [RFR], SVR-l, and SVR-g) for GS in 12 of the 19 rust datasets and the four yield datasets of wheat presented in Ornella et al. (2012). As in previous reports, RKHS had the best results in the four yield datasets, whereas RFR had the best results in nine of the 12 rust datasets (Table 2). This is an expected result, given that due to the additive nature of the trait (Singh et al., 2008; Rutkoski et al., 2014), one would expect PLR models (i.e., BL or RR) to produce the best results; major and minor gene/QTL with different effects are common in rust resistance. However, RFR has the capability of considering markers with large effects (see the Appendix), which is greater than the capability of linear models to capture additive effects. Within the ML group, RFR produced the best results in nine of the 12 rust datasets, RKHS produced the best results in one dataset and SVR-l was the best in one dataset, whereas within the PLR group, only the BL produced the best results in one dataset (Table 2). The performance of ML and PLR models on these rust data still supports the additive nature of rust statistical architecture. Note that RR and SVR-1 use the same regularization term in the cost function, but the algorithm for optimizing the parameters differs in each case because the objective function is different. In RR, a generalized least squares approach is used and in SVR-1, epsilon sensitive optimization is employed.

It should be mentioned that the precision of the classifiers depends on the number of individuals in a given class, which defines the extreme values. Small thresholds are associated with higher variabilities and worse results in the classifications. Thus by selecting a different α threshold, the conclusions may change. For example, Ornella et al. (2014) gave special emphasis to the analysis for α = 15%, because it is a percentile commonly used in

Table 1. Articles related to GS and Machine learning methods. Publications, datasets used, traits, models, and performance criteria.

Publications and datasets used	Traits†	Model‡	Performance criteria ¶
Ornella et al. (2012) Wheat datasets	PBW343xJuchi PBW343xPavon76 PBW343xMuu PBW343xKingbird PBW343xK-Nyangumi	BL, RR, SVR-l, SVR-g	*r*
González-Camacho et al. (2012); Ornella et al. (2014) 14 maize datasets 300 tropical lines genotyped with 55K SNPs 16 wheat datasets; 306 wheat lines genotyped with 1717 SNPs	GLS, FFL, MFL, ASI under SS or WW, GY-SS, GY-WW, Stem rust resistance, yellow rust resistance and GY	BL, RR, RKHS, RFR, SVR-l, SVR-g, RFC, SVC-l, SVC-g	*r*, κ, and *RE*
González-Camacho et al. (2012); maize datasets	GLS, FFL,MFL, ASI, SS or WW	BL, RHKS, RBFNN	*r* and PMSE
González-Camacho et al. (2016); 16 maize datasets; 17 wheat datasets	GLS, GY-SS, GY-WW, GY-LO, GY-HI, FFL, MFL, ASI with SS or WW, DTH	MLP, PNN	AUC and AUCpr

† Traits: female flowering time (FFL), male flowering time (MFL) and the MFL to FFL interval (ASI) under severe drought stress (SS) or in well-watered (WW) environments; grain yield (GY) under SS, WW, low (LO) and high (HI) yielding conditions; days to heading (DTH), gray leaf spot (GLS) resistance.

‡ Models: Bayesian Lasso (BL), Ridge Regression (RR), Reproducing Kernel Hilbert Spaces (RKHS), Random Forest Regression (RFR) and Support Vector Regression (SVR) with linear (l) and Gaussian kernels (g), Random Forest Classification (RFC) and Support Vector Classification (SVC) with linear (l) and Gaussian (g) kernels, radial basis function neural network (RBFNN), multi-layer perceptron (MLP) and Probabilistic neural network (PNN).

¶ Performance criteria: Pearson's correlation coefficient (r), Cohen's Kappa coefficient (κ), relative efficiency (RE), predictive mean squared error (PMSE), area under the receiver operating characteristic curve (AUC) and area under the recall-precision curve (AUCpr).

plant breeding programs; however, this threshold may lead to different results as compared to using smaller values of the threshold (e.g., 10 or 5%). In these cases, the problem would generally become more difficult but, in general, the regression approaches could improve, as the problem would start to resemble regression.

Based on our work and the bibliography, we found meaningful consistency in the performance of the methods and the main characteristics of the trait (architecture): a few major genes (rust) versus traits that are the result of the joint action of a large number of genes, each with small effects (e.g., yield), and epistasis versus additivity. In the following paragraphs, we will explore the relationship between the trait's architecture and the performance of some of the best-known GS methods: on the one hand, BL and RR as representatives of linear models; on the other hand, RKHS and RFR as examples of ML methods.

In Fig. 1A, we plot Pearson's correlations (r) of RKHS and RFR versus the r_{BL}/r_{RR} relationship in the 12 rust datasets already discussed. The 30 datasets (14 maize trials and 16 wheat trials) and the R and Java scripts used in this work are in the data repository http://repository.cimmyt.org/xmlui/handle/10883/2976 from Ornella et al. (2014). The ad hoc relationship r_{BL}/r_{RR} would measure the discrepancy in the estimated marker effects between the two methods. The BL induces marker-specific shrinkage of the estimated regression coefficient. The BL shrinks markers with near zero effects more than those with large effects; this leads to pseudo-variable selection when making predictions, whereas RR-BLUP assumes equal variance and shrinks all the marker effects to zero; this leads to potentially lower accuracy, especially when some large-effect QTLs are present close to or coincide with the markers (Thavamanikumar et al., 2015). Therefore, if there are no QTLs with large effects, the ratio holds

Table 2. Average Pearson's correlation (of 50 random partitions) of four regression models (RKHS, BL, RFR and SVR-l) applied to 16 wheat datasets. Bold numbers represent the models with the highest average (extracted from Ornella et al. 2014).

Dataset	RKHS†	BL	RFR	SVR-l
KBIRD-Srm	0.5	0.68	**0.75**	0.61
KBIRD-Sro	0.65	0.76	**0.8**	0.66
KNYANGUMI-Srm	0.35	0.38	**0.43**	0.39
KNYANGUMI-Sro	0.56	0.59	**0.68**	0.52
F6PAVON-Srm	0.5	0.6	**0.67**	0.46
F6PAVON-Sro	0.57	0.67	**0.71**	0.54
JUCHI-Ken	0.28	0.32	0.22	**0.42**
KBIRD-Ken	0.16	0.21	**0.45**	0.17
KBIRD-Tol	0.4	0.49	**0.53**	0.43
KNYANGUMI-Tol	0.14	0.28	**0.49**	0.21
F6PAVON-Ken	**0.33**	0.26	0.29	0.19
F6PAVON-Tol	0.51	**0.63**	0.56	0.54
GY-1	**0.57**	0.5	0.57	0.36
GY-2	**0.49**	0.49	0.45	0.36
GY-3	**0.41**	0.36	0.4	0.23
GY-4	**0.51**	0.44	0.49	0.34

† The models are RKHS (reproducing kernel Hilbert space); BL (Bayesian LASSO); RFR (random forest regression); SVR-l (support vector regression with linear kernel).

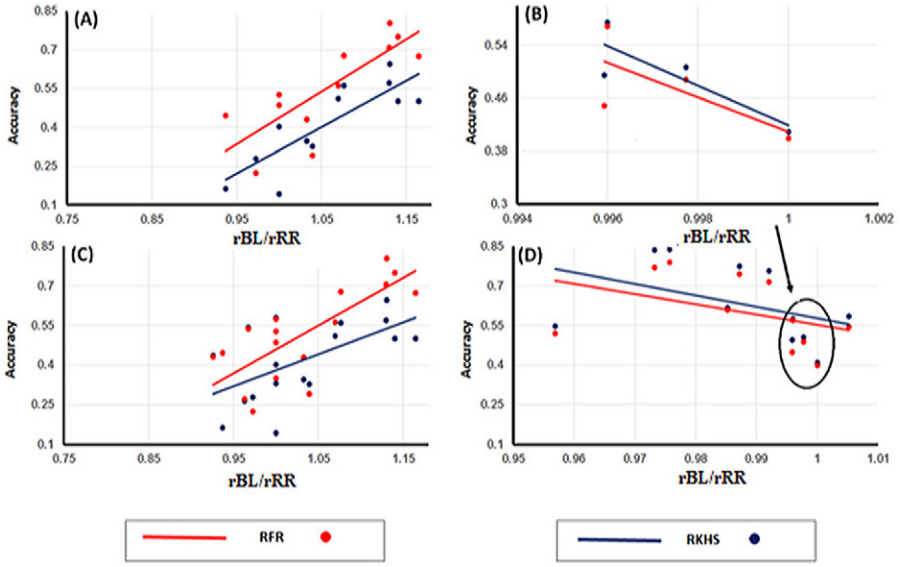

Fig. 1. Comparison of average Pearson's correlations (of 50 random partitions) of the RKHS and RFR models vs. the average ratio r_{BL}/r_{RR} for: (A) 12 rust datasets (B) 4 wheat yield datasets, (C) 12 rust datasets from (A) + 6 GLS datasets, (D) 4 datasets from (B) + 8 non-disease datasets from maize. RKHS (reproducing kernel Hilbert space), BL (Bayesian LASSO), RFR (random forest regression), RR, ridge regression, GLS (gray leaf spot) (adapted from Ornella et al., 2014).

close to one, whereas if there are large-effect QTLs, r_{BL}/r_{RR} would be greater than 1; the larger the effect of the QTLs, the greater the deviation from one should be (Thavamanikumar et al., 2015).

This concept was evaluated by analyzing the range of values on the *x* axis of Fig. 1. The four yield datasets in Fig. 1B present a very narrow r_{BL}/r_{RR} interval—from 0.99 to $\cong$1— whereas the range of the rust datasets (1A) varied from approximately 0.92 to more than 1.15. As a control, in Fig. 1D we added results of eight non-disease datasets (yield and flowering) from Ornella et al. (2014). Although the range on the *x* axis was broader, most ratios ranged between 0.97 and 1.01. We also included (Fig. 1C) the results of the five GLS (gray leaf spot) datasets of maize (also analyzed in Ornella et al., 2014), because this disease seems to be caused by a few major genes (Benson et al., 2015). Including these data points does not modify the range on the *x* axis $\cong$ [0.92, 1.2] nor the slope of the trend-lines. We included the performances of RKHS and RFR on the *y* axis with the intention of analyzing the response to the above-mentioned ratios of these two versatile methods that take epistasis into account. Interestingly, both methods showed a striking similarity in the slope of the trend-lines in the four datasets. Moreover, if large-effect QTLs are present (Fig. 1A and 1C), RFR outperforms RKHS. The situation is reversed if the trait is controlled by many small-effect QTLs (Fig. 1B and 1D). Although promising and consistent with the theory, our approach requires further research.

To conclude this section, we should mention that there are only a few other reports about GS for rust resistance. Daetwyler et al. (2014), for example, assessed the accuracy of GS for rust resistance in 206 hexaploid wheat (*Triticum aestivum*) landraces. Based on a five-fold cross-validation, the predicted mean correlations across years were 0.35, 0.27, and 0.44 for leaf rust (LR), stem rust (SR), and YR, respectively, using GBLUP, and 0.33, 0.38 and 0.30, respectively, using Bayes R (e.g., Gianola 2013). Genomic heritabilities estimated using restricted maximum likelihood were 0.49, 0.29, and 0.52 for LR, SR, and YR, respectively. To the best of our knowledge, no ML research has been done on these types of populations.

RESULTS AND DISCUSSION

Classification Models in Genomic Breeding for Rust Resistance

Although Pearson's correlation is usually applied for assessing the performance of model prediction in GS, it may not be an adequate performance criterion at the tails of the distribution function, where individuals are often selected. The correlation metric is sensitive to extreme values: thus by predicting well the worst and best rust tolerances individuals, a high correlation score will be obtained. This is also what the models optimize for in the training set, when Gaussian noise models are used. The classification methods, on the other hand, use all modeling resources to separate the classes. Thus from a pure ML perspective, when the scores are classification scores (kappa coefficient and relative efficiency), the classification methods should work better. Ornella et al. (2014) also evaluated the performance of the previously

described linear and nonlinear regression models for identifying individuals belonging to the best percentile of the distribution function (e.g., α = 15%), using the kappa coefficient (κ) and relative efficiency (*RE*).

The kappa coefficient is appropriate when data are unbalanced, for it estimates the proportion of cases that were correctly identified by taking into account coincidences expected from chance alone (Fielding and Bell 1997). κ is computed as:

$$\kappa = \frac{P_o - P_e}{1 - P_e}$$

where P_o is the agreement between observed and predicted values, computed by $\frac{tn+tp}{n}$, where *tn* is the number of true negatives; *tp* is the number of true positives; *n* is the total of individuals; P_e is the probability of agreement calculated as $P_e = \frac{\text{tp} + \text{fn}}{n} \times \frac{\text{tp} + \text{fp}}{n} + \frac{\text{fp} + \text{tn}}{n} \times \frac{\text{fn} + \text{tn}}{n}$, where *fp* is the number of false positives; and *fn* is the number of false negatives (Table 3). If there is complete agreement, then κ = 1 if there is no agreement among the raters other than what would be expected by chance, then κ ≤ 0.

The metric *RE* based on the expected genetic gain is computed by:

$$RE = \frac{\left(\sum_{\alpha'} y_i\right)/N_{\alpha'} - \left(\sum_{test} y_i\right)/N_{test}}{\left(\sum_{\alpha} y_i\right)/N_{\alpha} - \left(\sum_{test} y_i\right)/N_{test}}$$

where α′ and α represent the classes of extreme individuals selected from the observed ranking and predicted values; N_α and $N_{\alpha'}$ are the sizes of each class; y_i are the observed values; and $\left(\sum_{test} y_i\right)/N_{test}$ is the mean of the test dataset. The numerator in the *RE* equation represents the selection differential of individuals selected by GS, whereas the denominator represents the selection differential of individuals selected by traditional breeding (Falconer and Mackay, 1996).

Ornella et al. (2014) reported that *r* was a good performance measure of genomic-enabled prediction when replacing phenotypic selection for α = 15%. For the wheat and maize datasets evaluated in this paper, the relationships between *r* and κ and between *r* and *RE* were very similar (see Fig. 2 and 3 in Ornella et al., 2014). This close relationship is still present if we focus only on the rust data (i.e., with a peculiar distribution of the response variable). In Fig. 2A, we present a comparison of *r* vs. *RE* of the regression models (RKHS, RFR, RR, and BL) evaluated on the 12 rust datasets, whereas in Fig. 2B we use *r* vs. κ (α = 15%) as an example of the same approach and under the same conditions. From both figures, it seems that the ML and PLR models perform approximately the same when selecting the best individuals for a given level of correlation.

Classification models have been widely applied in research and industrial settings (Gareth et al., 2013).

Table 3. Confusion matrix for a binary classifier.

		Predicted value†		
		true	false	Sum
Observed value	**true**	*tp*	*fn*	*tp* + *fn*
	false	*fp*	*tn*	*fp* +*tn*
Sum		*tp* +*fp*	*fn* + *tn*	*n*

† *tp*: true positives, *fp*: false positives, *fn*: false negatives, *tn*: true negatives, and *n* is total of individuals.

Classifiers can capture nonlinear relationships between markers and phenotypes, especially in the neighborhood where breeders select the best lines, i.e., where we set the decision threshold. Ornella et al. (2014) also compared the performance of three classifiers for selecting the best individuals: support vector classifier (SVC-l) and (SVC-g) with linear and Gaussian kernels, respectively, and random forest classifier (RFC).

Tables 3 and 4 describe the performance measures (κ and *RE*) of the proposed regression and classification models for selecting the best 15% of individuals in the 16 wheat and 12 rust datasets. The SVC-l obtained the best κ in the SR datasets, in five YR datasets and in the GY-1 grain yield dataset (Table 4). This classifier outperformed F6PAVON-Srm and KNYANGUMI-Srm in the SR datasets, and F6PAVON-Ken in the YR and grain yield datasets. The SVC-g yielded the best κ in one YR dataset (KBIRD-Ken). Ridge regression had the best κ in GY-2, and RKHS gave the same value as BL in GY-3 and a higher value in GY-2 (Table 4).

Regarding *RE* (Table 5), SVC-l gave the highest *RE* in four SR datasets (both KBIRDs, KNYANGUMI-Srm and F6PAVON-Srm) and in all YR datasets except one (KBIRD-tol) (the best *RE* was RFR). The BL had the highest *RE* values in two SR datasets (KNYANGUMI-Sro and F6PAVON-Sro), whereas RKHS gave the best *RE* values in GY-2, GY-3, and GY-4. Finally, SVR-g had the best average *RE* in GY-1.

The binary classification model was better than the regression models on disease datasets probably due to the skewness of the distribution (Ornella et al., 2014; Montesinos-López et al., 2015a) or to how regression and classification models accounted for the complexity of the relationship between genotypes and phenotypes.

Concerning the impact of the distribution of the response variable (i.e., a discrete instead of a continuous variable) on the accuracy of prediction, Montesinos-López et al. (2015a, b) give several examples. In both papers, the authors introduced methods for dealing with discrete instead of continuous variables. Since these methods can be considered part of the PLR group, we refer the interested reader to the original references. For simplicity, we will finish this presentation by describing the ability of ANN to perform accurate multiclass classification.

González-Camacho et al. (2016) extended the work of Ornella et al. (2014) on assessing the performance of multi-layer perceptron (MLP) and probabilistic neural

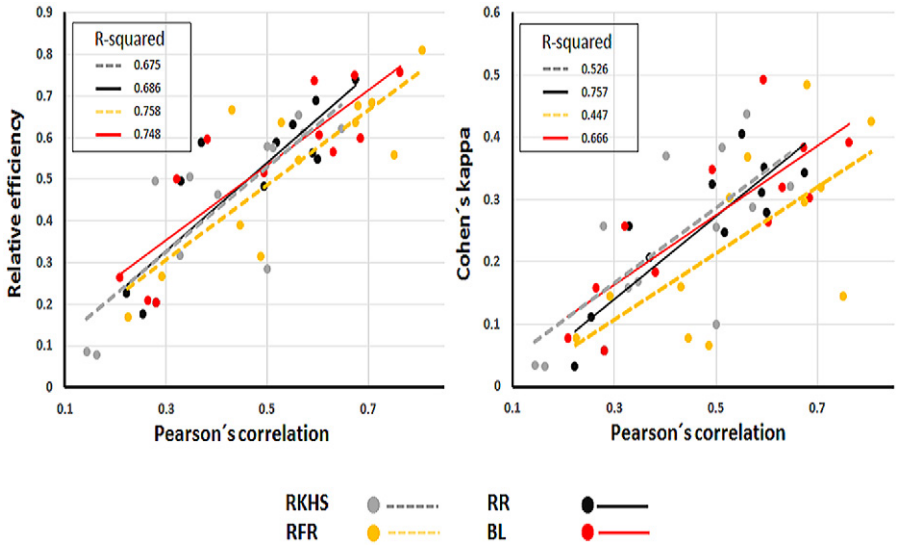

Fig. 2. Comparison of average relative efficiency and kappa coefficients (from 50 random partitions) for selecting the best 15% vs. average Pearson's correlations for 4 models evaluated on 12 rust datasets. BL (Bayesian LASSO), RFR (random forests regression), RKHS (reproducing kernel Hilbert space), RR (ridge regression) (adapted from Ornella et al., 2014).

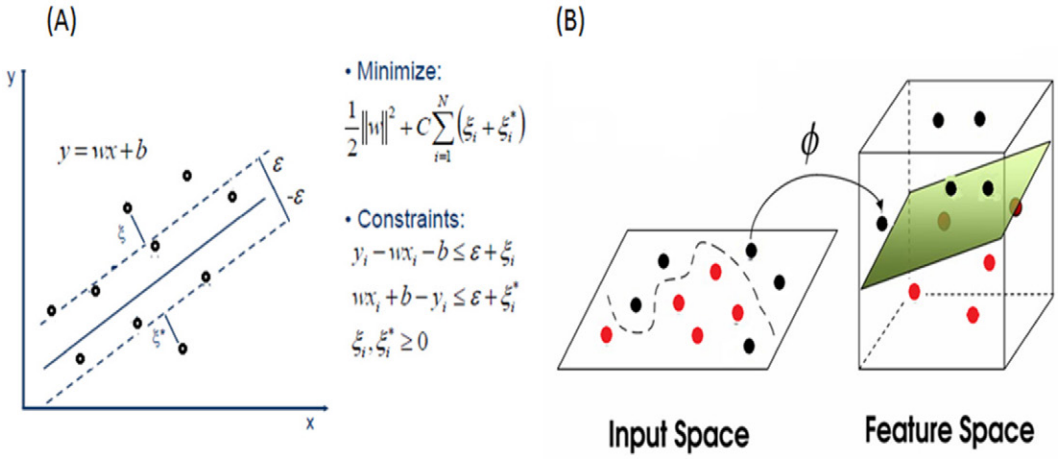

Fig. 3. Support vector machines. Examples of regression (A) and classification (B). (A) Support vector regression ignores residuals smaller than some constant ε (the ε tube) and assigns a linear loss function to larger errors. (B) Support vector classification: training data are mapped from the input space to the feature space, and a hyperplane is used to do the separation.

network (PNN) classifiers for estimating the probability of one individual belonging to a desired category. These classifiers were assessed using continuous traits grouped into three classes based on the empirical distribution function of traits. This is relevant for scales such as susceptible-moderate-resistant and can also be extended to more classes. González-Camacho et al. (2016) used genomic and phenotypic data from 16 maize datasets and 17 wheat datasets in various trait-environment conditions. The best individuals were selected from 15 and 30% of the upper class and similarly for the lower class. Wheat datasets were also evaluated as a binary class for the same percentiles in the upper and lower classes.

The performance of the two classifiers was assessed with two metrics: the area under the receiver operating characteristic (ROC) curve (AUC) and the area under the precision-recall curve (AUC*pr*) (Murphy 2012). The AUC produces a value between 0 and 1. The ROC curve is the plot of recall (*R*) versus the false positive rate (*fpr*) for a set of thresholds τ given.

Recall is the fraction of individuals correctly classified with respect to the observed individuals and is computed as:

$$R = \frac{tp}{tp + fn}$$

And *fpr* is defined as:

$$fpr = \frac{fp}{fp + tn}$$

The *AUCpr* measure is more informative with unbalanced classes (Keilwagen et al., 2014). The *AUCpr* is the

Table 4. Average kappa coefficients (of 50 random partitions) of three regression models (RKHS, BL, RR) and two classifiers (SVC-g, SVC-l) applied to 16 wheat datasets when the best 15% of the individuals are selected. Bold numbers indicate models with the highest average (extracted from Ornella et al., 2014).

Dataset	RKHS†	BL	RR	SVC-g	SVC-l
KBIRD-Srm	0.10	0.30	0.28	0.24	**0.44**
KBIRD-Sro	0.32	0.39	0.34	0.17	**0.53**
KNYANGUMI-Srm	0.17	0.18	0.21	0.23	**0.42**
KNYANGUMI-Sro	0.44	0.49	0.41	0.33	**0.49**
F6PAVON-Srm	0.26	0.26	0.25	0.36	**0.46**
F6PAVON-Sro	0.29	0.38	0.35	0.26	**0.46**
JUCHI-Ken	0.26	0.26	0.26	0.24	**0.28**
KBIRD-Ken	0.03	0.08	0.03	**0.23**	0.21
KBIRD-tol	0.37	0.35	0.33	0.30	**0.41**
KNYANGUMI-tol	0.03	0.06	0.06	0.29	**0.33**
F6PAVON-Ken	0.16	0.16	0.11	0.29	**0.35**
F6PAVON-tol	0.38	0.32	0.31	0.39	**0.42**
GY-1	0.23	0.14	0.14	0.27	**0.40**
GY-2	0.25	0.24	**0.26**	0.24	0.18
GY-3	0.23	**0.23**	0.23	0.22	0.15
GY-4	**0.42**	0.35	0.34	0.35	0.30

† The models are RKHS (reproducing kernel Hilbert space), BL (Bayesian LASSO), RR (ridge regression), SVC (support vector classification) with Gaussian (g) or linear (l) kernels.

Table 5. Average relative efficiency (of 50 random partitions) of three regression models and two classifiers (RKHS, BL, RFR, SVC-g, and SVC-l) applied to 16 wheat datasets when the best 15% of the individuals are selected. Bold numbers represent the highest values (extracted from Ornella et al., 2014).

Dataset	RKHS†	BL	RFR	SVC-g	SVC-l
KBIRD-Srm	0.28	0.60	0.56	0.12	**0.71**
KBIRD-Sro	0.62	0.76	0.81	0.47	**0.82**
KNYANGUMI-Srm	0.51	0.60	0.67	0.50	**0.81**
KNYANGUMI-Sro	0.65	**0.74**	0.68	0.48	0.73
F6PAVON-Srm	0.58	0.61	0.64	0.57	**0.78**
F6PAVON-Sro	0.61	**0.75**	0.69	0.49	0.74
JUCHI-Ken	0.50	0.50	0.17	0.26	**0.55**
KBIRD-Ken	0.08	0.27	0.39	0.19	**0.48**
KBIRD-tol	0.46	0.52	**0.64**	0.44	0.62
KNYANGUMI-tol	0.09	0.20	0.32	0.34	**0.55**
F6PAVON-Ken	0.32	0.21	0.27	0.26	**0.57**
F6PAVON-tol	0.58	0.57	0.55	0.48	**0.64**
GY-1	0.53	0.38	0.52	**0.62**	0.48
GY-2	**0.50**	0.46	0.39	0.45	0.32
GY-3	**0.46**	0.45	0.37	0.23	0.34
GY-4	**0.59**	0.48	0.56	0.47	0.36

† The models are RKHS (reproducing kernel Hilbert space), BL (Bayesian LASSO), RFR (random forest regression), SVC (support vector classification) with Gaussian (g) or linear (l) kernels.

area under the precision-recall curve, i.e., the plot of precision P vs. R for a set of thresholds τ. Precision P is computed as

$$P = \frac{tp}{tp + fp}$$

The PNN outperformed the MLP classifier for selecting on 15 and 30% upper classes, in 7 environments for GY, and for selecting on 15 and 30% lower classes in 10 environments for DTH (González-Camacho et al., 2016). In GS, it is usual that p markers >> n phenotype individuals, and PNN is promising because it has better generalization ability than MLP, with the advantage of being computationally faster than MLP in achieving optimal solutions. We must recall that, despite their good performance, both methods consider categorical response variables. Still to be explored are several methods in the ML repository that also take advantage of ordinal response variables (Hall and Frank, 2001).

CONCLUSIONS

The features of the architecture of rust resistance— i.e., their additive nature, the effective involvement of major and minor genes and the nature of the response variable (finite and discrete)— demand specific methods for dealing with the aforementioned characteristics. The flexibility of ML methods suggests ML is a valuable alternative to well-known parametric methods for predicting categorical and continuous responses in genomic selection.

We compared the performance of several regression/classification models against some parametric models (BL, ridge regression, etc.) on SR and YR. To show the broad horizon of ML capabilities, we also discussed the performance of ML methods on other traits, e.g., yield and days to heading in wheat or gray leaf spot resistance in maize. Results confirmed that ML methods could circumvent restrictions imposed by the statistical architecture of the trait. Further development is needed to adapt new approaches to wheat populations and/or environments.

Conflict of Interest Disclosure

The authors declare that there is no conflict of interest.

Appendix

A0. Reproducing Kernel Hilbert Space (RKHS) Models

The RKHS was initially proposed for animal breeding by Gianola et al. (2006); de los Campos et al. (2009) presented a genetic evaluation method where the values can be used for genetic evaluation using dense molecular

markers and pedigree. Furthermore, de los Campos et al. (2009) showed that the statistical models routinely used in genomic evaluations are particular cases of RKHS. De los Campos et al. (2010) give an overview of RKHS methods, as well as some ideas on how to set the parameters of the models and how to select the reproducing kernel; they also present an application in plant breeding. Other authors (e.g., Crossa et al., 2010; González-Camacho et al., 2012) have successfully applied RKHS models in plant breeding. The RKHS is the core mathematics used by all kernel methods; it is also the basis for the SVM and SVR methods, among others. Here the RKHS method additionally uses multiple kernel learning, which is an extension that combines several kernel functions.

A kernel is any smooth function K that defines a relationship between pairs of individuals through a covariance or by using similarity functions that can be defined in many ways. Several RKHS models have been proposed for GS (González-Recio et al., 2014). Here we describe the RKHS strategy with "kernel averaging" that was introduced in the field of quantitative genetics by de los Campos et al. (2010). The regression function takes the following form:

$$f(\mathbf{x}) = \beta_0 + \sum_{i'=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}_{i'})$$

where $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})'$ and $\mathbf{x}_{i'} = (x_{i'1}, \ldots, x_{i'p})'$ are vectors of dimension p markers; α_i are regression coefficients; and $K(.)$ is a positive definite function (the reproducing kernel, RK) evaluated in a pair of lines denoted by i and i'. In the case of a Gaussian kernel, $K(\mathbf{x}_i, \mathbf{x}_{i'}) = exp\{-h \| \mathbf{x}_i - \mathbf{x}_{i'} \|^2\}$, where $\| \mathbf{x}_i - \mathbf{x}_{i'} \|^2$ is the squared Euclidean distance between the vectors of markers for a pair of individuals (i, i'); and h is a parameter that is known as bandwidth, $h > 0$. The RK provides a set of n basis functions, which are nonlinear on marker genotypes; but the regression function is obtained by linearly combining the basis functions that are generated by the RK, with weights given by the corresponding regression coefficients.

To avoid over-fitting, the vector of regression coefficients is estimated using penalized or Bayesian methods. The set of basis functions is defined a priori via the choice of kernel; if the kernel is not selected properly, the ability of RKHS to capture complex patterns is affected. De los Campos et al. (2009) proposed kernel averaging to increase the accuracy of prediction. This multikernel approach was implemented using a Bayesian approach; for example, for three kernels:

$$\begin{cases} K_1(\mathbf{x}_i, \mathbf{x}_{i'}, h_1) = exp\left(-\frac{h_1}{q_{05}} \times d^2_{ii'}\right) \\ K_2(\mathbf{x}_i, \mathbf{x}_{i'}, h_2) = exp\left(-\frac{h_2}{q_{05}} \times d^2_{ii'}\right) \\ K_3(\mathbf{x}_i, \mathbf{x}_{i'}, h_2) = exp\left(-\frac{h_3}{q_{05}} \times d^2_{ii'}\right) \end{cases}$$

where $d^2_{ii'} = \sum_{j=1}^{p} \frac{(\mathbf{x}_{ij} - \mathbf{x}_{i'j})^2}{V_j}$ is a standardized squared Euclidean distance, V_j *is* the sample variance of the j-th marker, q_{05} is the fifth percentile of $d^2_{ii'}$, and $h_1 = 5$; $h_2 = 1$; $h_3 = 1/5$ are values of h, such that $K_1(\ldots)$ gives local basis functions and $K_3(\ldots)$ gives basis functions with a wider span.

A1. Support Vector Regression (SVR)

Support vector machine learning can be applied to classification or to regression problems (Cortes and Vapnik, 1995). Support vector regression (SVR) is a supervised learning algorithm capable of solving complex problems. The SVR aims to learn an unknown function based on the structural risk minimization principle. The SVR considers, in general, approximating functions of the form $f(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^{n} \mathbf{w}_i \varphi(\mathbf{x}_i)$. When we consider a linear regression model: $f(\mathbf{x}, \mathbf{w}) = \mathbf{x} \cdot \mathbf{w} + b$, where $\mathbf{x}, \mathbf{w} \in \mathrm{R}^p$ and $b \in \mathrm{R}$

In regression, usually an error of approximation is used instead of the margin between an optimal separating hyperplane and support vectors like in SVM. A linear loss function with ε-insensitivity zone is given by (Cortes and Vapnik 1995):

$$|y - f(\mathbf{x}, \mathbf{w})|_{\varepsilon} = \begin{cases} 0 & if\ |y - f(\mathbf{x}, \mathbf{w})| \le \varepsilon \\ |y - f(\mathbf{x}, \mathbf{w})| - \varepsilon & otherwise. \end{cases}$$

When the difference between the measured and predicted values is less than ε, then the error is equal to zero. The ε-insensitive loss function defines an ε tube as depicted in Fig. 3. When the predicted response is within the tube, the error is zero. For points outside the tube, the error is the difference between the predicted response and the radius ε of the tube.

In an SVR problem, the objective is to minimize the empirical risk and the squared norm of the weights simultaneously. That is, to estimate the hyperplane $f(\mathbf{x},\mathbf{w}) = \mathbf{x}^T\mathbf{w} + b$ by minimizing the empirical risk:

$$R = \frac{1}{2}\mathbf{w}^2 + C\left(\sum_{i=1}^{n} |y_i - f(\mathbf{x}_i, \mathbf{w})|_{\varepsilon}\right)$$

Minimizing R is equivalent to minimizing the risk as a function of slack variables (Cortes and Vapnik, 1995):

$$R = \frac{1}{2}\mathbf{w}^2 + C\left(\sum_{i=1}^{n} |y_i - f(\mathbf{x}_i, \mathbf{w})|_{\varepsilon}\right)$$

subject to constraints,

$$y_i - \mathbf{w}^T\mathbf{x}_i - b \le \varepsilon + \xi,\ i = 1, \ldots, n$$

$$\mathbf{w}^T\mathbf{x}_i + b - y_i \le \varepsilon + \xi^*,\ i = 1, \ldots, n$$

$$\xi \ge 0; \xi^* \ge 0,\ i = 1, \ldots, n$$

where ξ and ξ^* are slack variables. $\xi = y - f(\mathbf{x},\mathbf{w}) - \varepsilon$ for points above an ε tube; $\xi^* = f(\mathbf{x},\mathbf{w}) - y - \varepsilon$ for points below an ε tube; the constant C is a parameter defined by the user. Big C values restrict large errors, reduce the approximation errors and increase the weight vector norm $\| \mathbf{w} \|$. The latter increase does not favor good generalization performance of a model; ε is another design parameter that defines the size of an ε tube. Figure 3(A) describes a regression problem for which no linear solution exists, given the radius of an ε tube.

To perform a nonlinear regression, nonlinear functions f are defined by mapping the training dataset into a higher dimensional space, the feature space F (φ: X →F), and then solve the linear regression problem there (Fig. 3B). As in a classification problem, to make this approach computationally feasible, a symmetric kernel function can be used $K(\mathbf{x};y) = [\phi(\mathbf{x})] \cdot \phi(y)]$ that directly creates a dot product in the feature space (Hastie et al., 2009).

The linear kernel defined as $K(\mathbf{x};y) = (\mathbf{x} \times y)$ is the simplest kernel method. Another kernel method is the previously presented Gaussian kernel. The best values of C, ε and, eventually, h (Gaussian kernel) are commonly estimated by means of a grid search combined with a cross-validation procedure (Maenhout et al., 2007).

A2. Feedforward Artificial Neural Networks (ANN)

The ANN has the ability to capture linear and nonlinear relationships between predictor variables and response variables, including interactions between explanatory variables (Gianola et al., 2011; González-Camacho et al., 2012). The ANN used in GS are based on a single hidden layer for predicting phenotypic responses from genotypic inputs. The first layer contains the marker input vector; the hidden layer contains a varying number of neurons with nonlinear transfer functions. Each neuron in the hidden layer transforms the input and sends the result to the output layer. Thus the predicted responses of the neural network represent the weighted outputs from each neuron.

The ANN have internal parameters with two main model choices: (i) the number of neurons in the hidden layer, and (ii) the type of transfer function applied in each neuron. A low number of neurons in a neural network may not be able to capture complex patterns. On the other hand, a neural network with many neurons can have overfitting and poor predictive ability (Haykin, 1994).

The multilayer perceptron (MLP) and the radial basis function neural network (RBFNN) are the most frequently used feedforward neural network models. The MLP uses the back-propagation algorithm to estimate the weight vectors. In this case, the input data are iteratively given to the neural network. For each input data, the error between the predicted and desired response is computed. The error is then back-propagated to the ANN and employed to modify the weights; thus the error diminishes at each iteration. The RBFNN is defined by a hidden layer of processing units with Gaussian radial basis functions. Its structure for obtaining a phenotypic response from a marker input vector is described in Fig. 4 (González-Camacho et al., 2012).

A3. Multilayer Perceptron (MLP) Classifier

An MLP classifier maps a set of input data into C different disjoint classes (Fig. 5). In general, MLP is flexible because no assumptions are made about the joint distribution of inputs and outputs. The hidden layer contains M neurons. In each one, an S score is obtained using a linear combination of the input markers plus an intercept term, i.e.,

$$S = w_{m0} + \sum_{j=1}^{j=p} w_{mj} x_{ij} .$$

Then S is transformed using a nonlinear transfer function (e.g., a hyperbolic tangent sigmoid function), which maps from the real line to the interval [–1,1]. The sum layer contains C neurons (i.e., the number of classes) and each neuron processes the outputs of the hidden layer also using a linear combination of the outputs plus the intercept and a transformation using the tangent sigmoid function. Finally, the output of the MLP is a column vector of C elements. The index of the largest element in the vector specifies which of the C classes the vector represents (González Camacho et al., 2016).

The mean squared error between the predicted class $\hat{C}$ and the desired class C is commonly used to optimize an MLP classifier. $\hat{C}$ is a matrix of size S × n, with columns containing values in the [0,1] interval. Training of an MLP involves estimating all the parameters using the backpropagation method, based on the conjugate gradient method (Møller, 1993).

A4. Probabilistic Neural Network (PNN) Classifier

A PNN has a single hidden layer (Fig. 6), but conceptually it is different from MLP. The pattern layer has M neurons (M = number of individuals in the training dataset) and calculates the Euclidean norm between the input vector $\mathbf{x}_i$ and the center vectors $\mathbf{c}_m$ using a Gaussian kernel. The output of this pattern layer is a vector $\mathbf{u}_i$ with M elements, where

$$u_{mi} = \frac{\sqrt{-\ln 0.5}}{h} \| \mathbf{x}_i - \mathbf{c}_m \|$$

and h is the spread of the Gaussian function, showing how near $\mathbf{x}_i$ is to $\mathbf{c}_m$ (the m*th* training data). At that point, in the summation-output layer, each $\boldsymbol{u}_{mi}$ is transformed into a vector $\mathbf{z}_i \cdot \in \mathbb{R}^M$ whose elements are defined by $z_{mi} = \exp(-u_{mi})$. The $\mathbf{z}_{mi}$ values are then used to obtain the contribution for each k class, that is, $\nu_{ki} = \sum_{m=1}^{M} w_{km} z_{mi}$, where w_{km} are weights computed from the desired class $\mathbf{C}$ matrix of dimension $S \times n$ to produce a vector of probabilities $\hat{c}_i$ = softmax ($\mathbf{v}_i$) of size $S \times 1$ as its response; the softmax function σ (.) is given by

$$\sigma(\mathbf{v}_i) = \frac{\exp(v_k)}{\sum_{j=1}^{S} \exp(v_k)} \text{ for } k = 1,\ldots, \text{S classes}$$

where $\mathbf{v}_i$ is a desired vector of size $S \times 1$. For each k class, the softmax function transforms the outputs of

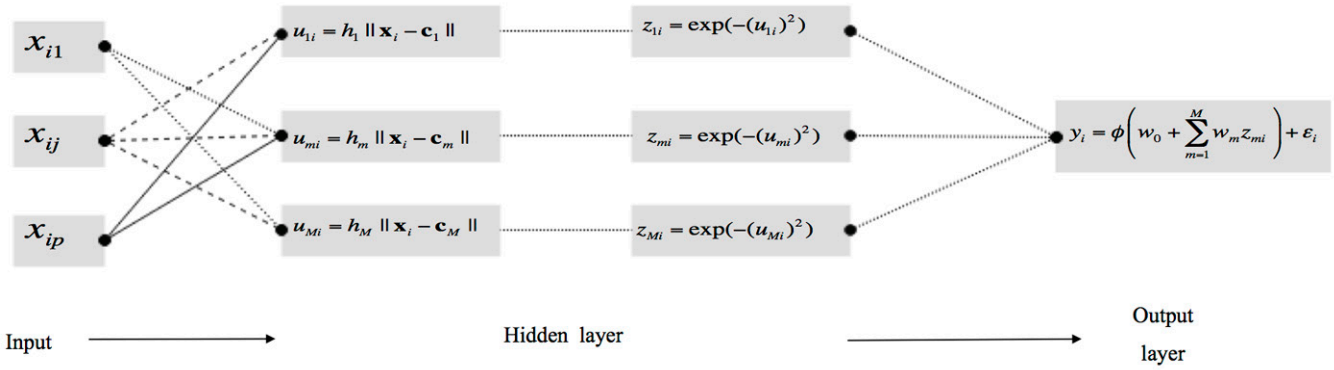

Fig. 4. Structure of a radial basis function neural network (RBFNN). In the hidden layer, each input vector $(\mathbf{x}_{i1},\ldots,\mathbf{x}_{ip})$ is summarized by the Euclidean distance between the input vectors x_i and the centers c_m, m = 1, …, M neurons, i.e., $h_m||x_i - c_m||$, where h_m is a bandwidth parameter. Then distances are transformed by the Gaussian kernel $\exp\left(-\left(\mathbf{h}_m \|\mathbf{x}_i - \mathbf{c}_m\|\right)^2\right)$ for obtaining the responses $y_i = \mathbf{w}_0 + \sum_{m=1}^{M} \mathbf{w}_m \mathbf{z}_{mi} + \varepsilon_i$, (extracted from González-Camacho et al. 2012).

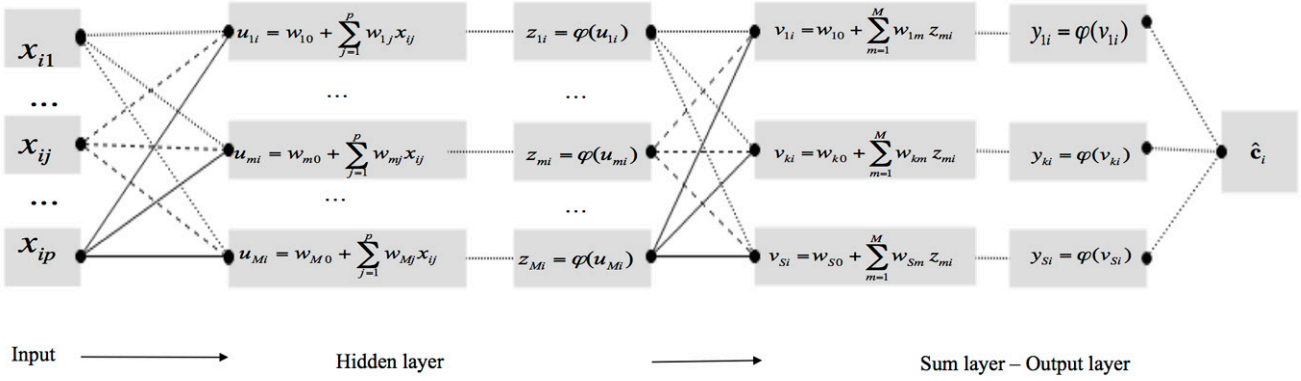

Fig. 5. Structure of a multi-layer perceptron (MLP) classifier. The hidden layer has M neurons whose outputs are computed by a linear combination of the input and weight vectors plus a bias term. In the output layer, the output of each of the S neurons (classes) is computed by a nonlinear transfer function and the phenotypic response is regressed from the data-derived features (extracted from González-Camacho et al. 2016).

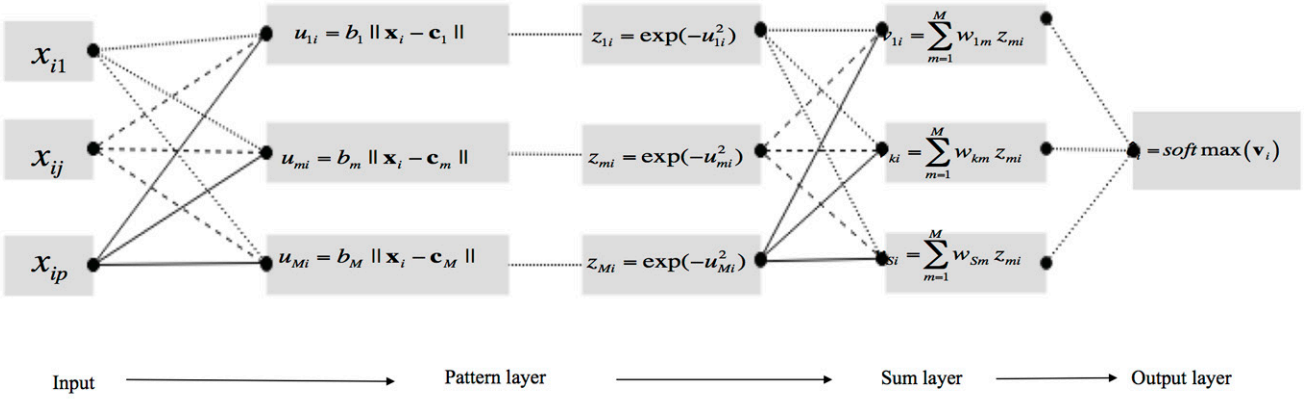

Fig. 6. Structure of a probabilistic neural network (PNN) classifier with a single hidden or pattern layer. In this layer, Euclidean distances are computed using a Gaussian kernel. Then the summation-output layer integrates the contributions for each class to generate a vector of probabilities (extracted from González-Camacho et al. 2016).

processing units on the output layer in the [0, 1] range (González Camacho et al., 2016).

An advantage of the PNN is that, for a given a value of *h,* it requires a single iteration over all the $\mathbf{x}_i$ in the training set. Furthermore, PNN produces as output (c_i) the posterior probabilities of each class membership within a short training time. The PNN converges to a Bayes classifier if the training dataset is large enough (Wasserman, 1993).

A5. Deep Learning

Conventional ANN models contain at least one hidden layer. Reaching beyond, deep learning architectures use many layers of nonlinear processing units (Goodfellow et al., 2016). The presence of more than one hidden layer allows the network to learn higher order interactions. The computing of the weight vector in each layer combines simple and complex features so that the most suitable hierarchical representations can be captured from the data.

Min et al. (2016) characterized traditional deep learning architectures into six types: deep neural network (DNN), multilayer perceptron (MLP), stacked auto-encoder, deep belief network, convolutional neural network (CNN) and recurrent neural network (RNN). They also defined three emergent architectures: convolutional auto-encoder, multi-dimensional recurrent neural networks, and deep spatiotemporal neural networks.

Each architecture has advantages. For instance, CNN is suitable for studying spatial information, DNN is suitable for analyzing internal correlations in high-dimensional data and RNN is suitable for analyzing sequential information.

A key element to take into account when training deep learning architectures is regularization, whose objective is to avoid over-fitting and attain good generalization performance. The two most common methods are weight decay, where a penalty term is imposed on the error loss function so that the weight vectors converge to smaller absolute values, and dropout, which randomly removes hidden units from the network.

A6. Random Forests (RF)

The RF is a non-parametric approach for solving regression and classification problems proposed by Breiman (2001) based on bagging, i.e., "bootstrap aggregated sampling." Gianola et al. (2014) applied the concept of bagging to GBLUP. The RF captures complex interactions and it is robust to over-fitting the data. However, variable importance measures are reported to be systematically affected by minor allele frequency (Walters et al., 2012). The RF combines the output of decorrelated decision trees from classification or regression generated from bootstrapped samples of the training dataset (Hastie et al., 2009).

For both classification and regression models, each tree is constructed using the following heuristic algorithm:

i. Draw samples with replacements from the entire dataset by bootstrapping so that records of the *i*th individual appear several times, or not at all, in the bootstrapped set. Root-node is the name given to each subset of the bootstrapped sample.
ii. Draw a certain number (*mtry*) of input variables (SNPs) at random, and select the *j*th SNP $j = 1,\ldots,mtry$ that minimizes a loss function. The entropy criterion is used for classification problems, whereas the mean squared error is used for regression problems.
iii. According to the genotypes of the *j*th SNP, the data in the node are separated into two new subsets.
iv. Repeat steps ii–iii for each new node with the data until a minimum node size (number of individuals) is reached. This number is usually five or less.
v. Construct a number *ntree* (> 50 – 100) of new trees by repeating steps I through iv and using new bootstrapped samples.

Finally, to make a prediction using a new example, the RF combines the outputs of the classification or regression trees based on bootstrapped samples of the dataset. In classification, the class of an unobserved example is predicted by counting the number of votes (usually one vote per decision tree is used) and assigning the class with the highest number of votes (Liaw, 2013). In regression, the method averages the *ntree* outputs.

Two main features can be tuned in RF:

1. Number of covariates (markers) sampled at random for each node. Although cross-validation strategies can be used to optimize *mtry*, the default values of $mtry = p / 3$ (p is the number of predictors) when building a regression tree, or $mtry = \sqrt{p}$ when building a classification tree, are commonly used (Gareth et al., 2013).
2. Number of trees; there is consensus that RF does not overfit with a growing number of trees; however, building each tree is very time-consuming. In our experience, an *ntree* range between 500 and 1000 is safe enough for plant datasets evaluated in GS.

Another key benefit of RF, as implemented in the R package "randomForests" (Liaw 2013), is that it provides two different measures for judging the relevance of predictor variables (e.g., marker or environmental effects). The first measure is computed by permuting markers from the OOB ("out of bag" sample) data, i.e., those individuals (around a third of the total) that were not selected in the bootstrapped sample. Briefly, after a tree is constructed, the OOB is passed down the tree and the prediction accuracy on this sample is calculated using several criteria (error rate for classification, mean squared error for regression problems). The genotypes for the *p*-th SNP are permuted in the OOB and the sample is again passed down the tree. The relative importance is computed as the difference between the prediction accuracy of the original OOB and the prediction accuracy of the OOB with the permuted variable. Then this step is repeated for each covariate (SNP) and the decrease in accuracy is averaged over all trees in the random forests (normalized by the standard deviation of

the differences). The SNPs that show a larger decrease are assumed to be more relevant (González-Recio et al., 2014).

The second measure is the reduction in impurities of the node from splitting the variable, averaged over all trees. For classification problems, node impurity is measured by the Gini index, whereas for regression, it is measured by the residual sum of squares. Impurity is calculated only at the node at which that variable is used for that split (Liaw, 2013).

References

Arruda, M.P., A.E. Lipka, P.J. Brown, A.M. Krill, C. Thurber, G. Brown-Guedira, Y. Dong, B.J. Foresman, and F.L. Kolb. 2016. Comparing genomic selection and marker-assisted selection for fusarium head blight resistance in wheat (*Triticum aestivum* L.). Mol. Breed. 36:84. doi:10.1007/s11032-016-0508-5

Bansal, U., H. Bariana, D. Wong, M. Randhawa, T. Wicker, M. Hayden, and B. Keller. 2014. Molecular mapping of an adult plant stem rust resistance gene *Sr56* in winter wheat cultivar Arina. Theor. Appl. Genet. 127:1441–1448. doi:10.1007/s00122-014-2311-1

Bassi, F.M., A.R. Bentley, G. Charmet, R. Ortiz, and J. Crossa. 2016. Breeding schemes for the implementation of genomic selection in wheat (*Triticum* spp.). Plant Sci. 242:23–36. doi:10.1016/j.plantsci.2015.08.021

Benson, J.M., J.A. Poland, B.M. Benson, E.L. Stromberg, and R.J. Nelson. 2015. Resistance to gray leaf spot of maize: Genetic architecture and mechanisms elucidated through nested association mapping and near-isogenic line analysis. PLoS Genet. 045. doi:10.1371/journal.pgen.1005

Bernardo, R. 2014. Genome-wide selection when major genes are known. Crop Sci. 54:68–75. doi:10.2135/cropsci2013.05.0315

Bernardo, R., and J. Yu. 2007. Prospects for genome-wide selection for quantitative traits in maize. Crop Sci. 47:1082–1090. doi:10.2135/cropsci2006.11.0690

Breiman, L. 2001. Random forests. Mach. Learn. 45:5–32. doi:10.1023/A:1010933404324

Cortes, C., and V. Vapnik. 1995. Support-vector networks. Mach. Learn. 20:273–297. doi:10.1007/BF00994018

Crossa, J., G. de los Campos, P. Pérez-Rodríguez, D. Gianola, J. Burgueño, J.L. Araus, et al. 2010. Prediction of genetic values of quantitative traits in plant breeding using pedigree and molecular markers. Genetics 186:713–724. doi:10.1534/genetics.110.118521

Crossa, J., P. Pérez-Rodríguez, J. Hickey, J. Burgueño, L. Ornella, et al. 2014. Genomic prediction in CIMMYT maize and wheat breeding programs. Heredity 112:48–60. doi:10.1038/hdy.2013.16

Daetwyler, H., U.K. Bansal, S. Harbans, H.S. Bariana, M.J. Hayden, and B.J. Hayes. 2014. Genomic prediction for rust resistance in diverse wheat landraces. Theor. Appl. Genet. 127:1795–1803. doi:10.1007/s00122-014-2341-8

de los Campos, G., D. Gianola, and G.J.M. Rosa. 2009. Reproducing kernel Hilbert spaces regression: A general framework for genetic evaluation. J. Anim. Sci. 87:1883–1887. doi:10.2527/jas.2008-1259

de los Campos, G., D. Gianola, G.J.M. Rosa, K.A. Weigel, and J. Crossa. 2010. Semi-parametric genomic-enabled prediction of genetic values using reproducing kernel Hilbert spaces methods. Genet. Res. 92:295–308. doi:10.1017/S0016672310000285

de los Campos, G., J.M. Hickey, R. Pong-Wong, H.D. Daetwyler, and M.P.L. Calus. 2013. Whole genome regression and prediction methods applied to plant and animal breeding. Genetics 193:327–345. doi:10.1534/genetics.112.143313

Ellis, J.G., E.S. Lagudah, W. Spielmeyer, and P.N. Dodds. 2014. The past, present and future of breeding rust resistant wheat. Frontier Plant Sc. 5:641.

Elshire, R.J., J.C. Glaubitz, Q. Sun, J.A. Poland, K. Kawamoto, E.S. Buckler, and S.E. Mitchell. 2011. A robust, simple genotyping-by-sequencing (GBS) approach for high diversity species. PLoS One 6(5):e19379. doi:10.1371/journal.pone.0019379

Endelman, J.B. 2011. Ridge regression and other kernels for genomic selection with R package rrBLUP. Plant Genome 4:250–255. doi:10.3835/plantgenome2011.08.0024

Falconer, D.S., and T.F.C. Mackay. 1996. Introduction to quantitative genetics. 4th Edition, Longman Group Ltd., London.

Fielding, A.H., and J.F. Bell. 1997. A review of methods for the assessment of prediction errors in conservation presence/absence models. Environ. Conserv. 24:38–49. doi:10.1017/S0376892997000088

Freund, Y., and R.E. Schapire. 1996. Experiments with a new boosting algorithm. In: Thirteenth International Conference on Machine Learning (ed. L. Saitta), p. 148–156. San Francisco, CA: Morgan Kaufmann. ISBN 1-55860-419-7.

Friedman, J.H. 2001. Greedy functions approximation: A gradient boosting machine. Ann. Stat. 29:1189–1232. doi:10.1214/aos/1013203451

Gareth J., D. Witten, T. Hastie, and R. Tibshirani. 2013. An introduction to statistical learning. Springer Science+Business Media New York (corrected at 6th printing 2015).

Gianola, D. 1980. A Method of Sire Evaluation for Dichotomies. J. Anim. Sci. 51:1266–1271. doi:10.2527/jas1981.5161266x

Gianola, D. 1982. Theory and Analysis of Threshold Characters. J. Anim. Sci. 54:1079–1096. doi:10.2527/jas1982.5451079x

Gianola, D. 2013. Priors in whole-genome regression: The Bayesian alphabet returns. Genetics 194:573–596. doi:10.1534/genetics.113.151753

Gianola, D., R. Fernando, and A. Stella. 2006. Genomic-assisted prediction of genetic values with semiparametric procedures. Genetics 173:1761–1776. doi:10.1534/genetics.105.049510

Gianola, D., H. Okut, K.A. Weigel, and G.J.M. Rosa. 2011. Predicting complex quantitative traits with Bayesian neural networks: A case study with Jersey cows and wheat. BMC Genet. 12:87. doi:10.1186/1471-2156-12-87

Gianola, D., K.A. Weigel, N. Krämer, A. Stella, and C.C. Schön. 2014. Enhancing genome-enabled prediction by bagging genomic BLUP. PLoS One. doi:10.1371/journal.pone.0091693

González-Camacho, J.M., G. de los Campos, P. Pérez-Rodríguez, D. Gianola, J.E. Cairns, G. Mahuku, R. Babu, and J. Crossa. 2012. Genome-enabled prediction of genetic values using radial basis function neural networks. Theor. Appl. Genet. 125:759–771. doi:10.1007/s00122-012-1868-9

González-Camacho, J.M., J. Crossa, P. Pérez-Rodríguez, L. Ornella, and D. Gianola. 2016. Genome-enabled prediction using probabilistic neural network classifiers. BMC Genomics 17:208. doi:10.1186/s12864-016-2553-1

González-Recio, O., and S. Forni. 2011. Genome-wide prediction of discrete traits using Bayesian regressions and machine learning. Genet. Sel. Evol. 43:7. doi:10.1186/1297-9686-43-7

González-Recio, O., G.J.M. Rosa, and D. Gianola. 2014. Machine learning methods and predictive ability metrics for genome-wide prediction of complex traits. Livest. Sci. 166:217–231. doi:10.1016/j.livsci.2014.05.036

Goodfellow, I., Y. Bengio, and A. Courville. 2016. Deep Learning. MIT Press, Cambridge, MA.

Hall, M., and E. Frank. 2001. A Simple Approach to Ordinal Classification. Machine Learning: ECML 2001. Lect. Notes Comput. Sci. 2167:145–156.

Hall, M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. 2009. The WEKA data mining software: An update. SIGKDD Explor. 11(1):10–18. doi:10.1145/1656274.1656278

Hastie, T., R. Tibshirani, and J. Friedman. 2009. The Elements of Statistical Learning. Data Mining, Inference, and Prediction. Second Edition: Corrected 10th printing. Springer-Verlag. doi:10.1007/978-0-387-84858-7

Haykin, S. 1994. Neural networks: A comprehensive foundation. MacMillan, New York, NY.

Heffner, E.L., J.L. Jannink, and M.E. Sorrells. 2011. Genomic selection accuracy using multifamily prediction models in a wheat breeding program. Plant Genome 4:65–75. doi:10.3835/plantgenome2010.12.0029

Keilwagen, J., I. Grosse, and J. Grau. 2014. Area under precision-recall curves for weighted and unweighted data. PLoS One 2014. doi:10.1371/journal.pone.0092209

Kizilkaya, K., R.L. Fernando, and D.J. Garrick. 2014. Reduction in accuracy of genomic prediction for ordered categorical data compared to continuous observations. Genet. Sel. Evol. 46:37. doi:10.1186/1297-9686-46-37

Kotsiantis, S.B. 2007. Supervised machine learning: A review of classification techniques. Informatica 31:249–268.

Kumar S., W. Travis, T.W. Banks, and S. Cloutier. 2012. SNP discovery through next-generation sequencing and its applications. Intern. J. of Pl. Genom. doi:10.1155/2012/831460.

Liaw, A. 2013. Package 'randomForest'. Breiman and Cutler's random forests for classification and regression (R package manual). http://cran.r-project.org/web/packages/randomForest/index.html. (accessed Oct. 2015).

Maenhout, S., B. De Baets, G. Haesaert, and E. Van Bockstaele. 2007. Support vector machine regression for the prediction of maize hybrid performance. Theor. Appl. Genet. 115:1003–1013. doi:10.1007/s00122-007-0627-9

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

Min, S., B. Lee, and S. Yoon. 2016. Deep learning in bioinformatics. Briefings in bioinformatics. 18(5):851–869. doi:10.1093/bib/bbw068

Møller, M.F. 1993. Scaled conjugate gradient algorithm for fast supervised learning. Neural Netw. 6:525–533. doi:10.1016/S0893-6080(05)80056-5

Montesinos-López O.A., A. Montesinos-López, P. Pérez-Rodríguez, G. de los Campos, K.M. Eskridge, and J. Crossa. 2015a. Threshold models for genome-enabled prediction of ordinal categorical traits in plant breeding. G3: Genes Genomes Genetics. 5:291–300.

Montesinos-López, O.A., A. Montesinos-López, P. Pérez-Rodríguez, K.M. Eskridge, X. He, P. Juliana, P. Singh, and J. Crossa. 2015b. Genomic prediction models for count data. J. Agric. Biol. Environ. Stat. 20:533–554. doi:10.1007/s13253-015-0223-4

Murphy, K.P. 2012. Machine learning: A probabilistic perspective. 1st ed. Cambridge, Massachusetts, London, England: The MIT Press.

Ogutu, J.O., H.P. Piepho, and T. Schulz-Streeck. 2011. A comparison of random forests, boosting and support vector machines for genomic selection. BMC Proc. 5(Suppl 3):S11. doi:10.1186/1753-6561-5-S3-S11

Ornella, L., S. Singh, P. Pérez-Rodríguez, J. Burgueño, R. Singh, E. Tapia, S. Bhavani, S. Dreisigacker, H.J. Braun, K. Mathews, and J. Crossa. 2012. Genomic prediction of genetic values for resistance to wheat rusts. Plant Genome 5:136–148. doi:10.3835/plantgenome2012.07.0017

Ornella, L., P. Pérez-Rodríguez, E. Tapia, J.M. González-Camacho, J. Burgueño, et al. 2014. Genomic-enabled prediction with classification algorithms. Heredity 112:616–626. doi:10.1038/hdy.2013.144

Pérez-Rodríguez P., D. Gianola, J.M. González-Camacho, J. Crossa, Y. Manès, and S. Dreisigacker. 2012. Comparison between linear and non-parametric regression models for genome-enabled prediction in wheat. G3: Genes Genomes Genetics. 2:1595–605.

Poland, J., and J. Rutkoski. 2016. Advances and Challenges in Genomic Selection for Disease Resistance. Annu. Rev. Phytopathol. 54:79–98. doi:10.1146/annurev-phyto-080615-100056

Polikar, R. 2006. Ensemble based systems in decision making. IEEE Circuits Syst. Mag. 6(3):21–45. doi:10.1109/MCAS.2006.1688199

R Core Team. 2016. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org/. (accessed Jul. 2017).

Roelfs, A.P., R.P. Singh, and E.E. Saari. 1992. Rust diseases of wheat: Concepts and methods of disease management. Mexico, D.F.: CIMMYT. 81 pp.

Rouse, M.N., L.E. Talbert, D. Singh, and J.D. Sherman. 2014. Complementary epistasis involving *Sr12* explains adult plant resistance to stem rust in Thatcher wheat (*Triticum aestivum* L.). Theor. Appl. Genet. 127:1549–1559. doi:10.1007/s00122-014-2319-6

Rutkoski, J., E. Heffner, and M. Sorrells. 2011. Genomic selection for durable stem rust resistance in wheat. Euphytica 179:161–173. doi:10.1007/s10681-010-0301-1

Rutkoski, J.E., J.A. Poland, R.P. Singh, J. Huerta-Espino, S. Bhavani, H. Barbier, M.N. Rouse, J.L. Jannink, and M.E. Sorrells. 2014. Genomic selection for quantitative adult plant stem rust resistance in wheat. Plant Genome. doi:10.3835/plantgenome2014.02.0006

Singh R.P., D.P. Hodson, J. Huerta-Espino, Y. Jin, P. Njau, R. Wanyera, S. Herrera-Foessel, and R.W. Ward. 2008. Will stem rust destroy the world's wheat crop? Adv. Agron. 98:271–309. doi:10.1016/S0065-2113(08)00205-8

Singh, R.P., D.P. Hodson, J. Huerta-Espino, Y. Jin, S. Bhavani, P. Njau, S. Herrera-Foessel, P.K. Singh, S. Singh, and V. Govindan. 2011. The emergence of Ug99 races of the stem rust fungus is a threat to world wheat production. Annu. Rev. Phytopathol. 49:465–481. doi:10.1146/annurev-phyto-072910-095423

Sonnenburg, S., M.L. Braun, C.S. Ong, S. Bengio, L. Bottou, G. Holmes, Y. LeCun, K. Muller, F. Pereira, C.E. Rasmussen, G. Rätsch, B. Scholkopf, A. Smola, P. Vincent, J. Weston, and R. Williamson. 2007. The need for open-source software in machine learning. J. Mach. Learn. Res. 8:2443–2466.

Sun, C., X.L. Wu, K.A. Weigel, G.J.M. Rosa, S. Bauck, B.W. Woodward, R.D. Schnabel, J.F. Taylor, and D. Gianola. 2012. An ensemble based approach to imputation of moderate density genotypes for genomic selection with application to Angus cattle. Genet. Res. 94:133–150. doi:10.1017/S001667231200033X

Thavamanikumar S., R. Dolferus, and B.L Thumma. 2015. Comparison of genomic selection models to predict flowering time and spike grain number in two hexaploid wheat doubled haploid populations. G3: Genes Genomes Genetics. 5:1991–1998.

Valiant, L.G. 1984. A theory of the learnable. Commun. ACM 27:1134–1142. doi:10.1145/1968.1972

Walters, R., C. Laurin, and G.H. Lubke. 2012. An integrated approach to reduce the impact of minor allele frequency and linkage disequilibrium on variable importance measures for genome-wide data. Bioinformatics 28:2615–2623. doi:10.1093/bioinformatics/bts483

Wasserman, P.D. 1993. Advanced methods in neural networks. New York: Van Nostrand Reinhold.