E-GEOD-57452

Seoyeon Oh and Tobias Heyman

3 december 2021

```
library(affy)
library(arrayQualityMetrics)
library(ArrayExpress)
library(limma)
library(siggenes)
```

E-GEOD-57452

General info

The array used for this dataset is A-AFFY-130 - Affymetrix GeneChip Mouse Gene 1.0 ST Array [MoGene-1_0-st-v1]. Mice were infected with influenza and RNA was extracted from the lungs after 10 days. We used samples involving susceptible mice after 10 days of infection with influenza from this dataset.

Intensity values

Read in the microarray data and display the head and dimensions of the intensity value matrix.

```
# make vector containing the sample class
samples <- c(replicate(3, "control"), "day0", "day1", "day2", replicate(2, "day3"), replicate(2, "day4")

# add the sample classes to the pData object
pData(data.raw_1)[,2] <- samples

colnames(pData(data.raw_1)) <- c("index", "treatment")


# filter control samples and samples taken after 3 days of infection
filter <- colnames(data.raw_1)[data.raw_1@phenoData@data$treatment=="control" | data.raw_1@phenoData@da

# apply filter
filtered <- data.raw_1[,filter]

# check dimentsions of filtered object
dim(exprs(filtered))
```

```
## [1] 1102500       6
```

```
## arrayQualityMetrics
arrayQualityMetrics(filtered,outdir="../Datasets/microarray1/raw",force=T)
```

```
## The report will be written into directory '../Datasets/microarray1/raw'.
```

```
## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute
```

```
## name(s): subscripts, group.number, group.value

## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute
## name(s): subscripts, group.number, group.value

## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute
## name(s): subscripts, group.number, group.value

## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute
## name(s): subscripts, group.number, group.value

## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute
## name(s): subscripts, group.number, group.value

## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute
## name(s): subscripts, group.number, group.value

## Warning in KernSmooth::bkde2D(x, gridsize = nbin, bandwidth = bandwidth):
## Binning grid too coarse for current (small) bandwidth: consider increasing
## 'gridsize'

## Warning in KernSmooth::bkde2D(x, gridsize = nbin, bandwidth = bandwidth):
## Binning grid too coarse for current (small) bandwidth: consider increasing
## 'gridsize'

## Warning in KernSmooth::bkde2D(x, gridsize = nbin, bandwidth = bandwidth):
## Binning grid too coarse for current (small) bandwidth: consider increasing
## 'gridsize'

## Warning in KernSmooth::bkde2D(x, gridsize = nbin, bandwidth = bandwidth):
## Binning grid too coarse for current (small) bandwidth: consider increasing
## 'gridsize'

## Warning in KernSmooth::bkde2D(x, gridsize = nbin, bandwidth = bandwidth):
## Binning grid too coarse for current (small) bandwidth: consider increasing
## 'gridsize'

## Warning in KernSmooth::bkde2D(x, gridsize = nbin, bandwidth = bandwidth):
## Binning grid too coarse for current (small) bandwidth: consider increasing
## 'gridsize'
```

```r
arrayQualityMetrics(filtered,outdir="../Datasets/microarray1/rawlog",force=T,do.logtransform=T)
```

```
## The report will be written into directory '../Datasets/microarray1/rawlog'.

## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute
## name(s): subscripts, group.number, group.value

## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute
## name(s): subscripts, group.number, group.value

## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute
## name(s): subscripts, group.number, group.value

## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute
## name(s): subscripts, group.number, group.value
```

```
## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute
## name(s): subscripts, group.number, group.value

## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute
## name(s): subscripts, group.number, group.value
```

```
# Preprocessing (using the oligo function because affy didnt work)
MouseRMA<- oligo::rma(filtered,background=T)
```

```
## Background correcting
## Normalizing
## Calculating Expression
```

```
## QC post preprocessing
arrayQualityMetrics(MouseRMA,outdir="../Datasets/microarray1/rma",force=T)              #RMA produces l
```

```
## The report will be written into directory '../Datasets/microarray1/rma'.
```

```
## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute
## name(s): subscripts, group.number, group.value

## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute
## name(s): subscripts, group.number, group.value

## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute
## name(s): subscripts, group.number, group.value

## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute
## name(s): subscripts, group.number, group.value

## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute
## name(s): subscripts, group.number, group.value

## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute
## name(s): subscripts, group.number, group.value
```

Data exploration

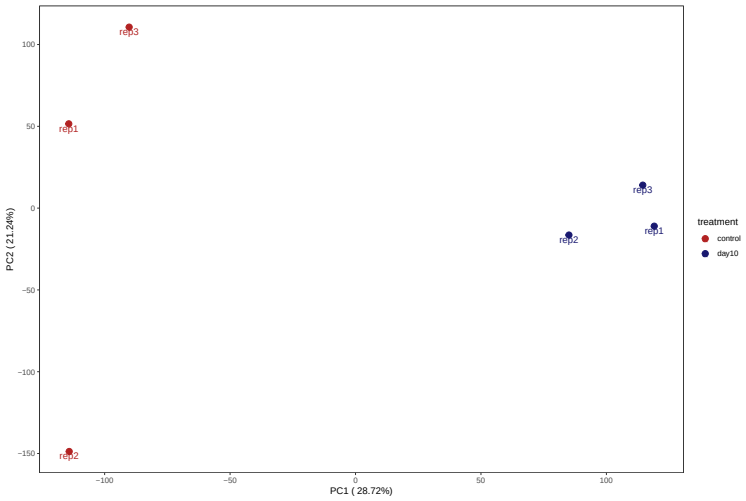
```
# transpose the data before PcA as this function requires the variables to b columns
data <- t(as.data.frame(MouseRMA@assayData$exprs))
pca <- prcomp(data, center = T, scale. = T)

summary(pca)
```

```
## Importance of components:
##                             PC1     PC2     PC3     PC4      PC5       PC6
## Standard deviation     117.2632 86.7380 83.3066 63.5689 57.45316 6.781e-13
## Proportion of Variance   0.3867  0.2116  0.1952  0.1137  0.09284 0.000e+00
## Cumulative Proportion    0.3867  0.5983  0.7935  0.9072  1.00000 1.000e+00
```

```
# save as dataframe and add treatment variable
pca_out <- as.data.frame(pca$x)
pca_out$treatment <- as.character(MouseRMA@phenoData@data$treatment)
pca_out$sample <- c("rep1", "rep2", "rep3", "rep1", "rep2", "rep3")
```

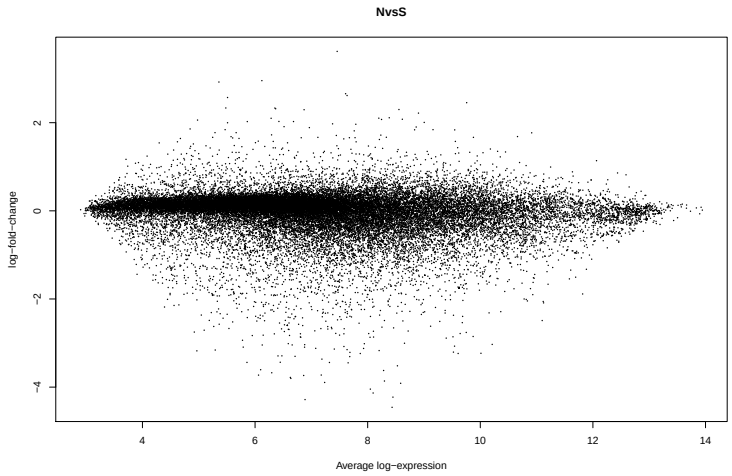
```r
# get lablels
percentage <- round(pca$sdev / sum(pca$sdev) * 100, 2)
percentage <- paste( colnames(pca_out), "(", paste( as.character(percentage), "%", ")", sep="") )

ggplot(data = pca_out)+
  geom_point(aes(x = PC1, y = PC2, colour = treatment, label=sample), size=3)+
  geom_text(aes(x = PC1, y = PC2, colour = treatment, label=sample),hjust=0.5, vjust=1.15)+
  theme_bw()+
  xlab(percentage[1])+
  ylab(percentage[2])+
  labs(colour = "treatment")+
  theme(plot.title = element_text(hjust = 0.5))+
  scale_colour_manual(values = c("firebrick", "midnightblue"))+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())
```

```
## Warning: Ignoring unknown aesthetics: label
```

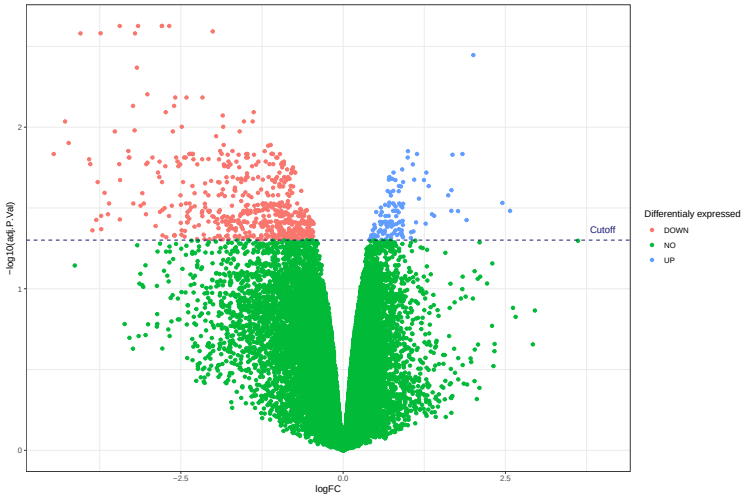


determine differential expression

```r
annot <- factor(pData(MouseRMA)[,2])

## Differential expression by LIMMA
# Method as stated in limma package (no intercept, easy for simple model designs)
design <- model.matrix(~0+annot)
colnames(design)<-c("control","infected")

# make linear model
fit <- lmFit(MouseRMA,design)

# create contrast matrix to get the differential expression between samples from infected mice and unin
cont.matrix <- makeContrasts(NvsS=control-infected,levels=design)
fit2 <- contrasts.fit(fit,cont.matrix)
fit2 <- eBayes(fit2)

# make MA plot for model with applied contrast matrix
limma::plotMA(fit2)
```




```
library(ggplot2)
# DE results with multiple testing correction (Benjamini-Hochberg = BH)
LIMMAout <- topTable(fit2,adjust="BH",number=nrow(exprs(MouseRMA)))



# add column indicating for all differntially expressed genes (adjusted p-value < 0.05) whether they're

LIMMAout$diffexpressed <- "NO"
LIMMAout$diffexpressed[LIMMAout$logFC > 0 & LIMMAout$adj.P.Val < 0.05] <- "UP"
LIMMAout$diffexpressed[LIMMAout$logFC < 0 & LIMMAout$adj.P.Val < 0.05] <- "DOWN"

# do the same but if we would not correct for multiple testing.
LIMMAout$diffexpressed_no_BH <- "NO"
LIMMAout$diffexpressed_no_BH[LIMMAout$logFC > 0 & LIMMAout$P.Value < 0.05] <- "UP"
LIMMAout$diffexpressed_no_BH[LIMMAout$logFC < 0 & LIMMAout$P.Value < 0.05] <- "DOWN"

# code to make volcano plots

ggplot(data = LIMMAout, aes(x= logFC, y = -log10(adj.P.Val), colour = diffexpressed)) +
 geom_point()+
 theme_bw()+
  geom_hline(yintercept = -log10(0.05), linetype="dashed", color="midnightblue")+
  annotate("text", min(4), 1.3, vjust = -1, label = "Cutoff", color="midnightblue")+
 labs(colour = "Differentialy expressed")
```




```
ggsave("volcanoplot.png", dpi=750)
```

```
## Saving 12 x 8 in image
```

```
ggplot(data = LIMMAout, aes(x= logFC, y = -log10(P.Value), colour = diffexpressed_no_BH)) +
 geom_point()+
 theme_bw()+
  geom_hline(yintercept = -log10(0.05), linetype="dashed", color="midnightblue")+
  annotate("text", min(4), 1.3, vjust = -1, label = "Cutoff", color="midnightblue")+
 labs(colour = "Differentialy expressed")
```


```
table(LIMMAout$diffexpressed)
```

```
##
##  DOWN    NO    UP
##   570 34863   123
```

Annotation

```
# get the annotation trough package (mogene10sttranscriptcluster.db) found at https://www.biostars.org/
columns(mogene10sttranscriptcluster.db)
```

```
##  [1] "ACCNUM"       "ALIAS"        "ENSEMBL"      "ENSEMBLPROT"  "ENSEMBLTRANS"
```

```
##  [6] "ENTREZID"     "ENZYME"       "EVIDENCE"     "EVIDENCEALL"  "GENENAME"
## [11] "GO"           "GOALL"        "IPI"          "MGI"          "ONTOLOGY"
## [16] "ONTOLOGYALL"  "PATH"         "PFAM"         "PMID"         "PROBEID"
## [21] "PROSITE"      "REFSEQ"       "SYMBOL"       "UNIGENE"      "UNIPROT"
```

```
annotTable <- select(
  mogene10sttranscriptcluster.db,
  keys = keys(mogene10sttranscriptcluster.db),
  column = c('PROBEID', 'SYMBOL', 'ENTREZID', 'ENSEMBL', 'GENENAME', 'PROSITE'),
  keytype = 'PROBEID')
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
## sort annotation data alphabetically on probe name

annotTable.filt <- annotTable[sort(annotTable$PROBEID,index.return=T)$ix,]

# merge information from multiple lines describing the same probe
probe <- "start"
position <- 0
for (i in 1:dim(annotTable.filt)[1]){
  if (annotTable.filt[i, 1] != probe){
    probe <- annotTable.filt[i, 1]
    position <- i
  }
  else{
    # concatenate the information o f the 2 lines with a ; as separator
    annotTable.filt[position,2:5] <- paste(annotTable.filt[position,2:5], annotTable.filt[i, 2:5], sep=
    # mark the line
    annotTable.filt[i,1] <- NA
  }
}

annotTable.filt <- annotTable.filt[!is.na(annotTable.filt$PROBEID),]

## Check if all probes are present in both sets
dim(annotTable.filt)
```

```
## [1] 35556     6
```

```
dim(LIMMAout)
```

```
## [1] 35556     8
```

```
 ## Double check => "Assumption is the mother of all fuck up's ;)"
sum(annotTable.filt$PROBEID!=sort(rownames(LIMMAout)))
```

```
## [1] 0
```

```
## Sort LIMMA output alphabetically on probe name
LIMMAout_sorted <- LIMMAout[sort(rownames(LIMMAout),index.return=T)$ix,]

## Add gene names to LIMMA output
LIMMAout_sorted$gene <- annotTable.filt$SYMBOL
LIMMAout_annot <- LIMMAout_sorted[sort(LIMMAout_sorted$adj.P.Val,index.return=T)$ix,]


# determine how many differentially expressed probes have an annotated gene
```

```r
table(is.na(LIMMAout_annot[LIMMAout_annot$diffexpressed != "NO",9]))
```

```
##
## FALSE  TRUE
##   512   181
```