Module 5: Genomic Selection and Dissecting G x E in R

Fundamentals of Genomic Prediction and Data-Drive Crop Breeding

(November 24-28, 2025)

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

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

Margaret Catolos
Associate Scientist
International Rice Research Institute
South-Asian Hub, Hyderabad
m.catolos@cgiar.org

November 20, 2025

Contents

Background Information

For this module, we will use a multi bi-parental mapping population derived by crossing multiple parents. The population is fixed at F_7 generation. The total number of genotypes in the population are **844**. The whole population has been genotyped with the **GBS SNP data** with total number of ~396511 SNP markers.

This mapping population has been divided into training set with 252 genotypes phenotyped across multiple enviornments for Grain yield and other traits. The rest of the 592 genotypes has only genotyped but not phenotyped. We will predict the performance of the 592 genotypes and estimate breeding values for grian yield for them.

What is Our Goal

- **Part 2**: Disect the G x E interactions and Estimate the Breeding Value

We will use R packages **Sommer** to fit different G x E models. More on this can be found here Click Link. We will aslo **BGLR** package to demonstrate the G x E dissection

NOTE: Due to IRRI's data policies, the actual names of lines is not given

Load the R Packages

```
>   library(AGHmatrix)
>   library(BGLR)
>   library(lme4)
>   library(ggplot2)
>   library(sommer)
```

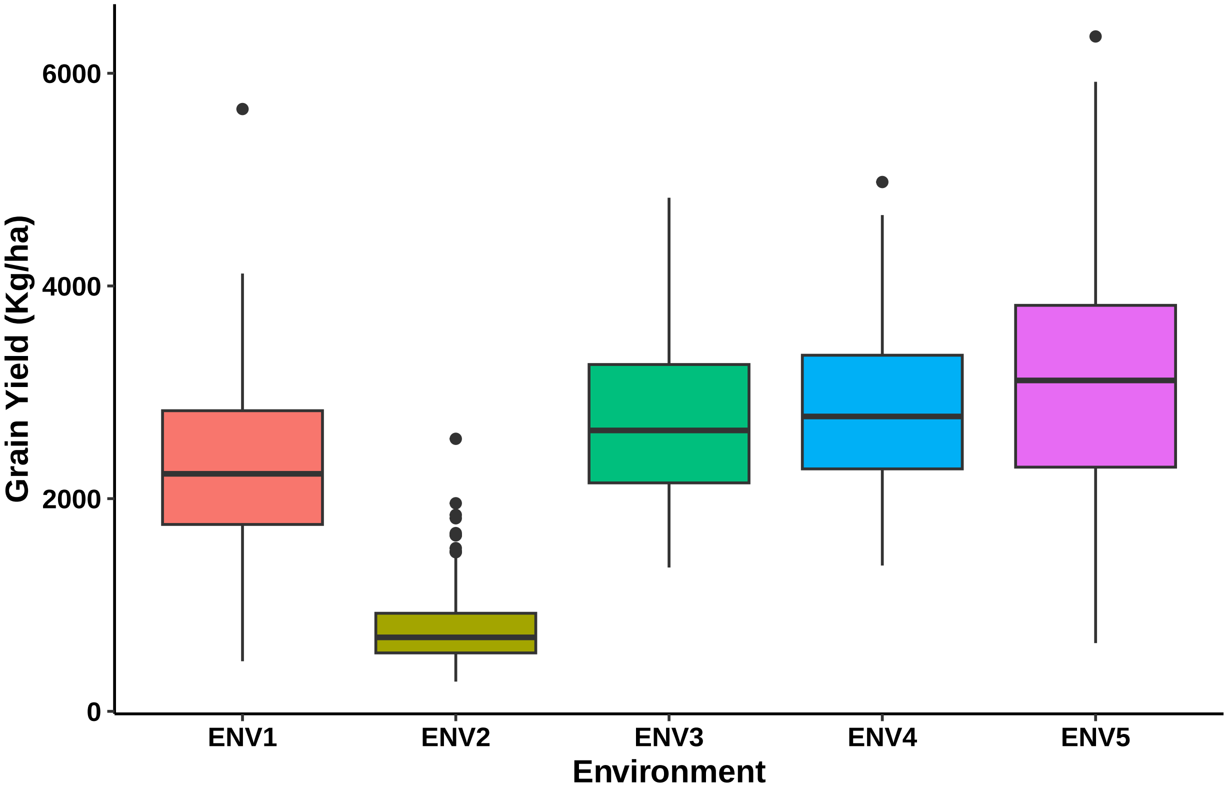
Read the Data Sets

The Data has 5 environments and has yield data. The data comes from the different locations in Bangladesh and India. BLUEs alrady extracted. We will upload the file and use it for analysis.

```
>   rm(list=ls()) # remove History
> # Read the phenotypic data
>   BLUEs.all<-read.csv(file="./Data/BLUES.ALL.csv")
```

Visualization of Data

```
> #png(file="~/boxplot.png", width=10, height =6, units = 'in',res=300)
>         boxplot<-ggplot(BLUEs.all, aes(x=Environment, y=BLUEs))+
+           geom_boxplot(aes(fill=Environment))+ # fill by timepoint to give different color
+           #scale_fill_manual(values = c("", ""))+
+           #scale_color_manual(values = c("", ""))+
+           theme_classic()+ #choose the theme for background
+           labs(title="",x="Environment", y = "Grain Yield (Kg/ha)")+#Add the labels to the plots
+           theme (plot.title = element_text(color="black", size=14, face="bold",hjust=0.5), # add and 
+                  axis.title.x = element_text(color="black", size=12, face="bold"), # add and modify t
+                  axis.title.y = element_text(color="black", size=12, face="bold")) + # add and modify
+           theme(axis.text= element_text(face = "bold", color = "black", size = 10))+ # modify the axi
+           theme(legend.position="none") # remove the theme from the plot
>           #aes(x = fct_inorder(timepoint))+ # order the levels
>           #dev.off()
>          # ggplotly(p)
>     boxplot
```



Read Genotype Data

This marker data as 844 genotypes with 396511 SNP Markers, and the file is saved as **.rds**. We will subset 252 genotypes and use it estimate the **GEBVs**.

```
>   geno<-readRDS("./Data/GBS_datav2.rds")
>   dim(geno)
```

```
[1]    844 396511
```

```
> # Match genotype with Phenotype
>   Ids<-unique(BLUEs.all$Genotype)
>   length(Ids)
```

```
[1] 252
```

```
> # Now subet the genotype Data based on IDs
>   geno<-geno[row.names(geno)%in%Ids,]
>   dim(geno)
```

```
[1]    252 396511
```

Build the G matrix

- Here we will construct the **Genomic Relationship Matrix (GRM)** using marker data. The GRM will be based on **VanRanden (2008)**.
- The steps used to create this GRM is:
  - Create a center of marker data (X matrix)
  - Create a Cross Product (XX)
  - Divide the (XX) by number of markers

$$GRM = XX^t/m$$

- More on relationship matrix can be found here Source 1, Source2
- We will use the AGHmatrix package to build G matrix.

```
>   GM<- Gmatrix(SNPmatrix=geno, missingValue=NA,
+                              maf=0.05, method="VanRaden")
```

```
Initial data:
    Number of Individuals: 252
    Number of Markers: 396511

Missing data check:
    Total SNPs: 396511
     0 SNPs dropped due to missing data threshold of 0.5
    Total of: 396511  SNPs

MAF check:
     24891 SNPs dropped with MAF below 0.05
    Total: 371620  SNPs

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

Summary check:
    Initial:  396511 SNPs
    Final:  371620  SNPs ( 24891  SNPs removed)

Completed! Time = 31.512  seconds
```

```
>   dim(GM)
```

```
[1] 252 252
```

Fit Various G x E models

The more description on G x E models I recommend going over these resources Resource 1 and Resource 2

Main MET Model

Here we will fit model using sommer R package.

The model assumes GxE doesn't exist and that the main genotype effect plus the fixed effect for environment is enough to predict the genotype effect in all locations.

```
> # Fit Model
>   gs.model1<- mmes(BLUEs~Environment, # Environment Fixed
+                  random= ~ vsm(ism(Genotype), Gu=GM), #vsm is covariance function to assign matrices
+                  rcov= ~ units, # Residuals have no-covariance
+                  data=BLUEs.all, verbose = FALSE)
> # Get summary
>   summary(gs.model1)
```

```
============================================================
         Multivariate Linear Mixed Model fit by REML
**********************  sommer 4.4  **********************
============================================================
         logLik      AIC      BIC Method Converge
Value -114.9267 239.8534 265.5438     AI     TRUE
============================================================
Variance-Covariance components:
                  VarComp VarCompSE Zratio Constraint
Genotype:GM:mu:mu   20735      7139  2.905   Positive
units:mu:mu        535135     22767 23.505   Positive
============================================================
Fixed effects:
          Estimate Std.Error t.value
Intercept   2315.7     46.18  50.146
ENV2       -1537.6     65.24 -23.570
ENV3         433.7     65.24   6.648
ENV4         525.7     65.24   8.058
ENV5         780.3     65.24  11.960
============================================================
Use the '$' sign to access results and parameters
```

```
> # Extract the GEBVs (random effects)
>   estimated.all<-data.frame(GEBVs= gs.model1$u) # stored in u
>   estimated.all$GEBVs<-estimated.all$GEBVs+ gs.model1$b[1] # adding intercept
>   gs.model1$AIC # Check AIC and BIC values
```

```
[1] 239.8534
```

```
>   gs.model1$BIC
```

```
[1] 265.5438
```

```
>   kable(head(estimated.all)) # View as table
```

	GEBVs
Genotype_10162	2051.910

	GEBVs
Genotype_10164	2431.821
Genotype_10169	2157.124
Genotype_10173	2057.640
Genotype_10175	2124.616
Genotype_10176	2174.124

MET: diagonal model (DG)

The diagonal model assumes that GxE exists and that the genotype variation is expressed differently at each location, therefore fitting a variance component for the genotype effect at each location. The main drawback is that this model assumes no covariance among locations, as if genotypes were independent.

```
> gs.model3<- mmes(BLUEs~Environment,
+                  random= ~ vsm(dsm(Environment),ism(Genotype), Gu=GM),
+                  rcov= ~ units,
+                  data=BLUEs.all, verbose = FALSE)
> summary(gs.model3)
```

```
===================================================================
             Multivariate Linear Mixed Model fit by REML
**************************  sommer 4.4  **************************
===================================================================
         logLik      AIC      BIC Method Converge
Value -20.67411 51.34823 77.03859     AI     TRUE
===================================================================
Variance-Covariance components:
                                  VarComp VarCompSE Zratio Constraint
Environment:Genotype:GM:ENV1:ENV1  133567     31523  4.237   Positive
Environment:Genotype:GM:ENV2:ENV2       0      7265  0.000   Positive
Environment:Genotype:GM:ENV3:ENV3  194228     39035  4.976   Positive
Environment:Genotype:GM:ENV4:ENV4  127188     30619  4.154   Positive
Environment:Genotype:GM:ENV5:ENV5  398478     61539  6.475   Positive
units:mu:mu                        286935     20286 14.144   Positive
===================================================================
Fixed effects:
          Estimate Std.Error t.value
Intercept   2317.2     33.86  68.432
ENV2       -1539.1     47.80 -32.197
ENV3         432.2     47.80   9.042
ENV4         524.2     47.80  10.965
ENV5         778.8     47.80  16.291
===================================================================
Use the '$' sign to access results and parameters
```

MET: unstructured model (US)

We assume that that GxE exists and that an environment-specific variance exists in addition to as many covariances for each environment-to-environment combinations. The main drawback is that is difficult to make this models converge because of the large number of variance components, the fact that some of these variance or covariance components are zero, and the difficulty in choosing good starting values. The fixed effect for environment plus the environment specific BLUP (adjusted by covariances) is used to predict the genotype effect in each location of interest.

```
>   gs.model6<- mmes(BLUEs~Environment,
+                      random= ~ vsm(usm(Environment),ism(Genotype), Gu=GM),
+                      rcov= ~ units,
+                      data=BLUEs.all, verbose = FALSE)
>   summary(gs.model6)
```

MET: compund symmetry model (CS)

The compound symmetry model assumes that GxE exists and that a main genotype variance-covariance component is expressed across all location. In addition, it assumes that a main genotype-by-environment variance is expressed across all locations. The main drawback is that the model assumes the same variance and covariance among locations.

```
>     E <- diag(length(unique(BLUEs.all$Environment)))
>     rownames(E) <- colnames(E) <- unique(BLUEs.all$Environment)
>     Ei <- solve(E)
>     Gi <- solve(GM)
>     EGi <- kronecker(Ei,Gi, make.dimnames = TRUE)
>     Ei <- as(as(as( Ei,  "dMatrix"), "generalMatrix"), "CsparseMatrix")
>     Gi <- as(as(as( Gi,  "dMatrix"), "generalMatrix"), "CsparseMatrix")
>     EGi <- as(as(as( EGi,  "dMatrix"), "generalMatrix"), "CsparseMatrix")
>     attr(Gi, "inverse")=TRUE
>     attr(EGi, "inverse")=TRUE
>     model5<- mmes(BLUEs~Environment,
+                    random= ~ vsm(ism(Genotype), Gu=Gi) + vsm(ism(Environment:Genotype), Gu=EGi),
+                    rcov= ~ units,
+                    data=BLUEs.all, verbose = FALSE)
>     summary(gs.model5)
```

Fit Same Model in BGLR

Here again we will explain in detail, y: vector of phenotypic values (Yield), $BLUES.all$ is a dataframe with columns Genotype, Environment, and Yield, Z is genotype incidence matrix GM is genomic relationship matrix (e.g., from markers) and $X$is environment design matrix. Before fitting model is BGLR we will build X and Z matrices, then we will create the Kernal Matrices of Genomic matrix and interaction matrices.

Then we will build ETA list to supply the matrices to function.

```
> # Convert Environment to design matrix
>   X <- model.matrix(~ Environment, data = BLUEs.all)
> # Convert Genotype to design matrix
>   Z <- model.matrix(~ Genotype - 1, data = BLUEs.all)
> # Genotype kernel
>   KG <- tcrossprod(Z %*%  GM)
> # GxE interaction kernel
>   KGE <- tcrossprod(X) * KG  # Element-wise multiplication
>
> # ETA list for BGLR Package
>     ETA <- list(
+       ENV = list(X = X, model = "FIXED"),
+       G   = list(K = KG, model = "RKHS"),
+       GxE = list(K = KGE, model = "RKHS")
+     )
> # Fit the Model
> modelGxE<- BGLR(y = BLUEs.all$BLUEs,
+            ETA = ETA,
+            nIter = 1000,
+            burnIn = 100,
+            thin = 2,
+            verbose = FALSE)
```

Extract the Results

```
> # Variance components
>   modelGxE$ETA$G$varU      # Genotype variance
```

```
[1] 2440.601
```

```
>   modelGxE$ETA$GxE$varU    # GxE variance
```

```
[1] 2721.165
```

```
>   modelGxE$varE            # Residual variance
```

```
[1] 570511.8
```

```
> # Breeding values (genotype effects)
>   GEBVs <- data.frame(GEBVs_All=modelGxE$ETA$G$u)
>   #BGLR directly outputs the genotype predictions as yHat
>   GEBVs2<- data.frame(modelGxE$yHat)
```

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

---

If your experiment needs a statistician, you need a better experiment - Ernest Rutherford