Introducción a Clase Práctica

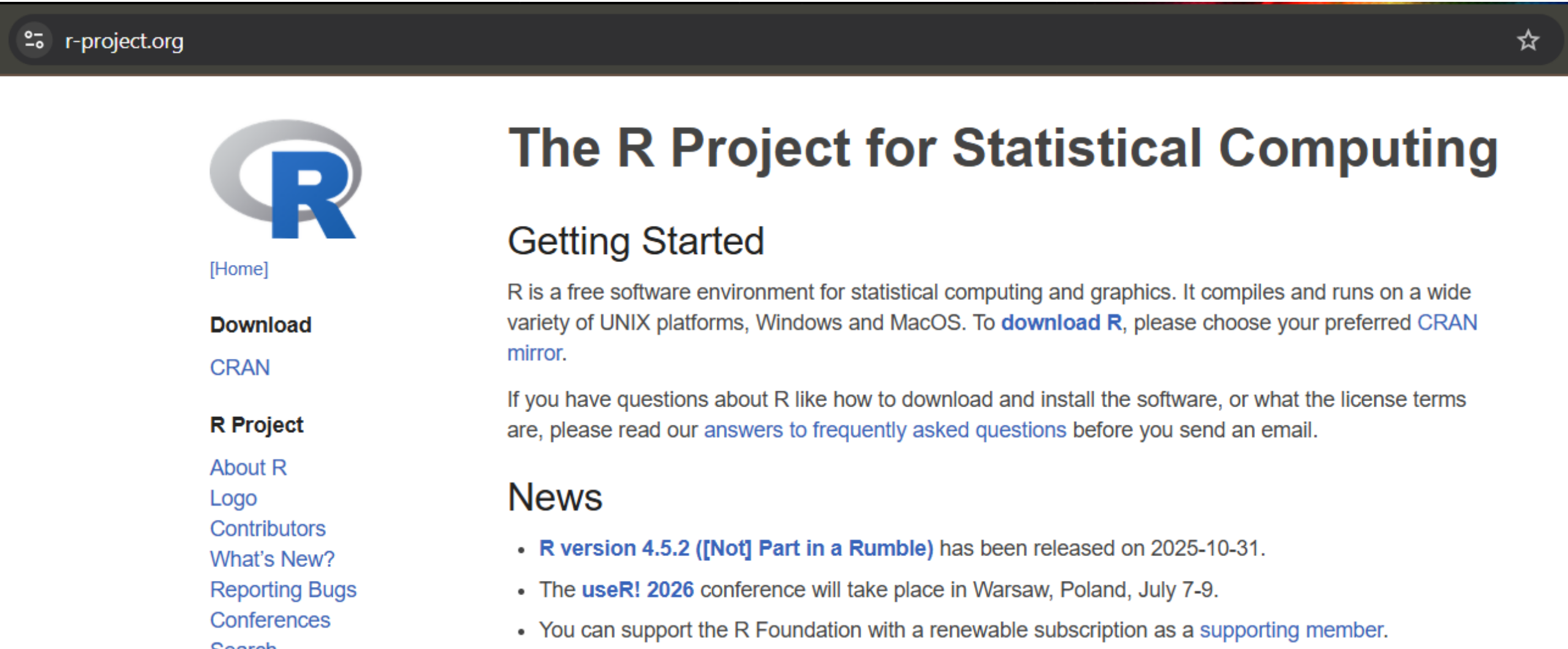
Docente: Dr. Martín Nahuel García

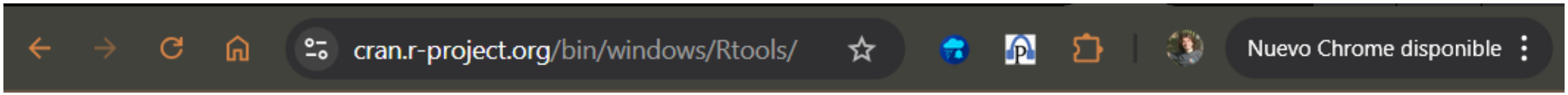
Mejoramiento Genético y Genómico Vegetal

Fecha: 11//2025

garcia.martin@inta.gob.ar

r-project.org



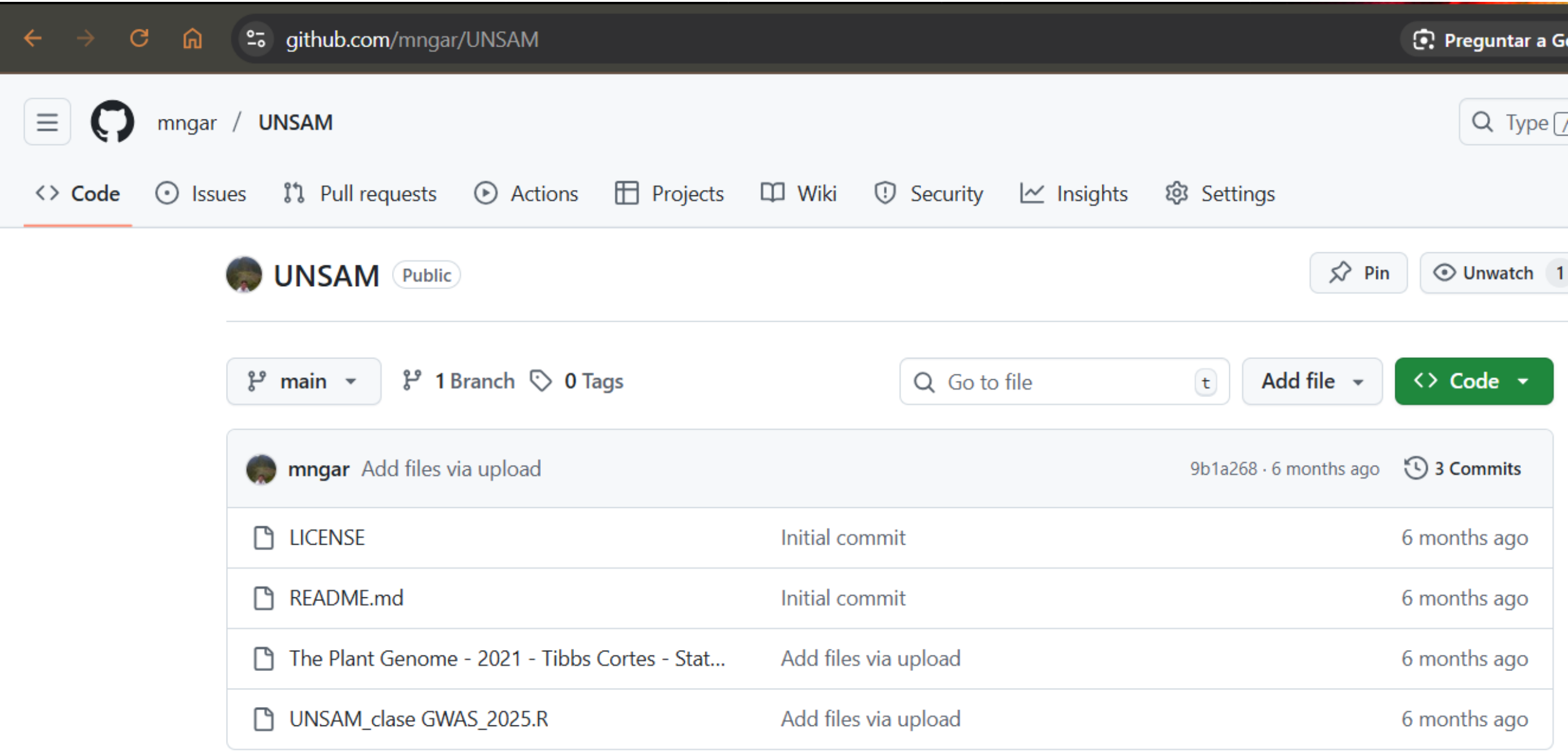


RTools: Toolchains for building R and R packages from source on Windows

Choose your version of Rtools:

RTools 4.5	for R versions from 4.5.0 (R-prerelease and R-devel)
RTools 4.4	for R versions 4.4.x (R-release)
RTools 4.3	for R versions 4.3.x (R-oldrelease)
RTools 4.2	for R versions 4.2.x
RTools 4.0	for R from version 4.0.0 to 4.1.3
old versions of RTools	for R versions prior to 4.0.0

github.com/mngar/UNSAM



github.com/mngar/UNSAM/blob/main/UNSAM_clase%20GWAS_2025.R

mngar / UNSAM

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

main | UNSAM / UNSAM_clase GWAS_2025.R

mngar Add files via upload

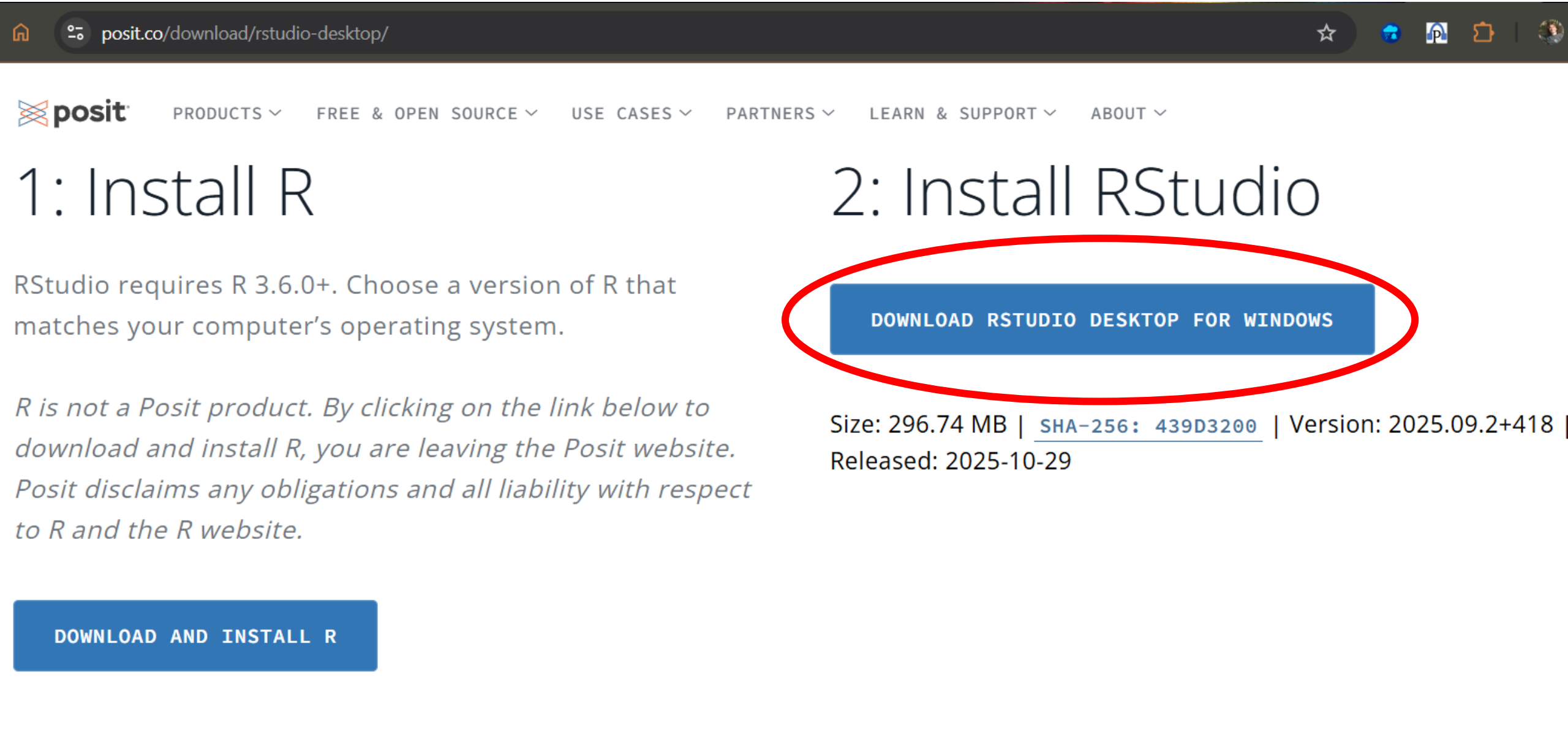
Code | Blame | 1014 lines (722 loc) · 34.2 KB

```
1   ##############################################
2   ## Clase práctica de GWAS                   ##
3   ## Docente invitado: Dr. Martin N. Garcia   ##
4   ## Fecha: 26/05/2025                        ##
5   ## Universidad Nacional de San Martin       ##
6   ##############################################
7
8   #CARGA DE PAQUETES
9   library(simulMGF)
10  library(tidyverse)
11  library(nortest)
12  library(rMVP)
13  library(data.table)
14  library(CJAMP)
15
16
17  #setear la semilla (para que los resultados puedan reproducirse exactamente)
18  set.seed(1234)
19
20
21  #SIMULACION DE DATOS
22  Nind <- 10000                  #numero de individuos
```

#Previamente instalar los paquetes

install.packages("simulMGF")
install.packages("tidyverse")
install.packages("nortest")
install.packages("rMVP")
install.packages("data.table")
install.packages("CJAMP")
install.packages("rrBLUP")
install.packages("tidyr")

https://posit.co/download/rstudio-desktop/



Consola de R

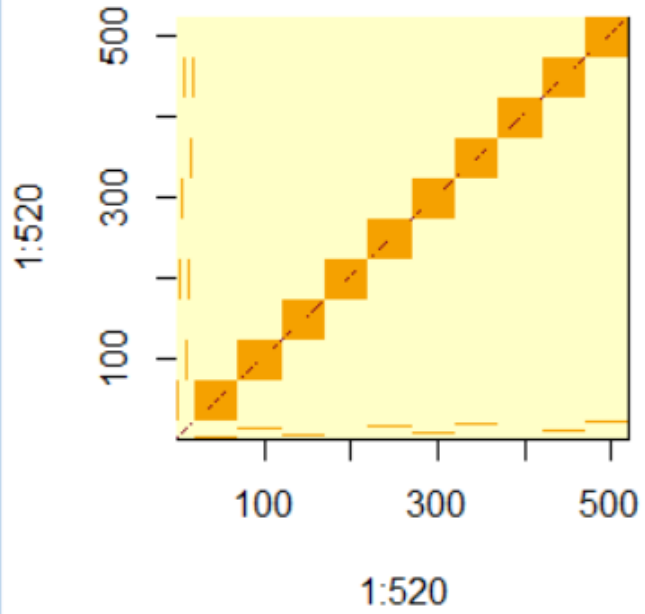
RGui

File History Resize Windows

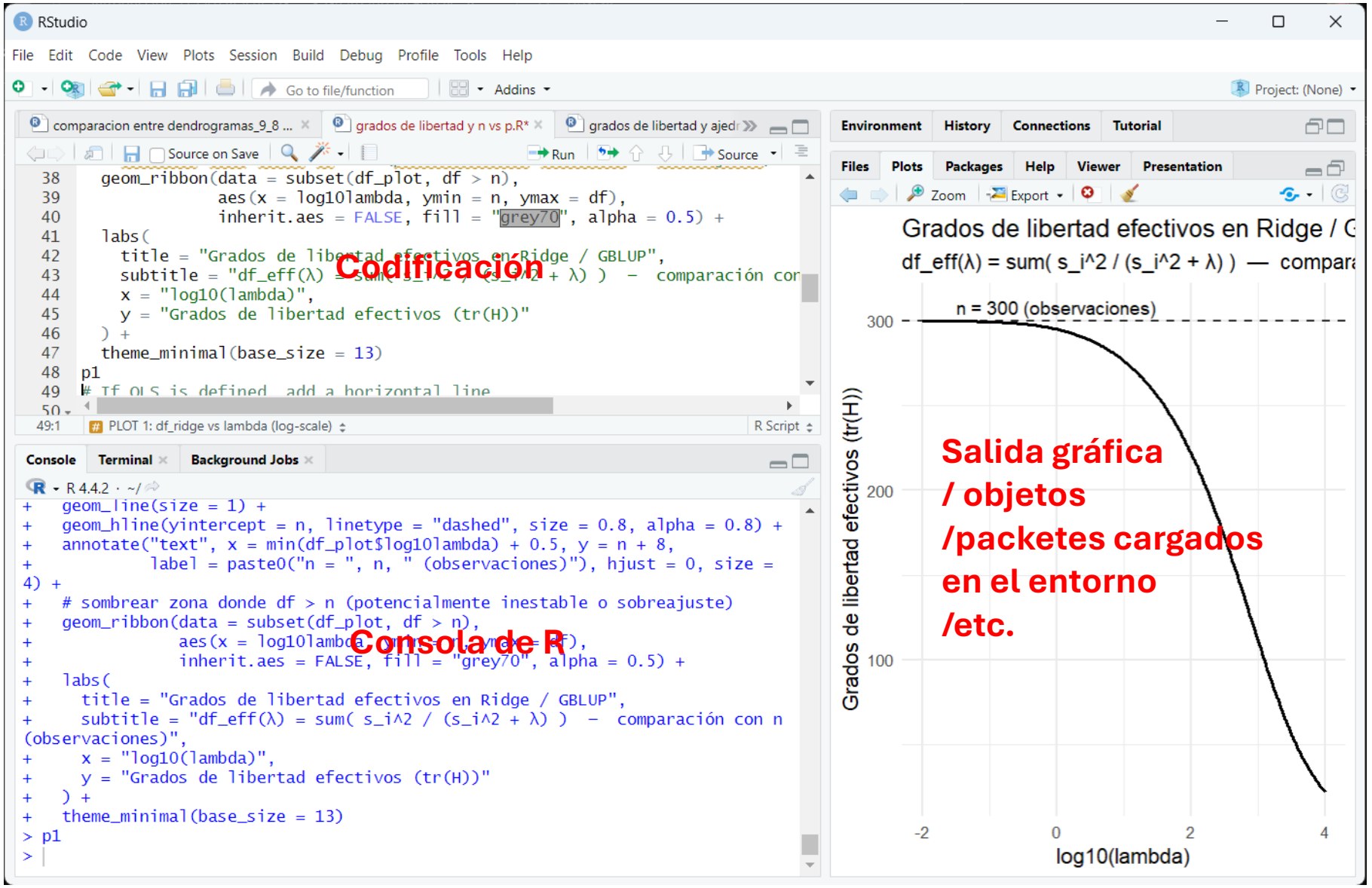
R Console

```
M10  0.0  0.0  0.0  0.0  0.0
P11  0.0  0.0  0.0  0.0  0.0
P12  0.0  0.0  0.0  0.0  0.0
P13  0.0  0.0  0.0  0.0  0.0
P14  0.5  0.5  0.5  0.5  0.5
P15  0.0  0.0  0.0  0.0  0.0
P16  0.0  0.0  0.0  0.0  0.0
P17  0.0  0.0  0.0  0.0  0.0
P18  0.0  0.0  0.0  0.0  0.0
P19  0.0  0.0  0.0  0.0  0.0
P20  0.0  0.0  0.0  0.0  0.0
> image(1:520, 1:520, A)
> A[1:5, 1:5]
   M1 M2 M3 M4 M5
M1  1  0  0  0  0
M2  0  1  0  0  0
M3  0  0  1  0  0
M4  0  0  0  1  0
M5  0  0  0  0  1
>
```

R Graphics: Device 2 (ACTIVE)



RStudio


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
comparacion entre dendrogramas_9_8 ...
grados de libertad y n vs p.R*
grados de libertad y ajed
Source on Save
Run
Source
38 geom_ribbon(data = subset(df_plot, df > n),
39 aes(x = log10lambda, ymin = n, ymax = df),
40 inherit.aes = FALSE, fill = "grey70", alpha = 0.5) +
41 labs(
42 title = "Grados de libertad efectivos en Ridge / GBLUP",
43 subtitle = "df_eff(λ) = sum(s_i^2 / (s_i^2 + λ)) — comparación con
44 x = "log10(lambda)",
45 y = "Grados de libertad efectivos (tr(H))"
46) +
47 theme_minimal(base_size = 13)
48 p1
49 # If OLS is defined, add a horizontal line
50
49:1
PLOT 1: df_ridge vs lambda (log-scale)
R Script
Codificación
Console
Terminal
Background Jobs
R 4.4.2 · ~/
+ geom_line(size = 1) +
+ geom_hline(yintercept = n, linetype = "dashed", size = 0.8, alpha = 0.8) +
+ annotate("text", x = min(df_plot$log10lambda) + 0.5, y = n + 8,
+ label = paste0("n = ", n, " (observaciones)"), hjust = 0, size =
4) +
+ # sombrear zona donde df > n (potencialmente inestable o sobreajuste)
+ geom_ribbon(data = subset(df_plot, df > n),
+ aes(x = log10lambda, ymin = n, ymax = df),
+ inherit.aes = FALSE, fill = "grey70", alpha = 0.5) +
+ labs(
+ title = "Grados de libertad efectivos en Ridge / GBLUP",
+ subtitle = "df_eff(λ) = sum(s_i^2 / (s_i^2 + λ)) — comparación con n
(observaciones)",
+ x = "log10(lambda)",
+ y = "Grados de libertad efectivos (tr(H))"
+) +
+ theme_minimal(base_size = 13)
> p1
>
Consola de R
Environment
History
Connections
Tutorial
Files
Plots
Packages
Help
Viewer
Presentation
Zoom
Export
Grados de libertad efectivos en Ridge / G
df_eff(λ) = sum(s_i^2 / (s_i^2 + λ)) — compara
n = 300 (observaciones)
300
200
100
Grados de libertad efectivos (tr(H))
-2
0
2
4
log10(lambda)
Salida gráfica
/ objetos
/packetes cargados
en el entorno
/etc.

Instalación de paquetes

```
install.packages("simulMGF")
install.packages("tidyverse")
install.packages("nortest")
install.packages("rMVP")
install.packages("data.table")
install.packages("CJAMP")
install.packages("ggplot2")
install.packages("AGHmatrix")
install.packages("rrBLUP")
install.packages("tidyr")
```

Clase práctica de GWAS

Docente: Dr. Martín Nahuel García

Mejoramiento Genético y Genómico Vegetal

Fecha: 11//2025

garcia.martin@inta.gob.ar

PUNTEO DE ACTIVIDADES

Carga de paquetes

Simulación de datos

```
library(simulMGF)
library(tidyverse)
library(nortest)
library(rMVP)
library(data.table)
library(CJAMP)
```

Descripción (CRAN)
Este paquete de R permite simular genotipos en una matriz de SNPs (polimorfismos de nucleótido único) para organismos diploides, usando números aleatorios de una distribución uniforme, codificados como 0, 1 o 2 (según Sikorska et al., 2013). También permite generar matrices de SNP para medios hermanos o hermanos completos a partir de datos de SNP reales o simulados de los padres, asumiendo segregación mendeliana. Además, simula rasgos fenotípicos controlados por loci de rasgo cuantitativo (QTL) con efectos extraídos de distribuciones normal o uniforme, bajo un modelo puramente aditivo. Esto es útil para probar modelos de asociación y predicción genómica o con fines educativos.

Gráficos / diseño

```
library(simulMGF)
library(tidyverse)
library(nortest)
library(rMVP)
library(data.table)
library(CJAMP)
```

Descripción (CRAN)

El *tidyverse* es un conjunto de paquetes que funcionan en armonía porque comparten representaciones de datos y un diseño de API comunes. Este paquete facilita la instalación y carga de múltiples paquetes del *tidyverse* en un solo paso. Más información en: https://www.tidyverse.org.

Pruebas de normalidad

```
library(simulMGF)
library(tidyverse)
library(nortest)
library(rMVP)
library(data.table)
library(CJAMP)
```

Descripción (CRAN)
Cinco pruebas globales (*omnibus*) para evaluar la hipótesis compuesta de normalidad.

Descripción (CRAN)

rMVP es una herramienta para estudios de asociación genómica amplia (GWAS) que destaca por su eficiencia en el uso de memoria, visualización mejorada y aceleración en paralelo. Sus principales características son:

1. Procesamiento eficiente de grandes volúmenes de datos.
2. Evaluación rápida de la estructura poblacional.
3. Estimación eficiente de componentes de varianza mediante varios algoritmos.
4. Pruebas de asociación aceleradas en paralelo mediante tres métodos.
5. Diseño computacional globalmente eficiente para GWAS.
6. Visualización mejorada de la información relacionada.

GWAS

```
library(simulMGF)
library(tidyverse)
library(nortest)
library(rMVP)
library(data.table)
library(CJAMP)
```

Incluye tres modelos: GLM, MLM y FarmCPU, y métodos para estimar componentes de varianza como EMMAX, FaSTLMM, SUPER y regresión HE.

Descripción (CRAN)

Permite la agregación rápida de grandes volúmenes de datos (por ejemplo, 100 GB en RAM), uniones ordenadas rápidas, y la adición, modificación o eliminación eficiente de columnas por grupo sin crear copias. Soporta columnas tipo lista y lectura/escritura rápida de archivos separados por caracteres. Ofrece una sintaxis natural y flexible que facilita y acelera el desarrollo.

Armado de tablas

```
library(simulMGF)
library(tidyverse)
library(nortest)
library(rMVP)
library(data.table)
library(CJAMP)
```

Descripción (CRAN)

Este paquete ofrece una implementación eficiente y robusta del método C-JAMP (Análisis Conjunto de Múltiples Fenotipos basado en Cópulas), recientemente propuesto. C-JAMP permite estimar y probar la asociación de uno o varios predictores sobre múltiples resultados en un modelo conjunto, con especial enfoque en estudios genómicos a gran escala con dos fenotipos. El uso de funciones de cópulas permite modelar una amplia gama de dependencias multivariadas entre fenotipos, y se ha demostrado que C-JAMP puede aumentar el poder de detección de variantes genéticas asociadas en comparación con métodos existentes. Además, se incluyen funciones para generar datos genéticos y fenotípicos, calcular la frecuencia del alelo menor (MAF) y estimar la varianza fenotípica explicada por los marcadores genéticos.

Cuanto explican los marcadores asociados de la variación del fenotipo?

simulMGF

simGeno: simula genotipos en matrices de SNP como valores aleatorios desde una distribución uniforme, para organismos diploides (codificados por 0, 1 y 2).

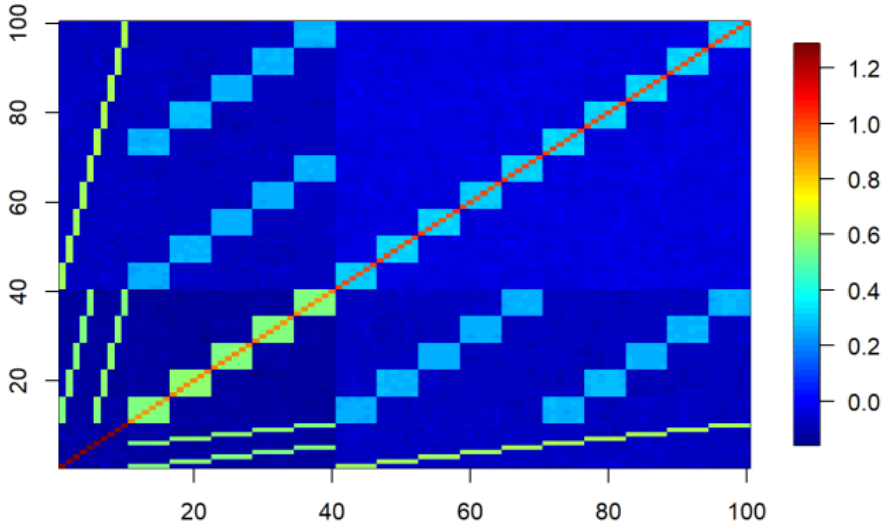
simPheno: simula un fenotipo desde una matriz de SNP con loci de caracteres cuantitativos (QTLs) con efectos muestreados desde una distribución Normal.

simulFS: simula los genotipos de progenies de hermanos completos desde el genotipo de los parentales.

simulHS: simula los genotipos de progenies de medios hermanos desde el genotipo de un parental.

simulN: simula una matriz de SNP y caracteres controlados por determinado número de QTLs y sus efectos muestreados desde una distribución Normal.

simulU: ídem simulN pero muestrea efectos desde una distribución Uniforme.



García, M.N. (2023). simulMGF: Simulate SNP Matrix, Phenotype and Genotypic Effects. R package version 0.1.1.
https://CRAN.R-project.org/package=simulMGF

Simulación de datos

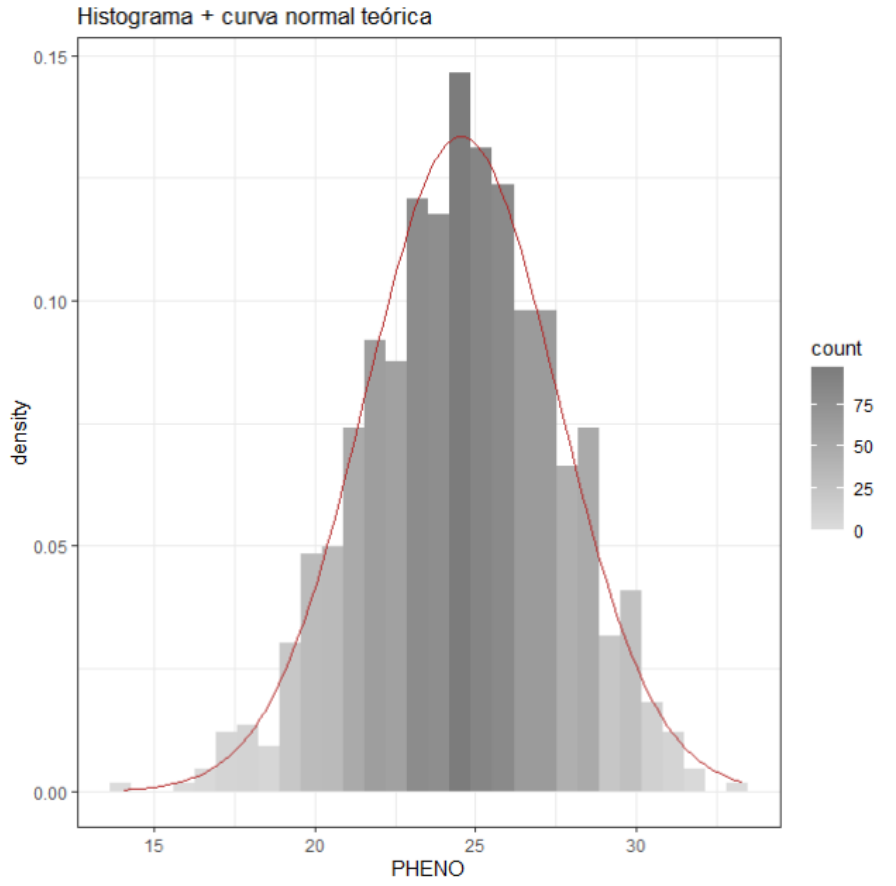
```
install.packages("simulMGF")

library(simulMGF)

set.seed(1234)

simulN(Nind = 1000, Nmarkers = 10000, Nqtl = 50,
Esigma = .5, Pmean = 25, Perror = .25)
```

Código disponible en: https://github.com/mngar/MGyGV

OBJETO “nsimout”

```
str(nsimout)
List of 4
 $ geno      : matriz de genotipos
 $ pheno     : vector de fenotipos
 $ QTN       : vector de marcadores asociados
 $ Meffects  : vector de efectos
```

$$y = Pmean + \Sigma\ QTN\ x\ Meffects + \varepsilon$$



Código disponible en: https://github.com/mngar/MGyGV

SIMULACION DE DATOS

```
Nind <- 10000
Nmarkers <- 10000
Nqtl <- 50
Esigma <- .5
Pmean <- 25
Perror <- .25

simulN(Nind, Nmarkers, Nqtl, Esigma, Pmean, Perror)
```

Simulamos una población de 10000 individuos que cuenta con 10000 marcadores polimórficos, de los cuales 50 están verdaderamente asociados a un carácter que se distribuye de manera Normal con media “25”.

El objeto generado **nsimout** contiene:
- una matriz de genotipos
- un vector de fenotipos
- un vector indicando las columnas con los marcadores asociados
- un vector con los efectos de esos marcadores asociados, que siguen una distribución $N\sim(0,(0.5)^2)$.

```
 str(nsimout)
List of 4
 $ geno    : num [1:10000, 1:10000] 0 1 1 1 2 1 0 0 1 1 ...
 $ pheno   : num [1:10000, 1] 29.5 24.6 26.3 28.6 28.8 ...
 $ QTN     : int [1:50] 2939 8550 469 5061 6087 6850 9650 1922 6252 830 ...
 $ Meffects: num [1:50] 0.4769 0.55592 -0.46926 -0.00734 -0.42578 ...
```

Llamamos pop a la matriz de datos genotípicos, la cual es una matriz de 10000 individuos (filas) x 10000 SNP (columnas).

```r
pop <- as.data.frame(nsimout$geno)
colnames(pop) <- paste0("M",c(1:10000))
rownames(pop) <- paste0("I",c(1:10000))

pop[1:10, 1:20]
```

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19	M20
I1	0	0	0	2	0	0	2	2	0	0	2	1	0	1	1	1	2	2	0	0
I2	1	1	2	2	0	2	0	0	2	0	0	2	0	1	0	0	0	1	2	0
I3	1	1	2	0	0	1	1	2	1	1	1	1	1	2	1	1	1	2	2	2
I4	1	2	1	0	0	1	0	0	0	0	2	1	1	1	1	2	1	1	0	2
I5	2	0	2	2	0	2	2	0	0	2	2	1	2	2	0	1	0	1	1	1
I6	1	0	1	2	2	0	0	0	2	1	0	2	1	1	1	1	2	2	1	1
I7	0	1	1	2	1	1	2	0	1	1	1	2	1	2	0	0	2	0	0	1
I8	0	0	1	2	0	0	1	0	1	1	1	1	1	1	2	1	0	0	0	2
I9	1	0	2	2	0	1	2	2	0	2	0	1	0	2	2	0	2	2	0	1
I10	1	1	0	2	0	1	1	1	1	0	0	0	0	0	0	2	1	2	0	0

Poblaciones de estudio, a partir de las cuales determinaremos cuales son los marcadores asociados al carácter que nos interesa:

- 200 individuos.
- 1000 individuos.
- 5000 individuos.

Población de validación, una población externa sobre la cual evaluaremos los modelos predictivos generados a partir de la información obtenida mediante GWAS:

- 5000 individuos

```r
#3 poblaciones de estudio
geno1 <- pop[1:200,]
geno2 <- pop[1:1000,]
geno3 <- pop[1:5000,]

feno1 <- nsimout$pheno[1:200]
feno2 <- nsimout$pheno[1:1000]
feno3 <- nsimout$pheno[1:5000]

feno1 <- data.frame(IND = rownames(geno1), Pheno = feno1)
feno2 <- data.frame(IND = rownames(geno2), Pheno = feno2)
feno3 <- data.frame(IND = rownames(geno3), Pheno = feno3)


#1 poblacion de validacion
genoV <- pop[5001:10000,]
fenoV <- nsimout$pheno[5001:10000]
fenoV <- data.frame(IND = rownames(genoV), Pheno = fenoV)
```

```r
QTL <- cbind(nsimout$QTN, nsimout$Meffects)
QTL <- cbind(QTL,abs(nsimout$Meffects))
colnames(QTL) <- c("marker", "effect", "effabs")
QTL <- as.data.frame(QTL)
QTL <- QTL[order(-QTL$effabs),]
QTL$SNP = paste0("M",QTL$marker)

head(QTL)
   marker     effect    effabs   SNP
31   7894  1.1029343 1.1029343 M7894
38   4253 -0.9660033 0.9660033 M4253
7    9650 -0.8596818 0.8596818 M9650
9    6252  0.7757577 0.7757577 M6252
26   1272  0.6476765 0.6476765 M1272
34   8271 -0.6093135 0.6093135 M8271
tail(QTL)
   marker        effect       effabs   SNP
45   8222  0.0224232204 0.0224232204 M8222
43   6837 -0.0194698251 0.0194698251 M6837
29   4816 -0.0154100050 0.0154100050 M4816
15   6917  0.0109210170 0.0109210170 M6917
4    5061 -0.0073401578 0.0073401578 M5061
23   6289 -0.0005240767 0.0005240767 M6289
```

Armaremos una tabla con los marcadores realmente asociados al fenotipo (al tratarse de datos simulados este dato es conocido), que nos servirá para evaluar la capacidad de las metodologías empleadas en detectarlos.

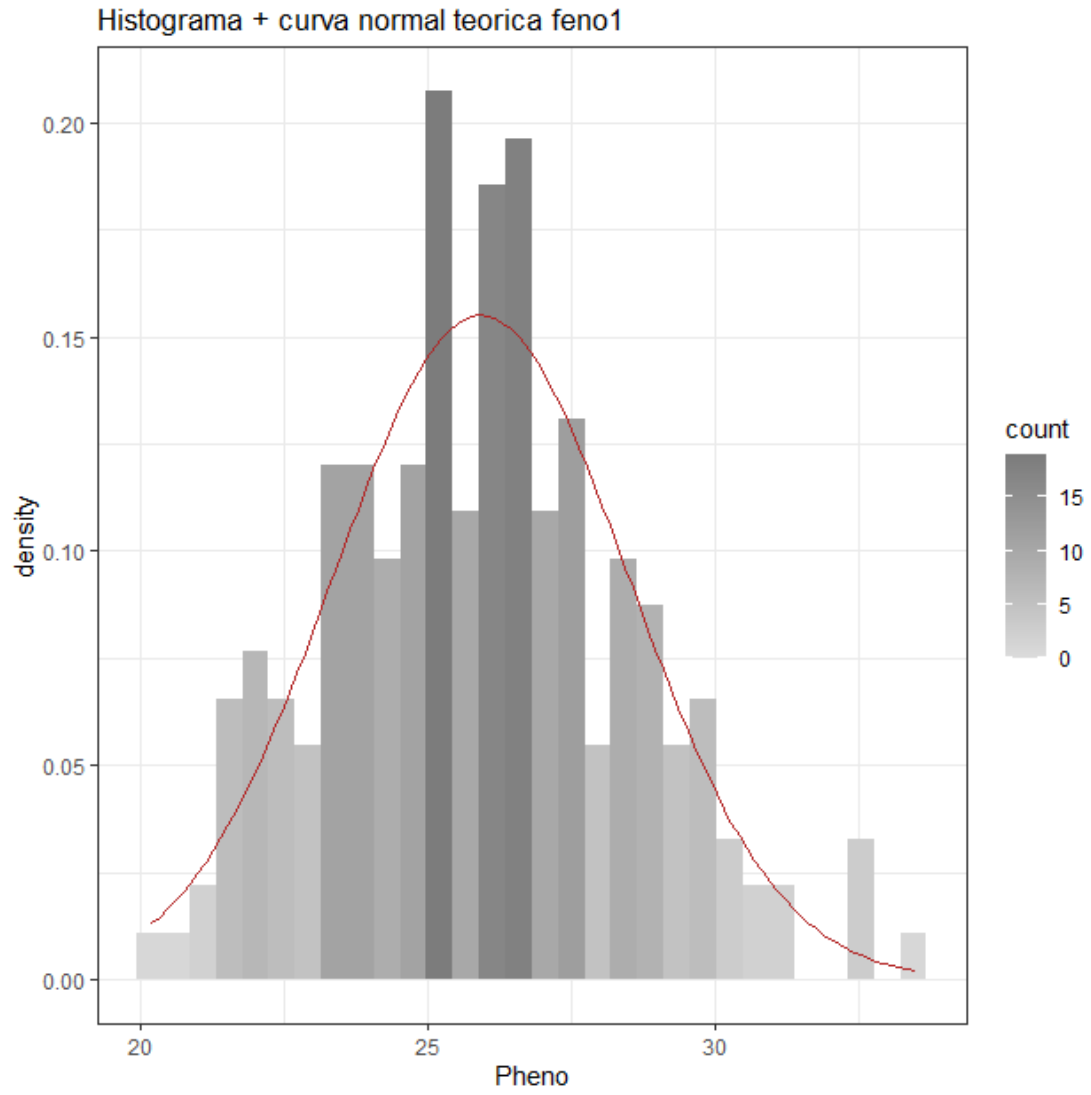
Generamos el objeto **map**, para indicar la posición de cada marcador.

En este caso distribuimos los SNPs en 5 cromosomas con el mismo número de marcadores.

```
#mapa
map <- data.frame (SNP = colnames(pop),
                   Chromosome = c(rep(1,(Nmarkers/5)),
                                  rep(2,(Nmarkers/5)),
                                  rep(3,(Nmarkers/5)),
                                  rep(4,(Nmarkers/5)),
                                  rep(5,(Nmarkers/5))),
                   Position = c(1:(Nmarkers/5),
                                1:(Nmarkers/5),
                                1:(Nmarkers/5),
                                1:(Nmarkers/5),
                                1:(Nmarkers/5)))

head(map)
  SNP Chromosome Position
1  M1          1        1
2  M2          1        2
3  M3          1        3

tail(map)
         SNP Chromosome Position
9998   M9998          5     1998
9999   M9999          5     1999
10000 M10000          5     2000
```

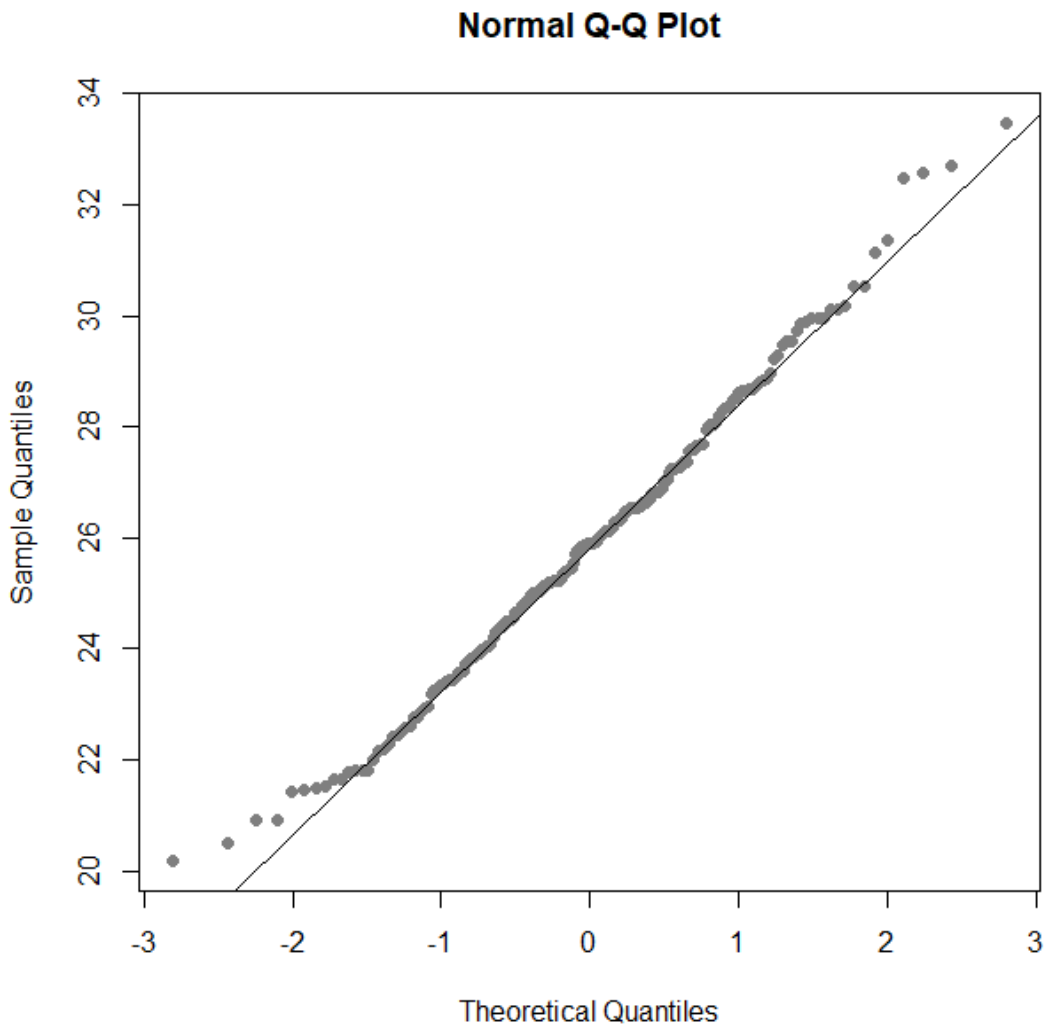
Checkeo visual de la normalidad del carácter mediante un histograma.

```r
ggplot(data = feno1, aes(x = Pheno)) +
  geom_histogram(aes(y = ..density.., fill = ..count..)) +
  scale_fill_gradient(low = "#DCDCDC", high = "#7C7C7C") +
  stat_function(fun = dnorm, colour = "firebrick",
                args = list(mean = mean(feno1$Pheno),
                            sd = sd(feno1$Pheno))) +
  ggtitle("Histograma + curva normal teorica feno1") +
  theme_bw()
```



Checkeo visual de la normalidad del carácter mediante un gráfico cuantil-cuantil.

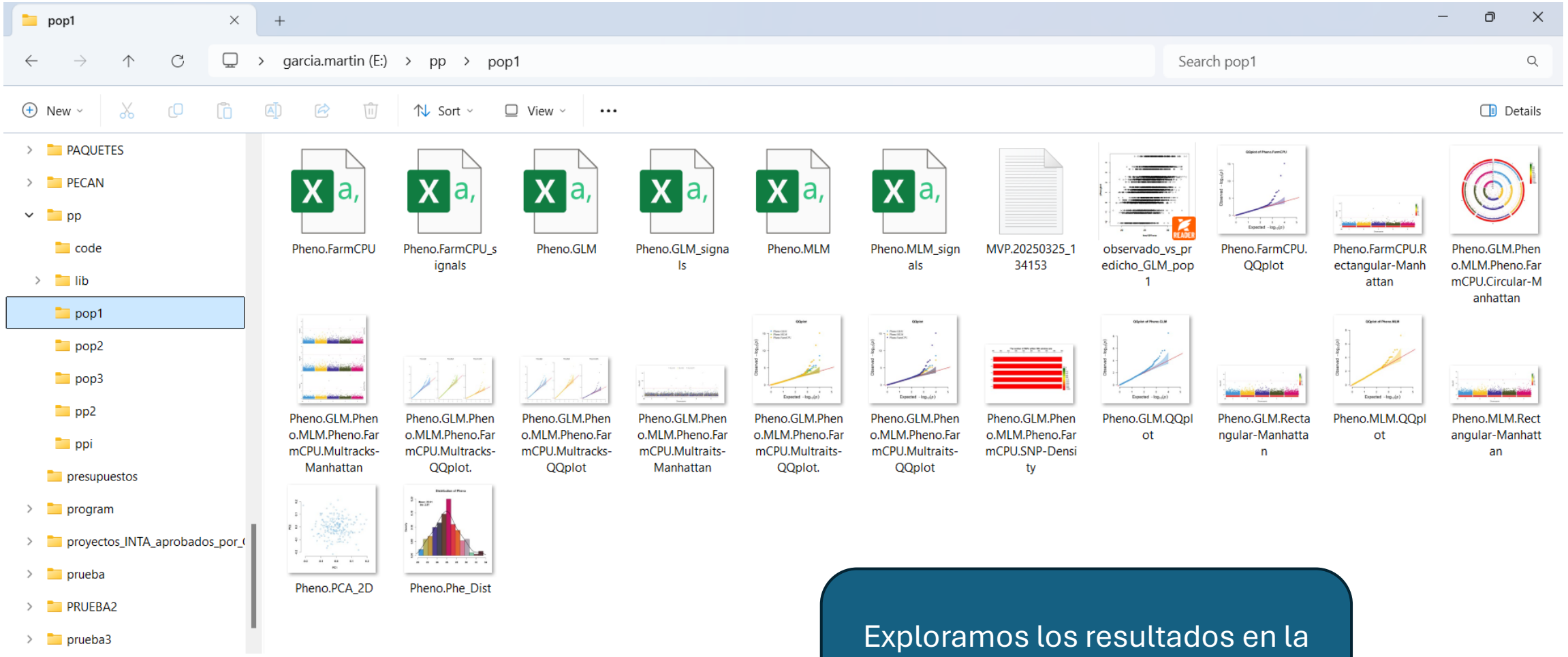
```r
qqnorm(feno1$Pheno, pch = 19, col = "gray50")
qqline(feno1$Pheno)
```



```r
dir.create(file.path("E:/pp/", "pop1"), showWarnings = FALSE)
setwd("E:/pp/pop1")

imMVP1 <- MVP(
  phe=feno1,
  geno=as.big.matrix(geno1),
  map=map,
  nPC.GLM=5,
  nPC.MLM=3,
  nPC.FarmCPU=3,
  maxLine=10000,
  vc.method="BRENT",
  method.bin="static",
  threshold=0.05,
  method=c("GLM", "MLM", "FarmCPU"),
  file.output=c("pmap", "pmap.signal", "plot", "log")
)
```

Creamos/elegimos el directorio donde saldrán los archivos de resultados.

imMVP1 será el objeto con los resultados del GWAS realizado con tres metodologías (GLM, MLM y FarmCPU) sobre la población 1 (200 individuos).



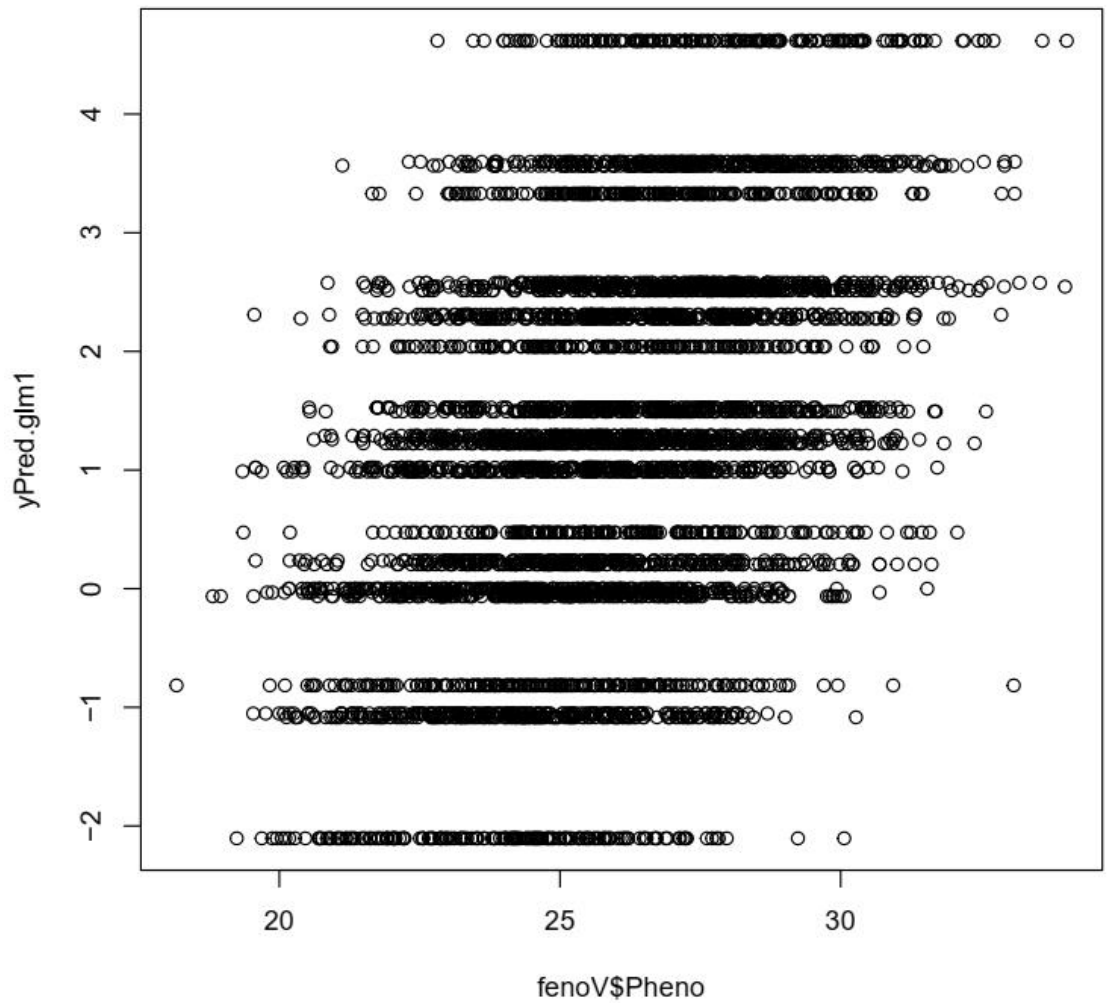
Exploramos los resultados en la carpeta que elegimos como destino de los archivos de salida.

```
setwd("E:/pp/pop1")
glm.signal1 <- read.csv("Pheno.GLM_signals.CSV")
glm.signal1
    SNP Chromosome Position    MAF    Effect        SE    Pheno.GLM
1 M6087          4       87 0.5000 -1.052039 0.2152592 2.143759e-06
2 M7894          4     1894 0.4550  1.288934 0.2073545 3.081871e-09
3 M9846          5     1846 0.4975  1.020128 0.2103834 2.548702e-06

glm.res1 = as.data.frame(imMVP1$glm.results)
caus.glm1 = c(rep(FALSE, 10000))
caus.glm1[c(6087, 7894, 9846)] = TRUE

pev.glm1 = compute_expl_var(genodata = geno1, phenodata = feno1$Pheno,
                            type = c("Rsquared_unadj", "Rsquared_adj"),
                            causal_idx = caus.glm1, effect_causal =
glm.res1$Effect)

pev.glm1
$Rsquared_unadj
[1] 0.3313
$Rsquared_adj
[1] 0.3210648
```

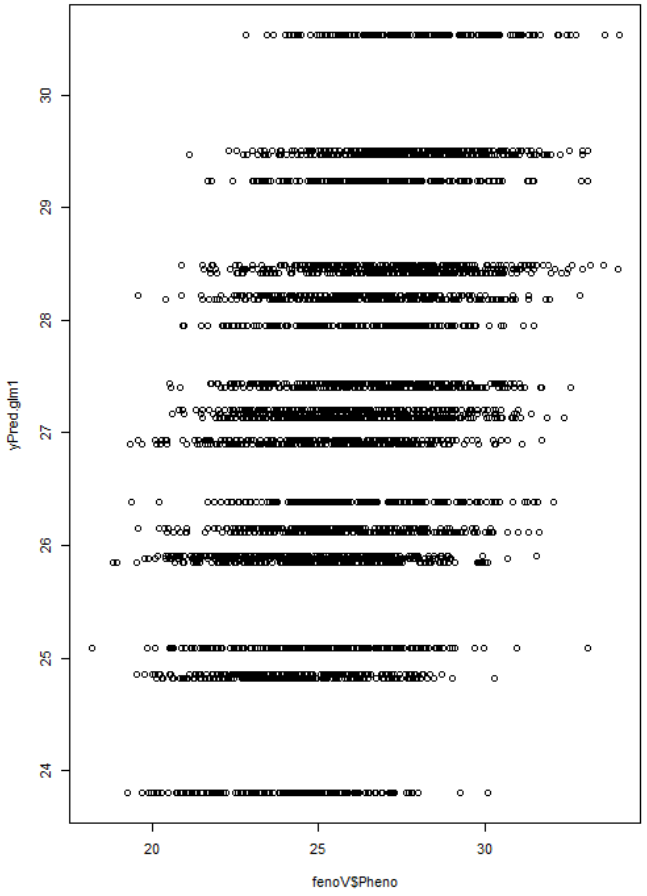
```r
e.glm1 = glm.signal1$Effect
e.glm1 = as.vector(e.glm1)
e.glm1

glm.geno.causal1 = as.matrix(genoV[,glm.signal1$SNP])
head(glm.geno.causal1)
      M6087 M7894 M9846
I5001     0     1     1
I5002     2     1     1
I5003     2     1     2
I5004     1     1     2
I5005     1     2     0
I5006     1     0     0

yPred.glm1 = glm.geno.causal1%*%e.glm1+mean(feno1$Pheno)
plot(fenoV$Pheno, yPred.glm1)
```

yPred.glm1 es nuestro modelo predictivo, definido como:
Fenotipo Predicho del individuo k = [promedio(feno1)] $+ \sum_{i=1}^{n} Z_{ki}\gamma_{ki}$

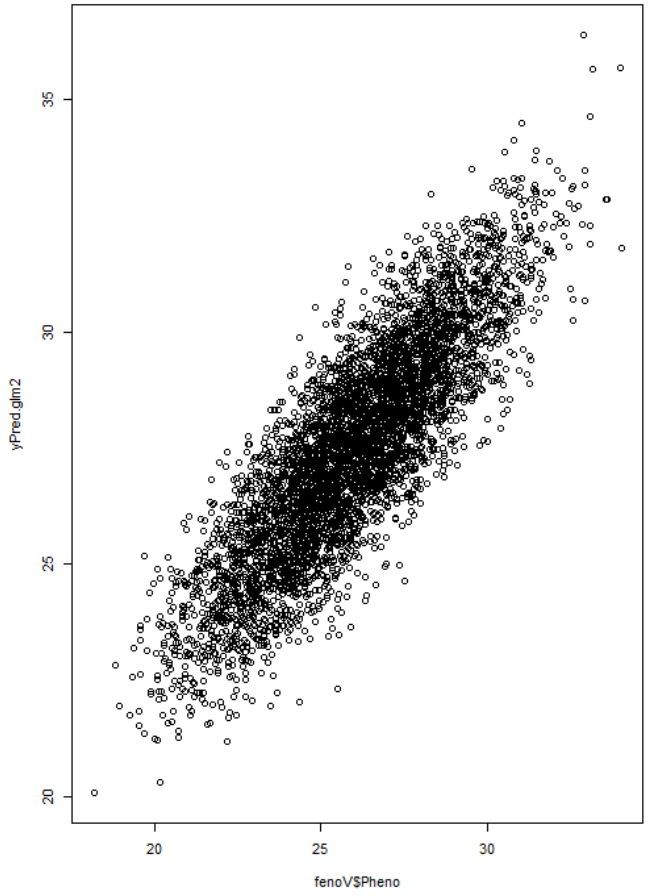
Donde Z_{ki} es el genotipo del i-ésimo SNP asociado y γ_i es su efecto;
y n es el número de SNPs asociados.


yPred.glm1
4
3
2
1
0
−1
−2
20
25
30
fenoV$Pheno

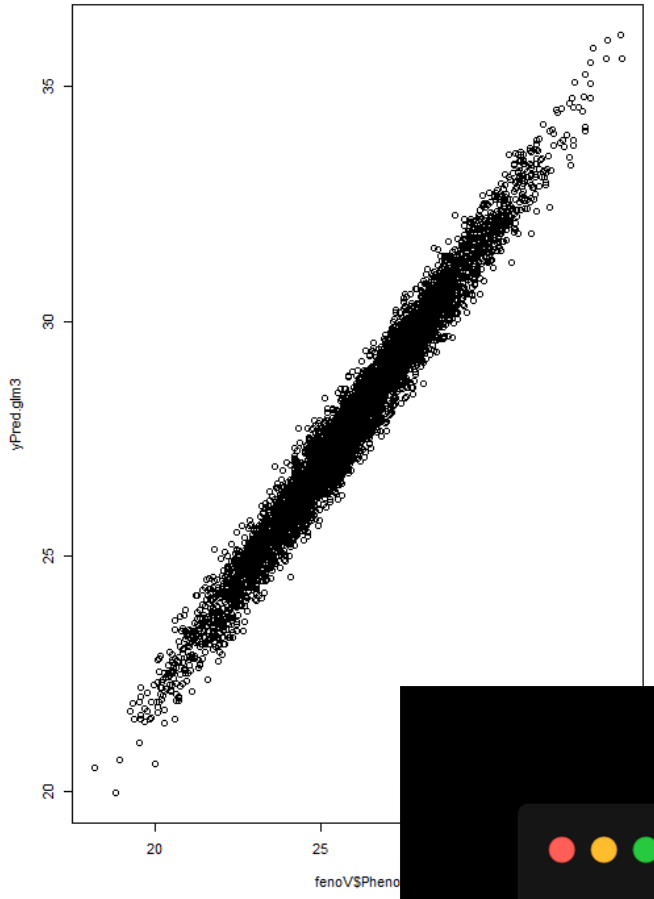
GLM_200ind_10000 SNPs



GLM_1000ind_10000 SNPs



GLM_5000ind_10000 SNPs

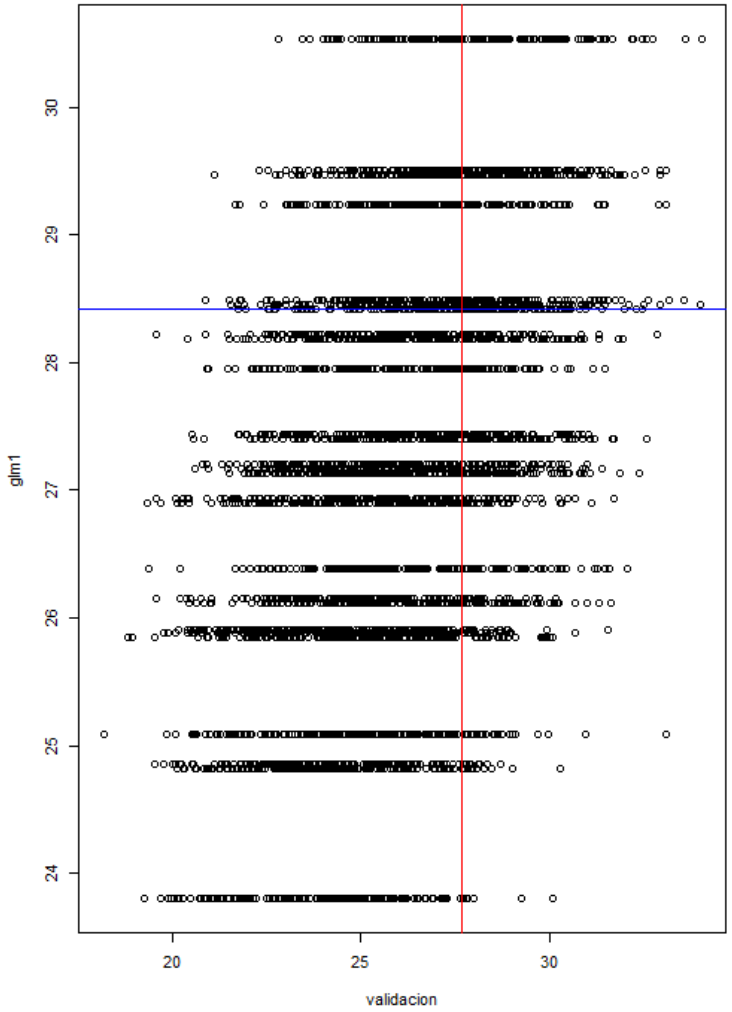


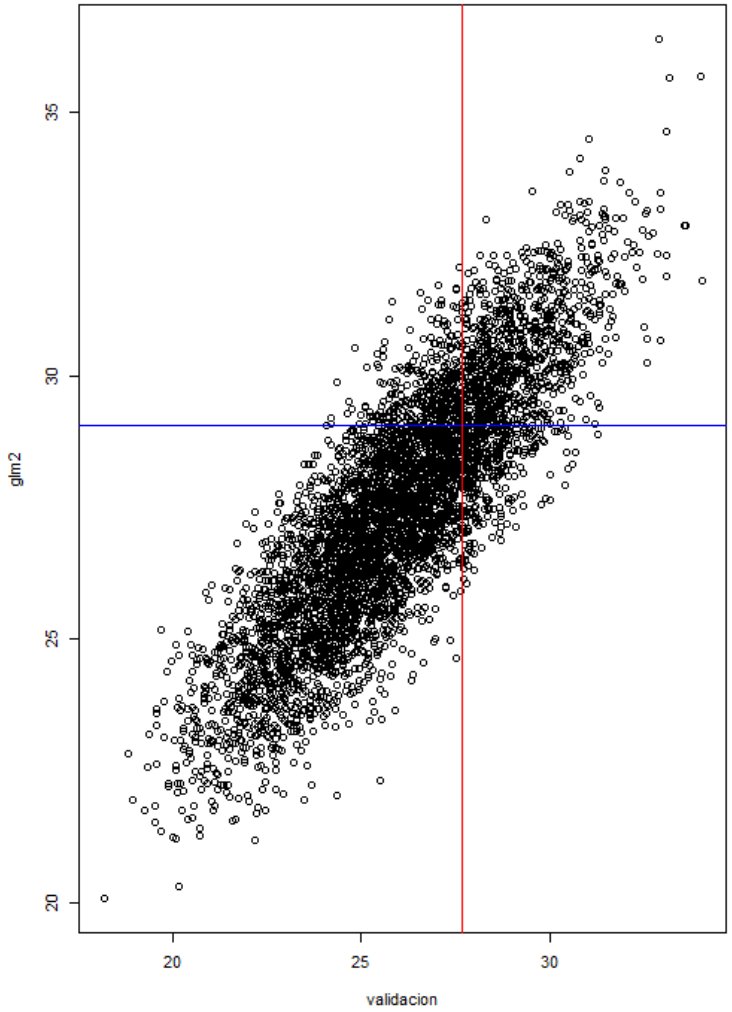
Podemos evaluar la precisión del modelo calculando la correlación entre “Fenotipos observados” y “Fenotipos predichos”

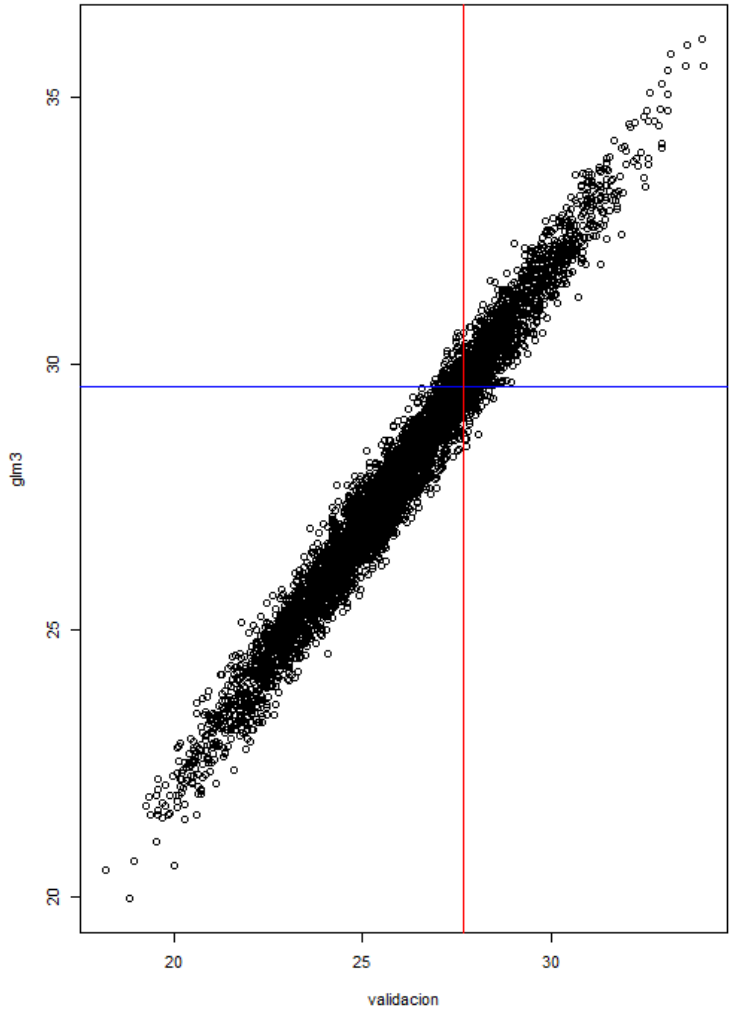
```
cor(fenoV$Pheno, yPred.glm1)
0.423
cor(fenoV$Pheno, yPred.glm2)
0.854
cor(fenoV$Pheno, yPred.glm3)
0.985
```

```r
resultados.glm = cbind(fenoV$Pheno, yPred.glm1)
resultados.glm = cbind(resultados.glm, yPred.glm2)
resultados.glm = cbind(resultados.glm, yPred.glm3)

colnames(resultados.glm) = c("validacion", "glm1", "glm2", "glm3")

head(resultados.glm)
      validacion     glm1     glm2     glm3
I5001   29.20532 28.21765 31.16691 31.60835
I5002   27.47146 26.11357 28.44232 29.28770
I5003   28.63221 27.13370 29.86830 31.14843
I5004   31.83360 28.18574 31.71103 33.52959
I5005   26.11438 27.43442 27.94379 28.03415
I5006   23.46066 24.85655 26.20686 25.82856
```

Si estos análisis fueran parte de un programa de mejoramiento y seleccionaramos el 25% superior del ranking de individuos obtenido con nuestros modelos (los que sobrepasan la línea horizontal azul) para que sean los parentales de la siguiente generación; podemos observar que el contenido de individuos que realmente pertenecen a ese “25% superior” (línea vertical roja) varia según el modelo predictivo.



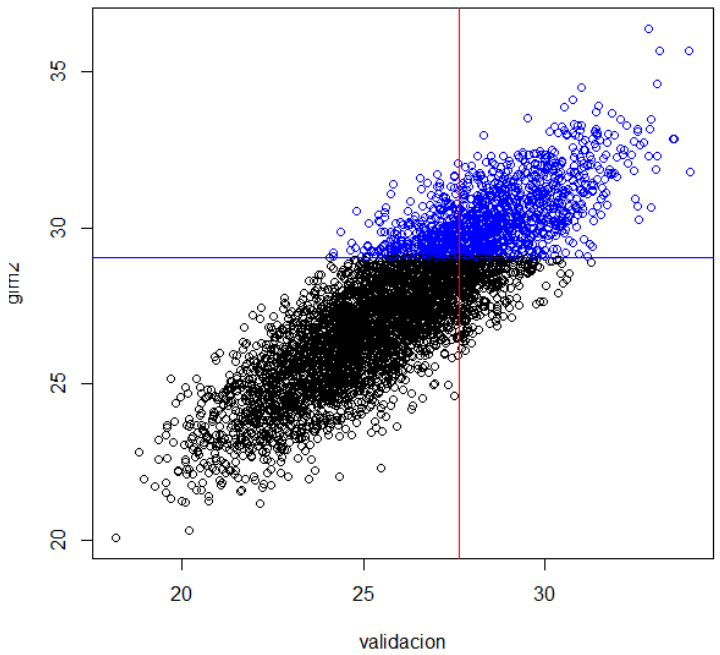


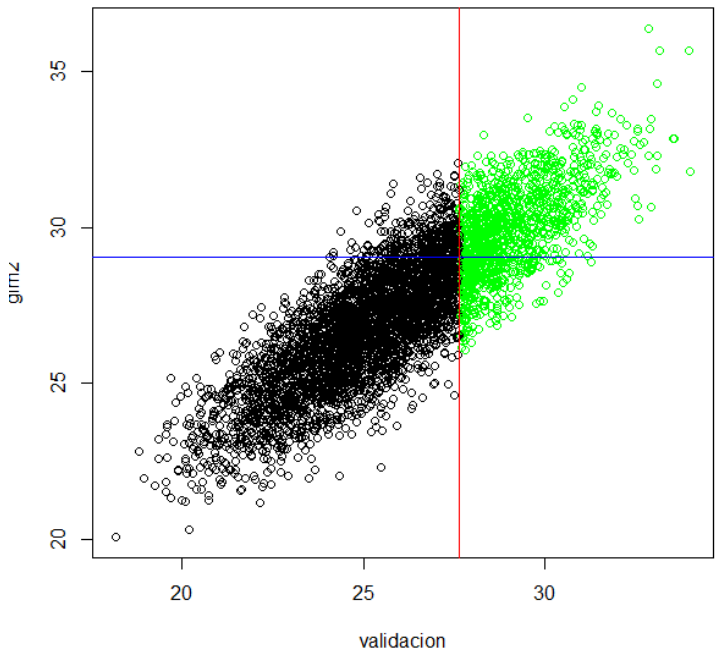


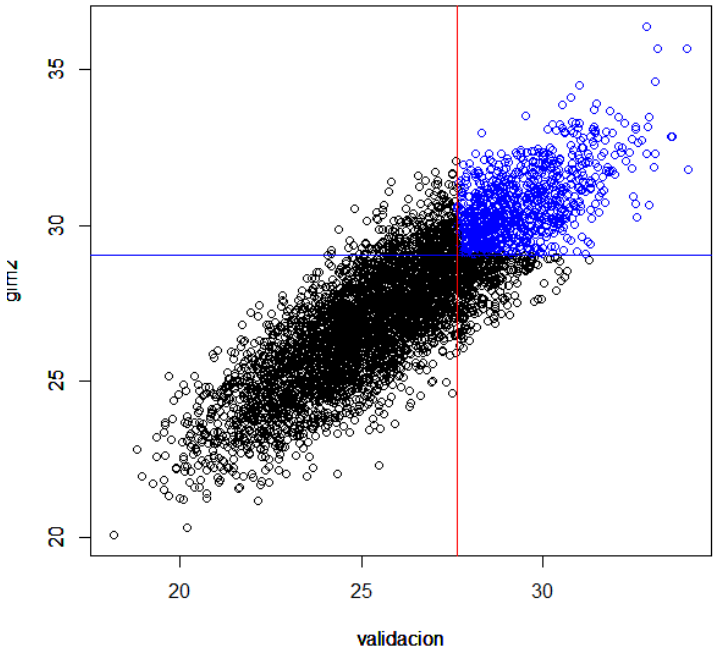
TOP25
PREDICHO
GLM2

TOP25
OBSERVADO

TOP25
PREDICHO
GLM2
∩
TOP25
OBSERVADO







Eficiencia de selección

La definimos (nosotros) como la proporción de los individuos seleccionados correctamente como superiores por el modelo evaluado.

```
ind.sup = rownames(resultados.glm[which(resultados.glm$validacion>quantile(resultados.glm$validacion,0.75)),])
sel.glm1 = rownames(resultados.glm[which(resultados.glm$glm1>quantile(resultados.glm$glm1,0.75)),])
sel.glm2 = rownames(resultados.glm[which(resultados.glm$glm2>quantile(resultados.glm$glm2,0.75)),])
sel.glm3 = rownames(resultados.glm[which(resultados.glm$glm3>quantile(resultados.glm$glm3,0.75)),])

ef.glm1 = length(intersect(ind.sup,sel.glm1))/length(sel.glm1)
ef.glm2 = length(intersect(ind.sup,sel.glm2))/length(sel.glm2)
ef.glm3 = length(intersect(ind.sup,sel.glm3))/length(sel.glm3)

ef.glm1
[1] 0.4390
ef.glm2
[1] 0.7144
ef.glm3
[1] 0.9096
```

Poder estadístico

Lo definimos como la proporción de QTLs verdaderos detectados por el modelo evaluado.

```
poder.glm1 = length(unique(glm.signal1$SNP,QTL$SNP))/Nqtl
poder.glm2 = length(unique(glm.signal2$SNP,QTL$SNP))/Nqtl
poder.glm3 = length(unique(glm.signal3$SNP,QTL$SNP))/Nqtl

poder.glm1
[1] 0.06
poder.glm2
[1] 0.36
poder.glm3
[1] 0.68
```

Falsos positivos

Marcadores no asociados a QTLs que controlan el caracter que son señalados como asociados por el modelo evaluado.

```
#falsos positivos
falpos.glm1 = abs(length(unique(glm.signal1$SNP,QTL$SNP))-length(glm.signal1$SNP))
falpos.glm2 = abs(length(unique(glm.signal2$SNP,QTL$SNP))-length(glm.signal2$SNP))
falpos.glm3 = abs(length(unique(glm.signal3$SNP,QTL$SNP))-length(glm.signal3$SNP))

falpos.glm1
[1] 0
falpos.glm2
[1] 0
falpos.glm3
[1] 0
```

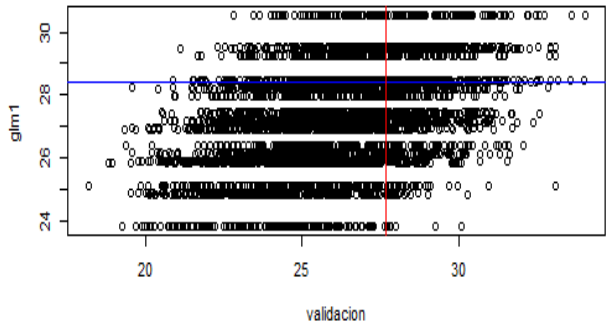
```r
### Cual es el minimo tamaño de efecto detectado por GLM en cada caso
#200 ind
mergexsnp_glm1_QTL = merge(glm.signal1,QTL, by = "SNP")
head(mergexsnp_glm1_QTL)
min(mergexsnp_glm1_QTL$effabs)
[1] 0.4257824

#1000 ind
mergexsnp_glm2_QTL = merge(glm.signal2,QTL, by = "SNP")
head(mergexsnp_glm2_QTL)
min(mergexsnp_glm2_QTL$effabs)
[1] 0.3326874

#5000 ind
mergexsnp_glm3_QTL = merge(glm.signal3,QTL, by = "SNP")
head(mergexsnp_glm3_QTL)
min(mergexsnp_glm3_QTL$effabs)
[1] 0.1557778
```

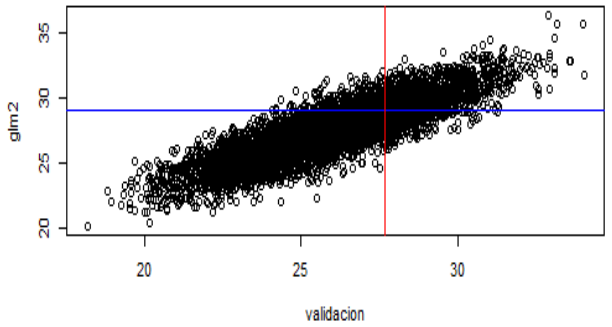
El incremento de poder estadístico se ve reflejado en la capacidad de detectar marcadores con efectos menores.

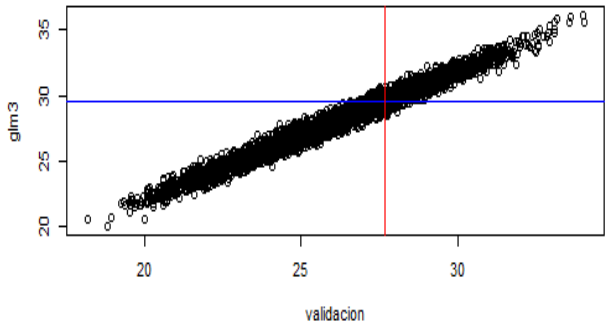
GLM

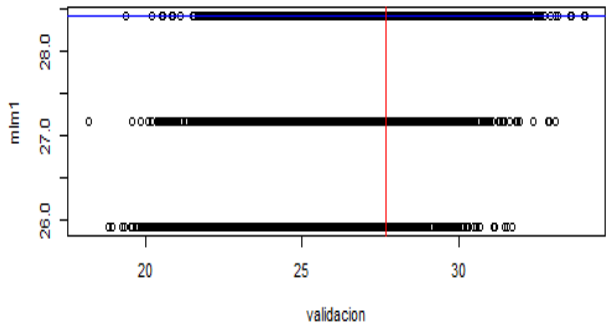
MLM

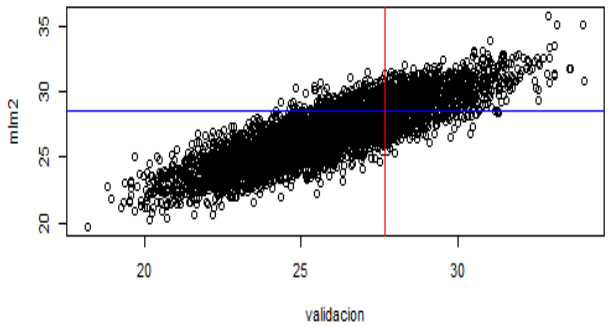
FarmCPU

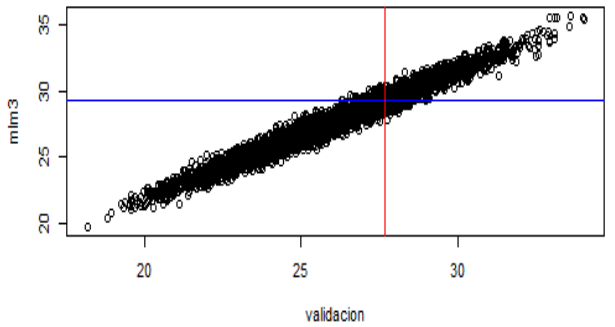


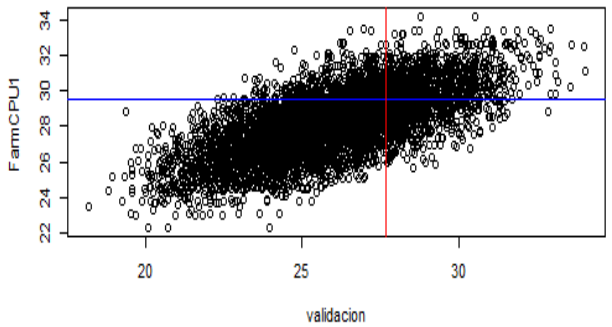


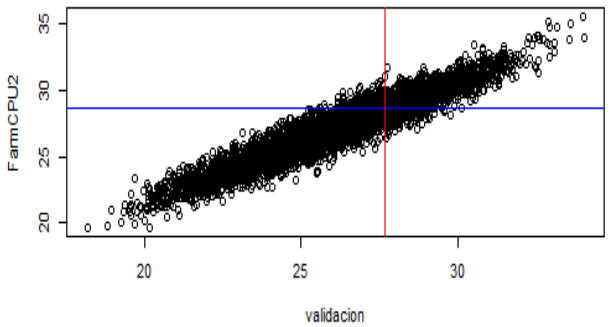


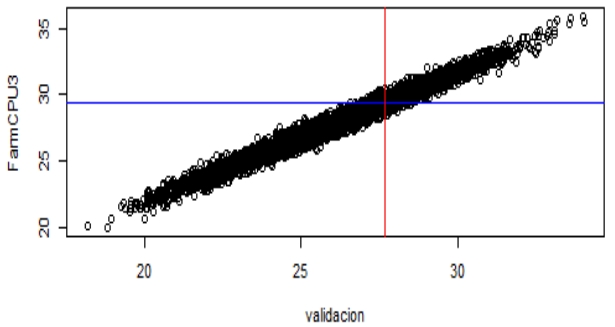




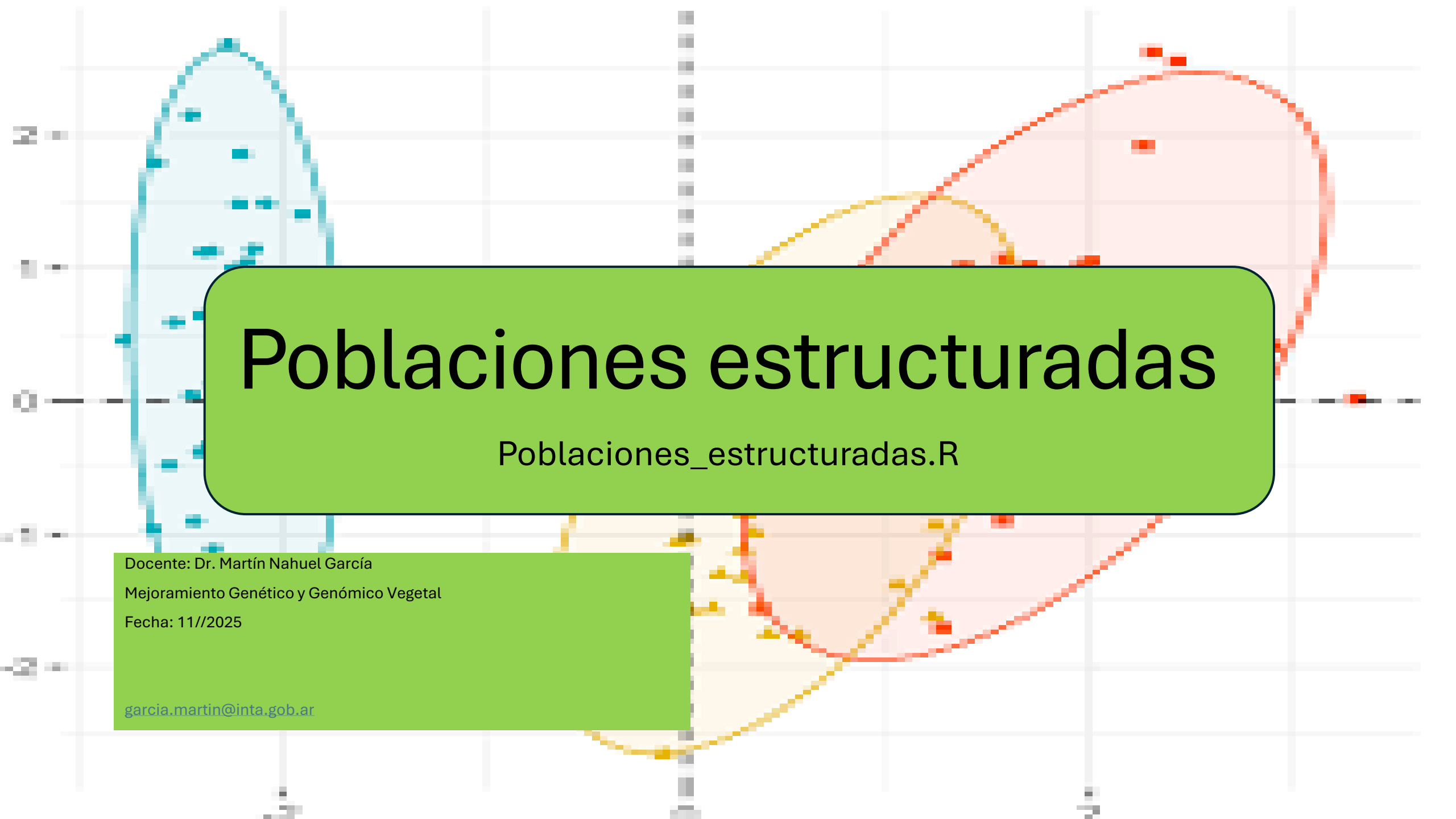








200 individuos	1000 individuos	5000 individuos



Poblaciones estructuradas

Poblaciones_estructuradas.R

Docente: Dr. Martín Nahuel García

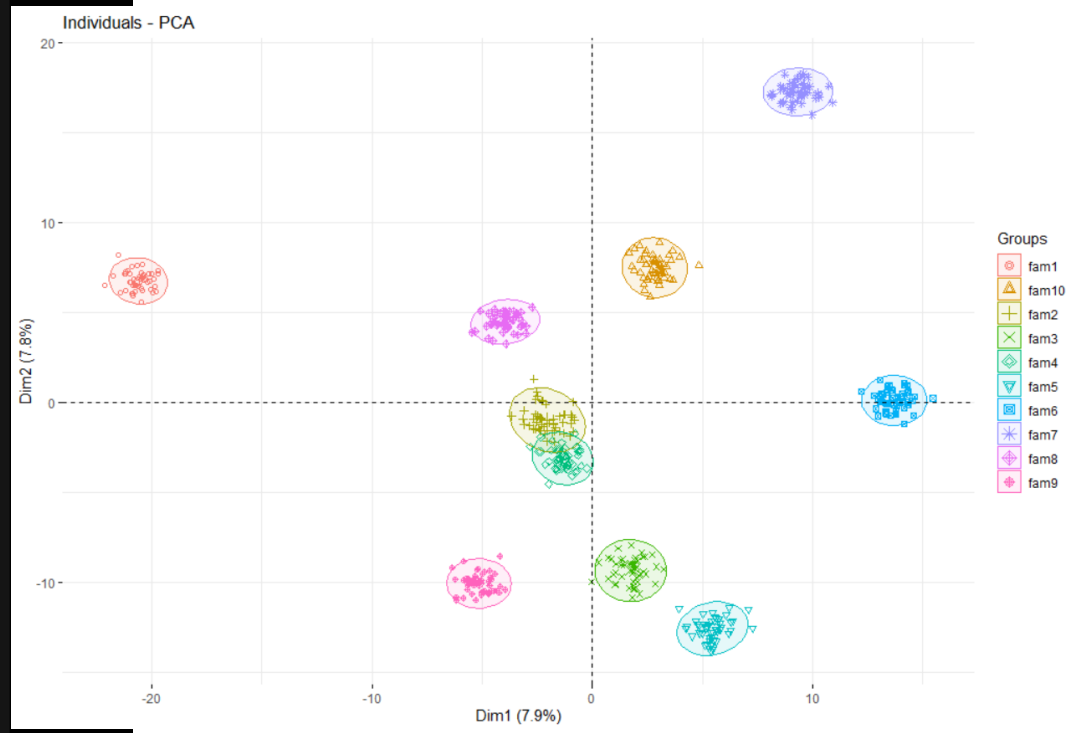
Mejoramiento Genético y Genómico Vegetal

Fecha: 11//2025

garcia.martin@inta.gob.ar

```r
library(simulMGF)
#library(factoMineR)
library(factoextra)

geno.madres = nsimout$geno[1:10,]
geno.padres = nsimout$geno[11:20,]

#generar matriz de genotipos de 10 familias de 50 hermanos completos
simulFS(geno.madres, geno.padres, 50)
#[1] "simulatedFS was generated"

scaled_data <- scale(simulatedFS, center = TRUE, scale = TRUE)
pca_result <- prcomp(scaled_data, graph = FALSE)

#rotulos de familias
familias = c(rep("fam1", 50),rep("fam2", 50),rep("fam3",
50),rep("fam4", 50),rep("fam5", 50), rep("fam6", 50),rep("fam7",
50),rep("fam8", 50),rep("fam9", 50),rep("fam10", 50))

str(familias)
# chr [1:500] "fam1" "fam1" "fam1" "fam1" ...

p <- fviz_pca_ind(pca_result, label="none", habillage=familias,
                  addEllipses=TRUE, ellipse.level=0.95)
print(p)
```

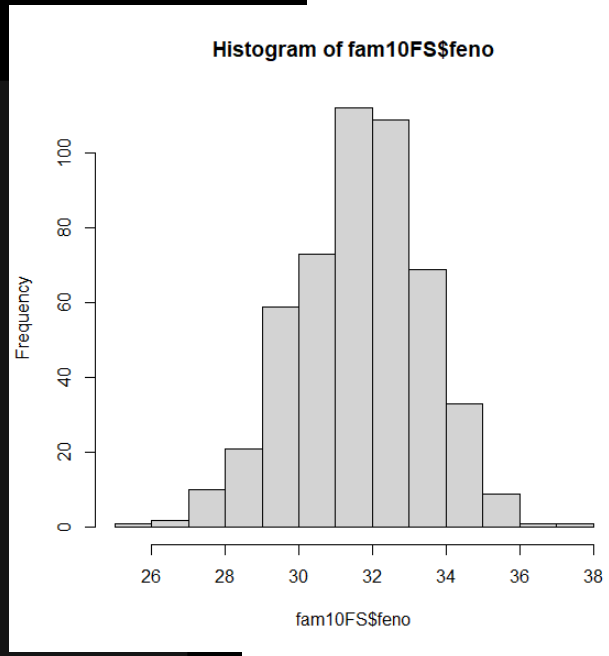
Simulación de 10 familias de 50 hermanos completes

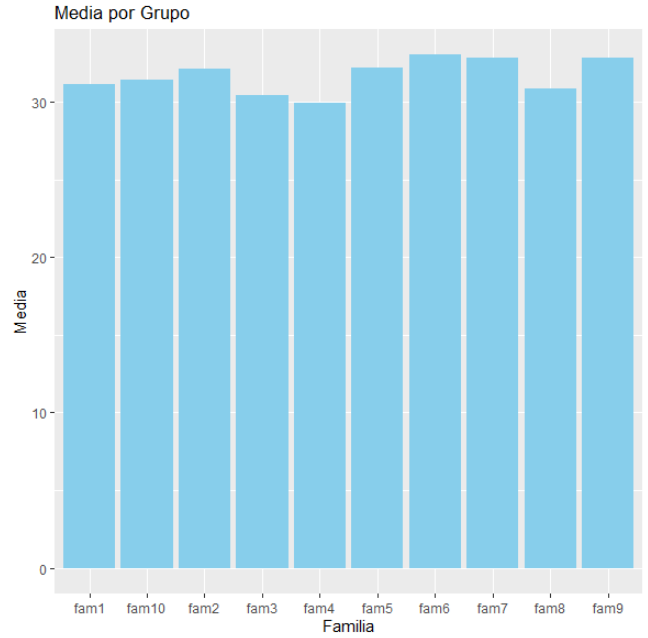


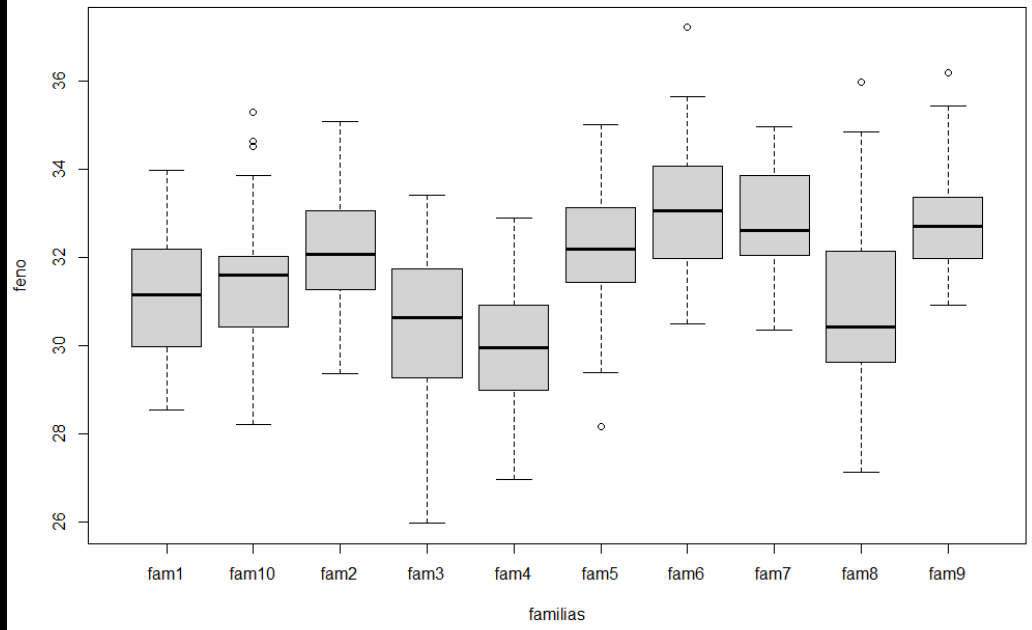
Nota: en el código del ejemplo el objeto nsimout fue generado previamente

```r
# Para generar un fenotipo usamos la función simPheno:
# simPheno(x, Nqtl, Esigma, Pmean, Perror)
# donde x = matriz de genotipos
# Nqtl = número de QTLs (marcadores asociados)
# Esigma = desvío estándar de la distribución Normal de donde se
muestrean los efectos
# Pmean = media del fenotipo
# Perror = desvío estándar del fenotipo

simPheno(simulatedFS, 50, 0.5, 30, 0.5)
#[1] "simP was generated"
str(simP)
#List of 3
# $ pheno   : num [1:500, 1] 31.8 28.5 29.3 ...
# $ QTN     : int [1:50] 516 250 349 821 ...
# $ Meffects: num [1:50] -0.2742 1.2245 -0.2029 0.2292 ...
```

```r
fam10FS = cbind(familias, simP$pheno[1:500])
fam10FS = as.data.frame(fam10FS)
colnames(fam10FS) = c("familias", "feno")
fam10FS$feno = as.numeric(fam10FS$feno)
head(fam10FS)
#  familias     feno
#1     fam1 31.78020
#2     fam1 28.53431
#3     fam1 30.42450

hist(fam10FS$feno)

library(dplyr)
library(ggplot2)

medias_por_grupo <- fam10FS %>%
  group_by(familias) %>%
  summarise(media_grupo = mean(feno))

ggplot(medias_por_grupo, aes(x = factor(familias), y = media_grupo)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Media por Grupo",
       x = "Familia",
       y = "Media")

boxplot(feno ~ familias, data = fam10FS)
```







```r
setwd("E:/MGYGV/2025_clase_practica")
ped = read.csv("pedigree.csv", header = T)

head(ped,30)
```

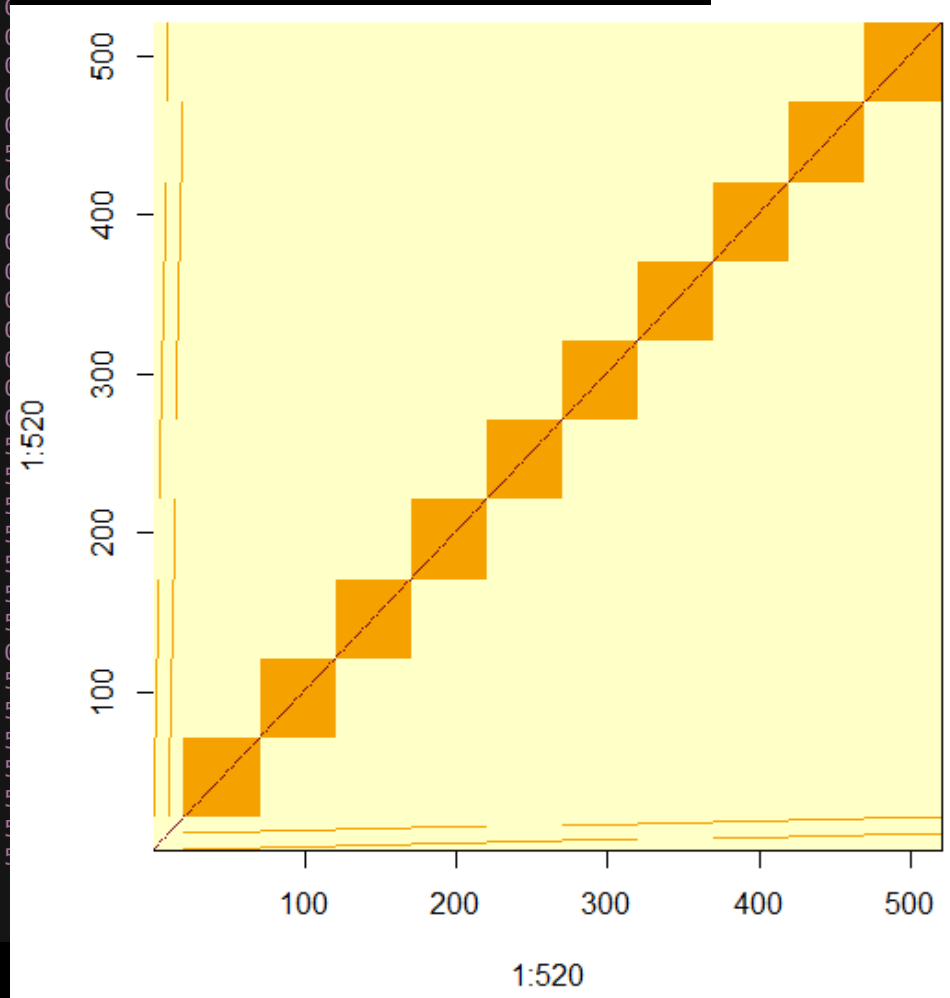
```
   Ind Par1 Par2
1   M1    0    0
2   M2    0    0
3   M3    0    0
4   M4    0    0
5   M5    0    0
6   M6    0    0
7   M7    0    0
8   M8    0    0
9   M9    0    0
10 M10    0    0
11 P11    0    0
12 P12    0    0
13 P13    0    0
14 P14    0    0
15 P15    0    0
16 P16    0    0
17 P17    0    0
18 P18    0    0
19 P19    0    0
20 P20    0    0
21  I1   M1  P11
22  I2   M1  P11
23  I3   M1  P11
24  I4   M1  P11
25  I5   M1  P11
26  I6   M1  P11
27  I7   M1  P11
28  I8   M1  P11
29  I9   M1  P11
30 I10   M1  P11
```

```r
tail(ped, 30)
```

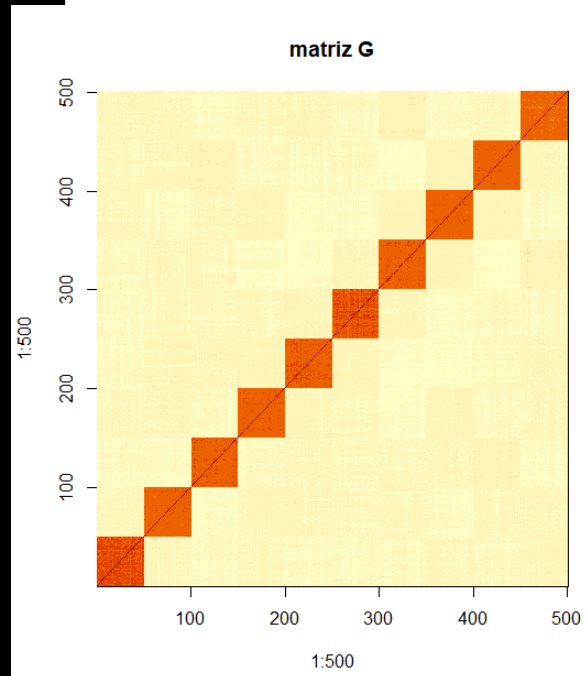
```
     Ind Par1 Par2
491 I471  M10  P20
492 I472  M10  P20
493 I473  M10  P20
494 I474  M10  P20
495 I475  M10  P20
496 I476  M10  P20
497 I477  M10  P20
498 I478  M10  P20
499 I479  M10  P20
500 I480  M10  P20
501 I481  M10  P20
502 I482  M10  P20
503 I483  M10  P20
504 I484  M10  P20
505 I485  M10  P20
506 I486  M10  P20
507 I487  M10  P20
508 I488  M10  P20
509 I489  M10  P20
510 I490  M10  P20
511 I491  M10  P20
512 I492  M10  P20
513 I493  M10  P20
514 I494  M10  P20
515 I495  M10  P20
516 I496  M10  P20
517 I497  M10  P20
518 I498  M10  P20
519 I499  M10  P20
520 I500  M10  P20
```

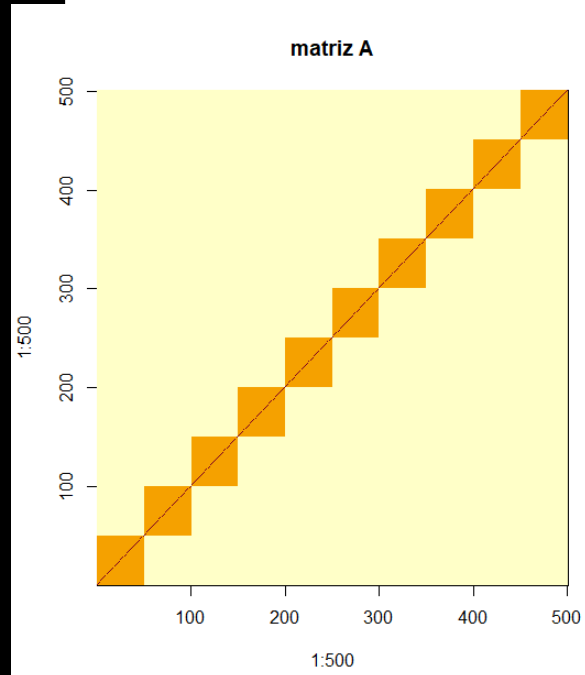
```
A = Amatrix(ped, ploidy=2)
 A[1:35, 1:35]
     M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20  I1  I2  I3  I4  I5  I6  I7  I8
M1  1.0  0  0  0  0  0  0  0  0   0 0.0   0   0   0   0   0   0   0   0   0 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
M2  0.0  1  0  0  0  0  0  0  0   0 0.0   0   0   0   0   0   0   0   0   0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
M3  0.0  0  1  0  0  0  0  0  0   0 0.0   0   0   0   0   0   0   0   0   0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
M4  0.0  0  0  1  0  0  0  0  0   0 0.0   0   0   0   0   0   0   0   0   0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
M5  0.0  0  0  0  1  0  0  0  0   0 0.0   0   0   0   0   0   0   0   0   0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
M6  0.0  0  0  0  0  1  0  0  0   0 0.0   0   0   0   0   0   0   0   0   0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
M7  0.0  0  0  0  0  0  1  0  0   0 0.0   0   0   0   0   0   0   0   0   0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
M8  0.0  0  0  0  0  0  0  1  0   0 0.0   0   0   0   0   0   0   0   0   0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
M9  0.0  0  0  0  0  0  0  0  1   0 0.0   0   0   0   0   0   0   0   0   0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
M10 0.0  0  0  0  0  0  0  0  0   1 0.0   0   0   0   0   0   0   0   0   0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
P11 0.0  0  0  0  0  0  0  0  0   0 1.0   0   0   0   0   0   0   0   0   0 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
P12 0.0  0  0  0  0  0  0  0  0   0 0.0   1   0   0   0   0   0   0   0   0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
P13 0.0  0  0  0  0  0  0  0  0   0 0.0   0   1   0   0   0   0   0   0   0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
P14 0.0  0  0  0  0  0  0  0  0   0 0.0   0   0   1   0   0   0   0   0   0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
P15 0.0  0  0  0  0  0  0  0  0   0 0.0   0   0   0   1   0   0   0   0   0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
P16 0.0  0  0  0  0  0  0  0  0   0 0.0   0   0   0   0   1   0   0   0   0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
P17 0.0  0  0  0  0  0  0  0  0   0 0.0   0   0   0   0   0   1   0   0   0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
P18 0.0  0  0  0  0  0  0  0  0   0 0.0   0   0   0   0   0   0   1   0   0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
P19 0.0  0  0  0  0  0  0  0  0   0 0.0   0   0   0   0   0   0   0   1   0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
P20 0.0  0  0  0  0  0  0  0  0   0 0.0   0   0   0   0   0   0   0   0   1 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
I1  0.5  0  0  0  0  0  0  0  0   0 0.5   0   0   0   0   0   0   0   0   0 1.0 0.5 0.5 0.5 0.5 0.5 0.5 0.5
I2  0.5  0  0  0  0  0  0  0  0   0 0.5   0   0   0   0   0   0   0   0   0 0.5 1.0 0.5 0.5 0.5 0.5 0.5 0.5
I3  0.5  0  0  0  0  0  0  0  0   0 0.5   0   0   0   0   0   0   0   0   0 0.5 0.5 1.0 0.5 0.5 0.5 0.5 0.5
I4  0.5  0  0  0  0  0  0  0  0   0 0.5   0   0   0   0   0   0   0   0   0 0.5 0.5 0.5 1.0 0.5 0.5 0.5 0.5
I5  0.5  0  0  0  0  0  0  0  0   0 0.5   0   0   0   0   0   0   0   0   0 0.5 0.5 0.5 0.5 1.0 0.5 0.5 0.5
I6  0.5  0  0  0  0  0  0  0  0   0 0.5   0   0   0   0   0   0   0   0   0 0.5 0.5 0.5 0.5 0.5 1.0 0.5 0.5
I7  0.5  0  0  0  0  0  0  0  0   0 0.5   0   0   0   0   0   0   0   0   0 0.5 0.5 0.5 0.5 0.5 0.5 1.0 0.5
I8  0.5  0  0  0  0  0  0  0  0   0 0.5   0   0   0   0   0   0   0   0   0 0.5 0.5 0.5 0.5 0.5 0.5 0.5 1.0
I9  0.5  0  0  0  0  0  0  0  0   0 0.5   0   0   0   0   0   0   0   0   0 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
I10 0.5  0  0  0  0  0  0  0  0   0 0.5   0   0   0   0   0   0   0   0   0 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
I11 0.5  0  0  0  0  0  0  0  0   0 0.5   0   0   0   0   0   0   0   0   0 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
I12 0.5  0  0  0  0  0  0  0  0   0 0.5   0   0   0   0   0   0   0   0   0 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
I13 0.5  0  0  0  0  0  0  0  0   0 0.5   0   0   0   0   0   0   0   0   0 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
I14 0.5  0  0  0  0  0  0  0  0   0 0.5   0   0   0   0   0   0   0   0   0 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
I15 0.5  0  0  0  0  0  0  0  0   0 0.5   0   0   0   0   0   0   0   0   0 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
```

```
library(psych)
image(1:520, 1:520, A)
```

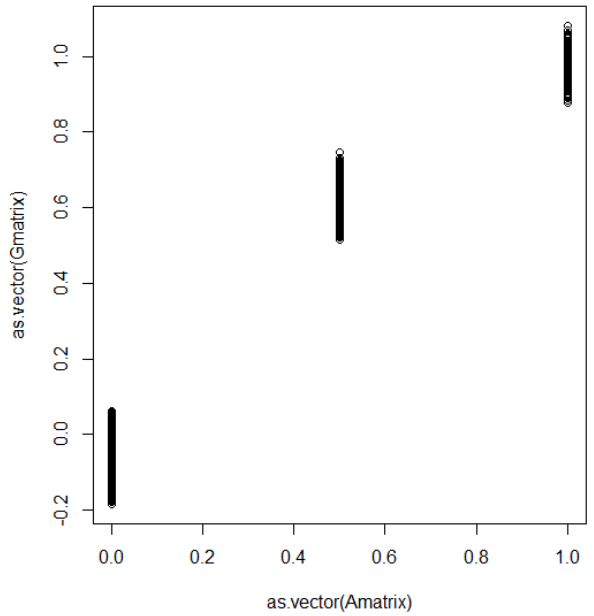


```r
### matriz G: matriz de relaciones realizadas

Gmatrix <- Gmatrix(simulatedFS, method="VanRaden", ploidy=2,
ratio=FALSE)
colnames(Gmatrix) = paste0("I", c(1:500))
rownames(Gmatrix) = paste0("I", c(1:500))
Gmatrix[1:5,1:5]
#          I1        I2        I3        I4        I5
#I1 1.0206027 0.6396616 0.6499071 0.6721293 0.6742828
#I2 0.6396616 1.0124361 0.6458238 0.6576928 0.6412104
#I3 0.6499071 0.6458238 1.0122208 0.6803622 0.6452440
#I4 0.6721293 0.6576928 0.6803622 1.0111109 0.6653956
#I5 0.6742828 0.6412104 0.6452440 0.6653956 0.9802168

image(1:500, 1:500, Gmatrix, main = "matriz G")

Amatrix = A[-c(1:20), -c(1:20)]
 dim(Amatrix)
#[1] 500 500
Amatrix[1:10,1:10]
#     I1  I2  I3  I4  I5  I6  I7  I8  I9 I10
#I1  1.0 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
#I2  0.5 1.0 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
#I3  0.5 0.5 1.0 0.5 0.5 0.5 0.5 0.5 0.5 0.5
#I4  0.5 0.5 0.5 1.0 0.5 0.5 0.5 0.5 0.5 0.5
#I5  0.5 0.5 0.5 0.5 1.0 0.5 0.5 0.5 0.5 0.5
#I6  0.5 0.5 0.5 0.5 0.5 1.0 0.5 0.5 0.5 0.5
#I7  0.5 0.5 0.5 0.5 0.5 0.5 1.0 0.5 0.5 0.5
#I8  0.5 0.5 0.5 0.5 0.5 0.5 0.5 1.0 0.5 0.5
#I9  0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 1.0 0.5
#I10 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 1.0

image(1:500, 1:500, Amatrix, main = "matriz A")
```

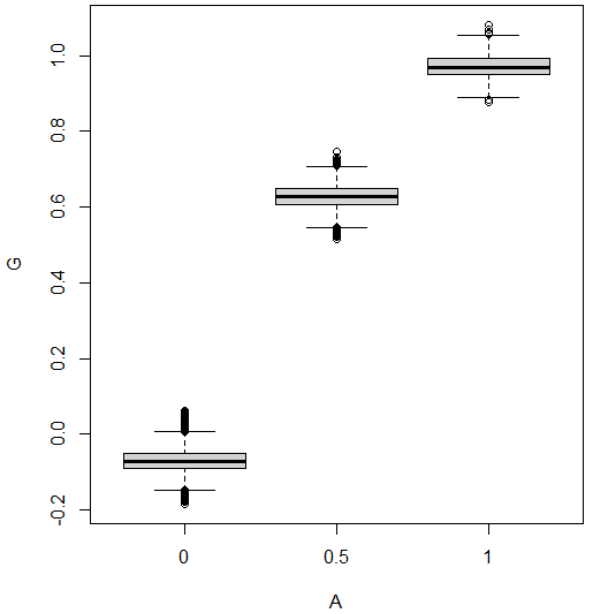




```r
cor(as.vector(Amatrix), as.vector(Gmatrix))
#[1] 0.9881251

plot(as.vector(Amatrix), as.vector(Gmatrix))
```




```r
pedG = cbind(as.vector(Amatrix), as.vector(Gmatrix))
colnames(pedG) = c("A", "G")
head(pedG)
boxplot(G ~ A, data = pedG)
```



Heredabilidad (h^2)

```r
str(simP)
#List of 3
 #$ pheno   : num [1:500, 1] 31.8 28.5 30.4 30.9 29.3 ...
 #$ QTN     : int [1:50] 516 250 349 185 218 480 936 177 821 389 ...
 #$ Meffects: num [1:50] -0.2742 1.2245 -0.2029 0.0898 0.2292 ...

# nuestra simulación está basada en:  y = mu + bX + e

y = simP$pheno
X = simulatedFS[, simP$QTN]
b = simP$Meffects

# Calcular valores genéticos
g <- X %*% b

# Calcular varianzas y heredabilidad
sigma2_g <- var(g)
sigma2_p <- var(y)
h2 <- sigma2_g / sigma2_p

print(paste("Heredabilidad estimada:", round(h2, 3)))
#[1] "Heredabilidad estimada: 0.913"
```

$$y = \mu + g + e$$

$$\sigma_p^2 = \mathrm{Var}(y)$$

$$h^2 = \frac{\sigma_g^2}{\sigma_p^2}$$

Alternativa (¡datos simulados!):

$$\sigma_p^2 = \sigma_g^2 + \sigma_e^2$$

$0.5^2 = 0.25$

```r
rrblup <- function(x, pheno, cv, part, projName){       # x es la matriz de genotipos
#generacion de index                                    # pheno es el phenotipo
train <- round(part/100*nrow(x), digits = 0)            # part es el porcentaje pob entrenamiento
test <- nrow(x)-train                                   # projName es el "nombre_del_proyecto"
if(isTRUE(test+train == nrow(x)) == TRUE) {
xtrain <- rep(1,train)
xtest <- rep(2,test)
xcv <- c(xtrain, xtest)
index <- matrix(nrow = nrow(x), ncol = cv, NA)
for (i in 1:cv) {
index[,i] <- sample(xcv)
}
} else {
print("train+test es distinto de nrow(x), adecuar valores")
}
##########################################################################
datas <- cbind(x, pheno)
sets1  <- index
datas2 <- population
datas3 <- feno
ntest <- test
for(fold in 1:cv){
 correlations <- matrix(NA,1,6)
 datos        <- matrix(NA,1,6)
```

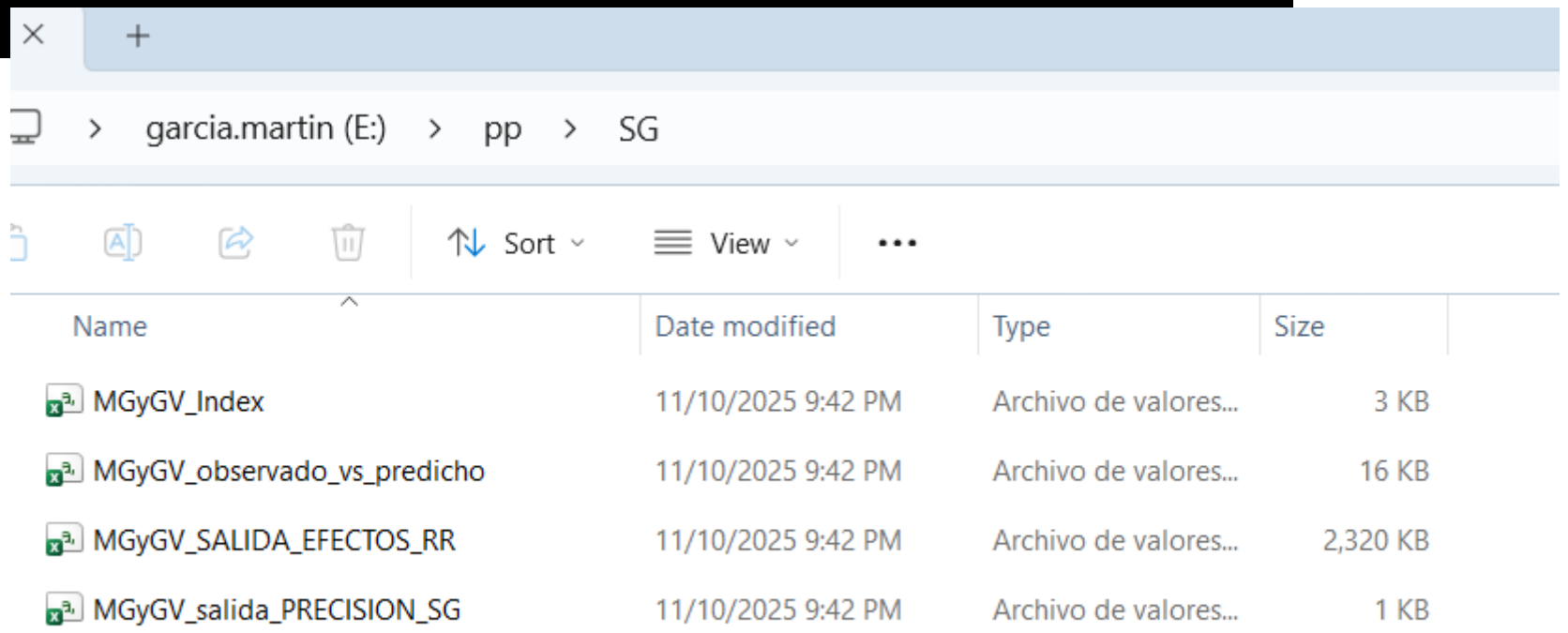
```r
##########################################################################
  itrain <- which(sets1[,fold]==1)
  itest  <- which(sets1[,fold]==2)
  test   <- datas[itest,]
  train  <- datas[itrain,]
  Xtest  <- test[,-ncol(test)]
  Ytest  <- test[,ncol(test)]
  Xtrain <- train[,-ncol(test)]
  Ytrain <- train[,ncol(test)]
##########################################################################
indice <- 1
for (columna1 in 1:(ncol(Xtrain)-1)){
    for (columna2 in (columna1+1):ncol(Xtrain)){
        dd <- sum(abs(Xtrain [,columna1]- Xtrain [,columna2]))
    if(dd==0){
        indice <- c(indice,columna1)
}}}
##  RR-BLUP       ########################################################
  X1     <- rbind(Xtrain,Xtest)
  X1     <- X1[,-indice]
  y1     <- c(Ytrain,Ytest)
  yNa1   <- y1
  train1 <- 1:nrow(Xtrain)
  f      <-  nrow(Xtrain)+1
  pred1  <- f:nrow(X1)
  ans    <- mixed.solve(y=y1[train1],Z=Xtrain) #By default K = I
  intercepto <- rep(ans$beta,ntest)
###
efectoRR  <- ans$u
efectoRR2 <- abs(efectoRR)
```

```
  y1     <- c(ytrain,ytest)
  yNa1   <- y1
  train1 <- 1:nrow(Xtrain)
  f      <-  nrow(Xtrain)+1
  pred1  <- f:nrow(X1)
  ans    <- mixed.solve(y=y1[train1],Z=Xtrain) #By default K = I
  intercepto <- rep(ans$beta,ntest)
###
efectoRR  <- ans$u
efectoRR2 <- abs(efectoRR)
num <- length(efectoRR)
rr  <- cbind(fold,1:num,efectoRR,efectoRR2)
newdata <- rr[order(-efectoRR2),]
yTestHat   <- intercepto + Xtest %*% ans$u
accuracyRR <- cor(Ytest,yTestHat)
prediccion <- paste(fold,Ytest,yTestHat, sep = ",")
write.table(prediccion, file = paste0(projName,
"_observado_vs_predicho.csv"),row.names=FALSE,col.names=FALSE,append=TRUE,sep=",")
write.table(newdata,file= paste0(projName,
"_SALIDA_EFECTOS_RR.csv"),row.names=FALSE,col.names=FALSE,append=TRUE,sep=",")
write.table(accuracyRR,file= paste0(projName,
"_salida_PRECISION_SG.csv"),row.names=FALSE,col.names=FALSE,append=TRUE,sep=",")
    }
write.table(index,file= paste0(projName, "_Index.csv"))
}
```

```r
#setear la semilla (VAMOS A SIMULAR MENOS DATOS QUE UTILIZAMOS EN LA
PRACTICA DE GWAS DEBIDO A LOS TIEMPOS DE COMPUTACION)
 set.seed(1234)

 #SIMULACION DE DATOS
 Nind <- 100
 Nmarkers <- 2000
 Nqtl <- 50
 Esigma <- .5
 Pmean <- 25
 Perror <- .25

 simulN(Nind, Nmarkers, Nqtl, Esigma, Pmean, Perror)
#[1] "nsimout was generated"
  str(nsimout)
#List of 4
# $ geno    : num [1:100, 1:2000] 0 1 1 1 2 1 0 0 1 1 ...
# $ pheno   : num [1:100, 1] 28.6 25.2 24.9 26.4 28 ...
# $ QTN     : int [1:50] 1042 1352 264 1097 1191 1884 1297 80 1793 1056 ...
# $ Meffects: num [1:50] -0.3551 -0.0335 0.1468 -0.5696 0.6242 ...

#los QTLs
   QTL <- cbind(nsimout$QTN, nsimout$Meffects)
   QTL <- cbind(QTL,abs(nsimout$Meffects))
   colnames(QTL) <- c("marker", "effect", "effabs")
   QTL <- as.data.frame(QTL)
   QTL <- QTL[order(-QTL$effabs),]
 head(QTL)
#   marker     effect    effabs
#17    526  0.8756740 0.8756740
#37   1199 -0.8462334 0.8462334
#36   1105  0.7025707 0.7025707
#47    505  0.7021062 0.7021062
#32   1964  0.6850347 0.6850347
#43   1310 -0.6259552 0.6259552
```

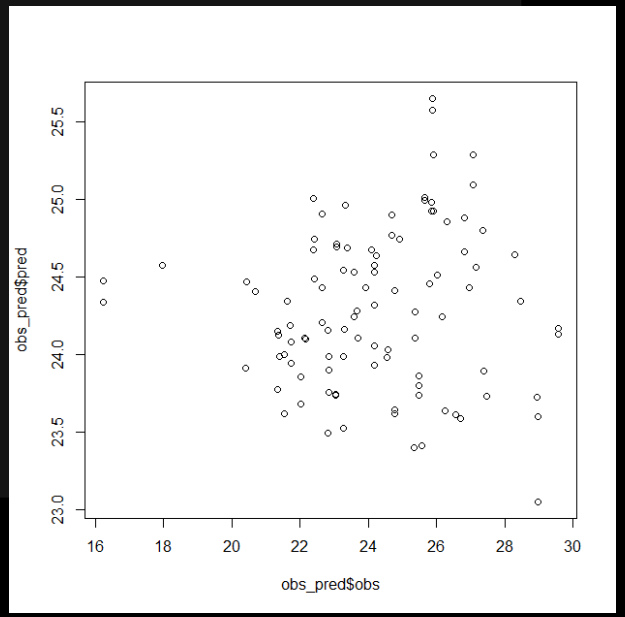
```r
#matriz de datos genomicos
population <- nsimout$geno
colnames(population) <- c(paste("M", 1:Nmarkers,sep = ""))
rownames(population) <- c(paste("IND", 1:Nind,sep = ""))

#matriz de datos fenotipicos
popfeno <- nsimout$pheno
colnames(popfeno) <- "PHENO"
rownames(popfeno) <- c(paste("IND", 1:Nind,sep = ""))
feno <- data.frame ( IND = rownames(popfeno),
                     Pheno = popfeno)

#mapa
map <- data.frame (SNP = colnames(population),
                   Chromosome = c(rep(1,(Nmarkers/5)),rep(2,(Nmarkers/5)),rep(3,
(Nmarkers/5)),rep(4,(Nmarkers/5)),rep(5,(Nmarkers/5))),
                   Position = c(1:(Nmarkers/5),1:(Nmarkers/5),1:(Nmarkers/5),1:
(Nmarkers/5),1:(Nmarkers/5))
                   )
```

```r
##########
##  SG  ##
##########
#setear el directorio de trabajo
setwd("E:/PP/SG")

rrblup(population, feno$PHENO, 10, 90, "MGyGV")        # el tiempo varia según la PC, entre 2 y 10 minutos
```

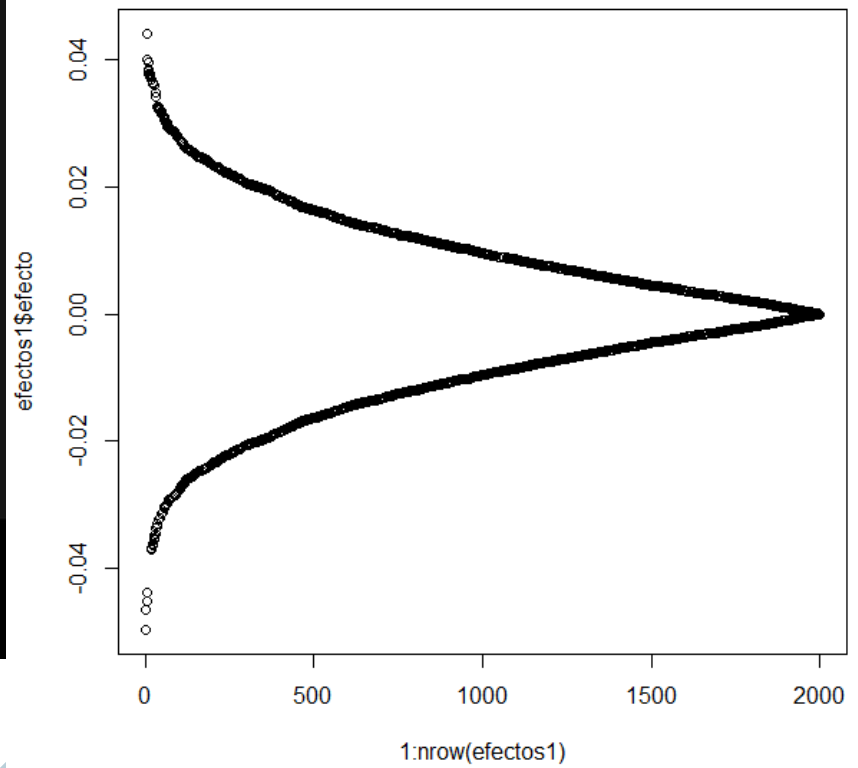


```
 particiones = read.csv("MGyGV_Index.csv", header = T,sep = " ")
dim(particiones)
#[1] 100  10
head(particiones)
#  V1 V2 V3 V4 V5 V6 V7 V8 V9 V10
#1  1  1  1  1  1  2  2  1  1   1
#2  1  1  1  1  1  1  1  1  2   1
#3  1  1  1  1  1  1  1  1  2   1
#4  1  2  2  1  1  1  1  1  1   1
#5  1  1  1  1  1  1  1  1  1   1
#6  1  1  1  1  1  1  1  1  1   1
```

```r
obs_pred = read.csv("MGyGV_observado_vs_predicho.csv", header = F)
library(tidyr)
obs_pred <- separate(obs_pred, col = V1, into = c("fold", "obs", "pred"), sep = ",")
dim(obs_pred)
#[1] 100   3
head(obs_pred)
#  fold              obs             pred
#1    1 28.9750471553632 23.0513864686126
#2    1 26.2600534821393 23.6357422188851
#3    1  28.959321814666 23.7242815760948
#4    1 25.6649445953692 24.9948838368234
#5    1 28.4614136754116 24.3454767743638
#6    1 23.0479323862354  23.744430075029

plot(obs_pred$obs, obs_pred$pred)
```



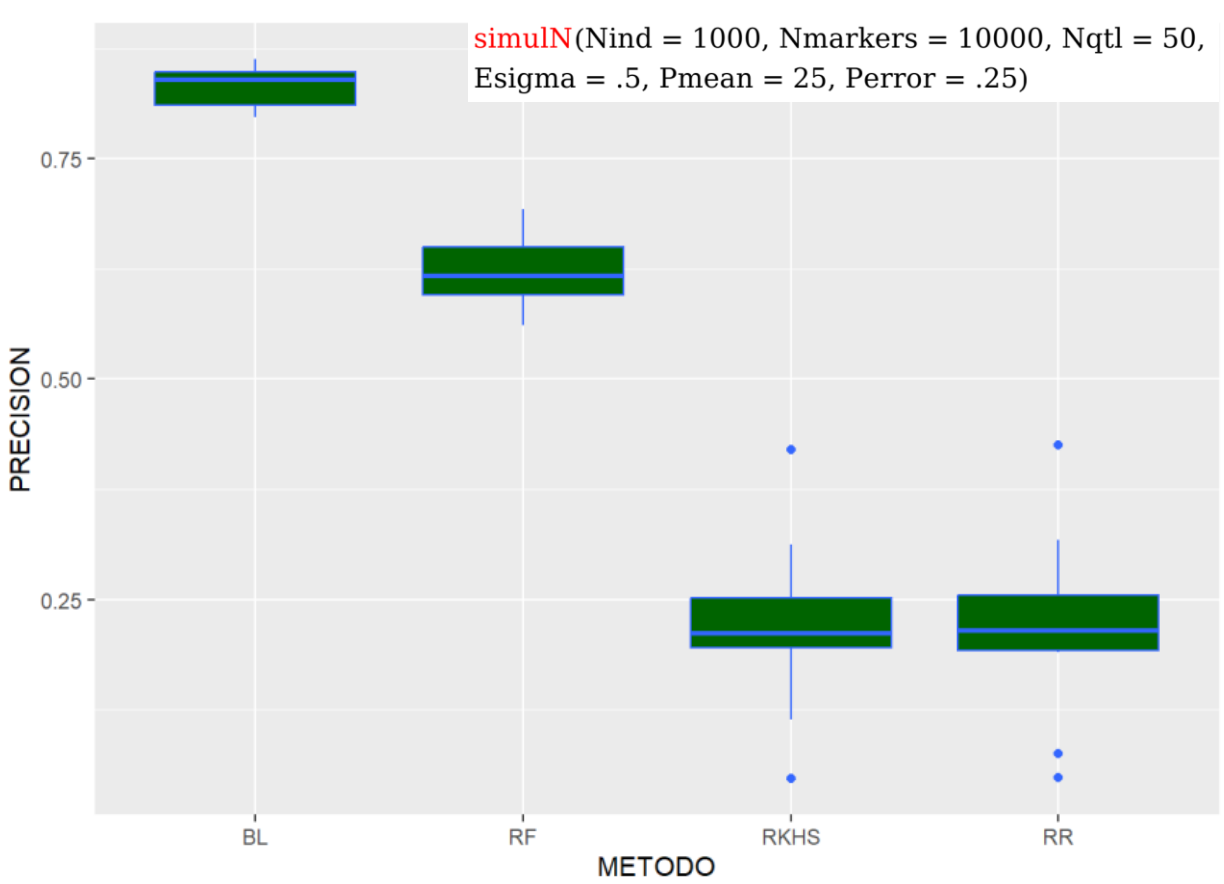
```r
efectos = read.csv("MGyGV_SALIDA_EFECTOS_RR.csv", header = F,sep = " ")
efectos <- separate(efectos, col = V1, into = c("fold", "SNP","efecto",
"efecto.absoluto"), sep = ",")
dim(efectos)
#[1] 20000     4
head(efectos)
#  fold  SNP              efecto    efecto.absoluto
#1    1 1240 -0.0497026093865756 0.0497026093865756
#2    1 1138 -0.0466129145611782 0.0466129145611782
#3    1  991 -0.0452910898941934 0.0452910898941934
#4    1 1949  0.0441564630714585 0.0441564630714585
#5    1 1521 -0.0439620807349064 0.0439620807349064
#6    1 1379  0.0401579246852856 0.0401579246852856
efectos1 = efectos[efectos$fold == "1", ]
plot(1:nrow(efectos1), efectos1$efecto)
```

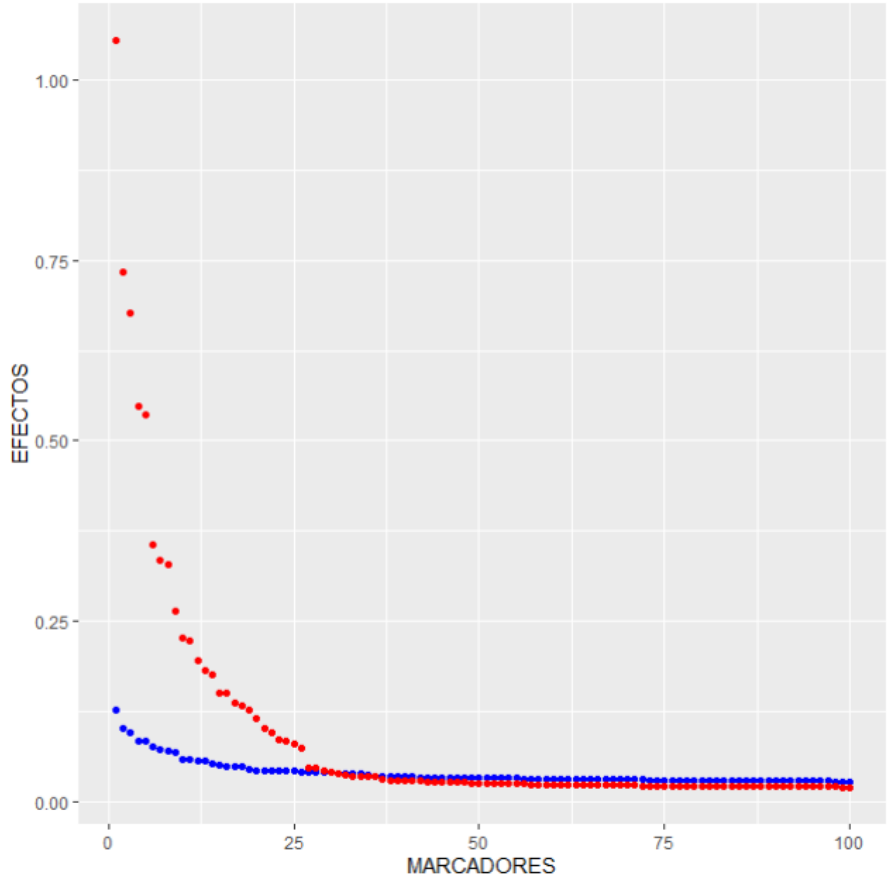


```r
precision = read.csv("MGyGV_salida_PRECISION_SG.csv", header = F,sep = " ")
dim(precision)
#[1] 10  1
precision
#            V1
#1  -0.48557006
#2   0.41243286
#3   0.44745927
#4   0.09754729
#5   0.63536423
#6   0.41240010
#7   0.65057955
#8  -0.17584973
#9  -0.24334551
#10  0.61555296
mean(precision$V1)
#[1] 0.2366571
```

SG

RR: Ridge Regression (regresión lineal)
BL: Bayesian LASSO (regresión lineal bayesiana)
RF: Random Forest (semiparamétrica)
RKHS: Reproducing Kernel Hilbert Spaces (no paramétrica)



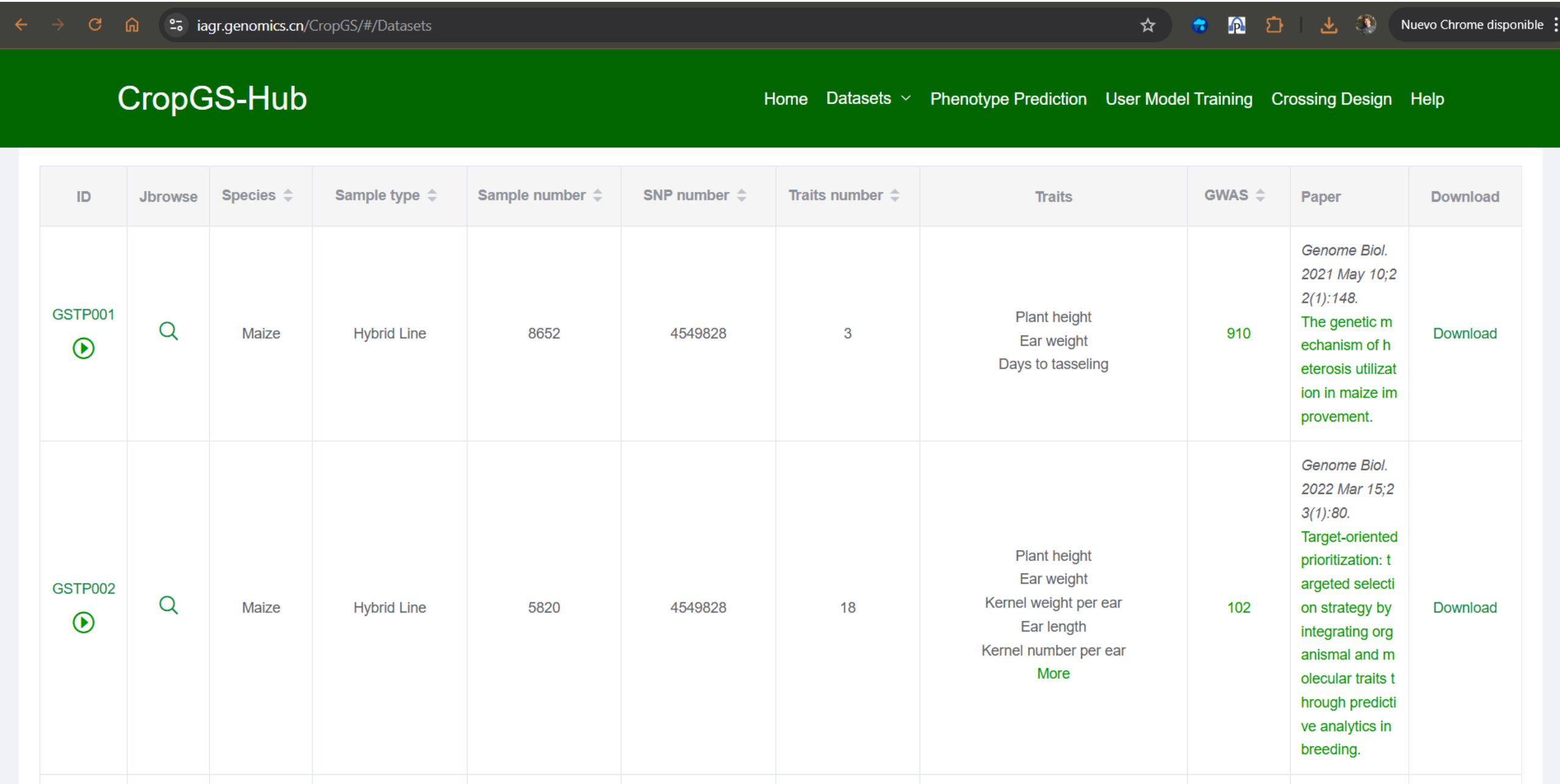


BL

RR

- Endelman, J. B. (2011). Ridge regression and other kernels for genomic selection with R package rrBLUP. The plant genome, 4(3).
- de Los Campos, et al., (2013). Genome-enabled prediction using the BLR (Bayesian Linear Regression) R-package. Genome-wide association studies and genomic prediction, 299-320.
- Breiman, L. (2001). Random forests. Machine learning, 45, 5-32.
- Pérez, P., & de Los Campos, G. (2014). Genome-wide regression and prediction with the BGLR statistical package. Genetics, 198(2), 483-495.

Extra: datos reales SNP: https://iagr.genomics.cn/CropGS/#/Datasets



ID	Jbrowse	Species	Sample type	Sample number	SNP number	Traits number	Traits	GWAS	Paper	Download
GSTP001		Maize	Hybrid Line	8652	4549828	3	Plant height Ear weight Days to tasseling	910	*Genome Biol. 2021 May 10;22(1):148.* The genetic mechanism of heterosis utilization in maize improvement.	Download
GSTP002		Maize	Hybrid Line	5820	4549828	18	Plant height Ear weight Kernel weight per ear Ear length Kernel number per ear More	102	*Genome Biol. 2022 Mar 15;23(1):80.* Target-oriented prioritization: targeted selection strategy by integrating organismal and molecular traits through predictive analytics in breeding.	Download

Imputation of Unordered Markers and the Impact on Genomic Selection Accuracy

Jessica E Rutkoski, Jesse Poland, Jean-Luc Jannink, Mark E Sorrells ✉ Author Notes

G3 Genes|Genomes|Genetics, Volume 3, Issue 3, 1 March 2013, Pages 427–439,
https://doi.org/10.1534/g3.112.005363
Published: 01 March 2013

Low-call-rate SNPs and presence–absence variation identified in the rice pan-genome can improve genomic prediction of rice gene bank accessions

Original Article | Open access | Published: 07 November 2025
Volume 138, article number 295, (2025) Cite this article

- Rutkoski, J. E., Poland, J., Jannink, J. L., & Sorrells, M. E. (2013). Imputation of unordered markers and the impact on genomic selection accuracy. *G3: Genes, Genomes, Genetics*, *3*(3), 427-439.
- Krusenbaum, L., Wissuwa, M. Low-call-rate SNPs and presence–absence variation identified in the rice pan-genome can improve genomic prediction of rice gene bank accessions. *Theor Appl Genet* **138**, 295 (2025). https://doi.org/10.1007/s00122-025-05080-x