Machine Learning (2020) 109:2195–2212
https://doi.org/10.1007/s10994-020-05881-9

Predicting rice phenotypes with meta and multi-target learning

Oghenejokpeme I. Orhobor[1] · Nickolai N. Alexandrov[3] · Ross D. King[1,2,4]

Received: 11 July 2019 / Revised: 14 December 2019 / Accepted: 7 May 2020 / Published online: 2 August 2020

Abstract
The features in some machine learning datasets can naturally be divided into groups. This is the case with genomic data, where features can be grouped by chromosome. In many applications it is common for these groupings to be ignored, as interactions may exist between features belonging to different groups. However, including a group that does not influence a response introduces noise when fitting a model, leading to suboptimal predictive accuracy. Here we present two general frameworks for the generation and combination of meta-features when feature groupings are present. Furthermore, we make comparisons to multi-target learning, given that one is typically interested in predicting multiple phenotypes. We evaluated the frameworks and multi-target learning approaches on a genomic rice dataset where the regression task is to predict plant phenotype. Our results demonstrate that there are use cases for both the meta and multi-target approaches, given that overall, they significantly outperform the base case.

Keywords Rice · Bioinformatics · Machine learning · Meta-learning · Multi-target learning

1 Introduction

Machine learning algorithms are increasingly being adapted for the prediction of plant phenotypes (Grinberg et al. 2016, 2019). This task is most commonly regression based as most agronomic phenotypes are quantitative. This observation is true of rice (Spindel et al. 2015), the most agronomically important crop in the world, as a significant proportion of the global population relies on it for their dietary needs (Maclean et al.

Editors: Larisa Soldatova, Joaquin Vanschoren.

✉ Oghenejokpeme I. Orhobor
oo288@cam.ac.uk

[1] Department of Chemical Engineering and Biotechnology, University of Cambridge, Cambridge CB3 0AS, United Kingdom

[2] Alan Turing Institute, 96 Euston Rd, London NW1 2DB, United Kingdom

[3] The International Rice Research Institute, Los Banos, Philippines

[4] Department of Biology and Biological Engineering, Chalmers University of Technology, 41296 Gothenburg, Sweden

2013). With a growing global population, estimates suggest that we need to double rice yields over the next few decades (Ray et al. 2013; UN 2015). Therefore, it is crucial that we develop high yielding varieties that are resilient to an increase in biotic and abiotic stresses caused by climate change (Tai et al. 2014). The predictive phenotype models built for such plant populations are most commonly used in genomic selection (GS). In GS, these predictive models are used to estimate the likelihood that an individual in a population will express a trait of interest. This likelihood is expressed as a genomic estimated breeding value (GEBV) and is used by plant breeders to select individuals that will serve as parents for the next generation of progeny. Therefore, it is desirable that the models used to estimate GEBVs are as accurate as possible.

GS has only been recently adopted in rice (Grenier et al. 2015), and a model which is based on a single learning algorithm is often used for phenotype prediction, most commonly a variant of the best linear unbiased predictor (Grenier et al. 2015; Onogi et al. 2015). In this context, we propose the use of meta-learning, which seeks to improve overall predictive accuracy by leveraging the predictive power of multiple learning algorithms, and has been shown in other domains to outperform a single learning algorithm if the goal is to optimize predictive accuracy (Jahrer et al. 2010). The process can be broadly split into two main steps, a meta-feature generation step and a meta-feature integration step. In the former, a set of base models are built using a collection of learning algorithms. Each base model is then used to predict meta-features, which are predictions of a phenotype of interest. In the latter, the meta-features generated in the previous step are combined using another learning algorithm to form the final prediction.

A vital consideration we make is that of the nature of the attributes or features present in the input data used in building phenotype prediction models. The input data is often genomic, with features that are representative of the genetic diversity present in a population and are at different loci across an organism's genome (Spindel et al. 2015). These features are themselves representative of genes which control phenotypes and are located in different chromosomes. Therefore, the features in such genomic data can naturally be grouped by chromosome. In typical predictive experiments, the feature groupings by chromosome in the genomic data are ignored when models are built. The advantage of this approach is that potential interactions between features belonging to different chromosomes are captured. However, this may lead to suboptimal predictive accuracy if the features are in a chromosome with genes that are not associated with a phenotype, which introduces noise in a built model. Therefore, it might be the case that systematically diminishing the effects of features in irrelevant chromosomes might lead to higher accuracy. To address this problem, we propose two meta-learning frameworks which seek to improve phenotype prediction accuracy. The first ignores the feature groupings present in the input genomic data, and the other does not (Orhobor et al. 2018).

Given that one is typically interested in predicting multiple phenotypes, we considered the viability of multi-target regression for phenotype prediction, where the interest lies in building models that simultaneously predict multiple outputs (Aho et al. 2012; Appice and Džeroski 2007; Kocev et al. 2009; Spyromitros-Xioufis et al. 2012; Tsoumakas et al. 2014). The key insight of this approach is that by jointly learning models for different outputs, one is able to leverage the relationships between the outputs, which may be correlated, in building better models. This approach has been applied in various fields, and like meta-learning, has been shown to outperform a single base model (Han et al. 2012; Kocev et al. 2009; Tuia et al. 2011).

The remainder of this paper is organized as follows. In Sect. 2 we present the different considerations in meta-feature generation and integration, and in Sect. 3, we describe the proposed frameworks. In Sect. 4, our experimental setup is given, detailing the learners used in our evaluation. In Sect. 5 we discuss the outcome of evaluating the proposed frameworks, where our results show that there are use cases for both. Lastly, we conclude in Sect. 6.

2 Background

Rather than using a single learning algorithm, meta-learning seeks to improve the predictive accuracy of models used to predict phenotype by combining the predictive power of a set of base learners utilizing a combining/meta-level learner. For example, assume a rice population with input genomic data (learning set) where one is interested in predicting grain width. Furthermore, assume that the goal is to improve predictive accuracy by combining the predictive power of random forests (Breiman 2001) (RF) and support vector regression (Cortes and Vapnik 1995) (SVR) using simple linear regression (LR). Therefore, RF and SVR are the base learners while LR is the combining learner.

To amalgamate the predictive power of RF and SVR, they are both independently used to build a model to predict grain width, and the predictions made by these models are considered as grain width meta-features. Meta-features are typically generated by resampling the learning set using v-fold cross-validation (Breiman 1996; Parmanto et al. 1996), where each fold serves as a validation set and the remainder as a training set. We adopt this approach in the proposed frameworks. The first advantage that v-fold cross-validation offers is in computational expense with regards to time. Given the advances in genotyping and sequencing technologies, the genomic data used in phenotype prediction experiments typically have input features in the order of a million features (Alexandrov et al. 2015). Therefore, building a single model takes a substantial amount of time, so other resampling methods like the Monte-Carlo cross-validation (Xu et al. 2007) may be infeasible. The second advantage is in the reduction of overfitting. As stated earlier, genomic data can have on the order of a million input features; therefore there is potential for overfitting as it is often the case that the number of features far outnumber the number of samples ($p \gg n$). Using our example, assume 3-fold cross-validation in the meta-feature generation step. In this case, both RF and SVR are used to build three models each on the different training sets and used to predict three meta-feature vectors on the validation sets. This means that we end up with three independent meta-feature matrices with columns corresponding to the number of base learners. Therefore, three sets of combining weights can be learned using LR and applied to the predictions made on unseen data. By doing this, we get combining weights that do not closely fit to one set of examples. A similar approach has been applied to positive effect in super learners (Van der Laan et al. 2007).

The diversity of the set of base models used in generating the set of meta-features is vital, as it is desirable for the base models to be incorrect in different ways (Caruana et al. 2004). That is, it is better for their predictions on some test set to be wrong on different samples, so that the amalgamation of their predictions yield improved results. There are two main ways of achieving this. The first is to use a set of different base learners, which has been alluded to in our example, as they would make different assumptions about the nature of the relationships between the features in the input data (Džeroski and Ženko 2004). For example, RF might make predictions based on nonlinear interactions amongst the features, whereas nearest neighbour techniques (Altman 1992) which consider the level of relatedness between samples might yield a unique perspective. The second way of achieving model diversity is by varying the input data. That is, the input data can be split into multiple datasets which have different subsets of the features from the original. A base learner can then be used to build models on each of these new datasets, which are then used in the generation of meta-features. This approach is used in the stacked interval partial least squares framework (Ni et al. 2009), where meta-features are combined from various intervals in spectral data using partial least squares. We have adopted the first approach to be used with both of the proposed frameworks. The second is used only in the framework for which feature groupings are considered. The main difference between what we propose and the work using partial least squares (Ni et al. 2009) is that we use an ensemble of base learners for each input data subset.

Having generated a set of meta-features the next step is to integrate them, creating the final prediction. Using our example, this entails integrating the meta-feature predictions by RF and SVR. Several integration methods have been proposed. However, most are better suited to classification rather than regression problems (Džeroski and Ženko 2002; Ting and Witten 1999). In a regression setting, meta-feature integration is done using weights. These weights are coefficients which determine how much each base learner's meta-feature will influence the final prediction. A constant or dynamic weighting approach can be used (Merz 1998). Constant weighting in its simplest form involves averaging the meta-feature values for each sample. If the meta-features generated by the base models are incorrect on different samples but are all mostly accurate, averaging the meta-features improves overall accuracy by adjusting the incorrectly predicted samples. A more sophisticated constant weighting approach is to learn the weights using a combining learner, which is LR in our example. Note that on a test set, the learned weights are uniformly applied to every sample. We utilize both of these constant weighting approaches in the proposed procedures. In contrast to constant weighting, dynamic weighting assigns individual weights to each sample in a test set. This is done by learning individual weights for each sample in the test set using only the most closely related samples in the learning set (Rooney et al. 2004). This approach is computationally expensive in terms of time, and we do not use it in the proposed procedures. However, we conjecture that it may yield interesting results, and will be a subject of future study.

The natural feature groupings present in the genomic data used for phenotype prediction can also be thought of as views in multi-view learning. This assertion is based on the fact that the groups in this context are chromosomes which have genes that may influence a phenotype of interest. Therefore, each group of features represents a different perspective/view in terms of gene-phenotype associations. Several approaches have been proposed in multi-view learning (Xu et al. 2013), and multiple kernel learning (MKL) (Sonnenburg et al. 2006) is the most closely related to the current discourse. In typical multi-view learning problems, the views are often distinct, with different underlying structures and distributions of the input features. In MKL, learning algorithms that are best suited to each distinct view are used, and their predictions are then combined (Cortes et al. 2009; Lanckriet et al. 2004). This approach is similar to what we propose, in that a combining learner is used to integrate the meta-features of different learners. However, our proposal differs in that multiple learners are used within each group or view to form a consensus on their influence on a trait.

As stated in the introduction, multi-target learning involves simultaneously learning models for different outputs to leverage output relatedness. Multi-target learning approaches have been classified into problem transformation methods and algorithm adaptation methods (Borchani et al. 2015). In problem transformation methods, the model building process is modified to accomodate several outputs, which usually involves augmenting the predictive features with the outputs before building the model. Examples of such methods are multi-target regressor stacking, ensemble of regressor chains, and ensemble of regressor chains corrected (Spyromitros-Xioufis et al. 2012). We considered all three of these methods in our evaluation as they are most closely related to the proposed frameworks in that they can be used independent of a particular learning algorithm. For algorithm adaptation methods, known algorithms such as SVR, which are typically used in single target problems, are adapted for a multi-target setting. Several of such methods have been proposed (Abraham et al. 2013; Appice and Džeroski 2007; Ikonomovska et al. 2011; Sánchez-Fernández et al. 2004), however, we do not consider them in our evaluation and this could be a subject of future study in the phenotype prediction domain.

3 Proposed frameworks

In this section, we describe two proposed meta-learning frameworks, frameworks A and B respectively. Framework A is for a situation in which the feature groupings present in an input dataset are ignored, and Framework B is for a situation in which feature groupings are considered.

3.1 Framework A

The motivation for this framework is the overall improvement of phenotype prediction accuracy by leveraging the predictive power of multiple learning algorithms. In this case, we assume that although the features in an input dataset can be grouped by chromosome, these groupings are ignored when building a predictive model. Regarding the description of the procedure, we first give a description using an example, followed by a more formal one.

Assume a scenario where there is a learning and test genomic dataset with the goal of predicting grain width. The test set contains samples for which we want to predict their phenotype, and it is not used to build models. The two base learners are RF and SVR, and the combining learner is LR. We also assume v folds. For the meta-feature generation step, first split the learning data into v folds. Using each fold as a validation set and the remainder as a training set, build an RF and SVR model for grain width on the training set then predict learning meta-features using the validation set and also predict the test meta-features using the test set. At the end of this, v sets of learning and test meta-feature matrices are generated, all with two columns which correspond to predictions made by RF and SVR.

For the integration step, form a single test meta-feature matrix, $\mathbf{T}_{avg}$, by averaging the v predictions made by each base model (RF and SVR). Using LR, learn combining weights with each of the v learning meta-feature matrices. This produces v sets of weights. Apply each of these weights to $\mathbf{T}_{avg}$, producing v predictions. Finally, average these v predictions to form the final prediction for grain width. More formally:

Assume a learning set, a test set with samples for which we want to predict their phenotype, a set of base learners, a combining learner, and v cross-validation folds.

Step 1

1. Split the learning set into v folds, aiming for approximately equal number of samples in each fold.
2. For each v fold:
   (a) *validation set* = current fold.
   (b) *training set* = the combination of the other folds.
   (c) build b base models using base learners on the training set.
   (d) predict the validation response using base models, generating a meta-feature matrix $\mathbf{V}_v \in \mathbb{R}^{m \times b}$, where m is the number of samples in the vth fold and b is number of base models.
   (e) predict the test response using base models, generating a meta-feature matrix $\mathbf{T}_v \in \mathbb{R}^{n \times b}$, where n is the number of samples in the test set and b is number of base models.
3. Output:
   (a) a set of validation meta-features $\mathcal{V} = (\mathbf{V}_1, \ldots, \mathbf{V}_v)$.
   (b) a set of test meta-features $\mathcal{T} = (\mathbf{T}_1, \ldots, \mathbf{T}_v)$.

Step 2 Using $\mathcal{V}$ and $\mathcal{T}$ from step 1 and a combining learner ϕ:

1. For each base model ψ, with $\psi_1, \ldots \psi_v$ predictions in $(\mathbf{T}_1, \ldots, \mathbf{T}_v) \in \mathcal{T}$, $\psi_{avg} = 1/v \sum_{i=1}^{v} \psi_i$. Therefore the average predictions for all base models in $\mathcal{T}$ can be represented as $\mathbf{T}_{avg} \in \mathbb{R}^{n \times b}$, where n is the number of samples and b is number of base models.
2. Learn combining weights on each validation meta-feature set in $\mathcal{V}$ using the combining learner ϕ. This produces v weight sets which are applied to $\mathbf{T}_{avg}$, producing $\phi_1, \ldots \phi_v$ predictions. The final prediction is given by $\phi_{avg} = 1/v \sum_{i=1}^{v} \phi_i$.
3. Output ϕ_{avg}.

3.2 Framework B

Like framework A, the motivation for this framework is also to improve overall phenotype predictive accuracy by leveraging the predictive power of multiple learning algorithms. However, in contrast to framework A, feature groupings present in the input genomic data are considered. The rationale for this is that for phenotype prediction, including features which are in regions that have genes that are not associated with a trait might only serve to introduce noise in a built model, leading to suboptimal predictive accuracy. Therefore, systematically diminishing the influence of such features might be better.

For a general genomic dataset, it is assumed that the group to which each feature belongs is known, and all features in the dataset have been separated into their respective groups, c. That is, for a general dataset $\mathbf{D} \in \mathbb{R}^{m \times f}$, where m is the number of samples and f is number of features, $\mathbf{D}$ has been separated into c subsets, $\mathcal{D} = \mathbf{D}^1, \ldots, \mathbf{D}^c$, such that the intersection between the features in any pair of subsets must be empty and the union of the features in all subsets must be equal to the features in $\mathbf{D}$.

The procedure for this framework can be described using the example in Sect. 3.1. However, we assume that both the learning and test datasets have been split into their c subsets by chromosome. For the meta-feature generation step, first split the learning set into v folds across all c data subsets, ensuring that across each c subset the same samples are in each v split. Using each fold as a validation set and the remainder as a training set in all c subsets, build an RF and SVR model for grain width on each c training set and then predict the learning meta-features using the corresponding c validation set and also predict the test meta-features using the corresponding c test set. At the end of this, v sets of learning and test meta-feature matrices are generated for the c subsets, all with two columns, p, which correspond to predictions made by RF and SVR. Therefore, there are $v \times c$ meta-feature matrices for the learning and test sets. For the learning meta-feature matrices, merge all c subsets for each v fold. This produces v learning meta-feature sets, where each set has c pairs of RF and SVR meta-features, or $c \times p$ meta-features. For the test meta-feature matrices, first form a single test meta-feature matrix for each c subset, $\mathbf{T}^c_{avg}$, by averaging the v predictions made by each base model (RF and SVR) within each c subset. These c averaged test meta-feature matrices are then merged in the same order the learning meta-feature matrices were, forming $\mathbf{T}_{merged}$.

Using LR, learn combining weights with each of the v merged learning meta-feature matrices. This produces v sets of weights. Apply each of these weights to $\mathbf{T}_{merged}$, producing v predictions. Finally, average these v predictions to form the final prediction for grain width. More formally:

Assume a learning and a test set that have been split into their c subsets using the chromosome to which features belong, a set of base learners, a combining learner, and v cross-validation folds.

Step 1

1. Split all c learning set subsets into v folds, aiming for approximately equal number of samples in each fold, and ensuring that the same samples are in each v-fold across all subsets.
2. For each v fold and in each c subset:

   (a) *validation set* = current fold.
   (b) *training set* = the combination of the other folds.
   (c) build b base models using base learners on the training set.
   (d) predict the validation response using all trained models, generating a meta-feature matrix $\mathbf{V}^c_v \in \mathbb{R}^{m \times b}$, where m is the number of samples in the vth fold and b is the number of base models.
   (e) predict the test response using all trained models, generating a meta-feature matrix $\mathbf{T}^c_v \in \mathbb{R}^{n \times b}$, where n is the number of samples in the test set and b is the number of base models.

3. Generating:

   (a) a set of validation meta-features for each c subset, $\mathcal{V}^1, \dots, \mathcal{V}^c$, where $\mathcal{V}^c = (\mathbf{V}^c_1, \dots, \mathbf{V}^c_v)$.
   (b) a set of test meta-features for each c subset, $\mathcal{T}^1, \dots, \mathcal{T}^c$, where $\mathcal{T}^c = (\mathbf{T}^c_1, \dots, \mathbf{T}^c_v)$.

4. Merge $\mathcal{V}^1, \dots, \mathcal{V}^c$ in order for all v validation meta-feature sets, creating v merged validation meta-feature sets $\mathcal{V}_{merged} = (\mathbf{V}_1, \dots, \mathbf{V}_v) \in \mathbb{R}^{m \times p}$, where p is $b \times c$.
5. For each test meta-feature set subset $\mathcal{T}^1, \dots, \mathcal{T}^c$, average the v predictions of each base learner in $\mathbf{T}^c_1, \dots, \mathbf{T}^c_v$. This produces the average prediction matrices of all base models

for all c subsets, $\mathbf{T}^1_{avg}, \ldots, \mathbf{T}^c_{avg}$. Merge all c average prediction matrices in order to form $\mathbf{T}_{merged} \in \mathbb{R}^{n \times p}$, where p is $b \times c$.

6. Output:

   (a) the set of v merged validation meta-feature matrices $\mathcal{V}_{merged}$.
   (b) the merged test meta-feature matrix $\mathbf{T}_{merged}$.

Step 2 Using $\mathcal{V}_{merged}$ and $\mathbf{T}_{merged}$ from step 1 and a combining learner ϕ:

1. Learn combining weights on each validation meta-feature set in $\mathcal{V}_{merged}$ using the combining learner ϕ. This produces v weight sets which are applied to $\mathbf{T}_{merged}$, producing $\phi_1, \ldots \phi_v$ predictions. The final prediction is given by $\phi_{avg} = 1/v \sum_{i=1}^{v} \phi_i$.
2. Output ϕ_{avg}.

4 Experimental setup

In this section, we discuss the dataset and methods used in our evaluation.

4.1 Dataset

We evaluated the proposed procedures using data from the 3000 rice genomes project (Alexandrov et al. 2015), downloaded from http://SNP-Seek.irri.org/_download.zul. For the genotype data, we used version 0.4 of the core single nucleotide polymorphism (SNP) subset of 3000 rice genomes, which consists of 3023 samples and 996,009 markers. It is a filtered SNP set with a fraction of missing data at <20%. Using linkage disequilibrium in Plink (Purcell et al. 2007), we pruned this dataset using a window of 50 SNPs, a step size of 5, and with an r^2 value of 0.001, where r^2 is the allowed correlation coefficient between the SNPs. This generated a smaller dataset with 12,286 features which represent the twelve rice chromosomes. The total proportion of missing values in this dataset is approximately 7%. We converted each SNP call for all varieties to numeric values; class 1 homozygotes are represented with 1, class 2 homozygotes as -1, and heterozygotes with 0. Missing values were imputed using column means, as it has been shown that mean imputation is sufficient in cases where less than 20% of the data for each marker is missing (Rutkoski et al. 2013).

Twelve quantitative traits were considered: culm diameter, culm length, culm number, grain length, grain width, grain weight, days to heading, ligule length, leaf length, leaf width, panicle length, and seedling height. Only 2266 samples in the genotype data are represented in the trait data. Of this 2266 samples in the trait data, some of them have missing values for some traits. We created two datasets. In the first, we excluded samples with unavailable or missing trait data for each trait experiment. We used this in the initial evaluation of the proposed frameworks. Therefore, a variable number of samples was used in each trait experiment. In the second, we removed all samples with missing data for any trait. This dataset consists of 1865 samples, and we used it in the evaluation of the proposed frameworks and the multi-target regression approaches. We refer to these datasets as I and II respectively. The raw and processed forms of the data used in our experiments are available in the Mendeley Data Repository at http://dx.doi.org/10.17632/86ygms76pb.1.

4.2 Setup

In our evaluation of the proposed approaches we used $v = 5$ folds and split the dataset into learning (75%), and testing (25%) sets with random sampling. For multi-target regressor stacking (MTRS), we generated the training output meta-features using 5-fold internal cross-validation and the test set meta-features using the models built for the base case. For the ensemble of regressor chains (ERC) and the ensemble of regressor chains corrected (ERCC), we used 10 chains, and for ERCC we used 5-fold internal cross-validation to generate the training output meta-features. Predictive accuracy was calculated as the coefficient of determination (R^2). All experiments were performed in R (Ihaka and Gentleman 1996). The code for the initial evaluation of the proposed framework is available at https://github.com/oghenejokpeme/DS2018. The code for the multi-target evaluation is available at https://github.com/oghenejokpeme/DSMLSE.

For the learners that require parameter tuning, we performed parameter selection using a grid search and cross-validation on the training data. We opted for grid search over random search (Bergstra and Bengio 2012) as the parameters which require tuning and the range of values we explored for these parameters were modest. This can be seen in the provided source code. We considered three sets of learners. Learners that take feature groupings into account, a set of base learners which do not take groupings into account and a set of combining learners.

4.3 Group learners

In our evaluation, we considered learners which take feature groupings into account. These learners are the group least absolute selection and shrinkage operator (Friedman et al. 2010) (GLASSO), group bridge-penalized regression (Huang et al. 2009) (GBRGE), and group minimax concave penalty (Breheny and Huang 2009) (GMCP). The optimal value for lambda along the regularization path was chosen using five-fold internal cross-validation for GLASSO. For GBRIDGE and GMCP, the Akaike Information Criteria was used as it has been shown to produce slightly better accuracies (Ogutu and Piepho 2014).

4.4 Base learners

The base learners used are the ridge regression best linear unbiased predictor (Endelman 2011) (RBLUP), random forests (RF), gradient boosted machines (Friedman 2001) (GBM), support vector regression (Cortes and Vapnik 1995) (SVR), k nearest neighbors (Altman 1992) (KNN), and eXtreme gradient boosting (Chen and He 2015) (XGB). RBLUP is specially designed for genomic predictions and has no parameters that require tuning. For RF the default of 1/3 the total number of variables is considered at each split, five observations are used for each terminal node, and 1000 trees were grown for each forest. For GBM we used a shrinkage parameter of 0.1, interaction depth of 6, 15 minimum number of observations in each node, and 1500 trees were grown. For SVR we used a radial basis kernel, and the hyperparameters were tuned using a grid search. XGB were also tuned with a grid search. Lastly, the optimal number of neighbors, n, used in the KNN models were chosen using cross-validation, where $1 \leq n \leq 30$.

4.5 Combining learners

The combining learners used are linear regression (LR), gradient descent (Kivinen and Warmuth 1997) (GD), kernel regularized least squares (Hainmueller and Hazlett 2014) (KRLS), ridge regression (Tibshirani 1996) (RR), and principal component regression (Jolliffe 1982) (PCR). The regularization parameter for RR was selected using internal cross-validation. A radial basis kernel was used with KRLS, and the bandwidth and regularization parameters were chosen using a grid search. For PCR the number of components used was chosen using internal cross-validation.

5 Results

In this section we discuss the results from the evaulation of the proposed frameworks and the multi-target regression approaches.

5.1 Evaluation of frameworks

The results discussed in this section are from the evaluation of the proposed approaches using dataset I (see Sect. 4.1).

5.1.1 Group and base learner performance

The group and base learner performances serve as a baseline for the performance of the combining learners on the proposed frameworks. For the twelve rice traits considered, a base learner which does not take feature groupings into account outperforms all other learners on ten of the twelve traits (Table 1). In general SVM and XGB outperform all other learners, even outperforming RBLUP, a learner designed for genomic predictions.

Table 1 Predictive accuracy (R^2) of the group and base learners

Trait	GLASSO	GBRGE	GMCP	RBLUP	RF	GBM	SVR	KNN	XGB
Culm diameter	0.164	–	–	0.163	0.155	0.100	0.179	0.097	**0.171**
Culm length	0.549	0.512	0.318	0.544	0.533	0.516	**0.559**	0.529	0.552
Culm number	0.213	–	–	0.216	0.218	0.191	0.217	0.217	**0.219**
Grain length	0.379	**0.387**	0.380	0.370	0.337	0.306	0.383	0.249	**0.387**
Grain width	0.462	0.458	0.455	0.483	0.446	0.439	0.480	0.379	**0.489**
Grain weight	0.363	0.325	0.318	0.370	0.353	0.299	**0.379**	0.281	**0.379**
Heading date	0.657	0.615	0.591	0.674	0.660	0.654	0.680	0.691	**0.693**
Ligule length	0.368	0.282	0.236	0.374	0.355	0.327	**0.380**	0.310	0.370
Leaf length	0.390	0.291	0.081	0.400	0.398	0.365	**0.419**	0.375	0.397
Leaf width	0.404	0.344	0.334	0.399	0.403	0.395	0.413	0.364	**0.423**
Panicle length	0.411	0.349	0.302	0.412	0.405	0.383	**0.437**	0.342	0.428
Seedling height	**0.225**	–	–	0.221	0.188	0.173	0.207	0.168	0.199

The best performing learner is in boldface. '–' are cases were the model building failed, which to the best of our knowledge was due to multicollinearity

We argue that this is the case for two reasons, (1) the traits considered are controlled by features with strong nonlinear interactions which RBLUP does not detect, and (2) SVM and XGB are better able to deal with a large number of irrelevant features. This is significant as recent advances in genotyping and sequencing technologies mean that genomic data is now being generated on the order of a million features, most of which are irrelevant in a built model. Therefore, rather than using traditional methods like RBLUP for phenotype prediction, more sophisticated methods like XGB should also be considered if one wants to use a single learning algorithm. The best performing group learner was GLASSO, which excludes features belonging to groups with low signal by assigning a zero coefficient to all features in such groups. It outperforms all other learners on one trait, seedling height, suggesting that it is indeed the case that some traits might benefit from excluding features from certain chromosomes. We assumed a null hypothesis that there is no difference in performance between GLASSO, the best performing group based method, and SVR and XGB, the best base learners. A sign test showed that with a significance level of 0.05, the null hypothesis can be rejected in both cases, as both comparisons (GLASSO-SVR and GLASSO-XGB) produced a p-value of 0.006.

5.1.2 Combining learner performance

In our evaluation of the proposed frameworks, the six base learners outlined in Sect. 4.4 were used to generate meta-features for twelve rice traits. To evaluate the frameworks five learning algorithms were then used as combining learners to integrate the generated meta-features. We found that in a meta-learning setting, some traits benefit when the feature groupings are ignored in the meta-feature generation and integration steps, while others benefit from having the feature groupings considered. We argue that the latter case occurs for two reasons. Firstly, each group has its own unique set of meta-features, generated by its own set of models. Therefore, noise is not introduced in these models from groups that may not be strongly associated with a phenotype. Secondly, the meta-features for a group represent the degree of association that a group has with a phenotype. Therefore, generating meta-features for each feature group in isolation before learning combining weights aids a combining learner in estimating the amount of influence each group has on a phenotype.

Comparing frameworks A and B based on the performance of the combining learners showed that for LR, framework A outperforms B on eleven of the twelve traits. For GD, framework A outperforms B in nine of the twelve traits. For KRLS, framework A outperforms B on eight of the twelve traits. For RR, framework A outperforms B on ten traits, they perform equally well on one trait, and framework B outperforms A on one trait. For PCR, framework A outperforms B on nine of the twelve traits, they perform equally well on two traits, and framework B outperforms A on one trait. See Table 2 for the results. These results suggest that on a per learner basis, framework A, in which feature groupings are ignored, is generally the better meta-learning approach. For each combining learner, we assumed a null hypothesis that there is no difference in performance between frameworks A and B. A sign test showed that with a significance level of 0.05, the null hypothesis cannot be rejected for GD, KRLS and PCR, with p-values of 0.146, 0.774, and 0.146 respectively. Whereas, the null hypothesis can be rejected for LR and RR, with p-values of 0.006 and 0.039 respectively. This suggests that the extent to which a given framework outperforms the other on a particular learner is learner dependent.

Evaluating the performance of the frameworks on a per trait basis irrespective of the combining learner tells a different story. In this case, framework A and B perform better on six

Table 2 Predictive accuracy (R^2) of the combining learners on frameworks A and B

	LR		GD		KRLS		RR		PCR	
Trait	A	B	A	B	A	B	A	B	A	B
Culm diameter	**0.175**	0.119	*0.178*	0.170	*0.178*	0.170	**0.177**	0.172	**0.177**	0.168
Culm length	**0.561**	0.552	**0.561**	–	0.566	*0.569*	**0.564**	0.563	**0.566**	**0.566**
Culm number	**0.236**	0.214	0.232	*0.242*	0.235	**0.239**	**0.233**	0.231	**0.236**	**0.236**
Grain length	**0.391**	0.378	**0.378**	0.348	**0.397**	0.388	**0.398**	0.388	*0.402*	0.383
Grain width	**0.497**	0.472	**0.477**	0.425	*0.499*	0.490	**0.497**	0.488	**0.498**	0.488
Grain weight	**0.379**	0.333	**0.371**	0.338	**0.376**	0.362	*0.382*	0.365	**0.380**	0.356
Heading date	0.692	**0.703**	**0.692**	–	0.698	*0.710*	0.699	**0.708**	0.699	**0.705**
Ligule length	**0.380**	0.374	0.381	*0.383*	0.381	**0.382**	**0.381**	0.375	**0.380**	0.372
Leaf length	**0.411**	0.385	**0.412**	0.398	*0.420*	0.411	**0.416**	0.411	**0.415**	0.409
Leaf width	**0.419**	0.389	**0.416**	0.401	0.419	*0.420*	**0.419**	0.416	**0.419**	0.409
Panicle length	**0.439**	0.394	0.429	*0.443*	**0.431**	0.429	**0.437**	**0.437**	**0.439**	0.438
Seedling height	*0.219*	0.168	**0.218**	0.193	**0.218**	0.214	**0.215**	0.210	**0.217**	0.210

The best performing framework for each learner is in boldface. The overall best performing learner-framework pair is in italics. '–' are cases where the model building failed, which to the best of our knowledge was due to multicollinearity

traits each. The results show that no particular learner performs better on any trait-framework pair. This suggests that if the proposed approaches are to be used, combining learners should be chosen based on the framework of choice and the trait one is interested in predicting. One way of making this decision might be to modify the well-known model selection procedure used to select a single model from a set of competing models. However, we acknowledge that this will be computationally expensive given the number of models that are built in both frameworks. It is also worth noting that GLASSO, a single model approach, outperforms both frameworks (Table 2). Therefore, one should also consider single learner approaches.

For each trait, we compared the best performing combining learner on both frameworks to the best performing base learner. For framework A, we found that the best performing combining learner performs just as well or outperforms the best performing base learner on ten of twelve traits. For framework B, we found that the best performing combining learner performs just as well or outperforms the best performing base learner on eight of twelve traits. See Table 3. These results show that it is not always the case that one of the meta-learning approaches outperforms a single base model. However, the best performing combining learner on at least one of the proposed meta-learning approaches outperforms the best performing single base learner on ten of the twelve traits. We assumed a null hypothesis that there is no difference in performance between the best performing learner on framework A and the base case, the best performing learner on framework B and the base case, and the best performing learner on both frameworks and the base case. A sign test showed that with a significance level of 0.05, the null hypothesis can be rejected for the first and third cases but not for the second case, with p-values of 0.039, 0.039 and 0.388 respectively. Therefore, we conclude that the proposed frameworks generally increase the accuracy by which plant phenotype can be predicted by leveraging the predictive power of multiple learning algorithms in scenarios where the feature groupings present in genomic data are considered and ignored.

Table 3 Predictive accuracy (R^2) of the best performing combining learners on frameworks A and B in comparison to the best performing base learner

Trait	A	B	Base
Culm diameter	**0.178**	0.172	0.171
Culm length	0.566	**0.569**	0.559
Culm number	0.236	**0.242**	0.219
Grain length	**0.402**	0.388	0.387
Grain width	**0.499**	0.490	0.489
Grain weight	**0.382**	0.365	0.379
Heading date	0.699	**0.710**	0.693
Ligule length	0.381	**0.383**	0.380
Leaf length	0.**420**	0.411	0.419
Leaf width	0.419	0.420	**0.423**
Panicle length	0.439	**0.443**	0.437
Seedling height	0.219	0.214	**0.221**

The best performing meta-learning or single model approach is in boldface

5.2 Comparison to multi-target learning

In this section, we discuss the results from evaluating the proposed frameworks and problem transformation multi-target approaches using dataset II (see Sect. 4.1). We used only SVR and XGB as the learners of choice for the experiments with the proposed frameworks and with the multi-target approaches as they were the two best performing learners in our initial evaluation (see Table 1). We used LR as the combining learner for the meta-features generated by both frameworks.

In the base case, we found that XGB performs just as well or outperforms SVR on nine of the twelve traits, which is consistent with the results in Table 1. However, one of either MTRS, ERC or ERCC outperforms the base case for both SVR and XGB for all traits (see Tables 4, 5), suggesting that even in a high dimensional setting such as this, where approximately 12,000 features are present in the genome data, the signal from the augmented

Table 4 Predictive accuracy (R^2) of support vector regression (SVR) on the base case for single traits, multi-target regressor stacking (MTRS), ensemble of regressor chains (ERC) and ensemble of regressor chains corrected (ERCC)

Trait	Base	MTRS	ERC	ERCC
Culm diameter	0.126	0.137	**0.138**	0.134
Culm length	0.555	0.563	0.561	**0.564**
Culm number	0.199	**0.206**	0.195	0.200
Grain length	0.448	0.443	**0.452**	0.446
Grain width	0.457	0.458	**0.460**	0.458
Grain weight	0.294	**0.295**	0.274	**0.295**
Heading date	0.660	**0.665**	0.664	0.660
Ligule length	0.355	**0.375**	0.359	0.369
Leaf length	0.375	**0.400**	0.399	0.397
Leaf width	0.406	0.403	**0.407**	0.404
Panicle length	0.410	**0.418**	**0.418**	0.415
Seedling height	0.222	0.226	**0.230**	0.225

The best performing approach is in boldface

Table 5 Predictive accuracy (R^2) of eXtreme gradient boosting (XGB) on the base case for single traits, multi-target regressor stacking (MTRS), ensemble of regressor chains (ERC) and ensemble of regressor chains corrected (ERCC)

Trait	Base	MTRS	ERC	ERCC
Culm diameter	0.164	0.165	0.145	**0.168**
Culm length	0.566	0.559	0.558	**0.570**
Culm number	0.201	0.216	0.208	**0.219**
Grain length	0.463	**0.477**	0.461	0.462
Grain width	0.511	0.506	0.511	**0.519**
Grain weight	0.304	**0.314**	0.297	0.301
Heading date	0.655	**0.661**	**0.661**	**0.661**
Ligule length	0.368	**0.376**	**0.376**	0.373
Leaf length	0.404	0.411	0.408	**0.414**
Leaf width	0.417	0.402	0.418	**0.420**
Panicle length	0.383	0.411	0.397	**0.412**
Seedling height	0.192	0.177	**0.201**	0.196

The best performing approach is in boldface

features can be identified amidst the noise. For both SVR and XGB we assumed a null hypothesis that there is no difference in performance in three cases: the base case and MTRS, the base case and ERC, and the base case and ERCC. For SVR, a sign test showed that the null hypothesis can be rejected for first and second cases but not for the third, with *p*-values of 0.039, 0.039 and 0.146 respectively. For XGB, a sign test showed that the null hypothesis cannot be rejected for the first and second cases with *p*-values of 0.388 and 0.774 respectively, but can be rejected for the third case with a *p*-value of 0.039. We also assumed a null hypothesis that there is no difference in performance between the base case and the best performing multi-target approaches. A sign test showed that the null hypothesis can be rejected for both SVR and XGB with a *p*-value of 0.0004 at a significance level of 0.05. These results demonstrate that in a multi-phenotype prediction setting, multi-target approaches should be used if one wants to optimize predictive accuracy.

In comparison to the proposed frameworks, one of either frameworks A or B outperforms base SVR on nine of the twelve traits and base XGB on eight of the twelve traits, which is also consistent with the results in the initial evaluation (see Table 3). But more interestingly, we compared the performance of frameworks A and B to that of the unweighted average predictions of SVR and XGB for MTRS, ERC and ERCC. We found that at least one of the multi-target approaches outperformed the frameworks on nine of the twelve traits (Table 6). We assumed a null hypothesis that there is no difference in performance between the proposed frameworks and the unweighted average predictions of SVR and XGB for MTRS, ERC and ERCC. With a signficance level of 0.05, a sign test showed that for framework A the null hypothesis cannot be rejected for MTRS, ERC, and ERCC, with *p*-values of 0.146, 0.388, and 0.146 respectively. For framework B, a sign test showed that the null hypothesis can be rejected for MTRS, ERC, and ERCC, with *p*-values of 0.006, 0.0004, and 0.0004 respectively. We argue that these results further demonstrate the utility of the multi-target approaches and highlights the need to consider weighted approaches for averaging predictions in a multi-target setting.

Table 6 Predictive accuracy (R^2) of frameworks A and B using SVR and XGB as the base learners and LR as the combining learners

Trait	A	B	$MTRS_{avg}$	ERC_{avg}	$ERCC_{avg}$
Culm diameter	**0.167**	0.143	0.162	0.152	0.160
Culm length	0.566	0.559	0.568	0.568	**0.573**
Culm number	**0.224**	0.206	0.220	0.208	0.217
Grain length	0.467	0.448	**0.481**	0.476	0.473
Grain width	0.495	0.463	0.497	0.499	**0.501**
Grain weight	0.292	0.292	**0.314**	0.293	0.305
Heading date	0.664	0.646	0.670	**0.669**	0.666
Ligule length	0.375	0.364	**0.382**	0.374	0.376
Leaf length	0.396	0.374	**0.412**	0.410	0.409
Leaf width	**0.419**	0.416	0.410	0.417	0.418
Panicle length	0.400	0.396	**0.421**	0.413	0.419
Seedling height	0.210	0.210	0.212	**0.222**	0.218

$MTRS_{avg}$, ERC_{avg}, $ERCC_{avg}$ correspond to the performance of the unweighted averaging of the predictions made by SVR and XGB in Tables 4 and 5. The overall best approach is in boldface

6 Conclusion

In this paper, we investigated the prediction of rice phenotypes. We argued that because rice is the most agronomically important crop in the world, the models used by plant breeders for the selection of the parents that will produce progeny with desirable traits should be as accurate as possible. We proposed that meta-learning, which leverages the predictive power of multiple learning algorithms, could improve the accuracy by which rice and plant phenotypes, in general, can be predicted. We noted that the genomic datasets often used in predicting phenotype consists of features that can naturally be separated into groups by chromosome and argued that including features from chromosomes which may not influence a trait might lead to suboptimal predictive accuracy, as it introduces noise in a built model. With this in mind, we proposed two meta-learning frameworks, one which does not consider feature groupings (framework A) and another which does (framework B). Our results show that framework A generally outperforms framework B on a per learner level of analysis, but that they perform equally well on a per trait level of analysis. But more importantly, the results show that the best performing meta-learner on at least one of the proposed meta-learning approaches outperforms the best performing single base learner on ten of the twelve traits. Furthermore, we evaluated three problem transformation multi-target learning approaches: multi-target regressor stacking, ensemble of regressor chains, and ensemble of regressor chains corrected. We demonstrated that in cases where a single learner is used or the predictions made by multiple learners are combined, the multi-target learning approaches performed best.

In future work, we intend to apply the proposed procedures to other agronomically relevant crops like wheat and barley, and possibly on human population data. Furthermore, we intend to extend the proposed procedures by introducing meta-feature pruning, which aids in the selection of the meta-features that will eventually be integrated (Mendes-Moreira et al. 2012). There are several methods (Caruana et al. 2004) that can be used to perform meta-feature pruning, and we conjecture that the different techniques will perform differently on the proposed frameworks. As stated in the discussion of considerations we made in developing the proposed frameworks (Sect. 2), we also intend to extend the proposed frameworks by introducing dynamic weighting for the integration of meta-features. It would also be interesting to apply these extentions to problem transformation multi-target approaches given a multiple learner scenario.

References

Abraham, Z., Tan, P. N., Winkler, J., Zhong, S., Liszewska, M., et al. (2013). Position preserving multi-output prediction. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 320–335). Springer.

Aho, T., Ženko, B., Džeroski, S., & Elomaa, T. (2012). Multi-target regression with rule ensembles. *Journal of Machine Learning Research*, *13*(Aug), 2367–2407.

Alexandrov, N., Tai, S., Wang, W., Mansueto, L., Palis, K., Fuentes, R. R., et al. (2015). SNP-Seek database of SNPs derived from 3000 rice genomes. *Nucleic Acids Research*, *43*(D1), D1023–D1027.

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, *46*(3), 175–185.

Appice, A., & Džeroski, S. (2007). Stepwise induction of multi-target model trees. In *European conference on machine learning* (pp. 502–509). Springer.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*(Feb), 281–305.

Borchani, H., Varando, G., Bielza, C., & Larrañaga, P. (2015). A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *5*(5), 216–233.

Breheny, P., & Huang, J. (2009). Penalized methods for bi-level variable selection. *Statistics and its Interface*, *2*(3), 369.

Breiman, L. (1996). Stacked regressions. *Machine Learning*, *24*(1), 49–64.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Caruana, R., Niculescu-Mizil, A., Crew, G., & Ksikes, A. (2004). Ensemble selection from libraries of models. In *Proceedings of the twenty-first international conference on machine learning* (p. 18). ACM.

Chen, T., & He, T. (2015). xgboost: eXtreme Gradient Boosting. R package version 0.4-2 (2015)

Cortes, C., Mohri, M., & Rostamizadeh, A. (2009). Learning non-linear combinations of kernels. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, & A. Culotta (Eds.), *Advances in neural information processing systems 22* (pp. 396–404). Curran Associates, Inc.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Džeroski, S., & Ženko, B. (2002). Stacking with multi-response model trees. In *International workshop on multiple classifier systems* (pp. 201–211). Springer.

Džeroski, S., & Ženko, B. (2004). Is combining classifiers with stacking better than selecting the best one? *Machine Learning*, *54*(3), 255–273.

Endelman, J. B. (2011). Ridge regression and other kernels for genomic selection with R package rrBLUP. *The Plant Genome*, *4*(3), 250–255.

Friedman, J., Hastie, T., & Tibshirani, R. (2010). A note on the group lasso and a sparse group lasso. arXiv preprint arXiv:1001.0736

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, *29*(5), 1189–1232.

Grenier, C., Cao, T. V., Ospina, Y., Quintero, C., Châtel, M. H., Tohme, J., et al. (2015). Accuracy of genomic selection in a rice synthetic population developed for recurrent selection breeding. *PloS ONE*, *10*(8), e0136594.

Grinberg, N. F., Lovatt, A., Hegarty, M., Lovatt, A., Skøt, K. P., Kelly, R., et al. (2016). Implementation of genomic prediction in *Lolium perenne* (L.) breeding populations. *Frontiers in Plant Science, 7*, 133.

Grinberg, N. F., Orhobor, O. I., & King, R. D. (2019). An evaluationof machine-learning for predicting phenotype: Studies in yeast, rice, and wheat. *Machine Learning*. https://doi.org/10.1007/s10994-019-05848-5.

Hainmueller, J., & Hazlett, C. (2014). Kernel regularized least squares: Reducing misspecification bias with a flexible and interpretable machine learning approach. *Political Analysis, 22*(2), 143–168.

Han, Z., Liu, Y., Zhao, J., & Wang, W. (2012). Real time prediction for converter gas tank levels based on multi-output least square support vector regressor. *Control Engineering Practice*, *20*(12), 1400–1409.

Huang, J., Ma, S., Xie, H., & Zhang, C. H. (2009). A group bridge approach for variable selection. *Biometrika*, *96*(2), 339–355.

Ihaka, R., & Gentleman, R. (1996). R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, *5*(3), 299–314.

Ikonomovska, E., Gama, J., & Džeroski, S. (2011). Incremental multi-target model trees for data streams. In *Proceedings of the 2011 ACM symposium on applied computing* (pp. 988–993). ACM.

Jahrer, M., Töscher, A., & Legenstein, R. (2010). Combining predictions for accurate recommender systems. In *Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 693–702). ACM.

Jolliffe, I. T. (1982). A note on the use of principal components in regression. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *31*(3), 300–303.

Kivinen, J., & Warmuth, M. K. (1997). Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, *132*(1), 1–63.

Kocev, D., Džeroski, S., White, M. D., Newell, G. R., & Griffioen, P. (2009). Using single-and multi-target regression trees and ensembles to model a compound index of vegetation condition. *Ecological Modelling*, *220*(8), 1159–1168.

Lanckriet, G. R., Cristianini, N., Bartlett, P., Ghaoui, L. E., & Jordan, M. I. (2004). Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, *5*(Jan), 27–72.

Maclean, J., Hardy, B., & Hettel, G. (2013). *Rice Almanac: Source book for one of the most important economic activities on earth*. Los Banos: International Rice Research Institute.

Mendes-Moreira, J., Soares, C., Jorge, A. M., & Sousa, J. F. D. (2012). Ensemble approaches for regression: A survey. *ACM Computing Surveys (CSUR)*, *45*(1), 10.

Merz, C. J. (1998). Classification and regression by combining models. Ph.D. thesis, University of California Irvine.

Ni, W., Brown, S. D., & Man, R. (2009). Stacked partial least squares regression analysis for spectral calibration and prediction. *Journal of Chemometrics*, *23*(10), 505–517.

Ogutu, J.O., & Piepho, H.P. (2014). Regularized group regression methods for genomic prediction: Bridge, MCP, SCAD, group bridge, group lasso, sparse group lasso, group MCP and group SCAD. In *BMC Proceedings* (vol. 8, p. S7). BioMed Central.

Onogi, A., Ideta, O., Inoshita, Y., Ebana, K., Yoshioka, T., Yamasaki, M., et al. (2015). Exploring the areas of applicability of whole-genome prediction methods for asian rice (*Oryza sativa* L.). *Theoretical and Applied Genetics*, *128*(1), 41–53.

Orhobor, O. I., Alexandrov, N. N., & King, R. D. (2018). Predicting rice phenotypes with meta-learning. In *International conference on discovery science* (pp. 144–158). Springer.

Parmanto, B., Munro, P. W., & Doyle, H. R. (1996). Reducing variance of committee prediction with resampling techniques. *Connection Science*, *8*(3–4), 405–426.

Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M. A., Bender, D., et al. (2007). PLINK: a tool set for whole-genome association and population-based linkage analyses. *The American Journal of Human Genetics*, *81*(3), 559–575.

Ray, D. K., Mueller, N. D., West, P. C., & Foley, J. A. (2013). Yield trends are insufficient to double global crop production by 2050. *PloS ONE*, *8*(6), e66428.

Rooney, N., Patterson, D., Anand, S., & Tsymbal, A. (2004). Dynamic integration of regression models. In *International workshop on multiple classifier systems* (Vol. 3077, pp. 164–173).

Rutkoski, J. E., Poland, J., Jannink, J., & Sorrells, M. E. (2013). Imputation of unordered markers and the impact on genomic selection accuracy. *G3: Genes, Genomes, Genetics*, *3*(3), 427–439.

Sánchez-Fernández, M., de Prado-Cumplido, M., Arenas-García, J., & Pérez-Cruz, F. (2004). Svm multiregression for nonlinear channel estimation in multiple-input multiple-output systems. *IEEE Transactions on Signal Processing*, *52*(8), 2298–2307.

Sonnenburg, S., Rätsch, G., Schäfer, C., & Schölkopf, B. (2006). Large scale multiple kernel learning. *Journal of Machine Learning Research*, *7*(Jul), 1531–1565.

Spindel, J., Begum, H., Akdemir, D., Virk, P., Collard, B., Redoña, E., et al. (2015). Genomic selection and association mapping in rice (*Oryza sativa*): effect of trait genetic architecture, training population composition, marker number and statistical model on accuracy of rice genomic selection in elite, tropical rice breeding lines. *PLoS Genetics*, *11*(2), e1004982.

Spyromitros-Xioufis, E., Tsoumakas, G., Groves, W., Vlahavas, I. (2012). Multi-label classification methods for multi-target regression (pp. 1159–1168). arXiv preprint arXiv:1211.6581.

Tai, A. P., Martin, M. V., & Heald, C. L. (2014). Threat to future global food security from climate change and ozone air pollution. *Nature Climate Change*, *4*(9), 817–821.

Tibshirani, R. (1996). Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288.

Ting, K. M., & Witten, I. H. (1999). Issues in stacked generalization. *Journal of Artificial Intelligence Research*, *10*, 271–289.

Tsoumakas, G., Spyromitros-Xioufis, E., Vrekou, A., & Vlahavas, I. (2014). Multi-target regression via random linear target combinations. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 225–240). Springer.

Tuia, D., Verrelst, J., Alonso, L., Pérez-Cruz, F., & Camps-Valls, G. (2011). Multioutput support vector regression for remote sensing biophysical parameter estimation. *IEEE Geoscience and Remote Sensing Letters*, *8*(4), 804–808.

UN. (2015). U.N.: World Population Prospects: The 2015 Revision, Key Findings and Advance Tables. Working Paper, No. ESA/P/WP. 241.

Van der Laan, M. J., Polley, E. C., & Hubbard, A. E. (2007). Super learner. *Statistical Applications in Genetics and Molecular Biology*, *6*(1). https://www.degruyter.com/view/journals/sagmb/6/1/article-sagmb.2007.6.1.1309.xml.xml.

Xu, C., Tao, D., & Xu, C. (2013). A survey on multi-view learning. arXiv preprint arXiv:1304.5634

Xu, L., Jiang, J. H., Zhou, Y. P., Wu, H. L., Shen, G. L., & Yu, R. Q. (2007). MCCV stacked regression for model combination and fast spectral interval selection in multivariate calibration. *Chemometrics and Intelligent Laboratory Systems*, *87*(2), 226–230.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.