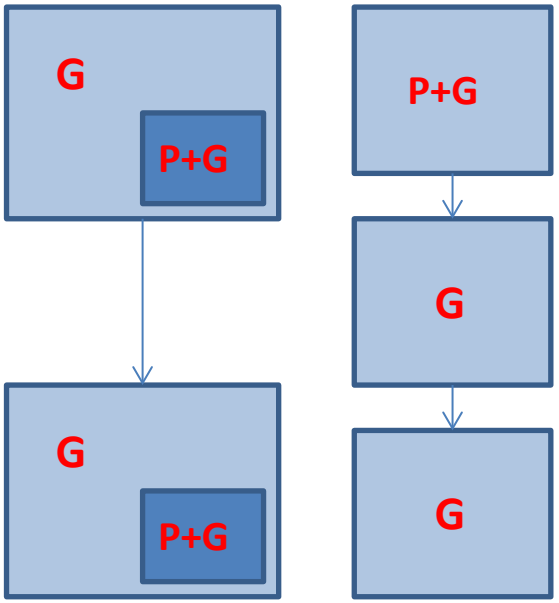
Basic concepts on GS

$$\Delta G = i\, r\, \sigma A / L$$

breeder → *i, r, L*

trade-offs → *r* <> *L*

maximize *r/L* [*i*], integrate (more) precise information more rapidly → GWE



GS addresses 3 of 4 components of genetic gain:

- **generation interval *L* : early evaluation**
- **selection intensity *i* : evaluation/costs**
- **accuracy *r* : information integration**

Basic concepts on GS

$$y = \mu + \sum x_{ij}\beta_j + \epsilon$$

$$u\ (BLUP) = X\beta$$

$$u \sim N(0, G\sigma^2{}_u)$$

$$G = XX'/2\sum pq$$

	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13
1	1	1	1	1	1	1	1	0	0	2	1	0
2	0	1	0	1	1	1	0	2	0	0	0	1
3	1	1	0	1	2	0	0	1	0	1	1	1
4	2	1	1	1	1	1	1	0	0	2	1	0
5	1	1	1	1	0	2	1	1	0	1	0	0
6	0	1	0	1	1	1	0	2	0	0	0	1
7	2	1	1	1	1	1	1	0	0	2	1	0
8	0	1	0	1	1	1	0	2	0	0	0	1
9	2	1	1	1	1	1	1	0	0	2	1	0
10	1	1	1	1	0	2	1	1	0	1	0	0
11	2	1	1	1	0	2	1	1	0	1	0	0
12	1	1	1	1	1	1	0	1	0	1	1	1
13	2	1	1	1	1	1	1	0	0	2	1	0
14	0	1	0	1	0	2	1	1	0	1	0	0
15	1	1	0	1	2	0	0	1	0	1	0	1

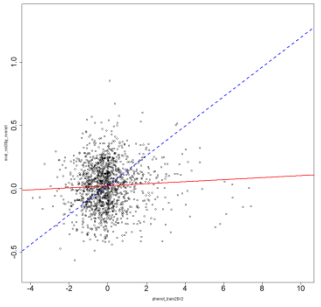
calculates G matrix

```
# calculates frequency of favourable allele per marker
Pi <- apply(X,2,sum)/(2*num_rec)
mat_Pi <- matrix(rep(Pi,num_rec),ncol=n_SNP,byrow=T)
W <- matrix(0,nrow=num_rec,ncol=n_SNP)
W <- X - (2*mat_Pi)
het <- 2*sum(Pi*(1-Pi))
G <- W%*%t(W) / het
# inverse is not needed
G_inv <- solve(G+diag(num_rec)*0.01)
```

Task scientific content: prediction accuracy

What is prediction accuracy?

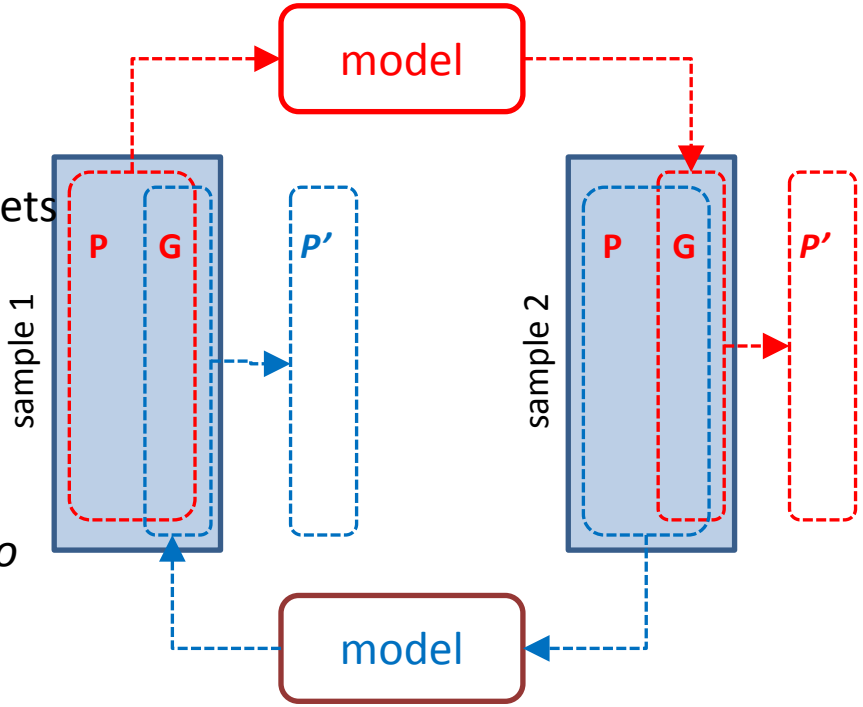
- most common metric to assess prediction accuracy is the correlation between estimated and true breeding values (or proxy)
- cross-validation



What affects prediction accuracy?

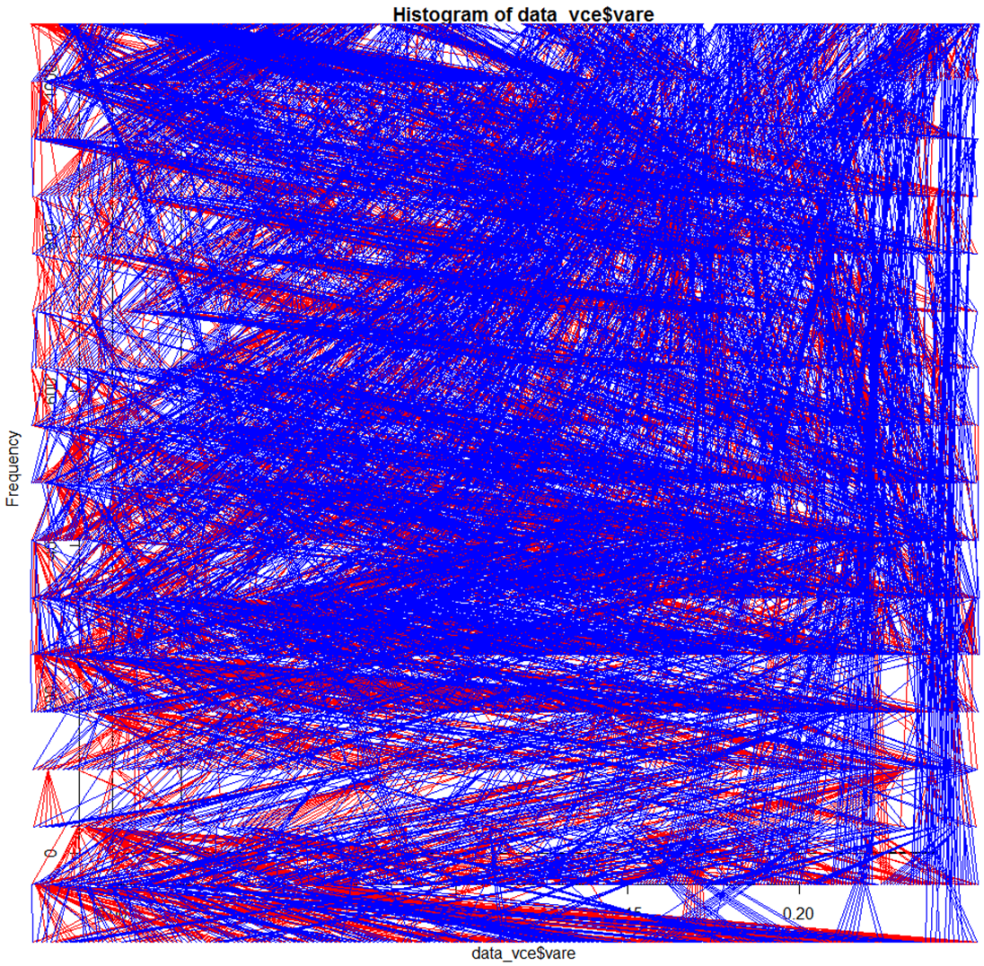
- relationships between training & prediction sets
- size of training & prediction sets
- heritabilities (& correlations when multiple traits)

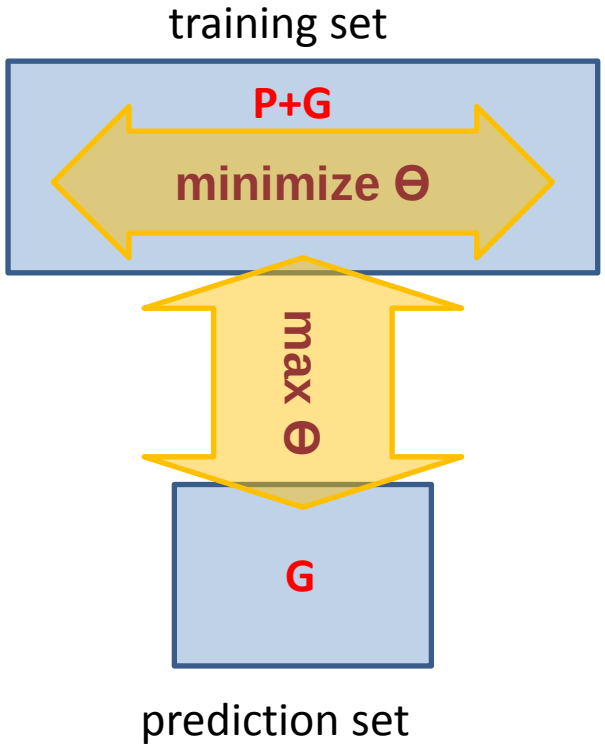
- marker density (when low)
- statistical model (clear with simulations, no *so* clear with real data)
- level of LD



GS evaluation with real data:

prediction accuracy *versus* training/prediction set sizes





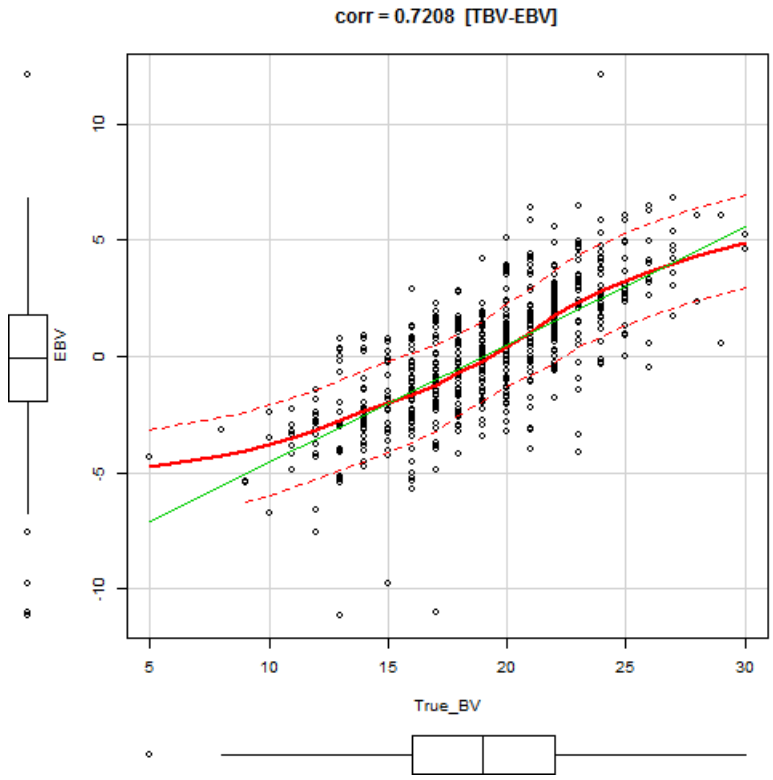
fonction drawPedigree [pedantics]

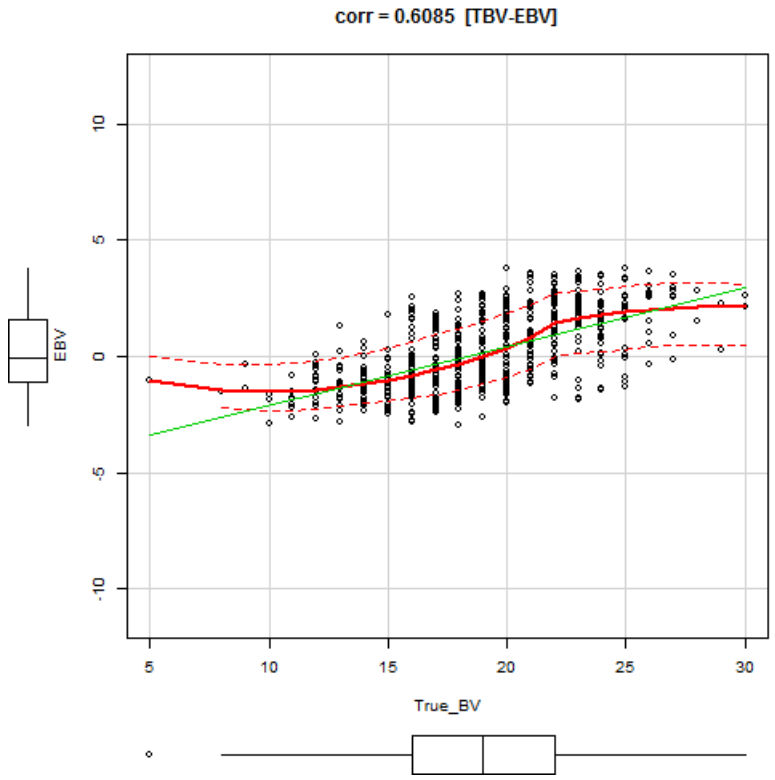
metagene_gs.R

Source on Save | Run | Source

```r
library(breedR)
library(pedigree)
library(pedigreemm)
library(pedantics)
library(car)
library(hexbin)

# import pedigree file already including founders
pheno_ped <- read.table("pheno_ped_case2.txt", header = T)
names(pheno_ped) <- c("self", "dad", "mum", "gen", "BV_X", "phe_X")
pheno_ped4 <- subset(pheno_ped, gen==4)
pedig <- build_pedigree(c('self', 'dad', 'mum'), data = pheno_ped)
num_tot <- as.numeric(dim(pheno_ped)[1])
num_rec <- as.numeric(dim(pheno_ped4)[1])
founders <- num_tot-num_rec

# check variables in data.frame
summary(pheno_ped)

# plot distributions for traits
colores <- rainbow(5)
plot(density(pheno_ped4$phe_X), col=colores[1], main="phe_X")

# run genetic model for each trait
pedmod_phe_X <- remlf90(fixed  = phe_X ~ 1,
                        genetic = list(model    = 'add_animal',
                                       pedigree = pedig,
```

1:1 (Top Level) R Script


corr = 0.7208 [TBV-EBV]
EBV
True_BV
10
5
0
-5
-10
5
10
15
20
25
30


corr = 0.6085 [TBV-EBV]
EBV
True_BV
10
5
0
-5
-10
5
10
15
20
25
30