Module 2: Relationship Matrices in Breeding

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@cgiar.org
whussain2.github.io

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

August 3, 2025

Contents

Introduction

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**.

Load the Required Library

```
> # install.packages("AGHmatrix")
>   #install.packages("AGHmatrix")
>   library ("AGHmatrix")
>   library(DT)
```

---

1. Pedigree Relationship Matrix

- We will use AHG matrix to build the A (Numerator Relationship Matrix).
- Matrix A is constructed using path coefficients but a recursive method (Henderson, 1976).
- Function **Amatrix()** process the pedigree and build the A-matrix related to that given pedigree.
- The matrix is built based in the recursive method presented in **Mrode (2014)** and described by **Henderson (1976)**
- We will use IRRI's drought breeding Program Pedigree Data. The details on this program can be found here **Click Here**.
- The description of file is given below. Need three columns. First column genotype ID's. Second column *Male* parent and third column *Female* Parent. If you do not know parnetage just put *0*. Mkaw sure that each unique genotype is present in the first column.

DESIGNATION	Female	Male
IR64	IR5657-33-2-1	IR2061-465-1-5-5
IRRI154	IR73012-137-2-2-2	PSBRC10(IR50404-57-2-2-3)
IRRI193	IR68077-82-2-2-2-3	IR00A117
IR05N412	IR72875-94-3-3-2	IR73707-45-3-2-3
IR05N419	IR72887-34-2-1-3	IR73707-45-3-2-3
IR06N155	IR72158-11-5-2-3_IR73	IR72875-94-3-3-2
IR09A220	IR72903-121-2-1-2	IR71606-1-1-4-2-3-1-2(NSIC110)
IR10A231	IRRI143_IR73718-23-2-1	IR00A110
IR10F559	IR80410-B-197-4_IRRI14	NSICRC158
IR10N237	IR01N111_IRRI164	IR72890-81-3-2-2
IR10N271	IR01W106	IR71676-90-2-2
IR11A282	IR04A427	BR29
IR11A303	IR04A427	IR72875-94-3-3-2
IR11A306	IR04A427	IR73006-12-3-3-2
IR11N121	IR05N341_IR64680-81-2	PSBRC10(IR50404-57-2-2-3)
IR11N202	IR05N173	BR29
IR12N135	IR01N149_IR64680-81-2	FEDEARROZ50
IR12F111	IR44004-74-3-2-3-3-3	IR70181-32-PMI1-1-5-1
BRRIDHAN53	BR10(BR51-46-5)_BR23	BR847-76-1-1
BRRIDHAN55	0	0
IRRI198	MEMBERANO	PADIABANGGOGO
IR58443-6B-10-3	AT401	IR31868-64-2-3-3-3
IR66946-3R-149-1-1	IR29	POKKALI
IRRI186	IR02A127	IR64
IR02A149	IR00A107	PSBRC54(IR60819-34-2-1)
IRRI214	IR71606-1-2-1-3-2-3-1	PSBRC64(IR59552-21-3-2-2)
IRRI180	IR73718-1-2-1-3	PSBRC10(IR50404-57-2-2-3)
IRRI179	IR73008-138-2-2-2_IR68	IR72870-19-2-2-3
IRRI181	IR02A127	JANAKI
IR10M210	IRRI123	IR68144-2B-2-2-3-1-127

```
> # Read the Pedigree Data
>   ped<-read.csv(file="./Data/ped.finalv2.csv", header = TRUE)
>   head(ped)
```

DESIGNATION	Female	Male
IR64	IR5657-33-2-1	IR2061-465-1-5-5
IRRI154	IR73012-137-2-2-2	PSBRC10(IR50404-57-2-2-3)
IRRI193	IR68077-82-2-2-2-3	IR00A117
IR05N412	IR72875-94-3-3-2	IR73707-45-3-2-3
IR05N419	IR72887-34-2-1-3	IR73707-45-3-2-3
IR06N155	IR72158-11-5-2-3_IR73707-45-3-2-3	IR72875-94-3-3-2

```
> # Use Amatrix function to build pedigree matirx
>   ped.matrix<-Amatrix(ped, ploidy=2)
Verifying conflicting data...
Organizing data...
To organize the data in a fast way wasn't possible...
Trying to organize in a slow (naive) way...
Your data was chronologically organized with success.
Processing a large pedigree data... It may take a couple of minutes...
Constructing matrix A using ploidy = 2
Completed! Time = 0.02873333  minutes
>   dim(ped.matrix)
[1] 4868 4868
>   ped.matrix[1:10,1:5]
               IR64    IRRI154    IRRI193   IR05N412   IR05N419
IR64     1.07666016 0.13361454 0.01675129 0.08966118 0.03874362
IRRI154  0.13361454 1.03618324 0.03123796 0.04905295 0.05164006
IRRI193  0.01675129 0.03123796 1.00000000 0.01018584 0.01580453
IR05N412 0.08966118 0.04905295 0.01018584 1.00000000 0.29291612
IR05N419 0.03874362 0.05164006 0.01580453 0.29291612 1.00000000
IR06N155 0.08966118 0.05100608 0.01018584 0.39858466 0.17475206
IR09A220 0.31167561 0.09983473 0.02137049 0.04587312 0.04730195
IR10A231 0.22385880 0.05423607 0.01408082 0.02706859 0.02323807
IR10F559 0.22907901 0.08667803 0.02460436 0.05484275 0.04891247
IR10N237 0.06115746 0.09206831 0.02595565 0.04761322 0.10361953
```

2. Genomic Relationship Matrices

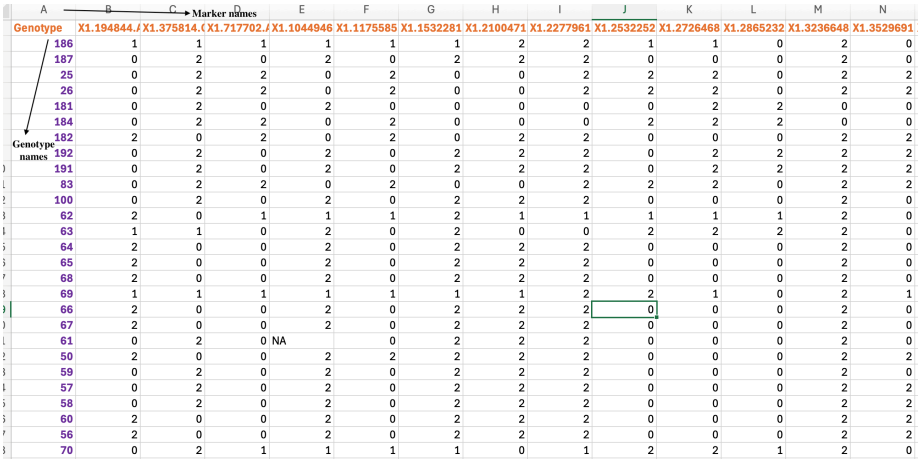
- Function **Gmatrix()** handles the molecular-marker matrix and builds the relationship matrices.
- Molecular markers data should be organized in a matrix format (individuals in rows and markers in columns) coded as 0 (BB), 1 (AB), 2 (AA) and missing data value (numeric or NA).
- Arguments in the function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

Additive Realtionship Matrix

- We will compute the additive relationship matrix based on **VanRanden (2008)**.
- The steps used to create this GRM is:
  - Create a center of marker data (Z matrix)
  - Create a Cross Product (ZZ)
  - Divide the (XX) by number of markers

$$G = \frac{ZZ'}{2\sum p_i(1-p_i)}$$

- More on additive relationship matrix based on **VanRaden (2008)** and **Yang et al. 2010**
- Sample data file is n x m matrix, where **n** is number of markers and **m** is number of genotypes.

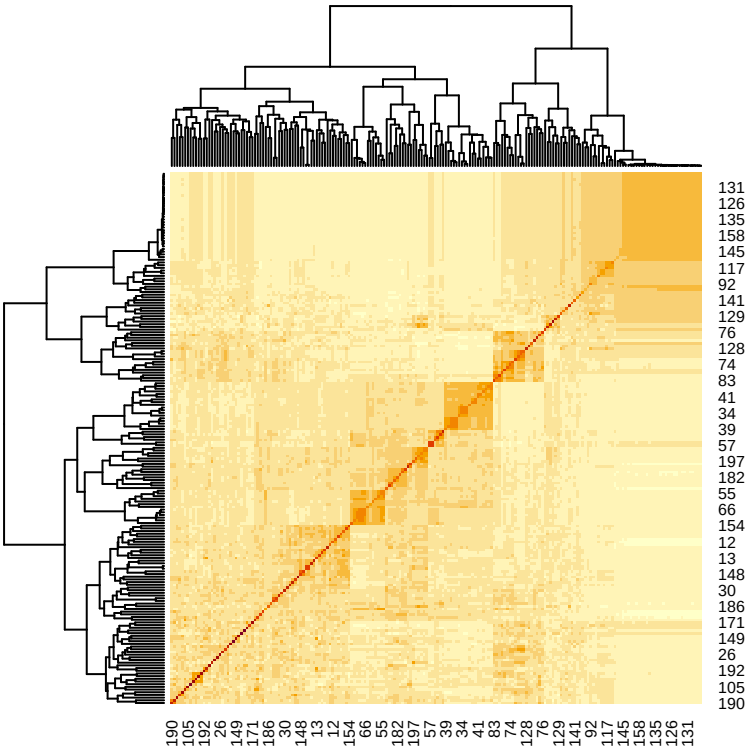


Genotype	X1.194844.A	X1.375814.C	X1.717702.A	X1.1044946	X1.1175585	X1.1532281	X1.2100471	X1.2277961	X1.2532252	X1.2726468	X1.2865232	X1.3236648	X1.3529691
186	1	1	1	1	1	1	2	2	1	1	0	2	0
187	0	2	0	2	0	2	2	2	0	0	0	2	0
25	0	2	2	0	2	0	0	2	2	2	0	2	2
26	0	2	2	0	2	0	0	2	2	2	0	2	2
181	0	2	0	2	0	0	0	0	0	2	2	0	0
184	0	2	2	0	2	0	0	0	2	2	2	0	0
182	2	0	2	0	2	0	2	2	0	0	0	2	2
192	0	2	0	2	0	2	2	2	0	2	2	2	2
191	0	2	0	2	0	2	2	2	0	2	2	2	2
83	0	2	2	0	2	0	0	2	2	2	0	2	2
100	0	2	0	2	0	2	2	2	0	0	0	2	0
62	2	0	1	1	1	2	1	1	1	1	1	2	0
63	1	1	0	2	0	2	0	0	2	2	2	2	0
64	2	0	0	2	0	2	2	2	0	0	0	2	0
65	2	0	0	2	0	2	2	2	0	0	0	2	0
68	2	0	0	2	0	2	2	2	0	0	0	2	0
69	1	1	1	1	1	1	1	2	2	1	0	2	1
66	2	0	0	2	0	2	2	2	0	0	0	2	0
67	2	0	0	2	0	2	2	2	0	0	0	2	0
61	0	2	0	NA	0	2	2	2	0	0	0	2	0
50	2	0	0	2	2	2	2	2	0	0	0	2	2
59	0	2	0	2	0	2	2	2	0	0	0	2	0
57	0	2	0	2	0	2	2	2	0	0	0	2	2
58	0	2	0	2	0	2	2	2	0	0	0	2	2
60	2	0	0	2	0	2	2	2	0	0	0	2	2
56	2	0	0	2	0	2	2	2	0	0	0	2	2
70	0	2	1	1	1	1	0	1	2	2	1	2	0

```
> # Read SNP data
>   geno<-read.csv(file="./Data/geno.data.csv", header = TRUE, row.names = 1)
>   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: 198
    Number of Markers: 1008

Missing data check:
    Total SNPs: 1008
     114 SNPs dropped due to missing data threshold of 0.5
    Total of: 894  SNPs

MAF check:
     85 SNPs dropped with MAF below 0.05
    Total: 809  SNPs

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

Summary check:
    Initial:  1008 SNPs
    Final:  809  SNPs ( 199  SNPs removed)

Completed! Time = 0.048  seconds
```

Visualize the Relationship matrix

```
>   heatmap(G_additive)
```



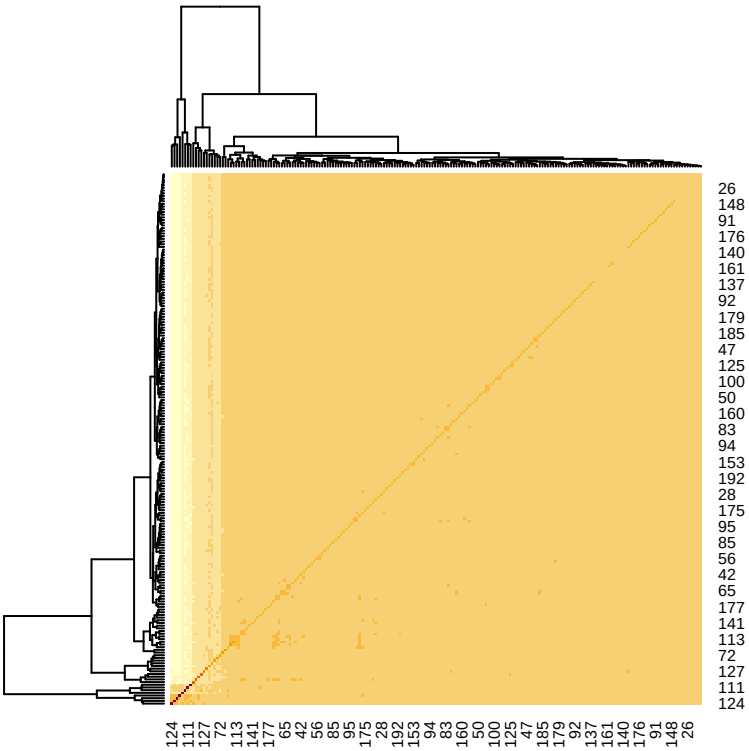
Dominance relationship matrix

- The function **Gmatrix()** can also construct the dominance relationship matrix either as proposed by **Su et al. (2012)** or as proposed by **Vitezica et al. (2013)**.

```
>   G_Dominace <- Gmatrix(SNPmatrix=geno, missingValue=NA,
+                       maf=0.05, method="Su")
Initial data:
    Number of Individuals: 198
    Number of Markers: 1008

Missing data check:
    Total SNPs: 1008
     114 SNPs dropped due to missing data threshold of 0.5
    Total of: 894  SNPs

MAF check:
     85 SNPs dropped with MAF below 0.05
    Total: 809  SNPs

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

Summary check:
    Initial:  1008 SNPs
    Final:  809  SNPs ( 199  SNPs removed)

Completed! Time = 0.052  seconds
```

Visualize the Domiance Relationship matrix

```
>   heatmap(G_Dominace)
```



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

---