DESeq2 Analysis Report

Code ▾

Lohit Marla

2025-02-11

Introduction

This report presents an RNA-seq differential expression analysis using DESeq2. Our objectives are to:

- Assess the impact of the treatment (indicated by the `dex` variable) on gene expression.
- Identify differentially expressed genes.
- Visualize the expression patterns of key genes.
- Explore overall sample variation using Principal Component Analysis (PCA).

The results described herein provide insight into the treatment effects on gene expression, and the workflow is designed for reproducibility and clarity.

Data Loading and Preparation

First, we load the necessary libraries and import our scaled count data and metadata. The count data represents gene expression measurements, while the metadata provides experimental conditions for each sample.

Differential Expression Analysis We construct a DESeq2 dataset object and run the DESeq analysis. The results include log2 fold changes, p-values, and adjusted p-values for each gene.

Hide

```
dds <- DESeqDataSetFromMatrix(countData = countData, colData = metaData, design = ~dex, tidy = TRUE)
```

```
converting counts to integer mode
Warning: some variables in design formula are characters, converting to factors
```

Hide

```
# Run the differential expression analysis
dds <- DESeq(dds)
```

```
estimating size factors
estimating dispersions
gene-wise dispersion estimates
mean-dispersion relationship
final dispersion estimates
fitting model and testing
```

Hide

```
# Extract results and view a summary
res <- results(dds)
head(results(dds, tidy = TRUE))
```

	row <chr>	baseMean <dbl>	log2FoldChange <dbl>	lfcSE <dbl>	stat <dbl>	pvalue <dbl>	pad <dbl
1	ENSG00000000003	747.1941954	-0.35070302	0.1682457	-2.0844697	0.03711747	0.1630348
2	ENSG00000000005	0.0000000	NA	NA	NA	NA	NA
3	ENSG00000000419	520.1341601	0.20610777	0.1010592	2.0394752	0.04140263	0.1760317
4	ENSG00000000457	322.6648439	0.02452695	0.1451451	0.1689823	0.86581056	0.9616942
5	ENSG00000000460	87.6826252	-0.14714205	0.2570073	-0.5725210	0.56696907	0.8158486
6	ENSG00000000938	0.3191666	-1.73228897	3.4936010	-0.4958463	0.62000288	NA

6 rows

Hide

```
summary(res)
```

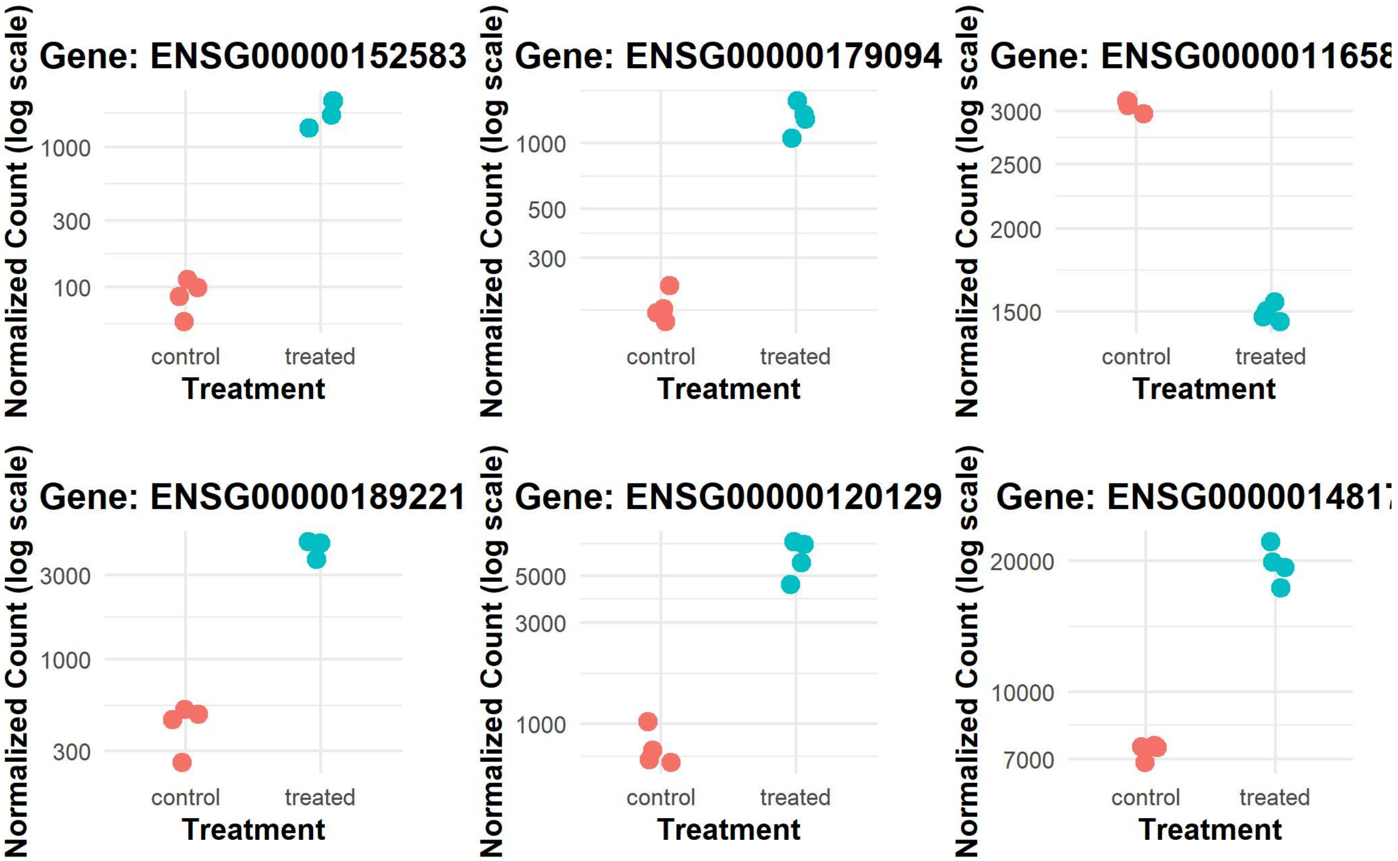
```

out of 25258 with nonzero total read count
adjusted p-value < 0.1
LFC > 0 (up)       : 1563, 6.2%
LFC < 0 (down)     : 1188, 4.7%
outliers [1]       : 142, 0.56%
low counts [2]     : 9971, 39%
(mean count < 10)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results
```

Visualization of Top Genes

The top six genes (with the smallest adjusted p-values) are selected for detailed visualization. Their normalized counts across treatment conditions are plotted on a log scale. We use ggplot2 for a polished appearance and gridExtra to arrange the plots in a grid.

Hide

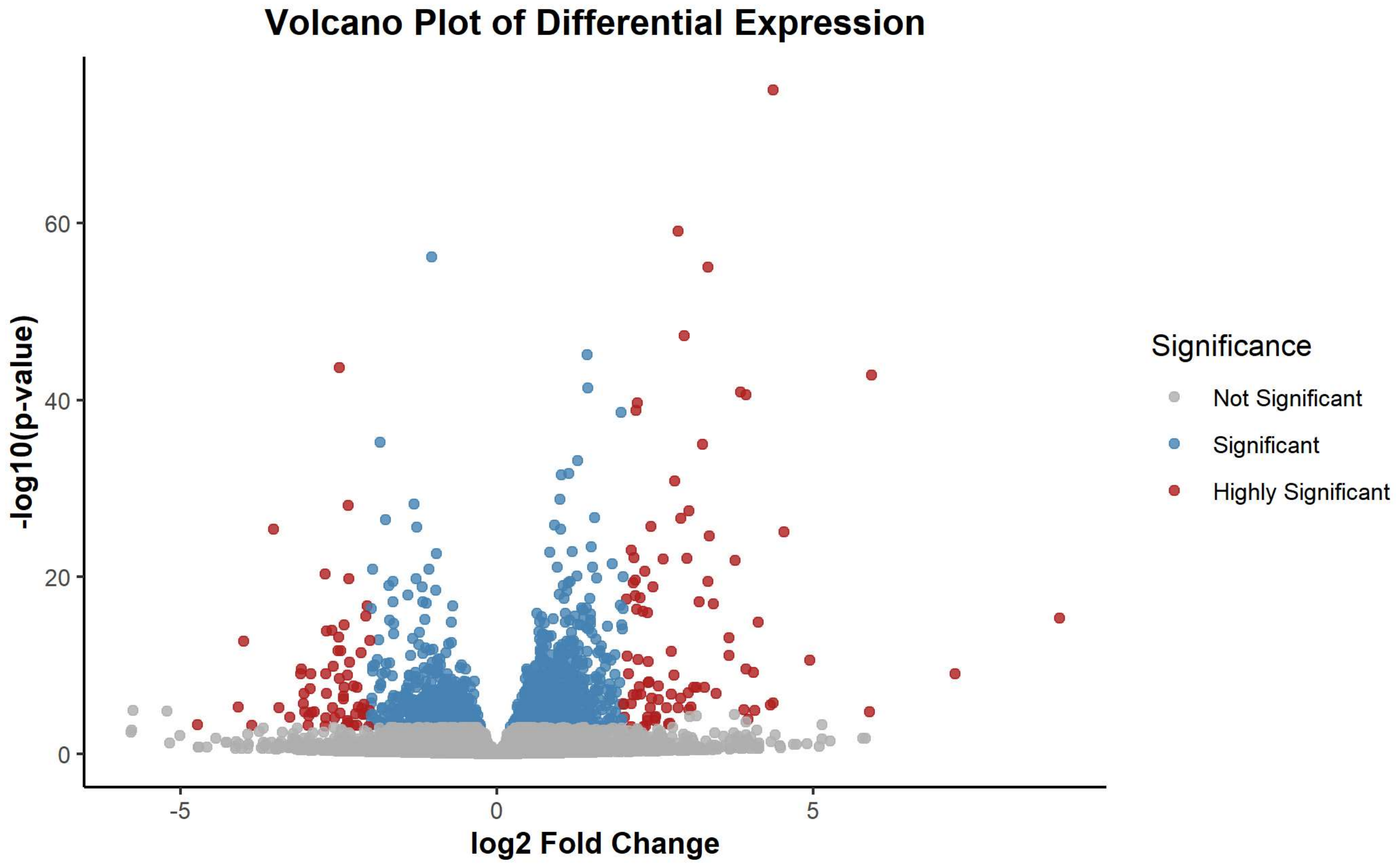
```
# Sort the results by padj and extract the top 6 genes
res <- res[order(res$padj), ]
top_genes <- rownames(res)[1:6]

# Create a list to store individual gene plots
plot_list <- list()

# Loop through each gene and create a plot for its normalized counts
for (g in top_genes) {
  # Get normalized counts for gene g
  data <- plotCounts(dds, gene = g, intgroup = "dex", returnData = TRUE)

  # Create a ggplot object with custom formatting and margins for spacing
  p <- ggplot(data, aes(x = dex, y = count, color = dex)) +
    geom_point(size = 3, position = position_jitter(width = 0.1)) +
    scale_y_log10() +
    labs(title = paste("Gene:", g),
         x = "Treatment",
         y = "Normalized Count (log scale)") +
    theme_minimal() +
    theme(plot.title = element_text(face = "bold", hjust = 0.5),
          axis.title = element_text(face = "bold"),
          legend.position = "none",
          plot.margin = unit(c(0.5, 0.5, 0.5, 0.5), "cm"))

  # Add the plot to our list
  plot_list[[g]] <- p
}

# Arrange the individual gene plots in a 2 x 3 grid with extra space between them
grid.arrange(grobs = plot_list, ncol = 3)
```



Volcano Plot of Differential Expression

The volcano plot summarizes the differential expression results across all genes. Genes are color-coded based on significance:

Grey: Not significant. Steelblue: Significant (padj < 0.01). Firebrick: Highly significant (padj < 0.01 and |log2FoldChange| > 2).



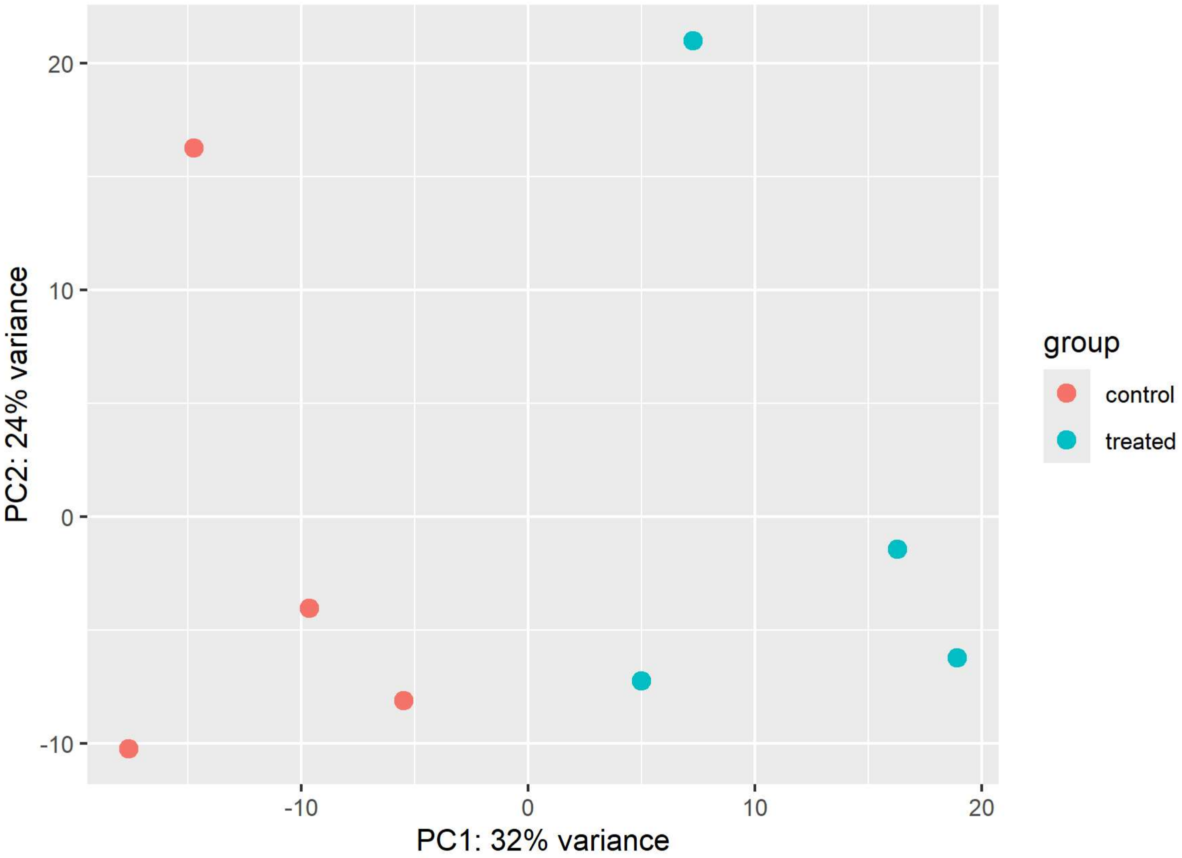
Principal Component Analysis (PCA)

PCA is used to explore overall sample variation and assess clustering based on the treatment condition. The variance-stabilizing transformation (VST) helps in visualizing the data on a comparable scale.

Hide

```
# Perform variance-stabilizing transformation
vsdata <- vst(dds, blind = FALSE)

# Plot PCA grouped by treatment condition
plotPCA(vsdata, intgroup = "dex")
```

```
using ntop=500 top features by variance
```



Conclusion

This report demonstrates our comprehensive DESeq2 analysis workflow:

Data Preparation: We loaded count data and metadata, ensuring that our samples are properly annotated. Differential Expression Analysis: DESeq2 was used to identify genes that respond to the treatment. Visualization: Detailed plots of top genes and a volcano plot summarize the differential expression results, while a PCA plot reveals sample clustering based on treatment. The analyses provide a robust foundation for further investigation and biological interpretation of the treatment effects. Please let me know if you have any questions or need further details regarding the workflow.