Run pSBVB addtional functions

14 de septiembre de 2018

Table of Contents

Purpose 1
Main features 2
Installation 2
Prepare your files 3
Function to create a pedigree to use as .ped file 3
Generate a pedigree based relationship matrix 4
Application examples 4
Example: Generate strawberry simulated genotype + phenotype dataset using the previous pedigree file and several options 4
Statistical Models: 6
Running Model 1 6
Running Model 2 8
Running Model 3 9
Examples with potato dataset: 10

Purpose

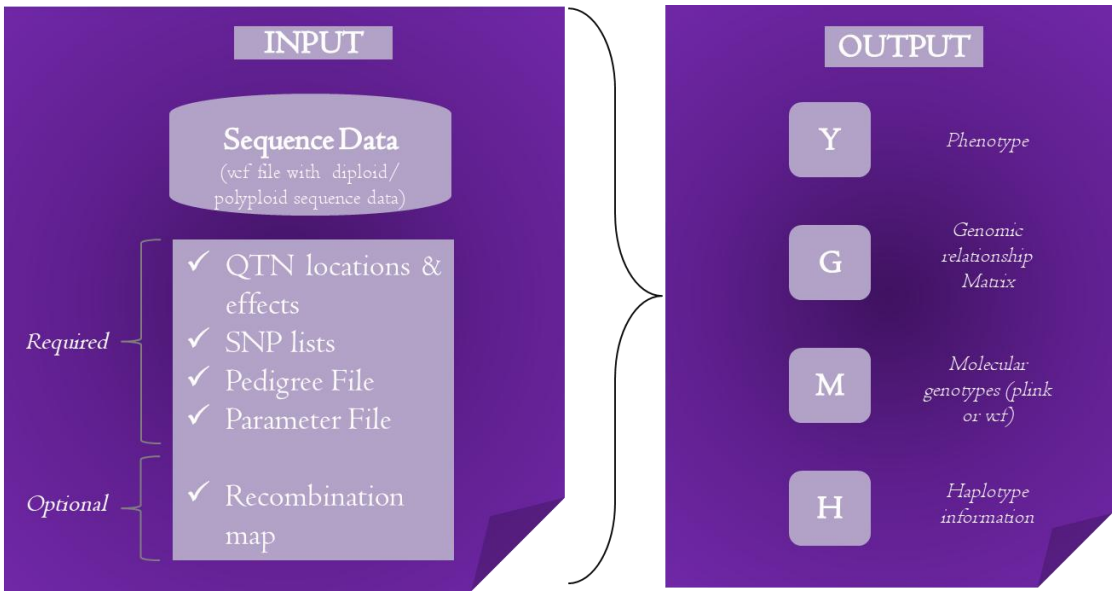
Polyploidy is a very common event in plants and in some fish species, too. However, the application of Genomic Selection (GS) in polyploids organism is still scarse and to the date, a few number of studies has been published.

Computer simulation allows the exploration of a wide range of hypothesis at low cost and they can help to interpret the outcome of selection in complex situations. The purpose of this study is to develop a flexible simulation tool and to propose several approaches to compute the molecular relationship matrix adapted to polyploids, as well as the implementation of phenotypes simulations.

The current framework shows how the program works using a toy example of octoploid strawberry (F. ananassa) SNPs and a toy dataset from autotetraploid potato.

As input, the software requiered a chipset (list of markers through the genome), a vcf file (lists with genotypes in vcf format), a pedigree file (file with crosses to perform simulations using gene dropping) and a parameter file. In order to facilitate the software usage, we recommend that you have all these files in a same folder. The software outputs are

phenotypes and genotypes matrices able to perform GS models. The next figure shows the general Software' view.



General Software' View

Main features

- Any number of traits.
- Tool adapted to work with both, auto and allo-polyploid organisms.
- Any number of QTNs, trait specific.
- Any number of additive and dominant effects.
- Can generate a correlation matrix to modelate meiosis in polyploid especies.
- Can generate correlated allelic effects and frequencies.
- Efficient algorithms to generate haplotypes and sample SNP genotypes.
- Computes genomic relationship matrices for any number of SNP arrays simultaneously.
- It allow to compute Genomic relationship matrix in several ways.
- Any number of chromosomes, allows for sex chromosomes and varying local recombination rates, that can be sex specific.

Installation

The source code, manual and examples can be obtained from https://github.com/lzingaretti/pSBVB

To compile:

```
gfortran -O3 kind.f90 ALliball.f90 aux_sub11.f90 psbvb.f90 -o sbvb -lblas
```

or

```
make
```

To install in /usr/local/bin

```
sudo make install
```

The program requires blas libraries but these are standard in any unix or OS mac system. We have tested pSBVB only in linux with gfortran compiler; intel ifort seems not working, but gfortran in mac OS looks ok. Usage

To run:

```
 file.vcf psbvb -isbvb.par
```

To run the program with the same random seed:

```
… | psbvb –isbvb.par –seed iseed
```

where iseed is an integer number

Prepare your files

Function to create a pedigree to use as .ped file

Pedigree file is required to simulations. The number of founder individuals have to be equal or higher than the number of inviduals in .gen (or .vcf) file. The founders are those individuals whith genomic information.

We provide a R function to generate pedigree file. You can choose any number of founder, generation, individuals by generation and sex specification is optional as well.

Example 1: Generate a pedigree file

The next pedigree file have 47 founders, 8 generations, 100 individuals of each generations from 1 to 7. The last generation have 1000 individuals. Sex is not considerer.

```
source("pedigree.R")
M<-pedgenerator(100,4,c(rep(100,3),150),
   sex=FALSE,
   path="~/Dropbox/DoctoradoCRAG/paper-1/article/software-help/reversinn1/pSB
VB/
   toy_strawberry/Additional_functions",exclude=47)
tail(M$pedigree)
```

Example 2: Generate a pedigree file with sex option

The next pedigree file have 47 founders, 8 generations, 100 individuals of each generations from 1 to 7. The last generation have 1000 individuals.

```
source("pedigree.R")
M<-pedgenerator(100,4,c(rep(100,3),150),sex=FALSE,
   path="~/Dropbox/DoctoradoCRAG/paper-1/article/software-help/reversinn1/pSB
VB/toy_strawberry/
   Additional_functions",exclude=47)

tail(M$pedigree)
```

Generate a pedigree based relationship matrix

We had implemented a R function to generate a Relationship matrix to compare with genomic relationship matrix. Our function could be used to compute both, additive and dominant relationship matrices. To do the calculation, a pedigree file is needed as input. You can to use the pedigree matrix generated with pedigree.R as input.

```
source("RelationshipMatrix.R")
data<-read.table("~/Dropbox/DoctoradoCRAG/paper-1/article/software-help/rever
sinn1/pSBVB
                 /toy_strawberry/Additional_functions/File_st.ped",header=FAL
SE)
#check the dimensions of dataset
dim(data)
A<-RelMatrix(data,dominance=FALSE,path= "~/Dropbox/DoctoradoCRAG/paper-1/arti
cle/software-help/reversinn1/pSBVB/toy_strawberry/Additional_functions/")
tail(A)
```

Application examples

Example: Generate strawberry simulated genotype + phenotype dataset using the previous pedigree file and several options.

The parameter file (see file_1.par) incorporate three "H2" parameter of 0.3, 0.4 and 0.5. 150 causals SNP's to simulate the phenotype. This file simulated simultaneosly three pheno types with three additive effects. The Genomic Relationship matrix is generated using Gtota, then the M values varying between 0 and 8. The others files have the next options:

- file_4.par –> heredability of 0.5, generate 1 trait. Use GT as genomic relationship matrix. Additive and dominat effects are simulated from a gamma distribution with mean=0.2, shape= 0.2 and mean= 0.2, shape= 0.5 parameters respectively.
- file_5.par –> heredability of 0.3, generate 1 trait. Use GT as genomic relationship matrix, 80 QTNS

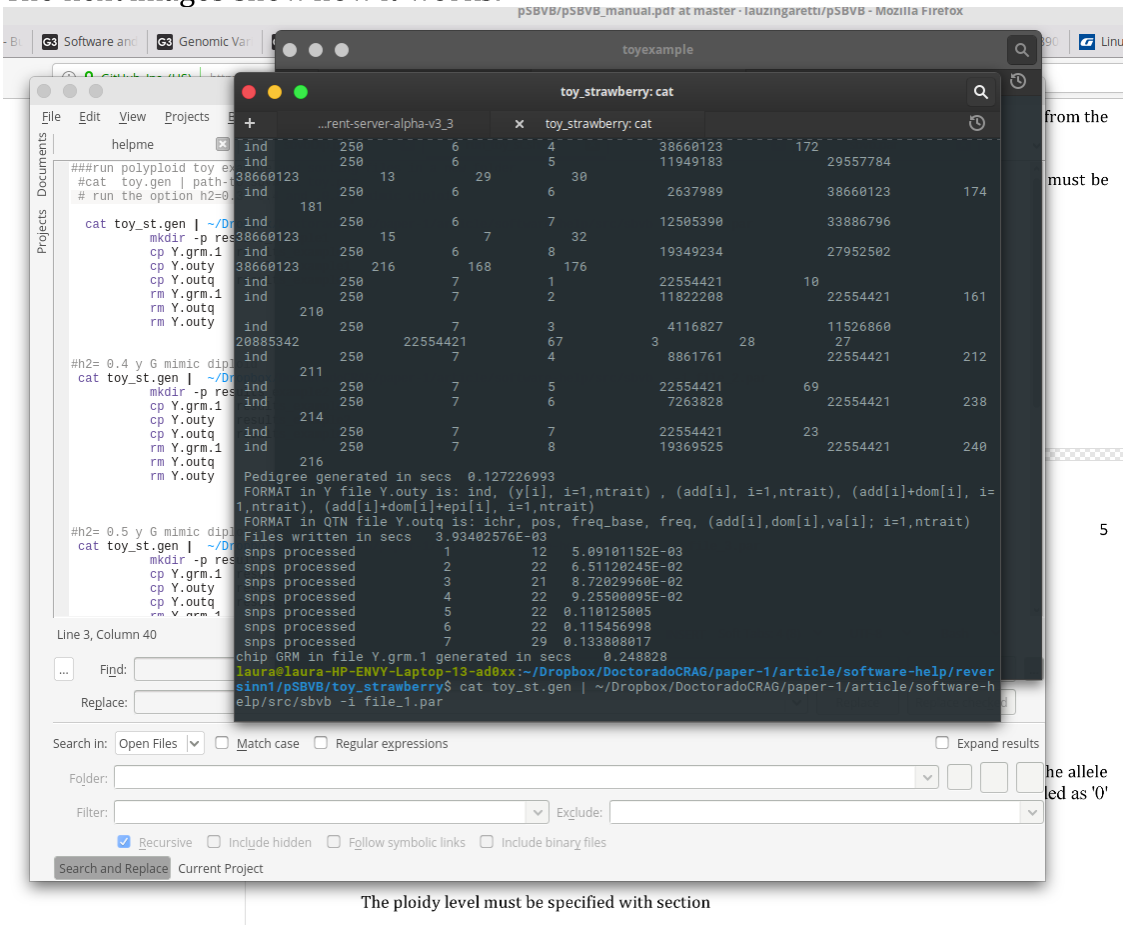
You can run all files simultaneosly using run_toy_st.sh file. You could run the program once:

```
cat toy_st.gen | sbvb -i file_1.par
```

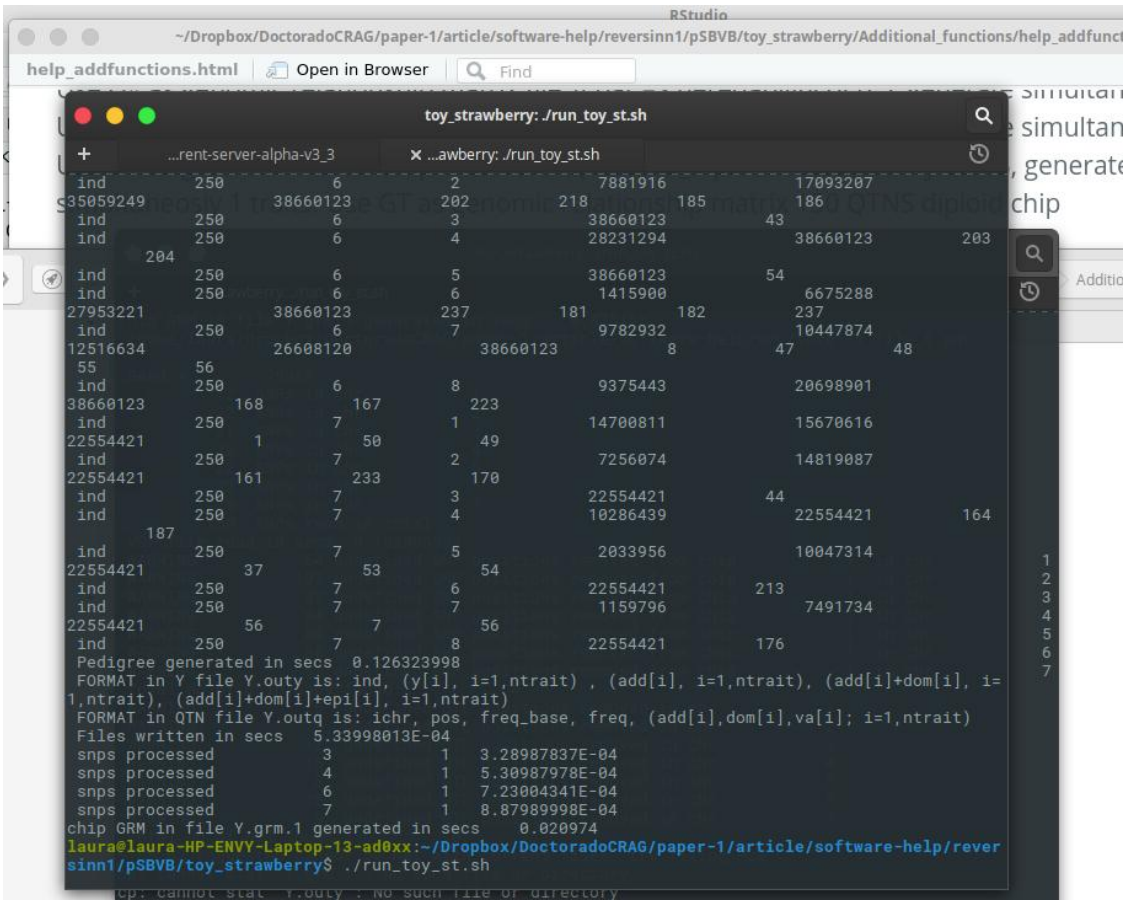
or using a .sh file:

```
./run_toy_st.sh
```

The next images show how it works:



The ploidy level must be specified with section



Run the program several times simultaneosly using a bash script

Statistical Models:

There are numerous GS methods that use genome-wide markers to predict breeding values and address the large small problem. Here, breeding values were predicted using GBLUP, which is a genomic-based extension of traditional BLUP (Henderson, 1984) and equivalent to Ridge Regression Model (RR-BLUP). The model is:

The R script to perform Predictive abilities is GBlupFunction.R

Running Model 1

Here, we include the output from three models simultaneosly and we compared them with Pedigree Based models. Finally, we plot the estimated vs. true values. We only graph the thirth model (heredability parameter was 0.5).

```
library(ggplot2)
source("GBlupFunction.R")
setwd("/home/laura/Dropbox/DoctoradoCRAG/paper-1/article/software-help/reversinn1/pSBVB/toy_strawberry/results_example1/")
```

```r
G<- read.table("Y.grm.1")
y<- read.table("Y.outy")
#check matrix dimention
dim(y)
```

```
## [1] 503  13
```

```r
dim(G)
```

```
## [1] 503 503
```

```r
#In the example one, we had been simulated three phenotypes simultaneosly. Th
e GBlup Prediction, could be used to evaluated predictive of those traits abi
lity simultaneosly.
h2=c(0.3,0.4,0.5)
ntraits=3
make_predictions=c(353:503)
library(ggplot2)
S<-GBlup_predict(G=G,y=y,h2=h2,make_predictions=make_predictions,ntraits=ntra
its)
S[[3]]$rho
```

```
## [1] 0.649779
```

```r
Phenohat<-S[[3]]$uhat[make_predictions] + S[[3]]$mean
Pheno<-S[[3]]$yout_corr[make_predictions]
cor(Pheno,Phenohat)
```

```
## [1] 0.649779
```

```r
datos<-data.frame(Phenohat,Pheno)
colnames(datos)<-c("Yhat","Y")
apply(datos,2,class)
```

```
##      Yhat         Y
## "numeric" "numeric"
```

```r
ggplot(datos, aes(y=Yhat, x=Y,colour="red")) +
geom_point(size=0.6) + scale_shape_manual(values=c(2,4)) +
ggtitle("Estimated values for testing set - Genomic matrix") +
geom_abline(intercept=lm(Yhat ~ Y,data=datos)$coefficients[1], slope=lm(Yhat
~ Y,data=datos)$coefficients[2])+
#scale_x_continuous(limits = c(min(datos[,1]), max(datos[,1])))+
#scale_y_continuous(limits = c(min(datos[,2]), max(datos[,2])))+

theme(panel.background = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), legend.position ="none")
```

Running Model 2

Model 4 incorporates dominant effects

```
source("GBlupFunction.R")
setwd("/home/laura/Dropbox/DoctoradoCRAG/paper-1/article/software-help/reversinn1/pSBVB/toy_strawberry/results_example4/")

G<- read.table("Y.grm.1")
y<- read.table("Y.outy")
#check matrix dimention
dim(y)

## [1] 503   5

dim(G)

## [1] 503 503

setwd("/home/laura/Dropbox/DoctoradoCRAG/paper-1/article/software-help/reversinn1/pSBVB/toy_strawberry/")

R<-read.table("Relationship.mat")
#check matrix dimention
dim(R)
```

```
## [1] 550 550

R<-R[c(48:550),c(48:550)]
#In the example one, we had been simulated three phenotypes simultaneosly. Th
e GBlup Prediction, could be used to evaluated predictive of those traits abi
lity simultaneosly.
h2=0.5
ntraits=1
make_predictions=c(354:503)

S<-GBlup_predict(G=G,y=y,h2=h2,make_predictions=c(354:503),ntraits=ntraits)
SR<-GBlup_predict(G=R,y=y,h2=h2,make_predictions=c(354:503),ntraits=ntraits)
S$rhoM

## [1] 0.6075785

SR$rhoM

## [1] 0.5100464
```

Running Model 3

```
source("GBlupFunction.R")
setwd("/home/laura/Dropbox/DoctoradoCRAG/paper-1/article/software-help/revers
inn1/pSBVB/toy_strawberry/results_example5/")

G<- read.table("Y.grm.1")
y<- read.table("Y.outy")
#check matrix dimention
dim(y)

## [1] 503   5

dim(G)

## [1] 503 503

setwd("/home/laura/Dropbox/DoctoradoCRAG/paper-1/article/software-help/revers
inn1/pSBVB/toy_strawberry/")

R<-read.table("Relationship.mat")
#check matrix dimention
dim(R)

## [1] 550 550

R<-R[c(48:550),c(48:550)]
#In the example one, we had been simulated three phenotypes simultaneosly. Th
e GBlup Prediction, could be used to evaluated predictive of those traits abi
lity simultaneosly.
h2=0.3
```

```
ntraits=1
make_predictions=c(354:503)

S<-GBlup_predict(G=G,y=y,h2=h2,make_predictions=c(354:503),ntraits=ntraits)
SR<-GBlup_predict(G=R,y=y,h2=h2,make_predictions=c(354:503),ntraits=ntraits)
S$rhoM

## [1] 0.5931802

SR$rhoM

## [1] 0.2669404
```

Examples with potato dataset:

```
setwd("/home/laura/Dropbox/DoctoradoCRAG/paper-1/article/software-help/reversinn1/pSBVB/toy_strawberry/Additional_functions/")
source("pedigree.R")
source("RelationshipMatrix.R")

## Completed! Time = 0.001266667  minutes

source("GBlupFunction.R")
source("generate_vcf_from_gen.R")
```

We also created a function to transformate genotypes into 'vcf' format. To do that, you need a dataset with genotypes (data varying between 0 and ploidy level) and a map file, containing the SNP's physical coordinates. We used genotypes from potato database (https://figshare.com/articles/Supplemental_Material_for_Enciso-Rodriguez_et_al_2018/6262214). Note that, our function generate the vcf file randomly from genotype (unphased data). In order to generate Linkage Disequilibrium, we could combine two pSBVB parameters: EXPAND_BASEPOP and INDFIRST, which simultaneosly exclude the initial individuals and generate a new set of founder.

Furthermore, the FilesGenerator.R file have all the functions needed to create a Chip, pedigree, QTNs location files and run potato example.

```
setwd("/home/laura/Dropbox/DoctoradoCRAG/paper-1/article/software-help/reversinn1/pSBVB/toy_potato/")
#read map file
map<-read.table("Sol_tet_mapa.map",sep="\t")
#read genotype
G<-read.delim("Pot_gen.gen",sep="\t")
#select a sub-sample from whole dataset
G=G[sample(c(1:nrow(G)),150),]
G=G[,sample(c(1:ncol(G)),500)]
G<-G[,colnames(G)%in%map[,1]]
dim(G)

## [1] 150 404
```

```r
#generate genotypes vcf format
A=GenotoVcf(G,p=4,map,path="NULL")
```

```
## Loading required package: data.table
```

```r
library(ggplot2)
setwd("/home/laura/Dropbox/DoctoradoCRAG/paper-1/article/software-help/reversinn1/pSBVB/toy_strawberry/Additional_functions/")
source("GBlupFunction.R")

data<-read.table("/home/laura/Dropbox/DoctoradoCRAG/paper-1/article/software-help/reversinn1/pSBVB/toy_potato/File_st.ped",header=FALSE)
#check the dimensions of dataset
#generate relationship matrix
dim(data)
```

```
## [1] 700   3
```

```r
A<-RelMatrix(data,dominance=FALSE,path="/home/laura/Dropbox/DoctoradoCRAG/paper-1/article/software-help/reversinn1/pSBVB/toy_potato/")
```

```
## Completed! Time = 0.00255  minutes
```

```r
setwd("/home/laura/Dropbox/DoctoradoCRAG/paper-1/article/software-help/reversinn1/pSBVB/toy_potato/results_potato/")

G<- read.table("Y.grm.1")
y<- read.table("Y.outy")
RelMatrix<-A[-c(1:150),-c(1:150)]



h2=0.5
ntraits=1

make_predictions=c(401:550)
library(ggplot2)
S<-GBlup_predict(G=G,y=y,h2=h2,make_predictions=make_predictions,ntraits=ntraits)
S$rho
```

```
## [1] 0.4194286
```

```r
Phenohat<-S$uhat[make_predictions] + S$mean
Pheno<-S$yout_corr[make_predictions]
cor(Pheno,Phenohat)
```

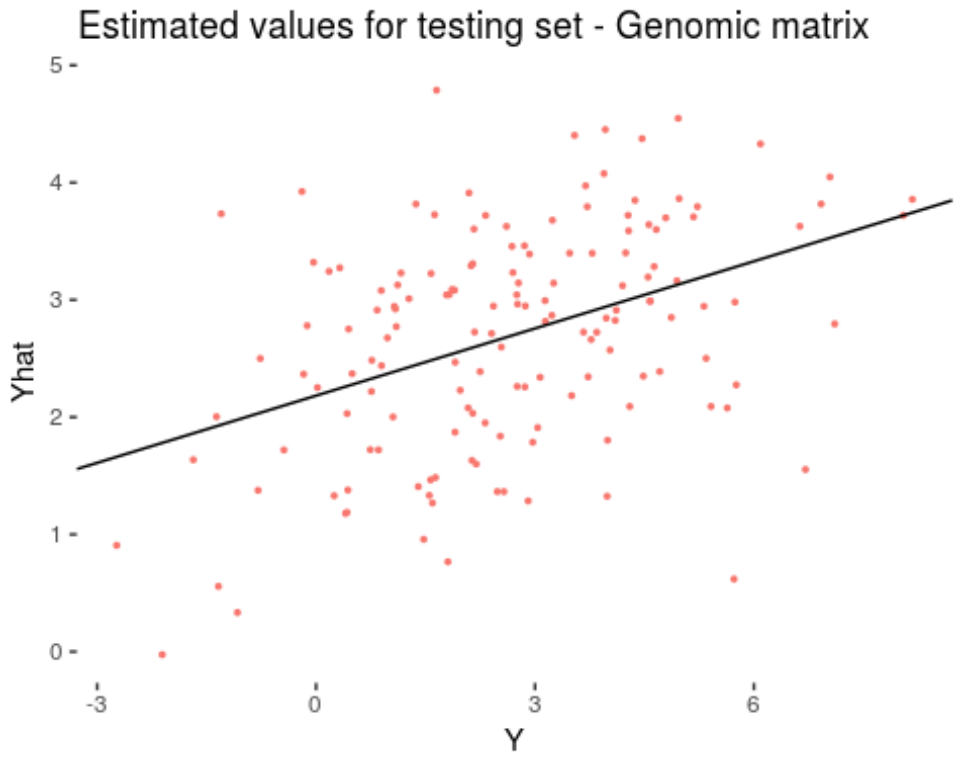
```
## [1] 0.4194286
```

```r
datos<-data.frame(Phenohat,Pheno)
colnames(datos)<-c("Yhat","Y")
apply(datos,2,class)
```

```
##      Yhat         Y
## "numeric" "numeric"
```

```
ggplot(datos, aes(y=Yhat, x=Y,colour="red")) +
  ggtitle("Estimated values for testing set - Genomic matrix")+
  geom_point(size=0.6) + scale_shape_manual(values=c(2,4)) +
  geom_abline(intercept=lm(Yhat ~ Y,data=datos)$coefficients[1], slope=lm(Yha
t ~ Y,data=datos)$coefficients[2])+
  #scale_x_continuous(limits = c(min(datos[,1]), max(datos[,1])))+
  #scale_y_continuous(limits = c(min(datos[,2]), max(datos[,2])))+

  theme(panel.background = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(), legend.position ="none")
```




```
make_predictions=c(401:550)


SR<-GBlup_predict(G=RelMatrix,y=y,h2=h2,make_predictions=c(401:550),ntraits=n
traits)
SR$rhoM
```

```
## [1] 0.3707248
```

```
Pheno<-SR$yout_corr[make_predictions]
datos_r<-data.frame(Phenohat,Pheno)
colnames(datos_r)<-c("Yhat","Y")
```

```r
#Pedigree Prediction
ggplot(datos_r, aes(y=Yhat, x=Y,colour="red")) +
  geom_point(size=0.6) + scale_shape_manual(values=c(2,4)) +
  geom_abline(intercept=lm(Yhat ~ Y,data=datos_r)$coefficients[1], slope=lm(Y
hat ~ Y,data=datos_r)$coefficients[2])+
   ggtitle("Estimated values for testing set - N Relationsip matrix")+
  theme(panel.background = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(), legend.position ="none")
```