Practical 3 Best Linear Unbiased Prediction (BLUP)

Winter School 2022 Module 4 Quantitative Genetics

2022-06-23

In this practical you will perform genomic prediction in a small data set using two equivalent BLUP models in R. We also provide code to run BLUP using GCTA on the larger simulated data set at the end. Feel free to give a try if you have time.

Data

The data set consists of a reference population of 325 bulls each has been genotyped for 10 SNPs. The trait was simulated in a way that the first SNP has an effect size of 2 and the 5th SNP has an effect size of 1 on the phenotype. The trait heritability is 0.1. There are a set of 31 calves who are selection candidates for this trait. Your task is to predict GEBV for these 31 selection candidates.

Let us start with reading in the data.

```
nmarkers <- 10      #number of markers
nrecords <- 325     #number of records

# data for reference population
X <- matrix(scan("/data/module4/prac3/xmat.inp"), ncol = nmarkers, byrow = TRUE)
y <- matrix(scan("/data/module4/prac3/yvec.inp"), byrow = TRUE)

# data for selection candidates
Xprog <- matrix(scan("/data/module4/prac3/xmat_prog.inp"), ncol = nmarkers, byrow = TRUE)
yprog <- matrix(scan("/data/module4/prac3/yvec_prog.inp"), ncol = 1, byrow = TRUE)
```

SNP-BLUP

In SNP-BLUP, we will need to predict the effects of the 10 SNPs in the reference population, using the equations:

$$\begin{bmatrix} 1_n'1_n & 1_n'X \\ X'1_n & X'X + I\lambda \end{bmatrix} \begin{bmatrix} \hat{\mu} \\ \hat{\alpha} \end{bmatrix} = \begin{bmatrix} 1_n'y \\ X'y \end{bmatrix}$$

where α are the SNP effects, 1_n is a vector of ones (325×1), X is a design matrix allocating SNP genotype to records, μ is the overall mean. We will use R to solve these equations.

First, we specify the value of lambda as $(1 - h^2)/(h^2/m) = 10$ where $m = 10$ is the number of SNPs. Then, we build the mixed model equations.

```
lambda   <- 10      # value for lambda

ones <- array(1, c(nrecords)) # a vector of ones
I <-diag(nmarkers)    # an identity matrix with dimention of m by m
```

```
# build coefficient matrix by blocks
coeff <- array(0, c(nmarkers + 1, nmarkers + 1))
coeff[1:1, 1:1] <- t(ones) %*% ones
coeff[1:1, 2:(nmarkers+1)] <- t(ones) %*% X
coeff[2: 2:(nmarkers+1), 1] <- t(X) %*% ones
coeff[2:(nmarkers+1), 2:(nmarkers+1)] <- t(X) %*% X + I * lambda

# build the right hand side
rhs = rbind(t(ones) %*% y, t(X) %*% y)

# get BLUP solution
solution_vec <- solve(coeff, rhs)
```

Question 1: Which SNP has the largest effect? Which SNP has the second largest effect? Is it consistent with the simulation setting? (hint: what's the LD correlation between SNP 2 and 3?)

```
print(solution_vec)
```

```
##              [,1]
##  [1,]  1.05907285
##  [2,]  1.62688323
##  [3,]  0.22899575
##  [4,]  0.22899575
##  [5,]  0.07916017
##  [6,] -0.04453153
##  [7,]  0.06071536
##  [8,]  0.29051596
##  [9,] -0.07916017
## [10,]  0.24355241
## [11,] -0.07916017
```

Question 2: The genotypes for the selection candidates are stored in `Xprog`. Can you write a small R script to calculate the GEBV?

```
GEBV = Xprog %*% solution_vec[-1]  # the first element is the intercept estimate
```

Question 3: Fours years later, all the selection candidates receive a phenotypic record from a progeny test. The results are in `yprog`. What is the correlation between your GEBV and the progeny test result?

```
cor(GEBV, yprog)
```

```
##           [,1]
## [1,] 0.7105889
```

GBLUP

We will analyse the same data using the GBLUP (BLUP based on the genomic relationship matrix) approach. The mixed model is

$$y = 1_n\mu + Zg + e$$

where terms are as above, and Z is a design matrix allocating records to individuals, and g is a vector of (genomic) breeding values. The g are random effects, assumed to be distributed $\mathcal{N}(0, G\sigma_g^2)$, where G is the genomic relationship matrix (GRM).

The solutions to the mixed model equations are

$$\begin{bmatrix}\hat{\mu}\\ \hat{g}\end{bmatrix} = \begin{bmatrix}1_n'1_n & 1_n'Z \\ Z'1_n & Z'Z + G^{-1}\frac{\sigma_e^2}{\sigma_g^2}\end{bmatrix}^{-1} \begin{bmatrix}1_n'y \\ Z'y\end{bmatrix}$$

The first step in building the mixed model equations is to build G.

G is constructed as

$$G = WW'/m$$

where m is the number of markers, $w_{ij} = (x_{ij} - 2p_j)/\sqrt{2p_j(1-p_j)}$ and p_j is the allele frequency of the 2nd allele for SNP j. Remember in the GBLUP, G (and Z) have to include all individuals, including the individuals with no phenotypes of their own that we wish to predict. So the X matrix has to include all individuals, including the progeny.

You can create this new X from the original X matrices in the first practical using the rbind (join by row command in R):

```
Xall <- rbind(X, Xprog)
nanims = nrow(Xall)
```

We next calculate GRM and its inverse.

```
# calculate allele frequency
p = colMeans(Xall)/2

# obtain standardised genotypes
W = matrix(0, nrow = nanims, ncol = nmarkers)
for(j in 1:nmarkers)
    W[,j] <- (Xall[,j] - 2*p[j])/sqrt(2*p[j]*(1-p[j]))

# compute GRM
G = W%*%t(W)/nmarkers
# The next line adds a small amount to the diagonal of G,
# otherwise G is not invertable in this small example!
G <- G + diag(nanims)*0.01

# compute the inverse of GRM
Ginv <- solve(G)
```

The only other matrix we do not have is Z, which is has dimensions nrecords $\times$ nanims. Z is a diagonal matrix for those animals with records, and a block of zeros for those animals without records (as there are no records to allocate these animals to).

Z can be constructed as

```
Z1 <-diag(nrecords)
Z2 <-matrix(0, 325, 31)
Z <- cbind(Z1, Z2)
```

Now go ahead and build the mixed model solution equations above. For $\frac{\sigma_e^2}{\sigma_g^2}$ use a value of 1.

```
# coeff
coeff <- array(0, c(nanims + 1, nanims + 1))
coeff[1:1, 1:1] <- t(ones) %*% ones
coeff[1:1, 2:(nanims+1)] <- t(ones) %*% Z
```

```
coeff[2: 2:(nanims+1), 1] <- t(Z) %*% ones
coeff[2:(nanims+1), 2:(nanims+1)] <- t(Z) %*% Z + Ginv

# right hand side
rhs = rbind(t(ones) %*% y, t(Z) %*% y)

# BLUP solution for the breeding value of each individual
gblup <- solve(coeff, rhs)
```

Question 4: What is the accuracy of the genomic predictions for the 31 selection candidates from the GBLUP, eg $r(\hat{g}, \text{yprog})$? How does this compare with the accuracy of SNP-BLUP?

```
# the genomic prediction for the 31 selection candidates is
yprog_pred = gblup[-c(1:326)]

# the accuracy is
cor(yprog, yprog_pred)
```

```
##           [,1]
## [1,] 0.7205602
```

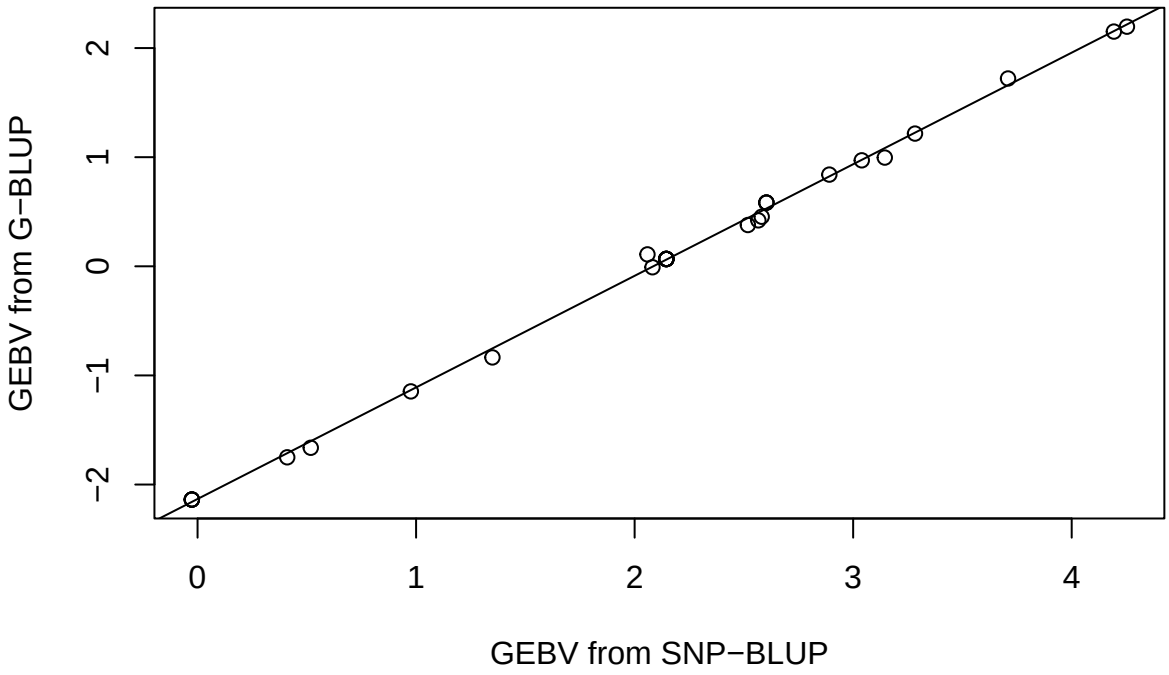
Comparison between SNP-BLUP and GBLUP

Question 5: If you plot GEBV for the SNP-BLUP prac against $\hat{g}$, do you get a regression line with a slope of 1, indicating these are equivalent models? Why, or why not (hint, did we centre and standardise the X matrix to the W matrix in SNP-BLUP)? If you use the same genotype matrix (W) in both SNP-BLUP and GBLUP are the genomic predictions identical?

```
#png("snpblup_vs_gblup.png")
plot(GEBV, yprog_pred, xlab = "GEBV from SNP-BLUP", ylab = "GEBV from G-BLUP")
lm(yprog_pred ~ GEBV)
```

```
##
## Call:
## lm(formula = yprog_pred ~ GEBV)
##
## Coefficients:
## (Intercept)         GEBV
##      -2.132        1.023
```

```
abline(lm(yprog_pred ~ GEBV))
```




```
#dev.off()
```

SNP-BLUP with standardised genotype matrix.

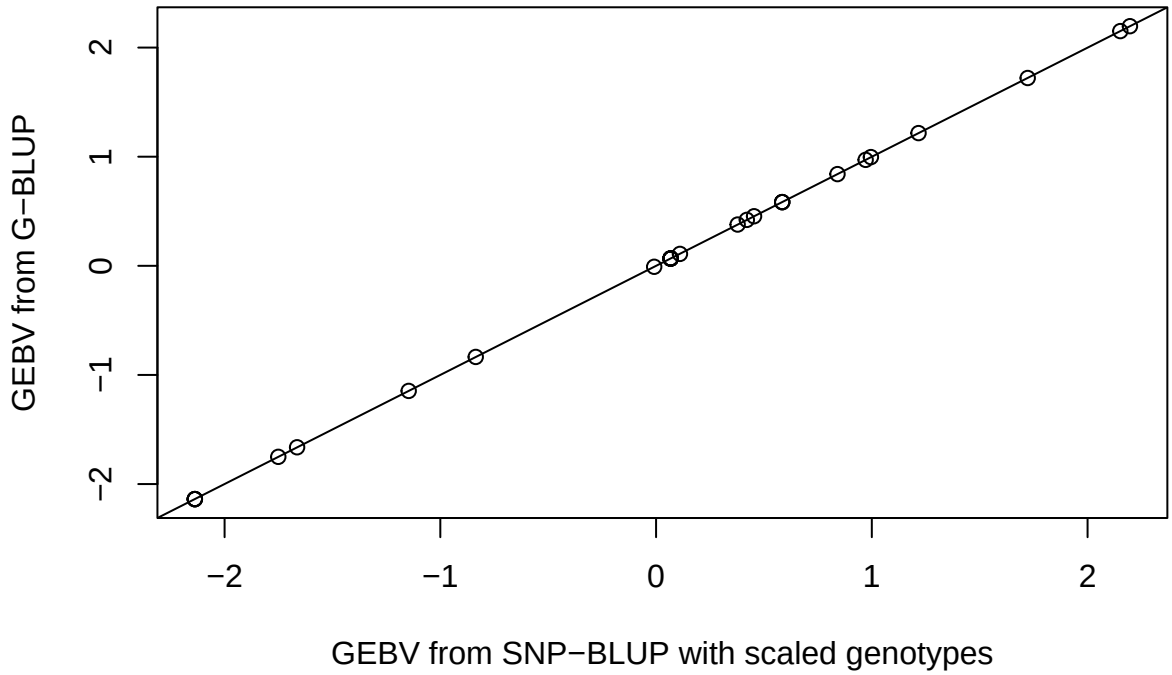
```
X = W[1:325,]
coeff <- array(0, c(nmarkers + 1, nmarkers + 1))
coeff[1:1, 1:1] <- t(ones) %*% ones
coeff[1:1, 2:(nmarkers+1)] <- t(ones) %*% X
coeff[2: 2:(nmarkers+1), 1] <- t(X) %*% ones
coeff[2:(nmarkers+1), 2:(nmarkers+1)] <- t(X) %*% X + lambda * I
rhs = rbind(t(ones) %*% y, t(X) %*% y)
solution_vec <- solve(coeff, rhs)
Wprog = W[-c(1:325),] #only W for the progeny
GEBV_W = Wprog %*% solution_vec[-1]
cor(yprog, GEBV_W)
```

```
##           [,1]
## [1,] 0.7206472
```

```
#png("snpblup_scaled_vs_gblup.png")
plot(GEBV_W, yprog_pred, xlab = "GEBV from SNP-BLUP with scaled genotypes", ylab = "GEBV from G-BLUP")
lm(yprog_pred ~ GEBV_W)
```

```
##
## Call:
## lm(formula = yprog_pred ~ GEBV_W)
##
## Coefficients:
## (Intercept)       GEBV_W
##  -6.829e-05    9.998e-01
```

```
abline(lm(yprog_pred ~ GEBV_W))
```




```
#dev.off()
```

BLUP using GCTA

You can use GCTA (https://yanglab.westlake.edu.cn/software/gcta/#BLUP) to run BLUP, which is more computationally efficient than R. We demonstrate GCTA-BLUP using same data in Practical 1. To save time, you can directly **run from Step 3**.

Step 1: build the genomic relationship matrix (GRM).

```
## code to generate GRM
bfile="/data/module4/prac3/gwas"
gcta --bfile $bfile --make-grm --threads 4 --out gwas
```

Step 2: run GBLUP

```
grm="/data/module4/prac3/gwas"
pheno="/data/module4/prac3/simu.phen"
covar="/data/module4/prac3/covariates.cov"
gcta --reml --grm $grm --pheno $pheno --reml-pred-rand --qcovar $covar --out simu
```

Step 3: obtain BLUP solutions for SNP effects

```
bfile="/data/module4/prac3/gwas"
indi_blp="/data/module4/prac3/simu.indi.blp"
gcta --bfile $bfile --blup-snp $indi_blp --out simu
```

Step 4: prediction

To compute the polygenic scores in the target population (do not need a testing population as required in C+PT for finding the optimal P-value threshold):

```
target="/data/module4/prac3/target"
plink --bfile $target --score simu.snp.blp sum 1 2 3 --out simu.snp.blp
```

Use R to evaluate the prediction accuracy

```
phenFile="/data/module4/prac3/simu.phen"
covFile="/data/module4/prac3/covariates.cov"
indlistFile="/data/module4/prac3/target.indlist"
prsFile="simu.snp.blp.profile"
Rscript /data/module4/prac3/get_pred_r2.R $phenFile $covFile $indlistFile $prsFile
```

Question 6: Is the prediction accuracy from BLUP higher than that from C+PT? Why or why not? (hint: what's the assumption for the SNP effect distribution in BLUP? Is the genetic model in the simulation consistent with this assumption?)