Cauliflower_GxE_selection_study

Christian Lampei

28.07.2014

Reading in data and preparing it for later use:

```r
data <- read.table("cauli_flower_data_all.csv", header=T)
data$genotype <- as.factor(data$genotype)
data$replicate <- as.factor(data$replicate)
data$time <- as.factor(data$time)
data$env <- as.factor(data$env)
data$year <- as.factor(data$year)
data$plant <- as.factor(data$plant)
str(data)
```

Removing accessions that were completely missing in one location and season or more. Interestingly all these accessions were from USDA.

```r
data$out <- 0
vect_na <- c(43, 48, 83, 101, 2, 6, 15, 25, 35, 37, 38, 39, 41, 45, 50, 56, 61,
             66, 67, 68, 74, 82)

for (i in 1:length(vect_na)) {
  data[which(data$genotype==vect_na[i]), ]$out <- 1
}
table(data$out)
data1 <- data[data$out!=1, ]
data1 <- droplevels(data1)
length(levels(data1$genotype))
length(levels(data$genotype))
```

Now the data is ready for use. Of originaly 200 genotypes, 178 genotypes remained in the data set.

In the next step the homogeneity of variances is tested using the flingner-killen test.

Test for cultivation methods:

```r
fligner.test(curd~location, data=data1)
fligner.test(days~location, data=data1)
```

Test for growing season:

```r
fligner.test(curd~time, data=data1)
fligner.test(days~time, data=data1)
```

And for both together in one environment factor (env):

```r
fligner.test(curd~env, data=data1)
fligner.test(days~env, data=data1)
```

Most is highly significant, only in days to budding the cultivation method is only marginally significant.

In the next step tested for significant effects of the factors cultivation method and growing season and their interaction. I used a mixed-effects model that accounted for heterogenous variances. We need the package "nlme".

First a variance structure needs to be modeled:

```r
require(nlme)

vf3 <- varComb(varIdent(form= ~1|location), varIdent(form= ~1|time))
```

A new variabel for a block in environment effect is needed:

```r
data1$blockinenv <- with(data1, factor(paste(env, replicate, sep=".")))
```

The model is formulated and evaluated. **Whatch out! Running the models may take several minutes (10-20 or more) depending on the speed of your computer or server.**

For curd width:

```r
m3 <- lme(curd~location*time*genotype, random= ~ 1|blockinenv,  weights=vf3, method="REML",
          data=data1[!is.na(data1$curd),])
anova(m3)
```

And for days to budding:

```r
m3d <- lme(days~location*time*genotype, random= ~ 1|blockinenv,  weights=vf3, method="REML",
           data=data1[!is.na(data1$days),])
anova(m3d)
```

After running the models we want to know which levels of the interaction are different from each other. For this aim we run a multiple comparison test using the multcomp package.

```r
require(multcomp)
vfA <- varIdent(form= ~1|env)
mx <- lme(curd ~ env, random= ~ 1|genotype, weights=vfA, data=data_means)
summary(glht(mx, linfct=mcp(env="Tukey"), ajust="BH"))

mx <- lme(days ~ env, random= ~ 1|genotype, weights=vfA, data=data_means)
summary(glht(mx, linfct=mcp(env="Tukey"), ajust="BH"))
```

Now an overview graph with indication of significant differences can be produced.

```
setEPS()
postscript("cauli_boxpl.eps", width=10, height=10)
par(mfrow=c(2,1), mar=c(1,5.5,1,2), mgp=c(3.5,1,0))
box <- boxplot(predict(m3) ~ location*time, data=data_means, notch=T,
               col=c("white", "grey70"), ylab="curd width", las=1,
               xaxt="n", cex.axis=1.5, cex.lab=1.5)
segments(5, 15.2, 5, 15.5, lty=1)
segments(6, 14.7, 6, 15.5, lty=1)
segments(5,15.5, 5.3, 15.5, lty=1)
text(5.5, 15.5, "***", cex=1.5)
segments(5.7,15.5, 6, 15.5, lty=1)
legend("topright", legend=c("conventional", "organic"),
       fill=c("white", "grey70"), bty="n", cex=1.5)
par(mar=c(3,5.5,1,2))
box <- boxplot(predict(md3) ~ location*time, data=data_means, notch=T,
               col=c("white", "grey70"), ylab="days to budding", las=1,
               names=c("2011 Jun", "2011 Jun", "2012 Apr", "2012 Apr", "2012 Aug", "2012 Aug"),
               cex.axis=1.5, cex.lab=1.5)
segments(1, 121, 1, 124, lty=1)
segments(2, 121, 2, 124, lty=1)
segments(1, 124, 1.3, 124, lty=1)
text(1.5, 124, "***", cex=1.5)
segments(1.7, 124, 2, 124, lty=1)
####
segments(3, 121, 3, 124, lty=1)
segments(4, 121, 4, 124, lty=1)
segments(3, 124, 3.3, 124, lty=1)
text(3.5, 124, "***", cex=1.5)
segments(3.7, 124, 4, 124, lty=1)

dev.off()
```

The next part of the analysis concerns the genetic correlations and the variance components.

First I calculated the mean for each replicate (5 plants from the same row).

```
data_means <- aggregate(data1, by=list(data1$year, data1$season, data1$time,
                                       data1$location, data1$genotype,
                                       data1$replicate), mean, na.rm=T)
data_means <- data_means[,-c(7,8,9, 10, 11, 12, 13, 14, 15)]
names(data_means)[1:6] <- c("year", "season", "time", "location", "genotype", "replicate")
str(data_means)
```

Now I splitted the data into organic and conventional since the variance component should be estimated in each cultivation method seperately.

```
org <- droplevels(data_means[data_means$location=="KL", ])
conv <- droplevels(data_means[data_means$location=="HD", ])
```

For the models I used the package “lme4”. The package requiers the detaching of the “nlme” package.

```
search()
detach(package:nlme)
require(lme4)
```

These models run quick. Afterwards for each model I extracted the variance components as a data frame to be able to use them for calculating heritability.

From here I do all steps for curd width. The days to budding script will be found uncommented in the end since it consists of the same steps:

```
curd_org <- lmer(curd ~  1 + (1|time/replicate) + (1|genotype) + (1|time:genotype), data=org)
var_cuorg <- data.frame(as.numeric(matrix(VarCorr(curd_org))))
rownames(var_cuorg) <- c("time:genotype", "genotype", "replicate:time", "time")

curd_conv <- lmer(curd ~ 1 + (1|time/replicate) + (1|genotype) + (1|time:genotype), data=conv)
var_cuconv <- data.frame(as.numeric(matrix(VarCorr(curd_conv))))
rownames(var_cuconv) <- c("time:genotype", "genotype", "replicate:time", "time")
```

Printing the variance components including the residual variance (sigma):

```
var_cuorg
sigma(curd_org)^2
var_cuconv
sigma(curd_conv)^2
```

Calculating the heritability:

```
(h_org <- var_cuorg[2,]/(var_cuorg[2,]+(var_cuorg[1,]/3)+(sigma(curd_org)^2/6)))
(h_conv <- var_cuconv[2,]/(var_cuconv[2,]+var_cuconv[1,]/3+sigma(curd_conv)^2/6))
```

After calculating the heritability for each cultivation method I want to estimate a standart error. This can be estimated by means of parametric bootstraping.

For curd width in organic cultivation

```
gg <- simulate(curd_org, 1000)
vh_orgs <- data.frame(Vgt=as.numeric(),
                      Vg=as.numeric(),
                      Ve=as.numeric(),
                      stringsAsFactors=FALSE)

for(i in 1:ncol(gg)){
mr <- refit(curd_org, gg[,i])
varmr <- data.frame(as.numeric(matrix(VarCorr(mr))))
vh_orgs[i,1] <- varmr[1,]
vh_orgs[i,2] <- varmr[2,]
vh_orgs[i,3] <- sigma(mr)^2
}

vh_orgs$h2  <- vh_orgs[,2]/(vh_orgs[,2]+vh_orgs[,1]/3+vh_orgs[,3]/6)
(SEvgt <- sd(vh_orgs$Vgt))
(SEvg <- sd(vh_orgs$Vg))
(SEve <- sd(vh_orgs$Ve))
(SEh2_org <- sd(vh_orgs$h2))
```

For curd width in conventional cultivation:

```
gg <- simulate(curd_conv, 1000)
vh_convs <- data.frame(Vgt=as.numeric(),
                       Vg=as.numeric(),
                       Ve=as.numeric(),
                       stringsAsFactors=FALSE)

for(i in 1:ncol(gg)){
  mr <- refit(curd_conv, gg[,i])
  varmr <- data.frame(as.numeric(matrix(VarCorr(mr))))
  vh_convs[i,1] <- varmr[1,]
  vh_convs[i,2] <- varmr[2,]
  vh_convs[i,3] <- sigma(mr)^2
}
vh_convs$h2  <- vh_convs[,2]/(vh_convs[,2]+vh_convs[,1]/3+vh_convs[,3]/6)
(SEvgtconv <- sd(vh_convs$Vgt))
(SEvgconv <- sd(vh_convs$Vg))
(SEveconv <- sd(vh_convs$Ve))
(SEh2_conv <- sd(vh_convs$h2))
```

To estimate the genetic correlation I now extracted the BLUPs from the model of each cultivation method.

```
bcurd_org <- ranef(curd_org)$genotype[[1]]
bcurd_conv <- ranef(curd_conv)$genotype[[1]]
```

Now it is possible to estimate the genetic correlation as a pearson correlation of the BLUPs.

```
cor.test(bcurd_org, bcurd_conv)
```

And I tested for the efficiency of direct (rd) or indirect (rid) selection for organic cultivation.

```
(rd <- h_org*sqrt(var_cuorg[2,]))
(rid <-h_conv*cor(bcurd_org, bcurd_conv)*sqrt(var_cuorg[2,]))
rid/rd
```

The next step is the selection of varieties. For this I used the BLUPs as estimate of yield and the genotype variance across growing seasons as a measure of stability.

I first calculate the genotype means for each cultivation methode and growing season:

```
data_means2 <- aggregate(data1, by=list(data1$year, data1$season, data1$time,
                                        data1$location, data1$genotype), mean, na.rm=T)
data_means2 <- data_means2[,-c(6,7,8,9,10,11,12,13)]
names(data_means2)[1:5] <- c("year", "season", "time", "location", "genotype")

org2 <- data_means2[data_means2$location=="KL" , ]
conv2 <- data_means2[data_means2$location=="HD" , ]
```

In a next step the variance across environments is calculated for each genotype to get a measure for yield stability (see article for more details). Both, the yield and the stability measures are than standardized to enable an equal selection fot yield and stability. This is done first for the organic cultivation.

```
stab_org_curd <- tapply(org2$curd, org2$genotype, var)
stab_org_curd1 <- (stab_org_curd-mean(stab_org_curd))/sd(stab_org_curd)
stab_org_curd2 <- stab_org_curd1*(-1)
bcurd_org1 <- (bcurd_org-mean(bcurd_org))/sd(bcurd_org)
```

For selection the standardized values of yield and stability are added, a dataframe with all three values for each genotype is constructed and sorted according the selection value. The 10 best genotypes are extracted in a seperate object.

```
sel_curd_org <- stab_org_curd2+bcurd_org1
sel_tab_curd_org <- cbind(sel_curd_org , stab_org_curd, bcurd_org)
sel_tab_curd_org <- sel_tab_curd_org[order(-sel_tab_curd_org[,1]),]
(sel10_tab_curd_org <- head(sel_tab_curd_org, 10))
```

Now the same needs to be done with the conventional cultivation.

```
#stabilty
stab_conv_curd <- tapply(conv2$curd, conv2$genotype, var)
stab_conv_curd1 <- (stab_conv_curd-mean(stab_conv_curd))/sd(stab_conv_curd)
stab_conv_curd2 <- stab_conv_curd1*(-1)
# blubs standardized
bcurd_conv1 <- (bcurd_conv-mean(bcurd_conv))/sd(bcurd_conv)
#selection value
sel_curd_conv <- stab_conv_curd2+bcurd_conv1
# data frame
sel_tab_curd_conv <- cbind(sel_curd_conv , stab_conv_curd, bcurd_conv)
sel_tab_curd_conv <- sel_tab_curd_conv[order(-sel_tab_curd_conv[,1]),]
(sel10_tab_curd_conv <- head(sel_tab_curd_conv, 10))
```

Before the selection plots can be produced the same analysis needs to be performed for days to budding:

```
#####################################
#days
days_org <- lmer(days ~ 1 + (1|time/replicate) + (1|genotype) + (1|time:genotype), data=org)
var_daorg <- data.frame(as.numeric(matrix(VarCorr(days_org))))
rownames(var_daorg) <- c("time:genotype", "genotype", "replicate:time", "time")

days_conv <- lmer(days ~ 1 + (1|time/replicate) + (1|genotype) + (1|time:genotype), data=conv)
var_daconv <- data.frame(as.numeric(matrix(VarCorr(days_conv))))
rownames(var_daconv) <- c("time:genotype", "genotype", "replicate:time", time)

##########
var_daorg
sigma(days_org)^2
var_daconv
sigma(days_conv)^2
# h2
(h_orgd <- var_daorg[2,]/(var_daorg[2,]+var_daorg[1,]/3+sigma(days_org)^2/6))
(h_convd <- var_daconv[2,]/(var_daconv[2,]+var_daconv[1,]/3+sigma(days_conv)^2/6))
##########
# simmulated standard error h2
# organic days
gg <- simulate(days_org, 1000)
vh_orgds <- data.frame(Vgt=as.numeric(),
                      Vg=as.numeric(),
                      Ve=as.numeric(),
                      stringsAsFactors=FALSE)
```

```
for(i in 1:ncol(gg)){
  mr <- refit(days_org, gg[,i])
  varmr <- data.frame(as.numeric(matrix(VarCorr(mr))))
  vh_orgds[i,1] <- varmr[1,]
  vh_orgds[i,2] <- varmr[2,]
  vh_orgds[i,3] <- sigma(mr)^2
}
vh_orgds$h2  <- vh_orgds[,2]/(vh_orgds[,2]+vh_orgds[,1]/3+vh_orgds[,3]/6)
(SEvgt <- sd(vh_orgds$Vgt))
(SEvg <- sd(vh_orgds$Vg))
(SEve <- sd(vh_orgds$Ve))
(SEh2_org <- sd(vh_orgds$h2))


# conventional days
gg <- simulate(days_conv, 1000)
vh_convds <- data.frame(Vgt=as.numeric(),
                       Vg=as.numeric(),
                       Ve=as.numeric(),
                       stringsAsFactors=FALSE)

for(i in 1:ncol(gg)){
  mr <- refit(days_conv, gg[,i])
  varmr <- data.frame(as.numeric(matrix(VarCorr(mr))))
  vh_convds[i,1] <- varmr[1,]
  vh_convds[i,2] <- varmr[2,]
  vh_convds[i,3] <- sigma(mr)^2
}
vh_convds$h2  <- vh_convds[,2]/(vh_convds[,2]+vh_convds[,1]/3+vh_convds[,3]/6)
(SEvgtconv <- sd(vh_convds$Vgt))
(SEvgconv <- sd(vh_convds$Vg))
(SEveconv <- sd(vh_convds$Ve))
(SEh2_conv <- sd(vh_convds$h2))
##########
# blups
bdays_org <- ranef(days_org)$genotype[[1]]
bdays_conv <- ranef(days_conv)$genotype[[1]]
#########
# genetic correlation
cor.test(bdays_org, bdays_conv)
# efficiecy of selection
(rd <- h_orgd*sqrt(var_daorg[2,]))
(rid <- h_convd*cor(bdays_org, bdays_conv)*sqrt(var_daorg[2,]))
rid/rd
#############################
```

```r
#organic
#stabilty
stab_org_days <- tapply(org2$days, org2$genotype, var)
stab_org_days1 <- (stab_org_days-mean(stab_org_days))/sd(stab_org_days)
stab_org_days2 <- stab_org_days1
# blubs standardized
bdays_org1 <- (bdays_org-mean(bdays_org))/sd(bdays_org)
#selection value
sel_days_org <- stab_org_days2+bdays_org1
# data frame
sel_tab_days_org <- cbind(sel_days_org , stab_org_days, bdays_org)
sel_tab_days_org <- sel_tab_days_org[order(sel_tab_days_org[,1]),]
(sel10_tab_days_org <- head(sel_tab_days_org, 10))
##############
#conventional
#stabilty
stab_conv_days <- tapply(conv2$days, conv2$genotype, var)
stab_conv_days1 <- (stab_conv_days-mean(stab_conv_days))/sd(stab_conv_days)
stab_conv_days2 <- stab_conv_days1
# blubs standardized
bdays_conv1 <- (bdays_conv-mean(bdays_conv))/sd(bdays_conv)
#selection value
sel_days_conv <- stab_conv_days2+bdays_conv1
# data frame
sel_tab_days_conv <- cbind(sel_days_conv , stab_conv_days, bdays_conv)
sel_tab_days_conv <- sel_tab_days_conv[order(sel_tab_days_conv[,1]),]
(sel10_tab_days_conv <- head(sel_tab_days_conv, 10))
```

The last thing remaining to do is a figure that gives overview over the selected varieties in respect to all varieties. For these plots the orginal values of yield and stability were used.

```r
setEPS()
postscript("selection_plots4.eps", width=15, height=15)
par(mfrow=c(2,2), cex=1.5,  mar=c(4,4,1,1), mgp=c(2.5,1,0))

plot(stab_org_curd~bcurd_org, type="n", las=1, ylab="Stability",
     xlab="Best linear unbiased prediction")
text(bcurd_org, stab_org_curd, names(stab_org_curd), cex=.6, col="grey60")
text(sel10_tab_curd_org[,3], sel10_tab_curd_org[,2], rownames(sel10_tab_curd_org),
     cex=.6, col="black")

text(-.8, 14, "A", cex=2)

plot(stab_conv_curd~bcurd_conv, type="n", las=1, ylab="Stability",
     xlab="Best linear unbiased prediction")
text(bcurd_conv, stab_conv_curd, names(stab_conv_curd), cex=.6, col="grey60")
text(sel10_tab_curd_conv[,3], sel10_tab_curd_conv[,2], rownames(sel10_tab_curd_conv),
     cex=.6, col="black")
text(-.7, 7.6, "B", cex=2)

plot(stab_org_days~bdays_org, type="n", las=1, ylab="Stability",
     xlab="Best linear unbiased prediction")
text(bdays_org, stab_org_days, names(stab_org_days), cex=.6, col="grey60")
text(sel10_tab_days_org[,3], sel10_tab_days_org[,2], rownames(sel10_tab_days_org),
     cex=.6, col="black")
text(-24, 680, "C", cex=2)

plot(stab_conv_days~bdays_conv, type="n", las=1, ylab="Stability",
     xlab="Best linear unbiased prediction")
text(bdays_conv, stab_conv_days, names(stab_conv_days), cex=.6, col="grey60")
text(sel10_tab_days_conv[,3], sel10_tab_days_conv[,2], rownames(sel10_tab_days_conv),
     cex=.6, col="black")
text(-25, 1100, "D", cex=2)

dev.off()
```