Module 3: Optimization of Training Sets Using R

Fundamentals of Genomic Prediction and Data-Drive Crop Breeding

(August 4-8, 2025)

Waseem Hussain
Senior Scientist-I
International Rice Research Institute
Rice Breeding Innovations Platfrom
waseem.hussain@irri.org
whussain2.github.io

Mahender Anumalla
Scientist-I
International Rice Research Institute
South-Asian Hub, Hyderabad
m.anumalla@irri.org

July 25, 2025

Contents

Load the Libraries

```
> # Load the Required Libraries
>   rm(list=ls()) # Remove previous work
>   library(AGHmatrix)
>   library(STPGA) # Package for Training Optimization
>   library(FactoMineR)
>   library(factoextra)
```

Here we will use R package **STPGA** to demonstrate how we can get optimized training set for performing genomic predictions**Additional Information**. We will use **CD Mean** to get the training set. There are various algorithms including **PEV Mean** and others to use for extracting the training. set.

Load Marker Data

The marker data is **1KRiCA** with 887 high quality SNP markers. The data is filtered and low quality markers and minor allele frequency markers has been already removed. The data is on **1**, **0** and **-1** format. For example for SNP1 if we have allele call as **A/T**, then we will have 3 types of genotypes **AA**, **TT** and **AT**. For this SNP, **[1==AA, 0==AT and TT== -1**.

More on filtering on How to filter low quality markers can be found here **How to Filter the SNP markers**

We will recode the marker data to **[2==AA, 1==AT and TT==0** formate for building the G matrix in **AGHMatrix** R package.

```
> # Genotype data
>   geno<-readRDS("./Data/geno.oyt.filtered.rds") # Upload marker data
>   dim(geno) # Check dimensions
```

```
[1] 616 887
```

```
>   geno[is.na(geno)] <-0
> # Recode the matrix in 2,1 and 0 numeric format
>   geno[geno==1]<-2
>   geno[geno==-1]<-0
>   geno[geno==0]<-0
>   kable(head(geno)[1:6,1:2])
```

	IRRI_SNP0001_CHR01_194844	IRRI_SNP0002_CHR01_375814
IR129391.B.12.B.3.1	0	0
IR129391.B.19.B.2.1	0	0
IR129391.B.44.B.4.1	2	2
IR129391.B.46.B.1.1	2	2
IR129391.B.6.B.4.1	0	0
IR129391.B.9.B.4.1	0	0

Build the Genomic Matrix

We will use AHGmatrix R package to build the various Relationship Matrices. More details on this package can be found here **AHGmatrix** and on **Github**.

- Function **Gmatrix()** handles the molecular-marker matrix and builds the relationship matrices.
- Molecular markers data should be organized in a matrix format (individuals in rows and markers in columns) coded as 0 (BB), 1 (AB), 2 (AA) and missing data value (numeric or NA).
- Arguments in the function to control are: **Minor allele frequency** (maf), **Threshold** for missing data, and **Method** which one should be the method used to build the kernel.
- To run the example quickly we will use only small set of 200 genotypes.

```
> # Get subset of first 200 genotypes
>   geno<-geno[1:200, ] # Subset
>   geno<-as.matrix(geno) # Convert as matrix
> # Build the VanRaden 2008 G matrix
>   G_additive <- Gmatrix(SNPmatrix=geno, missingValue=NA,
+                           maf=0.05, method="VanRaden")
```

```
Initial data:
    Number of Individuals: 200
    Number of Markers: 887

Missing data check:
    Total SNPs: 887
     0 SNPs dropped due to missing data threshold of 0.5
    Total of: 887  SNPs

MAF check:
     33 SNPs dropped with MAF below 0.05
    Total: 854  SNPs

Heterozigosity data check:
    No SNPs with heterozygosity, missing threshold of =  0

Summary check:
    Initial:  887 SNPs
    Final:  854  SNPs ( 33  SNPs removed)

Completed! Time = 0.056  seconds
```

Extract the Training Set

In the *STPGA* R package the function **GenAlgForSubsetSelectionNoTest** is available to optimize the training set. The function has several arguments and here we will use few arguments to obtain the training set.

Here we will use G matirx derived above to extract the training set.

```
> # Define the number of individuals to select for the training set
>   nSel <- 60  # Can be adjusted based on population size
>
> # Optimize the training set using CDmean in the argument errorstat
>
>     result <- GenAlgForSubsetSelectionNoTest(
+       P =  G_additive,  # marker matrix or relationship matrix
+       ntoselect = nSel, # Number of individuals in the training set
+       npop = 100,  # population size for the genetic algorithm
+       nelite = 5, # population size for the genetic algorithm
+       mutprob = 0.5, # probability of mutation for each generated solution
+       mutintensity = 1,
+       niterations = 50, # number of iterations.
+       minitbefstop = 50, # number of iterations before stopping
+       tabu = FALSE,
+       plotiters = FALSE, # plot the convergence
+       lambda = 1e-6, # scalar shrinkage parameter
+       errorstat = "CDMEAN"
+     )
```

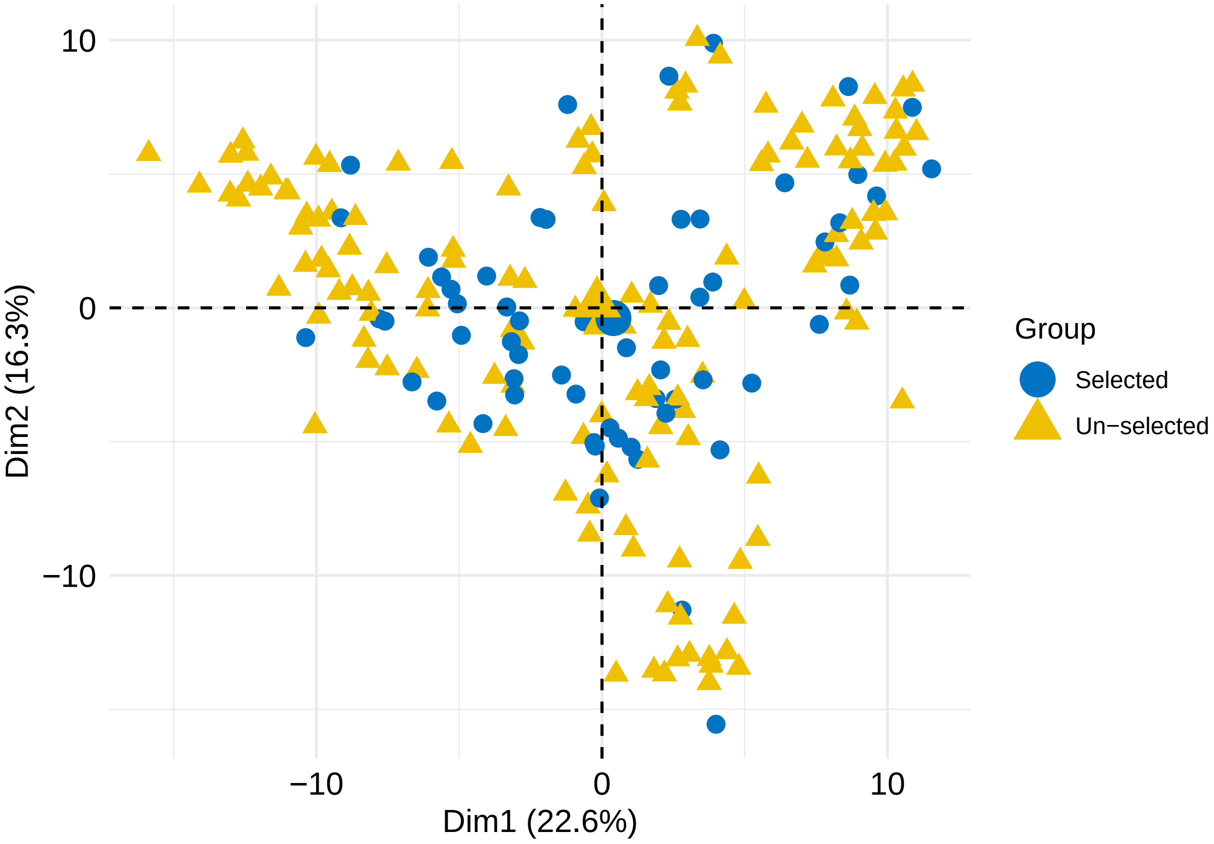
Extract the selected individuals

```
>   selected_ids <- data.frame(TrainingGenotypes=result$`Solution with rank 1`)
>   kable(head(selected_ids$TrainingGenotypes))
```

x
IR129391.B.12.B.3.1
IR129391.B.44.B.4.1
IR129391.B.6.B.4.1
IR129398.B.14.B.4.1
IR129398.B.3.B.2.1
IR129398.B.37.B.1.1

How Good Training Set Represents Whole Set

```
> #  Get lines from training set
>   line.names<-data.frame(Genotype=ifelse(row.names(G_additive)%in%selected_ids$TrainingGenotypes, "Sel
>   row.names( line.names)<-row.names(G_additive)
>
> # Perfom PCA on GM matrix
>   pca<- PCA(G_additive, graph = FALSE)
> # Get the Biplot
>   biplot.gm<-fviz_pca_biplot(pca, palette = "jco", axes = c(1, 2), geom="point",pointsize = 3,
+                          addEllipses = FALSE,col.ind = line.names$Genotype,
+                          geom.var= "none", label = "none", legend.title = "Group")+
+   theme_minimal()+
+   # add and modify the title to plot
+   theme (
+     plot.title = element_blank(),
+     # add and modify title to x axis
+     axis.title.x = element_text(color="black", size=12),
+     # add and modify title to y axis
+     axis.title.y = element_text(color="black", size=12)) +
+   # modify the axis text
+   theme(axis.text= element_text(color = "black", size = 12))
> biplot.gm
```



```
> #' ###################End########################################
```

Assignment

- Learn the various arguments annd Functions of **STPGA** R package.
- Also, run the example in **TrainSel** R package; Check Trainsel document as Example.

Additional Resources

- Maximizing the Reliability of Genomic Selection by Optimizing the Calibration Set of Reference Individuals: Comparison of Methods in Two Diverse Groups of Maize Inbreds (Zea mays L.)
- Selection of training populations with an accelerated genetic algorithm STPGA R-package

- A comparison of methods for training population optimization in genomic selection
- TrainSel: An R Package for Selection of Training Populations
- Training Set Optimization for Sparse Phenotyping in Genomic Selection: A Conceptual Overview

---

For any suggestions or comments, please feel to reach at waseem.hussain@irri.org; and m.anumalla@irri.org

---