BMC Bioinformatics

RESEARCH

Open Access

Haplotype-based autoencoders can reduce the dataset dimension and estimate haplotype block effects in different crop species

Philipp Georg Heilmann[1*], Emanuel Grosch[1], Matthias Frisch[1], Matthias Herrmann[2], Steffen Beuch[3], Vivek Kurra[4], Martin Mascher[5], Raz Avni[5], Klaus Oldach[6], Ina Röhrs[7], Anja Hanemann[8], Raja Ram Mehta[4], Carsten Reinbrecht[9], Albrecht Serfling[10], Andreas Stahl[10], Marco Stucke[9], Amine Abbadi[11], Tobias Kox[11] and Carola Zenke-Philippi[1]

*Correspondence:
Philipp Georg Heilmann
philipp.g.heilmann@agrar.uni-giessen.de

Full list of author information is available at the end of the article

Abstract

Background In plant breeding, many studies currently investigate the application of machine learning (ML) to genomic prediction, hoping for an improvement in prediction accuracy compared to standard models like Genomic Best Linear Unbiased Prediction (GBLUP). However, ML algorithms require much higher computational resources. This study aims to reduce the computational requirements and speed up training time by developing a novel autoencoder architecture inspired by haplotype blocks. Our approach incorporates prior knowledge on genetic linkage, inspired by haplotype block building, into the autoencoder architecture, resulting in a new encoded variable per haplotype block. We further modified our model into a semi-supervised version by adding available yield information. We used features extracted from the autoencoder's block layer as inputs for Random Forest and GBLUP models to predict the yield of hybrid and inbred crops.

Results Genomic prediction based on the extracted features maintained prediction accuracies equal to using the original marker data, even with a variable reduction of up to 98% and significantly reduced computation time. Prediction accuracy of the supervised component was in some cases equal to and in some lower than the prediction accuracy achieved using GBLUP. Effects estimated for haplotype block variants using our new method showed a high correlation to the blockwise sum of marker effects, which is the current standard approach for haplotype block effects. Correlation between the two block effect estimation approaches was very low for some blocks, which might indicate the incorporation of non-linear effects by the autoencoder.

Conclusions Our approach introduces a new perspective on processing haplotype blocks for genomic prediction, potentially providing more flexible modelling opportunities without the use of multiple binary dummy variables for each block variant. Additionally, training time for ML models may be significantly reduced by using the reduced feature sets generated using our method. By adding the semi-supervised component, the model is able to estimate values similar to marker effects

for each block on yield. In future work, this may provide a new way of quantifying the importance of haplotype blocks for selection and breeding.

Keywords Plant breeding, Machine learning, Genomic prediction, Neural networks, Autoencoders, Dimensionality reduction, Haplotype blocks

Introduction

With the introduction of marker-based prediction methods [1–3] and the development of high-throughput genotyping, plant breeding has increasingly complemented phenotyping with genotyping, allowing the selection and evaluation of untested genotypes. Methods such as genomic best linear unbiased predictor (GBLUP) provide robust frameworks for predicting the phenotypic values of traits based on genetic data. As field trials are expensive, accurately identifying the most promising genotypes greatly increases their efficiency and maximises the utility gained from the resources available. Therefore, research in plant breeding is heavily focused on developing new methods with higher prediction accuracies. To this day however, the original GBLUP proves to still be competitive and remains a highly reliable standard method that has not yet been consistently outperformed by other methods [4, 5].

Recently, machine learning (ML) has attracted attention in the field of plant breeding. The term ML encompasses a variety of algorithms that typically learn iteratively from training data. These algorithms can theoretically approximate any underlying function that describes the relationship between predictor variables and target variables within a dataset. Due to this inherent ability to also model nonlinear interactions, researchers hoped to find algorithms that would outperform GBLUP (a linear model by design). However, despite initial optimism, no ML approach has yet been shown to outperform traditional methods consistently in genomic prediction. While many studies demonstrate cases in which ML outperforms older approaches, such as GBLUP or Ridge Regression BLUP (RR-BLUP), its success appears to be species-, trait- and dataset-specific [4–10]. As the application of ML in plant breeding is still in its initial stage, this does not mean that ML cannot outperform classical methods, but rather that we need to develop more specific models and architectures for plant breeding [11].

However, ML algorithms have drawbacks that make them harder to train compared to linear mixed models for genomic prediction. While GBLUP generally uses a $n \times n$ genomic relationship matrix based on single nucleotide polymorphism (SNP) markers, where n is the number of genotypes, ML methods typically use the raw marker data, resulting in datasets of much higher dimensionality. Combined with the necessity of training many models for hyperparameter tuning and internal cross-validation [12], ML models often require much more training time and strong computer hardware. This slows down the application of ML in scientific studies compared to straightforward linear models. To address these issues, an initial study of 361 German winter wheat genotypes [8] was conducted, where we investigated the use of different sets of input variables, e.g. haplotype blocks and autoencoder features, to improve the training efficiency and increase the prediction accuracy of ML algorithms.

Haplotype blocks are said to have the potential to reduce the dimensionality of the dataset [13] and capture local epistatic effects [14]. To use blocks as features in prediction, every unique combination of marker alleles within a block (i.e. each variant of a block) is encoded using variables that represent the number of copies of each block

variant present in the genotype. When we used haplotype-based blocks as input features, we found that the prediction accuracy was comparable to or slightly higher than that of the full set of SNP markers for all algorithms for all traits except one [8]. However, due to the presence of many block variants, the haplotype-based blocks increased the dimension of the dataset rather than decreasing it [15], which further increased computation time.

As a second set of features, the output of the encoding layer of an autoencoder was used. Autoencoders are unsupervised neural networks optimized to encode a set of input features into a lower dimensional space in such a way that it is possible to reconstruct the original data from the encoded state [16]. Although this approach significantly reduced computation time, prediction accuracy declined for all traits and algorithms, presumably because the learned features were less informative than the original SNP markers [8].

Another study [17] proposed an alternative approach using a series of autoencoders instead of one autoencoder with fully connected layers. With this method, a fixed number of adjacent markers were grouped together and used as inputs for small autoencoders, which were then combined in a second stage to generate lower-dimensional representations for genomic prediction. This way, the prediction accuracy remained stable while the data dimension was reduced. The approach of grouping together a fixed number of adjacent markers is somewhat reminiscent of a very similar approach for building haplotype-based blocks using fixed window sizes. However, studies show that a fixed window size approach is usually not the best way of building blocks as prediction accuracy based on those is usually lower compared to predictions using blocks based on linkage disequilibrium (LD) [15, 18].

In this study, we build upon the method proposed in [17] by incorporating prior knowledge of the genetic linkage structure into the autoencoder structure. Instead of grouping markers based on a fixed window size, we define blocks using pairwise LD. Markers within a block are connected locally within the autoencoder, resulting in an architecture that more accurately reflects the underlying genetic structure. This approach enables each haplotype block to be reduced to a single unit in the encoding layer, i.e. a single variable after extraction, leading to a guaranteed dimensionality reduction compared to other haplotype block-based approaches.

Haplotype stacking has been proposed as a possible application for haplotype blocks in breeding [19]. However, this requires the estimation of effects for each block variant. An estimation method for effects and variances of haplotype blocks was first proposed in Voss-Fels et al.[19]. Effects of block variants are defined as the sum of the individual marker effects that belong to the same block. This approach has been adopted in other studies [20, 21]. Adding an output node of a target trait to our autoencoder architecture enables the model to estimate block variant effects by quantifying the contribution of a variant to the trait as the product of the outputs of the block layer and their respective weights. This provides a novel approach to estimating block variant effects beyond summing up individual effects. Additionally, it solves the problem of handling unseen block variants, which frequently occur in breeding programmes as newly generated material or crosses often contain variants that did not exist in the initial population.

The overall goal of this study was to develop an autoencoder-based method that integrates LD-based haplotype blocks into its architecture. By incorporating genetic linkage

information into the model, we aimed to improve the efficiency of dimensionality reduction for genomic prediction. Specifically, we aimed to: (1) characterise the features generated by the autoencoder during unsupervised training and evaluate their usefulness for genomic prediction; and (2) explore the potential of semi-supervised learning as an alternative approach for estimating the effects of variants within haplotype blocks.

Materials and methods

Materials

We applied our methodology to five datasets in total. Two of these datasets consisted of inbred/double haploid lines while the remaining three consisted of hybrids. All of the datasets contained yield data and genetic markers. We filtered the data, removing genotypes with more than 60% missing markers. Furthermore, we removed markers consisting of more than two alleles, with 10% or more missing values, or with an expected heterozygosity below 5%. If the remaining markers contained missing values, we imputed them using BEAGLE [22].

Dataset **Ra1** was provided by Norddeutsche Pflanzenzucht Hans-Georg Lembke KG and consisted of 746 rapeseed hybrids. After filtering 10,939 markers remained. Dataset **Mz1** was published previously [23] and accessed through the R package 'sommer 4.2.0' [24, 25]. It consisted of 1,254 maize hybrids. Genetic data was provided for the parental lines and 28,803 markers remained after filtering. Dataset **Mz2** is based on the data provided for the 2022 maize prediction competition hosted by the 'genomes 2 fields' project [26] and publicly available via the projects' website [27]. After preprossessing, it consisted of 4,689 maize hybrids and 267,818 markers for their respective parents. Dataset **Wh1** consisted of wheat lines which were generated from a factorial crossing design within the project MultiResistGS. After preprocessing, 293 lines and 17,221 markers remained. Dataset **Ot1** consisted of oat lines collected during the project FUGE, also from factorial crosses. After preprocessing, 234 lines and 29,457 markers remained. More information on the datasets is provided in Supplementary Table S1 and S2, and Supplementary Fig. S2.

For every dataset, adjusted entry means were either provided for each genotype or were calculated using an appropriate mixed linear model, factoring in environments and local field design. The specific mixed linear model design depended on the dataset.

Feature engineering

The haplotype blocks used in this study were built using the LD between adjacent markers with a threshold of 0.7 required for a marker to be assigned to a block. Pairwise LD between all markers was based on the r^2 measure [28]. We set a tolerance threshold of 1 marker within a block to be below this threshold.

The autoencoder that was used in this study was based on the generated haplotype blocks, as inputs are only connected locally within the borders of these blocks. Figure 1 displays a schematic illustration of an example autoencoder with this architecture. The unsupervised part of the autoencoder was used for feature engineering. It consisted of 4 to 5 layers, depending on the block size. Unsupervised learning refers to models that process input features without regard to any target trait Y. The input layer consisted of as many units as there were markers assigned to blocks. Blocks that only consisted of a single marker were not considered blocks and the markers were subsequently removed

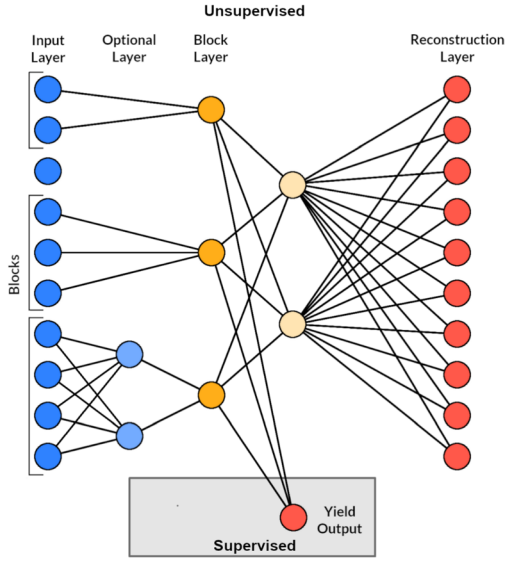

Fig. 1 Illustration of the autoencoder architecture

from the data input. The input layer was locally connected to either all units of an optional hidden layer or to the block layer. If a block contained ≥ 4 markers, an additional hidden layer was added between the input and the block layer to introduce additional non-linearity and modeling capabilities ("optional layer" in Fig. 1). The optional hidden layer consisted of $\lfloor \frac{n}{2} \rfloor$ units, where n was the size of the block. The output of those $\lfloor \frac{n}{2} \rfloor$ units was passed on to the block layer through a leaky ReLU activation function with a negative slope of 0.1. Local connections were limited to markers within the same block, resulting in one single unit per block in the block layer. The block layer was then fully connected to a hidden layer with 1000 units using the same leaky ReLU activation function as before. The hidden layer was connected to the reconstruction layer, which acted as the output of our unsupervised model. This layer returned the reconstructed marker data, which included markers that were previously removed due to a lack of block assignment. As the markers were encoded as -1, 0, and 1 to represent homozygosity for the minor allele, heterozygosity, and homozygosity for the major allele, respectively, the reconstruction layer used a tanh activation function that returned values in the range from -1 to 1.

The autoencoder was trained for 100 epochs. Model parameters were optimized using the Adam optimizer with a learning rate of 0.001 and the mean squared error (MSE) as the loss function. After training, the outputs of the block layer were extracted (first reduction step, AE1). As those were highly correlated, a second dimensionality reduction step (AE2) was added. The block layer output of the first autoencoder was used as the input for a new autoencoder with the same architecture which was then again trained for 100 epochs. Since it was not possible to form LD-based blocks in this scenario, blocks that showed a pairwise correlation >0.7 were grouped together to form 'meta blocks'. This is analogous to [17], where the same procedure was also applied a second time to the already reduced data. Model training was only needed once before the actual cross-validation, since the features could be used in all subsequent cross-validation runs as no phenotypic value was required at this stage.

As an additional reference, principal component analysis (PCA) was used to create principle components as input features, as this is a common and widespread method.

Genomic prediction

The first model we used to evaluate the influence of dimensionality reduction was a GBLUP. We used two different models for fitting the GBLUP, depending on the type of the crop. For datasets consisting of $i = 1, ..., n$ homozygous lines we used the model

$$\mathrm{y} = 1\beta_0 + \mathrm{Zu} + \mathrm{e}. \tag{1}$$

The vector y is the response vector with the phenotypic observations of the $i = 1, ..., n$ inbred lines, 1 is an $n \times 1$ column vector of ones, β_0 is a fixed intercept, u is a vector of normally distributed random genotypic values with $\mathrm{E(u)} = 0$ and $\mathrm{cov(u)} = \mathrm{G}\sigma^2_A$, 0 is an $n \times 1$ column vector of zeroes, and Z is the design matrix for the genotypic values. In models where y contains the adjusted treatment means of a field trial, Z is the identity matrix I_n. e is a vector of randomly distributed residuals with $\mathrm{E(e)} = 0$ and $\mathrm{cov(e)} = \mathrm{I}_n\sigma^2_R$. I_n is an identity matrix and σ^2_R is the residual variance.

For hybrids, we used a model that included the general combining ability (GCA) and specific combining ability (SCA) as described in [29] and applied in [23]:

$$\mathrm{y} = 1\beta_0 + \mathbf{Z_1u_1} + \mathbf{Z_2u_2} + \mathbf{Z_su_s} + \mathrm{e}. \tag{2}$$

The vectors y, 1, β_0, and e are defined as stated above, this time for the $i = 1, ..., n$ hybrids. $\mathbf{u_1}$ and $\mathbf{u_2}$ are vectors of normally distributed random GCA effects of parents 1 and 2, respectively, with $\mathrm{E}(\mathbf{u_1}) = \mathrm{E}(\mathbf{u_2}) = 0$, $\mathrm{cov}(\mathbf{u_1}) = \mathbf{G_1}\sigma^2_1$, and $\mathrm{cov}(\mathbf{u_2}) = \mathbf{G_2}\sigma^2_2$, $\mathbf{u_s}$ is a vector of normally distributed random SCA effects associated with the specific combinations of parents 1 and 2, with $\mathrm{E}(\mathbf{u_s}) = 0$ and $\mathrm{cov}(\mathbf{u_s}) = \mathbf{G_s}\sigma^2_s$, and the design matrices $\mathbf{Z_1}$, $\mathbf{Z_2}$, and $\mathbf{Z_s}$ link the observations in y to the corresponding GCA and SCA effects.

The genomic relationship matrices G, $\mathbf{G_1}$ and $\mathbf{G_2}$ were calculated using method I of VanRaden [30]. $\mathbf{G_s}$ is defined as the Kronecker product $\mathbf{G_1} \otimes \mathbf{G_2}$. The variance components were estimated by restricted maximum likelihood (REML).

As an additional reference to evaluate the influence of dimensionality reduction in the context of ML, we used Random Forests [31]. We trained Random Forest models based on 20 different sets of hyperparameters, generated using a maximum entropy grid [32, 33]. Each set of hyperparameters was evaluated using a 5-fold cross-validation based on the training set. We tuned the number of trees ([100, 1000]), the column sampling rate ($[\frac{ncol}{100}, \frac{ncol}{3}]$) and the minimum observations required for an additional split ([1, 20]). We then created a stacked ensemble based on the models trained for every hyperparameter combination [34]. We used the LASSO algorithm [35] as the super learner for the new model.

Estimation of haplotype block effects

Baseline block effects were calculated using the sum of individual marker effects belonging to the same block [19].

To estimate block effects using the autoencoder, the block layer was directly connected to an additional unit that returned a prediction for yield ("yield output" in Fig. 1), which added a supervised part to the autoencoder. While both supervised and unsupervised

parts of the model were optimized simultaneously, the supervised part was only optimised on the basis of observations included in the training set while the unsupervised part of the model was optimised on the full data set. This combination of supervised and unsupervised, or labelled and unlabelled, data is known as semi-supervised learning [36].

For the semi-supervised form, the autoencoder used a custom loss function

$$\mathrm{MSE}(X, \hat{X}) - \mathrm{cor}(y, \hat{y}) + (\mathrm{MSE}(y, \hat{y}) + \lambda ||w_Z||^2) \tag{3}$$

where X represents the true markers and $\hat{X}$ the reconstructed markers, y represents the true yield and $\hat{y}$ the predicted yield, and λ is the ridge parameter of the squared ℓ_2-norm of the parameters of the block layer w_Z. The term $\mathrm{MSE}(X, \hat{X})$ is the loss function for the unsupervised part of the autoencoder. It represents the difference between the output of the autoencoder, which are the reconstructed markers, and the true markers that were used as inputs. By minimizing this error, the model learns to form a meaningful latent representation of the marker data in the block layer. Minimizing the negative correlation between the observed and predicted yield is equal to maximizing the same correlation. It is important that the rankings of observed and predicted phenotypes match, especially for the best observed genotypes, as these are the ones selected in a breeding program. Since the correlation is scale-free, the term $(\mathrm{MSE}(y, \hat{y}) + \lambda ||w_Z||^2)$ is introduced which minimizes the MSE between predicted and observed trait values. This ensures that predictions are on the same scale as the observed values. Similar to the ridge parameter in methods like RR-BLUP, $\lambda ||w_Z||^2$ leads to a shrinkage of the parameters connecting the block layer and the supervised part. This shrinks the parameters of blocks with lesser importance for the final prediction. In addition, the constraint encourages the model to rely more on the shared block representation, thereby stabilizing training and reducing the risk of overfitting in the supervised part. We set λ to 0.001 for every dataset.

Evaluation

A Mantel test [37] was used to assess the autoencoders ability to capture the genomic information in a lower dimensional space. A Mantel test compares the similarity between the distance matrices of the target matrices, in our case the genomic relationship matrices generated from the full SNP data and both reduced feature sets. Genomic relationship matrices were calculated according to method I of VanRaden [30].

To assess the suitability of our data for genomic prediction, we used GBLUP and RF based on the full marker data as the reference models. The features from the first and second reduction step were rescaled to be between -1 and 1 to compute the genomic relationship matrices used in GBLUP as this format was required by the software. The RF model used untransformed features.

For cross-validation, 100 training and test sets based on 80%/20% of the data were created and tested with each combination of method (GBLUP and RF) and feature set (full SNPs, reduction step 1 and reduction step 2). Additionally, 100 training and test sets were created for data sets of hybrid crops (Ra1, Mz1, Mz2) for which the test sets only consisted of T0 hybrids, i.e. hybrids where both parental lines did not appear in the training set. The prediction accuracy of a model was defined as the correlation between the observed and predicted yield. Prediction accuracy and computation time were recorded for every cross-validation run.

Table 1 Absolute and relative dimension of the full set of SNPs (SNP), the output of the first autoencoder (AE1) and the second reduction step (AE2) for all datasets. Additionally, amounts of markers with no block assignment are listed

	SNP	AE1	AE2	Unassigned SNPs
Ra1	10939 (100%)	1458 (13.33%)	177 (1.62%)	1472
Mz1	28803 (100%)	4140 (14.37%)	534 (1.85%)	14061
Mz2	267818 (100%)	34643 (12.94%)	3819 (1.43%)	106500
Wh1	17221 (100%)	2353 (13.66%)	427 (2.48%)	5806
Ot1	29457 (100%)	2890 (9.81%)	432 (1.47%)	3090

Table 2 Mantel test of similarity between genomic relationship matrices derived from the full set of SNPs (SNP), the output of the first autoencoder (AE1) and the second reduction step (AE2) for all datasets. Results are provided in the form: *r*-value (*p*-value). The *r*-value stands for the correlation and ranges from -1 to 1 where 1 is perfect correlation

	SNP vs AE1	SNP vs AE2	AE1 vs AE2
Ra1	0.7559 (<0.001)	0.7176 (<0.001)	0.9669 (<0.001)
Mz1	0.9988 (<0.001)	0.9834 (<0.001)	0.9886 (<0.001)
Mz2	0.9499 (<0.001)	0.9348 (<0.001)	0.9855 (<0.001)
Wh1	0.7907 (<0.001)	0.8159 (<0.001)	0.9807 (<0.001)
Ot1	0.5322 (<0.001)	0.5392 (<0.001)	0.9419 (<0.001)

To explore the potential of semi-supervised autoencoders for genomic prediction and variant effect estimation, the prediction accuracy of the semi-supervised version of the autoencoder was compared to the standard GBLUP model using the same initial 100 training and test sets. Additionally, the haplotype block variant effects derived from the autoencoder were compared to the sum of the single marker effects belonging to the same block. The marker effects were derived from the GBLUP. As GBLUP and RR-BLUP are mathematically equivalent [38, 39], it is possible to transform the random effects of the genotypes into marker effects.

Software

Autoencoders were computed using Python 3.12 and PyTorch 2.5.1 [40] using an NVIDIA®Quadro RTX™4000 GPU. All other calculations were done using R 4.3.3 [41] on two Intel®Xeon®Platinum 8276 CPU. Marker filtering and haplotype block building was done using routines which we programmed in the C and R programming languages. Mantel test was done using the R package 'vegan' [42]. Genomic relationship matrices and GBLUP were calculated using 'sommer 4.2.0' [24, 25], for RF we used the 'tidymodels' framework [43] with 'ranger 0.16.0' [44]. Due to its large size, GBLUP for dataset Mz2 was calculated using ASReml 4.2.0.267 [45].

Results

Dimensionality reduction for genomic prediction

The magnitude of the dimension reduction is shown in Table 1. In general, the dimension of the features was reduced to 9-15% of the original dimension, where all datasets fell in the range of 13-14% with the exception of Ot1 (∼10%). After the second reduction step, only around 1-3% of the original dimensionality remained.

A Mantel test was used to compare the similarity between all the genomic relationship matrices (Table 2). All genomic relationship matrices had a significant *r*-value (*p*-value < 0.001) which indicated a high correlation between all tested matrices. The Mantel test

for comparing the full set of SNPs to the first autoencoder features showed high variability between the datasets. While the two maize datasetes showed very high similarity (values above 0.9), the oat dataset showed much lower similarity (around 0.5). The same pattern could be observed for the comparison of the SNPs to the features of the second autoencoder. The *r*-values generally showed the same tendencies as the previous test, with only small differences. Features AE1 and AE2 seemed to result in very similar genomic relationship matrices as the *r*-values were very high for all datasets (>0.9).

Results of the cross-validation for hybrids are shown in Fig. 2. For hybrids, the reduction in the data dimensionality did not lead to a decrease in prediction accuracy. In the cross-validation scenario where parents of the hybrids in the test set also appeared in the training set (T1/T2 hybrids), prediction accuracy was equivalent for different feature sets within algorithms. In the cases of Ra1 and Mz1, prediction accuracy was also equivalent between algorithms while RF performed slightly worse for dataset Mz2. In the case of the T0-scenario, prediction accuracies within algorithms showed slightly more variability between feature sets. The changes in the prediction accuracy were still very minor, with the only exception being GBLUP for Ra1, where prediction accuracy improved from 0.1 to 0.14 and 0.2 with the first and second reduction step. This was not observed for RF for the same dataset. Similar results were observed for the inbred datasets (Fig. 2). For lines, the prediction accuracy was equal between the two algorithms and across feature sets. Prediction accuracies were generally higher than when principle components were used as inputs (Figure S3).

The GBLUP was much faster to compute and computation time was mostly unaffected by the change in the data dimension (Table 3). The average computation time for one RF run, including grid search, was drastically reduced in both steps. While training the RF models with the full set of SNPs took several minutes minimum for all algorithms, even up to more than two hours per run in the case of Mz2, computation time of three of the

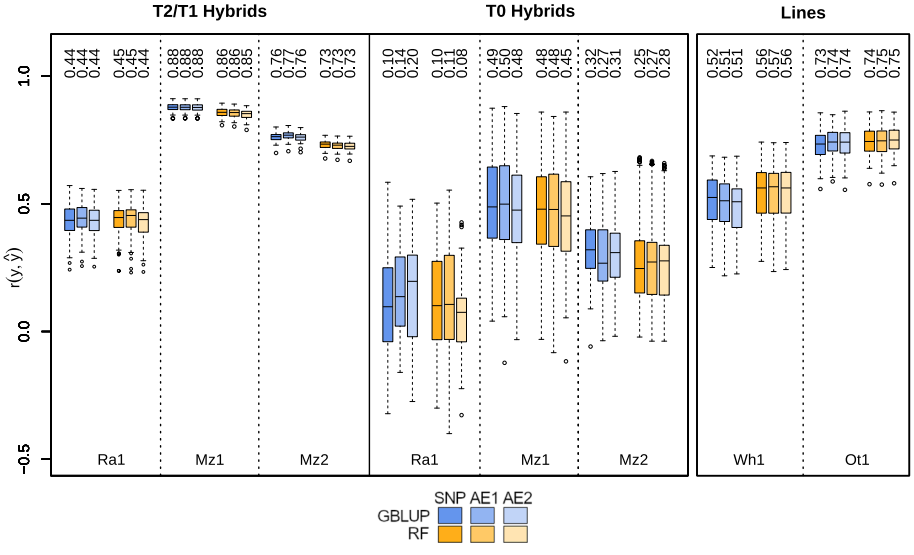

Fig. 2 Prediction accuracy of all cross-validation runs. Blue boxplots indicate GBLUP and orange indicate RF, wheares the reduced saturation indicates reduced feature sets. Numbers above the boxplots indicate the median prediction accuracy. Left shows T2/T1 hybrids, i.e. the 100 regular cross-validation runs where either one (T1) or both (T2) parental lines of the hybrids in the test set were available in the training set. The center shows 100 cross-validation runs where parental lines of hybrids in the test set were removed from the training set (T0). Plot on the right shows prediction accuracy for the cross-validation of the line datasets

Table 3 Mean computation time in minutes. GBLUP and RF refer to the methods used. SNP, AE1 and AE2 refer to the feature set used with each method and represent the full set of SNP markers (SNP), the output of the first autoencoder (AE1) and the second reduction step (AE2)

	GBLUP SNP	GBLUP AE1	GBLUP AE2	RF SNP	RF AE1	RF AE2
Ra1	0.23	0.08	0.09	4.19	0.94	0.20
Mz1	0.10	0.49	0.53	21.86	2.84	0.93
Mz2	7.37	7.22	9.77	162.49	34.32	4.35
Wh1	>0.01	>0.01	>0.01	2.64	0.57	0.18
Ot1	>0.01	>0.01	>0.01	3.98	0.79	0.21

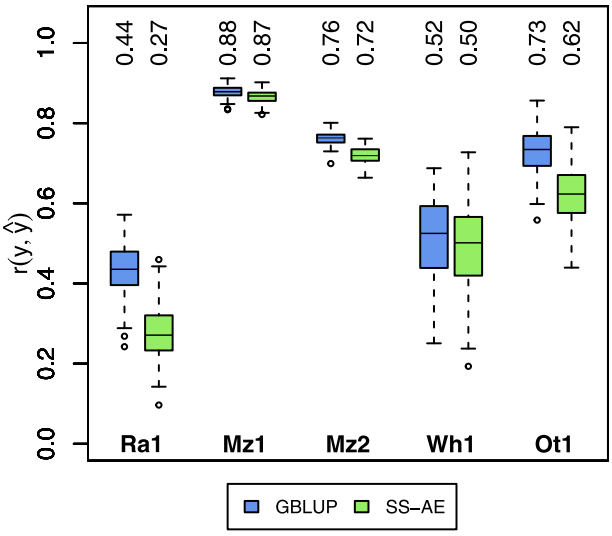

Fig. 3 Prediction accuracy of all cross-validation runs for all datasets. Green boxplots indicate the semi-supervised autoencoder (SS-AE), blue boxplots indicate GBLUP. Numbers above the boxplots indicate the medians of the prediction accuracy. For hybrids, training and test sets were identical to the T2/T1 scenario

datasets (Ra1, Ot1, Wh1) was reduced to under one minute after the first reduction step, and Mz1 requiring under one minute after the second reduction step. For dataset Mz2, a reduction from more than two hours to 30 to 4 min after the first and second reduction steps could be observed. Computation time after the second reduction step was even faster than the GBLUP for the same dataset with the full SNP data.

Semi-supervised estimation of block effects

We observed varying differences in the prediction accuracy between the semi-supervised autoencoder and the GBLUP (Fig. 3). Generally, the semi-supervised autoencoder performed worse than the GBLUP. While prediction accuracy for both methods was equal for datasets Mz1 and, to some degree, Mz2 and Wh1, the prediction accuracy of the autoencoder was lower compared to GBLUP for datasets Ra1 and Ot1.

We compared the haploblock variant effects estimated by the semi-supervised autoencoder to the marker effects estimated by GBLUP, summed up for every block. Figure 4 shows the distribution of the correlation between those variant effects for each block. The figure is exemplary and based on the effects estimated for the first cross-validation run for every dataset. The overall tendency was towards a positive correlation between both methods. However, different patterns in the distribution of the correlations could be observed. While Ra1 showed a tendency towards having more observations in the tails of the distribution, it also showed some similarity to a left-skewed distribution. The

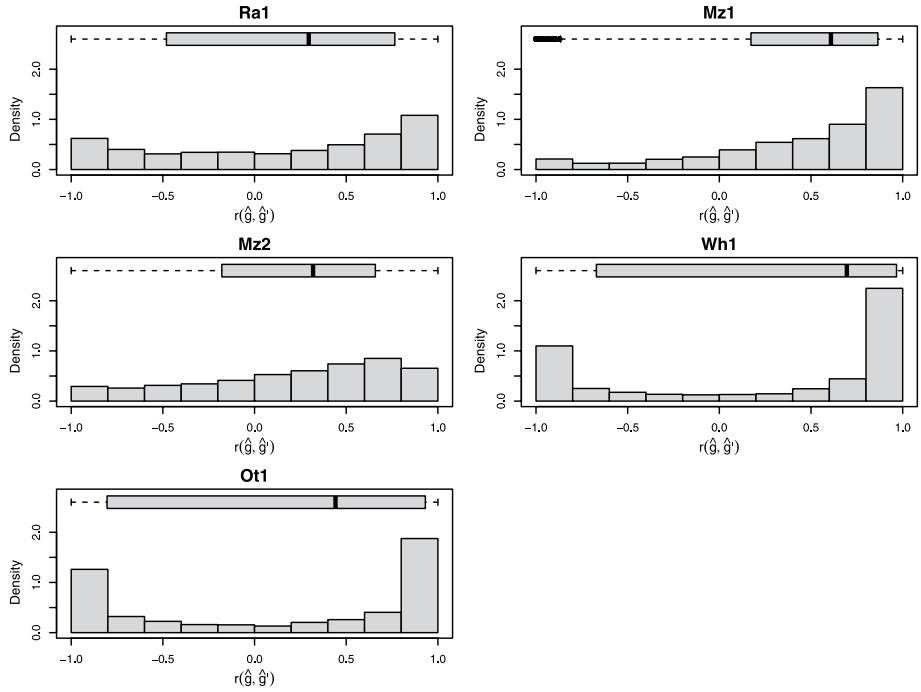

Fig. 4 Histograms and boxplots showing the distribution of correlation between effects of encoded blocks ($\hat{g}$) and summed up marker effects ($\hat{g}'$). For each block individually, correlation was recorded between autoencoder-based variant effects and marker effects summed up for every variant

distributions for Mz1 and Mz2 showed a clear left-skewed distribution with relatively few of the observed correlations falling below 0 and with each median correlation above 0.5. For the line datasets Wh1 and Ot1, the distribution showed a u-shape with most of the blocks falling into either a nearly perfect positive or negative correlation, with a stronger tendency towards a positive correlation.

A direct comparison of block variant effects based on the autoencoder and the block-wise sum of the marker effects is shown in Fig. 5. The observed values were scaled for both approaches due to the absence of an intercept in the autoencoder, resulting in larger effect sizes compared to those generated using GBLUP. The overall trends in effects were comparable, but the distribution of block variant effects was more symmetrical around 0 for single markers. For the encoded effects, there was a stronger tendency toward positive effects (above 0), while fewer negative effects, especially strong negative effects, were observed. Effects from the autoencoder showed some extreme positive values, relative to which most other effects appeared relatively small. Marker-based effects were more evenly distributed across all blocks, with fewer extreme values.

Discussion

Autoencoders for dimensionality reduction

In this study, we developed a novel autoencoder architecture based on the concept of haplotype-based blocks to compress high dimensional genetic data while preserving its core information content. Our first goal was to assess how much relevant information is retained within the encoded features extracted from the autoencoder and to determine their usefulness as inputs for prediction models. The Mantel test on the different feature sets showed a high similarity between the genomic relationship matrices generated with

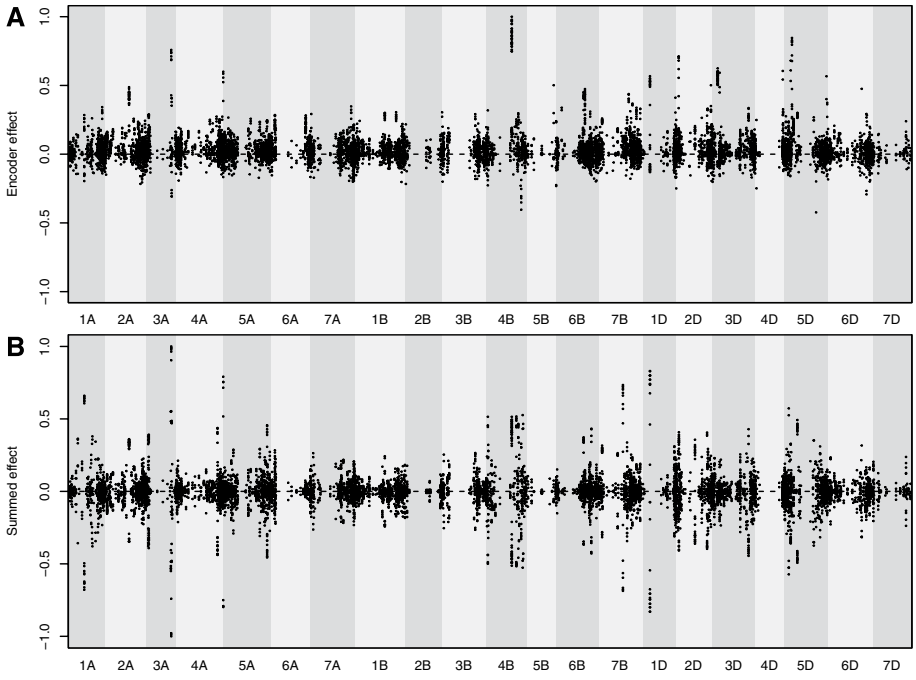

Fig. 5 Comparison of effects estimated for each haplotype block variant using the semisupervised autoencoder **A** to the block-wise sum of effects estimated using a GBLUP **B**. Estimation is based on the first cross-validation run for dataset Wh1. Differently highlighted areas indicate different chromosomes. Position on the x-axis indicates physical distance between blocks. The y-axis was scaled for both methods individually through dividing all effects by the largest effect of the respective method, thus scaling all effects from -1 to 1

those features (Table 2). While all genomic relationship matrices were significantly correlated with each other, the *r*-value between full and reduced data was comparatively low for dataset Ot1. For the maize datasets, the similarity was nearly perfect between full and reduced data. We conclude from this that the majority of genetic relationships contained within the marker data are retained during dimensionality reduction. Additionally, the genomic relationship matrices based on the first and second reduction steps were highly similar in all cases. This implies that reducing the already compressed data further does not result in the loss of any more information regarding the relationship structure in our data. This is reaffirmed by the consistency in the observed prediction accuracy for all datasets and algorithms. Compared to a widespread dimensionality reduction method, PCA, prediction accuracies of RF based on the autoencoder features were also higher (Figure S3). In all cases, prediction accuracy remained largely unaffected by the change in the dimensionality (Fig. 2). Through the dimensionality reduction of approximately 85 to 90% in the initial step and up to 98% in the second step (Table 1), computation time for ML algorithms was drastically reduced (Table 3). Computation time for GBLUP was unaffected as the dimension of the genomic relationship matrices depends only on the number of genotypes tested. Fluctuations in computation time may be related to a higher workload on the server side or other factors beyond the scope of this study.

In summary, we generated features in which each haplotype block is represented by a single variable, and the variables representing the blocks contain most, if not all, of the information from the full marker data. As our model guarantees dimension reduction, it overcomes the typical problem of an increased dataset dimension encountered with haplotype blocks [15]. This may help to avoid the potential problem of decreased

prediction accuracy due to multicollinearity when there are many block variants [46]. We conclude that locally connecting markers belonging to the same blocks based on flanking LD in an autoencoder is an effective way to reduce the dimensionality of genetic data. Our method therefore solves the problems faced in an earlier study [8], where autoencoders did not seem to be useful when using a standard architecture based on fully connected layers, as the prediction accuracy declined drastically. By reducing the number of features in the dataset training models for RF and other ML algorithms takes much less time and uses fewer computational resources. This can be advantageous in many situations, such as when computation time or model size are critical factors or related to costs (e.g. if rented cloud computing space is used or local space is limited), or during hyperparameter tuning, which is an integral part of ML [12]. A smaller model could be used to quickly find optimal regions in a very large hyperparameter space or create large ensembles of models. This is important as ensembles of models often perform better than individual models [7, 47, 48], especially when the ensemble is made up of diverse models [49]. The additional cost of training the unsupervised autoencoder is comparatively low as it only has to be trained once and does not require phenotypic measurements. Similar to the popular ML approach of transfer learning [50], a fully trained model can be stored and fine-tuned for each new cycle of a breeding program using newly generated data or for modeling new traits closely related to the original task in its semi-supervised form. This process allows the model to continually improve and adapt to the evolving germplasm within the breeding program.

Autoencoders are rarely used for genomic prediction in plant breeding. Existing studies focus on 2- and 3-dimensional [51, 52] or hyperspectral data [53, 54]. Tross et al. [54] used autoencoders to reduce the dimension of hyperspectral data, where some autoencoder features also showed similarity to known genes influencing leaf phenotypes. To our knowledge, the study that comes closest to ours is Islam et al [17]. Our findings regarding prediction accuracy are in accordance with the findings in their [17] study. However, our method has some key differences. The most notable is that we used a single autoencoder to train the full model, whereas Islam et al [17] used a series of small autoencoders. The equivalent to this would be training an individual autoencoder for each block, which our method can relatively easy be adapted to. However, as we have fully connected layers in the decoder part and also include the markers without block assignment in the output layer, we argue that having a single autoencoder comes with the advantage of the model learning the interconnectedness between blocks and all markers. This can be important as one possible reason for markers not being assigned to a block is that their physical position is not determined correctly and therefore actual existing blocks are broken up. We assume that in our case, a unit in the block layer would also be a good predictor for markers with wrong physical positioning and this information is therefore also accounted for in the model. Secondly, Islam et al. [17] chose to one-hot encode markers, therefore quadrupling the amount of input variables. The advantage is that their approach is not limited to biallelic SNP markers. However, as most markers are biallelic, we believe that this increases computation time unnecessarily in most cases.

Haplotype block effects

Our second goal was to enhance our model to create a novel approach of estimating the effects of haplotype block variants that goes beyond simply adding up the local effects of

markers. For this, we extended our model to include a yield output layer which was used only for those training samples that were in the training set, while the unsupervised part was using all of the data. This concept is called semi-supervised learning [36] and to our knowledge, this approach has not ben utilized in plant breeding before. The advantage of semi-supervised learning is that it uses all of the data in the model. In supervised models, genetic data without phenotypic data is usually excluded completely during model training. This is important because it is often hypothesised that datasets in plant breeding might be too small for ML to work in general [4, 11, 55]. By maximising data usage, breeders can get the most out of the available data.

Comparing GBLUP and semi-supervised autoencoders, the results showed that GBLUP was performing better, albeit by a smaller (Mz1, Mz2, Wh1) or larger (Ra1, Ot1) margin (Fig. 3). One possible explanation is that we used the same approach for every dataset. Typically, the number of epochs, the learning rate, and architectural details such as the size of the hidden layer before the output layer are adapted for each dataset. In our study we prioritised creating a general framework and prove the feasibility of a novel method of estimating haplotype block effects instead of working on fine-tuning the model for every dataset as this would have unnecessarily increased the complexity of the methodology. It is reasonable to assume that prediction accuracy would further improve with dataset-specific parameters.

Some authors argue that Neural Networks may not be suitable for estimating marker effects when used for genomic prediction [56]. However, by incorporating prior knowledge about genetic linkage into our model architecture and a direct connection between block and yield output layer, we ensure that the effects are, by design, in the same unit as the target variable. The effects derived from the autoencoders and the block-wise sums of marker effects [19] were often highly correlated (Fig. 4), especially for datasets Wh1 and Ot1. This agreement between the methods confirms that the autoencoder effects are plausible and meaningful. However, since there were also many cases where the correlation was lower, it appears that the autoencoder identified alternative regions with important contribution to estimating the final yield. This may indicate areas where non-additive effects may play a role. It has been shown that haplotype blocks capture local epistatic effects to some degree [14] although with varying success [57]. As we optimize our model towards two objectives (yield and the reconstructed marker data) and pass the inputs through multiple layers with non-linear activation functions, our model is designed in a way that could help capture local epistatic effects. While this was not directly tested here, future work could explore whether the model can also capture more complex non-linear interactions beyond local epistasis, such as global epistatic effects. Training on a combination of labelled and unlabelled data may also act as a form of regularisation, leading to better model generalisation [58]. Our approach therefore provides an alternative method of estimating the effects of haplotype blocks directly on the variant level instead of relying on estimating effects for markers first. This could be used for proposed methods like "haplotype stacking", where the goal is to bring favourable combinations of blocks together [19, 21], as this is currently relying on block-wise sums of marker effects.

Limitations

A current limitation of our model is that, due to the use of a (-1, 0, 1) marker encoding, some haplotype block variants are effectively lost for hybrid datasets. This encoding does not take into account on which chromosome an allele is positioned. The following example (Figure S1) illustrates the problem: The cross between two homozygous parental lines where one has the block variant CCAC and the other CAAA would result in a block variant encoded as (-1, 0, 1, 0). Another cross between two different homozygous parents with the variants CAAC and CCAA would equally result in a hybrid with marker data (-1, 0, 1, 0). As a result, our models assigns the same effect to some block variants that are functionally different. This problem does not occur in the case when lines are predicted as they are homozygous.

While this is a constraint when estimating block effects for hybrids, this problem can be circumvented in future studies. A potential solution would be to provide a tuple of the two alleles (i.e. two variables) instead of a single binary variable. This would effectively double the size of the input and unsupervised output layers, while the principle of the architecture remains unchanged. The idea of representing SNPs with two codes, typically one for additive and one for dominance effects, has long been used in quantitative genetics and genomic prediction [30]. Recent implementations in efficient, multilocus mixed models [59, 60] demonstrate the continued relevance of this approach for GWAS. A similar extension based on design or genomic relationship matrices could also be applied in our framework by using two separate matrices to represent the two parental haplotypes in a block, thereby addressing the problem of ambiguous encodings. Adjusting our method would be required if a diverse set of genotypes with a high rate of heterozygosity is used. Any aspect related to our first objective was not affected as the primary goal was dimensionality reduction while maintaining a stable prediction accuracy. Therefore, the actual values of the features are not interpreted.

Conclusion

Our method overcomes the limitations of autoencoders that were encountered in previous studies, where prediction accuracy decreased when features generated using an autoencoder where used as inputs in another model. The novel haplotype-based autoencoder successfully compresses high-dimensional genetic data while preserving genetic relationships, as shown by the high similarity in relationship information and consistent prediction accuracy compared to the standard GBLUP model. This dimensionality reduction significantly decreases computation time for ML algorithms, making them advantageous for scenarios where computational efficiency is important or extensive hyperparameter tuning is required. Additionally, the semi-supervised learning approach offers a new way of estimating the effects of haplotype blocks that goes beyond simply adding up the individual marker effects for each block.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-025-06323-w.

Supplementary file 1.

Author contributions

PGH conceived and conducted the study and prepared the manuscript. EG analysed the field data for oat. MH and CZP conceived the design for the project in which the oat data were generated. MH carried out the crossings and selfings

for oat. MH, MM and RA collected the genotypic data for oat. SB, VK, KO, and IR collected the field data for oat. AH, RM, CR, ASe, and MS carried out the factorial crossings for wheat. AH, RRM, CR, and MS collected the field data for wheat. AA and TK conceived the design for the project in which the data for rapeseed were generated. MF and ASt conceived the design for the project in which the wheat data were generated.

Funding
Open Access funding enabled and organized by Projekt DEAL. The project SEEH in which part of the oat data was generated is supported by funds of the Federal Ministry of Agriculture, Food and Regional Identity (BMLEH) based on a decision of the parliament of the Federal Republic of Germany via the Federal Office for Agriculture and Food (BLE) under the Federal Organic Farming Scheme (FKZ 2822OE188). The projects FUGE and MultiResistGS in which part of the oat data and the wheat data were generated were supported by funds of the Federal Ministry of Agriculture, Food and Regional Identity (BMLEH) based on a decision of the parliament of the Federal Republic of Germany via the Federal Office for Agriculture and Food (BLE) under the innovation support programme (FKZ 28AINO2A20 and FKZ 28A8411A18). Ra1 data were generated as part of the Project PROGReSs (BMBF, FKZ 031A297A-J), which was funded by the German Federal Ministry for Education and Research (BMBF).

Data availability
The datasets Ot1 and Wh1 were originally generated within the research projects SEEH, FUGE, and MultiResistGS. Datasets Ot1 and Wh1, together with the R and Python scripts used in this study, are available in the GitHub repository: https://github.com/PGHeilmann/Haploencoder. Dataset Mz1 is accessible through the R package "sommer" and was originally published in Technow et al. (2014). The dataset Mz2 was obtained from the Genomes to Fields (G2F) initiative and is publicly available through the official project website: https://www.genomes2fields.org/resources. Dataset Ra1 was provided by NPZ Innovation GmbH for internal use and restrictions apply to the availability of these data.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
Steffen Beuch is employed by Nordsaat Saatzucht GmbH. Vivek Kurra and Raja Mehta are employed by Saatzucht Bauer GmbH & Co. KG. Ina Röhrs is employed by Landbauschule Dottenfelderhof e.V. Klaus Oldach is employed by KWS Lochow GmbH. Anja Hanemann is employed by Saatzucht Josef Breun GmbH & Co. KG. Carsten Reinbrecht and Marco Stucke are employed by Saatzucht Streng-Engelen GmbH & Co. KG. Amine Abbadi and Tobias Kox are employed by NPZ Innovation GmbH. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential Conflict of interest.

Author details
[1]Institute of Agronomy and Plant Breeding II, Justus Liebig University, Heinrich-Buff-Ring 26-32, 35392 Giessen, Germany
[2]Research on Agricultural Crops, Federal Research Centre for Cultivated Plants, Institute for Breeding, Julius Kühn Institute, Rudolf-Schick-Platz 3a, 18190 Sanitz, Germany
[3]Saatzucht Granskevitz, Nordsaat Saatzucht GmbH, Granskevitz 3, 18569 Schaprode, Germany
[4]Saatzucht Bauer GmbH & Co. KG, Landshuter Str. 3a, 93083 Obertraubling, Germany
[5]Leibniz Institute of Plant Genetics and Crop Plant Research (IPK) Gatersleben, Corrensstraße 3, 06466 Seeland, Germany
[6]KWS Lochow GmbH, Ferdinand-von-Lochow-Str. 5, 29303 Bergen, Germany
[7]Research & Breeding Dottenfelderhof, Landbauschule Dottenfelderhof e.V., Dottenfelderhof 1, 61118 Bad Vilbel, Germany
[8]Saatzucht Josef Breun GmbH & Co. KG, Amselweg 1, 91074 Herzogenaurach, Germany
[9]Saatzucht Streng-Engelen GmbH & Co. KG, Aspachhof, 97215 Uffenheim, Germany
[10]Institute for Resistance Research and Stress Tolerance, Julius Kühn Institute, Erwin-Baur-Str. 27, 06484 Quedlinburg, Germany
[11]NPZ Innovation GmbH, Hohenlieth-Hof, 24363 Holtsee, Germany

Received: 4 July 2025 / Accepted: 7 November 2025
Published online: 02 December 2025

References

1. Bernardo R. Genetic Models for Predicting Maize Single-Cross Performance in Unbalanced Yield Trial Data. Crop Science. 1995;35(1) https://doi.org/10.2135/cropsci1995.0011183X003500010026x.
2. Whittaker JC, Thompson R, Denham MC. Marker-assisted selection using ridge regression. Genet Res. 2000;75(2):249–52. https://doi.org/10.1017/S0016672399004462.
3. Meuwissen THE, Hayes BJ, Goddard ME. Prediction of total genetic value using genome-wide dense marker maps. Genetics. 2001;157:1819–29. https://doi.org/10.1093/genetics/157.4.1819.
4. Azodi CB, Bolger E, McCarren A, Roantree M, de los Campos G, Shiu SH. Benchmarking Parametric and Machine Learning Models for Genomic Prediction of Complex Traits. G3: Genes, Genomes, Genetics. 2019 11;9(11):3691–3702. https://doi.org/10.1534/g3.119.400498.

5. Zingaretti LM, Gezan SA, Ferrão LFV, Osorio LF, Monfort A, Muñoz PR, et al. Exploring deep learning for complex trait genomic prediction in polyploid outcrossing species. Front Plant Sci. 2020. https://doi.org/10.3389/fpls.2020.00025.
6. John M, Haselbeck F, Dass R, Malisi C, Ricca P, Dreischer C, et al. A comparison of classical and machine learning-based phenotype prediction methods on simulated data and three plant species. Front Plant Sci. 2022. https://doi.org/10.3389/fpls.2022.932512.
7. Heilmann PG, Frisch M, Abbadi A, Kox T, Herzog E. Stacked ensembles on basis of parentage information can predict hybrid performance with an accuracy comparable to marker-based GBLUP. Front Plant Sci. 2023. https://doi.org/10.3389/fpls.2023.1178902.
8. Heilmann PG, Difabachew YF, Frisch M, Moritz AL, Stahl A, Wittkop B, et al. Machine learning for prediction of resistance scores in wheat (*Triticum aestivum* L.). Plant Breed. 2024;144(2):192–205. https://doi.org/10.1111/pbr.13235.
9. Lourenco VM, Ogutu JO, Rodrigues RA, Posekany A, Piepho HP. Genomic prediction using machine learning: a comparison of the performance of regularized regression, ensemble, instance-based and deep learning methods on synthetic and empirical data. BMC Genom. 2024;25(1):152. https://doi.org/10.1186/s12864-023-09933-x.
10. Montesinos-López A, Montesinos-López OA, Ramos-Pulido S, Mosqueda-González BA, Guerrero-Arroyo EA, Crossa J, et al. Artificial intelligence meets genomic selection: comparing deep learning and GBLUP across diverse plant datasets. Front Genet. 2025. https://doi.org/10.3389/fgene.2025.1568705.
11. Hayes BJ, Chen C, Powell O, Dinglasan E, Villiers K, Kemper KE, et al. Advancing artificial intelligence to help feed the world. Nat Biotechnol. 2023;41:1188–9. https://doi.org/10.1038/s41587-023-01898-2.
12. Probst P, Boulesteix AL, Bischl B. Tunability: importance of hyperparameters of machine learning algorithms. J Mach Learn Res. 2019;20(53):1–32.
13. Cuyabano BC, Su G, Lund MS. Genomic prediction of genetic merit using LD-based haplotypes in the Nordic Holstein population. BMC Genom. 2014;15(1):1171. https://doi.org/10.1186/1471-2164-15-1171.
14. Jiang Y, Schmidt RH, Reif JC. Haplotype-based genome-wide prediction models exploit local epistatic interactions among markers. G3: Genes, Genomes, Genetics. 2018;8(5):1687–99. https://doi.org/10.1534/g3.117.300548.
15. Difabachew YF, Frisch M, Langstroff AL, Stahl A, Wittkop B, Snowdon RJ, et al. Genomic prediction with haplotype blocks in wheat. Front Plant Sci. 2023. https://doi.org/10.3389/fpls.2023.1168547.
16. Goodfellow I, Bengio Y, Courville A. Deep Learning. Cambridge, MA.: MIT Press; 2016. Available from: http://www.deeplearningbook.org.
17. Islam T, Kim CH, Iwata H, Shimono H, Kimura A. DeepCGP: a deep learning method to compress genome-wide polymorphisms for predicting phenotype of rice. IEEE/ACM Trans Comput Biol Bioinf. 2023;20(3):2078–88. https://doi.org/10.1109/TCBB.2022.3231466.
18. Weber SE, Frisch M, Snowdon RJ, Voss-Fels KP. Haplotype blocks for genomic prediction: a comparative evaluation in multiple crop datasets. Front Plant Sci. 2023. https://doi.org/10.3389/fpls.2023.1217589.
19. Voss-Fels KP, Stahl A, Wittkop B, Lichthardt C, Nagler S, Rose T, et al. Breeding improves wheat productivity under contrasting agrochemical input levels. Nature Plants. 2019;5(7):706–14. https://doi.org/10.1038/s41477-019-0445-5.
20. Villiers K, Voss-Fels KP, Dinglasan E, Jacobs B, Hickey L, Hayes BJ. Evolutionary computing to assemble standing genetic diversity and achieve long-term genetic gain. Plant Genome. 2024;17(2):e20467. https://doi.org/10.1002/tpg2.20467.
21. Tong J, Tarekegn ZT, Jambuthenne D, Alahmad S, Periyannan S, Hickey L, et al. Stacking beneficial haplotypes from the Vavilov wheat collection to accelerate breeding for multiple disease resistance. Theor Appl Genet. 2024;137(12):274. https://doi.org/10.1007/s00122-024-04784-w.
22. Browning BL, Zhou Y, Browning SR. A one-penny imputed genome from next-generation reference panels. Am J Hum Genet. 2018;103(3):338–48. https://doi.org/10.1016/j.ajhg.2018.07.015.
23. Technow F, Schrag TA, Schipprack W, Bauer E, Simianer H, Melchinger AE. Genome properties and prospects of genomic prediction of hybrid performance in a breeding program of maize. Genetics. 2014;8(197):1343–55. https://doi.org/10.1534/genetics.114.165860.
24. Covarrubias-Pazaran G. Genome assisted prediction of quantitative traits using the R package sommer. PLoS ONE. 2016;11:1–15.
25. Covarrubias-Pazaran G. Software update: Moving the R package sommer to multivariate mixed models for genome-assisted prediction. biorxv. 2018
26. Lima DC, Washburn JD, Varela JI, Chen Q, Gage JL, Romay MC, et al. Genomes to fields 2022 maize genotype by environment prediction competition. BMC Res Notes. 2023;16(1):148. https://doi.org/10.1186/s13104-023-06421-z.
27. Genomes to Fields Initiative.: Resources. Accessed 20 May 2025. https://www.genomes2fields.org/resources/.
28. Zhao H, Nettleton D, Soller M, Dekkers JCM. Evaluation of linkage disequilibrium measures between multi-allelic markers as predictors of linkage disequilibrium between markers and QTL. Genet Res. 2005;86(1):77–87. https://doi.org/10.1017/S001667230500769X.
29. Stuber CW, Cockerham CC. Gene effects and variances in hybrid populations. Genetics. 1966;54(6):1279–86. https://doi.org/10.1093/genetics/54.6.1279.
30. VanRaden PM. Efficient methods to compute genomic predictions. J Dairy Sci. 2008;91(11):4414–23. https://doi.org/10.3168/jds.2007-0980.
31. Breiman L. Random forests. Mach Learn. 2001;10(45):5–32. https://doi.org/10.1023/A:1010933404324.
32. Shewry M, Wynn HP. Maximum entropy sampling. J Appl Stat. 1987;14:165–70.
33. Kuhn M, Frick H.: dials: Tools for Creating Tuning Parameter Values. R package version 1.2.1. Available from: https://CRAN.R-project.org/package=dials.
34. Wolpert DH. Stacked generalization. Neural Netw. 1992;5(2):241–59. https://doi.org/10.1016/S0893-6080(05)80023-1.
35. Tibshirani R. Regression shrinkage and selection via the lasso. J Roy Stat Soc: Ser B (Methodol). 1996;58(1):267–88.
36. Amini MR, Usunier N. Learning with Partially Labeled and Interdependent Data. Springer Cham; 2015.
37. Mantel N. The detection of disease clustering and a generalized regression approach. Can Res. 1967;27(2):209–20 (https://aacrjournals.org/cancerres/article-pdf/27/2_Part_1/209/2382183/cr0272p10209.pdf).
38. Shen X, Alam M, Fikse F, Rönnegård L. A novel generalized ridge regression method for quantitative genetics. Genetics. 2013;193(4):1255–68. https://doi.org/10.1534/genetics.112.146720 (https://academic.oup.com/genetics/article-pdf/193/4/1255/42118481/genetics1255.pdf.).

39. de Vlaming R, Groenen PJF. The current and future use of ridge regression for prediction in quantitative genetics. Biomed Res Int. 2015;2015:143712. https://doi.org/10.1155/2015/143712.
40. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. In: PyTorch: an imperative style, high-performance deep learning library. Red Hook, NY, USA: Curran Associates Inc.; 2019.
41. R Core Team.: R: A Language and Environment for Statistical Computing. Vienna, Austria. Available from: https://www.R-project.org/.
42. Oksanen J, Simpson GL, Blanchet FG, Kindt R, Legendre P, Minchin PR, et al.: vegan: Community Ecology Package. R package version 2.6-10. Available from: https://CRAN.R-project.org/package=vegan.
43. Kuhn M, Wickham H.: tidymodels: a collection of packages for modeling and machine learning using tidyverse principles. Available from: https://www.tidymodels.org.
44. Wright MN, Ziegler A. Ranger: a fast implementation of random forests for high dimensional data in C++ and R. J Stat Softw. 2017;77(1):1–17. https://doi.org/10.18637/jss.v077.i01.
45. The VSNi Team.: asreml: Fits Linear Mixed Models using REML. R package version 4.2.0.267.
46. Matias FI, Galli G, Correia Granato IS, Fritsche-Neto R. Genomic prediction of autogamous and allogamous plants by snps and haplotypes. Crop Sci. 2017;57(6):2951–8. https://doi.org/10.2135/cropsci2017.01.0022.
47. Liang M, Chang T, An B, Duan X, Du L, Wang X, et al. A stacking ensemble learning framework for genomic prediction. Front Genet. 2021. https://doi.org/10.3389/fgene.2021.600040.
48. Kick DR, Washburn JD. Ensemble of best linear unbiased predictor, machine learning and deep learning models predict maize yield better than each model alone. In silico Plants. 2023 09;5(2):diad015. https://doi.org/10.1093/insilicoplants/diad015.
49. Tomura S, Wilkinson MJ, Cooper M, Powell O. Improved genomic prediction performance with ensembles of diverse models. G3 Genes|Genomes|Genetics. 2025 03;15(5):jkaf048. https://doi.org/10.1093/g3journal/jkaf048.
50. Lu J, Behbood V, Hao P, Zuo H, Xue S, Zhang G. Transfer learning using computational intelligence: A survey. Knowledge-Based Systems. 2015;80:14–23. 25th anniversary of Knowledge-Based Systems. https://doi.org/10.1016/j.knosys.2015.01.010.
51. de Villiers HAC, Otten G, Chauhan A, Meesters L. Autoencoder-based 3D representation learning for industrial seedling abnormality detection. Comput Electron Agric. 2023. https://doi.org/10.1016/j.compag.2023.107619.
52. Jurado-Ruiz F, Rousseau D, Botia JA, Aranzana MJ. GenoDrawing: an autoencoder framework for image prediction from SNP markers. Plant Phenom 2023, https://doi.org/10.34133/plantphenomics.0113.
53. Powadi A, Jubery TZ, Tross MC, Schnable JC, Ganapathysubramanian B. Disentangling genotype and environment specific latent features for improved trait prediction using a compositional autoencoder. Front Plant Sci. 2024. https://doi.org/10.3389/fpls.2024.1476070.
54. Tross MC, Grzybowski MW, Jubery TZ, Grove RJ, Nishimwe AV, Torres-Rodriguez JV, et al. Data driven discovery and quantification of hyperspectral leaf reflectance phenotypes across a maize diversity panel. Plant Phenome J. 2024;7(1):e20106. https://doi.org/10.1002/ppj2.20106.
55. Montesinos-López OA, Montesinos-López A, Pérez-Rodríguez P, Barrón-López JA, Martini JWR, Fajardo-Flores SB, et al. A review of deep learning applications for genomic selection. BMC Genom. 2021;22:19. https://doi.org/10.1186/s12864-020-07319-x.
56. Ubbens J, Parkin I, Eynck C, Stavness I, Sharpe AG. Deep neural networks for genomic prediction do not estimate marker effects. Plant Genome. 2021;14(3):e20147. https://doi.org/10.1002/tpg2.20147.
57. Da Y, Liang Z, Prakapenka D. Multifactorial methods integrating haplotype and epistasis effects for genomic estimation and prediction of quantitative traits. Front Genet. 2022. https://doi.org/10.3389/fgene.2022.922369.
58. van Engelen JE, Hoos HH. A survey on semi-supervised learning. Mach Learn. 2020;109(2):373–440. https://doi.org/10.1007/s10994-019-05855-6.
59. Wang JT, Chang XY, Zhao Q, Zhang YM. FastBiCmrMLM: a fast and powerful compressed variance component mixed logistic model for big genomic case-control genome-wide association study. Brief Bioinform 2024 06;25(4):bbae290. https://doi.org/10.1093/bib/bbae290.
60. Wang J, Chen Y, Shu G, Zhao M, Zheng A, Chang X, et al. Fast3VmrMLM: a fast algorithm that integrates genome-wide scanning with machine learning to accelerate gene mining and breeding by design for polygenic traits in large-scale GWAS datasets. Plant Commun. 2025;6(7):101385. https://doi.org/10.1016/j.xplc.2025.101385.

Publisher's Note