1 Setup

1.1 Loading data

Set the working dir

```r
setwd("/mnt/picea/projects/arabidopsis/okeech/arabidopsis-fsg")
```

Librarius

```r
suppressPackageStartupMessages(library(DESeq)) # Normalisation
suppressPackageStartupMessages(library(vsn))
suppressPackageStartupMessages(library(gplots))
suppressPackageStartupMessages(library(RColorBrewer)) # Large color range
suppressPackageStartupMessages(library(tximport)) # Read Kallisto files
suppressPackageStartupMessages(library(scatterplot3d))
```

Define a palette (8 colors)

```r
pal <- brewer.pal(8,"Dark2")
```

Save the default margin parameters

```r
mar=par("mar")
```

source a few helper scripts
source("~/Git/UPSCb/src/R/featureSelection.R")
source("~/Git/UPSCb/src/R/plot.multidensity.R")

Read the sample file tab delimited

```r
samples <- read.delim("~/Git/UPSCb/projects/arabidopsis-fsg/doc/531vsRpn10vsCol_RNAseq.txt")
samples531 <- samples [-c(28:42),]
```

Load dataset

```r
countsRpn12a <- read.csv("analysis/kallisto/raw-unormalised-gene-expression_data.csv", header=T, row.names=1)
length(rowSums(countsRpn12a)) #32833 genes

## [1] 32833

length(which(rowSums(countsRpn12a) == 0)) #5783 genes without counts

## [1] 5783

geneswithcount <- which(rowSums(countsRpn12a) != 0) #genes with expression
write.csv(geneswithcount,file="analysis/kallisto/genes_with_counts531.csv")
```

The cumulative transcript coverage is deep, about 1000X

```r
countsRpn12a <- as.matrix(countsRpn12a)
plot(density(log10(rowMeans(countsRpn12a))),col=pal[1],
    main="gene mean raw counts distribution",
    xlab="mean raw counts (log10)")
```
The same is done for the individual samples colored by treatment. The samples are extremely similar. Light and IDL show a similar distribution for all samples.

```r
samples531 <- within(samples531, {
  Line <- as.factor(Line)
  Treatment <- as.factor(Treatment)
  Hours <- as.factor(Hours)
})

plot.multidensity(lapply(1:ncol(countsRpn12a), function(k) {log10(countsRpn12a)[,k]}),
  col=pal[as.integer(samples531$Treatment)],
  legend.x="topright",
  legend=levels(samples531$Treatment),
  legend.col=pal[1:nlevels(samples531$Treatment)],
  legend.lwd=2,
  main="sample raw counts distribution",
  xlab="per gene raw counts (log10)"
)
```
And colored by line

```r
# Genotypes show a similar distribution for all samples.
plot.multidensity(lapply(1:ncol(countsRpn12a), function(k) log10(countsRpn12a[,k])),
col=pal[as.integer(samples531$Line)],
legend.x="topright",
legend=levels(samples531$Line),
legend.col=pal[1:nlevels(samples531$Line)],
legend.lwd=2,
main="sample raw counts distribution",
xlab="per gene raw counts (log10)"
)```
1.2 Create the DESeq object

First, reorder the Line and Treatment factors

```r
samples531$Line <- relevel(samples531$Line, "Col_0")
samples531$Treatment <- relevel(samples531$Treatment, "Light")
```

```r
dds531 <- DESeqDataSetFromMatrix(countsRpn12a, samples531, ~Line*Hours)
```

Attach sample information

```r
save(dds531, file = "analysis/kallisto/DESeq-object-LineByHours_rpn12a.rda")
```

Estimating the size factor There is little difference in the size factor (0.8-1.3).

```r
dds531 <- estimateSizeFactors(dds531)
boxplot(colData(dds531)$sizeFactor, main = "Library size factor", ylab = "proportion")
abline(h = 1, lty = 2)
```
1.3 Variance Stabilising Transformation

Since there is almost no difference in library size, a VST is perfectly applicable. This is the situation prior to normalisation

```r
meanSdPlot(log2(counts(dds531)[rowSums(counts(dds531)) > 0,]))
```

```
## Warning: Removed 10217 rows containing non-finite values (stat_binhex).
```
Normalisation - blind, we give no prior as we want to assess quality (heteroscedastic)

```r
vst531 <- varianceStabilizingTransformation(dds531, blind=TRUE)
```

Extract the normalised counts

```r
vsd531 <- assay(vst531)
```

Look at the VST fit. It looks ok, around 0.5 sd on average (variation 0.1 to 0.7). Visualize the corrected mean - sd relationship. It is fairly linear, meaning we can assume homoscedasticity. The slight initial trend / bump is due to genes having few counts in a few subset of the samples and hence having a higher variability. This is expected.

```r
meanSdPlot(vsd531[rowSums(counts(dds531)) > 0,])
```
The VST introduces an offset

```r
grep('range(vsd531)')
```

```
## [1] 1.692855 19.078862
```

Which we remove, so that 0 means no expression

```r
vsd531 <- vsd531 - min(vsd531)
write.csv(vsd531, "analysis/kallisto/vst-blind-normalized-gene-expression_data-531.csv")
```

Transform to z-score

```r
vstz531 <- t(scale(t(vsd531)))
write.csv(vstz531,"analysis/kallisto/z-score_transformation_531.csv")
```

2 Quality Assessment

2.1 Principal Component Analysis
# PC1=49%, PC2=13%, PC3=10%, PC4=5%

```r
pc <- prcomp(t(vsd531))
percent <- round(summary(pc)$importance[2,]*100)
```

## 2.2 Two dimensions plots

### Coloring Genotypes

```r
plot(pc$x[,1],
     pc$x[,2],
     xlab=paste("Comp. 1 (",percent[1],")\%",sep=""),
     ylab=paste("Comp. 2 (",percent[2],")\%",sep=""),
     pch=c(19,17)[as.integer(samples531$Treatment)],
     col=pal[as.integer(samples531$Line)])
legend("topleft",pch=19,
       col=pal[1:nlevels(as.factor(samples531$Line))],
       legend=levels(as.factor(samples531$Line)))
legend("topright",pch=as.numeric(unique(samples531$Treatment)),
       legend=levels(factor(samples531$Treatment)))
```
plot(pc$x[,1],
    pc$x[,3],
    xlab=paste("Comp. 1 (",percent[1],")",sep=""),
    ylab=paste("Comp. 3 (",percent[3],")",sep=""),
    pch=c(19,17)[as.integer(samples531$Treatment)],
    col=pal[as.integer(samples531$Line)])
legend("topleft",pch=19,
    col=pal[1:nlevels(as.factor(samples531$Line))],
    legend=levels(as.factor(samples531$Line)))
legend("topright",pch=as.numeric(unique(samples531$Treatment)),
    legend=levels(factor(samples531$Treatment)))

plot(pc$x[,2],
    pc$x[,3],
    xlab=paste("Comp. 2 (",percent[2],")",sep=""),
    ylab=paste("Comp. 3 (",percent[3],")",sep=""),
    pch=c(19,17)[as.integer(samples531$Treatment)],
    col=pal[as.integer(samples531$Line)])
legend("topleft",pch=19,
    col=pal[1:nlevels(as.factor(samples531$Line))],
    legend=levels(as.factor(samples531$Line)))
legend("topright",pch=as.numeric(unique(samples531$Treatment)),
    legend=levels(factor(samples531$Treatment)))
Coloring Hours (instead of Treatments)

```r
f <- relevel(samples531$Hours, "6h")
samples531$Hours <- relevel(f, "0d")

plot(pc$x[,1],
     pc$x[,2],
     xlab=paste("Comp. 1 (", percent[1], ",%", sep=""),
     ylab=paste("Comp. 2 (", percent[2], ",%", sep=""),
     pch=c(19, 3, 17)[as.integer(samples531$Line)],
     col=pal[as.integer(samples531$Hours)])
legend("topleft", pch=19,
       col=pal[1:nlevels(as.factor(samples531$Hours))],
       legend=levels(as.factor(samples531$Hours)))
legend("topright", pch=as.numeric(unique(samples531$Line)),
       legend=levels(factor(samples531$Line)))
text(pc$x[,1],
     pc$x[,2],
     labels=samples$Rep, cex=.5, adj=-.3)
```
2.3 Expressed genes

```r
sels <- sapply(1:10, function(i) {
    featureSelect(vsd531, conditions = factor(paste0(samples531$Line, samples531$Treatment, samples531$Hours)), exp=i))
```

A cutoff at 2 seems reasonable.

```r
plot(colSums(sels), type="l", xlab="vst cutoff", main="number of genes selected at cutoff", ylab="number of genes")
```
sel <- sels[,2]

2.3.1 Hierarchical clustering of the data

plot(hclust(dist(t(vsd531[sel,]))),
     labels=paste(samples531$Line,samples531$Treatment,
                samples531$Hours,sep="_"))
2.3.2 Heatmap

Create a heatmap

```r
hpal <- colorRampPalette(c("blue","white","red"))(100)
```

z scale for gene counts (difference between mean expression and sample expression)

```r
s.vst <- t(scale(t(vsd531)))
```

```r
library(hyperSpec)
```

### Loading required package: lattice

### Loading required package: grid

### Loading required package: ggplot2

### Loading required package: xml2
## Package hyperSpec, version 0.100.0
##
## To get started, try
## vignette("hyperspec")
## package?hyperSpec
## vignette(package = "hyperSpec")
##
## If you use this package please cite it appropriately.
## citation("hyperSpec")
## will give you the correct reference.
##
## The project homepage is http://hyperspec.r-forge.r-project.org

## Attaching package: 'hyperSpec'

## The following object is masked from 'package:IRanges':
##
## collapse

heatmap.2(s.vst[sel,], distfun = pearson.dist, hclustfun = function(X){hclust(X,method="ward.D")}, trace="none", col=hpal, labRow = FALSE, labCol=paste(samples531$Line,samples531$Treatment, samples531$Hours, sep="_"), cexCol = 0.8, margins=c(7.1,0.1))
3 Session Info
| Package         | Version       |
|----------------|--------------|
| crayon          | 1.4.2        |
| annotate        | 1.72.0       |
| knitr           | 1.36         |
| XML             | 3.99-0.8     |
| latticeExtra    | 0.6-29       |
| vctrs           | 0.3.8        |
| purrr           | 0.3.4        |
| xfun            | 0.28         |
| tibble          | 3.1.6        |
| ellipsis        | 0.3.2        |
| Biostrings      | 2.62.0       |
| KEGGREST        | 1.34.0       |
| pillar          | 1.6.4        |
| glue            | 1.5.0        |
| BiocManager     | 1.30.16      |
| testthat        | 3.1.0        |
| assertthat      | 0.2.1        |
| xtable          | 1.8-4        |
| AnnotationDbi   | 1.56.1       |
| splines         | 4.1.1        |
| locfit          | 1.5-9.4      |
| geneplotter     | 1.72.0       |
| evaluate        | 0.14         |
| png             | 0.1-7        |
| gtable          | 0.3.0        |
| cachem          | 1.0.6        |
| survival        | 3.2-13       |
| memoise         | 2.0.0        |