RNA-Seq Analysis Pipeline

Daria Plewa

2025-02-03

```
library(DESeq2)
library(dplyr)
library(gplots)
library(RColorBrewer)
library(pheatmap)
library(knitr)
```

Preparing data

```
# Load raw count data from two different sequencing runs
data1 <- read.csv("count_matrix_se.txt", sep='\t', skip=1)
data2 <- read.csv("count_matrix.txt", sep='\t', skip=1)

# Combine relevant columns from the two datasets (genes x samples)
datann <- cbind(data1[, 7:10], data2[, 7:10])  # Keeping sample columns only

# Display the first few rows to inspect the structure
head(datann)
```

```
##   SRR3194428.bam SRR3194429.bam SRR3194430.bam SRR3194431.bam SRR3191542.bam
## 1              0              0              0              0              0
## 2              5             10             11              6             22
## 3              0              0              0              0              0
## 4              0              0              0              0              0
## 5              0              0              0              0              0
## 6              0              0              0              0              0
##   SRR3191543.bam SRR3191544.bam SRR3191545.bam
## 1              0              0              0
## 2              5              4             51
## 3              0              0              0
## 4              0              0              0
## 5              0              0              0
## 6              0              0              0
```

```
# Assign gene IDs as row names and define sample column names
rownames(datann) <- data1$Geneid
colnames(datann) <- c('SRR3191542', 'SRR3191543', 'SRR3191544', 'SRR3191545',
                      'SRR3194428', 'SRR3194429', 'SRR3194430', 'SRR3194431')
```

DESeq2

```
# Define sample conditions and sequencing platform information
samples <- colnames(datann)
condition <- factor(rep(c('Mock', 'Zika', 'Mock', 'Zika'), each = 2))
instrument <- factor(rep(c('MiSeq', 'NextSeq'), each = 4))

# Create metadata frame for DESeq2
colData <- data.frame(samples = samples, condition = condition, instrument = instrument)

# Create DESeq2 object using instrument as the design factor
dds <- DESeqDataSetFromMatrix(countData = datann, colData = colData, design = ~instrument)

# Normalize data using rlog transformation (logarithmic transformation to stabilize variance)
dds <- estimateSizeFactors(dds)
log_data <- rlog(dds)
norm_data <- assay(log_data)  # Extract normalized counts as a matrix
norm_data <- as.data.frame(norm_data)  # Convert to a data frame for easier manipulation

# Perform DE analysis and display a summary of the results
dds <- DESeq(dds)
res <- results(dds)
summary(res)
```

```
##
## out of 31648 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 788, 2.5%
## LFC < 0 (down)     : 597, 1.9%
## outliers [1]       : 37, 0.12%
## low counts [2]     : 15443, 49%
## (mean count < 4)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
# Combine normalized counts for samples from MiSeq and NextSeq platforms
Miseq <- c(norm_data$SRR3191542, norm_data$SRR3191543, norm_data$SRR3191544, norm_data$SRR3191545)
NextSeq <- c(norm_data$SRR3194428, norm_data$SRR3194429, norm_data$SRR3194430, norm_data$SRR3194431)
```

Comparison

```
# Perform correlation test to assess similarity between the two platforms
cor_result <- cor.test(Miseq, NextSeq)
cor_result
```

```
##
##  Pearson's product-moment correlation
##
## data:  Miseq and NextSeq
```

```
## t = 12347, df = 200022, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9993389 0.9993504
## sample estimates:
##       cor
## 0.9993447
```

```r
# Creating a table
table <- data.frame(
  Indicator = c("Correlation", "Significance (p)", "Interpretation"),
  Value = c("0.99", "< 2.2e-16", "Very high similarity")
)

# Generating the table
kable(table, format = "markdown", caption = "Summary of Correlation Results")
```

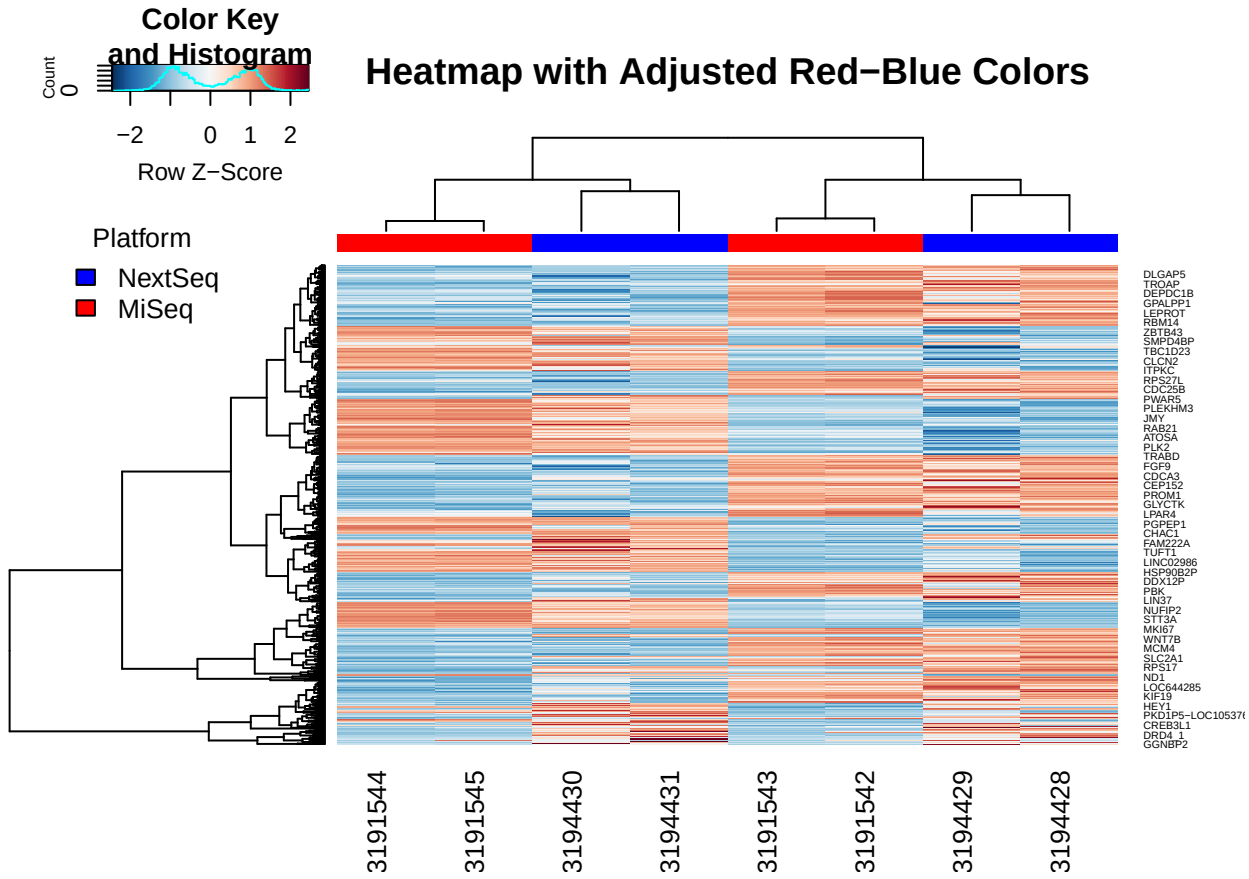
Table 1: Summary of Correlation Results

Indicator	Value
Correlation	0.99
Significance (p)	< 2.2e-16
Interpretation	Very high similarity

```
## Conclusion
The correlation returned a very high positive and significant result (0.99, p < 2.2e-16).
This indicates a very strong similarity between the sequences.
```

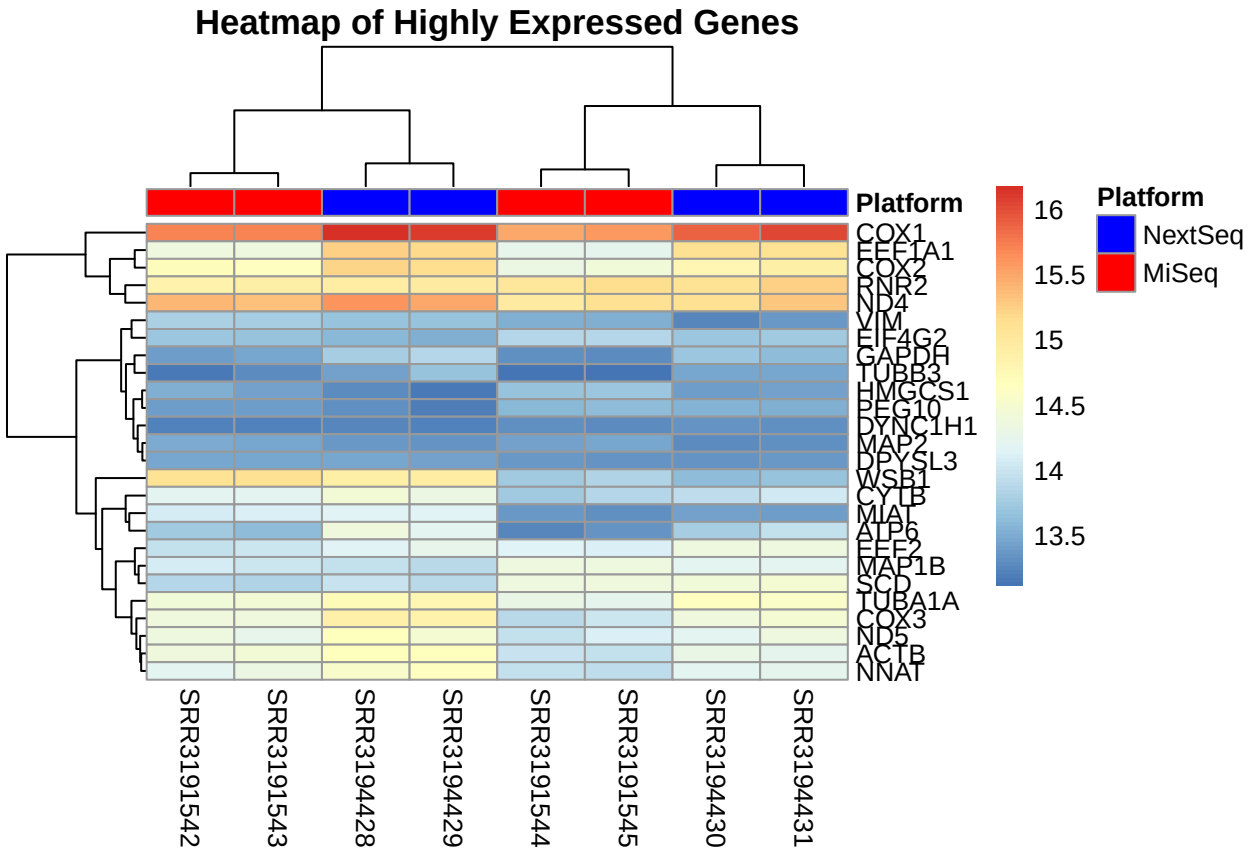
```r
# Define colors for annotation (legend)
ann_colors <- list(
  Platform = c("NextSeq" = "#0000FF", "MiSeq" = "#FF0000")  # Blue for NextSeq, Red for MiSeq
)

# Define a red-blue color scale with a smoother gradient
color <- brewer.pal(11, "RdBu")  # Use RdBu palette
morecols <- colorRampPalette(color)(100)  # Increase gradient resolution with 100 shades

# Assign red and blue colors based on the sequencing instrument used
col.inst <- ifelse(colData$instrument == "MiSeq", "#FF0000", "#0000FF")  # Red for MiSeq, Blue for Next

# Compute variance across genes and select the top 1000 most variable ones
countVar <- apply(norm_data, 1, var)
highVar <- order(countVar, decreasing = TRUE)[1:1000]
hmDat <- as.matrix(norm_data[highVar, ])  # Extract highly variable genes for heatmap

# Generate heatmap with improved color mapping
heatmap.2(hmDat, col = rev(morecols),  # Reverse RdBu scale for better visualization
          trace = "none",
          main = "Heatmap with Adjusted Red-Blue Colors",
          ColSideColors = col.inst,  # Add side color annotation for instrument types
          scale = "row")
```

```
# Add legend for color annotations (platform types)
legend(x = -0.1, y = 0.9, legend = names(ann_colors$Platform),
       fill = ann_colors$Platform, border = "black",
       title = "Platform", cex = 0.8, bty = "n", xpd = TRUE)
```




```
# Filter data: Keep only genes where all sample values are >= 13
filtered_data <- norm_data %>% filter_all(all_vars(. >= 13))
filtered_data <- as.matrix(filtered_data)

# Ensure annotation matches colData$instrument (correct assignment of colors)
annotation_col <- data.frame(
  Platform = factor(colData$instrument)  # Directly use instrument values
)
rownames(annotation_col) <- colnames(filtered_data)

# Define colors for annotation (ensure consistency with col.inst)
ann_colors <- list(
  Platform = c("NextSeq" = "#0000FF", "MiSeq" = "#FF0000")  # Blue for NextSeq, Red for MiSeq
)

# Assign colors to columns based on sequencing instrument (ensuring consistency)
col.inst <- ifelse(colData$instrument == "MiSeq", "#FF0000", "#0000FF")  # Red for MiSeq, Blue for Next
```

```r
# Generate heatmap with correctly matched annotation colors
pheatmap(filtered_data,
         cluster_rows = TRUE,
         cluster_cols = TRUE,
         show_rownames = TRUE,
         show_colnames = TRUE,
         main = "Heatmap of Highly Expressed Genes",
         annotation_col = annotation_col,  # Add the legend on top
         annotation_colors = ann_colors    # Apply consistent color scheme
)
```



Conclusion

Both sequencers produced very similar results.