MSc in Bioinformatics

Genomic Prediction of Complex Traits in Rice: Leveraging Structural Variation and Applying Deep Learning Methods

Alejandro Navarro Martínez

Research tutor:

Miguel Pérez Enciso

Centre de Recerca en Agrigenòmica (CRAG) CSIC-IRTA-UAB-UB

Academic tutor:

Raquel Egea Sánchez

Universitat Autònoma de Barcelona (UAB)

September 6th, 2022

Genomic Prediction of Complex Traits in Rice: Leveraging Structural Variation and Applying Deep Learning Methods

Abstract

Prediction of complex traits is challenging for plant breeders due to the high number of quantitative trait loci (QTLs) affecting them and because of the relevance of non-additive genetic effects for some traits. Therefore, it is important to fine-tune this process as much as possible. We present a case study of four traits in *Oryza sativa*, carried out with a dataset of 738 accessions representing the whole genetic diversity of the species. We address the problem of the high number of QTLs by introducing structural variants (SVs) as an additional source of genetic variation with respect to single-nucleotide polymorphisms (SNPs). The SVs dataset includes transposon insertion polymorphisms (TIPs), deletions, tandem duplications and inversions. The problem of the non-additive genetic effects is tackled by implementing deep learning methods, which are based on artificial neural network (ANN) models. These models can implement complex non-linear transformations of an input, so they should be able to capture the effects of dominance and epistasis. Results show that SVs can substantially improve the prediction of certain traits, as seen by the 7.40% improvement for leaf senescence. Deep learning methods proved to be competitive with linear models for regression tasks. They also performed well for classification, in which the linear models performed sub-optimally.

Table of Contents

List of Figures

List of Tables

Abbreviations

- ADM Admixed rice
- ANN Artificial neural network
- ARO Aromatic rice
- AUC Area under the curve
- AUS Aus rice
- CNN Convolutional neural network
- CNV Copy-number variant
- Cor Correlation
- CV Cross-validation
- DEL Deletion
- DL Deep learning
- DTX Terminal inverted repeat transposons
- DUP Tandem duplication
- GP Genomic prediction
- IND Indica rice
- INS Insertion
- INV Inversion
- JAP Japonica rice
- LD Linkage disequilibrium
- MAF Minor allele frequency
- MAS Marker-assisted selection
- MITE Miniature inverted-repeat transposable elements
- MLP Multi-layer perceptron
- MSE Mean-squared error
- PCA Principal component analysis
- QTL Quantitative trait locus
- RIX / LINE Long interspersed nuclear elements
- RKHS Reproducing Kernel Hilbert Spaces
- RLX Long terminal repeat retrotransposons
- SNP Single-nucleotide polymorphism
- SV Structural variant
- TIP Transposon insertion polymorphism

1. Introduction

1.1. Complex Traits and Genomic Prediction

Complex traits are phenotypes controlled by many genes and environmental factors; they are usually quantitative traits. Prediction of these traits is of vital importance for breeding, as it is a way to assess the performance of future individuals. The current paradigm revolves around genomic prediction (GP) (Meuwissen et al., 2001), which consists in employing genome-wide molecular marker data to predict a given complex trait. This approach overcomes the limitations of marker-assisted selection (MAS) (van-Arendonk et al., 1994), which only uses significantly associated markers. Association tests can only detect markers linked to quantitative trait loci (QTLs) with large effects, and thus the resulting marker set can fail to explain an important fraction of the real genetic variance of the trait: the problem known as "missing heritability" (Maher, 2008; de los Campos et al., 2015). In GP, the algorithms derive implicitly which are the most associated markers. However, this strategy results in models with a large number of parameters (*p*, due to the number of markers) and few observations (*n*, genotyped individuals) by comparison: the large-*p* small-*n* problem (de los Campos et al., 2013). Methods address this problem by implementing shrinkage of estimates, variable selection or a combination of both.

1.2. Deep Learning for Genomic Prediction

In recent years, deep learning (DL) methods have been considered for GP. The rationale is that linear model methods, unless specifically parametrized, only account for additive genetic effects. Nevertheless, complex traits can depend on non-additive genetic effects based on interactions between alleles: dominance and epistasis. DL models are artificial neural networks (ANNs) of a certain depth. ANNs are multilayer stacks of simple modules which compute non-linear transformations, and thus can implement complex functions of the inputs which account for these non-additive genetic effects (LeCun et al., 2015). Several GP studies implementing DL have been published (Montesinos-López et al., 2021), and results so far show that DL outperforms linear model methods for certain instances, but not always. There are different reasons by which DL can have a lower performance: not all traits have a non-additive genetic component, not all datasets are large enough to fit models of such complexity, and implementation may not be optimal (Montesinos-López et al., 2021). Implementation is quite intricate, as there are different architectures to choose from, like multi-layer perceptrons (MLPs) or convolutional neural networks (CNNs); there are many hyperparameters to tune, and the search for the optimal ones can be more or less exhaustive; and the experiment's design in terms of training, tuning and testing has to be done trying not to introduce any bias for the model's evaluation.

1.3. Leveraging Structural Variation

GP is based on building genome-wide dense molecular marker maps and exploiting the linkage disequilibrium (LD) between these markers and the QTLs. The denser the maps, the better representation of the genome's haplotype blocks there will be. It is for this reason that including additional sources of genetic variation (in addition to SNPs), like structural variants (SVs), can improve GP. This is especially true for transposon insertion polymorphisms (TIPs), which are ubiquitous in many plant genomes, corresponding to about 40% of the rice genome (our case study) (Jiang and Panaud, 2013). Different studies have evaluated the degree of

redundancy of this TIP data with that of SNPs, by measuring LD of TIPs with nearby SNPs. Results have shown low LD between TIPs and SNPs in *Arabidopsis*, tomato and rice (Stuart et al., 2016; Domínguez et al., 2020; Castanera et al., 2021); demonstrating the potential of this data to explain phenotypic variation previously unaccounted for. In fact, the tomato and rice studies (Domínguez et al., 2020; Castanera et al., 2021) found associations of TIPs with agronomic traits which could not be detected with SNPs. Vourlaki et al. (2022) went one step beyond and showed how TIPs explain an important fraction of the phenotypic variance of rice traits.

Nevertheless, this is not the only reason to include SV data, as other types of variants (like large indels and inversions), are not as widespread as TIPs across the genome. Given the size of these variants, they are more likely to either disrupt genes or cis-regulatory regions, thus affecting gene expression and consequently phenotypes. SVs also include copy-number variants (CNVs), which can affect gene expression by changing the gene dosage. Therefore, SVs are strong QTL candidates themselves, and do not have a value only as markers. There are documented cases of SVs being responsible for plant phenotypes (Gabur et al., 2019). Focusing on rice, we have examples like a CNV regulating grain length (Wang et al., 2015) and a small indel affecting root system architecture (Uga et al., 2013). TIPs are interesting in this regard, because there is evidence that transposition mechanisms are triggered by environmental stresses (Grandbastien, 2015; Carpentier et al., 2019), like the ones associated with domestication. Hence, TIPs are likely to be responsible for some of the trait changes occurred during domestication. In addition, because of the ability of DNA transposons to be excised, QTLs of this type would be difficult to track down with molecular markers.

1.4. Rice Breeding

Rice (*Oryza sativa*) is a staple food for many people across the world, so it is no surprise all the effort that has been put into obtaining better varieties. It is a diploid species with a relatively small genome of 430 Mb, what makes it a good model species for monocotyledons. There are four main populations of rice: japonica, indica, aus and aromatic (Wang et al., 2018). Japonica and indica are the largest cultivated groups, and they originate from different domestication events. There is evidence of the aus group having been domesticated separately, being more closely related to indica than to japonica rice (Carpentier et al., 2019). The main objectives of rice breeding can be divided into improving grain yield, grain quality and resistance to stresses (Bai et al., 2018). We will focus on four traits which are related to grain yield: time to flowering, as this conditions the adaptation of rice to different environments; grain weight, for it is related to grain size; culm diameter, as a thicker culm confers resistance to lodging; and leaf senescence, as delayed senescence is associated with more productive plants.

This work follows in the footsteps of Vourlaki et al. (2022), where they ascertained that TIP data could improve GP in rice for certain traits in specific prediction scenarios. We use the same dataset focusing on four traits, and we include additional structural variation data from Fuentes et al. (2019). In this study, we also implement a 10-fold cross-validation approach to assess model performance, which will give us and idea of the effect of population structure in model evaluation. Another expansion from the original work is the implementation of DL models. In summary, we are trying to determine whether structural variation data can improve GP, and what is the best statistical modelling implementation to achieve this.

2. Objectives

1. Assess if the addition of structural variation data improves the prediction of complex traits in comparison to using only SNPs.
2. Evaluate how well deep learning performs for genomic prediction in comparison to linear model methods.
3. Compare the performance of different deep learning implementations, considering different input options of genotype data, network architectures and single or multiple output models.

3. Materials and Methods

3.1. The Dataset

3.1.1. Accessions

Rice accessions are a subset of the 3000 Rice Genomes (3K-RG) project (The 3000 rice genomes project, 2014; Wang et al., 2018). The 738 accessions used in Castanera et al. (2021) were selected so as to use their TIP genotyping data. These accessions belong to the germplasm collection of the International Rice Research Institute (IRRI) and they have been sequenced at a minimum coverage of 15x. This is the same dataset used in Vourlaki et al. (2022). Accessions are classified into the 4 main rice population groups: indica (IND, N = 451), japonica (JAP, N = 166), aus (AUS, N = 75) and aromatic (ARO, N = 17); plus an admixed group (ADM, N = 29).

3.1.2. Genotypes

SNP genotyping data was obtained from the Rice SNP-Seek database (Mansueto et al., 2017), corresponding to the CoreSNP dataset for all chromosomes. In a previous work (Vourlaki et al., 2022), markers were filtered by MAF > 1% and missing rate < 1%. Missing genotypes were imputed with Beagle 5.2 (Browning et al., 2018).

As mentioned before, TIP data was obtained from Castanera et al. (2021). These variants were genotyped as absence/presence (0/1 coded). TIPs were filtered by MAF > 1% and organized into 2 groups to simplify posterior analyses: RLX-RIX and MITE-DTX. RLX-RIX is composed of retrotransposons (class I transposons), and comprises long terminal repeat (LTR) retrotransposons (RLX) and long interspersed nuclear elements (LINE; RIX). MITE-DTX is composed of DNA transposons (class II transposons), and comprises miniature inverted-repeat transposable elements (MITE) and other terminal inverted repeat (TIR) transposons (DTX).

Additional structural variation data was obtained from the Rice SNP-Seek database, corresponding to the work of Fuentes et al. (2019). There was data available for CNVs, insertions (INS), deletions (DEL), tandem duplications (DUP) and inversions (INV). Minimum size was 5 bp for INS, while it was 10 bp for DEL, DUP and INV. CNVs were discarded because there was data for only 661 accessions, and the missing rate per variant was too high (58% on average). INS were discarded too because they were available for just 390 accessions. DEL, DUP and INV were kept. All structural variants (SVs) were considered as absence/presence variants and filtered by MAF > 1%.

Genotypes from all sources were called against the Nipponbare IRGSP-1.0 reference genome assembly. The final number of markers for each of the marker sets is recorded on Table 1.

Table 1. Count for each of the marker sets.

Marker set	No. of markers
SNP	228,871
MITE-DTX	52,120
RLX-RIX	21,571
DEL	139,229
DUP	14,638
INV	6,083

3.1.3. Traits

Phenotypic data for 11 traits was procured from the Rice SNP-Seek database. Traits were selected to maximize data availability for the 738 accessions. After the genomic variance inference performed in section 3.3, and considering results from Vourlaki et al. (2022), 4 traits were selected for genomic prediction: time to flowering from sowing, grain weight, culm diameter of 1st internode and leaf senescence. A summary of the characteristics of these 4 traits is available on Table 2. Time to flowering was log transformed and categories of leaf senescence were binned to balance the number of observations per class. Similar transformations were applied for the rest of traits, as described in Vourlaki et al. (2022).

Table 2. Summary of the traits used for genomic prediction.

Variable type	Recoding	Units	Measurement Method	Description	Crop Ontology ID	Trait
Continuous	Log transformation	Days	The difference between the main heading date (80% flowering count) and the date of effective seeding	Time elapsed from date of sowing to flowering stage	CO_320:0000710	Time to flowering from sowing
Continuous	None	Grams (g)	Take a random sample of 100 well-developed, whole grains, dried to 13% moisture content, and weigh on a precision balance	Weight of a whole grain including the hull	CO_320:0000740	Grain weight
Binary	None	category_1: 1 = Thin < 5 mm category_2: 2 = Thick >= 5 mm	The average diameter of the basal portion of the main culm on three representative plants	Diameter of the basal portion of the main culm	CO_320:0000692	Culm diameter of 1st internode
Binary	Classes {3, 5, 7, 9} recoded as {2}	cat_1: 1 = Very early (all leaves lost their green colour before grain maturity) cat_2: 3 = Early (all leaves have lost their green colour at harvest) cat_3: 5 = Intermediate (one leaf still green at harvest) cat_4: 7 = Late (two or more leaves still green at harvest) cat_5: 9 = Very late (all leaves still green at harvest)	Estimate by observing all leaves below the flag leaf for their retention of greenness when the grains are ready for harvesting	Description of how soon leaves senesce	CO_320:0001014	Leaf senescence

3.2. Exploratory Analysis

Principal Component Analysis (PCA) of each of the genotype matrices was performed to assess the possibility of population structure in the molecular marker data. Histograms of the trait variables were plotted to check the normality of continuous traits and to verify a balanced distribution of categories for the binary traits. Distributions per rice population group were also evaluated.

Correlations between trait variables were also computed and plotted with the corrplot R package (Wei and Simko, 2021). Binary variables were treated as continuous for this analysis. This approximation was deemed adequate as the categorical variables are ordinal.

3.3. Genomic Variance Inference

The fraction of phenotypic variance explained by the different marker sets was estimated using the following model:

$$\mathbf{y} = \mathbf{1}\mu + \mathbf{u}_{\mathrm{SNP}} + \mathbf{u}_{\mathrm{SV}} + \mathbf{e} \quad (1)$$

where $\mathbf{y}$ is the vector of observations of the trait, μ is the intercept, $\mathbf{u}_{\mathrm{SNP}}$ is a vector of random effects for SNPs, $\mathbf{u}_{\mathrm{SV}}$ is a vector of random effects for one of the SVs (MITE-DTX, RLX-RIX, DEL, DUP or INV) and $\mathbf{e}$ is the vector of residuals. We assume the following distributions for the random variables: $\mathbf{u}_{\mathrm{SNP}} \sim N(\mathbf{0}, \mathbf{G}_{\mathrm{SNP}}\, \sigma^2_{\mathrm{SNP}})$, $\mathbf{u}_{\mathrm{SV}} \sim N(\mathbf{0}, \mathbf{G}_{\mathrm{SV}}\, \sigma^2_{\mathrm{SV}})$ and $\mathbf{e} \sim N(\mathbf{0}, \mathbf{I}\, \sigma^2_{\mathrm{e}})$. Model (1) was fitted separately for each of the five SV types, always including SNPs.

$\mathbf{G}_{\mathrm{SNP}}$ and $\mathbf{G}_{\mathrm{SV}}$ are genomic relationship matrices computed with the AGHmatrix R package (Amadeu et al., 2016) using the VanRaden (2008) method:

$$\mathbf{G} = \frac{\mathbf{Z}\,\mathbf{Z}^{\mathrm{T}}}{2\ \sum_{j=1}^{m} p_j\,(1-p_j)} \quad (2)$$

where $\mathbf{Z}$ is the $n \times m$ (n observations and m markers) centered matrix of marker genotypes, initially coded by minor allele dosage (0, 1, 2; marker genotype matrices of SVs were recoded from 0/1 to 0/2 to comply with this). p_j is the frequency of the minor allele of the j^{th} marker. $\mathbf{G}$ is basically a scaled covariance matrix of the individuals based on the marker information.

Models were fitted with a Bayesian Reproducing Kernel Hilbert Spaces (RKHS) regression implemented with the BGLR R package (Pérez and de los Campos, 2014). An RKHS regression function is based on a reproducing kernel (RK) function, which maps pairs of points in input space into a scalar and is positive semi-definite. A kernel function can be intuitively understood as a measure of similarity. Model (1) can be conceived in an RKHS framework by considering the kernel function:

$$K(\mathbf{x}_i, \mathbf{x}_{i'}) = \frac{\mathbf{x}_i^{\mathrm{T}}\,\mathbf{x}_{i'}}{2\ \sum_{j=1}^{m} p_j\,(1-p_j)} \quad (3)$$

where $\mathbf{x}_i$ and $\mathbf{x}_{i'}$ are vectors of centered genotypes (the ones from $\mathbf{Z}$) for any two individuals. Evaluation of this function over the whole input space returns the kernel matrix, which in this

case is $\mathbf{G}$. Model parameters were assigned the following priors according to default BGLR settings: $\mu \sim N(0,\ 10^{10})$ and $\sigma^2 \sim \chi^{-2}(\mathrm{df},\ S)$ (for σ^2_{e}, σ^2_{SNP} and σ^2_{SV}) where degrees of freedom $\mathrm{df} = 5$ and S is a scaling parameter automatically computed. Posterior distributions were obtained with a Gibbs sampler algorithm run for 100,000 iterations with a burn-in of 500 rounds and thinning every 5. Variance estimates were obtained as the mean of the posteriors. Heritability-like ratios for the proportion of phenotypic variance explained by marker genomic effects were calculated as: $h^2_{\mathrm{SNP}} = \frac{\sigma^2_{\mathrm{SNP}}}{\sigma^2_{\mathrm{y}}}$ and $h^2_{\mathrm{SV}} = \frac{\sigma^2_{\mathrm{SV}}}{\sigma^2_{\mathrm{y}}}$, where $\sigma^2_{\mathrm{y}} = \sigma^2_{\mathrm{SNP}} + \sigma^2_{\mathrm{sv}} + \sigma^2_{\mathrm{e}}$. Binary traits were considered as continuous for this analysis.

3.4. Genomic Prediction Frameworks

In order to evaluate how well the GP models generalize to new data, we need to divide our dataset into a training and test set. The training set is used for feature engineering (e.g. determining the mean and standard deviation for standardization), fitting the model, tuning hyperparameters and comparing models; while the test set is just used to obtain an evaluation of the final model's performance. Employing the test set for anything else would lead to an overfitting of the model and a too optimistic estimation of the generalization error.

In this regard, we proposed two frameworks to evaluate model performance:

First, a simple train/test split where the test set is composed by the rice accessions corresponding to the aromatic and admixed groups (N = 46). As our dataset presents population structure regarding the rice groups, evaluating the model on these two groups is interesting because ADM accessions are genetically heterogeneous as they are crosses of plants from other groups, and ARO accessions, although they are similar to each other, they are in-between the other rice groups (Figure 5A). In this way, accessions used for training are not very related to the ones used for testing, so the evaluation could give us a good estimate of generalization error. In a breeding context, this can be seen as an "across populations" prediction scenario. This scenario was labelled as "AroAdm", for short.

And second, a 10-fold cross-validation (CV) scenario where the dataset was partitioned into 10 subsets, and each subset was used as test set in turn. This approach was chosen because our dataset is small and structured, so model performance is likely to depend on the specific partition. With this methodology, as model performance is computed taking into account the whole dataset, it is less biased.

Figure 1. Diagrams of genomic prediction frameworks.
(A) data-splitting for the 10-fold cross-validation (CV) framework. **(B)** data-splitting for the AroAdm framework. TST = test set, VAL = validation set, TRN = train set. The test set of **(B)** corresponds to the aromatic and admixed rice accessions. The creation of a validation set and doing a hyperparameter search was only done for the artificial neural network models. Performance refers to the value of a given metric evaluated for a test set.

3.5. GWAS for Dimensionality Reduction

Combining all marker sets we have a total of 462,512 molecular markers. This number of variables is too high for parametric models which estimate marker effects (like Bayes C), or for ANNs, where the number of model parameters increases substantially. As most markers will not have an effect on the trait, we decided to pre-select the 10k most associated markers by means of a GWAS. This will decrease computation time for the aforementioned models and will avoid introducing noise (irrelevant data) into them. For every marker set, the following linear model was fitted to test association:

$$\mathbf{y} = \mathbf{1}\mu + \sum_{k=1}^{4} \mathbf{c}_k \boldsymbol{\beta}_k + \mathbf{x}g + \mathbf{e} \quad (4)$$

where $\mathbf{y}$ is the vector of standardized phenotypes (for the continuous ones), μ is the intercept, $\mathbf{c}_k$ is the vector of the k^{th} principal component of the $\mathbf{X}$ matrix of marker genotypes, $\boldsymbol{\beta}_k$ is the fixed effect of $\mathbf{c}_k$, $\mathbf{x}$ is the vector of standardized genotypes for the marker being tested with effect g and $\mathbf{e}$ is the vector of residuals. The model includes the first 4 principal components of the marker genotype matrix as fixed effects to correct for population stratification. For binary traits, a logistic regression was applied.

As we have 11 different training sets because of the "AroAdm" and 10-fold CV prediction frameworks, 11 independent GWAS were performed with the observations corresponding to each training set. *t*-tests were performed to assess significance of the marker effect and *p*-values were computed. New genotype matrices were created with the top 10k most associated markers and top 10k SNPs, sorting them by genomic position (important for the CNN models).

3.6. Genomic Prediction with Linear Models

Two linear model methods were implemented with BGLR: Bayesian RKHS (de los Campos et al., 2009) and Bayes C (Habier et al., 2011).

3.6.1. Bayesian RKHS

Two models were proposed to compare prediction with only SNP marker data and prediction with SNP plus SV data:

$$\mathbf{y} = \mathbf{1}\mu + \mathbf{u}_{\mathrm{SNP}} + \mathbf{e} \tag{5}$$

$$\mathbf{y} = \mathbf{1}\mu + \mathbf{u}_{\mathrm{SNP}} + \mathbf{u}_{\mathrm{MD}} + \mathbf{u}_{\mathrm{RR}} + \mathbf{u}_{\mathrm{DEL}} + \mathbf{u}_{\mathrm{DUP}} + \mathbf{u}_{\mathrm{INV}} + \mathbf{e} \tag{6}$$

where $\mathbf{u}_{\mathrm{x}}$ is a vector of marker random effects for which $\mathbf{u}_{\mathrm{x}} \sim N(\mathbf{0}, \mathbf{G}_{\mathrm{x}}\, \sigma^2_{\mathrm{x}})$ and subindex x denotes the marker set (MD = MITE-DTX, RR = RLX-RIX). Model (5) only employs SNP data and is labelled "RKHS_SNP", while (6) adds the information of all SVs and is labelled "RKHS_all". Despite of Bayesian RKHS models being able to accommodate non-linear relationships, these are linear models because the $\mathbf{G}$ matrices are covariance matrices, and thus represent the linear relationship between two elements.

3.6.2. Bayes C

The following Bayes C model was implemented:

$$\mathbf{y} = \mathbf{1}\mu + \mathbf{X}\boldsymbol{\beta} + \mathbf{e} \tag{7}$$

where $\mathbf{y}$ is the vector of phenotypes, μ is the intercept, $\mathbf{X}$ is a matrix of standardized marker genotypes with a $\boldsymbol{\beta}$ vector of effects and $\mathbf{e}$ is a vector of residuals which $\mathbf{e} \sim N(\mathbf{0}, \mathbf{I}\, \sigma^2_{\mathrm{e}})$. Bayes C induces shrinkage and selection of effects (small effects and many of them equal to 0) with the prior it assumes for them: $\boldsymbol{\beta} \sim \pi\, N(\mathbf{0},\ \mathbf{I}\sigma^2_{\beta}) + (1-\pi)\ \mathrm{DG}(0)$, a mixture of a Gaussian slab and a point of mass at 0 (degenerate distribution DG at 0). Hyperparameter π determines the probability of a marker of having an effect and is sampled from a prior $\pi \sim \mathrm{Beta}(\pi_0,\ p_0)$, a Beta distribution reparametrized so that π_0 represents the mean and p_0 a measure of dispersion. $p_0 = 5$ and $\pi_0 = 0.01$ were chosen, so we could say there is a probability of 1% of a marker entering the model. Default settings were used for everything else.

In Model "BayesC_SNP", the genotype matrix contained the 10k most associated SNPs as $\mathbf{X}$; and in model "BayesC_all", the top 10k markers, irrespective of their type, were employed. Standardization of marker genotypes was performed with the whole set of observations (train plus test). This is because in GP, the usual approach is to fit a new model every time that new data is available; instead of using a previously trained model to predict the new observations.

Therefore, in a normal GP experiment, test feature data would be available for feature engineering.

For both RKHS and Bayes C, regressions were run for binary traits with function parameter `response_type = "ordinal"`. The algorithm was run for 100,000 iterations, with burn-in of 500 and thinning of 5.

3.6.3. Performance Evaluation

To assess the performance of the trained models, different metrics were computed for the predictions on the test set. For the continuous traits, these were mean squared error (MSE) and correlation (cor). MSE:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (8)$$

where y_i is the real value of the trait and $\hat{y}_i$ is the predicted one for the i^{th} observation, and n is the number of observations on the test set. Cor:

$$\text{cor}_{y\hat{y}} = \frac{\sum_{i=1}^{n}(y_i - m_y)(\hat{y}_i - m_{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - m_y)^2}\sqrt{\sum_{i=1}^{n}(\hat{y}_i - m_{\hat{y}})^2}} \quad (9)$$

where m is the mean of vector $\mathbf{y}$ or $\hat{\mathbf{y}}$.

For the binary traits, binary cross-entropy and area under the curve (AUC) were computed. Binary cross-entropy:

$$\text{Binary crossentropy} = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)] \quad (10)$$

where y_i is the real value of the trait (coded as 0 or 1) and $\hat{y}_i$ is the predicted probability of class 1 for the i^{th} observation. AUC refers to the area under the receiver operating characteristic (ROC) curve. This curve represents the true positive rate (TPR) vs. the false positive rate (FPR). An ideal model will correctly classify all of the positive class elements (TPR = 1) and will not misclassify any of the negative class elements (FPR = 0). In this case the AUC would be equal to 1. If the model were randomly assigning the classes, the AUC would be of about 0.5.

MSE and binary cross-entropy provide a measure of how far the predicted values are from the real ones, and they are calculated directly from the output of the model, so they are robust metrics to make comparisons between models. Cor and AUC give us an idea of how good the predictions are, to assess if they are actually useful, better than random guessing.

3.7. Genomic Prediction with Artificial Neural Networks

3.7.1. Algorithm Overview

ANNs are multilayer stacks of neurons. They are composed of input layers, which introduce data into the model; hidden layers, which compute some transformation of their inputs; and output layers, which return that which we are trying to predict. Therefore, an ANN can be seen as a nested function of all the transformations happening in the hidden layers.

Input data can be *n*-dimensional, but it needs to be flattened to 1 dimension before being fed to a fully connected "dense" layer. Dense layers are the only type of hidden layer used in the most representative ANN architecture: the MLP (Figure 2). Any given neuron computes a weighted sum of its inputs plus a bias, and applies an activation function to it (which is often non-linear), like:

$$z = g(\sum_{i=1}^{n} w_i \, x_i + b) \tag{11}$$

where $g(.)$ is the activation function, x_i is the i^{th} input with its weight w_i and b is the bias. It is thanks to this non-linear activation function that these models can learn complex patterns of the data. Then, the output of a whole layer can be expressed as:

$$\mathbf{z}_k = g(\mathbf{W}_{k-1} \, \mathbf{z}_{k-1} + \mathbf{b}_{k-1}) \tag{12}$$

where it can be seen how the output of layer k depends on the output of the previous layer. The output layers work as a dense layer which is constructed differently depending on the prediction problem. For regression, the layer will have one neuron per target variable, and the activation function will be linear (an identity function). For classification, one target variable will require as many neurons as categories it has, and the activation function will be softmax, as each neuron will output the probability of each category. Binary targets only require one neuron and a sigmoid activation function.

To fit the model, it is necessary to choose a loss function to compute the error, and an optimization algorithm. These algorithms are based on backpropagation, which consists in calculating the gradient of the loss function with respect to the model's parameters, and then making small changes in the parameter's values. Backpropagation is repeated for a certain number of epochs, that is, the number of times the whole training set has been used to adjust the model's parameters.

This is just an overview of the algorithm, for more in depth information, see LeCun et al. (2015) and Pérez-Enciso and Zingaretti (2019).

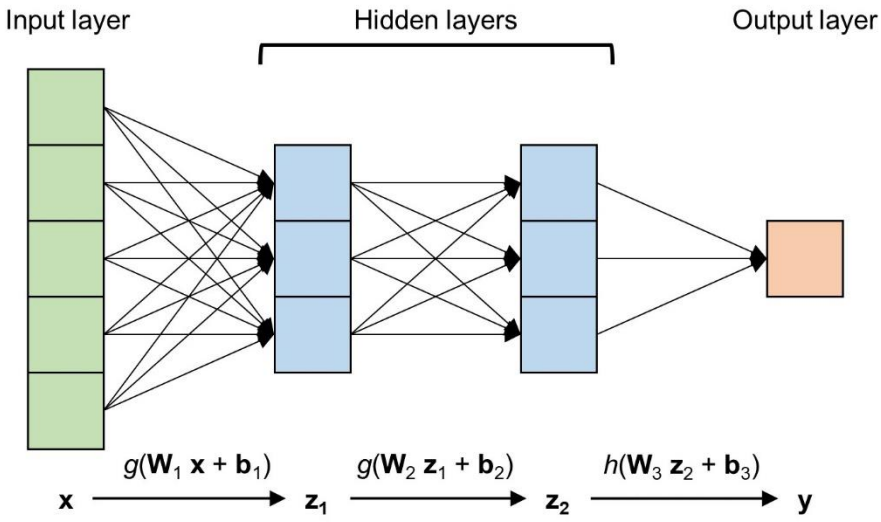


Figure 2. Multi-layer perceptron (MLP) diagram.
The input layer receives the vector of one observation, where every element corresponds with the value for one of the features. In the hidden layers, weighted sums of the inputs are performed (plus a bias), and this is passed to an activation function *g*(.). For the output layer, a different activation function *h*(.) is chosen depending on if the problem is regression or classification.

3.7.2. Input Options

Two approaches were tested to introduce the molecular marker data into the ANN models. First, the "top-m" approach, which uses the genotype matrices with the top 10k markers or top 10k SNPs, as in Bayes C.

Second, the "kerPC" approach, which is an attempt to reduce the dimensionality of the data. This method consists in taking the $\mathbf{G}$ matrices computed for the Bayesian RKHS models, and creating new "kerPC" matrices with their eigenvectors. These eigenvectors give a measure of relatedness between individuals (individuals with a high loading for a given eigenvector will be more similar to each other). Taking all eigenvectors, the new feature matrices for each marker set have 738 variables, a huge decrease from the thousands of variables of the genotype matrices. The exact methodology was to standardize the $\mathbf{G}$ matrices, add a small constant to the diagonal (diag(**G**) * 1.0001) to make sure these matrices are positive definite, and then eigenvectors were computed.

Within this approach, the scenario where only SNP data is used for prediction was set up by just using the kerPC matrix corresponding to $\mathbf{G}_{\mathrm{SNP}}$. The "all" marker data scenario was separated in two: models with a single input layer ("single.in"), where all kerPC matrices were concatenated together; and models with multiple input layers ("multi.in") connected to an additional dense layer, one for each kerPC matrix.

All features were standardized using the whole set of observations (train plus test), for the same reasons argued for Bayes C.

3.7.3. Network Architectures

Feed-forward MLP and CNN architectures were implemented. MLPs were used for both top-m and kerPC inputs. The general architecture consists in an input layer (or multiple with an extra dense layer), a "core" of dense layers, a dropout layer and the output layer.

Regularization of the weights was applied to all dense layers. Loss function chosen for continuous traits was MSE, and binary cross-entropy for binary traits. Adam optimizer (Kingma and Ba, 2017) was used.

CNNs were used just for the top-m inputs. This is because CNNs have the advantage of taking into account the spatial correlation between features, while reducing the number of parameters of the model. Genome-wide marker data can present some degree of correlation between adjacent markers in the form of linkage disequilibrium, so theoretically a CNN would be a better choice to model this kind of data. CNNs process their input in a convolutional layer, where a certain number of filters (or kernels) of given dimensions perform convolutions over the input with a certain stride (Figure 3). A convolution consists in computing dot products of the filter and a subset of the input, while advancing over the input a “stride” at a time. A bias term can be added after each dot product, and the result can be given to an activation function, just like with the neurons of dense layers. In the end, each filter generates a convolved feature. These convolved features can be fed into a pooling layer, which “summarizes” the information of the convolved feature into a lower space by taking the maximum or average of contiguous positions (pool size and stride can be defined). Then, the output is usually flattened and fed to dense layers like in an MLP.

The specific CNN architecture implemented here consists in a 1-dimensional convolutional layer (Conv1D) with various filters, filter size of 3 and stride of 1, bias, activation function and regularization; connected to a maximum pooling layer with pool size of 3 and stride equal to pool size. Then the output is flattened and introduced in an MLP with the same characteristics as described above.

All models were implemented in python with TensorFlow 2.9.1 (Abadi et al., 2015) and the Keras API (Chollet and others, 2015).

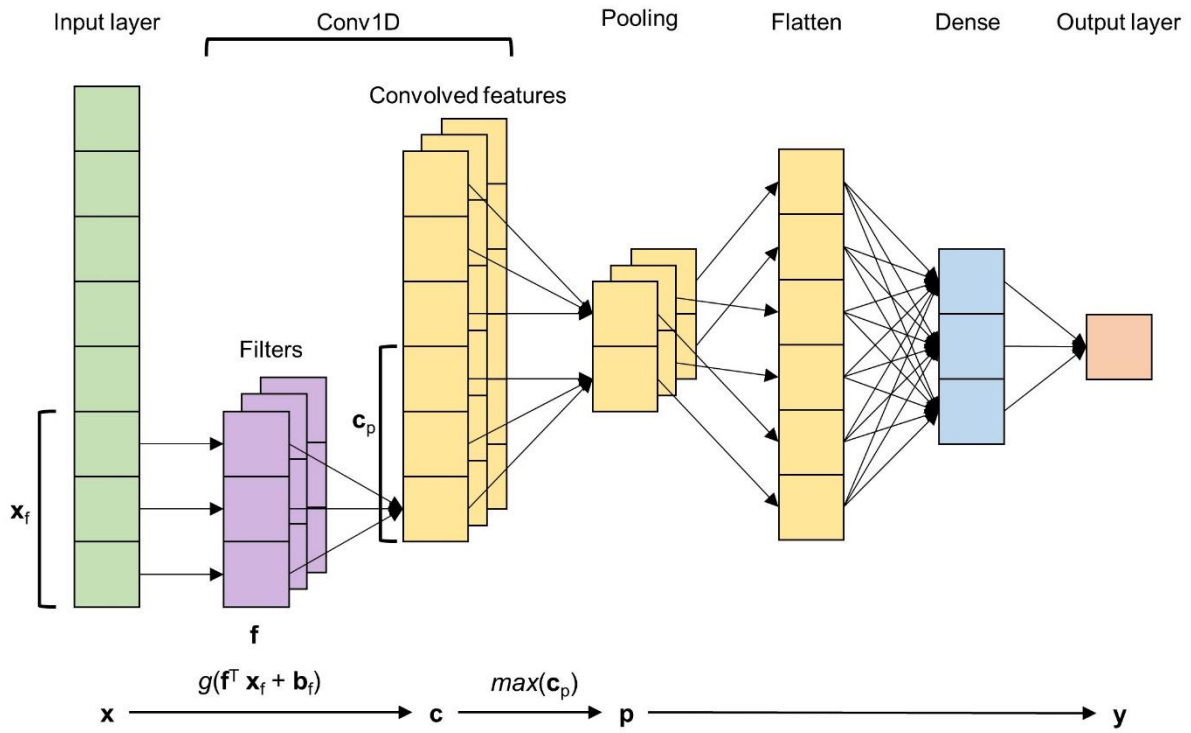


Figure 3. One-dimensional convolutional neural network (CNN) diagram.
The input layer receives the vector of one observation, where every element corresponds with the value for one of the features. In a one-dimensional convolutional layer (Conv1D), filters of size = 3 are applied to subsets of the input ($\mathbf{x}_f$) with a stride of 1 (where $\mathbf{b}_f$ is the bias of the filter, and $g(.)$ is an activation function) to form convolved features. In a pooling layer with pool size = 3 and stride = pool size, the maximums of subsets $\mathbf{c}_p$ of the convolved features are taken. The output of the pooling layer is flattened to one dimension and fed to a perceptron.

3.7.4. Multiple Output Models

Finally, we also tested whether predicting more than one trait simultaneously could improve prediction, as statistical models would be able to exploit the correlation between traits. Three multi-trait models were proposed: one with the two continuous traits (“2.cont.traits”), another one with the two binary (“2.bin.traits”) and a third one with the 4 traits (“4.traits”). These models were only implemented for MLPs. In the case of “top-m” input option, the marker genotype matrices associated to each of the traits were introduced in different input layers (with a connected dense layer). For the “kerPC” input option, both “single.in” and “multi.in” models were tested. In all models, an independent dense layer was added before each output layer.

In all, our ANN models can be summarized with a code indicating “input-option_architecture_input-number”, resulting in 5 different main model types. The specific implementation of this main types creates submodels based on the number of outputs and the marker set information used (Table 3). The resulting topologies can be summarized with the graphs shown on Figure 4.

Table 3. Summary of the ANN models.

Input option	Architecture	Input number	Output number	Marker sets
kerPC	MLP	single.in	single.out / multi.out	SNP / all
kerPC	MLP	multi.in	single.out / multi.out	all
top-m	MLP	single.in	single.out	SNP / all
top-m	MLP	multi.in	multi.out	SNP / all
top-m	CNN	single.in	single.out	SNP / all

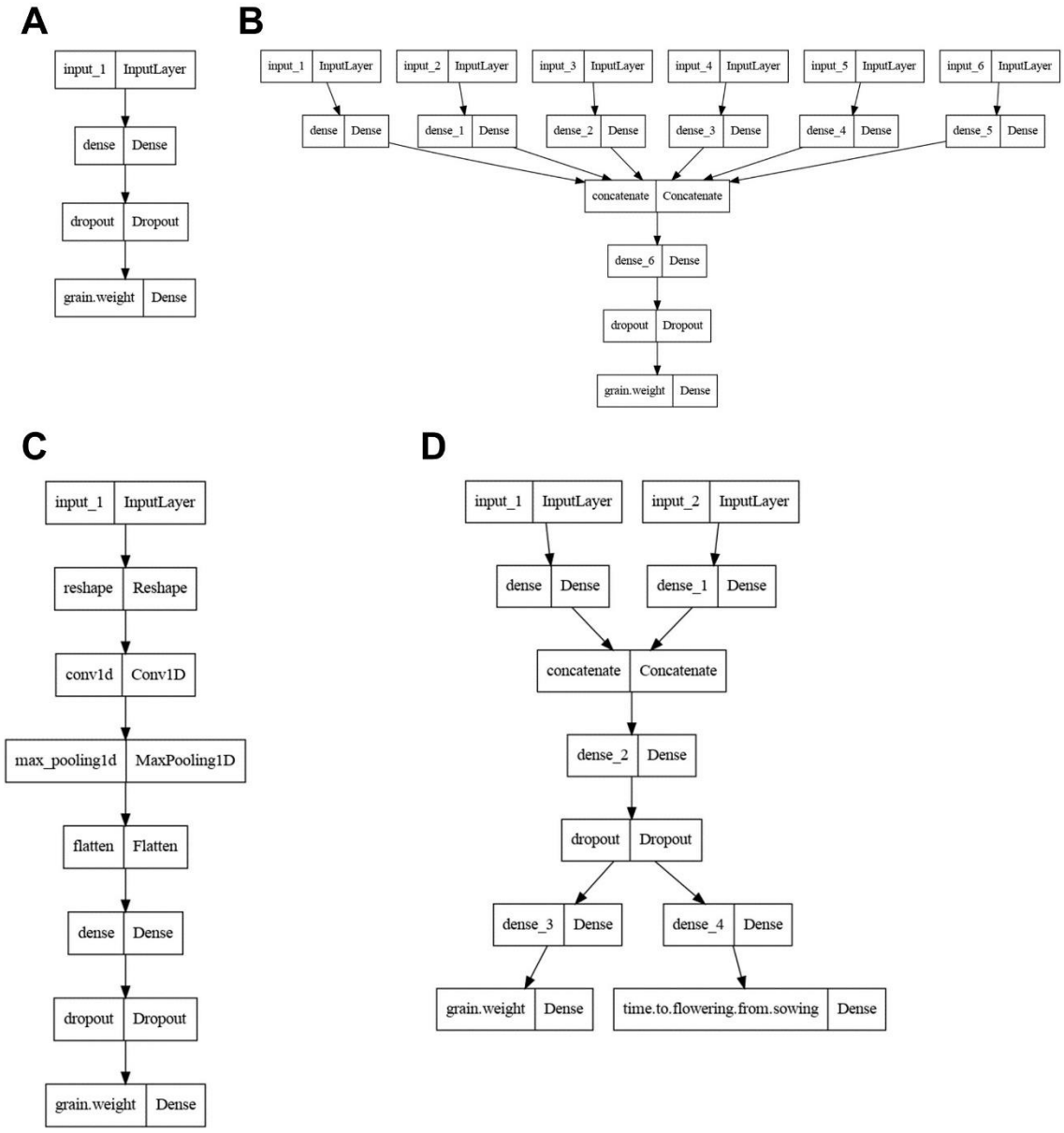


Figure 4. Graphs of ANN models.
Representative graphs of the different artificial neural network (ANN) topologies implemented, obtained with the Keras API. Note that only one dense layer has been represented before the dropout layer for simplification. **(A)** kerPC_MLP_single.in and top-m_MLP_single.in models. **(B)** kerPC_MLP_multi.in model. **(C)** top-m_CNN_single.in model. **(D)** top-m_MLP_multi.in model. Note it is a multi-trait model.

3.7.5. Hyperparameter Tuning

ANNs offer enormous flexibility in their construction, but that requires careful hyperparameter optimization. We searched the hyperparameter space of Table 4 looking for the optimal ones, for each model and trait (or combination of them). For the AroAdm GP framework, a holdout validation was performed, that is, 20% of the training data was set aside as a validation set, and hyperparameter configurations were evaluated on it. For the 10-fold-CV, no hyperparameter search was performed, and instead the best ones from the AroAdm case were chosen. This decision was due to the high computational cost of the search, and the impossibility of carrying it out for 10 different train/test splits. For the search, a Hyperband algorithm (Li et al., 2018) was employed with a maximum number of epochs of 50, 40 hyperband iterations (20 for the CNNs) and an early stopping callback.

Table 4. Hyperparameter search space.
Hyperparameters in order: activation function (used for all dense and convolutional layers), regularization type and rate, No. of dense layers, No. of neurons of dense layer x (is conditional to n_layers), dropout rate, No. of neurons of dense layers connected to independent inputs, No. of neurons of dense layers connected to independent outputs, No. of filters of a convolutional layer. Values in-between square brackets indicate a choice. Models in parenthesis refer only to their multi-trait cases.

Hyperparameter	Values	Models
activation	['relu', 'tanh', 'linear', 'softplus']	All
regularization_type	['L2', 'L1']	All
regularization_rate	[0.0, 0.1, 0.01, 0.001]	All
n_layers	'min_value': 1, 'max_value': 5, 'step': 1	All
units_x	[10, 20, 40, 80, 160]	All
dropout_rate	'min_value': 0.0, 'max_value': 0.3, 'step': 0.05	All
first_units	[10, 20, 40, 80, 160]	kerPC_MLP_multi.in, top-m_MLP_multi.in
last_units	[10, 20, 40, 80, 160]	top-m_MLP_multi.in, (kerPC_MLP_single.in), (top-m_MLP_single.in)
n_filters	[16, 32, 64, 128]	top-m_CNN_single.in

3.7.6. Prediction

Models were built with their respective best hyperparameters. Validation sets were created with 20% of the training data. Models were fitted with the smaller training set for 50 epochs with an early stopping callback (min_delta = 0.001, patience = 10). Predictions were computed with the test set, and performance metrics were calculated (the same as for linear models). This process was repeated 5 times and the average of the metrics was taken.

All the code used for data processing and model implementation can be found at https://github.com/anavartinez/rice-GP_SV_DL.

4. Results

4.1. Exploratory Analysis

PCA of the marker genotype matrices (Figure 5) revealed presence of population structure according to the rice variety group. Regarding the matrix of SNPs, the one with the largest number of markers spread across the genome, rice groups are genetically distinct from each other, being indica and japonica the furthest apart. Admixed accessions are spread along the first PC, as expected. Almost the same structure can be seen for the RLX-RIX matrix. For DEL and MITE-DTX it is also similar, but for DEL there is a greater spread of the indica group, and for MITE-DTX there is very little separation by the second PC. DUP and INV marker sets show quite different maps. For DUP there is still a slight separation by variety group, with a subset of the indica group splitting from the rest; and for INV, most accessions are almost the same, except for a subset of indica again. This could be due to them being the smallest datasets, so the genome-wide pattern captured by SNPs does not show here. In any case, we have seen that our data is structured, and that rice variety group will need to be considered in downstream analyses.

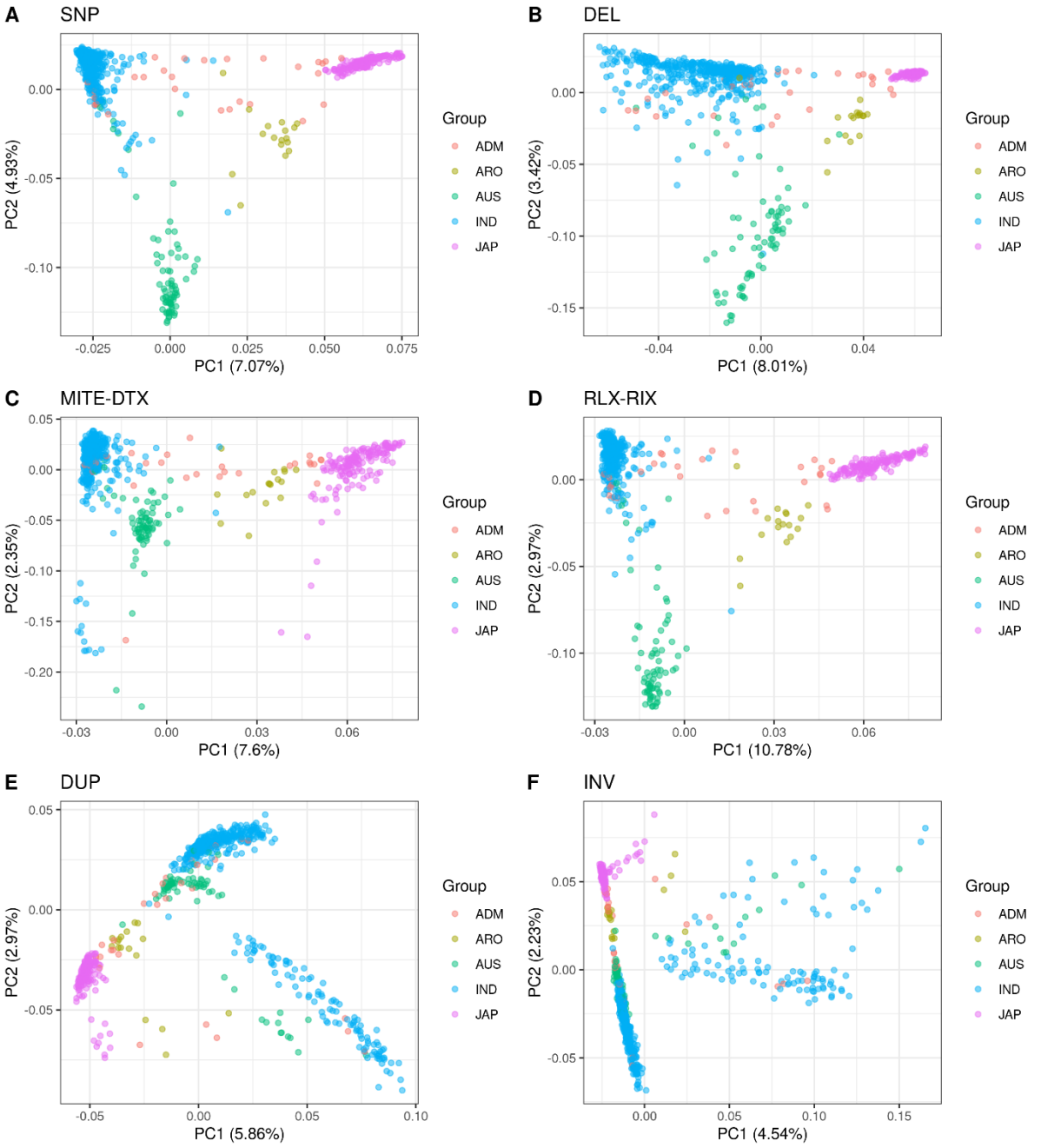


Figure 5. PCA of the marker genotype matrices.
Observations are colored by rice population group.

Regarding the distributions of the traits of interest (Figure 6), histogram of time to flowering shows that the log transformation was successful in attaining a bell-shaped distribution. Histogram of leaf senescence also makes clear that the binning of the original categories produced balanced classes.

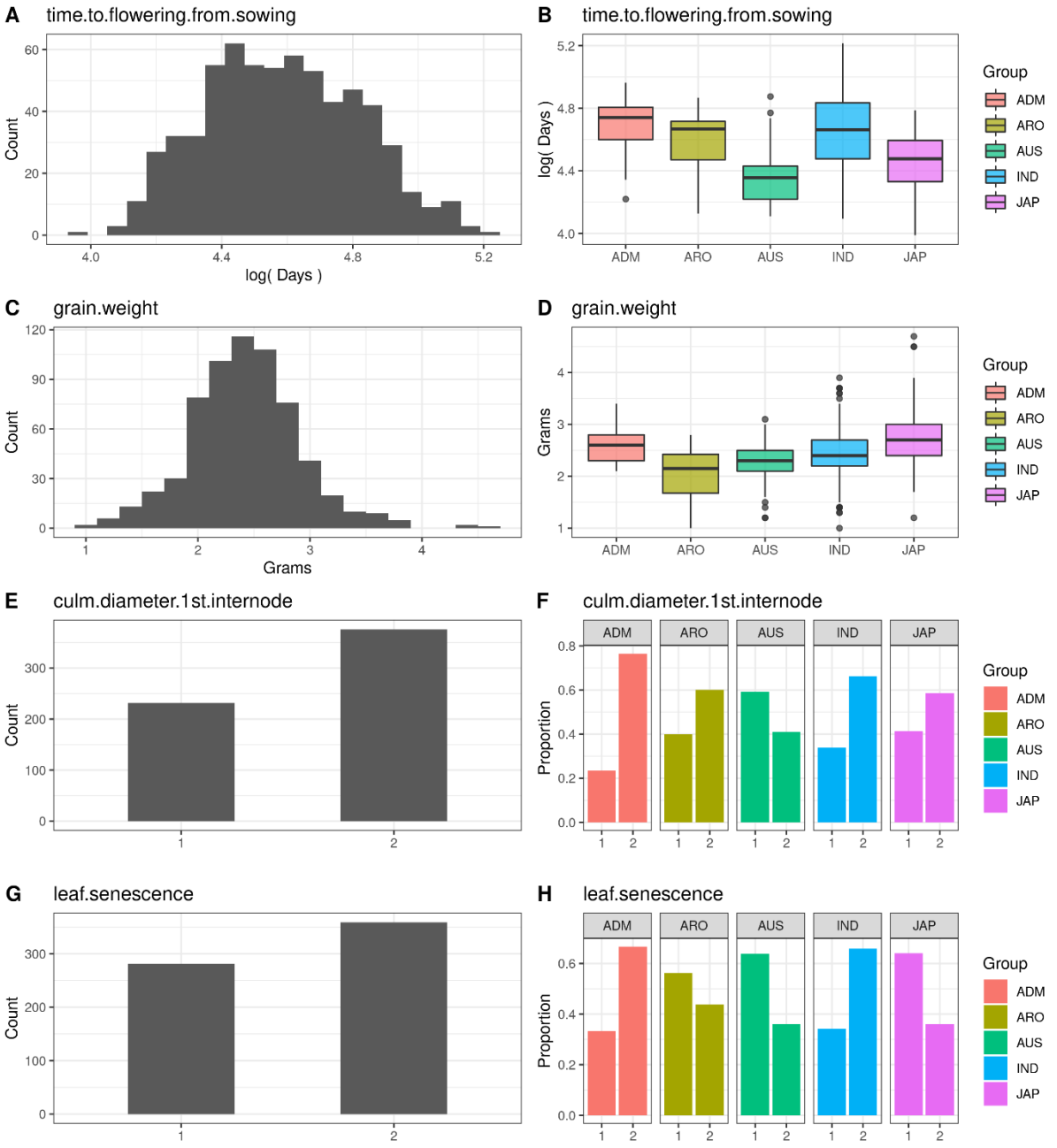


Figure 6. Trait variable distributions.
Distributions per rice population group shown on the right.

Finally, correlations between traits were computed (Figure 7). Culm diameter and time to flowering are modestly correlated. Otherwise, correlations were weak.

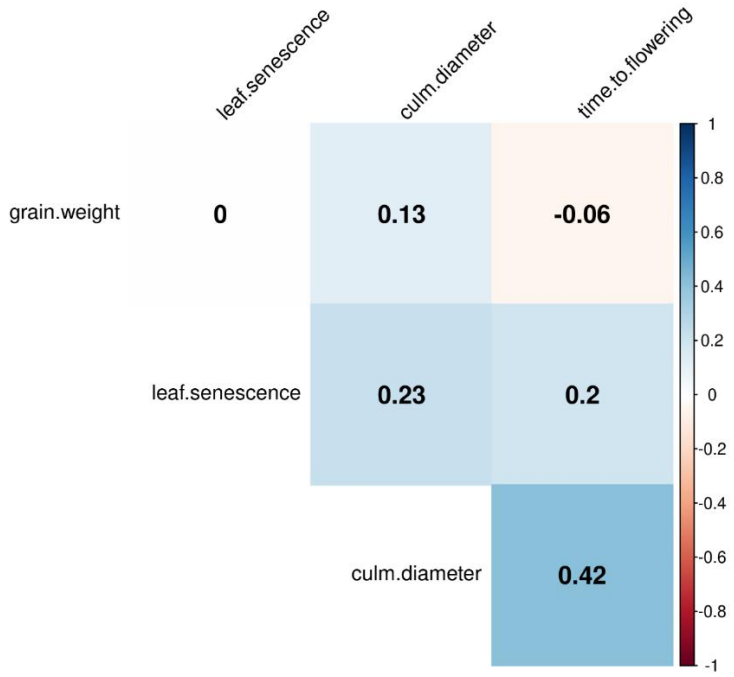


Figure 7. Correlation between traits.

4.2. Genomic Variance Inference

Pairwise estimates of the fraction of phenotypic variance explained by SNPs and by another structural variant's effects can be seen on Figure 8. The value of the h^2 ratio for SNPs is the average of all pairwise comparisons. By adding up any two of these pairwise ratios, we can get an estimate of the genomic heritability of the trait (h_g^2), that is, the proportion of phenotypic variance explained by the regression on markers (de los Campos et al., 2015). h_g^2 is quite high for time to flowering, but mainly because of the SNP component, so our expectation is that adding the SV data will probably not amount to a substantial increase in predictive accuracy. A similar trend is observed for grain weight. For culm diameter and leaf senescence, SNPs explain a lower proportion of the trait's variance, so in this case we expect the possible prediction improvement to be more noticeable, especially for leaf senescence.

Earlier results from Vourlaki et al. (2022) showed very good predictive accuracy for time to flowering, so we included this trait in the analysis to have a control case where GP has already been implemented successfully. The same study showed poor results for grain weight, so we chose the trait to see if predictive accuracy could be improved. The other two traits were chosen as candidates for which SV data could make a difference in prediction.

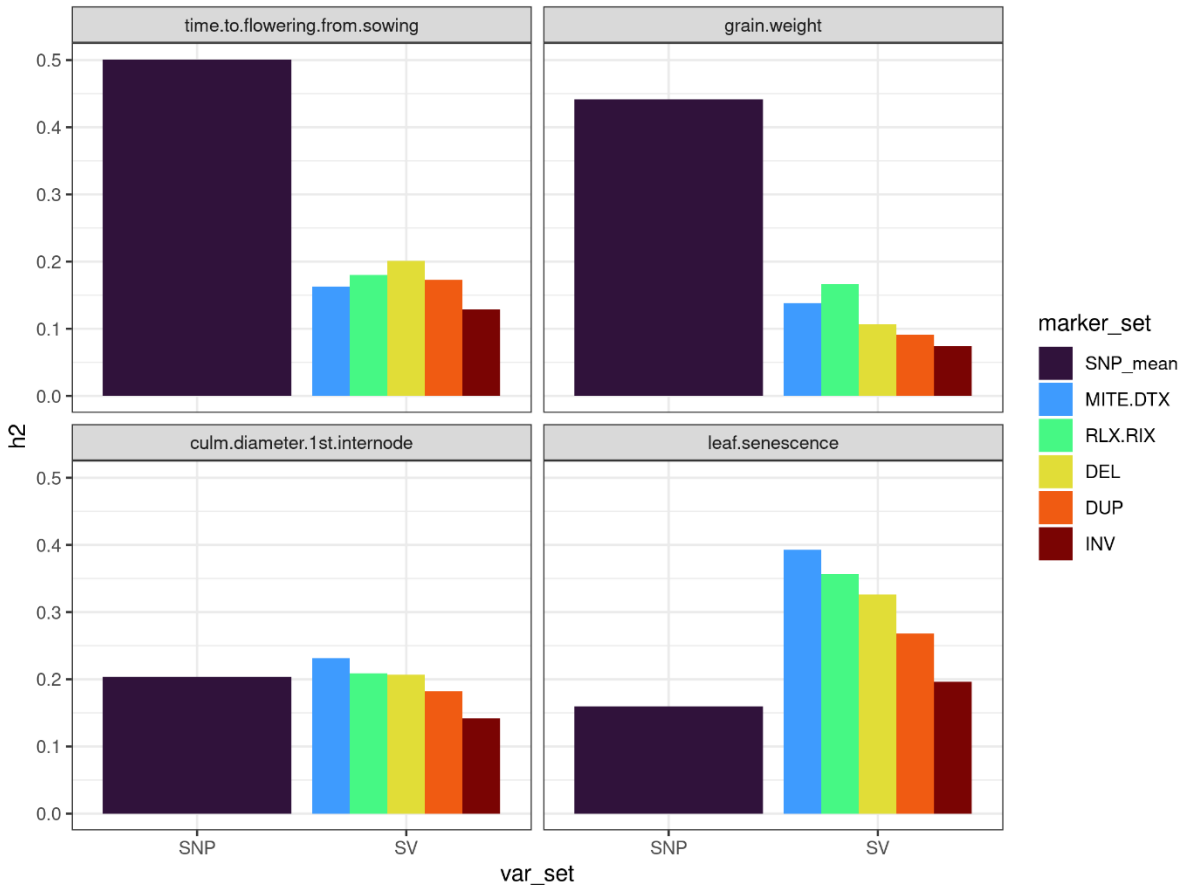


Figure 8. Fractions of phenotypic variance explained by marker sets.
Fractions were obtained as pairwise estimates using the SNP marker set and a structural variant (SV) one. The value shown for SNP is the average of all pairwise estimates.

4.3. GWAS

Genotype matrices with the 10k most associated markers to each trait were explored regarding their marker type composition (Figure 9). In general, there are more top marker hits of the marker types which were more abundant in the first place (SNP > DEL > MITE-DTX > RLX-RIX > DUP > INV). This is especially true for SNP and DEL, of which the original number of markers is considerably larger than the rest. Interestingly, the top 10k markers matrix of grain weight is composed primarily of SNPs, having few hits for the SVs. Time to flowering and leaf senescence present a substantial proportion of associated MITE-DTX markers, compared to the other traits.

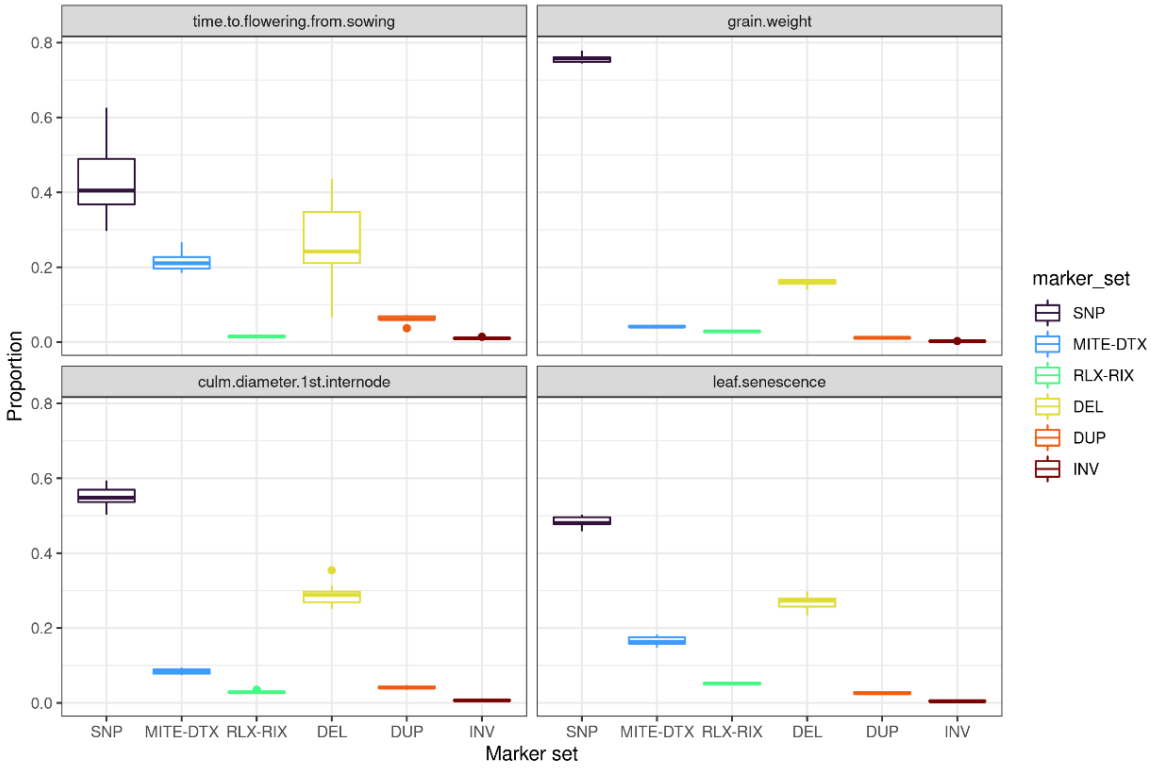


Figure 9. Marker type proportions in the 10k most associated markers to each trait.
Box plots represent the 11 observations obtained from performing GWAS with different training sets (10 observations from the 10-fold cross-validation setting and 1 from AroAdm).

4.4. Genomic Prediction

We chose to compare the different models' performances in terms of the correlation and AUC metrics (Figure 10), as with them we have a sense of scale (correlation from 0 to 1, AUC from 0.5 to 1). Results based on the error metrics are in the appendix (Figure S1). There is a general trend of prediction working better for the 10-fold CV than for the AroAdm partition. For all traits, the maximum performance achieved by 10-fold CV (maximum of the medians) is greater than the maximum for AroAdm. This confirms our supposition that prediction would be more challenging for AroAdm because of training data not being very related to test data. Another general tendency is that kerPC ANN models perform worse than linear and top-m ANN models. Moreover, Bayesian RKHS and Bayes C have a similar performance for all cases.

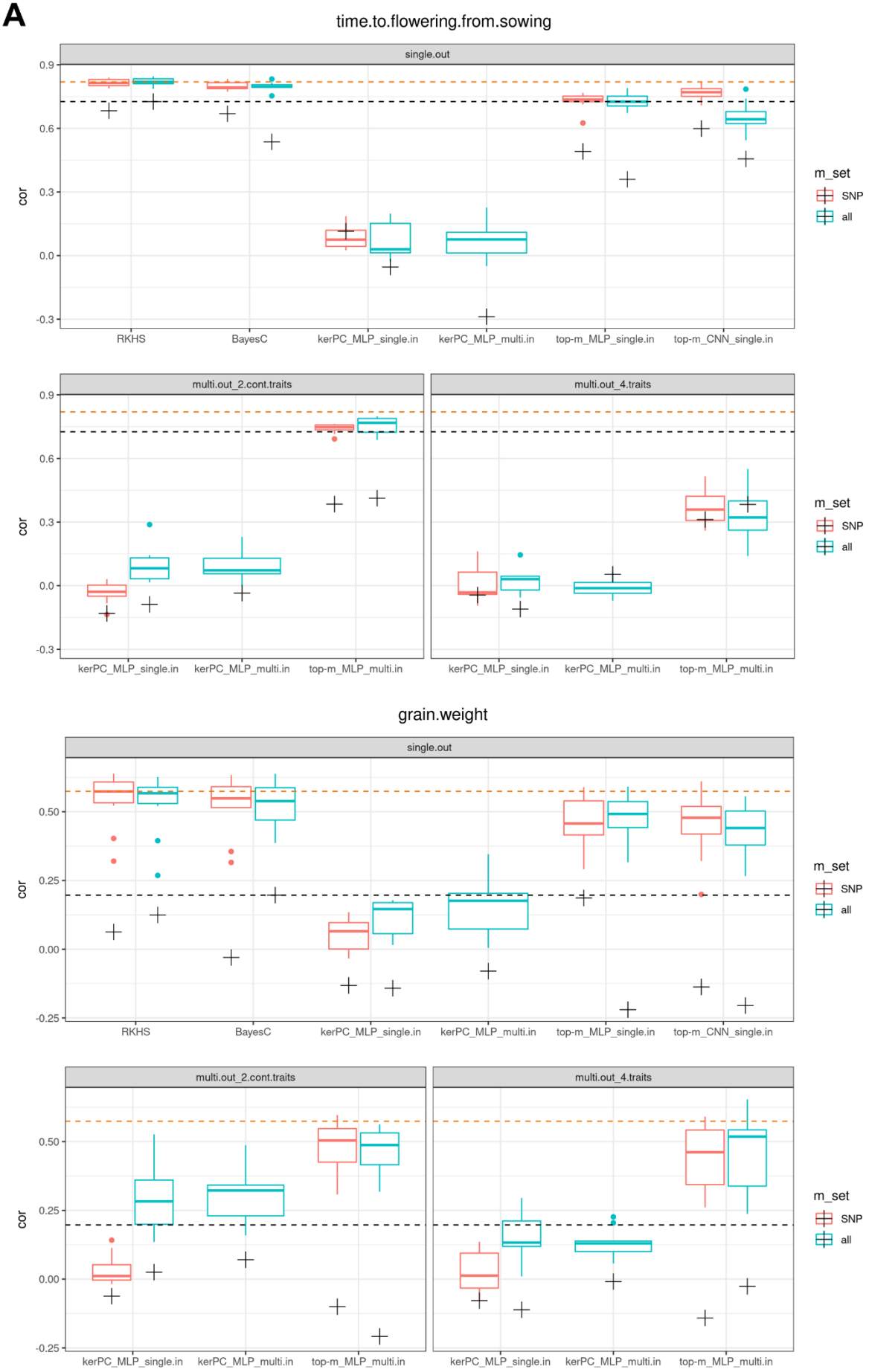
Focusing on time to flowering (Figure 1A), correlations are quite high, up to ~0.8. The best predictions were obtained with the linear models, being RKHS a better option because of how it fared with the AroAdm partition. SVs do not seem to improve prediction. Top-m models are a small downgrade compared to the linear models, and the multi.out_2.cont.traits case performed like the single-trait models did, so it appears that the uncorrelated trait grain weight is not helping prediction, as expected. KerPC does not work at all for this trait, with correlations of about 0.

For grain weight, prediction based on molecular markers appears to be very difficult. Relatively good correlations can be achieved for the 10-fold CV scenario (about 0.5), with a similar trend of linear models > top-m > kerPC, but then prediction fails for the AroAdm case (correlation of 0 or even negative). This shows how the trained models struggle to generalize to new data; it is like the underlying genetic architecture of the trait cannot be learnt from the observations of IND, JAP and AUS groups.

Culm diameter's best AUCs for the 10-fold CV case are achieved by top-m ANN models (AUCs near 0.7) (Figure 10B). However, there is the same problem as with grain weight: prediction for AroAdm is not reliable. Adding SV data does not appear to have an impact, despite our expectation based on the genomic variance inference; and multi-trait models also do not seem to pose an advantage. In addition, neither linear models nor kerPC ANNs were able to produce accurate predictions.

Regarding leaf senescence, the only models capable of making correct classifications are top-m ANNs again. AUCs are higher in this case, up to ~0.78. For this trait, SVs do seem to improve prediction, in accordance with our observation that SV datasets could explain a larger fraction of the phenotypic variance than SNPs. Moreover, classification is still good for the AroAdm framework, especially with the models including SV data. MLPs appear to work a little better than CNNs, and multi-trait models do not improve prediction. The increase in the AUC for the MLP model when including SVs, considering the medians of the 10-fold CV, is of 7.40%.

A


time.to.flowering.from.sowing
single.out
multi.out_2.cont.traits
multi.out_4.traits
cor
m_set
SNP
all
RKHS
BayesC
kerPC_MLP_single.in
kerPC_MLP_multi.in
top-m_MLP_single.in
top-m_CNN_single.in
top-m_MLP_multi.in
grain.weight

B

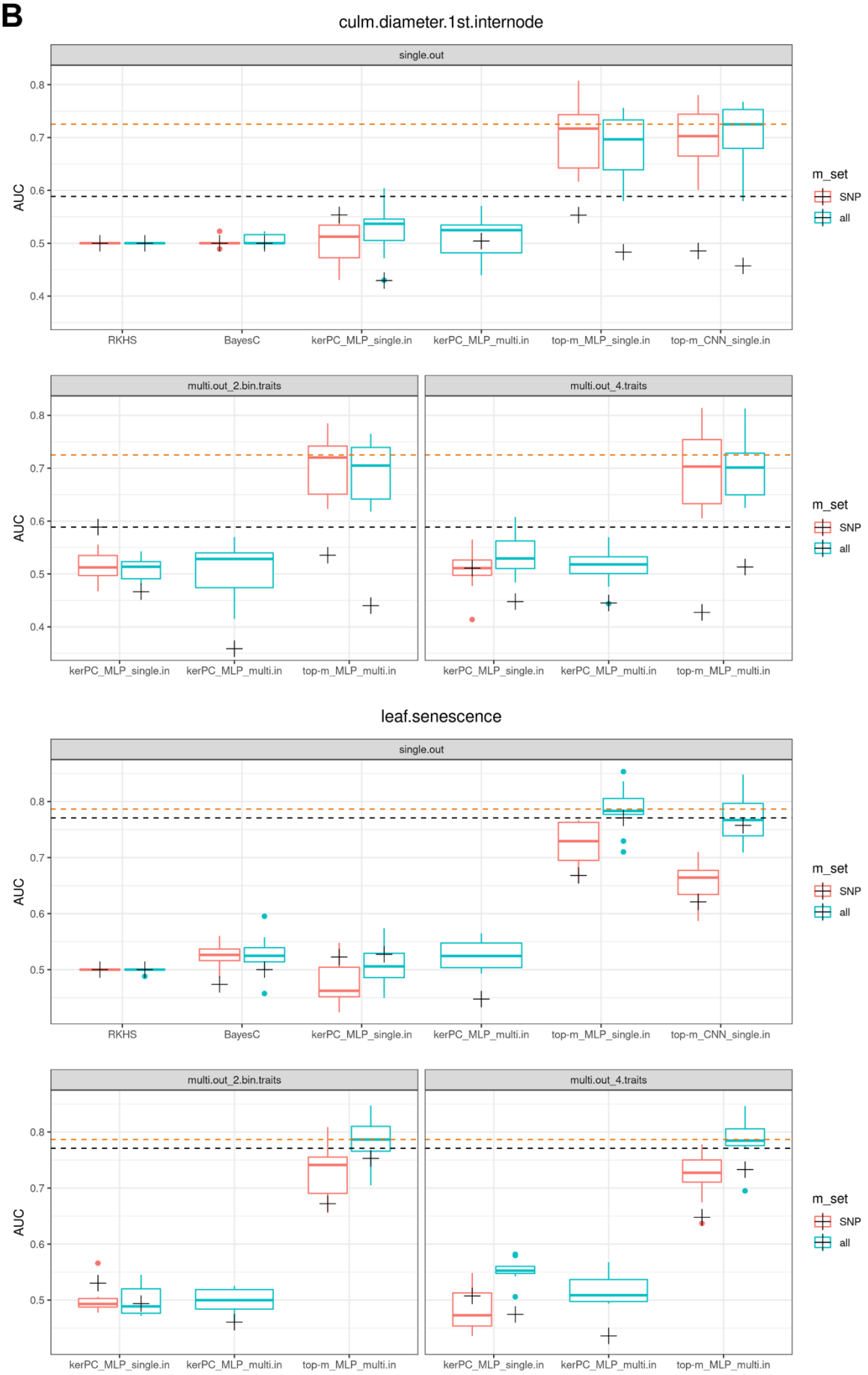

culm.diameter.1st.internode
single.out
multi.out_2.bin.traits
multi.out_4.traits
AUC
m_set
SNP
all
RKHS
BayesC
kerPC_MLP_single.in
kerPC_MLP_multi.in
top-m_MLP_single.in
top-m_CNN_single.in
top-m_MLP_multi.in
leaf.senescence

Figure 10. Genomic prediction results.
(A) displays the correlation between predicted and observed data of the continuous traits, and **(B)** the area under the curve (AUC) of the binary trait predictions. For each trait, results for single-trait (single.out) and multi-trait models with the two continuous traits (multi.out_2.cont.traits) and the four traits (multi.out_4.traits) are shown. X axis represents the model used, where the artificial neural network models are coded like “input-option_architecture_input-number”, as described in the text. Models trained with SNP data or data from “all” marker sources are distinguished. Results for the 10-fold cross-validation framework are represented with box plots, and the ones for the “AroAdm” case are represented by plus (+) signs. The orange dotted line shows the maximum of the median values achieved in the 10-fold cross-validation, and the black dotted line describes the maximum value for the “AroAdm” case.

5. Discussion

We have performed GP of four agronomically relevant rice traits in a dataset with 738 accessions, representing the main populations of rice. In each of the implemented methods, we have obtained two measurements of the performance of GP: an overall measurement, obtained by performing 10-fold CV over the whole dataset, where accessions belonging to the different rice variety groups are randomly distributed across partitions; and an across-populations measurement, where accessions from the aromatic and admixed groups are predicted using the rest of the groups, which are more distantly related, making this a challenging prediction scenario.

Regarding whether SVs can improve GP, we observed this to be the case for leaf senescence (> 7% increase). Depending on the trait, SVs can be an important source of genetic variation in addition to SNPs. For this reason, it might be worth genotyping SVs in breeding programs. With our implementation, that is, doing a pre-selection of the most associated markers of all kinds, results did not seem to worsen when including the SV data, so this methodology could offer an advantage if implemented as a standard.

Of the different tested models, we have observed that linear models excel at regression tasks (time to flowering and grain weight). Considering their simplicity and robustness, it is sensible to use them as a benchmark on a prediction experiment. ANNs are appropriate if dominant and epistatic effects play an important role for the traits. However, the performance of linear models for classification (culm diameter and leaf senescence) was quite poor. This might be due to additive genetic effects being small for these traits, or due to the implementation in BGLR. We suspect the R package because prediction results considering both variables as continuous were relatively decent (Vourlaki et al., 2022). Conversely, ANNs performed well for these traits.

With respect to the ANN implementations, the kerPC input option was inadequate for all traits. It seems like too much information is lost when transforming the original genotypes to this measure of relatedness between individuals. The top markers input option behaved reasonably well, reaching competitive results with the linear models for regression and obtaining the best results for classification. Differences between MLP and CNN performances were not large. Multi-trait models did not improve prediction, and in fact, they were worse for some of the cases, so they are not recommended. They might still be useful for highly correlated traits.

Focusing on the traits themselves, it appears prediction in an across-populations scenario is more challenging for grain weight and culm diameter, as the models did not generalize well after training with the JAP, IND and AUS accessions. This information should be taken into account when designing breeding experiments. Time to flowering and leaf senescence were not affected as much by this.

6. Conclusions

- SVs can improve GP for traits in which SVs are a significant source of genetic variance.
- Linear models perform very well for regression, but less so with classification.
- ANN models are competitive with linear models for regression, and they could be a good option for traits with important non-additive genetic effects. They can also deal with classification tasks.
- The eigenvectors of genomic relationship matrices are not good input options for ANNs.
- For the traits analyzed here, CNNs and multi-trait models do not seem to pose an advantage versus MLPs and single-trait models, respectively.
- Prediction across populations is more difficult for grain weight and culm diameter than for time to flowering and leaf senescence.

Bibliography

Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, et al. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. https://www.tensorflow.org/.

Amadeu RR, Cellon C, Olmstead JW, Garcia AAF, Resende Jr. MFR, Muñoz PR. 2016. AGHmatrix: R Package to Construct Relationship Matrices for Autotetraploid and Diploid Species: A Blueberry Example. The Plant Genome. 9(3):plantgenome2016.01.0009. doi:10.3835/plantgenome2016.01.0009.

van-Arendonk JAM, Tier B, Kinghorn BP. 1994. Use of Multiple Genetic Markers in Prediction of Breeding Values. Genetics. 137(1):319–329.

Bai S, Yu H, Wang B, Li J. 2018. Retrospective and perspective of rice breeding in China. Journal of Genetics and Genomics. 45(11):603–612. doi:10.1016/j.jgg.2018.10.002.

Browning BL, Zhou Y, Browning SR. 2018. A One-Penny Imputed Genome from Next-Generation Reference Panels. The American Journal of Human Genetics. 103(3):338–348. doi:10.1016/j.ajhg.2018.07.015.

de los Campos G, Gianola D, Rosa GJM. 2009. Reproducing kernel Hilbert spaces regression: A general framework for genetic evaluation1. Journal of Animal Science. 87(6):1883–1887. doi:10.2527/jas.2008-1259.

de los Campos G, Hickey JM, Pong-Wong R, Daetwyler HD, Calus MPL. 2013. Whole-Genome Regression and Prediction Methods Applied to Plant and Animal Breeding. Genetics. 193(2):327–345. doi:10.1534/genetics.112.143313.

de los Campos G, Sorensen D, Gianola D. 2015. Genomic Heritability: What Is It? PLoS Genet. 11(5):e1005048. doi:10.1371/journal.pgen.1005048.

Carpentier M-C, Manfroi E, Wei F-J, Wu H-P, Lasserre E, Llauro C, Debladis E, Akakpo R, Hsing Y-I, Panaud O. 2019. Retrotranspositional landscape of Asian rice revealed by 3000 genomes. Nat Commun. 10(1):24. doi:10.1038/s41467-018-07974-5.

Castanera R, Vendrell-Mir P, Bardil A, Carpentier M-C, Panaud O, Casacuberta JM. 2021. Amplification dynamics of miniature inverted-repeat transposable elements and their impact on rice trait variability. The Plant Journal. 107(1):118–135. doi:10.1111/tpj.15277.

Chollet F, others. 2015. Keras. https://keras.io.

Domínguez M, Dugas E, Benchouaia M, Leduque B, Jiménez-Gómez JM, Colot V, Quadrana L. 2020. The impact of transposable elements on tomato diversity. Nat Commun. 11(1):4058. doi:10.1038/s41467-020-17874-2.

Fuentes RR, Chebotarov D, Duitama J, Smith S, De la Hoz JF, Mohiyuddin M, Wing RA, McNally KL, Tatarinova T, Grigoriev A, et al. 2019. Structural variants in 3000 rice genomes. Genome Res. 29(5):870–880. doi:10.1101/gr.241240.118.

Gabur I, Chawla HS, Snowdon RJ, Parkin IAP. 2019. Connecting genome structural variation with complex traits in crop plants. Theor Appl Genet. 132(3):733–750. doi:10.1007/s00122-018-3233-0.

Grandbastien M-A. 2015. LTR retrotransposons, handy hitchhikers of plant regulation and stress response. Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms. 1849(4):403–416. doi:10.1016/j.bbagrm.2014.07.017.

Habier D, Fernando RL, Kizilkaya K, Garrick DJ. 2011. Extension of the bayesian alphabet for genomic selection. BMC Bioinformatics. 12(1):186. doi:10.1186/1471-2105-12-186.

Jiang N, Panaud O. 2013. Transposable Element Dynamics in Rice and Its Wild Relatives. In: Zhang Q, Wing RA, editors. Genetics and Genomics of Rice. New York, NY: Springer. (Plant Genetics and Genomics: Crops and Models). p. 55–69. [accessed 2022 Sep 3]. https://doi.org/10.1007/978-1-4614-7903-1_5.

Kingma DP, Ba J. 2017. Adam: A Method for Stochastic Optimization. doi:10.48550/arXiv.1412.6980. [accessed 2022 Aug 29]. http://arxiv.org/abs/1412.6980.

LeCun Y, Bengio Y, Hinton G. 2015. Deep learning. Nature. 521(7553):436–444. doi:10.1038/nature14539.

Li L, Jamieson K, DeSalvo G, Rostamizadeh A, Talwalkar A. 2018. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. Journal of Machine Learning Research. 18(185):1–52.

Maher B. 2008. Personal genomes: The case of the missing heritability. Nature. 456(7218):18–21. doi:10.1038/456018a.

Mansueto L, Fuentes RR, Borja FN, Detras J, Abriol-Santos JM, Chebotarov D, Sanciangco M, Palis K, Copetti D, Poliakov A, et al. 2017. Rice SNP-seek database update: new SNPs, indels, and queries. Nucleic Acids Research. 45(D1):D1075–D1081. doi:10.1093/nar/gkw1135.

Meuwissen TH, Hayes BJ, Goddard ME. 2001. Prediction of total genetic value using genome-wide dense marker maps. Genetics. 157(4):1819–1829.

Montesinos-López OA, Montesinos-López A, Pérez-Rodríguez P, Barrón-López JA, Martini JWR, Fajardo-Flores SB, Gaytan-Lugo LS, Santana-Mancilla PC, Crossa J. 2021. A review of deep learning applications for genomic selection. BMC Genomics. 22:19. doi:10.1186/s12864-020-07319-x.

Pérez P, de los Campos G. 2014. Genome-Wide Regression and Prediction with the BGLR Statistical Package. Genetics. 198(2):483–495. doi:10.1534/genetics.114.164442.

Pérez-Enciso M, Zingaretti LM. 2019. A Guide on Deep Learning for Complex Trait Genomic Prediction. Genes. 10(7):553. doi:10.3390/genes10070553.

Stuart T, Eichten SR, Cahn J, Karpievitch YV, Borevitz JO, Lister R. 2016. Population scale mapping of transposable element diversity reveals links to gene regulation and epigenomic variation. Zilberman D, editor. eLife. 5:e20777. doi:10.7554/eLife.20777.

The 3000 rice genomes project. 2014. The 3,000 rice genomes project. GigaScience. 3(1):2047-217X-3–7. doi:10.1186/2047-217X-3-7.

Uga Y, Sugimoto K, Ogawa S, Rane J, Ishitani M, Hara N, Kitomi Y, Inukai Y, Ono K, Kanno N, et al. 2013. Control of root system architecture by DEEPER ROOTING 1 increases rice yield under drought conditions. Nat Genet. 45(9):1097–1102. doi:10.1038/ng.2725.

VanRaden PM. 2008. Efficient Methods to Compute Genomic Predictions. Journal of Dairy Science. 91(11):4414–4423. doi:10.3168/jds.2007-0980.

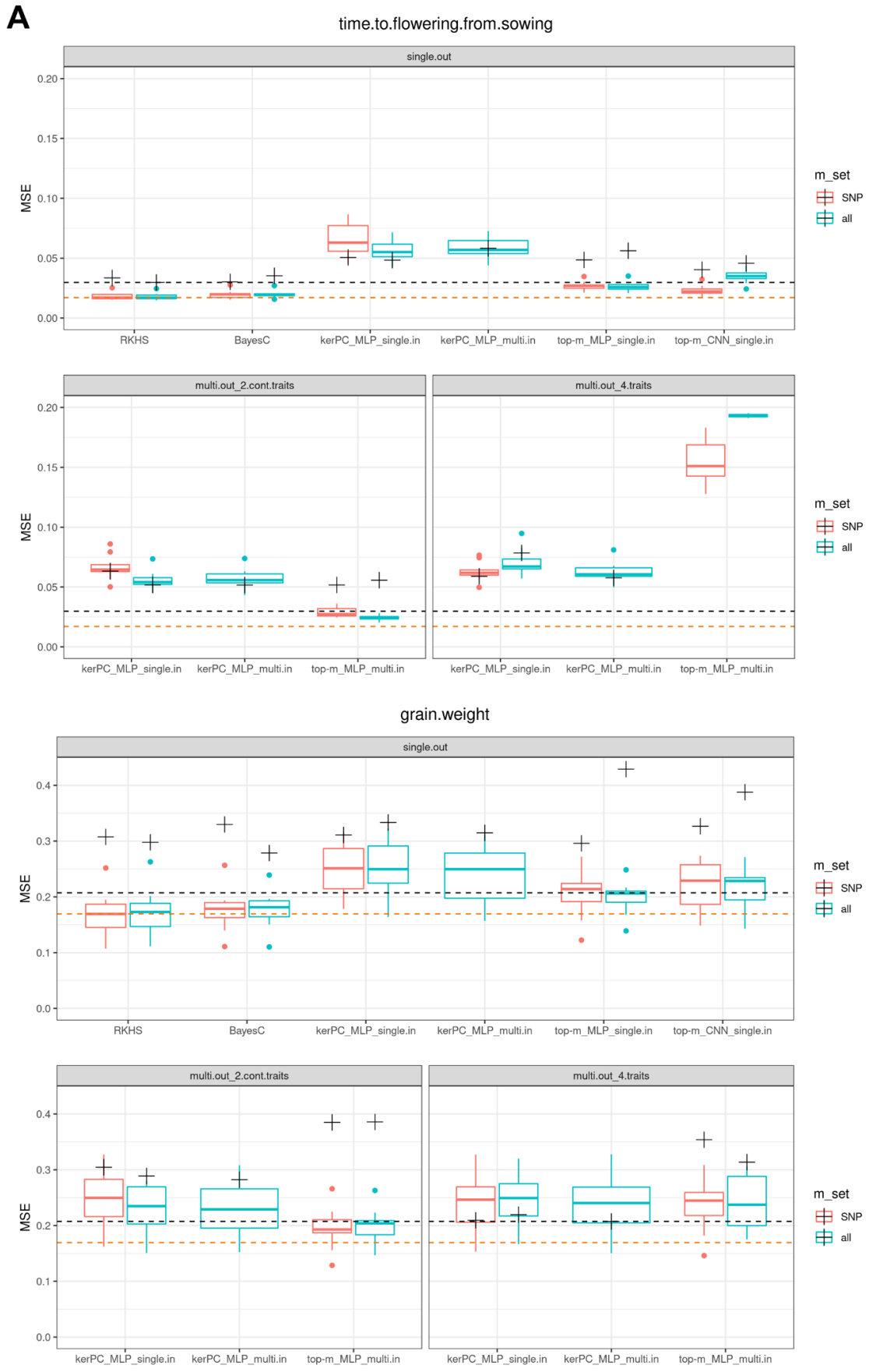
Vourlaki I-T, Castanera R, Ramos-Onsins SE, Casacuberta JM, Pérez-Enciso M. 2022 Aug 5. Transposable element polymorphisms improve prediction of complex agronomic traits in rice. Theor Appl Genet. doi:10.1007/s00122-022-04180-2. [accessed 2022 Aug 8]. https://doi.org/10.1007/s00122-022-04180-2.

Wang W, Mauleon R, Hu Z, Chebotarov D, Tai S, Wu Z, Li M, Zheng T, Fuentes RR, Zhang F, et al. 2018. Genomic variation in 3,010 diverse accessions of Asian cultivated rice. Nature. 557(7703):43–49. doi:10.1038/s41586-018-0063-9.

Wang Yuexing, Xiong G, Hu J, Jiang L, Yu H, Xu Jie, Fang Y, Zeng L, Xu E, Xu Jing, et al. 2015. Copy number variation at the GL7 locus contributes to grain size diversity in rice. Nat Genet. 47(8):944–948. doi:10.1038/ng.3346.

Wei T, Simko V. 2021. R package “corrplot”: Visualization of a correlation matrix. https://github.com/taiyun/corrplot.

Appendix


A
time.to.flowering.from.sowing
single.out
multi.out_2.cont.traits
multi.out_4.traits
MSE
m_set
SNP
all
RKHS
BayesC
kerPC_MLP_single.in
kerPC_MLP_multi.in
top-m_MLP_single.in
top-m_CNN_single.in
top-m_MLP_multi.in
grain.weight

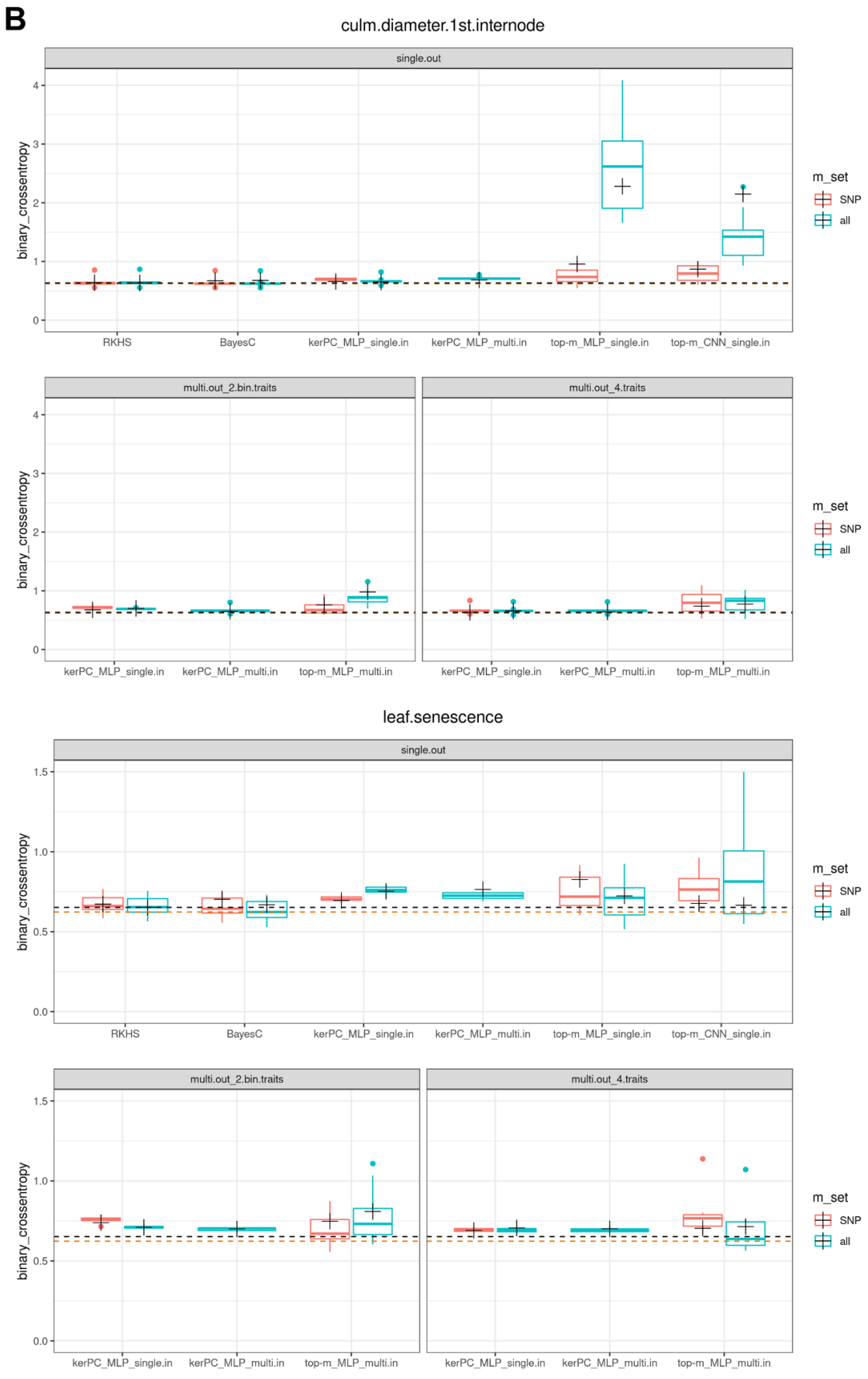


Figure S1. Genomic prediction results using error metrics.

(A) displays the mean squared error (MSE) of the continuous trait prediction, and **(B)** the binary cross-entropy of the binary trait predictions. For each trait, results for single-trait (single.out) and multi-trait models with the two continuous traits (multi.out_2.cont.traits) and the four traits (multi.out_4.traits) are shown. X axis represents the model used, where the artificial neural network models are coded like “input-option_architecture_input-number”, as described in the text. Models trained with SNP data or data from “all” marker sources are distinguished. Results for the 10-fold cross-validation framework are represented with box plots, and the ones for the “AroAdm” case are represented by plus (+) signs. The orange dotted line shows the minimum of the median values achieved in the 10-fold cross-validation, and the black dotted line describes the minimum value for the “AroAdm” case.

Acknowledgements

I want to give my thanks to Miguel for giving me the opportunity to research a topic I was really interested in, despite my lack of training in statistics, and for all the patience and continuous support. I also want to thank Ioanna for her mentoring, for answering all my questions about the methods and helping me with the scripts. I want to say thank you to Joan, María and Leo, I really enjoyed the group meetings and having you teach me about your research. Thanks for making my stay at CRAG so pleasant and I wish you good luck with your projects! I'm also grateful for how warm and welcoming was everyone else: Jesús, Cris, María L, Antonia, Elena, Endika, Yron, Oriol, Manuel, Alice, Magí, thanks for all the moments together! And lastly, I thank my parents for supporting me during what has been a really long year in Barcelona.