# Multi-trait and multi-environment genomic assisted predictions using the SKM library

## Talk given to the Nacional Association of Plant Breeders (NAPB), USA.

Dr. Osval Antonio Montesinos López
oamontes1@ucol.mx
Facultad de Telemática
Universidad de Colima
México

May, 15, 2023.

# Outline

- Introduction to multi-trait and multi-enviroment analysis

- Introduction to the SKM library

- Illustrative examples for implementing multi-trait, multi-environment genomic assisted predictin models in the SKM library under Bayesian methods, generalized linear models (GLM) and Partial least squares (PLS) methods.

# Introduction

The R code is available at:

https://github.com/osval78/SKM_Multivariate_NAPB

# Introduction to multi-trait analysis

- Most of the research done for genomic selection (GS) has been aimed at developing univariate-trait (UT) models and very little at developing multi-trait (MT) models.

- UT models are trained to predict the value of a single continuous (continuous, binary, categorical or count) phenotype in a testing dataset, while MT models are trained to predict at least two traits simultaneously.

# Introduction to multi-trait analysis

- *MT models should be preferred since they allows to*:

1) represent complex relationships between traits more efficiently,

2) exploit not only the correlation between lines, but also the correlation between traits,

3) are much more interpretable than a series of UT models,

4) are more computationally efficient to train than multiple UT models individually.

# Introduction to multi-trait analysis

5) improve index selection because they allow more precise estimates of random effects of lines and genetic correlation between traits,

6) can improve indirect selection because they increase the precision of genetic correlation parameter estimates between traits, and

7) improve the power of hypothesis testing better than UT models. Reducing type I and II errors. A type I error (false-positive) occurs if an investigator rejects a null hypothesis that is actually true in the population; while a type II error (false-negative) occurs if the investigator fails to reject a null hypothesis that is actually false in the population.

# Introduction to multi-trait analysis

For these reasons, MT models produce in general more accurate parameter estimates and better predictions than UT models.

For example, Schulthess et al. (2017) found empirical evidence that MT models improve parameter estimates.

Calus and Veerkamp (2011), Jia and Jannink (2012), and other authors found that MT models outperform UT models in terms of prediction accuracy.

# Introduction to multi-trait analysis

However, these authors also found that when the correlation between traits is low using MT models, it is not really advantageous since the lower the degree of relatedness between traits, the lower the benefits of MT models.

MT models are also known as multivariate analyses in statistical literature, and due to their clear advantages over UT models, they have been applied for solving a great diversity of problems in areas like environmental science, education, chemistry, telecommunications, psychology, medicine, communications, engineering, food science, among others.

# Introduction to multi-trait analysis

The use of MT models is not as popular as the use of UT models due to some of the following reasons:

1) there is less software available for performing MT analyses.

2) fitting MT models is computationally intensive and much more demanding than fitting UT models.

3) MT models are complex, as traits and variables have different response patterns in different environments and therefore create very complex genotype×environment interactions (G×E).

# Introduction to multi-trait analysis

4) MT results are based on more assumptions than UT results and may be difficult to assess and achieve.

5) MT models increase the problems of convergence when they are fitted with classic methods like maximum likelihood or restricted maximum likelihood, among others.

6) Analyzing MT models in the context of GS is very challenging due to the size and complexity of the underlying datasets, which nowadays is common in many breeding programs; for these reasons, MT models require much more computational effort than UT models.

# Introduction to multi-trait analysis

In general MT models are of paramount importance since allows breeders to select individuals that excel in multiple traits simultaneously, leading to more efficient and effective breeding programs.

For this reason, by combining multi-trait and multi-environment analysis, GS can provide breeders with a more comprehensive and accurate prediction of an individual's performance potential. It helps identify genotypes that exhibit superior performance across a range of traits and environments, leading to the development of improved varieties or breeds.

# Introduction to multi-trait analysis

Additionally, it enables breeders to select for specific trait combinations or optimize breeding strategies for different target environments, ultimately accelerating the rate of genetic gain and enhancing the efficiency of breeding programs.

However, due to the fact that GS is a predictive methodology still the implementation of multi-trait, multi-environment genomic assisted predictions in many breeding programs is challenging.

For this reason, some research is being conducted to facilitate the implementation of multi-trait, multi-environment models to improve the accuracy of the GS methodology.

# Introduction to multi-trait analysis

However, statistical machine learning methods are key for the sucessful implementation of the GS methodology.

However, due to lack of time and not enought training, it is difficult for breeders and scientists of related fields to learn all the fundamentals of prediction algorithms.

Fortunately, with smart or highly automated software, it is possible for these professionals to appropriately implement any state-of-the-art statistical machine learning method.

# SKM library

# *Sparse kernel methods (SKM) library*

- SKM is an R library that allows us to implement seven of the most powerful state-of -the-art algorithms.

  - Bayesian methods
  - Random Forest (RF)
  - Support vector Machine (SVM)
  - Generalized linear models (GLM)
  - Grandient boosting machine (GBM)
  - Partial Least Squares (PLS)
  - Feed-forward artificial neural networks

# *Sparse kernel methods (SKM) library*

- SKM not implement the algorithms itself but internally uses some already popular libraries for this purpose, which are known to have efficient and complete implementations of the algorithms.

- SKM use the same format for implementing 6 out of the 7 state of the art statistical machine learning algorithms.

- For algorithms with hyperparameters, SKM has an easy way to specify the conditions for hyperparameter tuning.

## *Sparse kernel methods (SKM) library*

- Also, has many easy-to-use functions for computing metrics for evaluation of prediction accuracy.

- Additionally provides many function for implementing many cross-validation strategies and for making summaries of prediction accuracy.

# *Hyperparameter tuning in statistical machine learning algorithms*

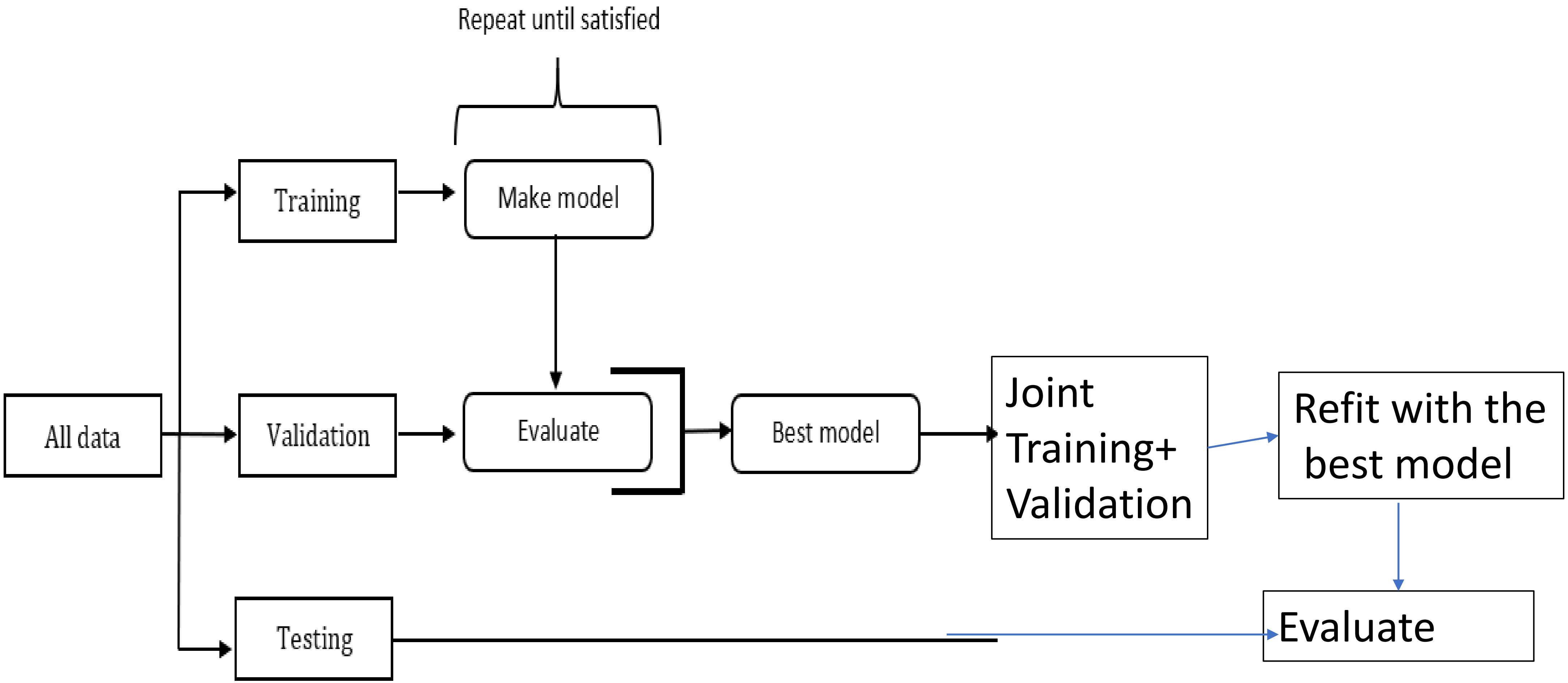
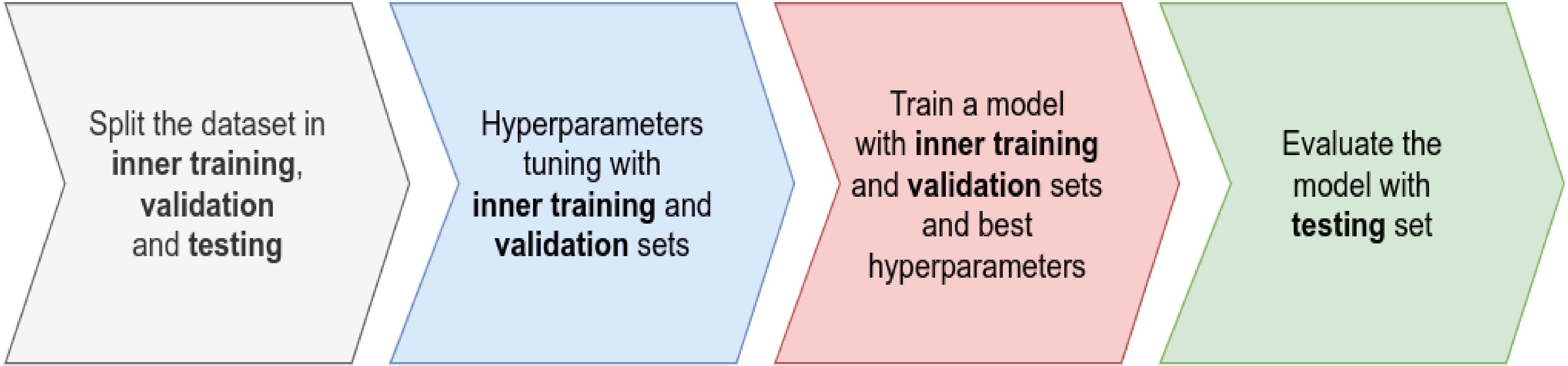


Figure 5.9. Schematic representation of the training, validation and testing sets adapted by Cook (2017).

# *Hyperparameter tuning in statistical machine learning algorithms*



Split the dataset in **inner training**, **validation** and **testing**

Hyperparameters tuning with **inner training** and **validation** sets

Train a model with **inner training** and **validation** sets and best hyperparameters

Evaluate the model with **testing** set

*Figure 1: Machine learning algorithm evaluation in one iteration of cross validation.*

# Installation of the SKM library

```
# Script to install all the required packages for SKM library

is_installed <- function(package) {

  return(all(package %in% rownames(installed.packages())))

}

is_windows_os <- function() {

  return(.Platform$OS.type == "windows")

}

ok <- TRUE

if (is_windows_os()) {

  if (!is_installed("installr")) {

    install.packages("installr")

  }

   ok <- install.Rtools(check = TRUE, check_r_update = FALSE)

}
```

# Installation of the SKM library

```
if (ok) {
  if (!is_installed("devtools")) {
    install.packages("devtools")
  }

  devtools::install_github("gdlc/BGLR-R", upgrade = "default")
  devtools::install_github("rstudio/tensorflow", upgrade = "default")
  devtools::install_github("rstudio/keras", upgrade = "default")
  devtools::install_github("brandon-mosqueda/SKM", upgrade = "default")
}
```

# Installation of the SKM library

If devtools can not be installed try:

https://cran.r-project.org/bin/windows/Rtools/rtools42/files/rtools42-5355-5357.exe

# Kernel implementation on the SKM library

- One important aspect of the SKM library is the availability of kernels to capture non-linear patterns in data, giving the library its name.

- In the SKM library, you can find seven kernels that can be implemented: Linear, Polynomial, Sigmoid, Gaussian, Exponential, Arc-Cosine 1 and Arc-Cosine L (with L = 2, 3, …).

## Evaluation of Prediction accuracy

*Metrics for regression problems available in SKM*

Mean Squared Error (`mse`),

Root Mean Squared Error (`rmse`),

Normalized Root Mean Squared Error (`nrmse`),

Mean Absolute Error (`mae`)

Mean Arctangent Absolute Percentage Error (`maape`)

`numeric_summary()` provides all the above numeric metrics

# Evaluation of Prediction accuracy

## *Metrics for classification problems available in SKM*

Accuracy (accuracy),

precision (precision),

recall (recall),

sensitivity (sensitivity),

specificity (specificity),

Brier's score (brier_score).

Details of most of these metrics can be found in Montesinos-López et al. (2022a)

# Evaluation of Prediction accuracy

## *Metrics for classification problems available in SKM*

F1-score (f1_score),

Kappa coefficient (kappa_coeff),

Matthews coefficient (matthews_coeff),

ROC area under the curve (roc_auc) and

precision-recall area under curve (pr_auc)

categorical_summary() provides all the above categorical metrics

# Evaluation of Prediction accuracy

## *Specific functions for genomic prediction*

gs_summaries is a function specific for the context of genomic prediction

gs_summaries expects a data frame with 5 columns: *Fold*, the fold number; *Line*, the line; *Env*, the environment; *Observed* and *Predicted*.

The type of response variable is automatically inferred so numeric or categorical metrics are computed and reported depending on that.

This function returns three different summaries computed by line, by environment and by fold.

# Evaluation of Prediction accuracy

The following block of code shows how we can use the function `gs_summaries` with the predictions in the data frame generated.

```
# Errors metrics for prediction performance
Summaries <- gs_summaries(Predictions)
# Print summaries by line, environment and fold
Summaries$line
Summaries$env
Summaries$fold
```

## Types of cross-validation

**K-fold**: cv_kfold() is the function in SKM

**Stratified k-fold**: cv_kfold_strata() is the name of the function in SKM

**Random partition**: `cv_random()` is the function to implement this cross-validation in SKM.

**Random stratified partitions**: `cv_random_strata()`

# Types of cross-validation

**Leave-one-group-out**: `cv_leve_one_group_out()`

**Random line**: `cv_random_line()`

**Missing**: `cv_na()`

# Hyperparameters tuning

Five algorithms in SKM require the estimation of hyperparameters (with the exception of bayesian_model and partial_least_squares that require hyperparameters but they are computed internally efficiently).

- `tune_type`: It can be “Grid_search” or “Bayesian_optimization”.

- `tune_cv_type`: Cross validation for tuning, it can be “K_fold” or “Random” that call a random partition.

# Hyperparameters tuning

```
# Grid search
model <-SKM::generalized_linear_model(
# Predictor variables
X_training,    # Response variable
y_training,
# Tunable hyperparameters
alpha =c(0,.2,0.4,0.6,1), ###Elastic_Net
# Tune configuration parameters
tune_type = "Grid_search",    tune_folds_number = 5,
tune_grid_proportion = 1,    tune_bayes_samples_number = 10,
tune_bayes_iterations_number = 10,
 # Other algorithm's parameters    lambdas_number = 100,
standardize = TRUE)
```

# Hyperparameters tuning

```
# Bayesian optimization
 model <-SKM::generalized_linear_model(
# Predictor variables
X_training,    # Response variable
y_training,
  # Tunable hyperparameters
alpha = list(min = 0, max = 1),
# Tune configuration parameters
tune_type = "bayesian_optimization",
 tune_bayes_samples_number = 5,
 tune_bayes_iterations_number = 10,
 # Other algorithm's parameters
 lambdas_number = 100,    standardize = TRUE)
```

## Hyperparameters tuning

```r
# Show all the evaluated combinations with
its loss value
model$hyperparams_grid


# Show the best combination used for
fitting the final model
model$best_hyperparams
```

# Illustrative examples

# Illustrative examples

In this section we will cover multi-trait and multi-environment genomic assited predictions under Bayesian, GLM and PLS  models.

All examples are illustrated with the SKM library.

## Bayesian models

- In SKM we refer to Bayesian models to all those models of the BGLR library (Perez & de los Campos, 2014).

- To implement these models can be used the function bayesian_model() of SKM, but, in contrast with the other six algorithms in SKM, the expected format of the predictor consist of a list with the different components of the model.

- Furthermore, instead of partitioning the data in training and testing sets, all the data is provided and the positions used for testing are set to NA or specified in a parameter included for this purpose.

## Bayesian models

$$\boldsymbol{Y} = \boldsymbol{1}_n\boldsymbol{\mu}^T + \boldsymbol{X}_E\boldsymbol{\beta}_E + \boldsymbol{Z}_L\boldsymbol{g} + \boldsymbol{Z}_{EL}\boldsymbol{gE} + \boldsymbol{\epsilon}$$

where $\boldsymbol{Y}$ is the matrix of phenotypic response variables of order $n \times n_T$,
$\boldsymbol{\mu}^T$ is a vector of intercepts for each trait of length $n_T$,
$\boldsymbol{\beta}_E$ is the matrix of beta coefficients for environments of order $I \times n_T$,
$\boldsymbol{g} \sim MN_{J\times n_T}(\boldsymbol{0}, \boldsymbol{G}, \boldsymbol{\Sigma}_T)$,
$\boldsymbol{gE} \sim MN_{JI\times n_T}\left(\boldsymbol{0}, \boldsymbol{X}_E\boldsymbol{X}_E^T \odot \boldsymbol{Z}_g\boldsymbol{G}\boldsymbol{Z}_g^T, \boldsymbol{\Sigma}_{T_2}\right)$, where $\boldsymbol{\Sigma}_{T_2}$ is the variance-covariance matrix of traits of order $n_T \times n_T$, $\odot$ denotes the Hadamard product.
$\boldsymbol{\epsilon} \sim MN_{n\times n_T}\left(\boldsymbol{0}, \boldsymbol{I}_{IJ}, \boldsymbol{R}\right)$, where $\boldsymbol{R}$ is the residual variance-covariance matrix of order $n_T \times n_T$.

# Bayesian models

```
X <- list(
  list(x =X_e, model = “FIXED"),
  list(x =K_g, model = "BGBLUP"),
  list(x =K_ge, model = "BGBLUP")
)
# Sample of 20% of the observations used as testing set
testing_indices <- sample(nrow(X), nrow(X) * 0.2)

SKM::bayesian_model(
  X,
  y,
```

# Bayesian models

```
 iterations_number = 100,
 burn_in = 50,
 thinning = 5,

testing_indices = testing_indices)
```

# Bayesian models

**Examples:**

Ex_Bayesian_1_Multi_trait_Continous_BRR_New_SVD

Ex_Bayesian_2_Multi_trait_Continous_BRR_No_Interaction_SVD

Ex_Bayesian_3_Multi_trait_Continous_GBLUP

Ex_Bayesian_4_Multi_trait_Continous_GBLUP_No_Interaction

Ex_Bayesian_5_Multi_trait_Continous_GBLUP_Kernel

The R code is available at:

https://github.com/osval78/SKM_Multivariate_NAPB

# Conclusions

- Bayesian methods are powerful for prediction since they incorpórate prior information. The prior could reflect:
  - Uncertainty about a parameter that is actually fixed OR
  - The variety of values that a truly stochastic parameter could take.

- They are robust methods that without a complex tuning process produces competitive predictions.

- They allow to specified specific model assumptions and check model quality and sensitivity to these assumptions in a straightforward way.

- They belong to the class of linear models, but also they work with kernels and in this way they are able to capture non-linear patterns in the data.

- No convergents problems are faced with Bayesian methods, but they are more demanding in terms of computational resources.

- Bayesian methods require a bit more mathematical formalism, but with friendly software they can be implemented by many users of different backgrounds.

# Generalized linear models (GLM)

Under generalized linear models, we use a set of $p$ predictor variables and a response variable to fit a model of the form:

*Distribution*: $y_i$ is distributed among those which are considered exponential families of probability distributions (like normal, binomial, Poisson, etc.)

*Linear predictor*: $\eta_i = \beta_0 + X_{i1}\beta_1 + \cdots + X_{ip}\beta_p$

*Link function*: $\mathrm{g}(\mu_i) = \eta_i$

where $y_i$ is the response variable for the i-th individual (sample), $x_i$ is the predictor variable, $\beta_0$ denotes and intercept term, $\beta_i$ is the beta coefficient corresponding to the predictor $x_i$.

# Generalized linear models (GLM)

The loss function that is minimized is

$$minimize\{-\ell(\boldsymbol{\beta};\boldsymbol{y}) + 0.5\lambda\sum_{j=1}^{p}\{(1-\alpha)\beta_j^2 + \alpha|\beta_j|\}$$

Where $\lambda$ is the penalization parameter, when $\alpha$=0 a Ridge penalization is implemented, while when $\alpha$=1 a Lasso penalization is implemented and when 0<$\alpha$ <1 the Elastic Net penalization is implemented. Can be implemented prediction models for continuous response variables (identity link function) for binary (logit link function) for categorical (generalized logit link function) and for counts (exponential logit link function) (Friedman, Hastie, & Tibshirani, 2010).

# Generalized linear models (GLM)

- Under the SKM library for continuous response variables, it is possible to train multi-trait models.

- The function to implement all these models is the generalized_linear_model() function.

- The required parameters for this function are shown in the following block of code:

# Generalized linear models (GLM)

```
SKM::generalized_linear_model(
   X_training, y_training,

  # Tunable hyperparameters
  alpha = 1, ###Lasso Regression

  # Tune configuration parameters
  tune_type = "Grid_search",
  tune_folds_number = 5,
  tune_grid_proportion = 1,
  tune_bayes_samples_number = 10,
  tune_bayes_iterations_number = 10,
```

# Generalized linear models (GLM)

```
# Other algorithm's parameters
  lambdas_number = 100,

  # Seed for reproducible results
  seed = NULL,
  verbose = TRUE
)
```

## Generalized linear models (GLM)

**Examples:**

Ex_GLM_1_Bay_Opt_Multi_Trait

Ex_GLM_2_Bay_Opt_Multi_Trait_NO_GE

Ex_GLM_3_Grid_Search_Multi_Trait

Ex_GLM_4_Grid_Search_Multi_Trait_NO_GE

Ex_GLM_5_Lasso_Multi_Trait

**The R code is available at:**

https://github.com/osval78/SKM_Multivariate_NAPB

# Conclusions

- Generalized Linear Models (GLM) are a power framework for implementing GS models for continous, binary, categorical and count response variables.

- Also, in the SKM library can be implemented multi-trait models for continous outcomes.

- In the SKM library can be implemented the following three types of penalization under GLM: Ridge, Lasso and Elastic net.

- The implementation of GLMs is quite efficient in terms of time of implementation.

- However, GLMs are unable to capture non linear patterns in the data.

- The tuning process is quite efficient for this models since only the penalization hyperparameter and alpha parameters are tuned which is done quicly in the SKM library.

# Partial Least Squares (PLS)

- Partial Least Square (PLS) regression is one of the most popular in biological sciences, because it can model complex biological events, it is flexible for considering different factors, and it is unaffected by data collinearity.

- For this reason, some authors suggest that PLS is a potentially valuable method for modeling high-dimensional biological data (as derived from genomics, proteomics and peptidomics) (Palermo et al., 2009).

- PLS can model multiple responses, while efficiently dealing with multicollinearity.

# Partial Least Squares (PLS)

## Examples:

Ex_PLS_1_Multi_trait_Continous_New_SVD

Ex_PLS_2_Multi_trait_Continous_New_SVD_NO_GE

# Conclusions

- PLS regression is somewhat close to Principal Components regression (PCR).

- Like PCR, PLS regression involves projecting the response onto uncorrelated components (i.e. linear combinations of predictors).

- Unlike PCR, the way PLS components are extracted is by taking into account the response variable.

- We can conveniently re-express the solution in terms of the original predictors.

- PLS regression is not based on any optimization criterion. Rather it is based on an interative algorithm (which converges)

- Simplicity in its algorithm: no need to invert any matrix, no need to diagonalize any matrix. All you need to do is compute simple regressions. In other words, you just need inner products.

- Also works for the multivariate response variables of various responses.

- Handles cases where we have more predictors than observations ($p \gg n$).

- It is quite efficient for correlated inputs.

# Areas of opportunities in multi-trait and multi-environment modelling

- To develop more efficient models from the computational point of view.

- To develop more efficient models to capture better the correlation between environments and traits.

- To develop models for mixed response variables (Binary, continuous, categorical and counts).

- To develop these type of models for bigger data sets.

# Final comments

MT models are progressively assuming a pivotal position within the realm of data analysis for plant and animal genomic-assisted breeding due to:

(1) the continuous and rapid advancement in computational power,

(2) the substantial advantages offered by MT models in refining parameter estimates and enhancing prediction accuracy.

For this reason, MT models are key to the timely identification and meticulous selection of the most advantageous candidate genotypes.

# Final comments

MT models provide advantages over single trait modeling, facilitating enhanced prediction accuracy through the incorporation of correlated traits, as well as for obtaining an optimal and simplified total merit selection index (Okeke et al. 2017).

In scenarios where the aim is to predict intricate or economically burdensome traits that exhibit a correlation with less costly secondary traits, the incorporation of MT models can prove instrumental in formulating improved strategies for genomic selection

# Final comments

Similarly, improvement of the accuracy of prediction for low-heritability key traits can follow from the use of high-heritability secondary traits (Jia & Jannink 2012; Muranty et al., 2015).

Furthermore, this can be combined with the information of traits obtained using the speed breeding methodology to shorten the breeding cycles and accelerate breeding programs (Ghosh et al., 2018; Watson et al., 2019).

# Final comments

While the advantage of the multi-trait model is clearly documented, larger data sets and more computing resources are required, as there are additional parameters that need to be estimated (genetic and error covariances), which may affect the accuracy of genomic prediction.

Additionally, convergence problems often arise when implementing complex mixed linear models and especially when small data sets are used, but this issues are not a problem under a Bayesian framework.

# Conclusions

The SKM library provides an opportunity to utilize various statistical machine learning techniques for multi-trait, multi-environment Genomic prediction without the need for extensive expertise in statistics and programming.

Additionally, in SKM the users can incorporating other types of omics data, such as genomic, phenomics and other omics information, which has been shown to enhance predictive accuracy based on empirical evidence.

# Conclusions

In order to democratize the implementation of genomic prediction, there is a need for software that is easier to use, more automatic, and open source.

However, still is challenging the implementation of multi-trait, multi-environment genomic assisted predictions in the context of many traits, large inputs and small sample size.

For these reason more research, effort and resources are required to solve the current challenges.

Questions ?

Thanks!

# REFERENCES

Montesinos-López A, Montesinos-López OA, Montesinos-López JC, Flores-Cortes CA, de la Rosa R, Crossa J. (2021). A guide for kernel generalized regression methods for genomic-enabled prediction. Heredity (Edinb). 126(4):577-596. doi: 10.1038/s41437-021-00412-1.

Montesinos-López O. A., Montesinos-López A., Crossa, J. (2022a). Multivariate Statistical Machine Learning methods for genomic prediction. Springer Nature, Switzerland, ISBN: 978-3-030-89012-4.

Montesinos-López, O.A., Mosqueda-González, B.A., Palafox-González, A., Montesinos-López, A. and Crossa, J. (2022). A General-Purpose Machine Learning R Library for Sparse Kernels Methods With an Application for Genome-Based Prediction. Front. Genet. 13:887643. doi: 10.3389/fgene.2022.887643

# References

- Montesinos-López, O. A., Montesinos-López, A., Crossa, J., Toledo, F. H., Pérez-Hernández, O., Eskridge, K. M., and Rutkoski, J. (2016). A genomic Bayesian multi-trait and multi-environment model. *G3: Genes, Genomes, Genetics*, *6*(9), 2725-2744.

- Montesinos-López, O. A., Montesinos-López, A., Crossa, J., Cuevas, J., Montesinos-López, J. C. Salas-Gutiérrez, Z., Lillemo, M., Philomin, J. and Singh, R. (2019). A Bayesian Genomic Multi-output Regressor Stacking Model for Predicting Multi-trait Multi-environment Plant Breeding Data. G3-Genes Genomes Genetics. 9(10): 3381-3393.

- Montesinos-López, O. A., Montesinos-López, J.C., Montesinos-López, A., Ramírez-Alcaraz, J.M., Poland, J., Singh, R., Dreisigacker, S., Crespo, L., Mondal, S., Govidan, V., Juliana, P., Huerta-Espino, J., Shrestha, S., Varshney, R. K., Crossa, J. (2022). Bayesian multitrait kernel methods improve multi-environment genome-based prediction, G3 Genes|Genomes|Genetics, 12(2), jkab406.

# References

- Montesinos-López, O. A., Montesinos-López, A.  and Crossa, J. (2022a).  Linear Mixed Models. In Multivariate Statistical Machine Learning Methods for Genomic Prediction; Montesinos López, O.A., Montesinos López, A., Crossa, J., Eds.; Springer International Publishing: Cham, Switzerland, 2022; pp. 141–170, ISBN 978-3-030-89010-0.

- Montesinos-López, O. A., Montesinos-López, A.  and Crossa, J. (2022b). Bayesian Genomic Linear Regression. In Multivariate Statistical Machine Learning Methods for Genomic Prediction; Montesinos López, O.A., Montesinos López, A., Crossa, J., Eds.; Springer International Publishing: Cham, Switzerland, 2022; pp. 171–208, ISBN 978-3-030-89010-0.

- Okeke, U. G., Akdemir, D., Rabbi, I., Kulakow, P., & Jannink, J. L. (2017). Accuracies of univariate and multivariate genomic prediction models in African cassava. *Genetics Selection Evolution*, 49(1), 88.

- Atanda, S. A.,  Steffes, J.,  Lan, Y., Bari, Md. A.,  Kim, J.,  Morales, M.,  Johnson, J.,  Saludares, R. A.,  Worral, H.,  Piche, L.,  Ross, A.,  Grusak, M. A., Coyne, C. J.,  McGee, R. J., Rao, J.,  Bandillo, N. (2022). Multi-trait genomic prediction improves selection accuracy for enhancing seed mineral concentrations in pea (Pisum sativum L.). doi: https://doi.org/10.1101/2022.04.11.487944

# References

- Jia, Y., & Jannink, J. L. (2012). Multiple-trait genomic selection methods increase genetic value prediction accuracy. *Genetics*, 192(4), 1513-1522.

- Muranty, H., Troggio, M., Sadok, I. B., Al Rifaï, M., Auwerkerken, A., Banchi, E., ... & Kumar, S. (2015). Accuracy and responses of genomic selection on key traits in apple breeding. *Horticulture research*, 2(1), 1-12.

- Ghosh, S., Watson, A., Gonzalez-Navarro, O. E., Ramirez-Gonzalez, R. H., Yanes, L., Mendoza-Suárez, M., ... & Hafeez, A. (2018). Speed breeding in growth chambers and glasshouses for crop breeding and model plant research. *Nature protocols*, 13(12), 2944-2963.

- Watson, A., Hickey, L. T., Christopher, J., Rutkoski, J., Poland, J., & Hayes, B. J. (2019). Multivariate Genomic Selection and Potential of Rapid Indirect Selection with Speed Breeding in Spring Wheat. *Crop Science*, 59(5), 1945-1959.