Annex 5: DESeq2 DE analysis

Juan Manuel Medina Méndez

```r
# Data reading
setwd("~/ProjectX")

# Library
library("DESeq2")
library("ggplot2")
library("pheatmap")
library("RColorBrewer")

# Data reading
matrix_plus <- read.table("EM.plus.tab", h = F, stringsAsFactors = F)
matrix_minus <- read.table("EM.minus.tab", h = F, stringsAsFactors = F)

# Removal of the SIZE column
matrix_plus <- matrix_plus[, -2]
matrix_minus <- matrix_minus[, -2]
matrix_control <- data.frame(matrix_plus[, c(1,2,3,4,5)], matrix_minus[, c(2,3,4,5)])
matrix_kd <- data.frame(matrix_plus[, c(1,6,7,8,9)], matrix_minus[, c(6,7,8,9)])

# Column renaming
names(matrix_plus) <- c("POSITION", "C0+", "C1+", "C2+", "C3+", "KD0+", "KD1+", "KD2+", "KD3+")
names(matrix_minus) <- c("POSITION", "C0-", "C1-", "C2-", "C3-", "KD0-", "KD1-", "KD2-", "KD3-")
names(matrix_control) <- c("POSITION", "C0+", "C1+", "C2+", "C3+", "C0-", "C1-", "C2-", "C3-")
names(matrix_kd) <- c("POSITION", "KD0+", "KD1+", "KD2+", "KD3+", "KD0-", "KD1-", "KD2-", "KD3-")

## MATRIX PLUS: CONTROL (+) VS KD (+)
## MATRIX MINUS: CONTROL (-) VS KD (-)
## MATRIX CONTROL: CONTROL (+) VS CONTROL (-)
## MATRIX KD: KD (+) VS KD (-)

# Positions as rownames
cts_matrix_plus <- matrix_plus[, -1]
rownames(cts_matrix_plus) <- matrix_plus[, 1]
cts_matrix_minus <- matrix_minus[, -1]
rownames(cts_matrix_minus) <- matrix_minus[, 1]

cts_matrix_control <- matrix_control[, -1]
rownames(cts_matrix_control) <- matrix_control[, 1]
cts_matrix_kd <- matrix_kd[, -1]
rownames(cts_matrix_kd) <- matrix_kd[, 1]

# Metadata
metadata_plus <- colnames(cts_matrix_plus)
condition_plus <- c(rep("control", 4), rep("kd", 4))
metadata_minus <- colnames(cts_matrix_minus)
condition_minus <- c(rep("control", 4), rep("kd", 4))

metadata_control <- colnames(cts_matrix_control)
condition_control <- c(rep("control_plus", 4), rep("control_minus", 4))
```

```r
metadata_kd <- colnames(cts_matrix_kd)
condition_kd <- c(rep("kd_plus", 4), rep("kd_minus", 4))

# Design of the METADATA DFs
coldata_plus <- data.frame(colnames(cts_matrix_plus), metadata_plus, condition_plus)
coldata_minus <- data.frame(colnames(cts_matrix_minus), metadata_minus, condition_minus)
coldata_control <- data.frame(colnames(cts_matrix_control), metadata_control, condition_control)
coldata_kd <- data.frame(colnames(cts_matrix_kd), metadata_kd, condition_kd)

# Samples as rownames
rownames(coldata_plus) <- coldata_plus[, 1]
coldata_plus <- coldata_plus[, -1]
rownames(coldata_minus) <- coldata_minus[, 1]
coldata_minus <- coldata_minus[, -1]

rownames(coldata_control) <- coldata_control[, 1]
coldata_control <- coldata_control[, -1]
rownames(coldata_kd) <- coldata_kd[, 1]
coldata_kd <- coldata_kd[, -1]

# Sanity check: columns of cts and rows of coldata have to be the SAME and be in the SAME ORDER
all(colnames(cts_matrix_plus) %in% rownames(coldata_plus))
all(colnames(cts_matrix_plus) == rownames(coldata_plus))
all(colnames(cts_matrix_minus) %in% rownames(coldata_minus))
all(colnames(cts_matrix_minus) == rownames(coldata_minus))
all(colnames(cts_matrix_control) %in% rownames(coldata_control))
all(colnames(cts_matrix_control) == rownames(coldata_control))
all(colnames(cts_matrix_kd) %in% rownames(coldata_kd))
all(colnames(cts_matrix_kd) == rownames(coldata_kd))

# Rounding: no floats accepted, only integers
cts_matrix_plus <- round(cts_matrix_plus)
cts_matrix_minus <- round(cts_matrix_minus)
cts_matrix_control <- round(cts_matrix_control)
cts_matrix_kd <- round(cts_matrix_kd)

# Construction of DESeqDataSet objects:
dds_plus <- DESeqDataSetFromMatrix(countData = cts_matrix_plus,
                                   colData = coldata_plus,
                                   design = ~ condition_plus)

dds_minus <- DESeqDataSetFromMatrix(countData = cts_matrix_minus,
                                    colData = coldata_minus,
                                    design = ~ condition_minus)

dds_control <- DESeqDataSetFromMatrix(countData = cts_matrix_control,
                                      colData = coldata_control,
                                      design = ~ condition_control)

dds_kd <- DESeqDataSetFromMatrix(countData = cts_matrix_kd,
                                 colData = coldata_kd,
                                 design = ~ condition_kd)
```

```
# Pre-filtering: remove rows in which there are no tags
dds_plus <- dds_plus[rowSums(counts(dds_plus)) >= 1, ]
dds_minus <- dds_minus[rowSums(counts(dds_minus)) >= 1, ]
dds_control <- dds_control[rowSums(counts(dds_control)) >= 1, ]
dds_kd <- dds_kd[rowSums(counts(dds_kd)) >= 1, ]

# Set factor levels (which level represents the control group and the exosome kd one,
# or the plus/minus)
dds_plus$condition_plus <- factor(dds_plus$condition_plus, levels = c("control", "kd"))
dds_minus$condition_minus <- factor(dds_minus$condition_minus, levels = c("control", "kd"))
dds_control$condition_control <- factor(dds_control$condition_control,
                                        levels = c("control_plus","control_minus"))
dds_kd$condition_kd <- factor(dds_kd$condition_kd, levels = c("kd_plus", "kd_minus"))

## DE ANALYSIS ##
dds_plus <- DESeq(dds_plus)
dds_minus <- DESeq(dds_minus)
dds_control <- DESeq(dds_control)
dds_kd <- DESeq(dds_kd)

## RESULTS (DEFAULT ALPHA = 0.1, here is changed to 0.05) ##
res_plus <- results(dds_plus, alpha = 0.05)
res_minus <- results(dds_minus, alpha = 0.05)
res_control <- results(dds_control, alpha = 0.05)
res_kd <- results(dds_kd, alpha = 0.05)
res_plus; res_minus; res_control; res_kd

# Information of each column of the results
mcols(res_plus)$description

# Summarization of the results
summary(res_plus); summary(res_minus); summary(res_control); summary(res_kd)

# Number of significantly expressed windows (adjusted p-values less than 0.05)
sum(res_plus$padj < 0.05, na.rm = TRUE)
sum(res_minus$padj < 0.05, na.rm = TRUE)
sum(res_control$padj < 0.05, na.rm = TRUE)
sum(res_kd$padj < 0.05, na.rm = TRUE)


# Number of total windows
nrow(dds_plus); nrow(dds_minus); nrow(dds_control); nrow(dds_kd)


## DATA EXPLORATION ##

# MA PLOT
plotMA(res_plus, ylim = c(-2, 2))
```

```
plotMA(res_minus, ylim = c(-2, 2))
```

```
plotMA(res_control, ylim = c(-2, 2))
plotMA(res_kd, ylim = c(-2, 2))
```

```
# Customized implementations of the MA plot
plot_res_plus <- res_plus
plot_res_plus$significant <- as.factor((plot_res_plus$padj < .05))
plot_res_plus$significant[is.na(plot_res_plus$significant)] = F
plot_res_minus <- res_minus
plot_res_minus$significant <- as.factor((plot_res_minus$padj < .05))
plot_res_minus$significant[is.na(plot_res_minus$significant)] = F

plot_res_control <- res_control
plot_res_control$significant <- as.factor((plot_res_control$padj < .05))
plot_res_control$significant[is.na(plot_res_control$significant)] = F
plot_res_kd <- res_kd
plot_res_kd$significant <- as.factor((plot_res_kd$padj < .05))
plot_res_kd$significant[is.na(plot_res_kd$significant)] = F

ggplot(as.data.frame(plot_res_plus), aes(x = log2(baseMean), y = log2FoldChange, color = significant)) +
  geom_point() +
  geom_hline(color = "blue3", yintercept = 0) +
  stat_smooth(se = FALSE, method = "loess", color = "red3") +
  scale_color_manual(values = c("Black", "Red"))
```

```
ggplot(as.data.frame(plot_res_minus), aes(x = log2(baseMean), y = log2FoldChange, color = significant))
  geom_point() +
  geom_hline(color = "blue3", yintercept = 0) +
  stat_smooth(se = FALSE, method = "loess", color = "red3") +
  scale_color_manual(values = c("Black", "Red"))
```

```
ggplot(as.data.frame(plot_res_control), aes(x = log2(baseMean), y = log2FoldChange, color = significant
  geom_point() +
  geom_hline(color = "blue3", yintercept = 0) +
  stat_smooth(se = FALSE, method = "loess", color = "red3") +
  scale_color_manual(values = c("Black", "Red"))
```

```
ggplot(as.data.frame(plot_res_kd), aes(x = log2(baseMean), y = log2FoldChange, color = significant)) +
  geom_point() +
  geom_hline(color = "blue3", yintercept = 0) +
  stat_smooth(se = FALSE, method = "loess", color = "red3") +
  scale_color_manual(values = c("Black", "Red"))
```

```
# MA PLOT (of shrunken log2 FC, obtained with a bayesian prior assumption that removes the noise
# associated with log2 FC from low count windows without requiring arbitrary filtering thresholds)
res_plusLFC <- lfcShrink(dds_plus, coef = 2, res = res_plus)
plotMA(res_plusLFC, ylim = c(-2, 2))
```

```
res_minusLFC <- lfcShrink(dds_minus, coef = 2, res = res_minus)
plotMA(res_minusLFC, ylim = c(-2, 2))
```

```
res_controlLFC <- lfcShrink(dds_control, coef = 2, res = res_control)
plotMA(res_controlLFC, ylim = c(-2, 2))
```

```
res_kdLFC <- lfcShrink(dds_kd, coef = 2, res = res_kd)
plotMA(res_kdLFC, ylim = c(-2, 2))
```

```r
# Customized implementations of the shrunken MA plots
plot_res_LFC_plus <- res_plusLFC
plot_res_LFC_plus$significant = as.factor((plot_res_LFC_plus$padj < .05))
plot_res_LFC_plus$significant[is.na(plot_res_LFC_plus$significant)] = F
plot_res_LFC_minus <- res_minusLFC
plot_res_LFC_minus$significant = as.factor((plot_res_LFC_minus$padj < .05))
plot_res_LFC_minus$significant[is.na(plot_res_LFC_minus$significant)] = F

plot_res_LFC_control <- res_controlLFC
plot_res_LFC_control$significant = as.factor((plot_res_LFC_control$padj < .05))
plot_res_LFC_control$significant[is.na(plot_res_LFC_control$significant)] = F
plot_res_LFC_kd <- res_kdLFC
plot_res_LFC_kd$significant = as.factor((plot_res_LFC_kd$padj < .05))
plot_res_LFC_kd$significant[is.na(plot_res_LFC_kd$significant)] = F

ggplot(as.data.frame(plot_res_LFC_plus), aes(x = log2(baseMean), y = log2FoldChange,
                                             color = significant)) +
  geom_point() +
  geom_hline(color = "blue3", yintercept = 0) +
  stat_smooth(se = FALSE, method = "loess", color = "red3") +
  scale_color_manual(values=c("Black","Red")) +
  ggtitle("MA plot of shrunken LFC \n (control & KD: plus strand)")
```

```r
ggplot(as.data.frame(plot_res_LFC_minus), aes(x = log2(baseMean), y = log2FoldChange,
                                              color = significant)) +
  geom_point() +
  geom_hline(color = "blue3", yintercept = 0) +
  stat_smooth(se = FALSE, method = "loess", color = "red3") +
  scale_color_manual(values=c("Black","Red")) +
  ggtitle("MA plot of shrunken LFC \n (control & KD: minus strand)")
```

```r
ggplot(as.data.frame(plot_res_LFC_control), aes(x = log2(baseMean), y = log2FoldChange,
                                                color = significant)) +
  geom_point() +
  geom_hline(color = "blue3", yintercept = 0) +
  stat_smooth(se = FALSE, method = "loess", color = "red3") +
  scale_color_manual(values=c("Black","Red")) +
  ggtitle("MA plot of shrunken LFC \n (controls of both strands)")
```

```r
ggplot(as.data.frame(plot_res_LFC_kd), aes(x = log2(baseMean), y = log2FoldChange,
                                           color = significant)) +
  geom_point() +
  geom_hline(color = "blue3", yintercept = 0) +
  stat_smooth(se = FALSE, method = "loess", color = "red3") +
  scale_color_manual(values=c("Black","Red")) +
  ggtitle("MA plot of shrunken LFC \n (KDs of both strands)")
```

```r
# PLOT COUNTS
# Count of reads for one single window across the groups, normalizing counts by sequencing depth
# and adding a pseudocount
plotCounts(dds_plus, gene = which.min(res_plus$padj), intgroup = "condition_plus")
```

```r
d_plus <- plotCounts(dds_plus, gene = which.min(res_plus$padj),
                     intgroup = "condition_plus", returnData = TRUE)
```

```
ggplot(d_plus, aes(x = condition_plus, y = count)) +
  geom_point(position = position_jitter(w = 0.1, h = 0)) +
  scale_y_log10(breaks = c(25, 100, 400, 1600))
```

```
# Regularized log transformation to remove the dependence of the variance on the mean, particularly
# the high variance of the logarithm of count data when the mean is low
rld_plus <- rlog(dds_plus, blind = FALSE)
rld_minus <- rlog(dds_minus, blind = FALSE)
rld_control <- rlog(dds_control, blind = FALSE)
rld_kd <- rlog(dds_kd, blind = FALSE)

# Variance stabilizing transformation
vsd_plus <- varianceStabilizingTransformation(dds_plus, blind = FALSE)
vsd_minus <- varianceStabilizingTransformation(dds_minus, blind = FALSE)
vsd_control <- varianceStabilizingTransformation(dds_control, blind = FALSE)
vsd_kd <- varianceStabilizingTransformation(dds_kd, blind = FALSE)

# HIERARCHICAL CLUSTERING: overview over similarities and dissimilarities between samples

# Create distance matrices
sampleDists_plus <- dist(t(assay(rld_plus)))
sampleDistMatrix_plus <- as.matrix(sampleDists_plus)
rownames(sampleDistMatrix_plus) <- paste(rld_plus$condition_plus, rld_plus$type, sep = "+")
colnames(sampleDistMatrix_plus) <- NULL
sampleDists_minus <- dist(t(assay(rld_minus)))
sampleDistMatrix_minus <- as.matrix(sampleDists_minus)
rownames(sampleDistMatrix_minus) <- paste(rld_minus$condition_minus, rld_minus$type, sep = "-")
colnames(sampleDistMatrix_minus) <- NULL

sampleDists_control <- dist(t(assay(rld_control)))
sampleDistMatrix_control <- as.matrix(sampleDists_control)
rownames(sampleDistMatrix_control) <- paste(rld_control$condition_control, rld_control$type)
colnames(sampleDistMatrix_control) <- NULL
sampleDists_kd <- dist(t(assay(rld_kd)))
sampleDistMatrix_kd <- as.matrix(sampleDists_kd)
rownames(sampleDistMatrix_kd) <- paste(rld_kd$condition_kd, rld_kd$type)
colnames(sampleDistMatrix_kd) <- NULL

# Create colors (using the default ones)
colors <- colorRampPalette(rev(brewer.pal(9, "Blues")))(255)

pheatmap(sampleDistMatrix_plus,
         clustering_distance_rows = sampleDists_plus,
         clustering_distance_cols = sampleDists_plus,
         col = colors,
         main = "Heatmap of RLT count data \n (control & KD: plus strand)")
```

```
pheatmap(sampleDistMatrix_minus,
         clustering_distance_rows = sampleDists_minus,
         clustering_distance_cols = sampleDists_minus,
         col = colors,
         main = "Heatmap of RLT count data \n (control & KD: minus strand)")
```

```r
pheatmap(sampleDistMatrix_control,
         clustering_distance_rows = sampleDists_control,
         clustering_distance_cols = sampleDists_control,
         col = colors,
         main = "Heatmap of RLT count data \n (controls of both strands)")
```

```r
pheatmap(sampleDistMatrix_kd,
         clustering_distance_rows = sampleDists_kd,
         clustering_distance_cols = sampleDists_kd,
         col = colors,
         main = "Heatmap of RLT count data \n (KDs of both strands)")
```

```r
# Default PCAs of the regularized log transform
plotPCA(rld_plus, intgroup = c("condition_plus", "metadata_plus"))
```

```r
plotPCA(rld_minus, intgroup = c("condition_minus", "metadata_minus"))
```

```r
plotPCA(rld_control, intgroup = c("condition_control", "metadata_control"))
```

```r
plotPCA(rld_kd, intgroup = c("condition_kd", "metadata_kd"))
```

```r
# Customized PCAs
pcaData_plus <- plotPCA(rld_plus, intgroup = c("condition_plus", "metadata_plus"),
                        returnData = TRUE)
percentVar_plus <- round(100 * attr(pcaData_plus, "percentVar"))

pcaData_minus <- plotPCA(rld_minus, intgroup = c("condition_minus", "metadata_minus"),
                        returnData = TRUE)
percentVar_minus <- round(100 * attr(pcaData_minus, "percentVar"))

pcaData_control <- plotPCA(rld_control, intgroup = c("condition_control", "metadata_control"),
                           returnData = TRUE)
percentVar_control <- round(100 * attr(pcaData_control, "percentVar"))

pcaData_kd <- plotPCA(rld_kd, intgroup = c("condition_kd", "metadata_kd"),
                      returnData = TRUE)
percentVar_kd <- round(100 * attr(pcaData_kd, "percentVar"))

ggplot(pcaData_plus, aes(PC1, PC2, shape = condition_plus, color = metadata_plus)) +
  geom_point(size = 3) +
  xlab(paste0("PC1: ", percentVar_plus[1], "% variance")) +
  ylab(paste0("PC2: ", percentVar_plus[2], "% variance")) +
  coord_fixed() +
  ggtitle("PCA of RLT count data \n (control & KD: plus strand)") +
  theme(plot.title = element_text(size = 10), legend.text = element_text(size = 8))
```

```r
ggplot(pcaData_minus, aes(PC1, PC2, shape = condition_minus, color = metadata_minus)) +
  geom_point(size = 3) +
  xlab(paste0("PC1: ", percentVar_minus[1], "% variance")) +
  ylab(paste0("PC2: ", percentVar_minus[2], "% variance")) +
  coord_fixed() +
  ggtitle("PCA of RLT count data \n (control & KD: minus strand)") +
  theme(plot.title = element_text(size = 10), legend.text = element_text(size = 8))
```

```r
ggplot(pcaData_control, aes(PC1, PC2, shape = condition_control, color = metadata_control)) +
  geom_point(size = 3) +
  xlab(paste0("PC1: ", percentVar_control[1], "% variance")) +
  ylab(paste0("PC2: ", percentVar_control[2], "% variance")) +
  coord_fixed() +
  ggtitle("PCA of RLT count data \n (controls of both strands)") +
  theme(plot.title = element_text(size = 10), legend.text = element_text(size = 8))
```

```r
ggplot(pcaData_kd, aes(PC1, PC2, shape = condition_kd, color = metadata_kd)) +
  geom_point(size = 3) +
  xlab(paste0("PC1: ", percentVar_kd[1], "% variance")) +
  ylab(paste0("PC2: ", percentVar_kd[2], "% variance")) +
  coord_fixed() +
  ggtitle("PCA of RLT count data \n (KDs of both strands)") +
  theme(plot.title = element_text(size = 10), legend.text = element_text(size = 8))
```