Swiss Institute of Bioinformatics

Isabelle Dupanloup
BCF - Bioinformatics Core Facility

Day 2 hands-on: exploration of gene counts and normalization

Loading R packages

```
library(ggplot2)
library(gridExtra)
library(reshape2)
library(mixOmics)
library(RColorBrewer)
library(DESeq2)
library(edgeR)
library(devtools)

if necessary:
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

BiocManager::install("mixOmics")
```

Data description and importation

The dataset used in this practical session corresponds to 6 mouse samples at 2 different developmental stages: newborn and adult.
The dataset corresponds to 3 different files:

- Day2_dataset_genecounts.txt : contains the gene counts for each sample and for each gene
- Day2_dataset_genelength.txt : contains the gene length in kilobases
- Day2_dataset_infossamples.txt : contains the information relative to the samples

The files can be loaded in R using:

```
raw_counts <- read.table("Day2_dataset_genecounts.txt", header = TRUE,
                         row.names = 1)
raw_counts <- as.matrix(raw_counts)
gene_lengths <- scan("Day2_dataset_genelength.txt")
design <- read.table("Day2_dataset_infossamples.txt", header = TRUE)
```

Basic exploratory analysis of raw counts

The number of genes and samples can be obtained with:

```
dim(raw_counts)
```

We start the analysis by filtering out the genes for which no count has been found:

```
raw_counts_wn <- raw_counts[rowSums(raw_counts) > 0, ]
dim(raw_counts_wn)
```

It is often useful, to visualize the count distribution, to compute log-transformed counts:

This practical has been adapted from online tutorials developed by members of the teaching team at the Harvard Chan Bioinformatics Core (HBC): https://github.com/hbctraining/DGE_workshop

```
log_counts <- log2(raw_counts_wn + 1)
head(log_counts)

df_raw <- melt(log_counts, id = rownames(raw_counts_wn))
names(df_raw)[1:2]<- c("id", "sample")
df_raw$method <- rep("Raw counts", nrow(df_raw))
head(df_raw)
```

Count distribution

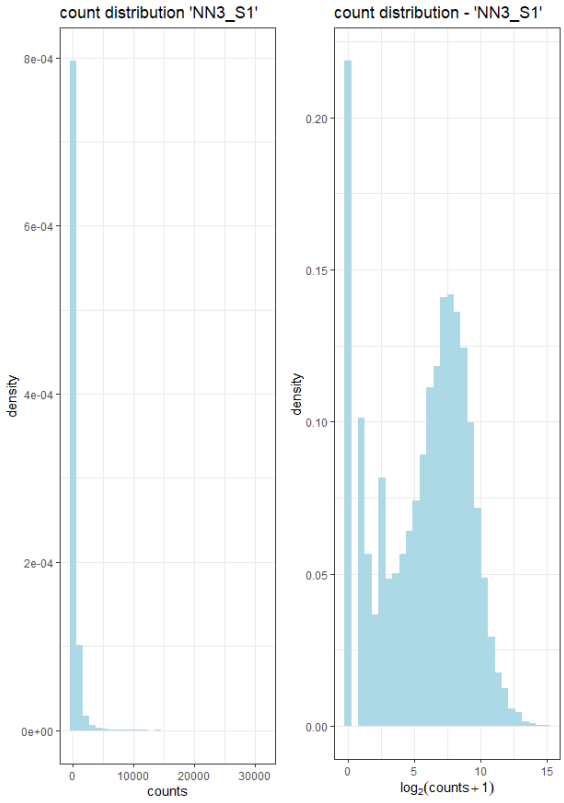
Let's have an overview of the distribution of the genes counts for the first sample by plotting the histograms of raw counts and log_counts:

```
df <- data.frame(rcounts = raw_counts_wn [ ,1], lcounts = log_counts[ ,1])

p1 <- ggplot(data=df, aes(x = rcounts, y = ..density..))
p1 <- p1 + geom_histogram(fill = "lightblue")
p1 <- p1 + theme_bw()
p1 <- p1 + ggtitle(paste0("count distribution '", design$labels[1], "'"))
p1 <- p1 + xlab("counts")

p2 <- ggplot(data=df, aes(x = lcounts, y = ..density..))
p2 <- p2 + geom_histogram(fill = "lightblue")
p2 <- p2 + theme_bw()
p2 <- p2 + ggtitle(paste0("count distribution - '", design$labels[1], "'"))
p2 <- p2 + xlab(expression(log[2](counts + 1)))

grid.arrange(p1, p2, ncol = 2)
```

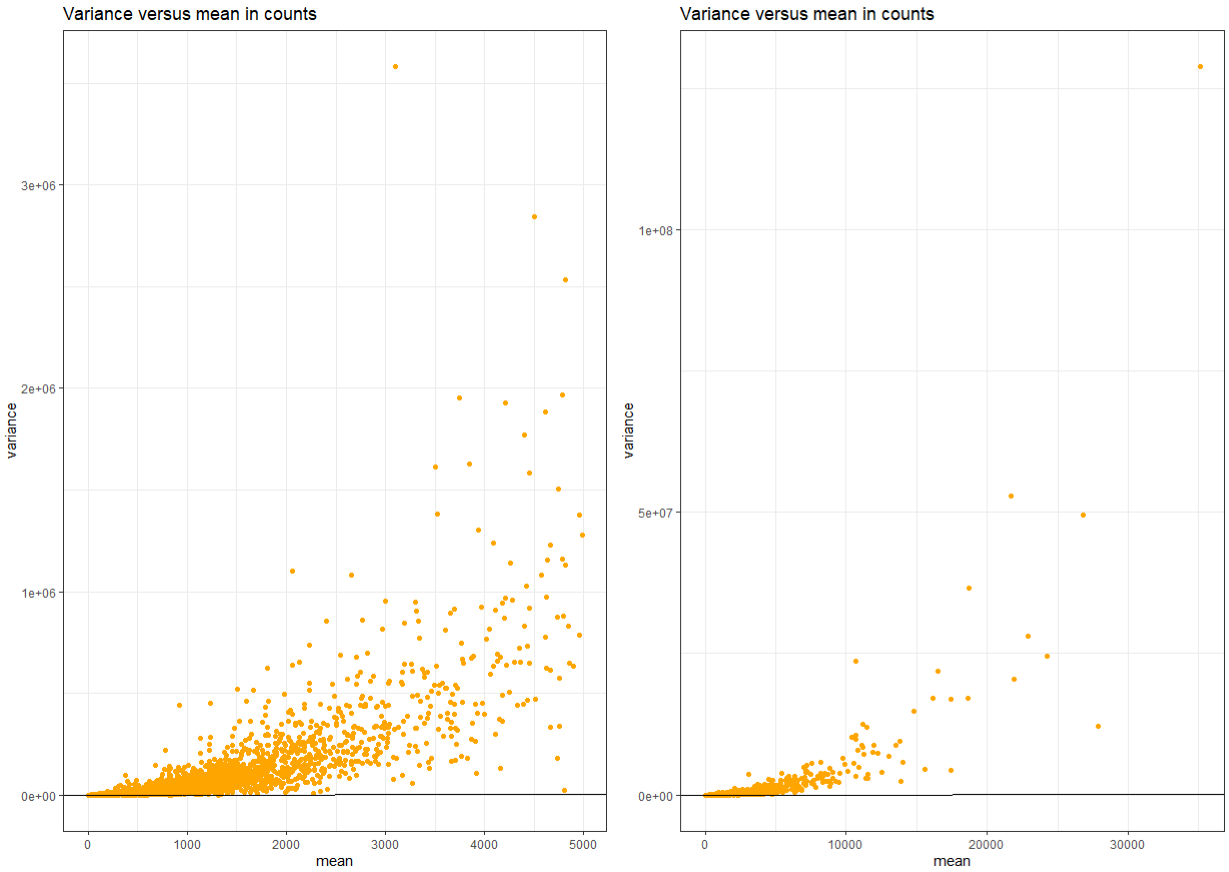


This figure shows that the count distribution is highly skewed with a large proportion of genes with a count equal to 0 and a few number of genes with a very large number of counts.

Relation between mean and variance

Another important feature of rna-seq data is the fact that they are overdispersed. This means that the variance of the count of a given gene over different biological samples within a given condition is larger than the average count for the same gene. This feature is illustrated by plotting the graphics of the mean vs variance for condition “newborns” for all genes with an average count smaller than 5000 (otherwise the chart cannot be read easily):

```
df <- data.frame(mean = apply(raw_counts_wn[ ,design$group == "newborns"], 1, mean),
                 var = apply(raw_counts_wn[ ,design$group == "newborns"], 1, var))
df <- df[df$mean <= 5000, ]
p <- ggplot(data=df, aes(x = mean, y = var))
p <- p + geom_point(colour = "orange")
p <- p + theme_bw()
p <- p + geom_abline(aes(intercept=0, slope=1))
p <- p + ggtitle("Variance versus mean in counts") + ylab("variance")
print(p)
```



In this figure, the black line is the "y=x" diagonal. It is easy to see that, for most genes, the variance is much larger than the mean.

Normalization

DESeq

```
groups <- factor(design$group)
dds <- DESeqDataSetFromMatrix(raw_counts_wn, DataFrame(groups), ~ groups)
dds

dds <- estimateSizeFactors(dds)
sizeFactors(dds)

deseq_normcount <- counts(dds, normalized = TRUE)

pseudo_deseq <- log2(deseq_normcount + 1)
df_deseq <- melt(pseudo_deseq, id = rownames(raw_counts_wn))
names(df_deseq)[1:2]<- c("id", "sample")
df_deseq$method <- rep("DESeq", nrow(df_raw))
```

edgeR

```
dge2 <- DGEList(raw_counts_wn)
dge2

dge2 <- calcNormFactors(dge2, method = "TMM")

pseudo_TMM <- log2(cpm(dge2) + 1)

df_TMM <- melt(pseudo_TMM, id = rownames(raw_counts_wn))
names(df_TMM)[1:2] <- c ("id", "sample")
df_TMM$method <- rep("TMM", nrow(df_TMM))
```

RPKM

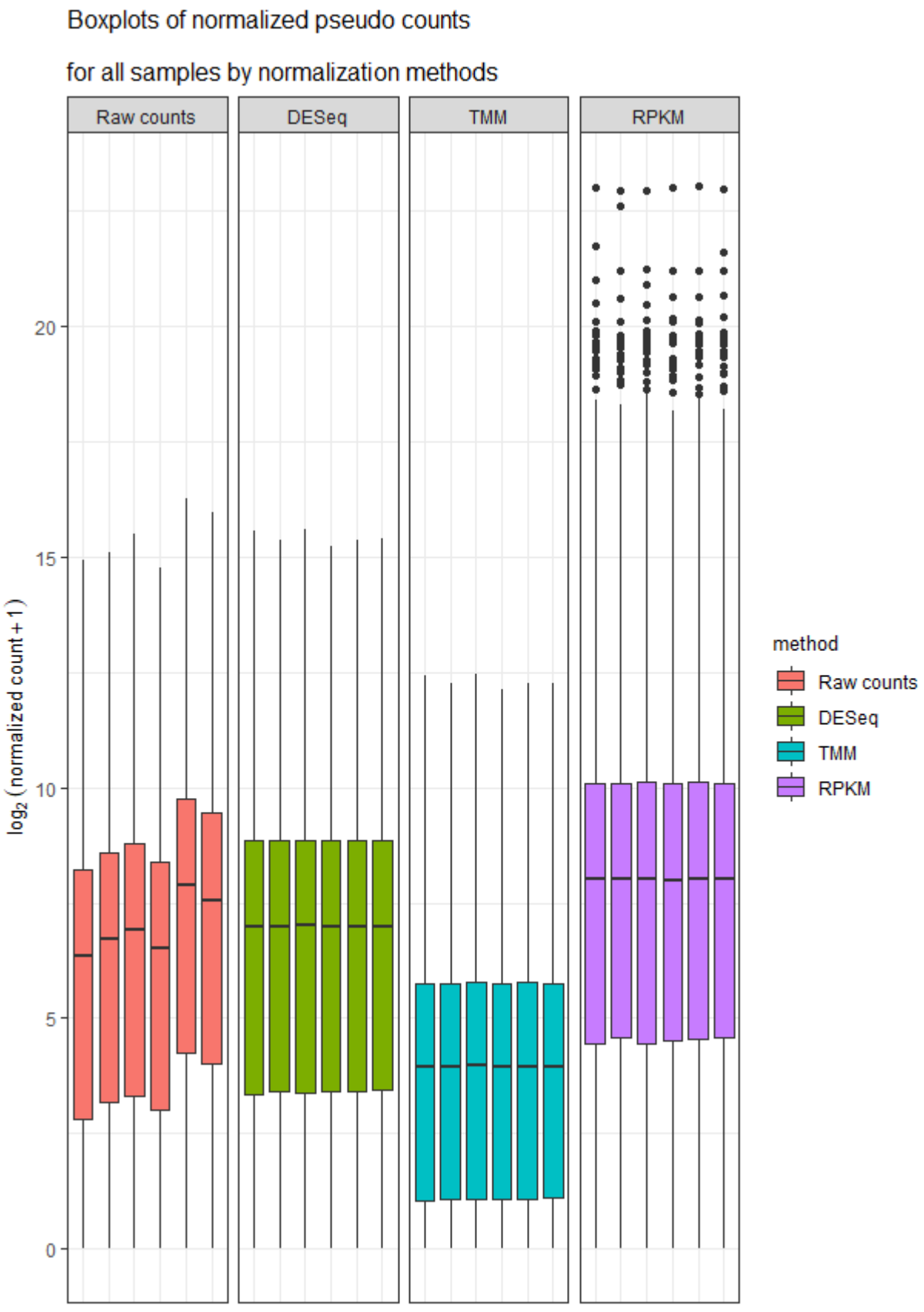
```
gene_lengths_wn <- gene_lengths[rowSums(raw_counts) > 0]
pseudo_RPKM <- log2(rpkm(dge2, gene.length = gene_lengths_wn) + 1)

df_RPKM <- melt(pseudo_RPKM, id = rownames(raw_counts_wn))
names(df_RPKM)[1:2] <- c ("id", "sample")
df_RPKM$method <- rep("RPKM", nrow(df_RPKM))
```

Comparison

```
df_allnorm <- rbind(df_raw, df_deseq, df_TMM, df_RPKM)
df_allnorm$method <- factor(df_allnorm$method,
                            levels = c("Raw counts", "DESeq", "TMM", "RPKM"))

p <- ggplot(data=df_allnorm, aes(x=sample, y=value, fill=method))
p <- p + geom_boxplot()
p <- p + theme_bw()
p <- p + ggtitle("Boxplots of normalized pseudo counts\n
for all samples by normalization methods")
p <- p + facet_grid(. ~ method)
p <- p + ylab(expression(log[2] ~ (normalized ~ count + 1))) + xlab("")
p <- p + theme(title = element_text(size=10), axis.text.x = element_blank(),
               axis.ticks.x = element_blank())
print(p)
```


Boxplots of normalized pseudo counts
for all samples by normalization methods
Raw counts
DESeq
TMM
RPKM
20
15
10
5
0
log2 (normalized count + 1)
method
Raw counts
DESeq
TMM
RPKM