Statistical Methods for the Prediction of Genetic Values Exercises on Corn Borer Example

Christina Lehermeier

November 18, 2015

Exercises on Corn Borer example (Table 1) based on book chapter "Statistical Methods for the Prediction of Genetic Values" from Schön and Wimmer (2014) using `R` and package `synbreed` (Wimmer et al. 2012). Exercises are partly based on course slides from Valentin Wimmer (http://synbreed.r-forge.r-project.org/).

Table 1: Pedigree, phenotypic values, and marker genotypes for eight simulated maize individuals (*Simulated SNP effect)

Cycle	Individual	Pedigree	Tunnel length [cm]	SNP1 (0)*	SNP2 (1)	SNP3 (-4)	SNP4 (4)
1	I1	P1 × P2	13	2	2	0	1
1	I2	P3 × P4	17	0	0	0	1
1	I3	-	1	0	1	2	0
2	I4	I1 × I2	17	1	1	0	2
2	I5	I1 × I2	11	1	1	0	1
2	I6	I2 × I3	6	0	1	1	0
2	I7	I1 × I2	-	1	1	0	1
2	I8	I1 × I2	-	1	1	0	0

Exercises

1. Transfer the pedigree structure of the eight simulated maize individuals from Table 1 into an object of class pedigree using the `synbreed` R package. Plot the pedigree.

2. Combine all data of the corn borer example in an object of class `gpData` called `cbData`. Include pedigree, phenotypes, and genotypes (SNPs 1 to 4) and add the names for markers and individuals for all objects. Additionally include the true genetic values of all individuals within the `covar` object.

3. Use the `summary` method for this object. Is everything correct?

4. Generate a new object called `cbData2` excluding all individuals without phenotypes.

5. Use this data to compute a single marker regression for each SNP. Which markers are significant at the 5% error rate?

6. Set up a multiple marker regression model using (i) all SNPs and (ii) only SNPs 3 and 4. Compare the results and discuss which model you would choose.

7. Predict the tunnel length for individuals I7 and I8 using effects from single marker regression of SNP4 and from multiple marker regression with SNPs 3 and 4.

8. Fit a mixed model for the 6 phenotyped individuals from the corn borer example by including all markers as random effects and an intercept as fixed effect (RR-BLUP). Assume different shrinkage factors λ for the marker effects.

9. Derive the additive relationship matrix of the 8 individuals using pedigree information

10. Predict the genetic values of individuals I7 and I8 using a mixed model including pedigree information (PBLUP model). Assume a trait heritability of 0.5.

11. Predict the genetic values of all individuals using the RR-BLUP model with all markers as random effects and shrinkage factor $\lambda = 2$. Compare the predicted genetic values with those from the PBLUP model and with the true genetic values.

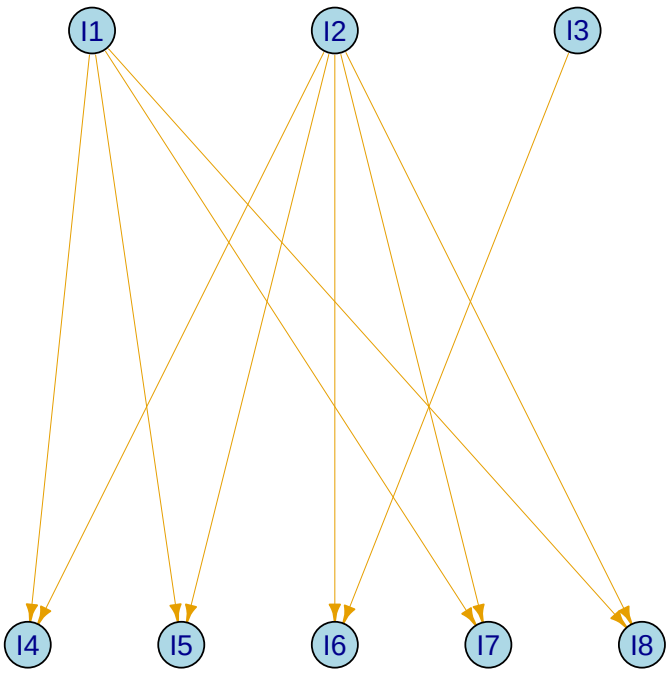
12. Calculate the fraction of genetic variance which is explained by the RR-BLUP model and the PBLUP model, respectively.

Solutions

1. Create the pedigree of corn borer example using function `create.pedigree` from `R` package `synbreed` and plot it.

```
> IDs <- paste("I", seq(1:8), sep="")
> Parent1 <- c("P1", "P3", NA, "I1", "I1", "I2", "I1", "I1")
> Parent2 <- c("P2", "P4", NA, "I2", "I2", "I3", "I2", "I2")
> Cycle <- c(1,1,1,2,2,2,2,2)
> ped <- create.pedigree(ID=IDs,Par1=Parent1,Par2=Parent2,gener=Cycle)
> plot(ped)

IGRAPH DN-- 8 10 --
+ attr: name (v/c), Par1 (v/c), Par2 (v/c), gener (v/n)
+ edges (vertex names):
 [1] I1->I4 I1->I5 I2->I6 I1->I7 I1->I8 I2->I4 I2->I5 I3->I6 I2->I7 I2->I8
IGRAPH DN-- 8 10 --
+ attr: name (v/c), Par1 (v/c), Par2 (v/c), gener (v/n)
+ edges (vertex names):
 [1] I1->I4 I1->I5 I2->I6 I1->I7 I1->I8 I2->I4 I2->I5 I3->I6 I2->I7 I2->I8
```



2. Create gpData object for Corn borer example using function `create.gpData` from `R` package `synbreed`

```
> # generate matrix of genotypic values
> geno <- matrix(c(2,2,0,1,
+                  0,0,0,1,
+                  0,1,2,0,
+                  1,1,0,2,
+                  1,1,0,1,
+                  0,1,1,0,
+                  1,1,0,1,
+                  1,1,0,0), nrow = 8, ncol = 4, byrow=T)
> # give rownames
> rownames(geno) <- IDs
> # give colnames
> colnames(geno) <- paste("SNP", seq(1:4), sep = "")
> # generate vector of phenotypic values
> pheno <- data.frame(TunnelLength=c(13, 17, 1, 17, 11, 6))
> rownames(pheno) <- IDs[1:6]
> # calculate true genetic values based on simulated SNP effects
> SNPeff <- c(0,1,-4,4)
> tgv <- data.frame(tgv = geno %*% SNPeff)
> rownames(tgv) <- IDs
> # create gpData object
> cbData <- create.gpData(pheno=pheno, geno=geno,pedigree=ped, covar=tgv)
```

3.
```
> summary(cbData)

object of class 'gpData'
covar
         No. of individuals 8
                 phenotyped 6
                  genotyped 8
pheno
         No. of traits:                  1

  TunnelLength
 Min.   : 1.00
 1st Qu.: 7.25
 Median :12.00
 Mean   :10.83
 3rd Qu.:16.00
```

```
 Max.    :17.00

geno
          No. of markers 4
          genotypes 0 1 2
          frequencies 0.40625 0.46875 0.125
          NA's 0.000 %
map
          No. of mapped markers
          No. of chromosomes     0

          markers per chromosome
         NULL

pedigree
Number of
          individuals  8
          Par 1        4
          Par 2        4
          generations  2
```

4. Use function `discard.individuals()` from `R` package `synbreed` to discard individuals without phenotypic information

```
> cbData2 <- discard.individuals(cbData,
+             which = cbData$covar$id[!cbData$covar$phenotyped])
```

5. Use function `lm()` in `R` to compute a single marker regression for each SNP. Look at output from `summary()` and `anova()`.

```
> # Estimate effect of SNP 1
> Mod1 <- lm(cbData2$pheno[,"TunnelLength",1] ~  cbData2$geno[, "SNP1"])
> summary(Mod1)


Call:
lm(formula = cbData2$pheno[, "TunnelLength", 1] ~ cbData2$geno[,
    "SNP1"])

Residuals:
  I1   I2   I3   I4   I5   I6
-2.1  8.3 -7.7  5.1 -0.9 -2.7
```

```
Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)              8.700      3.536   2.460   0.0697 .
cbData2$geno[, "SNP1"]   3.200      3.536   0.905   0.4166
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.456 on 4 degrees of freedom
Multiple R-squared:   0.17,        Adjusted R-squared:  -0.03755
F-statistic: 0.819 on 1 and 4 DF,  p-value: 0.4166

> anova(Mod1)

Analysis of Variance Table

Response: cbData2$pheno[, "TunnelLength", 1]
                      Df  Sum Sq Mean Sq F value Pr(>F)
cbData2$geno[, "SNP1"]  1  34.133  34.133   0.819 0.4166
Residuals               4 166.700  41.675

> # Estimate effect of SNP 2
> Mod2 <- lm(cbData2$pheno[,"TunnelLength",1] ~  cbData2$geno[, "SNP2"])
> summary(Mod2)

Call:
lm(formula = cbData2$pheno[, "TunnelLength", 1] ~ cbData2$geno[,
    "SNP2"])

Residuals:
     I1      I2      I3      I4      I5      I6
 4.1667  4.1667 -9.8333  6.1667  0.1667 -4.8333

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)             12.833      5.669   2.264   0.0863 .
cbData2$geno[, "SNP2"]  -2.000      4.910  -0.407   0.7046
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.943 on 4 degrees of freedom
Multiple R-squared:  0.03983,        Adjusted R-squared:  -0.2002
F-statistic: 0.1659 on 1 and 4 DF,  p-value: 0.7046
```

```
> anova(Mod2)

Analysis of Variance Table

Response: cbData2$pheno[, "TunnelLength", 1]
                      Df Sum Sq Mean Sq F value Pr(>F)
cbData2$geno[, "SNP2"]  1   8.00   8.000  0.1659 0.7046
Residuals               4 192.83  48.208

> # Estimate effect of SNP 3
> Mod3 <- lm(cbData2$pheno[,"TunnelLength",1] ~  cbData2$geno[, "SNP3"])
> summary(Mod3)

Call:
lm(formula = cbData2$pheno[, "TunnelLength", 1] ~ cbData2$geno[,
    "SNP3"])

Residuals:
     I1      I2      I3      I4      I5      I6
-1.3333  2.6667  0.6667  2.6667 -3.3333 -1.3333

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)             14.333      1.321  10.847  0.00041 ***
cbData2$geno[, "SNP3"]  -7.000      1.447  -4.836  0.00842 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.708 on 4 degrees of freedom
Multiple R-squared:  0.8539,        Adjusted R-squared:  0.8174
F-statistic: 23.39 on 1 and 4 DF,  p-value: 0.008425

> anova(Mod3)

Analysis of Variance Table

Response: cbData2$pheno[, "TunnelLength", 1]
                      Df  Sum Sq Mean Sq F value   Pr(>F)
cbData2$geno[, "SNP3"]  1 171.500 171.500  23.386 0.008425 **
Residuals               4  29.333   7.333
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> # Estimate effect of SNP 4
> Mod4 <- lm(cbData2$pheno[,"TunnelLength",1] ~  cbData2$geno[, "SNP4"])
> summary(Mod4)

Call:
lm(formula = cbData2$pheno[, "TunnelLength", 1] ~ cbData2$geno[,
    "SNP4"])

Residuals:
     I1      I2      I3      I4      I5      I6
 0.9412  4.9412 -3.7059 -2.4118 -1.0588  1.2941

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)              4.706      2.215   2.125    0.101
cbData2$geno[, "SNP4"]   7.353      2.050   3.586    0.023 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.451 on 4 degrees of freedom
Multiple R-squared:  0.7628,        Adjusted R-squared:  0.7034
F-statistic: 12.86 on 1 and 4 DF,  p-value: 0.02304

> anova(Mod4)

Analysis of Variance Table

Response: cbData2$pheno[, "TunnelLength", 1]
                      Df  Sum Sq Mean Sq F value  Pr(>F)
cbData2$geno[, "SNP4"]  1 153.186 153.186   12.86 0.02304 *
Residuals               4  47.647  11.912
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In the following, the same single marker regression model for each marker is performed as above, but a `for` loop in `R` is used. Estimated SNP effects and p-values from the models are extracted. Note that `for` loops in `R` are not very efficient and should be avoided for larger number of markers, instead function `apply` can be used.

```
> # generate empty vector were effects can be saved
> betaHat <- vector(length=ncol(cbData2$geno))
> names(betaHat) <- colnames(cbData2$geno)
```

```
> # generate empty vector were p-values can be saved
> pVal <- vector(length=ncol(cbData2$geno))
> names(pVal) <- colnames(cbData2$geno)
> for(i in 1:4){
+ SMR <- lm(cbData2$pheno[,"TunnelLength",1] ~  cbData2$geno[, i])
+ betaHat[i] <- coefficients(SMR)[2]  # extract SNP effect from each SMR model
+ pVal[i] <-  anova(SMR)$Pr[1]        # extract p-value from each SMR model
+ }
> # which markers significant at 5% error rate (p value < 0.05)
> #(Note: not corrected for multiple testing!)
> which(pVal <0.05)

SNP3 SNP4
   3    4
```

6.
```
> # Multiple marker model using all SNPs
> MMR_all <-  lm(cbData2$pheno[,"TunnelLength",1] ~  cbData2$geno)
> summary(MMR_all)

Call:
lm(formula = cbData2$pheno[, "TunnelLength", 1] ~ cbData2$geno)

Residuals:
        I1          I2          I3          I4          I5          I6
 1.333e+00   1.333e+00   3.223e-16  -1.218e-16  -2.667e+00  -4.548e-16

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       12.3333     5.4975   2.243    0.267
cbData2$genoSNP1  -0.6667     8.8443  -0.075    0.952
cbData2$genoSNP2  -1.3333     9.1409  -0.146    0.908
cbData2$genoSNP3  -5.0000     4.6188  -1.083    0.475
cbData2$genoSNP4   3.3333     3.7712   0.884    0.539

Residual standard error: 3.266 on 1 degrees of freedom
Multiple R-squared:  0.9469,        Adjusted R-squared:  0.7344
F-statistic: 4.457 on 4 and 1 DF,  p-value: 0.3396

> anova(MMR_all)

Analysis of Variance Table
```

```
Response: cbData2$pheno[, "TunnelLength", 1]
             Df  Sum Sq Mean Sq F value Pr(>F)
cbData2$geno  4 190.167  47.542   4.457 0.3396
Residuals     1  10.667  10.667

> # Multiple marker model using SNP3 and SNP4
> MMR_SNP34 <-  lm(cbData2$pheno[,"TunnelLength",1] ~
+                  cbData2$geno[,c("SNP3", "SNP4")])
> summary(MMR_SNP34)

Call:
lm(formula = cbData2$pheno[, "TunnelLength", 1] ~ cbData2$geno[,
    c("SNP3", "SNP4")])

Residuals:
      I1       I2       I3       I4       I5       I6
-0.72727  3.27273 -0.09091  0.09091 -2.72727  0.18182

Coefficients:
                                     Estimate Std. Error t value Pr(>|t|)
(Intercept)                            10.545      3.151   3.346   0.0442 *
cbData2$geno[, c("SNP3", "SNP4")]SNP3  -4.727      2.196  -2.152   0.1204
cbData2$geno[, c("SNP3", "SNP4")]SNP4   3.182      2.441   1.303   0.2834
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.498 on 3 degrees of freedom
Multiple R-squared:  0.9068,        Adjusted R-squared:  0.8446
F-statistic: 14.59 on 2 and 3 DF,  p-value: 0.02847

> anova(MMR_SNP34)

Analysis of Variance Table

Response: cbData2$pheno[, "TunnelLength", 1]
                                Df  Sum Sq Mean Sq F value  Pr(>F)
cbData2$geno[, c("SNP3", "SNP4")]  2 182.106  91.053  14.586 0.02847 *
Residuals                          3  18.727   6.242
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

7.
```
> # Predict values based on SMR of SNP4
> X <- cbind(c(1,1), cbData$geno[c("I7", "I8"), c("SNP4")])
> betaHatMod4 <- coefficients(Mod4)
> yHatMod4 <- X %*% betaHatMod4
> # Predict values based on MMR
> X <- cbind(c(1,1), cbData$geno[c("I7", "I8"), c("SNP3", "SNP4")])
> betaHatMMR <- coefficients(MMR_SNP34)
> yHatMMR <- X %*% betaHatMMR
```

8. Use function `MME` from `synbreed` package to fit mixed model equation.

```
> X <- matrix(rep(1, times=6),nrow=6,ncol=1)
> W <- cbData2$geno
> lambda <- 2
> MME1 <- MME(X=X,Z=W, GI=diag(4)*lambda, RI=diag(6), y=cbData$pheno[,1,1])
> # fixed effect
> MME1$b

[1] 11.2114

> # vector of random SNP effects:
> MME1$u

[1]  0.5130641 -1.2565321 -3.1235154  2.5178147
```

9. Derive the additive relationship matrix for all 8 individuals using function `kin()`

```
> A <- kin(cbData, ret="add")
> A

    I1  I2  I3   I4   I5   I6   I7   I8
I1 1.0 0.0 0.0 0.50 0.50 0.00 0.50 0.50
I2 0.0 1.0 0.0 0.50 0.50 0.50 0.50 0.50
I3 0.0 0.0 1.0 0.00 0.00 0.50 0.00 0.00
I4 0.5 0.5 0.0 1.00 0.50 0.25 0.50 0.50
I5 0.5 0.5 0.0 0.50 1.00 0.25 0.50 0.50
I6 0.0 0.5 0.5 0.25 0.25 1.00 0.25 0.25
I7 0.5 0.5 0.0 0.50 0.50 0.25 1.00 0.50
I8 0.5 0.5 0.0 0.50 0.50 0.25 0.50 1.00
attr(,"info")
[1] "This relationshipMatrix was calculated by synbreed version 0.11-26"
attr(,"type")
```

```
[1] "add"
attr(,"class")
[1] "relationshipMatrix" "matrix"
```

10.
```
> # Fit PBLUP model for all 8 individuals
> X <- matrix(1,nrow=6,ncol=1)
> W <- cbind(diag(6),rep(0,times=6), rep(0,times=6))
> lambda <- 1
> PBLUP <- MME(X=X,Z=W, GI=solve(A)*lambda, RI=diag(6), y=cbData$pheno[,1,1])
> # predicted genetic values of all 8 individuals
> gPBLUP <- PBLUP$u
> names(gPBLUP) <- rownames(A)
> gPBLUP
```

```
       I1        I2        I3        I4        I5        I6        I7        I8
 1.835351  3.292978 -5.128329  3.985472  1.985472 -2.002421  2.564165  2.564165
```

11.
```
> X <- matrix(rep(1, times=6),nrow=6,ncol=1)
> W <- cbData2$geno
> lambda <- 2
> MME1 <- MME(X=X,Z=W, GI=diag(4)*lambda, RI=diag(6), y=cbData$pheno[,1,1])
> # fixed effect
> MME1$b
```

```
[1] 11.2114
```

```
> # vector of random SNP effects:
> MME1$u
```

```
[1]  0.5130641 -1.2565321 -3.1235154  2.5178147
```

```
> # predict genetic values of all 8 individuals based on SNP effects from RRBLUP
> gRRBLUP <- cbData$geno %*% MME1$u
> names(gRRBLUP) <- rownames(A)
> # generate a table including true genetic values (tgv),
> # predicted genetic values from PBLUP (gPBLUP) and
> # from RRBLUP (gRRBLUP) for all individuals
> (Tab <- data.frame(tgv=tgv, gPBLUP=gPBLUP, gRRBLUP=gRRBLUP))
```

```
     tgv    gPBLUP    gRRBLUP
I1     6  1.835351  1.0308789
I2     4  3.292978  2.5178147
I3    -7 -5.128329 -7.5035629
I4     9  3.985472  4.2921615
I5     5  1.985472  1.7743468
I6    -3 -2.002421 -4.3800475
I7     5  2.564165  1.7743468
I8     1  2.564165 -0.7434679
```

12.
```
> # calculate R^2 for PBLUP
> (R2PBLUP <- cor(Tab$tgv, Tab$gPBLUP)^2)

[1] 0.84743

> # calculate R^2 for RRBLUP
> (R2RRBLUP <- cor(Tab$tgv, Tab$gRRBLUP)^2)

[1] 0.9542008
```