Data analysis: Genomic prediction

Azam et al.

Genomic prediction

```
library(asreml)
library(StageWise)

effects <- data.frame(name=c("block"),
                      fixed=c(FALSE),
                      factor=c(TRUE))

model_cs <- Stage1(filename="FileS5.csv", traits="Scab_score",
                   spline = c("row","range"), solver = "spats", effects = effects)
```

```
## Warning in stat_boxplot(outlier.color = "red"): Ignoring unknown parameters:
## ‘outlier.colour‘
```

```
# BLUEs and vcov matrix from Stage1

Stage1.blues <- model_cs$blues
Stage1.vcov <- model_cs$vcov

# Read geno file

geno <- read_geno(
  filename = "FileS1.csv", ploidy = 4, map = TRUE, min.minor.allele = 5,
  dominance = T)
```

```
## Minor allele threshold = 5 genotypes
## Number of markers = 1273
## Number of genotypes = 439
```

```
# Separate the ids and make 5 folds for each population (DP and ED)

id <- data.frame(id = unique(Stage1.blues$id))
id_cntrl <- id$id[c(1,164,165,166,167,431)]
id_cntrl
```

```
## [1] "Desiree"    "Electra"    "Jubel"      "MarisPiper" "MayanGold"
## [6] "Picasso"
```

```r
id <- id$id[-c(1,164,165,166,167,431)]

id_ED <- unique(id[substr(id,1,1)=="E"])
n_ED <- length(id_ED)
n_ED
```

```
## [1] 166
```

```r
id_DP <- unique(id[substr(id,1,1)=="P"])
id_DP_sample <- sample(id_DP, 166)
n_DP <- length(id_DP_sample)
n_DP
```

```
## [1] 166
```

```r
ED.folds <- split(sample(id_ED),cut(1:n_ED,breaks=5))
DP.folds <- split(sample(id_DP_sample),cut(1:n_DP,breaks=5))

# Variance components estimation

ans2b <- Stage2(data=Stage1.blues, vcov=Stage1.vcov, geno = geno,
                non.add = "g.resid", silent = FALSE)
ans2c <- Stage2(data=Stage1.blues, vcov=Stage1.vcov, geno = geno,
                non.add = "dom", silent = FALSE)

ans2b$aic
```

```
## [1] 2405.093
```

```r
ans2c$aic
```

```
## [1] 2401.013
```

As there is not much difference in the AIC of genetic residual and dominant model, we choose the genetic residual model.

```r
summary(ans2b$vars)
```

```
##               Variance   PVE
## env               1.30    NA
## additive          0.23 0.141
## g.resid           0.12 0.073
## g x env           0.57 0.350
## Stage1.error      0.71 0.436
```

```r
# BLUP estimation without marker data = observed values

ans2 <- Stage2(data=Stage1.blues, vcov=Stage1.vcov, silent = FALSE)
prep_b <- blup_prep(data=Stage1.blues,
                        vcov=Stage1.vcov,
                        vars=ans2$vars)
GV_obs <- blup(prep_b, what="GV")
GV_obs[1:5,1:3]
```

```
##        id    value        r2
## 1 Desiree 3.914775 0.6872260
## 2      E1 5.220876 0.5543410
## 3     E10 5.049699 0.5543507
## 4    E100 4.711003 0.5537079
## 5    E101 4.217328 0.5537407
```

```
# BLUP estimation with marker data = predicted values
## ED training population
GV_E <- NULL
GV_ED_E <- NULL
GV_DP_E <- NULL
R_ED_E <- NULL
R_DP_E <- NULL

for (i in 1:5) {

  ### ED validation population

  prep <- blup_prep(Stage1.blues,Stage1.vcov,geno,vars=ans2b$vars,
                    mask=data.frame(id=c(id_DP,id_cntrl,ED.folds[[i]])))
  GV_E[[i]] <- blup(prep,geno,what="GV")
  GV_ED_E[[i]] <- GV_E[[i]][GV_E[[i]]$id %in% ED.folds[[i]],]

  pred <- GV_ED_E[[i]]
  merge <- merge(GV_obs, pred, by = "id")
  #### Correlation of each fold between observed and predicted values.
  R_ED_E[i] <- cor(merge$value.x,merge$value.y)


  ### DP validation population

  GV_DP_E[[i]] <- GV_E[[i]][GV_E[[i]]$id %in% DP.folds[[i]],]
  pred <- GV_DP_E[[i]]
  merge <- merge(GV_obs, pred, by = "id")
  #### Correlation of each fold between observed and predicted values.
  R_DP_E[i] <- cor(merge$value.x,merge$value.y)
}
```

```
## DP training population

GV_P <- NULL
GV_DP_P <- NULL
GV_ED_P <- NULL
R_ED_P <- NULL
R_DP_P <- NULL

for (i in 1:5) {

  ##DP validation population

  prep <- blup_prep(Stage1.blues,Stage1.vcov,geno,vars=ans2b$vars,
                    mask=data.frame(id=c(id_ED,DP.folds[[i]],id_cntrl,
                                         setdiff(id_DP, id_DP_sample))))
```

```
  GV_P[[i]] <- blup(prep,geno,what="GV")
  GV_DP_P[[i]] <- GV_P[[i]][GV_P[[i]]$id %in% DP.folds[[i]],]

  pred <- GV_DP_P[[i]]
  merge <- merge(GV_obs, pred, by = "id")#
  #### Correlation of each fold between observed and predicted values.
  R_DP_P[i] <- cor(merge$value.x,merge$value.y)


  ##ED validation population

  GV_ED_P[[i]] <- GV_P[[i]][GV_P[[i]]$id %in% ED.folds[[i]],]
  pred <- GV_ED_P[[i]]
  merge <- merge(GV_obs, pred, by = "id")
  #### Correlation of each fold between observed and predicted values.
  R_ED_P[i] <- cor(merge$value.x,merge$value.y)
}
```

We performed 10 iteration of the previous code and the output was saved in the dataframe CV.

```
CV2 <- data.frame(
  Training_population = rep(c("DP","ED","ED","DP"), each = 5),
  Validation_population = rep(c("DP","ED"), each =10),
  Sibs = rep(c("Two shared parents", "One shared parent", "Two shared parents",
               "One shared parent"), each = 5),
  PA = c(R_DP_P, R_DP_E, R_ED_E, R_ED_P),
  Iteration = 10, Folds = rep(1:5)
)
```

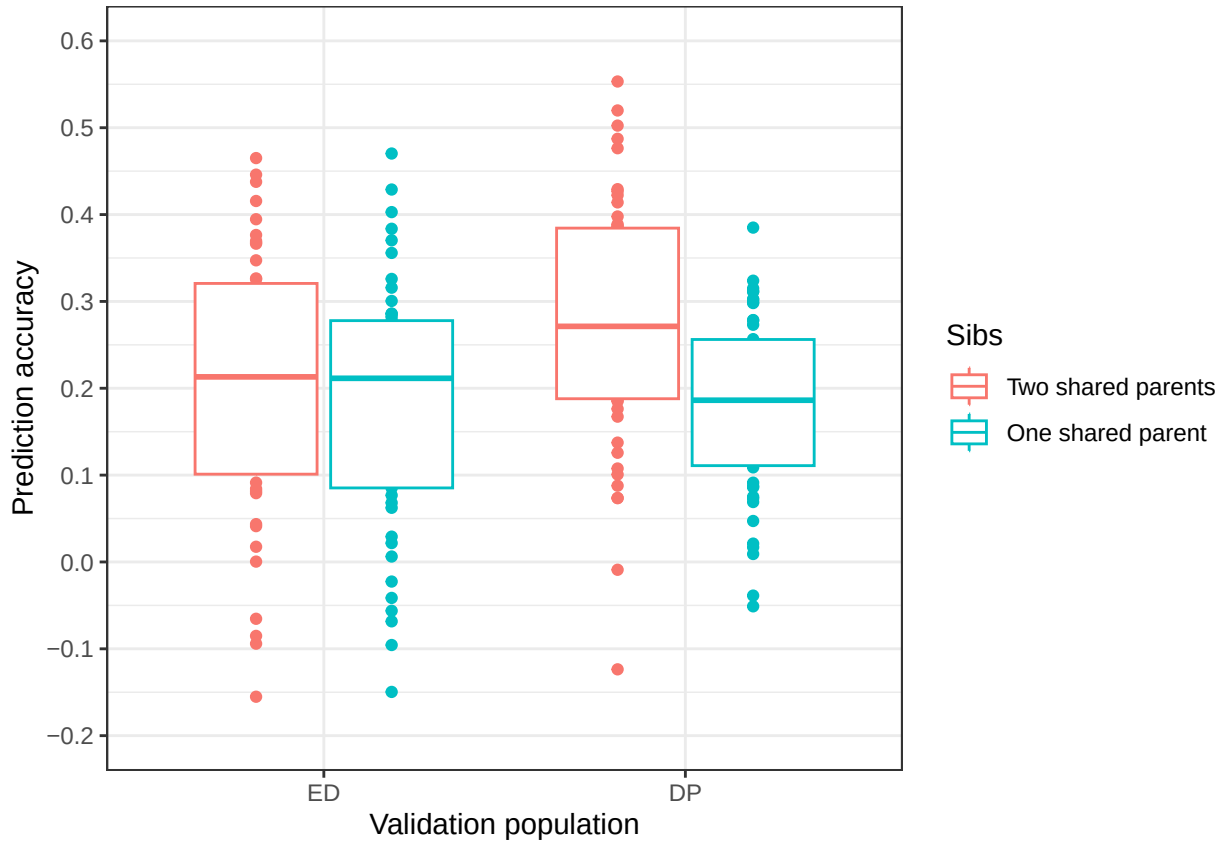
For each iteration, new folds are created for ED, and for DP, a new sample of 166 is drawn before generating the folds. After the first iteration, CV is updated to CV2, and the iteration number is incremented for each subsequent iteration before being bound back to the original CV dataframe.

```
CV <- rbind(CV, CV2)

CV$Validation_population <- factor(CV$Validation_population, levels = c("ED", "DP"))
CV$Sibs <- factor(CV$Sibs, levels = c("Two shared parents", "One shared parent"))
```

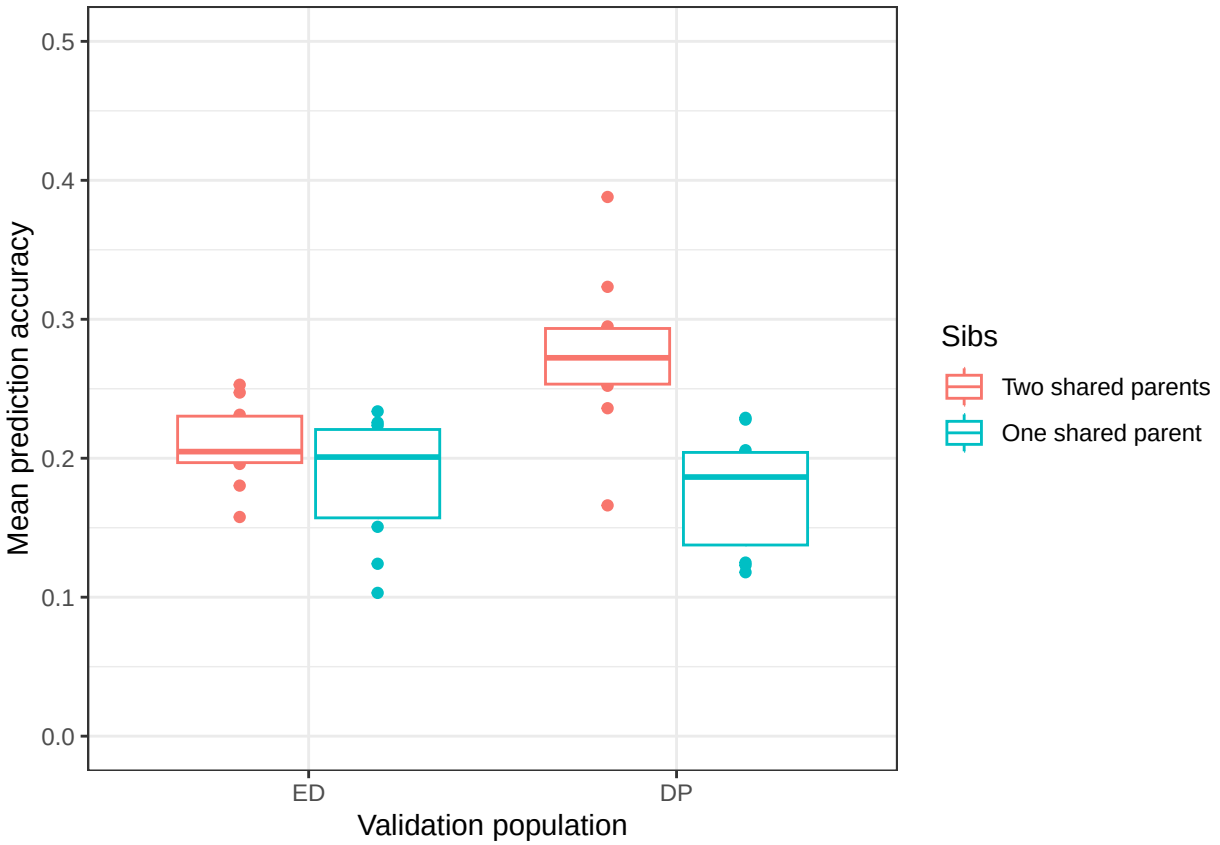
Plots of prediction accuracy for TP having two shared parents compared to one shared parent.

```
library(ggplot2)
library(scales)
library(dplyr)

ggplot(CV,aes(x=Validation_population,y=PA, color = Sibs)) + geom_boxplot(coef=0) +
  theme_bw() +
  #facet_wrap(~Method)+
  scale_y_continuous(
    breaks = seq(-0.2, 0.6, 0.1),
    limits = c(-0.2, 0.6),
    labels = number_format(accuracy = 0.1)
  ) + ylab("Prediction accuracy") + xlab("Validation population")
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```




```
# Plot mean accuracy for each iteration

CV_mean <- CV %>%
  group_by(Iteration, Training_population, Validation_population,Sibs) %>%
  summarise(Average_PA = mean(PA, na.rm = TRUE), .groups = 'drop')

ggplot(CV_mean,aes(x=Validation_population,y=Average_PA, color =Sibs)) +
  geom_boxplot(coef=0) +
  #facet_wrap(~Method)+
  theme_bw() + scale_y_continuous(
    breaks = seq(0, 0.5, 0.1),
    limits = c(0, 0.5),
    labels = number_format(accuracy = 0.1)
  ) + ylab("Mean prediction accuracy") + xlab("Validation population")
```




```
ggsave("Figure5.png",device="png",dpi=600,height=5,width=6)
```

Paired-t test to see whether the differences between using siblings sharing one parents and siblings sharing two parents are significant.

```
library(correctR)

##One shared parent vs two shared parents PA of the ED validation population

CV_ED <- CV[CV$Validation_population == "ED",]

DF_ED <- CV_ED[,c(3:6)]

colnames(DF_ED)[3] <- "r"
colnames(DF_ED)[2] <- "values"
colnames(DF_ED)[1] <- "model"
colnames(DF_ED)[4] <- "k"

DF_ED$r <- as.numeric(DF_ED$r)
DF_ED$k <- as.numeric(DF_ED$k)

repkfold_ttest(data = DF_ED, n1 = 132, n2=34, k=5, r=10,tailed = "one",
               greater = "Two shared parents")
```

```
##   statistic   p.value
## 1 0.2364555 0.4070325
```

There are no significant difference between same folds of ED population predicted with siblings sharing two parents vs siblings sharing two parents.

```
##One shared parent vs two shared parents PA of the DP validation population

CV_DP <- CV[CV$Validation_population == "DP",]

DF_DP <- CV_DP[,c(3:6)]

colnames(DF_DP)[3] <- "r"
colnames(DF_DP)[2] <- "values"
colnames(DF_DP)[1] <- "model"
colnames(DF_DP)[4] <- "k"

DF_DP$r <- as.numeric(DF_DP$r)
DF_DP$k <- as.numeric(DF_DP$k)

repkfold_ttest(data = DF_DP, n1 = 132, n2=34, k=5, r=10,tailed = "one",
               greater = "Two shared parents")
```

```
##   statistic   p.value
## 1  1.013543 0.1578899
```

There are no significant difference between same folds of DP population predicted with siblings sharing two parents vs siblings sharing two parents.