```
#########################
# Exercises on Corn Borer Example based on book chapter "Statistical Methods for the Prediction of Genetic Values" from Schoen and Wimmer (2014)
# Created with "synbreed" R package (Wimmer et al. 2012) version 0.11-26 and R version 3.2.2
# Author: Christina Lehermeier
# Date: Nov. 18 2015
#########################

# install and load synbreed package
install.packages("synbreed", repos="http://R-Forge.R-project.org")
library(synbreed)

##
# (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)
# create pedigree
ped <- create.pedigree(ID=IDs,Par1=Parent1,Par2=Parent2,gener=Cycle)
# plot pedigree
plot(ped)

##
# (2) Create gpData object for Corn borer example using function \texttt{create.gpData} from \texttt{R} package \texttt{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) Use summary method for gpData object cbData
##

summary(cbData)


##
# (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)
anova(Mod1)

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

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

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

```
summary(Mod4)
anova(Mod4)

# fitting single marker regression for every marker succesively and extract SNP
effects and p-values using for loop (Note: in R for loops are not efficient and
using function apply() should be preferred)

# 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)

####
# (6) Set up a multiple marker regression model
####

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

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

####
# (7) Predict tunnel length for individuals I7 and I8
####

# 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
# vector of random SNP effects:
MME1$u

####
# (9) Derive the additive relationship matrix for all 8 individuals using function
\texttt{kin()}
####

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

####
# (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

####
# (11) Predict genetic values of all individuals using the RR-BLUP model
```

```
####

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$b
# vector of random SNP effects:
MME1$u
# 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))


####
# (12) Calculate the fraction of genetic variance which is explained by the RR-
BLUP model and the PBLUP model
####

# calculate R^2 for PBLUP
(R2PBLUP <- cor(Tab$tgv, Tab$gPBLUP)^2)
# calculate R^2 for RRBLUP
(R2RRBLUP <- cor(Tab$tgv, Tab$gRRBLUP)^2)
```