G3 Genes | Genomes | Genetics

GENOMIC PREDICTION

Benchmarking Parametric and Machine Learning Models for Genomic Prediction of Complex Traits

Christina B. Azodi,* Emily Bolger,† Andrew McCarren,‡ Mark Roantree,‡ Gustavo de los Campos,§,,††,1 and Shin-Han Shiu*,‡‡,1**

†Department of Mathematics, Moravian College, Bethlehem, PA, ‡Insight Centre for Data Analytics, School of Computing, Dublin City University, Dublin 9, Ireland, *Department of Plant Biology , §Department of Epidemiology & Biostatistics, **Department of Statistics & Probability, ††Institute for Quantitative Health Science and Engineering, and ‡‡Department of Computational, Mathematics, Science, and Engineering, Michigan State University, East Lansing, MI, 48824

ORCID IDs: 0000-0002-6097-606X (C.B.A.); 0000-0002-7297-0984 (A.M.); 0000-0001-5692-7129 (G.d.l.C.); 0000-0001-6470-235X (S.-H.S.)

ABSTRACT The usefulness of genomic prediction in crop and livestock breeding programs has prompted efforts to develop new and improved genomic prediction algorithms, such as artificial neural networks and gradient tree boosting. However, the performance of these algorithms has not been compared in a systematic manner using a wide range of datasets and models. Using data of 18 traits across six plant species with different marker densities and training population sizes, we compared the performance of six linear and six non-linear algorithms. First, we found that hyperparameter selection was necessary for all non-linear algorithms and that feature selection prior to model training was critical for artificial neural networks when the markers greatly outnumbered the number of training lines. Across all species and trait combinations, no one algorithm performed best, however predictions based on a combination of results from multiple algorithms (*i.e.*, ensemble predictions) performed consistently well. While linear and non-linear algorithms performed best for a similar number of traits, the performance of non-linear algorithms vary more between traits. Although artificial neural networks did not perform best for any trait, we identified strategies (*i.e.*, feature selection, seeded starting weights) that boosted their performance to near the level of other algorithms. Our results highlight the importance of algorithm selection for the prediction of trait values.

KEYWORDS
Genomic selection
artificial neural network
genotype-to-phenotype
Genomic Prediction
GenPred
Shared Data Resources

The ability to predict complex traits from genotypes is a grand challenge in biology and is accelerating the speed of crop and livestock breeding (Heffner *et al.* 2009; Lorenz *et al.* 2011; Jonas and de Koning 2013; Desta and Ortiz 2014). Genomic Prediction (GP, aka Genomic Selection), the use of genome-wide genetic markers to predict complex traits, was originally proposed by Meuwissen *et al.* (Meuwissen *et al.* 2001) as a solution to the limitations of Marker-Assisted Selection (MAS) where only a limited number of previously identified markers with the strongest associations are used to select the best lines. GP is particularly well-suited for the prediction of quantitative traits controlled by many small-effect alleles (Ribaut and Ragot 2007). A major challenge in using GP is estimating the effects of a large number of makers (p) using phenotype information of a comparatively limited number of individuals (n) (*i.e.*, $P >> n$) (Meuwissen *et al.* 2001). To address this challenge, Meuwissen *et al.* first presented three statistical methods for GP (Meuwissen *et al.* 2001). The first was a linear mixed model called ridge regression Best Linear Unbiased Prediction (rrBLUP), which uniformly shrinks the marker effects. The other two were Bayesian approaches, BayesA (BA) and BayesB (BB), which both differentially shrink the marker effects and with BB also performing variable selection. Since then, additional approaches have been shown to be useful for GP, including Least Absolute Angle and Selection Operator (LASSO) (Usai *et al.* 2009), Elastic Net (Zou and Hastie 2005), Support Vector Regression with a linear kernel (SVR_{lin})

doi: https://doi.org/10.1534/g3.119.400498
Manuscript received July 3, 2019; accepted for publication September 9, 2019; published Early Online September 18, 2019.

Supplemental material available at FigShare: https://doi.org/10.25387/g3.9855590.
[1]Corresponding authors: Michigan State University, 775 Woodlot Dr., Office 1311, East Lansing, MI 48824-1312. E-mail: gustavoc@msu.edu. Shin-Han Shiu, Michigan State University, Plant Biology Laboratories, 612 Wilson Road, Room 166, East Lansing, MI 48824-1312. E-mail: shius@msu.edu.

(Moser *et al.* 2009; Xu *et al.* 2018), and additional Bayesian methods including Bayesian LASSO (BL), BayesCπ, and BayesDπ (de los Campos *et al.* 2009; Habier *et al.* 2011).

While these approaches perform well when dealing with high dimensional data (*i.e.*, $P >> n$), they are all based on a linear mapping from genotype to phenotypes, and therefore may not fully capture non-linear effects (*e.g.*, epistasis, dominance), which are likely to be important for complex traits (Holland 2007; Monir and Zhu 2018). To overcome this limitation, non-linear approaches, including reproducing kernel Hilbert spaces (RKHS) regression (Gianola *et al.* 2006; de los Campos *et al.* 2010), Support Vector Regression with non-linear kernels (*i.e.*, polynomial SVR_{poly} and radial basis function SVR_{rbf} (Long *et al.* 2011; Kasnavi *et al.* 2017)), and decision tree based algorithms such as Random Forest (RF) (González-Recio and Forni 2011; Spindel *et al.* 2015) and Gradient Tree Boosting (GTB) (González-Recio *et al.* 2013) have been applied to GP problems. In previous efforts to compare the performance of multiple linear and non-linear approaches (Heslot *et al.* 2012; Neves *et al.* 2012; Blondel *et al.* 2015; Ramstein *et al.* 2016; Roorkiwal *et al.* 2016), no single method performs best in all cases. Rather, factors such as the size of the training data set, marker type and number, trait heritability, effective population size, the number of causal loci, as well as genetic architecture (the locus effect size distribution) can all affect algorithm performance (Meuwissen 2009; Riedelsheimer *et al.* 2013; Spindel *et al.* 2015; Norman *et al.* 2018). This highlights the importance of comparing new algorithms across a diverse range of datasets.

With improvements in computing speeds, the development of graphics processing units (GPUs), and breakthroughs in algorithms for backpropagation learning (Rumelhart *et al.* 1986; Parker 1987), there has been a resurgence of research using deep learning (*i.e.*, artificial neural networks (ANNs)) to model complex biological processes (Angermueller *et al.* 2016; Webb 2018). ANNs are a class of machine learning methods that perform layers of transformations on features to create abstraction features, known as hidden layers, which are used for predictions. The first application of ANNs for GP was presented in 2011, when Okut *et al.* trained fully connected ANNs (*i.e.*, each node in a layer is connected to all nodes in surrounding layers) containing one hidden layer to predict body mass index in mice (Okut *et al.* 2011). Since 2011, more complex ANN architectures have been used for GP including radial basis function neural networks (González-Camacho *et al.* 2012) deep neural networks (Ehret *et al.* 2015; Bellot *et al.* 2018), deep recurrent neural networks (Pouladi *et al.* 2015), probabilistic neural network classifiers (González-Camacho *et al.* 2016, 2018), and convolutional neural networks (CNNs) (Ma *et al.* 2018). With only one exception (Bellot *et al.* 2018), these ANNs have been applied to datasets with relatively few genetic markers (<60k), however, as sequencing continues to become less expensive, whole-genome marker datasets are becoming larger with some breeding programs generating data for hundreds of thousands of markers. Because of the internal complexity of ANN models, training an ANN with so many markers can result in sub-optimal solutions (*i.e.*, underfitting). Therefore, it is especially important to benchmark ANNs against other GP statistical approaches on datasets with high dimensionality where underfitting may occur.

GP has yielded promising results for breeders. However, a comprehensive comparison of GP algorithms, particularly ANNs, on a wide range of GP problems is missing (Figure 1A). Here we compared the ability of 12 GP algorithms (see **Methods**, Figure 1B) to predict a diverse range of physiological traits in six plant species (maize, rice, sorghum, soy, spruce, and switchgrass; Figure 1C). These six data sets (referred to as the benchmark data sets) represent a wide range of GP data types, with the size of the training data set ranging from 327 to 5,014 individuals, and 4,000 to 332,000 markers derived from array-based approaches or sequencing. Compared to the linear algorithms included in the study, the non-linear algorithms, especially ANNs, require more pre-modeling tuning (*e.g.*, hyperparameter selection, feature selection). Therefore, before comparing algorithm performance across all 18 combinations of species and traits, we first focused on predicting plant height in each species in order to establish best practices for model building. Because ANNs are underrepresented in GP comparison studies and our first attempts to use ANNs for GP performed relatively poorly, we focus on methods to improve ANN performance, including reducing model complexity using feature selection and combining relationships learned from linear algorithms into the more complex ANN architectures (*i.e.*, a seeded ANN approach and convolutional layers (*i.e.*, CNNs)). Then, using lessons learned from predicting height, we compared the performance of all GP algorithms across all species and traits.

MATERIALS AND METHODS

Genotype and phenotype data

Genotypic data from six plant species were used to predict 3 traits from each species (Figure 1C). The maize phenotypic (Hansey *et al.* 2011) and genotypic (Hirsch *et al.* 2014) data were from the pan-genome population, maize trait values were averaged over replicate plots. The rice data were from elite breeding lines from the International Rice Research Institute irrigated rice breeding program (Spindel *et al.* 2015), and dry season trait data averaged over four years were used. The sorghum data were generated from sorghum lines from the US National Plant Germplasm System grown in Urbana, IL (Fernandes *et al.* 2017) and trait values were averaged over two blocks for this study. The soybean data were generated from the SoyNAM population containing recombinant inbred lines (RILs) derived from 40 biparental populations (Xavier *et al.* 2016). The white spruce data were obtained from the SmartForests project team, using a SNP-chip developed by Quebec Ministry of Forest Wildlife and Parks (Beaulieu *et al.* 2014). Switchgrass phenotypic (Lipka *et al.* 2014) and genotypic (Evans *et al.* 2018) data were generated from the Northern Switchgrass Association Panel (Evans *et al.* 2015) which contains clones or genotypes from 66 diverse upland switchgrass populations.

The genotype data were obtained in the form of biallelic SNPs with missing marker data already dropped or imputed by the original authors. Marker calls were converted when necessary to [-1,0,1] corresponding to [aa, Aa, AA] where A was either the reference or the most common allele. Genome locations of maize SNPs were converted from assembly AGPv2 to AGPv4, with AGPv2 SNPs that did not map to AGPv4 being removed, leaving 332,178 markers for the maize analysis. Phenotype values were normalized between 0 and 1. Lines with missing phenotypic value for any of the three traits were removed.

Genomic selection algorithms

To assess what statistical approaches are most frequently used for genomic selection, we conducted a literature search of papers applying genomic selection methods to crop or simulated data from January 2012-February 2018. We recorded what statistical approach(es) was(were) applied in each study (Table S1), allowing us to calculate both the total number of times an approach had been applied and how many times any two approaches were directly compared (Figure 1A). Based on the results from this literature search, nine commonly used statistical approaches were included in this study: rrBLUP, Bayes A (BA), Bayes B (BB), Bayesian LASSO, Bayesian-RR, RF, SVR with a

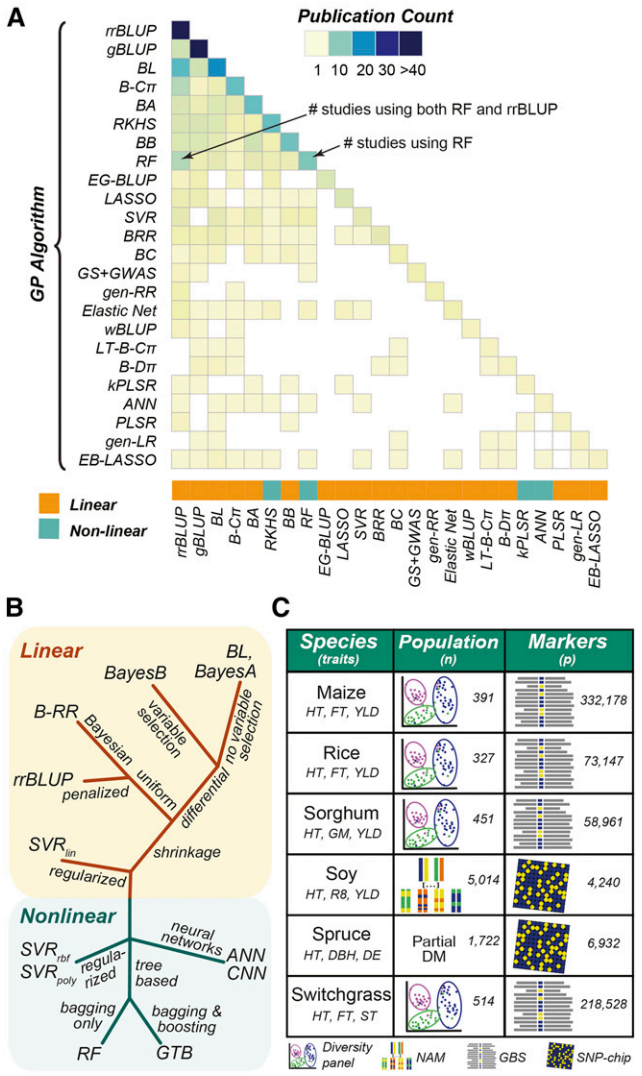

Figure 1 Algorithms used and compared in past GP studies and algorithms and data included in the GP benchmark. (A) Number of times a GP algorithm was utilized (diagonal) or directly compared to other GP algorithms (lower triangle) out of 91 publications published between 2012-2018 (Table S1). GP algorithms were included if they were utilized in >1 study. (B) A graphical representation of the GP algorithms included in the study and their relationship to each other. Colors designate if the algorithm identifies only linear (orange) or linear and non-linear (green) relationships. The placement of each algorithm on the tree designates (qualitatively) the relationship between different algorithms. The labels at each branch provide more information about how algorithms in that branch differ from others. rrBLUP, ridge regression Best Linear Unbiased Predictor; BRR, Bayesian Ridge Regression; BA, BayesA; BB, BayesB; BL, Bayesian LASSO; SVR, Support Vector Regression (kernel type: lin, linear; poly, polynomial; rbf, radial basis function); RF, Random Forest; GTB, Gradient Tree Boosting; ANN, Artificial Neural Network; CNN, Convolutional Neural Network. (C) Species and traits included in the benchmark with training population types and sizes and marker types and numbers for each dataset. NAM: Nested Association Mapping. DM: partial diallel mating. GBS: genotyping by sequencing. SNP: single nucleotide polymorphism. HT: height. FT: flowering time. YLD: yield. GM: grain moisture. R8: time to R8 developmental stage. DBH: diameter at breast height. DE: wood density. ST: standability.

linear kernel (SVR_{lin}), SVR with polynomial kernel (SVR_{poly}), SVR with radial basis function kernel (SVR_{rbf}). Three additional machine learning approaches, gradient tree boosting (GTB), artificial neural networks (ANN), and convolutional neural networks (CNN), were also included because of their ability to model non-linear relationships.

Most linear algorithms were implemented in R packages rrBLUP (Endelman 2011) and BGLR (for Bayesian methods including BRR: Bayesian RR, BA: Bayes A, BB: Bayes B, and BL: Bayesian LASSO) (Pérez and de los Campos 2014). These algorithms vary in what approach they use to address the $P >> n$ problem (Figure 1B), for example rrBLUP performs uniform shrinkage on all marker coefficients to reduce variance of the estimator, while BB performs differential shrinkage of the marker coefficients and variable selection. The differences between these algorithms have been thoroughly reviewed previously (de los Campos *et al.* 2013). Models for Bayesian methods were trained for 12,000 iterations using a burn-in of 2,000.

Non-linear algorithms (SVR_{poly}, SVR_{rbf}, RF, and GTB) and SVR_{lin} were implemented in python using the Scikit-Learn library (Pedregosa *et al.* 2011). For SVR algorithms, the marker data are mapped into a new feature space using linear or non-linear kernels (*i.e.*, poly, rbf) and then linear regression within that feature space is performed with the goal of minimizing error outside of a margin of tolerated error. The RF algorithm works by averaging the predictions from a "forest" of bootstrapped regression trees, where each tree contains a random subset of the lines and of the markers (Breiman 2001). Related to RF, GTB algorithm uses the principle of boosting (Friedman 2001) to improve predictions from weak learners (*i.e.*, regression trees) by iteratively updating the learners to minimize a loss function, therefore generating better weak learners as training progresses.

Artificial Neural Networks (ANNs) were implemented in python using TensorFlow (Girija 2016). The input layer for the ANNs contained the genetic markers for an individual (x; Figure 1B), the nodes in the hidden layers were all fully connected to all nodes in the previous and following layers (*i.e.*, Multilayer Perceptron). A non-linear activation function (selected during the grid search, see below) was applied to each node in the input and hidden layers, except the last hidden layer, which was connected with a linear function to the output layer, the predicted trait value (y). To reduce the likelihood of vanishing gradients, when the error gradient, which controls the degree to which the weights are updated during each iteration of training, becomes so small the weights stop updating thus halting model training, in the ANN, the starting weights (w) were scaled relative to the number of input markers using the Xavier Initializer (Glorot and Bengio 2010). Weights were then optimized using the Adam Optimizer (Kingma and Ba 2014) with a learning rate selected by the grid search (described below). To determine the optimal stopping time for training (*i.e.*, number of epochs), an early stopping approach was used (Prechelt 1998), where the training set was further divided into training and validation, and early stopping occurred when the change in mean squared error (MSE) for the validation set was $< 0.1\%$ for 10 epochs using a 10 epoch burn-in. Occasionally, due to poor random initialization of weights, the early stopping criteria would be reached before the network started to converge and the resulting network would predict the same trait value for every line. When this was observed in the validation set the training process was repeated starting with new initialized weights.

Convolutional Neural Networks (CNNs) were implemented in Python 3.6 using Tensorflow 2.0. The input layer for the CNNs consisted of the genetic markers for an individual one-hot-encoded so that each possible allele at each locus was represented as present or absent. Because of the large size of the possible hyperparameter

space (Table S2), a randomized search (using RandomizedSearchCV from Scikit-Learn with 5 folds) was performed on rice for predicting height on one replicate, and the best combination of hyperparameters (lowest average mean squared error) from this one search was used for all other species, traits, and replicates. The input data first passed through a convolutional layer, followed by a maximum pooling layer, a dropout layer, a dense (*i.e.*, fully connected) layer, a batch normalization layer, and finally to the output layer containing one node with the predicted trait value. The EarlyStopping function in Keras (https://keras.io/callbacks/#earlystopping) was used to avoid overfitting (min_delta = 0, patience = 10). To reduce the time and memory requirements, CNN models were trained using a batch size = 100 and run for a maximum of 1,000 epochs. As with ANN models, if the early stopping criteria was reached before the network started to converge, the model would be re-run starting with new initialized weights.

To incorporate predictions from multiple algorithms into one summary prediction, an ensemble approach was used where the ensemble predicted trait value was the mean predicted trait value from 11 algorithms (EN_{11}: rrBLUP, BRR, BA, BB, BL, SVR, SVRpoly, SVRrbf, RF, GTB, ANN) or five algorithms (EN_5: rrBLUP, BL, SVRpoly, RF, ANN). The subset of five consisted of algorithms with differing statistical bases, where rrBLUP represented penalized methods, BL represented the Bayesian approaches, SVRpoly represented non-linear regularized functions, RF represented decision tree based methods, and ANN represented the deep learning approach. This ensemble predicted trait value was then compared to the true trait values to generate performance metrics. A Repeated Measures Analysis of variance (ANOVA) implemented in R was used to compare model performance, where performance of each model on each replicate test set were considered related.

Hyperparameter grid search using cross-validation

To obtain the best possible results from each algorithm, a grid search approach was used to determine the combination of hyperparameters that maximized performance for each trait/species combination. No hyperparameter needed to be defined for rrBLUP, BL, or BRR. For rrBLUP, the R package estimates the regularization and kernel parameters from the data. For BL or BRR, parameters for these Bayesian regression methods were also estimated from the data. Between one and five hyperparameters were tested for the remaining algorithms (Table S2).

To avoid biasing our hyperparameter selection, an 80/20 training/testing approach was used, where 20% of the lines were held out from each model as a testing set and the grid search was performed on the remaining 80% of training lines. For RF, SVR_{lin}, SVR_{poly}, SVR_{rbf}, and GTB algorithms, 10 replicates of the grid search were run using the GridSearchCV function from Scikit-Learn with fivefold cross validation. Ten replicates of the grid search were also run for ANN models, where for each replicate 80% of the training data were randomly selected for training the network with each combination of hyperparameters and the remaining 20% used to select the best combination. This whole process (train/test split, grid search) was replicated 10 times, with a different 20% of lines selected as the test set for each replicate. ANOVA implemented in R was used to determine which hyperparameters significantly impacted model performance for each species.

Assessing predictive performance

The predictive performance of the models was compared using two metrics. For the grid search analysis, the mean squared error (MSE) between the predicted ($\hat{Y}$) and the true (Y) trait value was used. For the model comparisons, Pearson correlation coefficient (r) between the predicted ($\hat{Y}$) and the true trait value (Y) was used as it is the standard metric for GP performance (Heffner *et al.* 2009; Heslot *et al.* 2012; Riedelsheimer *et al.* 2013). It was computed using the cor() function in R for rrBLUP and the Bayesian approaches or the numpy corrcoef() function in Python for the ML and ANN approaches. Only predicted trait values for lines from the test set were considered when calculating r. Summary performance metrics (% of best r, rank, variance) were calculated using the mean predictive performance (r) across all replicates for each GP algorithm for each species/trait combination.

Feature selection

The top 10, 50, 100, 250, 500, 1000, 2000, 4000, and 8000 markers were selected using three different feature selection algorithms: Random Forest (RF), Elastic Net (EN), and BayesA (BA). RF and EN feature selection were implemented in Scikit-Learn and BA was implemented in the BGLR package in R. The EN feature selection algorithm requires tuning of the hyperparameter that controls the ratio of the L1- and L2-penalties (*e.g.*, L1:L2 = 1:10 = 0.1). Because the L1 penalty function performs variable selection by shrinking some coefficients to zero, we started with an initial weight on the L1 penalty of 0.1 and then, if fewer than 8,000 markers remained after variable selection, we reduced it in steps of 0.02 until that criteria was met (a 4,000 marker threshold was used for spruce and soy, which only had 6,932 and 4,240 markers available, respectively).

To avoid bias during feature selection, the 80:20 training/testing approach described above was used, where feature selection was performed on the training data and the ultimate performance of models built using the selected markers was scored on the testing set. This was repeated for all 10 testing sets. A repeat measures ANOVA was conducted to compare feature selection algorithms, the number of features selected, and GP algorithms (*i.e.*, independent variables) on model performance (*i.e.*, dependent variable) where replicates were considered repeat measures as they used the same testing set. One-sided, paired Wilcoxon Signed-Rank tests were conducted to determine if model performance (*i.e.*, dependent variable) increased after feature selection (all *vs.* top 4,000 for soy and spruce, all *vs.* top 8,000 for other species) (*i.e.*, independent variable). Resulting p-values were corrected for multiple testing (q-value) (Benjamini and Hochberg 1995).

Initializing ANN starting weights seeded from other GP algorithms

In addition to building ANNs with randomly initialized starting weights, we tested the usefulness of seeding the starting weights with information from other GP algorithms (*i.e.*, rrBLUP, BB, BL, or RF). This is an ensemble-like approach in that it utilizes multiple algorithms to make a final prediction. Ensemble approaches often perform better than single algorithm approaches (Dietterich 2000). First, after the data were divided into training, validation, and testing sets and, for species with large p:n ratios (*i.e.*, maize, rice, sorghum, switchgrass) the top 8,000 markers were selected, we applied a GP algorithm (rrBLUP, BB, BL, or RF) to the training data. From that model we extracted the coefficients/importance scores assigned to each marker and used those as the starting weights for 25% of the nodes in the first hidden layer. We also tested seeding starting weights for 50% of the nodes to predict height in all 6 species but found this significantly increased the model error (MSE) on the validation set (ANOVA; p-value= 0.04), so only results from seeding 25% were included. Because we still needed

to reduce the likelihood of vanishing gradients, described above, we manually adjusted the scale of the coefficients/importance scores to match the distribution of the starting weights assigned the remaining 75% of the nodes in the first hidden layer by Xavier Initialization. Finally, to reduce bias in the ANN, random noise was introduced to the seeded nodes by multiplying each starting weight with a random number from a normal distribution with a mean =0 and the standard deviation equal to the standard deviation of weights from Xavier Initialization.

After the training data were used to determine these seeded starting weights, it was used to train the ANN model, the validation set was used to select the best set of hyperparameters and the early stopping point. Then the final trained model was applied to the testing set and performance metrics were calculated. A repeat measures ANOVA was conducted to test if the seeded or the unseeded ANN models (*i.e.*, independent variable) differed in the amount of variation (standard deviation) in model performance across replicates (*i.e.*, dependent variable), with each species acting as a repeat measurement.

Data availability

For reproducibility, all six datasets along with training/testing designations are available on Dryad (https://doi.org/10.5061/dryad.xksn02vb9) and scripts to run all of the algorithms included in this study on GitHub for future benchmarking. All code used in this study is available on GitHub (https://github.com/ShiuLab/Manuscript_Code/tree/master/2019_GP_Comparison). A README file is included, which provides detailed instructions on how to use the code to generate GP models. Supplemental material available at FigShare: https://doi.org/10.25387/g3.9855590.

RESULTS

Hyperparameter grid search is critical, particularly among non-linear algorithms

We selected six linear and five non-linear algorithms (note, CNNs are discussed separately) to compare their performance in GP problems (see **Methods**). While some model parameters can be estimated from the data (de los Campos *et al.* 2013), other parameters, referred to as hyperparameters, have to be user-defined (Chapelle *et al.* 2002; Kuhn and Johnson 2013). This was the case for eight of the algorithms in our study: BA, BB, SVR_{lin}, SVR_{poly}, SVR_{rbf}, RF, GTB, and ANN. For these algorithms we conducted a grid search to evaluate the prediction accuracy of models using every possible combination of hyperparameter values (for lists of hyperparameters, see Table S2). To produce unbiased estimates of prediction accuracy the grid search was performed within the training set so that no data from the testing set was used to select hyperparameter values. Then we used the best set of hyperparameters from the grid search to build models using genotype and phenotype data from six plant species. This allowed us to compare the predictive performance of all algorithms included in the benchmark datasets.

To determine which hyperparameters significantly impacted model performance, we tested for changes in model performance (mean squared error; MSE) across the hyperparameter space for each algorithm/species/trait combination using Analysis of Variance (ANOVA). The degrees of freedom hyperparameter for BA and BB, both linear algorithms, that influences the shape of the prior density of marker effects (de los Campos *et al.* 2013) had no significant impact on model performance (ANOVA: p-value= 0.41∼1.0; Table S3). Other parameters for the Bayesian algorithms were determined using rules built into the BGLR package that account for factors such as phenotypic variance and the number of markers (p) (Pérez and de los Campos 2014) and were therefore not considered in our grid search. However, 15 of 16 of the hyperparameters tested for the non-linear algorithms significantly impacted performance in at least one species (Table S3, Figure S1A-C). Using height in maize as an example, we found that SVR_{poly} algorithm performed better (*i.e.*, lower MSE) using 2nd degree polynomials compared to using up to 3rd degree polynomials (p-value = $1^{*}10^{-21}$, Figure 2A). For RF-based models, the maximum depth (max depth) of decision trees allowed significantly impacted performance (p-value = $1^{*}10^{-3}$, Table S3), with shallower trees typically performing better (Figure 2B). This pattern was also observed in RF models predicting height for rice, spruce, and soy (p-value= $1^{*}10^{-66}$∼$5^{*}10^{-4}$, Table S3, Figure S1B). Because shallower decision trees are less complex, they tend not to overfit, suggesting the best hyperparameters for RF are those that reduce overfitting. The only hyperparameter from the non-linear algorithms that did not impact performance was the rate of dropout (a useful regularization technique to avoid overfitting) for ANN models, where there was no significant change in model performance when two different rates (10% and 50%) were used (p-value= 0.24 ∼0.97, Table S3).

ANN is the most significantly impacted by hyperparameter choice

Hyperparameters for SVR_{lin}, SVR_{poly}, SVR_{rbf}, RF, and GTB tended to have moderate effects on MSE, while ANN hyperparameters often caused substantial changes in MSE (Figure 2A-C; Figure S1A-C). Across the six species, the median variance in MSE across the hyperparameter space for ANN was $6^{*}10^{6}$, but ranged from $3^{*}10^{-3}$- 0.1 for the other GP algorithms (Figure S1D) For example, for predicting height in maize, SVR_{poly} models built using the 2nd degree polynomial outperformed those built using the 3rd degree polynomial with a decrease in MSE ∼0.05 (Figure 2A), while for ANN models, hyperparameter combinations that performed the best (*i.e.*, Sigmoid activation function and no L2 regularization) resulted in models with MSEs that were >500 lower than the worst performing model (Rectified Linear Unit (ReLU) activation function, no L2 regularization, and large numbers of hidden nodes; Figure 2C). This highlighted that, while hyperparameter selection is necessary for all non-linear algorithms, it is especially critical for building ANNs for GP problems.

Using the best set of hyperparameters for each model, we next compared the predictive performance (Pearson's correlation coefficient, r, between predicted and true trait values) of each algorithm on plant height. As with past efforts to benchmark GP algorithms (Heslot *et al.* 2012; Neves *et al.* 2012), no one algorithm always performed the best (white bolded; Figure 2D). For example, while rrBLUP performed best for maize, sorghum, and switchgrass, BA performed best for soy, and RF performed best for rice and spruce. Notably, ANNs substantially underperformed compared to other non-linear algorithms, with a median performance at 84% of the best r for each of the six species (*i.e.*, 16% below the best performing algorithm for that trait/species).

Notably, among the six species, ANN performed the best in soy (r = 0.44) relative to the species best algorithm BA (r = 0.47, Figure 2D). Soy has the largest number of training lines among the six species (5,014) and has a marker to training line ratio close to one (Figure 1C). Thus, we hypothesized the poor performance of the ANN models was in part due to our inability to train a network with so many features (markers) and so little training data (lines). During ANN model training, the weights assigned to each connection between nodes in neighboring layers of the network have to be estimated. Because

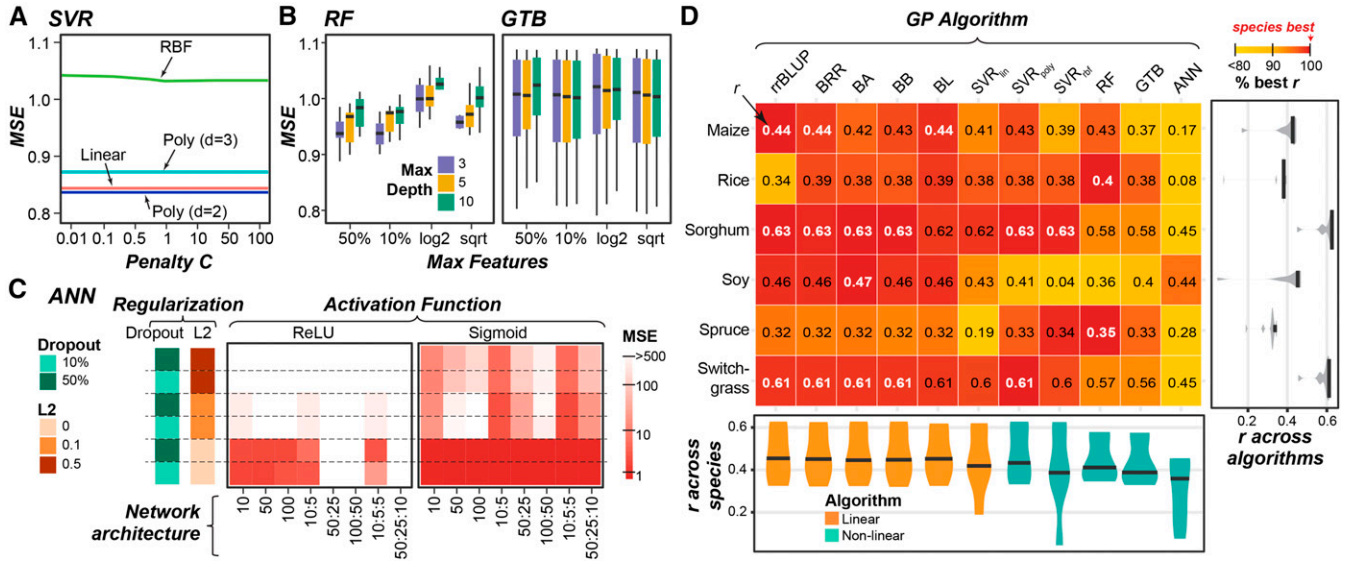

Figure 2 Grid search results for height in maize and overall GP algorithm performance for predicting height across species. (A) Average of mean squared error (MSE) over hyperparameter space (penalty, *C*) for Support Vector Regression (SVR) based models predicting height in maize. SVR_{rbf} and SVR_{poly} results are shown using gamma = $1x10^{-5}$ and $1x10^{-4}$, respectively. Poly: polynomial. RBF: Radial Basis Function. (B) Distribution of MSEs across hyperparameter space for Random Forest (RF; left) and Gradient Tree Boosting (GTB; right) as the maximum features available to each tree (Max Features) and maximum tree depth (color) change. GTB results are shown using a learning rate = 0.01. (C) Average MSE across hyperparameter space for ANN models with different network architectures, degrees of regularization (dropout or L2), using either the Rectified Linear Unit (ReLU; left) or Sigmoid (right) activation function. (D) Mean performance (Pearson's Correlation Coefficient: *r*, text) for predicting height and percent best *r* (colored box, top algorithm for each species = 100% (red)). White text: the best *r* values. Violin-plots show the median and distribution of *r* values for each trait (right) and algorithm (bottom).

every input marker is connected to every node in the first hidden layer, including more markers in the model will require more weights to be estimated, resulting in a more complex network that is more likely to underfit. In an ideal situation, to account for the complexity in these large networks, five to ten times more instances (lines) than features (markers) would need to be available for training (Klimasauskas 1993). Alternatively, one can reduce model complexity by only including markers that are most likely to be associated with the trait using feature selection methods.

Feature selection improves performance of ANN models

ANNs and sometimes other non-linear algorithms performed poorly compared to linear methods, which could be due to an insufficient number of training lines relative to the number of markers. To address this, we used feature selection to identify and select the markers most associated with trait variation. Because the number of markers associated with a trait is dependent on the genetic architecture of the trait and is not typically known, models were built using a range of numbers of markers (P = 10~8,000) and were compared to models built using all available markers from each species. Because performing feature selection on the training and testing data can artificially inflate prediction accuracies (Bermingham *et al.* 2015), feature selection was conducted on the training set only. This was repeated 10 times, using a different subset of lines for testing for each replicate (**see Methods**).

Three feature selection algorithms (RF, BayesA, and Elastic Net (EN)) were compared to predict height in maize, the species with the largest number of markers (p) relative to training lines (n) (p:n = 850, Figure 1C). While each algorithm selected a largely different subset of markers (Figure 3A, Figure S2A), the degree of overlap was significantly greater than random expectation. To demonstrate this, we randomly selected three sets of 8,000 maize markers and counted how many markers were present in all three sets 10,000 times and found that the 99th percentile of overlap was equal to 10, however we observed an average of 220 overlapping markers across replicates using these three feature selection approaches. When the different feature selection subsets were used to predict height in maize, there was a significant interaction between the number of available markers (p) and the feature selection method (repeat measures ANOVA: p-value = $1.7*10^{-12}$). Exploring this interaction further, we found that, while feature selection algorithms performed similarly with large n, RF tended to perform the best when fewer markers were selected for GP (Figure 3B; Figure S2B) and was therefore used to test the impact of feature selection on predicting height in the other five species.

For species with a low p:n ratio (*i.e.*, soy and spruce), for all GP algorithms tested, as p increased the model performance tended to increase continuously (*e.g.*, all GP algorithms in sorghum) or, in some cases, the model performance reached a maximum (or a plateau) quickly (*e.g.*, in soy after 2,500 markers were used) (Figure 3C). For these species, there was no significant improvement in performance after feature selection (all *vs.* top 4,000) using any GP algorithm (one-sided, paired Wilcoxon Signed-Rank test: q-value = 0.98 ~0.99; Figure 3D). For example, ANNs built using all 6,932 spruce markers performed no better than those built using the top 4,000 markers (p-value= 0.98).

For species with a large p:n ratio (*i.e.*, maize, rice, sorghum, and switchgrass), a similar pattern was observed for rrBLUP, SVR_{lin}, and GTB, where performance increased or reached a plateau as p increased and no significant improvement in performance was found after feature selection (P = 8,000) (q-value = 0.28 ~0.99; Figure 3D). However, for these four species, feature selection improved the

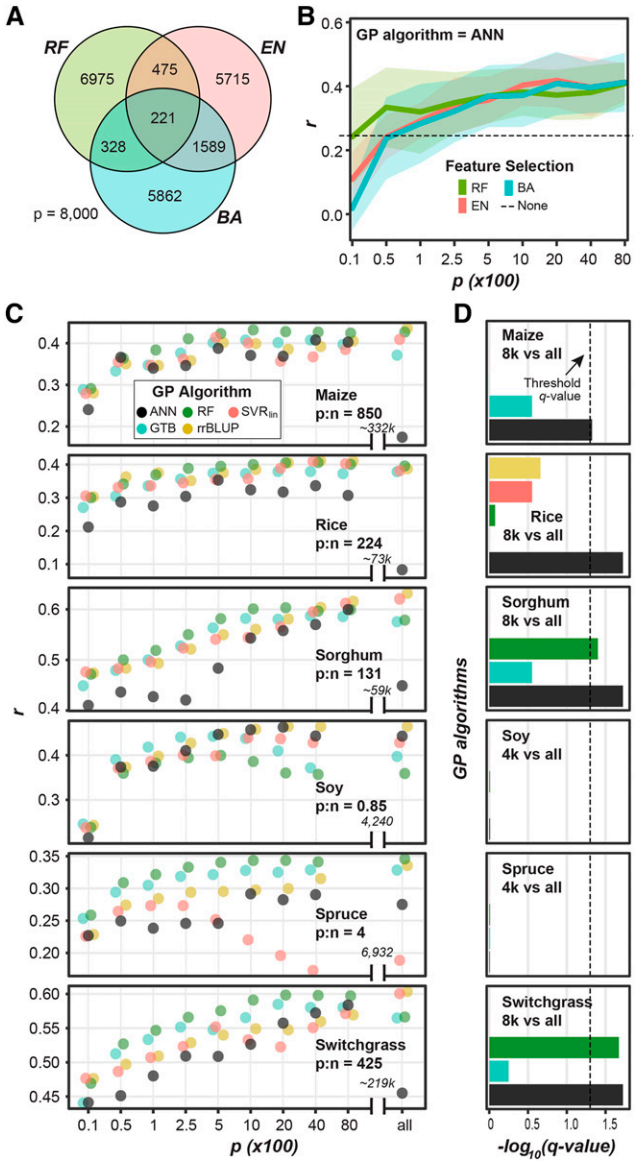

Figure 3 Impact of feature selection on GP algorithm performance. (A) Average number of overlapping markers in the top 8,000 markers selected by three feature selection algorithms for predicting height in maize across ten replicates. EN: Elastic Net. (B) Change in ANN predictive performance (*r*) at predicting height in maize as the number of input markers (p) selected by three feature selection algorithms (BayesA: BA, EN, and Random Forest: RF) increases. Dashed line: mean *r* when all 332,178 maize markers were used. (C) Mean *r* of rrBLUP, SVR_{lin}, RF, GTB, and ANN models for predicting height using subsets or all (X-axis) markers as features across 10 replicate feature selection and ML runs for each of six species with their ratios of numbers of markers (p) to numbers of lines (n) shown. Data points were jittered horizontally for ease of visualization. (D) The significance ($-\log_{10}$(*q*-value), paired Wilcoxon Signed-Rank test) of the difference in *r* between models from different GP algorithms (colored as in Figure 3C) generated using a subset of 4,000 or 8,000 and all markers as input. Dotted line designates significant differences (*p*-value < 0.05).

performance of ANN models (*q*-value= 0.019 ~0.047; Figure 3D). For example, after feature selection prediction of height in maize using ANNs improved from *r*=0.17 to 0.41, a 141% increase. Ultimately, performing feature selection prior to ANN training for these four datasets with large p:n ratios, improved ANN performance (median *r* at 89% of the best *r* for each of the six species) compared to ANNs without feature selection (84% of the best *r*). Therefore, for the GP benchmark analysis, feature selection was performed prior to model building for additional traits for maize, rice, sorghum, and switchgrass and the top 8,000 markers were used. Because feature selection only improved the performance of RF models in sorghum and switchgrass, we did not perform feature selection before training RF models in the full benchmark study.

While feature selection notably improved ANN performance, ANNs still often underperformed compared to other GP algorithms (Figure 3C), meaning the they were unable to learn even the linear relationships between markers and traits that were found using the linear-based algorithms. Because ANNs should theoretically at least match the performance of linear algorithms, this suggests that the ANN hyperparameters are not optimal. Furthermore, we found that, even after feature selection, there was greater variation in performance across replicates for ANN models compared to rrBLUP, SVR_{lin}, RF, and GB (Figure S2C-D), indicating the ANN models did not always converge on the best solution. One potential reason for the is that the final trained network can be heavily influenced by the initial weights used in ANN, which are selected randomly. In addition, while random weight initialization, a procedure we have used thus far, reduces bias in the network, it can also result in some networks converging on a local, rather than global, optimal solution.

Non-random initialization of ANN starting weights and convolutional layers improve ANN performance for some species

To reduce the likelihood of ANNs converging to locally optimal solutions, we developed an approach that allowed the ANNs to utilize the relationship between markers and traits determined by another GP algorithm. In this approach, a GP algorithm was applied to the training lines, and the coefficient or importance score assigned to each marker from this algorithm was used to seed the starting weights (Figure 4A). Four GP algorithms were tested to seed the weights: rrBLUP, BB, BL, and RF (referred to as ANN_{rrBLUP}, ANN_{BB}, ANN_{BL}, and ANN_{RF}, respectively). Because this approach could predispose the networks to only learn the relationship already identified by the seed algorithm, two steps were taken to re-introduce randomness into the network (see **Methods**). First, the seeded approach was only used to initialize starting weights for 25% of the nodes in the first hidden layer, while connection weights to the remaining 75% of nodes were initialized randomly as before. Second, noise was infused into the starting weights for the 25% of nodes that were seeded.

Applying this approach to predict plant height we found that ANN performance improved for three of six species (Figure 4B). For example, the average performance for rice without seeding (ANN) was $r = 0.25$ and with seeding from BL (ANN_{BL}) was $r = 0.32$, a 28% improvement, while for sorghum, ANN_{BL} had <0.1% improvement over the original ANN methods. Seeding ANN models did not significantly reduce the amount of variation in model performance across replicates (repeated measure ANOVA: *p*-value= 0.39, Table S4). Ultimately, seeded ANN models had a median performance between 89–90% of the best *r* for each species (compared to 89% with random initialization, Figure 4B). While this represented only a moderate improvement, we included the seeded ANN approach in the benchmark analysis because of how substantial the improvement was for some species (*i.e.*, rice).

Another deep learning strategy for reducing the complexity of GP problems and consequently decreasing the likelihood of converging on local optimum is to use convolutional and pooling layers to summarize

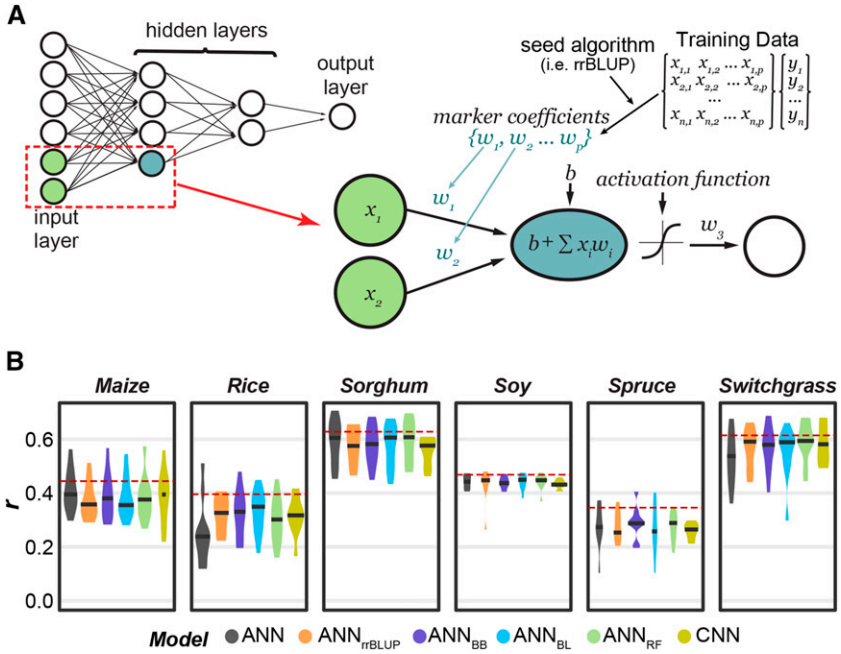

Figure 4 Description and performance results of the seeded ANN approach. (A) An overview of the seeded ANN approach. The network in the top left is an example of a fully connected ANN with 6 input nodes (*i.e.*, 6 markers), two hidden layers, and one output layer (*i.e.*, predicted trait value). The blue node in the first hidden layer represents an example node that will have seeded weights. For this node, the weights (*w*) connecting each input node to the hidden node will be seeded from the coefficient/importance for each marker as determined by another GP algorithm using the training data. *b*: bias, which helps control the value at which the activation function will trigger. (B) The distribution of model performance (*r*) using only all random (None) or 25% seeded (rrBLUP, BayesB, BL, RF) weight initialization, and convolutional neural networks (CNNs). The mean performance of the overall top performing algorithm (*i.e.*, not necessary ANN) shown as dotted red line.

local patterns of genetic markers and learn from these summaries (Ma *et al.* 2018). We tested this approach by training Convolutional Neural Networks (CNNs) to predict plant height (Figure S3A). Notably, feature selection (n= 8,000) had either no or a negative impact on CNN performance. For example, the average performance of CNNs at predicting height in maize, the species with the most genetic markers, was $r = 0.39$, but dropped to $r = 0.37$ after feature selection. CNNs performed better than ANNs at predicting height in two of six species (yellow; Figure 4B), with the biggest improvement in rice where the average performance increased from $r = 0.25$ using ANNs to $r = 0.32$ using CNNs, a 32% improvement. While CNN models did not reduce the amount of variation in model performance across replicates (repeated measure ANOVA: p-value = 0.08, Table S4), we included CNNs in the final benchmark analysis because of the promising results in rice and switchgrass.

No one GP algorithm performs best for all species and traits

Having established best practices for hyperparameter and feature selection for our datasets, we next compared the performance of all GP algorithms for predicting three traits in each of the six species. For maize, rice, and soy, these traits included height, flowering time, and yield (Figure 1C). For species where data were not available for one or more of these traits, other traits were used (see the panel labeled "Others", Figure 5A). As with past efforts to benchmark GP algorithms (Heslot *et al.* 2012; Neves *et al.* 2012), different algorithms performed best for different species/trait combinations (Figure 5A; Table S5). Thus, we utilized the predictive power of multiple algorithms to establish an ensemble prediction using all (except CNN: EN_{11}) or a subset of five (EN_5) algorithms (see **Methods**). The ensemble models consistently performed well, with EN_5 or EN_{11} being the best (three) or tied for the best (nine) algorithm for 12 of the 18 species/trait combinations included in the benchmark and had a median performance rank of 3 (Figure 5B; Table S6). For the remaining 6 species/trait combinations where EN_5 or EN_{11} weren't among the best performers, they tended to perform only slightly worse (median % of best $r = 99.2\%$, Figure 5A). This suggests that ensemble-based predictions are more stable and more likely to result in better trait predictions than a single algorithm.

Focusing on the species/trait combinations where one of the non-ensemble algorithms was or tied for best, we found that a linear algorithm performed best for five of the species/trait combinations, a non-linear algorithm performed best for four species/trait combinations, and both a linear and a non-linear algorithm performed equally well for the remaining six species/trait combinations (Figure 5B). This finding suggests that linear and non-linear algorithms are equally well suited for GP. The linear algorithms BRR and BA performed best overall, being among the top performers for 9 and 8 traits, respectively, and with the top two median ranks of five and 4.5, respectively (Table S6). The top performing non-linear algorithm was SVR_{poly}, which was among the top performers for 8 traits and had a median rank of 6. There was notably greater performance variation across species/traits for non-linear algorithms (mean variance = 1.03%) compared linear algorithms (mean variance = 0.65%) (Table S6). For example, SVR_{rbf} performed poorly at predicting developmental timing traits (median 83% of the best *r*), however it had or was tied for the best prediction for three of the four "other" traits (median 100% of the best *r*) (Figure 5A). Results from ANN models using randomly initialized (ANN) and BB seeded (ANN_{BB}) weights are shown because ANN_{BB} had the best performance of the seeded ANN models (see Table S5, S6 for results from other seeded ANNs). Notably, none of the randomly initialized ANN (median rank = 13.5), the ANN_{BB} (median rank = 13), or the CNN (median rank = 15.5) models performed best for any trait (Table S6).

One limitation of comparing the mean score or performance rank is that small but consistent differences in model performance could be missed. To account for this, we also calculated the number of times an algorithm outperformed another algorithm for each trait across the replicates. Using this metric, we were able to identify algorithms that consistently outperformed others for a given trait/species combination (Figure 5C, Figure S4). We frequently observed that linear algorithms had higher win percentages than nonlinear algorithms, this was the case for all three traits in maize and soybean for example (Figure S4). However, there were plenty of exceptions. RF and SVR_{rbf} had higher win

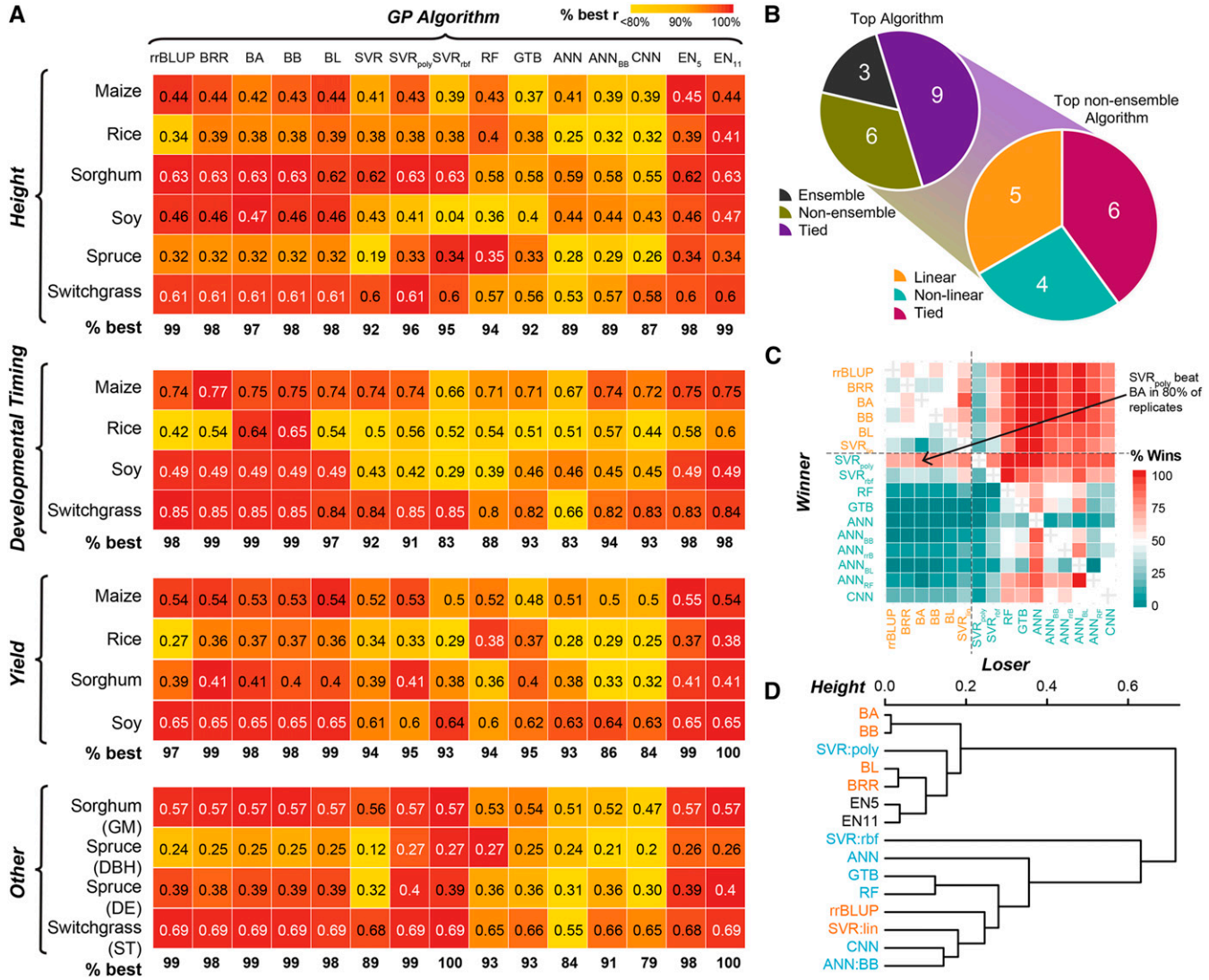

Figure 5 Comparison of algorithms for predicting additional traits. (A) Mean model performance (*r*; text) for each species/trait combination (y-axis) for each GP algorithm (x-axis). White text: *r* of the best performing algorithm(s) for a species. Colored boxes: percent of best performance (*r*) for a species, with the top algorithm for each species = 100% (red). The median % of best performance for each GP algorithm for each type of trait (*i.e.*, height, developmental timing, yield, other) is shown below each heatmap. GM: sorghum grain moisture. DBH and DE: diameter at breast height and wood density, respectively, for spruce. ST: standability for switchgrass. (B) Top left: summary of the number of species/trait combinations that were predicted best by an ensemble (gray) or a non-ensemble model (yellow), or predicted equally well by both (purple). Bottom right: among non-ensemble models that performed or tied for the best, the number of species/trait combinations that were predicted best by a linear (blue) or a non-linear model (green) or predicted equally well by both (orange). (C) Percent of replicates where one GP algorithm (y-axis, winner) outperformed another GP algorithm (x-axis, loser) for predicting height in switchgrass. Orange and cyan texts: linear and non-linear algorithms, respectively. (D) Hierarchical clustering of GP algorithms based on mean predictive performance across all species/trait combinations. Algorithm colored as in (C).

percentages than linear algorithms for predicting height and diameter at breast height (DBH) in spruce and ANN_{BB} had a higher win percentage than all algorithms except BA and BB for predicting flowering time in rice (Figure S3). In a few cases, assessing win percentages allowed us to identify winners when mean predictive performance (*r*) was tied. For example, for predicting height in switchgrass. SVR_{poly} had the same average performance ($r = 0.61$) as multiple of the linear algorithms (*i.e.*, rrBLUP, BA, etc.), however, it outperformed those algorithms in 70–80% of replicates (Figure 5C).

In order to determine which algorithms perform similarly, we performed hierarchical clustering of the algorithms based on their performance across the 18 species/trait combinations (from Figure 5A). Interestingly, linear and non-linear algorithms did not clearly separate from each other (Figure 5D). For example, rrBLUP and SVR_{lin} were more similar to the neural network based models (*i.e.*, CNN and ANN_{BB}), than they were to the linear Bayesian algorithms (*i.e.*, BA, BB, BL, and BRR). Notably, while the Bayesian algorithms tended to cluster together closely performance-wise, the non-linear algorithms tended to have a greater distance between them. Finally, in order to identify if algorithm performance was similar for specific types of traits (*e.g.*, whether similar algorithms perform well at predicting traits related to developmental timing) or across species/population

composition (*e.g.*, whether similar algorithms perform well on diversity panels), we performed hierarchical clustering of each species/trait based on performance of all 14 algorithms (from Figure 5A). Surprisingly, species/trait combinations with similar patterns of algorithm performance were often not the same species, trait, or population type (Figure S5), suggesting that we cannot generalize easily the differences in performance based on species, trait, or population type.

DISCUSSION

We conducted a benchmarking comparison of GP algorithms on 18 species/trait combinations that differ in the type and size of the training data set and of the marker data available. Similar to previous GP algorithm benchmark studies conducted on smaller datasets (Heslot *et al.* 2012; Blondel *et al.* 2015), a key result from this analysis is that no one model performs best for all species and all traits. We further demonstrate that, while similar algorithms perform similarly across the 18 species/trait combinations, algorithm performance was not clearly related to the trait type or population composition. With that said, linear algorithms tend to perform consistently well, while the performance of non-linear algorithms varied widely by trait. Studies of gene networks have shown that non-additive interactions (*e.g.*, epistasis, dominance) are important for development and regulation of complex traits (Holland 2007; Monir and Zhu 2018). One may expect approaches that can consider non-linear combinations would therefore be better suited for modeling complex trait. This was not the case and we found the inconsistency of non-linear algorithms surprising.

We have three, non-mutually exclusive, explanations for why linear algorithms often outperform non-linear algorithms. First, the traits included in this study vary in their genetic architecture (*i.e.*, the number and distribution of allele effects), therefore we may be observing that linear algorithms outperform non-linear algorithms when the trait has a predominantly additive genetic basis. Second, there is evidence that even highly complex biological systems generate allelic patterns that are consistent with a linear, additive genetic model because of the discrete nature of DNA variation and the fact that many markers have extreme allele frequencies (Hill *et al.* 2008). The proportion of dominance and epistatic variance that can be captured by an additive (*i.e.*, linear) model increases when allele frequencies are extreme (Hill *et al.* 2008). This phenomenon is even more important with inbred lines (*e.g.*, soy and rice); where, at each locus there are only 2 possible variants (*e.g.*, AA and TT); thus, the additive model fully captures the single-locus genetic variance. However, the fraction of epistatic variance that can be captured by an additive model depends on how many multi-locus genotypes are present in the data and this depends on allele frequencies. Thus, the distribution of allele frequency (which due to mutation, selection, and drift is often enriched at extreme values) is one of the reasons why additive models often capture and perform very well at predicting traits that at the biological level are affected by complex epistatic networks. Finally, a third explanation is that the amount of training data available for most GP problems was insufficient for learning non-linear interactions between large numbers of markers, therefore the linear models, which focus on modeling linear relationships, outperform the non-linear models.

Three findings from our study suggest that limited training data plays a role. First, we found that non-linear algorithms performed better at predicting traits in species with a small marker number to population size (p:n) ratio. For example, RF, SVR_{poly}, and SVR_{rbf} performed best at predicting traits in spruce and ANN models tended to perform better at predicting traits in soy, the species with the second smallest and smallest p:n, respectively. Second, the ANN models significantly improved after feature selection. This was not the case for other algorithms in our study or with previous efforts to use feature selection for GP (Vazquez *et al.* 2010; Bermingham *et al.* 2015). For example, for predicting traits in Holstein cattle, the top 2,000 markers had only 95% of the predictive ability of all the markers using BL (Vazquez *et al.* 2010). With a fixed training data size, prediction accuracy is a function of how much genetic variation is captured by markers in linkage disequilibrium with quantitative trait loci and the accuracy of the estimated effects (Goddard 2009). Because feature selection removes markers from the model, such decreases in performance after feature selection for non-ANN models are likely due to the reduction in the amount of genetic variation captured without a subsequent increase in the accuracy of the estimated effects. However, we hypothesize that feature selection significantly improved performance for ANNs because it improved the accuracy of the estimated effects (*i.e.*, the connection weights) more than it reduced the amount of genetic variation captured. Third, ANNs that have been trained on small datasets often have unstable performance likely because ANNs are sensitive to the initialized weight values when they do not have enough training data to learn from (LeBaron and Weigend 1998; Shaikhina and Khovanova 2017). We observed greater instability in performance across replicates for ANNs compared to other algorithms (Figure S2C-D), suggesting that our ANN models may have benefitted from additional training data.

However, a recent study involving large sample size (n∼80,000) in humans compared linear models with two types of ANN algorithms, multilayer perceptron and convolutional neural networks, and did not find any clear superiority of the ANN methods relative to linear models, if anything the linear model offered higher predictive power than the ANNs (Bellot *et al.* 2018). While they also found that feature selection improved the performance of their ANN models, using the top 10k of the 50k markers, these models still did not outperform the linear models (Bellot *et al.* 2018). Given that these results are from a single study in humans, we believe it will be informative to benchmark ANNs on a larger crop dataset in the future.

While there is a great deal of excitement about the uses of deep learning in the field of genetics, there is still much work to be done to improve performance of deep learning-based models. In this study we identified dimensionality as a major limitation to training ANNs for GP. Additional areas of deep learning research also need to be further explored. For example, in this study we limited the ANN hyperparameter space searched because the grid search method was too computationally intensive to be more thorough. Because changes in hyperparameters had a large impact on model performance, further hyperparameter tuning could lead to better performing models. For example, we limited our search to include nine possible network architectures with between one and three hidden layers each containing between 5-100 nodes (Table S2), but it is possible that ANNs with different network architectures, such as more hidden layers, or different combinations of layer sizes, could have performed better. Similarly, given that the hyperparameter space for CNN models was only tested for one species and trait (height in rice), it is likely that model-specific hyperparameter selection could improve the performance of CNN models beyond what we were able to achieve here.

In summary, we provided a thorough comparison of 12 GP algorithms and two ensembles for predicting diverse traits in six plant species with a range of marker types and numbers and population types and sizes. We found that no GP algorithm was best for all species/trait combinations and that trait type or population type were not closely associated with which algorithms worked best. While neural network approaches did not tend to outperform linear or other non-linear

models, strategies to tailor neural networks for GP problems (*e.g.*, non-random initialization of stating weights, convolutional and pooling layers) show promise. Unlike previous GP algorithm benchmark studies (Heslot *et al.* 2012), we found that the performance of ensemble models, generated by combining predictions from multiple individual GP algorithms, consistently tied with or exceeded the performance of the best individual algorithm. Taken together, these finds lead us to recommend that breeders test the performance of multiple algorithms on their training population to identify which algorithm or combination of algorithms performs best for traits important to their breeding program.

ACKNOWLEDGMENTS

We thank Peipei Wang and John Lloyd from the Shiu lab, Gabriel Rovere from the MSU QuantGen group, and Fouad Bahrpeyma from the Insight Center for their valuable suggestions to our project. This work was supported by the National Science Foundation (NSF) Graduate Research Fellowship [Fellow ID: 2015196719], Graduate Research Opportunities Abroad (GROW) Fellowship to C.B.A.; NSF PlantGenomics Research Experiences for Undergraduate to E.B.; the U.S. Department of Energy Great Lakes Bioenergy Research Center [BER DE-SC0018409] and National Science Foundation [IOS-1546617, DEB-1655386] to S.-H.S.; and the National Institutes of Health [R01GM099992, R01FM101219] to G.D.L.C.

LITERATURE CITED

Angermueller, C., T. Pärnamaa, L. Parts, and O. Stegle, 2016 Deep learning for computational biology. Mol. Syst. Biol. 12: 878. https://doi.org/10.15252/msb.20156651

Beaulieu, J., T. K. Doerksen, J. MacKay, A. Rainville, and J. Bousquet, 2014 Genomic selection accuracies within and between environments and small breeding groups in white spruce. BMC Genomics 15: 1048. https://doi.org/10.1186/1471-2164-15-1048

Bellot, P., G. de los Campos, and M. Pérez-Enciso, 2018 Can Deep Learning Improve Genomic Prediction of Complex Human Traits? Genetics genetics.2018.

Benjamini, Y., and Y. Hochberg, 1995 Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. J R Stat Soc Ser. B Stat Methodol 57: 289–300.

Bermingham, M. L., R. Pong-Wong, A. Spiliopoulou, C. Hayward, I. Rudan *et al.*, 2015 Application of high-dimensional feature selection: evaluation for genomic prediction in man. Sci. Rep. 1–12. https://doi.org/10.1038/srep10312

Blondel, M., A. Onogi, H. Iwata, and N. Ueda, 2015 A Ranking Approach to Genomic Selection. PLoS One 10: e0128570. https://doi.org/10.1371/journal.pone.0128570

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

de los Campos, G., D. Gianola, G. J. 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. https://doi.org/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. https://doi.org/10.1534/genetics.112.143313

de los Campos, G., H. Naya, D. Gianola, J. Crossa, A. Legarra *et al.*, 2009 Predicting quantitative traits with regression models for dense molecular markers and pedigree. Genetics 182: 375–385. https://doi.org/10.1534/genetics.109.101501

Chapelle, O., V. Vapnik, O. Bousquet, and S. Mukherjee, 2002 Choosing Multiple Parameters for Support Vector Machines. Mach. Learn. 46: 131–159. https://doi.org/10.1023/A:1012450327387

Desta, Z. A., and R. Ortiz, 2014 Genomic selection: genome-wide prediction in plant improvement. Trends Plant Sci. 19: 592–601. https://doi.org/10.1016/j.tplants.2014.05.006

Dietterich, T. G., 2000 Ensemble methods in machine learning. Int. Workshop Mult. Classif. Syst. https://doi.org/10.1007/3-540-45014-9_1

Ehret, A., D. Hochstuhl, D. Gianola, and G. Thaller, 2015 Application of neural networks with back-propagation to genome-enabled prediction of complex traits in Holstein-Friesian and German Fleckvieh cattle. Genet. Sel. Evol. 47: 22. https://doi.org/10.1186/s12711-015-0097-5

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

Evans, J., E. Crisovan, K. Barry, C. Daum, J. Jenkins *et al.*, 2015 Diversity and population structure of northern switchgrass as revealed through exome capture sequencing. Plant J. 84: 800–815. https://doi.org/10.1111/tpj.13041

Evans, J., M. D. Sanciangco, K. H. Lau, E. Crisovan, K. Barry *et al.*, 2018 Extensive Genetic Diversity is Present within North American Switchgrass Germplasm. Plant Genome. 11 https://doi.org/10.3835/plantgenome2017.06.0055

Fernandes, S. B., K. O. G. Dias, D. F. Ferreira, and P. J. Brown, 2017 Efficiency of multi-trait, indirect, and trait-assisted genomic selection for improvement of biomass sorghum. Theor. Appl. Genet. 131: 747–755.

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

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

Girija, S. S., 2016 Tensorflow: Large-scale machine learning on heterogeneous distributed systems.

Glorot, X., and Y. Bengio Understanding the difficulty of training deep feedforward neural networks. 2010.

Goddard, M., 2009 Genomic selection: prediction of accuracy and maximisation of long term response. Genetica 136: 245–257. https://doi.org/10.1007/s10709-008-9308-0

González-Camacho, J. M., G. de los Campos, P. Pérez, D. Gianola, J. E. Cairns *et al.*, 2012 Genome-enabled prediction of genetic values using radial basis function neural networks. Theor. Appl. Genet. 125: 759–771. https://doi.org/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. https://doi.org/10.1186/s12864-016-2553-1

González-Camacho, J. M., L. Ornella, P. Pérez-Rodríguez, D. Gianola, S. Dreisigacker *et al.*, 2018 Applications of Machine Learning Methods to Genomic Selection in Breeding Wheat for Rust Resistance. Plant Genome 11: 170104. https://doi.org/10.3835/plantgenome2017.11.0104

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

González-Recio, O., J. A. Jiménez-Montero, and R. Alenda, 2013 The gradient boosting algorithm and random boosting for genome-assisted evaluation in large data sets. J. Dairy Sci. 96: 614–624. https://doi.org/10.3168/jds.2012-5630

Habier, D., R. L. Fernando, K. Kizilkaya, and D. J. Garrick, 2011 Extension of the bayesian alphabet for genomic selection. BMC Bioinformatics 12: 186. https://doi.org/10.1186/1471-2105-12-186

Hansey, C. N., J. M. Johnson, R. S. Sekhon, S. M. Kaeppler, and N. de Leon, 2011 Genetic diversity of a maize association population with restricted phenology. Crop Sci. 51: 704–715. https://doi.org/10.2135/cropsci2010.03.0178

Heffner, E. L., M. E. Sorrells, and J.-L. Jannink, 2009 Genomic Selection for Crop Improvement. Crop Sci. 49: 1–12. https://doi.org/10.2135/cropsci2008.08.0512

Heslot, N., H.-P. Yang, M. E. Sorrells, and J.-L. Jannink, 2012 Genomic Selection in Plant Breeding: A Comparison of Models. Crop Sci. 52: 146–160. https://doi.org/10.2135/cropsci2011.06.0297

Hill, W. G., M. E. Goddard, and P. M. Visscher, 2008 Data and theory point to mainly additive genetic variance for complex traits. PLoS Genet. 4: e1000008. https://doi.org/10.1371/journal.pgen.1000008

Hirsch, C. N., J. M. Foerster, J. M. Johnson, R. S. Sekhon, G. Muttoni *et al.*, 2014 Insights into the maize pan-genome and pan-transcriptome. Plant Cell 26: 121–135. https://doi.org/10.1105/tpc.113.119982

Holland, J. B., 2007 Genetic architecture of complex traits in plants. Curr. Opin. Plant Biol. 10: 156–161. https://doi.org/10.1016/j.pbi.2007.01.003

Jonas, E., and D.-J. de Koning, 2013 Does genomic selection have a future in plant breeding? Trends Biotechnol. 31: 497–504. https://doi.org/10.1016/j.tibtech.2013.06.003

Kasnavi, S. A., M. A. Afshar, M. M. Shariati, N. E. J. Kashan, and M. Honarvar, 2017 Performance evaluation of support vector machine (SVM)-based predictors in genomic selection. Indian J. Anim. Sci. 87: 1226–1231.

Kingma, D. P., and J. Ba, 2014 Adam: A Method for Stochastic Optimization. arXiv. https://arxiv.org/abs/1412.6980

Klimasauskas, C. C., 1993 Applying Neural Networks In: Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real World Performance, pp. 64–65. R. R. Trippi and E. Turban, editors, Probus, Chicago. ISBN: 1557384525.

Kuhn, M., and K. Johnson, 2013 Over-Fitting and Model Tuning. Applied Predictive Modeling. Springer, New York, NY. 61–92. https://doi.org/10.1007/978-1-4614-6849-3_4

LeBaron, B., and A. S. Weigend, 1998 A bootstrap evaluation of the effect of data splitting on financial time series. IEEE Trans. Neural Netw. 9: 213–220. https://doi.org/10.1109/72.655043

Lipka, A. E., F. Lu, J. H. Cherney, E. S. Buckler, M. D. Casler *et al.*, 2014 Accelerating the Switchgrass (Panicum virgatum L.) Breeding Cycle Using Genomic Selection Approaches. PLoS One 9: e112227. https://doi.org/10.1371/journal.pone.0112227

Long, N., D. Gianola, G. J. M. Rosa, and K. A. Weigel, 2011 Application of support vector regression to genome-assisted prediction of quantitative traits. Theor. Appl. Genet. 123: 1065–1074. https://doi.org/10.1007/s00122-011-1648-y

Lorenz, A. J., S. Chao, F. G. Asoro, E. L. Heffner, T. Hayashi *et al.*, 2011 *Genomic Selection in Plant Breeding: Knowledge and Prospects*, chap 2. Elsevier, Amsterdam, Netherlands https://doi.org/10.1016/B978-0-12-385531-2.00002-5

Ma, W., Z. Qiu, J. Song, J. Li, Q. Cheng *et al.*, 2018 A deep convolutional neural network approach for predicting phenotypes from genotypes. Planta 248: 1307–1318. https://doi.org/10.1007/s00425-018-2976-9

Meuwissen, T. H. E., 2009 Accuracy of breeding values of'unrelated'individuals predicted by dense SNP genotyping. Genet. Sel. Evol. 41: 35. https://doi.org/10.1186/1297-9686-41-35

Meuwissen, T. H.., B. J. Hayes, and M. E. Goddard, 2001 Prediction of Total Genetic Value Using Genome-Wide Dense Marker Maps. Genetics 157: 1819–1829.

Monir, M. M., and J. Zhu, 2018 Dominance and Epistasis Interactions Revealed as Important Variants for Leaf Traits of Maize NAM Population. Front. Plant Sci. 9: 627. https://doi.org/10.3389/fpls.2018.00627

Moser, G., B. Tier, R. E. Crump, M. S. Khatkar, and H. W. Raadsma, 2009 A comparison of five methods to predict genomic breeding values of dairy bulls from genome-wide SNP markers. Genet. Sel. Evol. 41: 56. https://doi.org/10.1186/1297-9686-41-56

Neves, H. H., R. Carvalheiro, and S. A. Queiroz, 2012 A comparison of statistical methods for genomic selection in a mice population. 13: 100. https://doi.org/10.1186/1471-2156-13-100

Norman, A., J. Taylor, J. Edwards, and H. Kuchel, 2018 Optimising Genomic Selection in Wheat: Effect of Marker Density, Population Size and Population Structure on Prediction Accuracy. G3 (Bethesda) 8: 2889–2899. https://doi.org/10.1534/g3.118.200311

Okut, H., D. Gianola, G. J. M. Rosa, and K. A. Weigel, 2011 Prediction of body mass index in mice using dense molecular markers and a regularized neural network. Genet. Res. 93: 189–201. https://doi.org/10.1017/S0016672310000662

Parker, D. B., 1987 Optimal algorithms for adaptive networks: Second order backpropagation, second order direct backpropagation, and second order hebbing learning. Proceedings of the IEEE First International Conference on Neural Networks. 2: 593–600.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion *et al.*, 2011 Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. 12: 2825–2830.

Pérez, P., and G. de los Campos, 2014 Genome-wide regression and prediction with the BGLR statistical package. Genetics 198: 483–495. https://doi.org/10.1534/genetics.114.164442

Pouladi, F., H. Salehinejad, and A. M. Gilani, 2015 Deep Recurrent Neural Networks for Sequential Phenotype Prediction in Genomics. arXiv:1511.02554

Prechelt, L., 1998 Early Stopping - But When? pp. 55–69 in *Neural Networks: Tricks of the Trade*, edited by G. B. Orr and K.-R. Müller. Springer Berlin Heidelberg, Berlin, Heidelberg.

Ramstein, G. P., J. Evans, S. M. Kaeppler, R. B. Mitchell, K. P. Vogel *et al.*, 2016 Accuracy of Genomic Prediction in Switchgrass (Panicum virgatum L.) Improved by Accounting for Linkage Disequilibrium. G3 (Bethesda) 6: 1049–1062. https://doi.org/10.1534/g3.115.024950

Ribaut, J.-M., and M. Ragot, 2007 Marker-assisted selection to improve drought adaptation in maize: the backcross approach, perspectives, limitations, and alternatives. J. Exp. Bot. 58: 351–360. https://doi.org/10.1093/jxb/erl214

Riedelsheimer, C., F. Technow, and A. E. Melchinger, 2013 Comparison of whole-genome prediction models for traits with contrasting genetic architecture in a diversity panel of maize inbred lines. BMC Genomics 13: 452. https://doi.org/10.1186/1471-2164-13-452

Roorkiwal, M., A. Rathore, R. R. Das, M. K. Singh, A. Jain *et al.*, 2016 Genome-Enabled Prediction Models for Yield Related Traits in Chickpea. Front. Plant Sci. 7: 1666. https://doi.org/10.3389/fpls.2016.01666

Rumelhart, D. E., G. E. Hinton, and R. J. Williams, 1986 Learning internal representation by error propagation, Parallel Distributed Processing, DE Rumelhart and JL McClelland, eds. ISBN: 0-262-68053-X. 318–326.

Shaikhina, T., and N. A. Khovanova, 2017 Handling limited datasets with neural networks in medical applications: A small-data approach. Artif. Intell. Med. 75: 51–63. https://doi.org/10.1016/j.artmed.2016.12.003

Spindel, J., H. Begum, D. Akdemir, P. Virk, B. Collard *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 Genet. 11: e1004982. https://doi.org/10.1371/journal.pgen.1004982

Usai, M. G., M. E. Goddard, and B. J. Hayes, 2009 LASSO with cross-validation for genomic selection. Genet. Res. 91: 427–436. https://doi.org/10.1017/S0016672309990334

Vazquez, A. I., G. J. M. Rosa, K. A. Weigel, G. de los Campos, D. Gianola *et al.*, 2010 Predictive ability of subsets of single nucleotide polymorphisms with and without parent average in US Holsteins. J. Dairy Sci. 93: 5942–5949. https://doi.org/10.3168/jds.2010-3335

Webb, S., 2018 Deep learning for biology. Nature 554: 555–557. https://doi.org/10.1038/d41586-018-02174-z

Xavier, A., W. M. Muir, and K. M. Rainey, 2016 Assessing Predictive Properties of Genome-Wide Selection in Soybeans. G3 (Bethesda) 6: 2611–2616. https://doi.org/10.1534/g3.116.032268

Xu, Y., X. Wang, X. Ding, X. Zheng, Z. Yang *et al.*, 2018 Genomic selection of agronomic traits in hybrid rice using an NCII population. Rice (N. Y.) 11: 32. https://doi.org/10.1186/s12284-018-0223-4

Zou, H., and T. Hastie, 2005 Regularization and variable selection via the elastic net. J. R. Stat. Soc. Ser. 67: 301–320. http://www.jstor.org/stable/3647580

Communicating editor: J. Birchler