Stat 892 - phenotype preparation

Malachy Campbell

11/15/2017

```
library(plyr)
library(reshape2)
library(gplots)
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
library(BGLR)
```

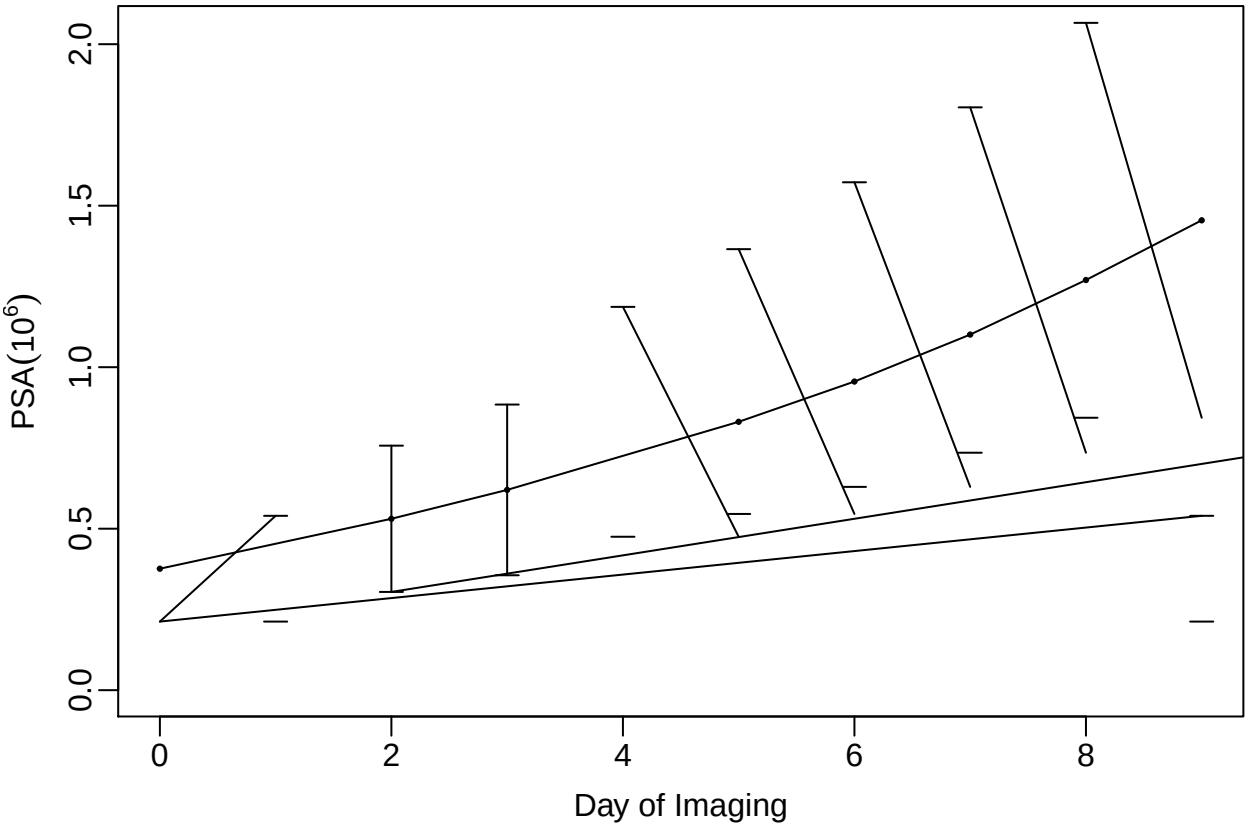
Introduction

Aus 2014 PSA data

Here, this data set that was collected using a high throughput phenomics platform (Lemnatex Scanalyzer 3D). From the images, we've quantified the number of plant pixels and have summed these for each plant. This sum we refer to as the projected shoot area (PSA) and use this a measure of shoot growth. PSA recorded over a period of eight days for 359 rice lines. Plants that had abberant growth patterns were removed from this dataset. The data consists of three independent experiments (Exp), each experiment has 357 lines (NSFTV.ID). In each experiment a subset of < 100 lines were randomly selected and replicated twice (Rep). Thus, for the three experiments there will be some lines that have six replicates. The plants were randomly assigned to positions on the conveyor belts in two smart houses (this is an automated greenhouse).

```
PSA=read.csv("~/Desktop/Stat892/Phenotypes/Aus2014_PSA.csv")
#Get the mean PSA at each time point
ddPSA=ddply(PSA, .(DayOfImaging), summarise, Mean=mean(PSA/100000, na.rm=T), SD=sd(PSA/100000, na.rm=T))


par(mar=c(3,3,1,.2), mgp=c(1.8,0.5,0))
plot(ddPSA$DayOfImaging, ddPSA$Mean, pch=19, cex=0.3, ylab=expression(PSA (10^6)), xlab="Day of Imaging")
lines(ddPSA$DayOfImaging, ddPSA$Mean, col="black")
segments(ddPSA$DayOfImaging, ddPSA$Mean - ddPSA$SD, 1:10, ddPSA$Mean + ddPSA$SD, lwd=1)
segments(1:10 - 0.1, ddPSA$Mean - ddPSA$SD, 1:10 + 0.1, ddPSA$Mean - ddPSA$SD, lwd=1)
segments(1:10 - 0.1, ddPSA$Mean + ddPSA$SD, 1:10 + 0.1, ddPSA$Mean + ddPSA$SD, lwd=1)
```



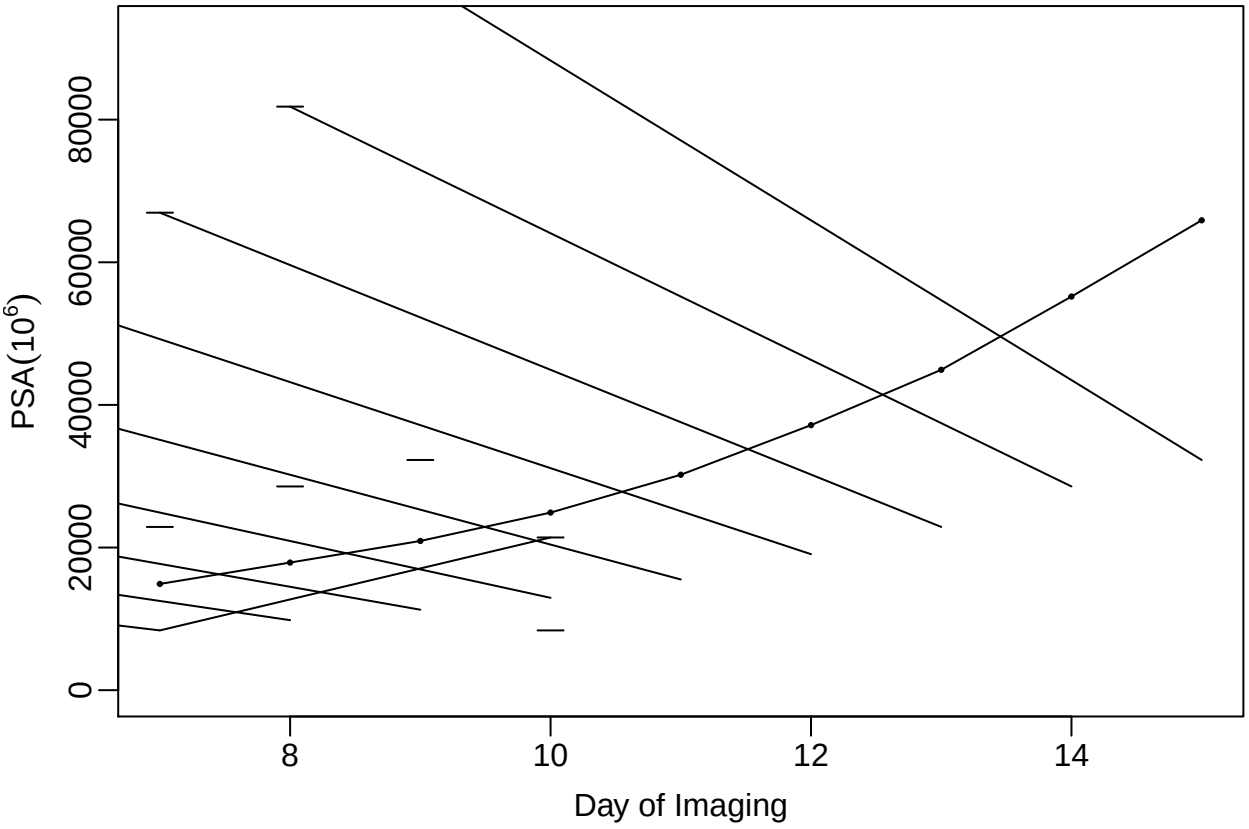
Aus 2013 PSA data

This dataset follows the same design as that described above. Here 361 rice lines are phenotyped over a period of ten days. The developmental period here is early than that above.

```
PSA=read.csv("~/Desktop/Stat892/Phenotypes/Aus2013_PSA.csv")

ddPSA=ddply(PSA, .(DayOfImaging), summarise, Mean=mean(PSA, na.rm=T), SD=sd(PSA, na.rm=T))


par(mar=c(3,3,1,.2), mgp=c(1.8,0.5,0))
plot(ddPSA$DayOfImaging, ddPSA$Mean, pch=19, cex=0.3, ylab=expression(PSA (10^6)), xlab="Day of Imaging
lines(ddPSA$DayOfImaging, ddPSA$Mean, col="black")
segments(ddPSA$DayOfImaging, ddPSA$Mean - ddPSA$SD, 1:10, ddPSA$Mean + ddPSA$SD, lwd=1)
segments(1:10 - 0.1, ddPSA$Mean - ddPSA$SD, 1:10 + 0.1, ddPSA$Mean - ddPSA$SD, lwd=1)
segments(1:10 - 0.1, ddPSA$Mean + ddPSA$SD, 1:10 + 0.1, ddPSA$Mean + ddPSA$SD, lwd=1)
```



Prep genotype data

These 357 lines are part of the Rice Diveristy Panel 1 that was developed by Susan McCouch at Cornell (Zhao et al 2011). These lines have been genotyped with a 44k SNP chip. Here I will load the SNP data in PLINK format, and generate the genomic relationship matrix.

```
FAM=read.table("~/Desktop/Stat892/Genotypes/sativas413.fam")[1:2]
MAP=read.table("~/Desktop/Stat892/Genotypes/sativas413.map")

setwd("~/Desktop/Stat892/Genotypes/")
PED = read_ped("sativas413.ped")
m=PED$p
n=PED$n
PED=PED$x

##SNPs in PED are coded as 0, 1, 2, 3. 2 is missing data. 1 are heterozygous, 0 and 3 are homozygous fo
PED[PED == 2] <- NA
PED[PED==0]=0
PED[PED==1]=1
PED[PED==3]=2


W = t(matrix(PED, nrow=m, ncol=n, byrow = T))
colnames(W)=MAP$V2

rownames(W) <- FAM$V2
```

```
for (j in 1:ncol(W)) {
  W[, j] = ifelse(is.na(W[, j]), mean(W[, j], na.rm = TRUE), W[, j])
}

W.orig=W


W=W.orig[row.names(W.orig) %in% PSA$NSFTV.ID ,]
freq = colMeans(W) / 2
maf = ifelse(freq > 0.5, 1-freq, freq)
maf.index = which(maf < 0.05)
length(maf.index)
```

```
## [1] 3245
```

```
W = W[, -maf.index]

####Estimate GRM using VanRaden's method
##NOTE that in the standalone of asreml the inverse of G is
##done after loading. So DO NOT take the inverse of G here!!!
Zsc = scale(x=W,center=T,scale=T)
G = tcrossprod(Zsc)/ncol(W)
G = G + diag(nrow(W))*0.001
G=G[match(unique(PSA$NSFTV.ID), row.names(G) ) ,]
G=G[, match(unique(PSA$NSFTV.ID), colnames(G) )]

##This chunk of code writes the GRM in a format that ASREML likes.
G.final <- as.data.frame(which(row(G)>=col(G),arr.ind=TRUE))
G.final$G <- G[lower.tri(G,diag=T)]
G.final <- G.final[,c(2,1,3)]
G.final <- G.final[order(G.final[,2],G.final[,1]),]
G.final <- G.final[,c(2,1,3)]
colnames(G.final)[1:2]=c("Row", "Column")
attr(G.final, "rowNames") = row.names(G)

write.table(G.final, "~/Desktop/Stat892/ASREML/G2.grm",col.names=F,row.names=F,quote=F,sep="\t")
```