Zika DESeq Demo

Generated by Bryce Watson for Van Doorslaer Lab

1/18/2021

Contents

Setup and Data Import

Instruction and rationale for how to conduct RNA-seq using the DESeq2 R package is outlined in the Bioconductor vignette: **Analyzing RNA-seq data with DESeq2**. Script developed with guidance of **Simon Cockell**.

First we load the library consisting of BiocManager,DESeq2, Biobase, ggplot2, ggrepel, apeglm, pheatmap, genefilter,plotly, tibble, and rmarkdown. Then we perform initial data import of raw count files from a csv.

The dataset for this RNA Seq experiment can be found on the **NCBI GEO Database**

Make DESeq Dataset

Once un-normalized read counts are loaded from a .csv some simple organization is required before performing DESeq operations. In order to make a DESeq Data Set (dds), a .txt or .csv meta file specifying the experimental conditions and samples needs to be built and loaded into the 'col_data' vector.

Once that is done we can build the basic dds object with the function described below. Then using the results() function we can generate a results table with log2 fold changes, p values and adjusted p values.

```
dds = DESeqDataSetFromMatrix(countData = cts,
                             colData = col_data,
                             design = ~ Condition)

#dds prefiltering
ddsf <- dds[ rowSums(counts(dds)) > 1, ]

#DeSeq Dataset
dds <- DESeq(ddsf)
res <- results(dds)
res_df <- as.data.frame(res)
filter_df <- res_df[complete.cases(res_df),] # Filters out incomplete rows.
```

Table 1: Header of Results Dataframe

	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj
ENSG00000225972	161.47154	-1.9686827	3.4137485	-0.5766924	0.5641473	0.8316329
ENSG00000225630	126.91552	-0.1622252	0.9440682	-0.1718363	0.8635662	0.9587603
ENSG00000237973	529.31391	0.0697169	1.0716423	0.0650561	0.9481293	0.9852001
ENSG00000248527	2075.74428	-0.5715542	0.8589732	-0.6653924	0.5057995	0.8013907
ENSG00000228794	24.97574	-0.2476499	1.1566025	-0.2141184	0.8304547	0.9449305

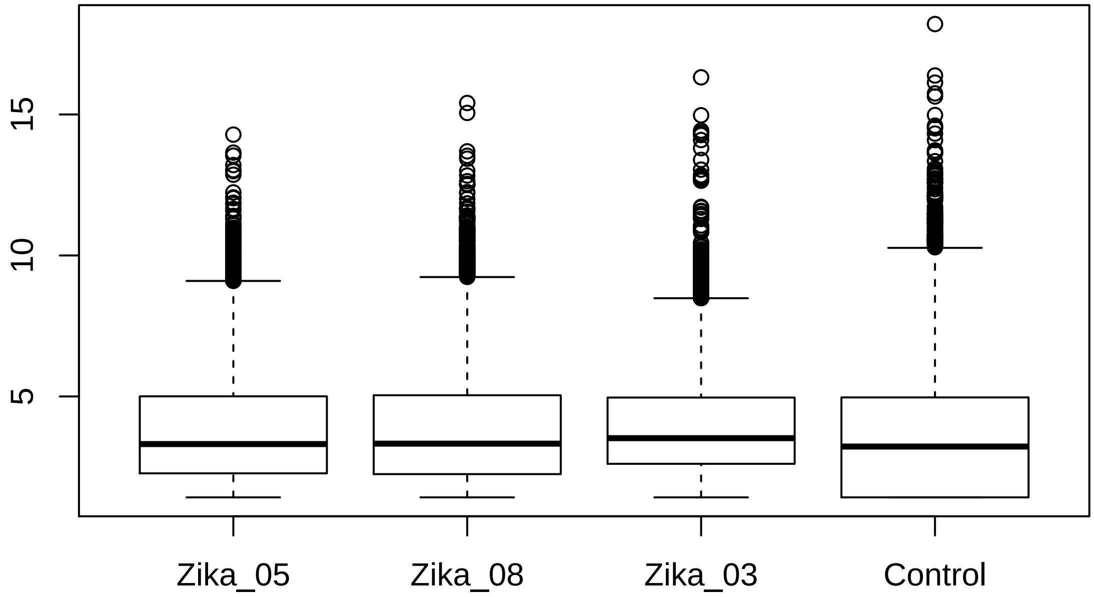
Normalization and Data Merging

The next step is to normalize data with the VarianceStabilizingTransformation() to produce a *DESeqTransform* object. This object will be mainly used for variance measurements going forward such as a PCA plot, MA plot and heatmap.

```
#Normalization
vst = varianceStabilizingTransformation(dds)
vsd <- vst(dds, blind = FALSE)
```

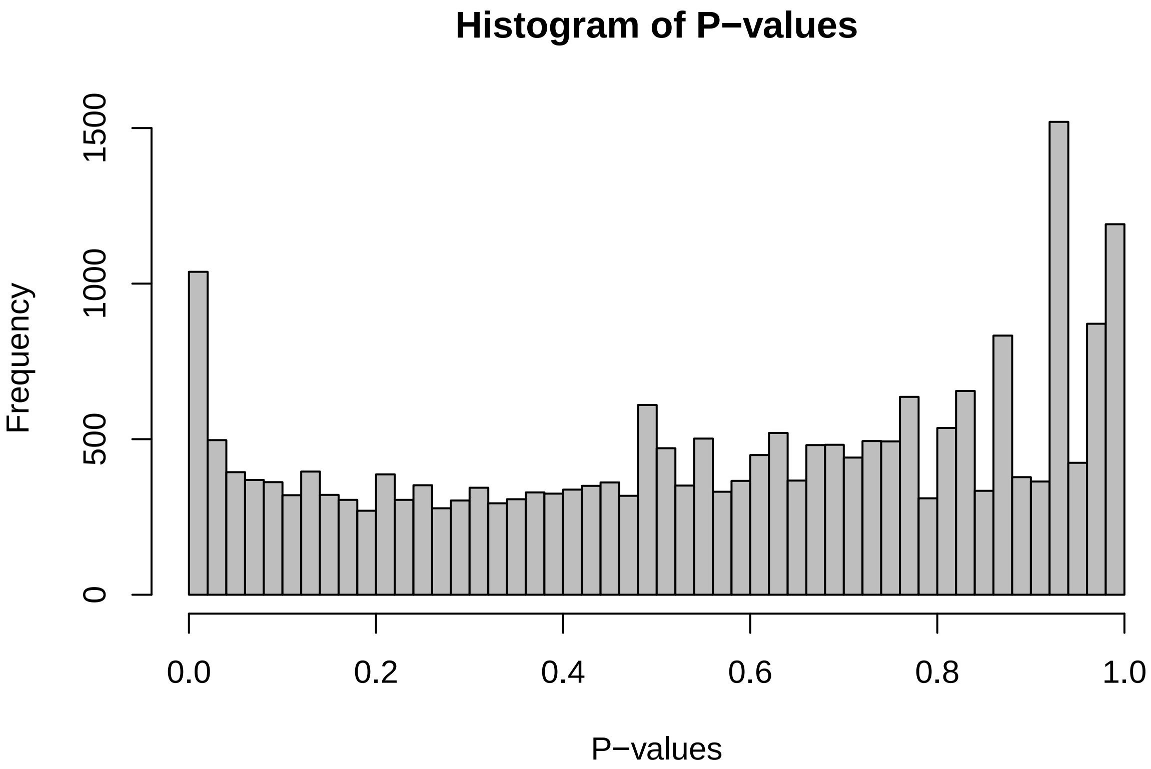
Boxplot

An easy start to get an overview of the data is with a simple boxplot of the counts frequency.



P-Value Histogram

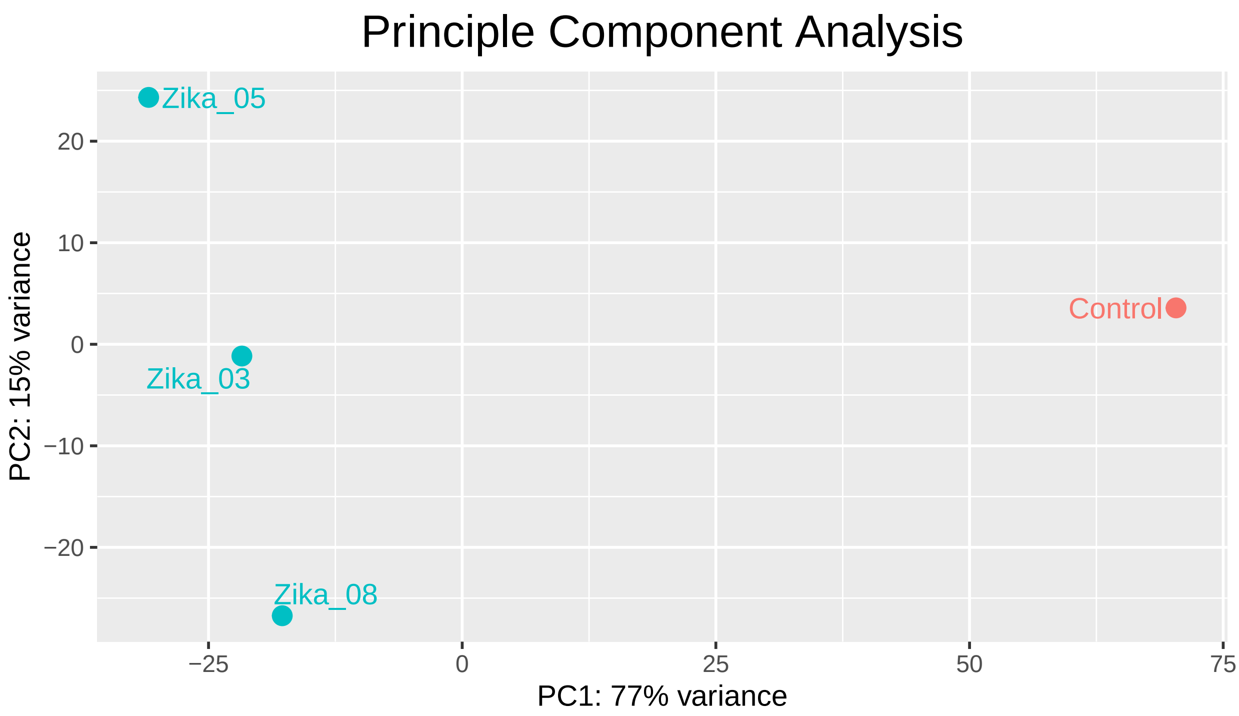
A histogram of the distribution of P-values in the DESeq results table allows us to see a simplified pattern of p-value spread.



PCA Plot

The first overview we will use to visualize distances between samples is a principle component analysis (PCA). The plotPCA() function comes with the DESeq2 package and is built on ggplot2. So, it is also compatible with ggplot-adjacent packages such as ggrepel. Here we use the labeling functions of these packages to differentiate between groups.

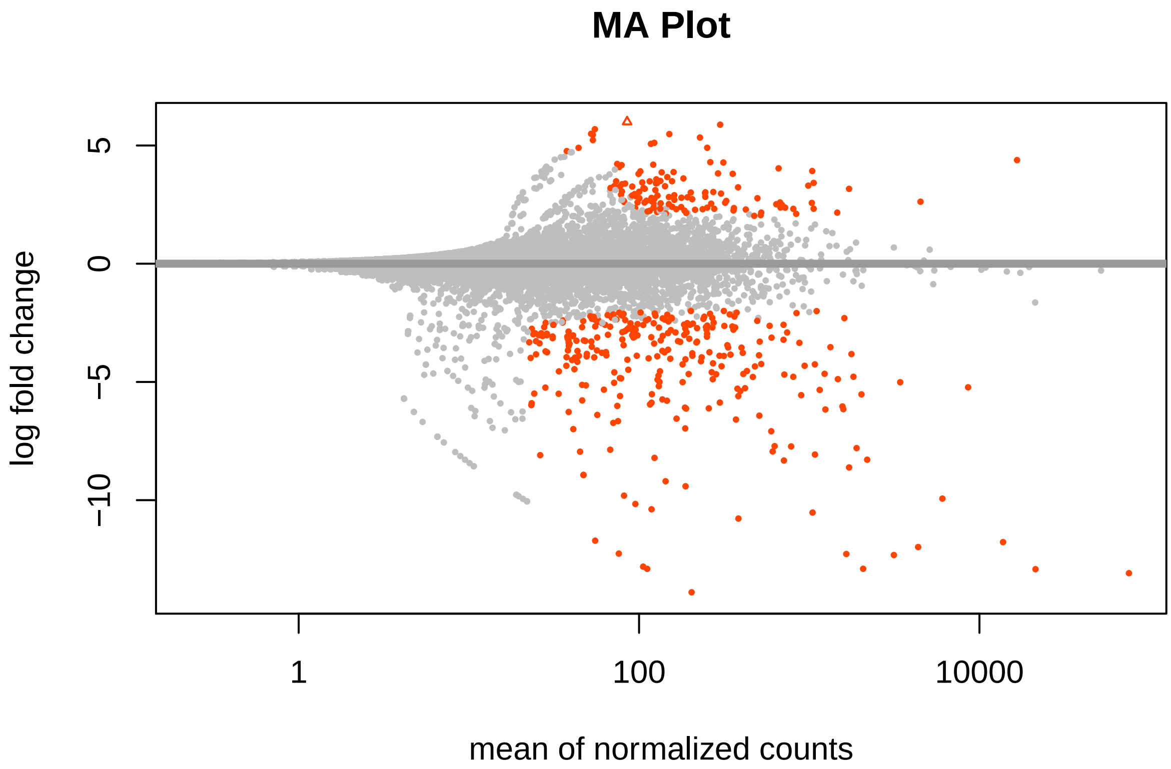
```r
plotPCA(vst, intgroup='Condition') +
  geom_text_repel(aes(label=name)) +
  ggtitle("Principle Component Analysis") +
  theme(legend.position="none", plot.title = element_text(size = rel(1.5), hjust = 0.5))
```



MA Plot

The next step is to get an overview of the experiment with plotMA(). MA plots represent each gene as a point on a graph. The X-axis plots the mean of the gene's expression across all samples. The Y-axis plots the average of counts normalized by size factor or the log2 fold change. The default alpha threshold for adjusted p-values is 0.1, adjusted here to match the 0.05 padj values in later plots.

```
plotMA(res, alpha=0.05, main='MA Plot', ylim=c(-14,6),
       colNonSig = "gray",
       colSig = "orangered",
       colLine = "grey60")
```



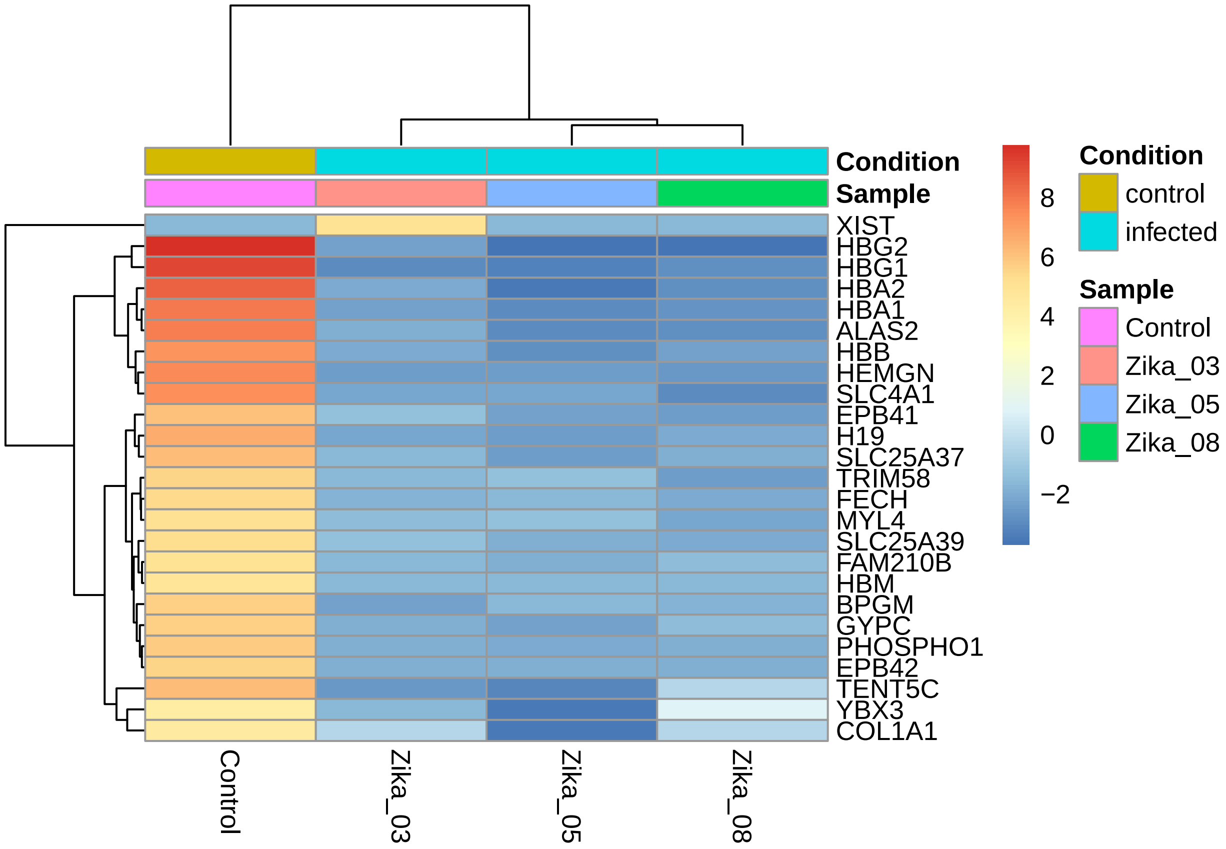
Gene Clustering Heatmap

For the next two plots we will need to convert Ensembl IDs to more readable gene IDs. Here the biomaRt package is used to convert to HUGO gene nomenclature (hgnc). Other R packages such as the organism annotation package can accomplish similar conversions.

```
#Genes of Interest Annotation
topVarGenes <- head(order(rowVars(assay(vsd)), decreasing = TRUE), 25)
mat  <- assay(vsd)[ topVarGenes, ]
mart <- useDataset("hsapiens_gene_ensembl", useMart("ensembl"))
mat  <- mat - rowMeans(mat)
gns <- getBM(c("hgnc_symbol", "ensembl_gene_id"), "ensembl_gene_id", row.names(mat), mart=mart, useCache
row.names(mat)[match(gns[,2], row.names(mat))] <- gns[,1]
```

Using the annotated gene IDs generated above we can now build a heatmap very simply with the pheatmap package. The 25 most variable genes are highlighted, but this is easily changed my modifying the 'topVarGenes' variable and running the Ens ID to hgnc conversion again.

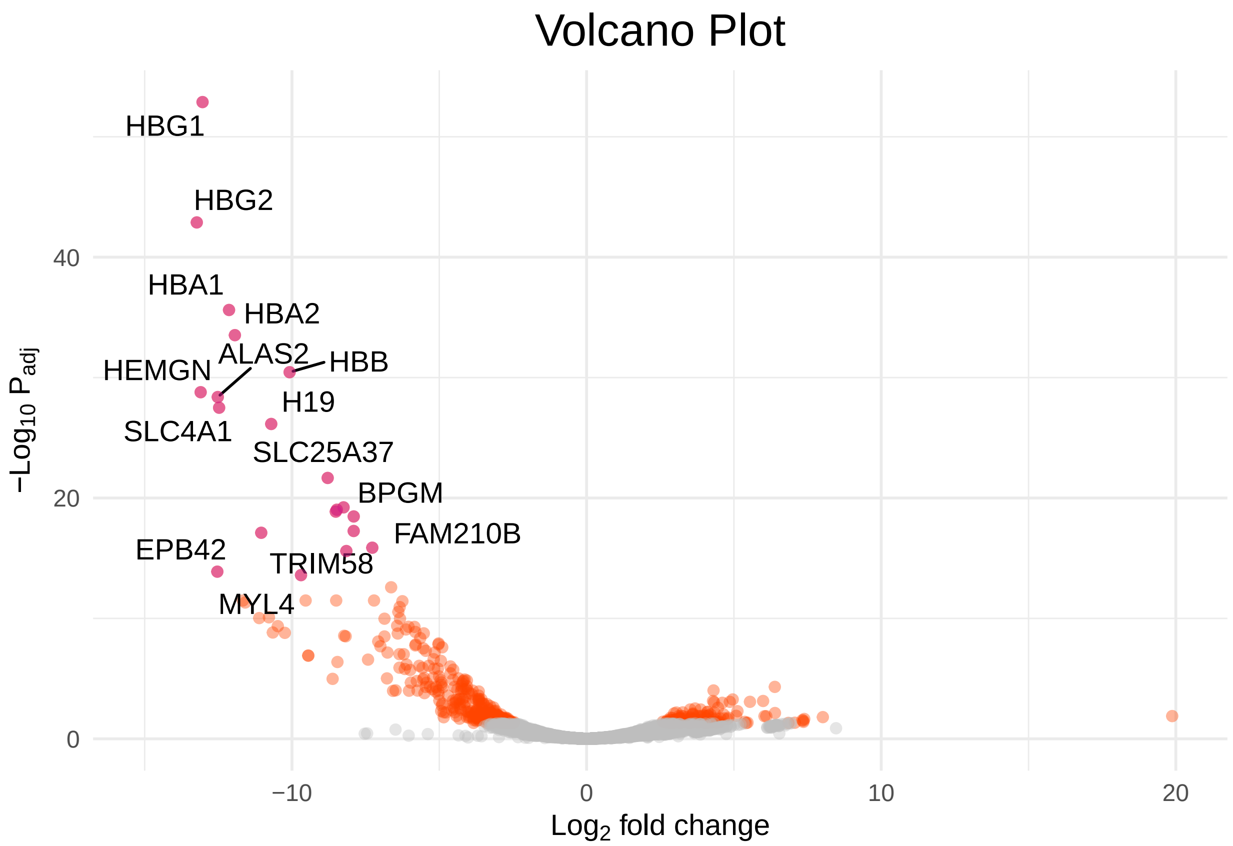
```
#Make Heatmap
anno <- as.data.frame(colData(vsd)[, c("Sample","Condition")])
pheatmap(mat, annotation_col = anno)
```



Volcano Plot

The final plot to get an overview of the data is a Volcano Plot. The Log2Fold change is on the X-axis. This shows how much a gene has changed. The djusted P-value of each gene is on the Y-axis, showing how significant that change is. This gives a good spread of change in gene expression for our dataset along with the chosen measure of significance (p-adj in this case).

```
ggplot(filter_df, aes(x=log2FoldChange, y=-log10(padj))) +
  geom_point(aes(color=test), size=1.5, alpha=0.4) +
  scale_color_manual(values=c('violetred', 'gray', 'orangered')) +
  xlim(-15, 20) +
  ggtitle('Volcano Plot') +
  labs(y=expression('-Log'[10]*' P'[adj]), x=expression('Log'[2]*' fold change')) +
  geom_text_repel(data=topSigGenes, force=5,aes(x = log2FoldChange, y = -log10(padj),label=Gene))+
  geom_point(data=topSigGenes,aes(x = log2FoldChange, y = -log10(padj), color='black',alpha=0.4))+
  theme_minimal() +
  theme(legend.position="none", plot.title = element_text(size = rel(1.5), hjust = 0.5))
```



Conclusion

RNA-Seq analysis shows high variability and significance in similar gene groups. Substantial overlap in findings with the original research article shows this exploratory analysis to be a success. As reported in (Aguiar et al. 2020) collagen-encoding genes and genes which code for extra-cellular matrix proteins are among the most variable. This supports the theory that some Zika congenital abnormalities are associated with increased permeability of the blood-brain barrier. This may heighten risk of ischemic stroke, intracranial bleeding, and allow for permeation of Zika virus into developing neonate cells.

References

Aguiar, Renato S., Fabio Pohl, Guilherme L. Morais, Fabio C. S. Nogueira, Joseane B. Carvalho, Letícia Guida, Luis W. P. Arge, et al. 2020. "Molecular Alterations in the Extracellular Matrix in the Brains of Newborns with Congenital Zika Syndrome." *Science Signaling* 13 (635). https://doi.org/10.1126/scisignal.aay6736.

Allaire, JJ, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, Winston Chang, and Richard Iannone. 2020. *Rmarkdown: Dynamic Documents for R*. https://github.com/rstudio/rmarkdown.

Durinck, Steffen, and Wolfgang Huber. 2020. *BiomaRt: Interface to Biomart Databases (I.e. Ensembl)*.

Durinck, Steffen, Yves Moreau, Arek Kasprzyk, Sean Davis, Bart De Moor, Alvis Brazma, and Wolfgang Huber. 2005. "BioMart and Bioconductor: A Powerful Link Between Biological Databases and Microarray Data Analysis." *Bioinformatics* 21: 3439–40.

Durinck, Steffen, Paul T. Spellman, Ewan Birney, and Wolfgang Huber. 2009. "Mapping Identifiers for the Integration of Genomic Datasets with the R/Bioconductor Package biomaRt." *Nature Protocols* 4: 1184–91.

Gentleman, R., V. Carey, W. Huber, and F. Hahne. 2019. *Genefilter: Methods for Filtering Genes from High-Throughput Experiments*.

Gentleman, R., V. Carey, M. Morgan, and S. Falcon. 2019. *Biobase: Base Functions for Bioconductor*.

Huber, W., V. J. Carey, R. Gentleman, S. Anders, M. Carlson, B. S. Carvalho, H. C. Bravo, et al. 2015. "Orchestrating High-Throughput Genomic Analysis with Bioconductor." *Nature Methods* 12 (2): 115–21. http://www.nature.com/nmeth/journal/v12/n2/full/nmeth.3252.html.

Kolde, Raivo. 2019. *Pheatmap: Pretty Heatmaps*. https://CRAN.R-project.org/package=pheatmap.

Love, Michael, Simon Anders, and Wolfgang Huber. 2019. *DESeq2: Differential Gene Expression Analysis Based on the Negative Binomial Distribution*. https://github.com/mikelove/DESeq2.

Love, Michael I., Wolfgang Huber, and Simon Anders. 2014. "Moderated Estimation of Fold Change and Dispersion for Rna-Seq Data with Deseq2." *Genome Biology* 15 (12): 550. https://doi.org/10.1186/s13059-014-0550-8.

Morgan, Martin. 2019. *BiocManager: Access the Bioconductor Project Package Repository*. https://CRAN.R-project.org/package=BiocManager.

Müller, Kirill, and Hadley Wickham. 2020. *Tibble: Simple Data Frames*. https://CRAN.R-project.org/package=tibble.

Sievert, Carson. 2020. *Interactive Web-Based Data Visualization with R, Plotly, and Shiny*. Chapman; Hall/CRC. https://plotly-r.com.

Sievert, Carson, Chris Parmer, Toby Hocking, Scott Chamberlain, Karthik Ram, Marianne Corvellec, and Pedro Despouy. 2020. *Plotly: Create Interactive Web Graphics via Plotly.js*. https://CRAN.R-project.org/package=plotly.

Slowikowski, Kamil. 2020. *Ggrepel: Automatically Position Non-Overlapping Text Labels with Ggplot2*. https://github.com/slowkow/ggrepel.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. 2020. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. https://CRAN.R-project.org/package=ggplot2.

Xie, Yihui. 2014. "Knitr: A Comprehensive Tool for Reproducible Research in R." In *Implementing Reproducible Computational Research*, edited by Victoria Stodden, Friedrich Leisch, and Roger D. Peng. Chapman; Hall/CRC. http://www.crcpress.com/product/isbn/9781466561595.

———. 2015. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. https://yihui.org/knitr/.

———. 2019. "TinyTeX: A Lightweight, Cross-Platform, and Easy-to-Maintain Latex Distribution Based on Tex Live." *TUGboat*, no. 1: 30–32. http://tug.org/TUGboat/Contents/contents40-1.html.

———. 2020a. *Knitr: A General-Purpose Package for Dynamic Report Generation in R*. https://yihui.org/knitr/.

———. 2020b. *Tinytex: Helper Functions to Install and Maintain Tex Live, and Compile Latex Documents*. https://github.com/yihui/tinytex.

Xie, Yihui, J. J. Allaire, and Garrett Grolemund. 2018. *R Markdown: The Definitive Guide*. Boca Raton, Florida: Chapman; Hall/CRC. https://bookdown.org/yihui/rmarkdown.

Xie, Yihui, Christophe Dervieux, and Emily Riederer. 2020. *R Markdown Cookbook*. Boca Raton, Florida: Chapman; Hall/CRC. https://bookdown.org/yihui/rmarkdown-cookbook.

Zhu, Anqi, Joseph G. Ibrahim, and Michael I. Love. 2018. “Heavy-Tailed Prior Distributions for Sequence Count Data: Removing the Noise and Preserving Large Differences.” *Bioinformatics*. https://doi.org/10.1093/bioinformatics/bty895.

Zhu, Anqi, Joseph Ibrahim, and Michael Love. 2019. *Apeglm: Approximate Posterior Estimation for Glm Coefficients*.