Heritability and spatial distribution of mummies aphids in ears wheat

C.MELOT ; C.LE BOUAR ; L.RIZZI

20/06/2021

# Analysis

## Tools

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble 3.1.1 v dplyr 1.0.6  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.1  
## v purrr 0.3.4

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

## The following object is masked from 'package:purrr':  
##   
## compact

## Phenotypic data

Phenotypic data are loaded

setwd("C:/Users/YonaW/Desktop/2A/JRL/analyse R/script/") # modify according to the directory place  
  
A<-read.table("donnee/sortie2.csv", sep=";", header = T)  
  
  
  
# Link between passage and lines  
A$passageGD<-A$passage  
levels(A$GD)

## NULL

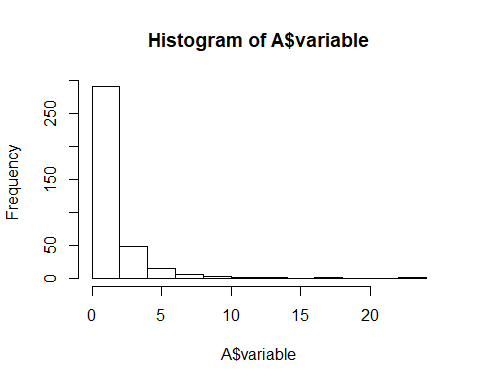
A$passageGD[A$GD=="D"]<-A$passageGD[A$GD=="D"]+0.5

## Data visualization

Vizualisation of the variable wanted. For the mummies aphids : “somme\_PP\_TOTAL” For terminal mummies aphids : “somme\_Nbpucerons\_vides\_momies”

For living mummies aphids: “somme\_Nbpucerons\_parasites”

A$variable<-A$somme\_PP\_TOTAL  
variable1<-"somme\_PP\_TOTAL"  
A$variable=as.numeric(A$variable)  
hist(A$variable)

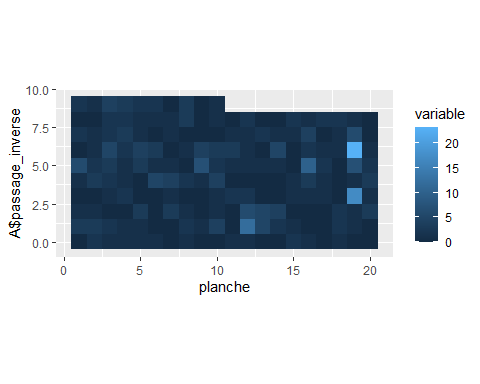


## Brut cartography

Here is the spatial distribution of the brut cartography.

A$passage\_inverse<-max(A$passageGD)-A$passageGD  
zp1 <- ggplot(A, aes(x =planche, y =A$passage\_inverse, fill = variable))  
zp1 <- zp1 + geom\_tile()  
zp1 <- zp1 + coord\_equal()  
zp1

## Warning: Use of `A$passage\_inverse` is discouraged. Use `passage\_inverse` instead.



## Spatial analysis

A$R <- as.factor(A$planche)  
A$C <- as.factor(A$passageGD)  
#genotype is fixed  
m0\_A\_F <- SpATS(response = variable1,   
 spatial = ~ SAP( passageGD, planche, nseg = c(10,20),   
 degree = 3, pord = 2),  
 genotype = "Genotype", genotype.as.random = FALSE,  
 random = ~ R + C,  
 data = A,  
 control = list(tolerance = 1e-03))

## Effective dimensions  
## -------------------------  
## It. Deviance R Cf(passageGD,planche)|passageGDf(passageGD,planche)|planche  
## 1 942.960107 1.003 0.852 10.378 14.072  
## 2 609.046311 2.600 0.412 6.935 10.379  
## 3 597.684278 5.336 0.231 4.418 7.252  
## 4 588.661677 8.291 0.151 2.651 4.510  
## 5 583.901093 10.250 0.114 1.591 2.430  
## 6 582.605766 11.058 0.091 1.093 1.369  
## 7 582.351912 11.286 0.075 0.866 0.970  
## 8 582.269831 11.345 0.061 0.732 0.778  
## 9 582.232012 11.357 0.051 0.640 0.666  
## 10 582.211025 11.356 0.042 0.571 0.593  
## 11 582.197988 11.352 0.035 0.517 0.543  
## 12 582.189217 11.348 0.029 0.473 0.506  
## 13 582.182938 11.344 0.024 0.436 0.479  
## 14 582.178204 11.342 0.020 0.405 0.458  
## 15 582.174468 11.339 0.017 0.377 0.443  
## 16 582.171398 11.337 0.014 0.353 0.431  
## 17 582.168781 11.335 0.012 0.331 0.422  
## 18 582.166473 11.333 0.010 0.311 0.415  
## 19 582.164377 11.332 0.008 0.292 0.411  
## 20 582.162424 11.330 0.007 0.275 0.407  
## 21 582.160565 11.328 0.006 0.258 0.406  
## 22 582.158762 11.327 0.005 0.243 0.405  
## 23 582.156988 11.325 0.004 0.228 0.405  
## 24 582.155223 11.323 0.004 0.213 0.406  
## 25 582.153452 11.322 0.003 0.200 0.407  
## 26 582.151665 11.320 0.003 0.186 0.409  
## 27 582.149854 11.319 0.002 0.173 0.412  
## 28 582.148014 11.317 0.002 0.160 0.415  
## 29 582.146144 11.316 0.002 0.147 0.418  
## 30 582.144244 11.314 0.001 0.135 0.421  
## 31 582.142319 11.312 0.001 0.124 0.425  
## 32 582.140372 11.311 0.001 0.112 0.428  
## 33 582.138412 11.310 0.001 0.101 0.432  
## 34 582.136450 11.308 0.001 0.091 0.436  
## 35 582.134497 11.307 0.001 0.081 0.439  
## 36 582.132568 11.306 0.001 0.071 0.443  
## 37 582.130677 11.304 0.001 0.063 0.446  
## 38 582.128840 11.303 0.000 0.055 0.449  
## 39 582.127074 11.302 0.000 0.047 0.452  
## 40 582.125394 11.302 0.000 0.040 0.455  
## 41 582.123815 11.301 0.000 0.034 0.458  
## 42 582.122351 11.300 0.000 0.029 0.461  
## 43 582.121012 11.299 0.000 0.024 0.463  
## 44 582.119810 11.299 0.000 0.020 0.465  
## 45 582.118750 11.299 0.000 0.016 0.467  
## 46 582.117838 11.298 0.000 0.013 0.469  
## Timings:  
## SpATS 3.96 seconds  
## All process 4.36 seconds

#genotype is random  
m0\_A\_R <- SpATS(response = variable1,   
 spatial = ~ SAP( passageGD, planche, nseg = c(10,20),   
 degree = 3, pord = 2),  
 genotype = "Genotype", genotype.as.random = TRUE,  
 random = ~ R + C,  
 data = A,  
 control = list(tolerance = 1e-03))

## Effective dimensions  
## -------------------------  
## It. Deviance Genotype R Cf(passageGD,planche)|passageGDf(passageGD,planche)|planche  
## 1 1399.781490 116.025 1.327 1.085 12.039 15.991  
## 2 1015.634786 90.078 3.340 0.540 8.481 11.979  
## 3 999.835889 78.476 6.028 0.276 5.635 8.278  
## 4 990.368620 72.376 8.352 0.155 3.698 5.200  
## 5 985.261135 69.061 9.742 0.096 2.573 2.933  
## 6 983.291864 67.094 10.333 0.064 2.083 1.613  
## 7 982.714496 65.706 10.473 0.043 1.980 1.032  
## 8 982.510050 64.640 10.471 0.028 1.976 0.759  
## 9 982.424816 63.809 10.438 0.018 1.974 0.612  
## 10 982.383890 63.161 10.405 0.011 1.965 0.522  
## 11 982.361974 62.658 10.377 0.007 1.953 0.461  
## 12 982.349424 62.269 10.356 0.004 1.941 0.419  
## 13 982.341927 61.967 10.340 0.003 1.932 0.387  
## 14 982.337313 61.734 10.328 0.002 1.924 0.363  
## 15 982.334404 61.553 10.319 0.001 1.918 0.345  
## 16 982.332530 61.413 10.312 0.001 1.913 0.329  
## 17 982.331299 61.304 10.307 0.000 1.908 0.317  
## 18 982.330475 61.219 10.302 0.000 1.905 0.307  
## Timings:  
## SpATS 1.74 seconds  
## All process 1.92 seconds

# Brief summary  
m0\_A\_F

##   
## Spatial analysis of trials with splines   
##   
## Response: somme\_PP\_TOTAL  
## Genotypes (as fixed): Genotype   
## Spatial: ~SAP(passageGD, planche, nseg = c(10, 20), degree = 3, pord = 2)  
## Random: ~R + C   
##   
##   
## Number of observations: 368  
## Number of missing data: 0  
## Effective dimension: 198.78  
## Deviance: 582.118

m0\_A\_R

##   
## Spatial analysis of trials with splines   
##   
## Response: somme\_PP\_TOTAL  
## Genotypes (as random): Genotype   
## Spatial: ~SAP(passageGD, planche, nseg = c(10, 20), degree = 3, pord = 2)  
## Random: ~R + C   
##   
##   
## Number of observations: 368  
## Number of missing data: 0  
## Effective dimension: 77.73  
## Deviance: 982.330

# More information: dimensions  
#summary(m0\_A\_F, which = "dimensions")   
#summary(m0\_A\_R, which = "dimensions")   
  
# More information: variances  
#summary(m0\_A\_F, which = "variances")  
#summary(m0\_A\_R, which = "variances")  
  
# More information: all  
#summary(m0\_A\_F, which = "all")  
#summary(m0\_A\_R, which = "all")  
  
# Plot results  
plot(m0\_A\_F)

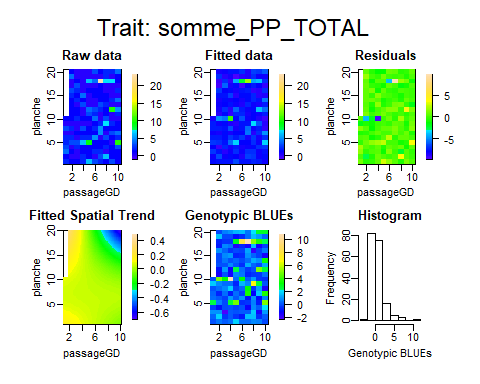
## Warning in matrix(df$ONE, ncol = length(columns), nrow = length(rows)): la  
## longueur des données [378] n'est pas un diviseur ni un multiple du nombre de  
## lignes [20]

## Warning in matrix(df$response, ncol = length(columns), nrow = length(rows)): la  
## longueur des données [378] n'est pas un diviseur ni un multiple du nombre de  
## lignes [20]

## Warning in matrix(df$fitted, ncol = length(columns), nrow = length(rows)): la  
## longueur des données [378] n'est pas un diviseur ni un multiple du nombre de  
## lignes [20]

## Warning in matrix(df$residuals, ncol = length(columns), nrow = length(rows)):  
## la longueur des données [378] n'est pas un diviseur ni un multiple du nombre de  
## lignes [20]

## Warning in matrix(df$geno.pred, ncol = length(columns), nrow = length(rows)):  
## la longueur des données [378] n'est pas un diviseur ni un multiple du nombre de  
## lignes [20]



plot(m0\_A\_R)

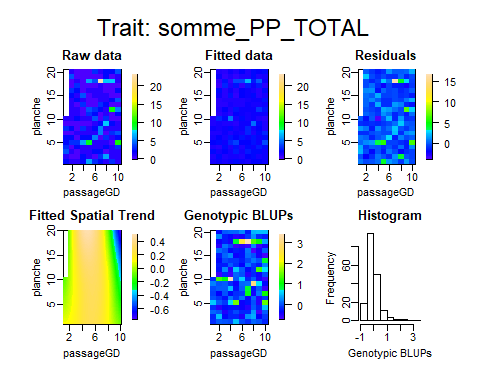
## Warning in matrix(df$ONE, ncol = length(columns), nrow = length(rows)): la  
## longueur des données [378] n'est pas un diviseur ni un multiple du nombre de  
## lignes [20]

## Warning in matrix(df$response, ncol = length(columns), nrow = length(rows)): la  
## longueur des données [378] n'est pas un diviseur ni un multiple du nombre de  
## lignes [20]

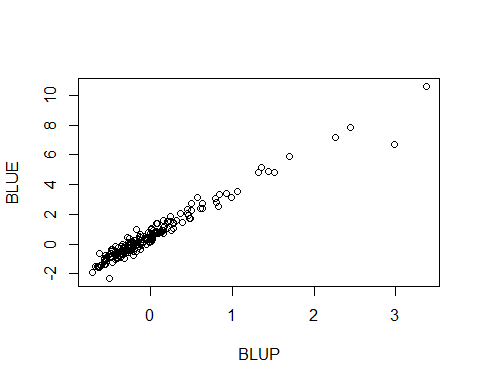
## Warning in matrix(df$fitted, ncol = length(columns), nrow = length(rows)): la  
## longueur des données [378] n'est pas un diviseur ni un multiple du nombre de  
## lignes [20]

## Warning in matrix(df$residuals, ncol = length(columns), nrow = length(rows)):  
## la longueur des données [378] n'est pas un diviseur ni un multiple du nombre de  
## lignes [20]

## Warning in matrix(df$geno.pred, ncol = length(columns), nrow = length(rows)):  
## la longueur des données [378] n'est pas un diviseur ni un multiple du nombre de  
## lignes [20]



# Compute the variogram\*  
#var.m0\_A\_F <- variogram(m0\_A\_F)  
  
# Plot the variogram  
#plot(var.m0\_A\_F)  
  
#creation of BLUE and BLUP vector  
BLUE<-m0\_A\_F$coeff[1:184]  
BLUP<-m0\_A\_R$coeff[1:184]  
  
BLUE[which(names(BLUE)=="Intercept")]<-0  
names(BLUE)[which(names(BLUE)=="Intercept")]<-names(BLUP)[1]  
BLUE<-BLUE[order(names(BLUE))]  
  
plot(BLUP, BLUE)



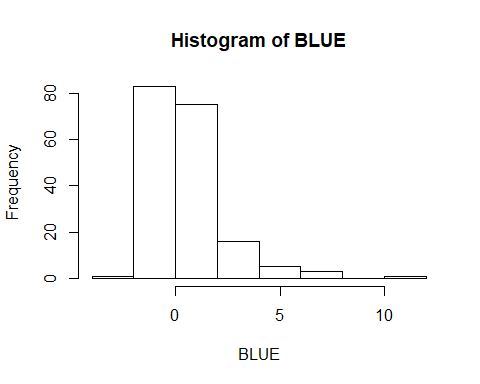
which(BLUP>0.8)

## ELAX\_130 ELAX\_198 ELAX\_227 ELAX\_237 ELAX\_288 ELAX\_303 ELAX\_427 ELAX\_428   
## 12 32 39 44 61 67 116 117   
## ELAX\_435 ELAX\_451 ELAX\_460 ELAX\_49 ELAX\_68 ELAX\_94 GQAX\_76   
## 118 123 126 142 148 159 178

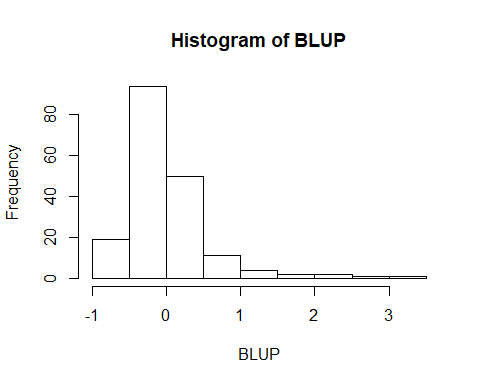
BLUE[118]

## ELAX\_435   
## 6.672005

hist(BLUE)



hist(BLUP)



BLUE and BLUPS data are saved in a csv file.

donnee<-data.frame(blue=BLUE, blup=BLUP)  
  
write.table(donnee, "BlueBlupS2.csv", sep=";",col.names = TRUE, row.names = TRUE)

## Heritability

Heritability with and without correction are estimated

### Without spatial correction

Individual heritability (for each genotype)

library(lme4)

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':  
##   
## expand, pack, unpack

mnaif<-lmer(variable ~ rep +(1|Genotype), data=A)  
VG<-as.data.frame(VarCorr(mnaif))$sdcor[1]\*\*2  
VE<-as.data.frame(VarCorr(mnaif))$sdcor[2]\*\*2  
#mnaif  
VG/(VG+VE)#heritability

## [1] 0.17761

VG #genotypic variance

## [1] 0.9956764

VE #environmental variance

## [1] 4.610294

Design heritability (mean of the two genotype)

VG/(VG+VE/2)

## [1] 0.3016449

### With spatial correction

summary(m0\_A\_R, which = "variances")

##   
## Spatial analysis of trials with splines   
##   
## Response: somme\_PP\_TOTAL  
## Genotypes (as random): Genotype   
## Spatial: ~SAP(passageGD, planche, nseg = c(10, 20), degree = 3, pord = 2)  
## Random: ~R + C   
##   
##   
## Number of observations: 368  
## Number of missing data: 0  
## Effective dimension: 77.73  
## Deviance: 982.330  
##   
## Variance components:  
## Variance SD log10(lambda)  
## Genotype 1.097e+00 1.047e+00 0.57723  
## R 3.672e-01 6.060e-01 1.05241  
## C 4.818e-06 2.195e-03 5.93450  
## f(passageGD,planche)|passageGD 2.992e-01 5.470e-01 1.14134  
## f(passageGD,planche)|planche 1.007e-03 3.173e-02 3.61431  
##   
## Residual 4.143e+00 2.035e+00

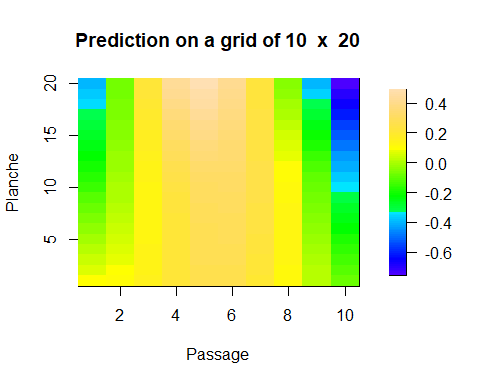
getHeritability(m0\_A\_R)

## Genotype   
## 0.33

# Fit spatial covariable

Creation of a covariable to see links between number of mummies independantly of their genotype. The function obtain.spatialtrend is utilised.

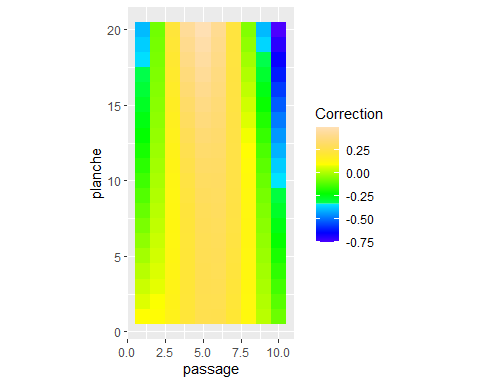
# a value for each plot is created  
nrowA<-nlevels(as.factor(A$passageGD))  
ncolA<-nlevels(as.factor(A$planche))  
  
spat.trend.2 <- obtain.spatialtrend(m0\_A\_R, grid = c(nrowA, ncolA))  
  
colors = topo.colors(100)  
  
fields::image.plot(spat.trend.2$col.p, spat.trend.2$row.p, t(spat.trend.2$fit),  
 main = paste("Prediction on a grid of", nrowA, " x ", ncolA),  
 col = colors, xlab = "Passage", ylab = "Planche")



#results are saved in a table  
ligne<-NA  
colonne<-NA  
correction<-NA  
  
for ( i in 1:length(spat.trend.2$col.p)) {  
 for (j in 1:length(spat.trend.2$row.p)) {  
 ligne<-append(ligne,spat.trend.2$col.p[i])  
 colonne<-append(colonne,spat.trend.2$row.p[j])  
 correction<-append(correction, t(spat.trend.2$fit)[i,j])  
 }  
}  
  
ACor<-data.frame(passage=ligne, planche=colonne, Correction=correction)   
ACor<-ACor[-1,]  
ACor[1:10,]

## passage planche Correction  
## 2 1 1 0.09017982  
## 3 1 2 0.06475577  
## 4 1 3 0.03916485  
## 5 1 4 0.01326214  
## 6 1 5 -0.01294175  
## 7 1 6 -0.03931903  
## 8 1 7 -0.06577411  
## 9 1 8 -0.09225720  
## 10 1 9 -0.11872376  
## 11 1 10 -0.14508787

zp1 <- ggplot(ACor, aes(x =passage, y =planche, fill = Correction))  
zp1 <- zp1 + geom\_tile()  
zp1<- zp1 + scale\_fill\_gradientn(colors=colors)  
zp1 <- zp1 + coord\_equal()  
zp1



File is saved in a csv file.

write.table(ACor, "Covariable.csv", sep=";",col.names = TRUE, row.names = FALSE)