Chapter 5

CropGBM: An Ultra-Efficient Machine Learning Toolbox for Genomic Selection-Assisted Breeding in Crops

Yuetong Xu, John D. Laurie, and Xiangfeng Wang

Abstract

Continued improvement and falling costs of DNA sequencing have accelerated the increase in genomic resources for crop plants. From these efforts, considerable genetic diversity has been found and is aiding in the identification of markers for breeding purposes. High-density molecular markers have allowed for marker-assisted selection of quantitative traits that are controlled by a small number of genes. Recently, whole genomic selection has been proposed where markers genome-wide are used to estimate the contribution of all loci to traits of interest. In this chapter we outline the steps needed to perform genomic selection using machine learning. We describe our method called Crop Genomic Breeding Machine (CropGBM) and demonstrate its use on diverse maize lines containing high-density markers.

Key words Marker-assisted selection, Genomic selection, Machine learning, Artificial intelligence

1 Introduction

Traditional breeding is primarily based on phenotype selection. Breeders select excellent offspring by observing crop phenotypes to achieve genetic improvement of target traits. Advances in molecular genetic technology have identified a wide range of genetic variations in crop genomes, from which a large number of molecular markers associated with traits are selected to accelerate the breeding process. Marker-assisted selection (MAS) [1] has been commonly utilized to assist breeding, and is usually only effective for qualitative traits controlled by single genes with major effects. For many important agronomic traits controlled by numerous quantitative trait loci (QTLs) in which each QTL contributes a minor effect to the phenotype, MAS is usually not effective [2–4]. First of all, markers adjacent to QTL sites with insignificant effect may be omitted. Secondly, due to the insufficient coverage

Software website: https://ibreeding.github.io/

Andriy Bilichak and John D. Laurie (eds.), *Accelerated Breeding of Cereal Crops*, Springer Protocols Handbooks,
https://doi.org/10.1007/978-1-0716-1526-3_5, © Springer Science+Business Media, LLC, part of Springer Nature 2022

and low-density markers, the accumulative influence of the QTLs contributing to the traits cannot be accurately estimated. Therefore, MAS is usually unable to direct breeding for quantitative traits, such as yield and resistance traits. With the rapid advance of sequencing technology that enables high-density molecular markers, whole genomic selection (GS) technology has been proposed. GS utilizes genome-wide markers, which may avoid the omission of small-effect markers, to estimate the contribution of all loci in the genome to traits, based on the calculation of a genomic estimated breeding value (GEBV) derived from a GS predictive models with the genome-wide markers as input [5]. Assisted with the predicted phenotype or GEBV from the GS model, breeders thus can precisely select breeding material or design breeding schemes to greatly shorten the breeding cycle. For crops with heterosis, breeders evaluate inbred lines not only based on parental phenotypes but also the potential for creating superior hybrids. In the seed industry, GS usually uses 20% of the total samples as training population to construct a predictive model to predict the phenotype of the offspring of untested hybrid combinations, from which the general combining ability (GCA) is estimated to accelerate screening [6]. Also, GS can infer the genotype of the F_1 hybrid offspring based on their parental genotypes, which may greatly reduce the cost for genotyping [7].

Currently, the most commonly used GS algorithm is to apply regression analysis to predict phenotypes, which are mainly divided into two categories: the BLUP (best linear unbiased prediction) method represented by gBLUP (genomic BLUP) and rrBLUP (ridge regression BLUP) and the Bayesian method represented by Bayes-A and Bayes-B. The traditional BLUP-type method constructs a pedigree matrix inferred from the samples based on the breeding history, and then uses MLM (mixed linear model) to calculate the EBV (estimated breeding value) [8]. In contrast, gBLUP computes a pedigree matrix containing the genomic relationship between each pair of the samples based on the genotype of all the samples in one breeding population, and then uses this correlation matrix instead of the kinship matrix to genomically estimate EBV (referred to as GEBV) for each sample. Additionally, rrBLUP treats the labeling effect as a random effect, assuming it is a standard normal distribution, and then sums up the labeling effects to estimate GEBV [9]. The Bayesian method assumes that the labeling effect obeys a certain prior distribution, so problems such as hyper-parameter optimization for the prior distribution exist. However, if the mark set is too large, the performance of the Bayesian method is not comparable to the BLUP-based method [5]. Although gBLUP has been widely used in building GS models due to its high efficiency and accuracy, disadvantages still exist especially when given extra-large sample sets. First, processing of genotypic data is complicated, as gBLUP requires the conversion of genotypes from the character forms of A, T, G, and C to digital forms of 0, 1, and 2, according to the minor allele frequencies (MAFs) for each SNP site; when the population is changed, this conversion has to be redone which may cost significant amounts of computing time. Second, gBLUP struggles to capture the complex nonlinear relationships between genotype and phenotypes, as gBLUP is based on linear mixed-effect models. Third, the kinship matrix derived from one designated population is non-extensible, which means that if the training population is changed, the kinship matrix has to be reconstructed, especially when the genetic composition and population structure are complex; otherwise, problematic model fitting may occur that generates a great deal of false-positive results. Fourth, the linear model of gBLUP is efficient at solving regression problems; however, many agronomic traits are multi-classification problems which require a more appropriate method for solving rather than using a regression model.

In the recent years, high-throughput genotyping and phenotyping technologies have been rapidly advancing concomitantly with lowering costs. This has resulted in the rapid accumulation of genotypic and phenotypic data from breeding materials, inbred lines, and F_1 hybrids collected from actual breeding programs. Thus, the ever-accumulating large datasets are gradually forming a Big Data environment, which increases the feasibility of utilizing machine learning-based paradigms to perform data-driven decision-making for breeding. To overcome the innate shortcomings of traditional GS methods, adoption of machine learning (ML) theory and algorithms for genomic selection has been highly anticipated. Several types of ML algorithms including SVM (support-vector machine), RF (random forest), GB (gradient boost), LGB (light gradient boost), and other machine learning methods are gradually being applied to biological problems such as genomic selection [10–12].

ML methods do not require the distribution and variance of labeling effects, and can fully explore the nonlinear relationship between labels by continuously learning data and optimizing parameters. According to our unpublished research (Yan et al., unpublished), an ensemble learning paradigm – gradient boost (GB) in ML outperformed other types of ML models. One of its variants LightGBM (gradient boosting machine) exhibited great advantages over gBLUP, rrBLUP, and Bayesian methods, as well as other types of ML algorithms. A great advantage is its ultra-efficiency in terms of computing when training models. Performance testing on a large population containing 100,000 samples showed that, while rrBLUP may take almost 1 month to accomplish model training, LightGBM only used 9 min on a small-scale server. Thus, we implemented the LightGBM algorithm as a toolbox, called CropGBM (Crop Genomic Breeding Machine), to carry out genomic selection for breeding. In this chapter, we illustrate the usage of CropGBM in terms of utilizing machine learning for genomic selection-assisted breeding in crops, including streamline analysis of data preprocessing, population structure analysis, feature selection, and phenotype prediction. The CropGBM toolbox and example dataset are freely available for academic research proposes at https://ibreeding.github.io.

2 Example Dataset

The example dataset used here for demonstrating the functionality of CropGBM is composed of 6210 maize F_1 hybrids generated by crossing 207 maternal lines with 30 paternal lines, including genotypes of 4903 SNPs and phenotypes of flowering time (DTT, days to tasseling), plant stature (PH, plant height), and kernel yield (EW, ear weight). The 4903 SNPs were selected from a total of 14.8 million SNPs generated from whole-genome resequencing of 1458 inbred lines that were used for constructing the F_1 hybrid population. The example dataset are freely available at https://github.com/YuetongXU/CropGBM_Tutorial-data.

2.1 Hardware Configuration of the Server

There is no specific hardware requirement for running CropGBM programs. The choice of hardware is determined according to the sample size and number of SNPs. The configuration of the server used for developing CropGBM is as follows.

CPU	Xeon CPU (E5–2665 2.40GHz, 8 Cores) × 2
Memory	128 Gb
GPU	NVIDIA GeForce GTX-P8 1080 × 4

2.2 Software Environment and Preinstalled Libraries

Linux operating system: Ubuntu 16.04.5 LTS, 18.04.2 LTS, is recommended.

CropGBM: A toolbox that is a one-stop solution for genomic selection-assisted breeding based on the gradient boosting algorithm with ensemble learning paradigm. It features a streamlined analytical pipeline with genotype processing, phenotype processing, population structure analysis, SNP screening, genotype-to-phenotype (G2P) prediction, data visualization, and other functional modules. Additionally, it includes t-SNE and OPTICS to analyze and visualize population structure. The kernel package used in CropGBM is the "LightGBM" package to perform regression and classification problem. Installation instruction of CropGBM may be accessed via https://ibreeding.github.io/.

Plink1.9: PLINK is required to perform a set of large-scale analyses in a computationally efficient manner, to process the genotype and phenotype data. The tool of Plink1.9 may be obtained from the download address http://www.cog-genomics.org/plink/1.9/.

GPU: CropGBM supports GPU acceleration to speed up the process of model training with three to five times increase. If your server is equipped with a GPU card, you may want to check the ability of the GPU in terms of scientific computing via https://developer.nvidia.com/cuda-gpus. This website may return the GPU type, and help the users select the appropriate compute unified device architecture (CUDA) toolkit to provide the necessary environment for GPU-accelerated computing. The CUDA download address is https://developer.nvidia.com/cuda-downloads.

3 Analytical Procedure of Running CropGBM

CropGBM supports two modes to configure parameters prior to running the program, using configuration files to perform streamlined analysis or direct command lines to add parameters. The configuration file documents the parameters and the corresponding values in one single file, so that it is convenient for users to uniformly manage and reuse a large number of parameters.

3.1 Preprocessing of Genotype and Phenotype Data

CropGBM calls the software Plink1.9 for preprocessing of genotypic data. To ensure a high quality of SNP set, the following steps need to be done, including screening of incomplete samples and SNP sites according to deletion rate and MAFs, imputation of missing genotypes based on high-frequency genotypes, and removal of redundant SNPs based on linkage. Most importantly, CropGBM converts the character-based genotypes (such as AA, AC, CC) to digit-based genotypes (0, 1, 2). Additionally, CropGBM also supports statistical analysis of genotype data such as missing rate, heterozygosity rate, and MAF distribution, so that the users may examine the quality of the genotype data. Moreover, system-level phenotype variation may exist to influence G2P prediction, which is caused by population stratification. Thus, for certain situations, preprocessing of phenotype data is also required. CropGBM utilizes the Z-score algorithm to remove systematic bias of phenotype across different populations. The detailed usage is shown below.

3.2 Genotype Data Preprocessing

3.2.1 Filter and View Overall Data

```
$ cropgbm -c configfile.params -o cropgbm_result/ -pg all --fileprefix genofile --fileformat bed --snpmaxmiss 0.10 --samplemaxmiss 0.10 --maf_max 0.05 --r2_cutoff 0.7
```

This command filters the data based on the SNP miss rate (*--snpmaxmiss 0.10*), the sample miss rate (*--samplemaxmiss 0.10*), the minimum allele frequency (*--maf_max 0.05*), and r ^ 2 (*–r2_cutoff 0.7*) to generate files in BED and PED formats and distribution histograms showing the overall situation of the data (Fig. 1).

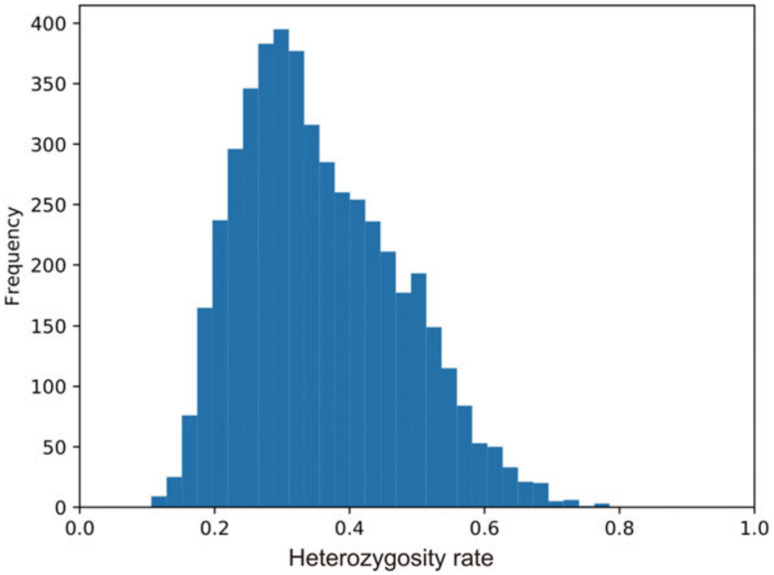

Fig. 1 Distribution of heterozygosity rate

3.2.2 Extract and Remove Specific ID of Samples and SNP Data

```
$ cropgbm -c configfile.params -o cropgbm_result/ -pg filter --fileprefix genofile --keep-sampleid-path ksampleid_file.txt --extract-snpid-path ksnpid_file.txt
$ cropgbm -c configfile.params -o cropgbm_result/ -pg filter --fileprefix genofile --remove-sampleid-path rsampleid_file.txt --exclude-snpid-path rsnpid_file.txt
```

This command extracts and removes the data of specific samples according to the sample ID by --keep-sampleid-path or --remove-sampleid-path, and the data of a specific SNP in all samples according to the SNP ID by --extract-snpid-path or --exclude-snpid-path.

3.3 Conversion of Character-Based Genotypes to Digits

```
$ cropgbm -c configfile.params -o cropgbm_result/ -pg filter --fileprefix genofile --recode --remove-sampleid-path rsampleid_file.txt --exclude-snpid-path rsnpid_file.txt
```

The parameter --recode indicates that the recoding operation is performed on the genotype data. The change rule is 00-> 0, 01-> 1, 10-> 1, 11-> 2. The genofile_filter.geno in the output file is converted genotype file.

3.4 Running the CropGBM with Command-Line Parameters as a Pipeline

```
$ cropgbm -c configfile.params -o cropgbm_result/ -pg all --fileprefix genofile --fileformat ped --snpmaxmiss 0.10 --snpmax-miss 0.10 --maf_max 0.05 --r2_cutoff 0.7 --recode --keep-sampleid-path ksampleid_file.txt --extract-snpid-path ksnpid_file.txt
```

Except for the -c and –pg parameters, all of the other parameters can be set through the configuration file and omitted on the command line.

4 Phenotype Data Preprocessing

4.1 Extract and Visualize Phenotype Data

```
$ cropgbm -c configfile.params -o cropgbm_result/ -pp --phe-plot
--phefile-path phefile.txt --phefile-header --phefile-sep
```

The phenotype file contains at least two columns of data. By default, CropGBM treats the first column as the sample ID and the second column as the phenotype data which can be visualized as a histogram distribution (Fig. 2).

4.2 Extract and Visualize the Phenotype Data According to Sample ID

```
$ cropgbm -c configfile.params -o cropgbm_result/ -pp --phe-plot
--phefile-path   phefile.txt   --ppgroupfile-path   phefile.txt   --
ppgroupfile-sep ’,’ --ppgroupid-column 3 --ppgroupfile-header
```

This command generates a phefile_scatter.pdf file to facilitate examining whether population stratification of phenotype occurs in different groups of samples (Fig. 3).

4.3 Normalization of Phenotype Data with Z-Score

```
$  cropgbm  -c  configfile.params  -o  cropgbm_result/  -pp  -phe-
plot  -phefile-path  phefile.txt  -ppgroupfile-path  phefile.txt  -
ppgroupfile-sep  ‘,’  -ppgroupid-column  3  -ppgroupfile-header  -
phe_norm -norm-mode z-score
```

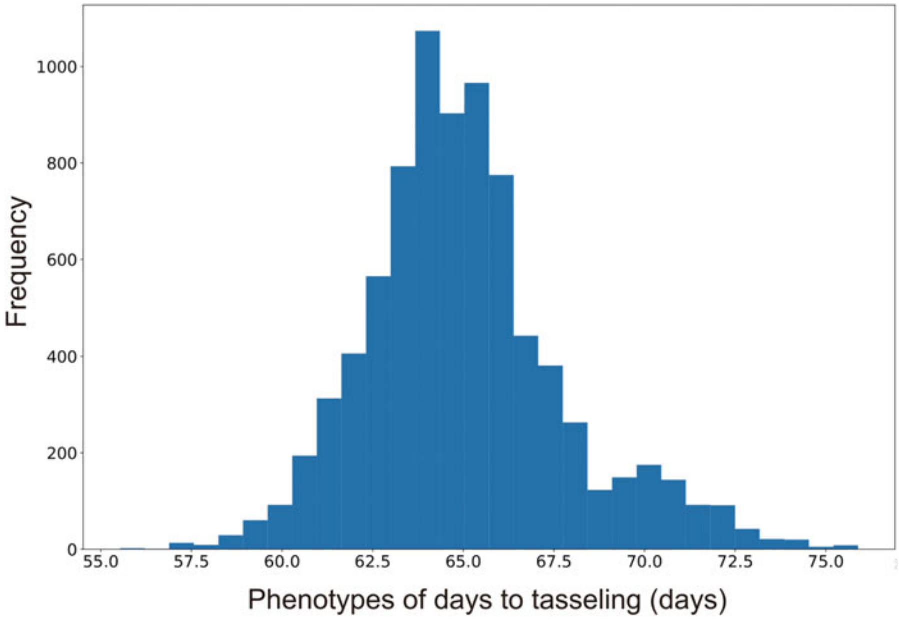

Fig. 2 Phenotype distribution represented by histogram plot

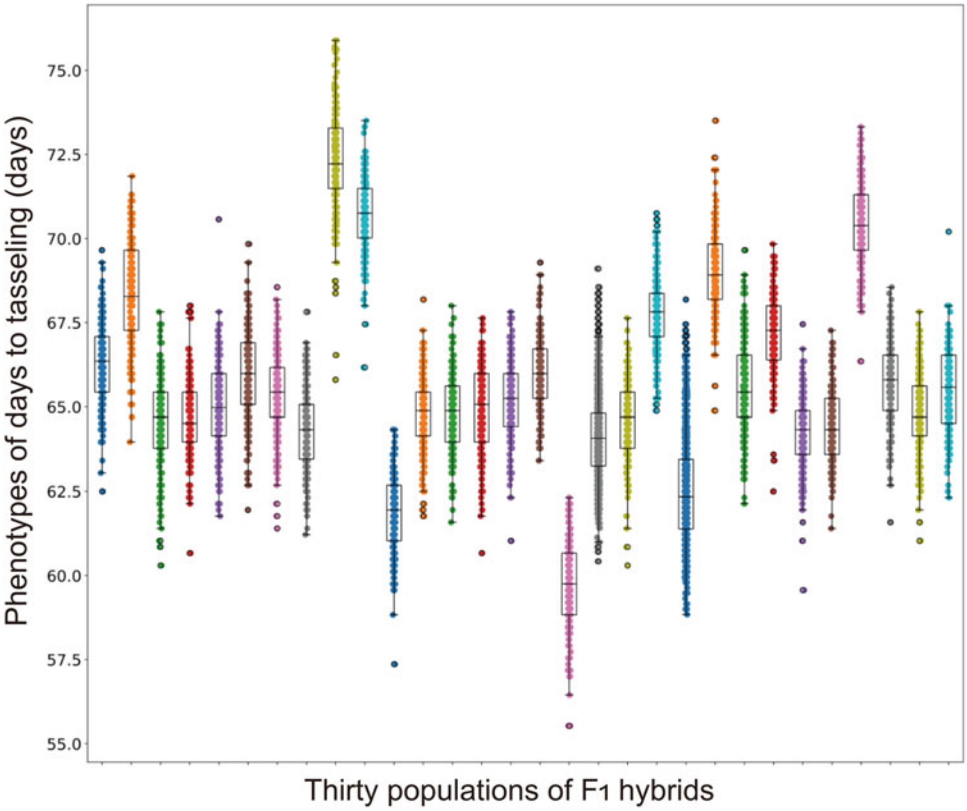

Fig. 3 Phenotype distribution of 30 populations of F_1 hybrids before normalization

This command performs Z-score normalization on phenotype data to remove between-group stratification (*--norm-mode z-score*) (Fig. 4).

4.4 Conversion of Phenotype Data

```
# recode phenotype data into continuous non-negative integer
$ cropgbm -c configfile.params -o cropgbm_result/ -pp -phe-recode word2num -phefile-path phefile.txt
```

This command converts phenotype data into continuous non-negative integers. The parameter *--phe-recode* specifies the recoding operation on the phenotype data. The optional value is [word2-num, num2word]. Word2num means conversion of phenotype data into continuous nonnegative integer form, and num2word means reconversion of continuous nonnegative integer form into phenotype data. This requires a conversion table corresponding to integers and phenotype. When performing classification tasks, LightGBM only accepts consecutive integers with example labels [0, N]. If the training samples are from 5 groups, [0, 1, 2, 3, 4] is needed as the label of the five groups, but this usually does not match the group name. Using this parameter, the program can

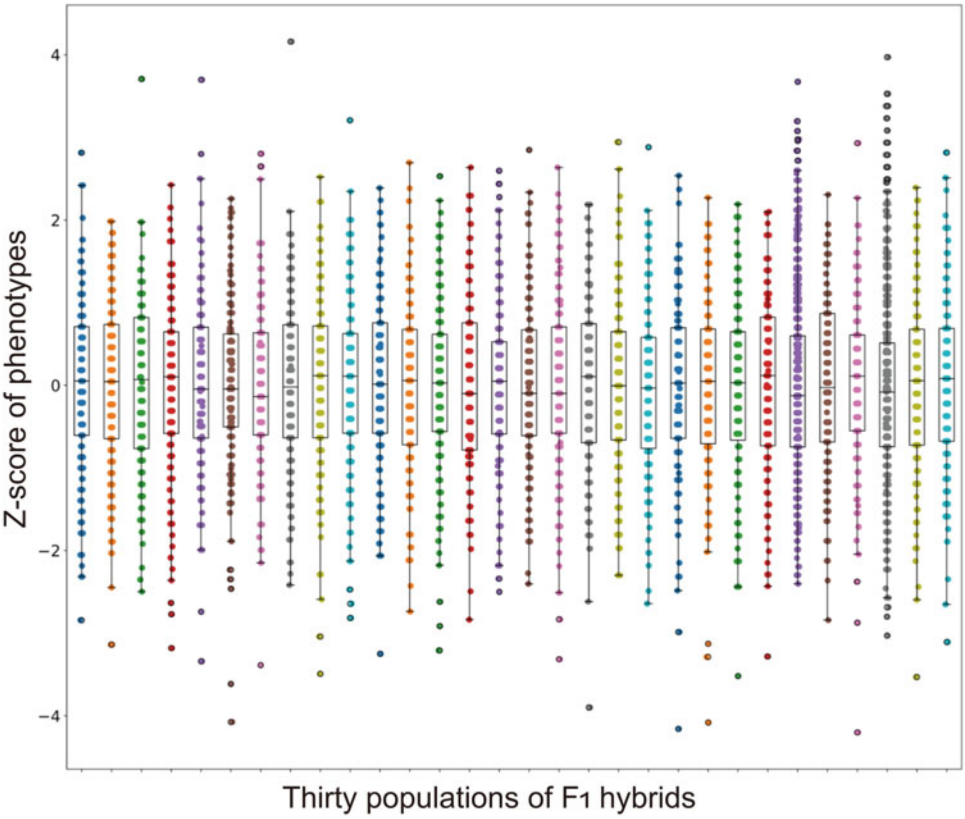

Fig. 4 Phenotype distribution of 30 populations of F_1 hybrids after Z-score normalization

implement a reversible conversion between sample labels and [0, N] consecutive integers, providing compatible phenotype data for downstream classification tasks. The phefile.word2num in the output file is a correspondence file between phenotype data and continuous nonnegative integer.

5 Population Structure Analysis

As population stratification may cause model overfitting, it is important to understand population structure and the genetic composition of the samples before constructing the model using CropGBM. Then, the training and testing dataset may be appropriately partitioned to avoid overfitting. In addition to the commonly used PCA (principal component analysis) and K-means clustering algorithm, CropGBM also integrates nonlinear algorithms—t-SNE and OPTICS—to visualize the population structure.

5.1 PCA and K-Means Clustering Analysis and Visualization of Population Structure

```
$ cropgbm -c configfile.params -o cropgbm_result/ -s --genofile-path filename_filter.geno --structure_plot --redim-mode pca --cluster-mode kmeans --n-clusters 30
```

This command clusters genotype data based on user-specified dimensionality reduction (*--redim-mode pca*) and clustering (*--cluster-mode kmeans*) algorithms. The output files are filename.cluster, filename_redim.pdf, and filename_cluster.pdf. The filename.cluster file is the clustering result. The filename_redim.pdf displays the dimensionality reduction results in the form of a scatter plot (Fig. 5). The filename_cluster.pdf displays the clustering results in the form of a scatter plot, with different categories indicated by different colors (Fig. 6).

5.2 t-SNE and OPTICS Clustering Analysis and Visualization

```
$ cropgbm -c configfile.params -o cropgbm_result/ -s --genofile-path filename_filter.geno --structure_plot --redim-mode tsne --window-size 5 --cluster-mode optics
```

The filename_reachability.pdf in the output file displays the reachable distance between each sample in the form of a scatter plot, which is output only when the *--redim-mode* value is t-SNE

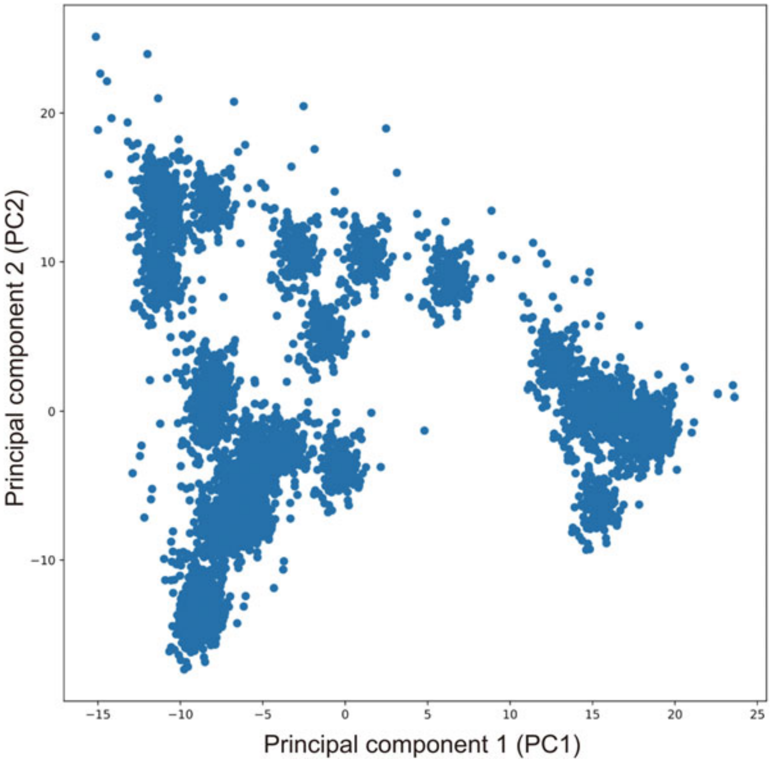

Fig. 5 PCA representation of the 30 populations indicates strong population stratification

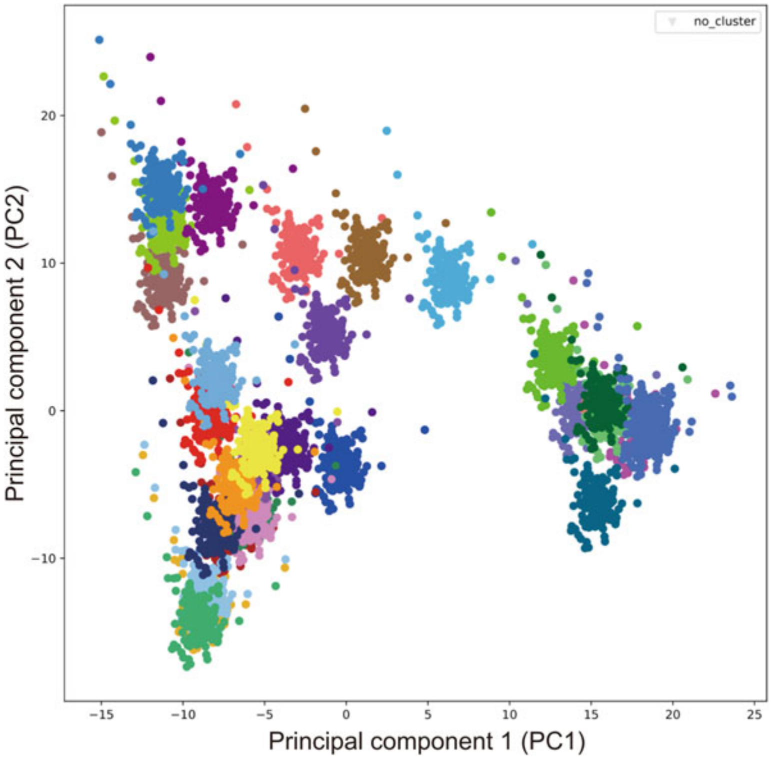

Fig. 6 PCA representation of the 30 populations. Samples in the same population are colored by the same colors

(Fig. 7). Different categories are represented by different colors, and discrete points are represented by black dots. At the same time, a plot of population structure constructed by the t-SNE algorithm is also generated with different categories represented by different colors (Fig. 8).

6 Constructing Genomic Selection GS Model with Training Samples

Construction of the GS model is the core functionality of CropGBM. Similar to other ML algorithms, the quality of the parameters is essential to the precision and performance of the predictive model to ensure the robustness and precision of the model. To avoid problematic overfitting, multiple validation sets are highly recommended to derive the most optimal parameters with the cross-validation analysis.

6.1 Cross-Validation Analysis

```
$ cropgbm -c configfile.params -o cropgbm_result/ -e -cv --traingeno train.geno --trainphe train.phe --cv-nfold 5 --min-detal 0.5
```

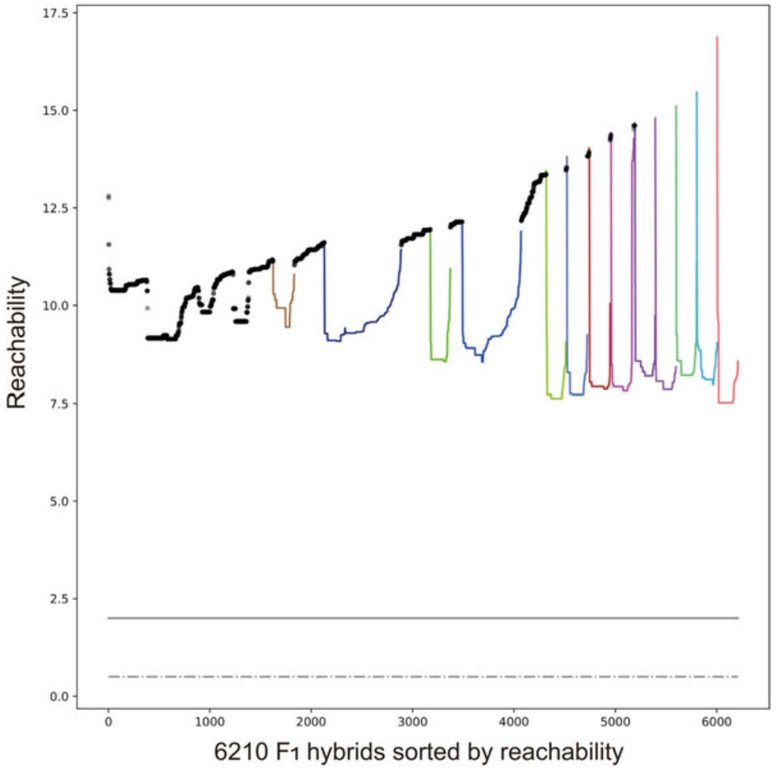

Fig. 7 OPTICS plot of the 6210 F_1 hybrids to illustrate the genetic similarity among samples

This command will estimate the accuracy of the model under different iterations. The accuracy is measured by mean square error (MSE) between the prediction result and the actual phenotype value. It's worth noting that, when the difference between training and testing set is considerably large, model precision may not be objective and it is necessary to sacrifice precision to ensure model robustness.

6.2 Training Model

```
$ cropgbm -c configfile.params -o cropgbm_result/ -e -t --train-geno train.phe --trainphe train.phe --validgeno valid.geno --validphe valid.phe
```

This command will estimate the prediction accuracy of the model on the validation set at different iterations. The output model file is train.lgb_model, in which model structure and parameters are documented for the subsequent prediction of the phenotypes of testing samples. If a validation set is not provided, the *--validgeno* and *--validphe* parameters can be omitted. It is important to note that CropGBM only recognizes different SNPs by index, not by column name. If the SNP of the same column is different

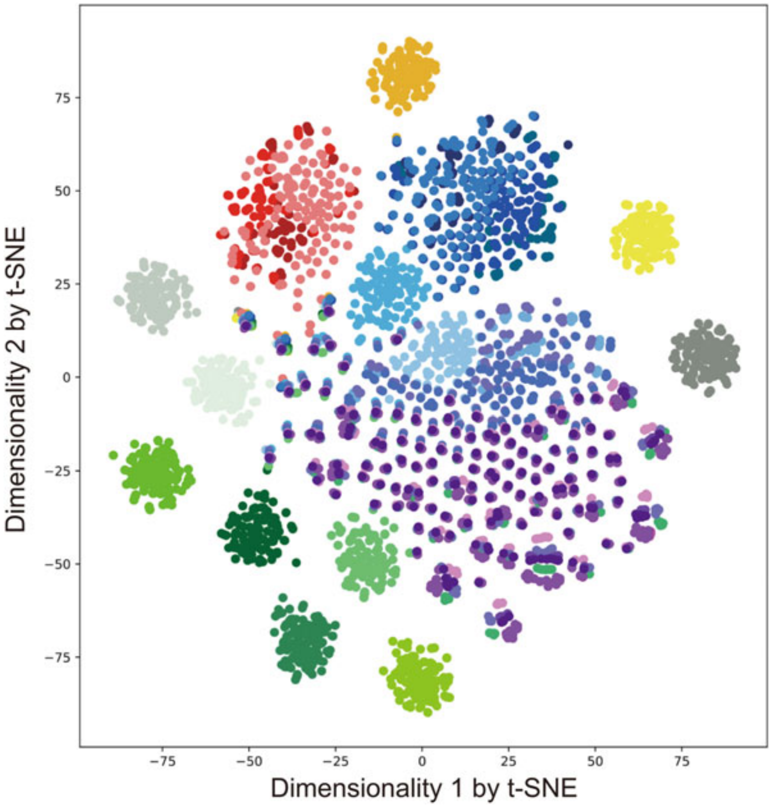

Fig. 8 t-SNE plot to illustrate the population stratification of the 6210 F_1 hybrids

between the two datasets, the program cannot recognize it. Therefore, the training set, testing set, and validation set must be consistent in terms of using the same SNP index; otherwise, the prediction result is non-reference.

7 Feature Selection Functionality

Similar to other ML algorithms, CropGBM may perform feature selection analysis during the training process, so that features with high predictive effectiveness may be automatically identified. CropGBM derives a so-called parameter of information gain (IG) to rank the predictive effectiveness for each feature. As a matter of fact, the higher the IG, the higher the probability of the genotype being associated with the trait. Therefore, feature selection in CropGBM is similar to the GWAS analysis, which can be used to identify SNPs with significant association to a target trait. This means that a large number of nonrelevant SNPs may be removed and a highly condensed marker panel may be constructed according to the feature selection functionality in CropGBM.

```
$ cropgbm -c configfile.params -o cropgbm_result/ -e -t -sf
--bygain-boxplot --traingeno train.geno --trainphe train.phe
--gain-min 0.05 --colorbar-max 0.6 --cv-times 5
```

This command will generate five output files, including train.lgb_model, train.feature, train_bygain.pdf, train_random.pdf, and train_heatmap.pdf. The train.lgb_model is the model file. The tree structure of each training and the gain value of each node are recorded. The train.feature contains the selected feature with high information gain. The information gain value of each SNP in each decision tree is recorded. The train_bygain.pdf shows the variation of the model error with the addition of SNP in the form of a scatter plot. The program repeated the fivefold cross-validation on the training set using the added SNP (Fig. 9). The x-axis coordinate in the figure is the new SNP ID added in the model, and the order is added according to the featureGain_sum value of the SNP in the train.feature file from largest to smallest. The y-axis coordinate is the prediction error. The train_random.pdf shows the change of

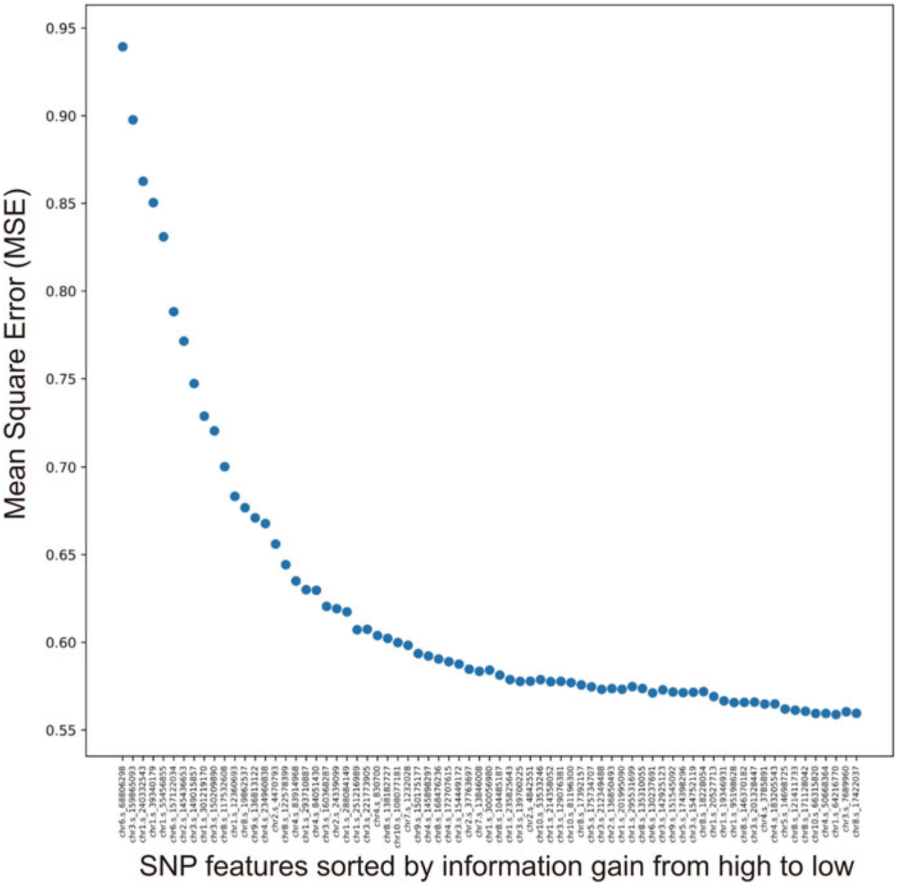

Fig. 9 SNP features sorted by feature importance based on the value of information gain inferred by CropGBM

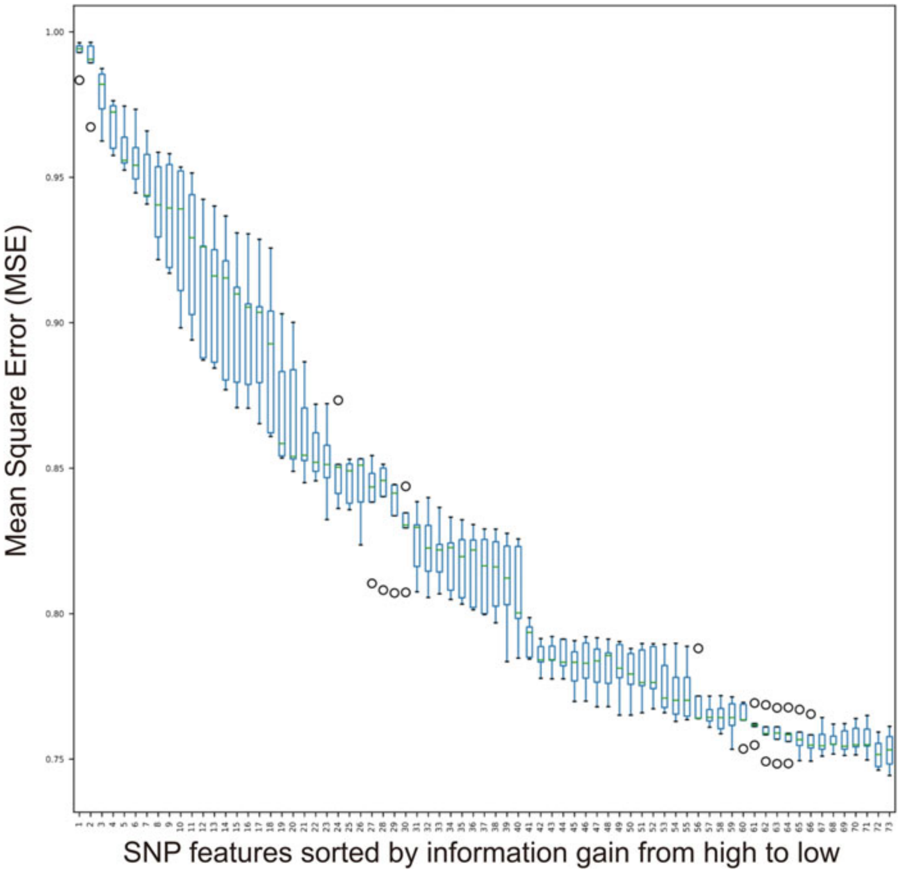

Fig. 10 Randomized SNP features sorted by feature importance based on the value of information gain inferred by CropGBM

model error with the addition of SNP in the form of boxplots (Fig. 10). The x-axis in the figure is the number of SNPs used by the model. Since the SNP used in each cross-validation is randomly extracted from all SNP, there is no SNP ID. The y-axis is the prediction error. The train_heatmap.pdf files shows the gain value and change regularity of each SNP in different decision tree in the form of a heat map, that is, the information in the train.feature file (Fig. 11). The x-axis coordinate in the figure is SNP ID, which is arranged in descending order from the featureGain_sum value of the SNP in the train.feature file; the y-axis coordinate is the index of the decision tree.

Additionally, CropGBM may use different sets of SNPs during the modeling process, especially when the SNP set is excessively redundant. Thus, SNPs significantly associated with phenotypes may not be accurately identified. Therefore, it is recommended that users select no more than 10,000 SNPs for modeling to ensure the accuracy of the feature selection functionality.

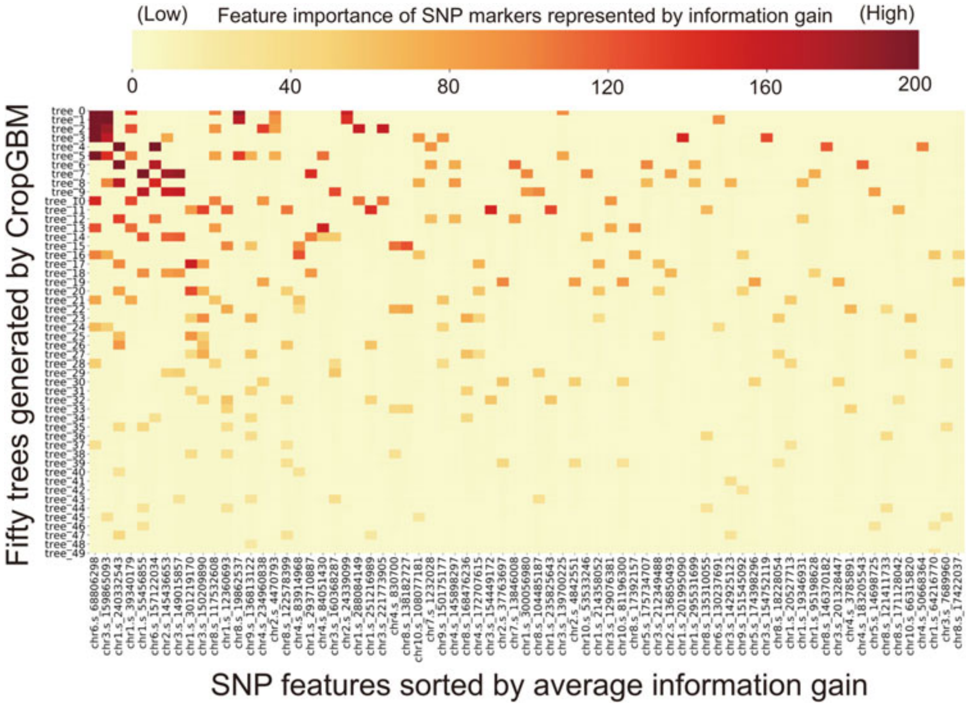

Fig. 11 Heat map of importance SNP markers selected by CropGBM. The SNP markers are sorted by feature importance based on information gain inferred by CropGBM

7.1 Prediction of the Phenotypes of Testing Samples

After the model is well trained by CropGBM, the final step is to perform phenotype prediction with genotypes as input from the testing population. As long as the prediction step is accomplished, the entire pipeline of CropGBM is finished. The trained model may be repeatedly used for prediction, and the result file is recorded in train.predict:

```
$ cropgbm -c configfile.params -o cropgbm_result/ -e -p --
testgeno test.geno --modelfile-path train.lgb_model
```

8 Concluding Remarks

In this chapter, we demonstrate the basic functionality and utility of CropGBM with an example dataset generated from a maize breeding program to illustrate the power of machine learning to perform genomic selection-assisted breeding. According to our benchmark testing, the most advantageous merit of CropGBM is ultra-high efficiency in terms of model training compared to rrBLUP and gBLUP. For example, rrBLUP took over 17 hours and 116 Gb memory to finish model training. In comparison, CropGBM only took 8 minutes and 20 Gb memory on the same server. When the sample size increased to 100,000 samples, rrBLUP failed to train the model. On the same dataset, CropGBM only used 15 minutes and 40 Gb memory with only CPU computing used. If GPU acceleration is enabled, this procedure may be reduced to only 4 min. With the rapid advance of genotyping technology and drone-carried machine vision systems to automatically collect crop traits, phenotypes will not be the limit to traditional agronomic traits, but expansion to include physiological traits captured by hyperspectral cameras will be possible [13]. Therefore, it is foreseeable that breeding data may explode in the near future forming a Big Data environment for the seed industry. Thus, the ultra-high efficient CropGBM toolbox will be an important, one-stop solution to construct data-driven decision-making models for crop breeding.

In addition, gBLUP or rrBLUP derives the linear, additive effect of alleles in a binary fashion (biallelic), namely, presence or absence corresponding to 0 or 1. They also require converting the genotypes into 0, 1, and 2 for regression analysis. However, when the number of alleles exceeds two, the biallelic effect may not be properly inferred by a linear model when the SNP is in a non-biallelic status. When the crop species features a polyploidy genome such as wheat, the situation becomes even more complicated. If the prediction is only carried on biallelic SNPs, a large amount of genetic information may be lost. Thus, one critical feature of CropGBM is that it may use one-hot coding scheme to represent genotypes rather than only using 0, 1, 2 to represent the frequency of bi-alleles in the population. This is likely another important reason why the precision of CropGBM is slightly higher than rrBLUP for a certain circumstance.

Moreover, the feature selection functionality is another advantageous merit of CropGBM compared to rrBLUP, which may be used to identify trait-associated SNPs. By this means, the initial SNP set containing tens of thousands of markers can be significantly condensed to a small marker panel containing less than 100 markers. Therefore, this genotyping platform with the ability for multiplexing samples may be used to further reduce genotyping cost. Our analysis proved that the SNPs selected by CropGBM with high predictive effectiveness are mostly located within the QTL of functionally known genes previously identified by GWAS or QTL mapping. Thus, CropGBM offers an avenue for gene discovery and identification of important genetic variation in breeding programs. In summary, the successful application of CropGBM in genomic selection indicated that machine learning facilitated with artificial intelligence is a promising technology to facilitate Big Data-driven breeding.

Acknowledgments

This work was supported by the National Science Foundation of China (31871706): Utilization of Machine Learning Strategy to Predict Maize Heterosis.

References

1. Lande R, Thompson R (1990) Efficiency of marker-assisted selection in the improvement of quantitative traits. Genetics 124 (3):743–756
2. Bernardo R (2008) Molecular markers and selection for complex traits in plants: learning from the last 20 years. Crop Sci 48:1649–1664. https://doi.org/10.2135/cropsci2008.03.0131
3. Xu Y, Crouch J (2008) Marker-assisted selection in plant breeding: from publications to practice. Crop Sci 48:391–407. https://doi.org/10.2135/cropsci2007.04.0191
4. Heffner E, Sorrells M, Jannink J-L (2009) Genomic selection for crop improvement. Crop Sci 49:1–12. https://doi.org/10.2135/cropsci2008.08.0512
5. Meuwissen THE, Hayes BJB, Goddard MEM (2001) Prediction of total genetic value using genome-wide dense marker maps. Genetics 157:1819–1829
6. Xu S, Zhu D, Zhang Q (2014) Predicting hybrid performance in rice using genomic best linear unbiased prediction. Proc Natl Acad Sci U S A 111:12456–12461. https://doi.org/10.1073/pnas.1413750111
7. Kadam DC, Potts SM, Bohn MO, Lipka AE, Lorenz AJ (2016) Genomic prediction of single crosses in the early stages of a maize hybrid breeding pipeline. G3 (Bethesda, Md) 6 (11):3443–3453. https://doi.org/10.1534/g3.116.031286
8. Henderson CR (1984) Applications of linear models in animal breeding. University of Guelph, Guelph
9. Bernardo R, Yu J (2007) Prospects for genomewide selection for quantitative traits in maize. Crop Sci 47:1082–1090. https://doi.org/10.2135/cropsci2006.11.0690
10. Maros M, Capper D, Jones D, Hovestadt V, Deimling A, Pfister S, Benner A, Zucknick M, Sill M (2020) Machine learning workflows to estimate class probabilities for precision cancer diagnostics on DNA methylation microarray data. Nat Protoc 15:479–512. https://doi.org/10.1038/s41596-019-0251-6
11. Dorman SN, Baranova K, Knoll JHM, Urquhart Brad L, Mariani G, Carcangiu ML, Rogan Peter K (2016) Genomic signatures for paclitaxel and gemcitabine resistance in breast cancer derived by machine learning. Mol Oncol 10 (1):85–100. https://doi.org/10.1016/j.molonc.2015.07.006
12. Qiu Z, Cheng Q, Song J, Tang Y, Ma C (2016) Application of machine learning-based classification to genomic selection and performance improvement, vol 9771. doi:https://doi.org/10.1007/978-3-319-42291-6_41
13. Shakoor N, Lee S, Mockler T (2017) High throughput phenotyping to accelerate crop breeding and monitoring of diseases in the field. Curr Opin Plant Biol 38:184–192. https://doi.org/10.1016/j.pbi.2017.05.006