

**WORKSHOP
DE BIOINFORMÁTICA
APLICADA À GENÔMICA E
MELHORAMENTO ANIMAL**



AULA PRÁTICA 2

RR-BLUP e G-BLUP



Fabyano Fonseca e Silva

**Prof. Adjunto IV - Dep Zootecnia – UFV
Estatística Genômica e Bioinformática**

Campo Grande, 14/07 a 15/07 de 2014

#Método RR-BLUP: pacote rrBLUP do R

```
gen=read.table("gs1_gen.txt",h=T) #lendo arquivo genotipos
dim(gen)
head(gen)

freq=read.table("gs1_alefreq.txt",h=T) #lendo arquivo freq alel
dim(freq)
head(freq)

fen=read.table("gs1_fen.txt",h=T) #lendo arquivo fenotipos
dim(fen)
head(fen)

#corrigindo fenótipo para efeitos fixos - de los Campos et al. (2013)

y1=mean(fen$y) + lm(y~ factor(ym_slg) + factor(farm) + hcw, data=fen)$residuals

library(rrBLUP)
M=as.matrix(gen) # matriz de marcadores
dim(M)
head(M)

rrblup = mixed.solve(y1, Z=M, SE=TRUE)
va=rrblup$Vu #variância genética aditiva
sig2u=sum(freq$pq)*va #variância genética aditiva
sig2e=rrblup$Ve #Variância residual
h2_r=sig2u/(sig2u+sig2e) #herdabilidade
```

estimativas para os efeitos SNPs

```
a_hat_r=rrblup$u #vetor de efeitos estimados SNPs
snp_rrblup=cbind(colnames(M),a_hat_r)
colnames(snp_rrblup)=c("SNP","a_rrblup")
write.table(snp_rrblup,"snp_rrblup.txt", row.names=FALSE,quote=FALSE)
plot(a_hat_r) #gráfico dos efeitos estimados SNPs
```

Valores genéticos genômicos - GEBVs

```
u_hat_r=M%*%a_hat_r
rownames(u_hat_r)=fen$id
plot(hist(u_hat_r)) #distribuição dos GEBVs
top10=quantile(u_hat_r[,1], probs = c(0.9)) #top10%
top10
# individuos selecionados e respectivos GEBVs
sel_id=data.frame(u_hat_r[,1][u_hat_r[,1]>=top10])
colnames(sel_id)=c("GEBV")
```

```
library(gap) #pacote que contem o grafico Manhattan
```

```
map=read.table("mapa_gs1.txt", h=T)
```

```
snp_rrblup=read.table("snp_rrblup.txt",h=T)
```

```
#interseção de arquivos (SNPs com efeitos estimados e mapeados)
```

```
snp_rrblup1=merge(map,snp_rrblup,by=intersect("SNP","SNP"))
```

```
snp_rrblup2=snp_rrblup1[with(snp_rrblup1, order(chr,pos)),]
```

```
#ordenar chr e posicao
```

```
par(las=2, xpd=TRUE, cex.axis=0.9, cex=0.8)
```

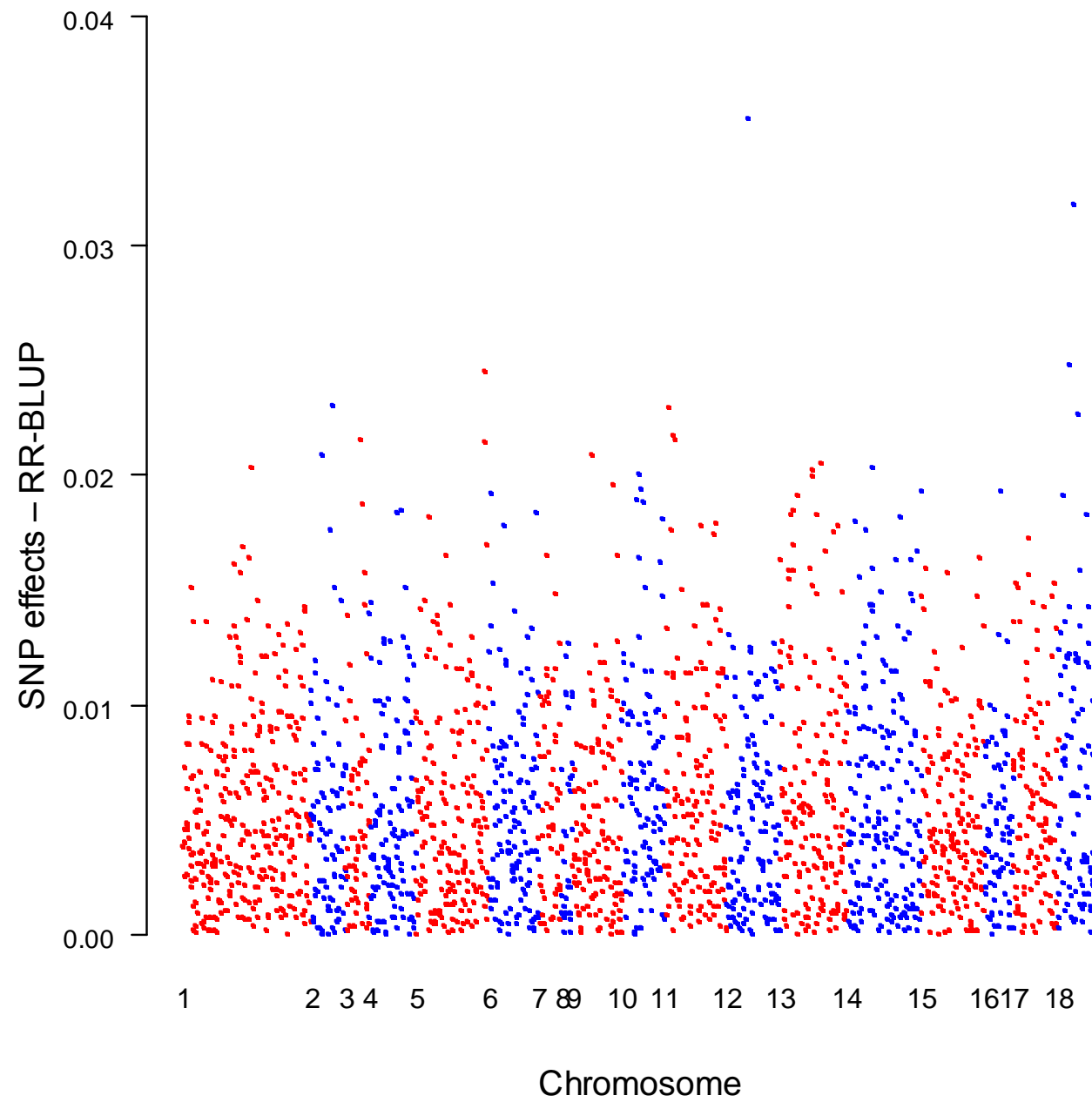
```
color = cbind(rep(c("red","blue"),9),c("blue"))
```

```
#funcao mhplot
```

```
mhtplot(abs(snp_rrblup2[,-1]),mht.control(logscale=FALSE,
```

```
colors=color,labels=cbind(seq(1,18),c("x")),srt=0), ylab="SNP  
effects - RR-BLUP",pch=19)
```

```
axis(2, at = seq(0, 0.04, by = 0.01))
```



```
#####ajustando GBLUP#####
library(rrBLUP)
gblup= mixed.solve(y1, K=A.mat(M),SE=T, method="REML",
bounds=c((sig2e/sig2u)-0.01, (sig2e/sig2u)+0.01)) #facilitar convergencia

#OBS. bounds é uma estratégia para evitar problemas de convergência,
#assume um limite inf. e sup. para o param. shrinkage (lambda=sig2e/sig2u
#do rrblup)

sig2u_g=gblup$Vu #variância genética aditiva
sig2e_g=gblup$Ve #Variância residual
h2_g=sig2u_g/(sig2u_g+sig2e_g) #herdabilidade
u_hat_g=gblup$u #vetor de GEBV estimados
cor(u_hat_r,u_hat_g) #cor entre GEBV rrblup e gblup
library(MASS)
a_hat_g=ginv(t(M)%*%M)%*%t(M)%*%u_hat_g #vetor de efeitos estimados SNPs
cor(a_hat_r,a_hat_g) #cor entre Efeitos SNPs rrblup e gblup
#acurácia dos valores genéticos genômicos
r=sqrt(1-((gblup$u.SE)^2/sig2u_g))
```

OBS. calculando acurácia por meio de SE para EBV – Manual ASREML3

Note that the error presented for the estimate of a random effect is the square root of the prediction error variance. In a genetic context for example where a relationship matrix A is involved, the accuracy is $\sqrt{1 - \frac{s_i^2}{(1+f_i)\sigma_A^2}}$ where s_i is the standard error reported with the BLUP (u_i) for the i th individual, f_i is the

Método RR-BLUP: populações treinamento e validação

```
gen=read.table("gs1_gen.txt",h=T) #lendo arquivo genotipos
dim(gen)
head(gen)

freq=read.table("gs1_alefreq.txt",h=T) #lendo arquivo freq alel
dim(freq)
head(freq)

fen=read.table("gs1_fen.txt",h=T) #lendo arquivo fenotipos
dim(fen)
head(fen)

#corrigindo fenótipo para efeitos fixos

y1=mean(fen$y) + lm(y~ factor(ym_slg) + factor(farm) + hcw, data=fen)$residuals

#####ajustando RR-BLUP#####

library(rrBLUP)

M=as.matrix(gen) # matriz de marcadores

M1=M[1:500,]
M2=M[501:634,]

y11=y1[1:500]
y12=y1[501:634]

rrblup = mixed.solve(y11, Z=M1)

gebv2_r=M2%*%rrblup$u #GEBV para indivíduos da pop 2
cor(gebv2_r,y12) #acurácia dada pela cor obs e pred
```