Exploratory plots

Read counts to DGE, Part II

Friederike Dündar

February 25, 2020

Contents

Similarity assessments and clustering

One of the very first sanity checks one should do after obtaining **normalized read counts** is to see if the biological replicates are more similar to each other than samples from different conditions.

Attaching packages for handling the data sets and for plotting.

```
library(DESeq2)
library(magrittr)
library(ggplot2)
```

Loading the `DESeq2` object and the rlog-transformed values that we generated previously (see the first part of our "Counts to DGE" documents).

```
load("RNAseqGierlinski.RData")
ls()
```

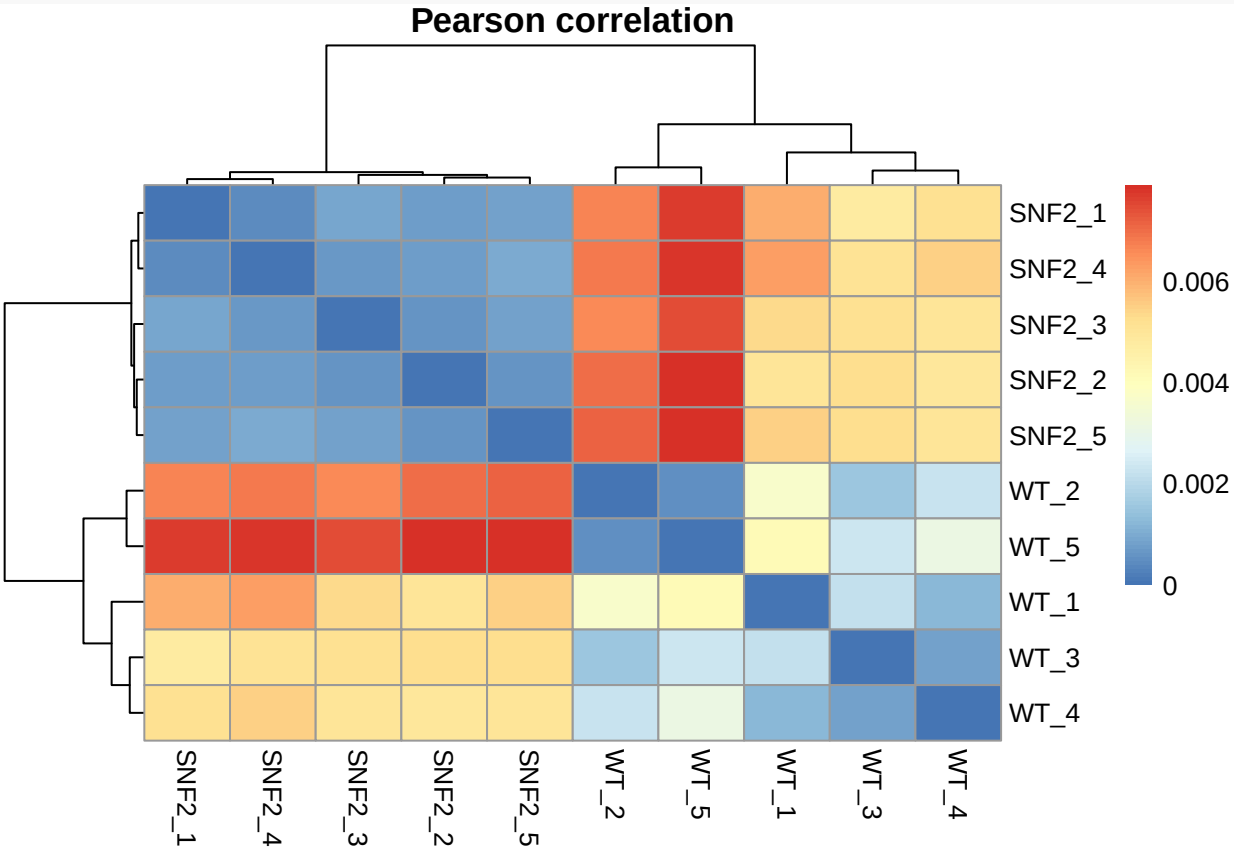
```
##  [1] "counts.sf_normalized" "def.chunk.hook"       "DESeq.ds"
##  [4] "DESeq.rlog"           "folder"               "keep_genes"
##  [7] "log.counts"           "log.norm.counts"      "msd_plot"
## [10] "orig_names"           "readcounts"           "rlog.norm.counts"
## [13] "sample_info"
```

Sample clustering using Pearson correlation

The ENCODE consortium recommends that *"for messenger RNA, (. . .) biological replicates [should] display greater than 0.9 correlation for transcripts/features"*.
The Pearson correlation coefficient is a measure of the strength of the linear relationship between two variables and is often used to assess the similarity of RNA-seq samples in a pair-wise fashion. It is defined as the **covariance of two variables divided by the product of their standard deviation**.

You can use the **pheatmap** package to generate a clustered heatmap of correlation coefficients:

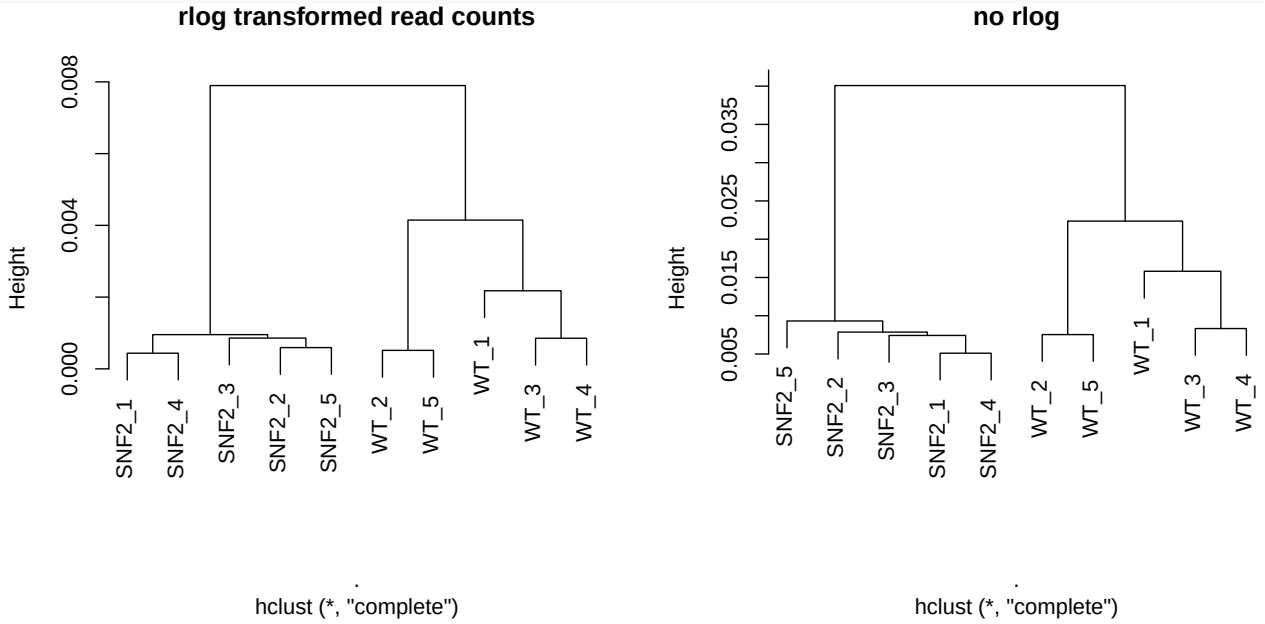
```
corr_coeff <- cor(rlog.norm.counts, method = "pearson")
as.dist(1-corr_coeff, upper = TRUE) %>%
  as.matrix %>%
  pheatmap::pheatmap(., main = "Pearson correlation")
```



Just plot the dendrogram, comparing the effects of the `rlog` transformation.

```
par(mfrow=c(1,2))
# Pearson corr. for rlog.norm values
as.dist(1 - corr_coeff) %>% hclust %>%
  plot( ., labels = colnames(rlog.norm.counts),
        main = "rlog transformed read counts")

# Pearson corr. for log.norm.values
as.dist( 1 - cor(log.norm.counts, method = "pearson")) %>%
  hclust %>% plot( ., labels = colnames(log.norm.counts),
                   main = "no rlog")
```

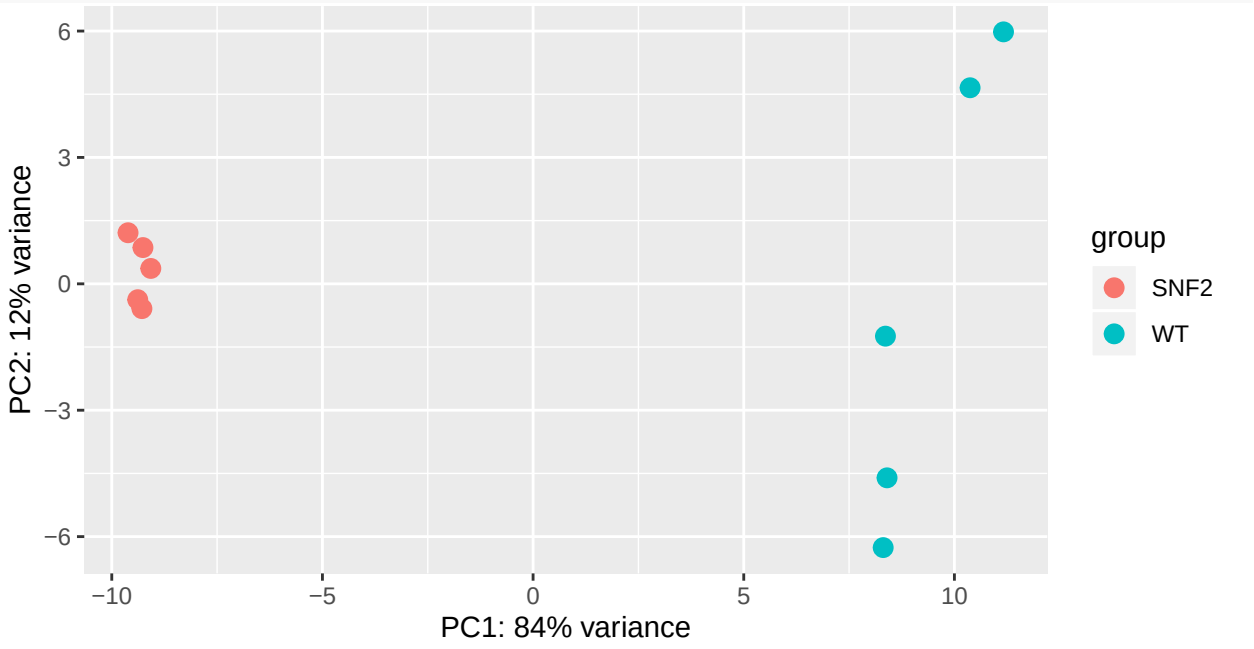


PCA

Principal Component Analysis is typically done on the most variably detected genes. Take note that the matrix has to be flipped via the `t()` function in order to determine the eigenvectors based on the gene expression values.

```
rv <- rowVars(assay(DESeq.rlog)) # equivalent to rowVars(rlog.norm.counts)
top_variable <- order(rv, decreasing = TRUE)[seq_len(500)]
pca <- prcomp(t(assay(DESeq.rlog)[top_variable, ]))
head(pca$x)
```

```
plotPCA(DESeq.rlog)
```



pcaExplorer

`pcaExplorer` lets you interact with the DESeq2-based plots and analyses. It has included hierarchical clustering of samples and PCA.

```
#BiocManager::install("pcaExplorer")
pcaExplorer::pcaExplorer(dds = DESeq.ds, rlt = DESeq.rlog)
```

Consult their vignette for more details.