Final Project

Jared Tesar

11/16/2021

1.0: DESeq Object Creation

```
DESeqDataSet = DESeqDataSetFromMatrix(countData = rnaCounts, colData = sampleAnnotation,
      design = ~ time + genotype + time:genotype)

DESeqDataSet = DESeq(DESeqDataSet, parallel=FALSE, test = "LRT", reduced = ~ time + genotype)
```

This chunk creates a malleable DESeq object containing the count data from rnaCounts, and generates p-values for the time and genotype interaction term.

1.1: False Discovery Rates

```
res = results(DESeqDataSet)

nonAdj = res %>%
  as.data.frame() %>%
  filter(pvalue <= 0.1)

meanP = (nonAdj  %>% drop_na() %>% summarize(mean = mean(pvalue)))

#Total Gene Count
res %>% as.data.frame %>% count()
```

This chunk creates a data frame from the DESeq object. Then the dataset is magnified to show just the genes with a false positive likelihood of less than 10%. Finally, the number of genes expected to be false positives was calculated.

```
##       n
## 1 18399
```

```
#Gene Count With p-value <= 0.1
nonAdj %>% count()
```

```
##      n
## 1 4265
```

```
#Mean of p-value for genes with a p-value <= 0.1
meanP
```

```
##         mean
## 1 0.03120147
```

```
#Expected number of false positive discoveries within expected significant interactions
meanP[,1] * (nonAdj %>% count())
```

```
##          n
## 1 133.0743
```

Of the 18,399 total genes in the given dataset, 4,265 of them show evidence of a significant time:genotype interaction term, given a false discovery rate of 10% and below.

The mean false discovery rate of these 4,265 genes with evidence of a significant interaction term is 0.0312, or about 3.1%. Therefore, the total number of genes expected to be false positives is 133.07.

2.0: Normalization of Counts

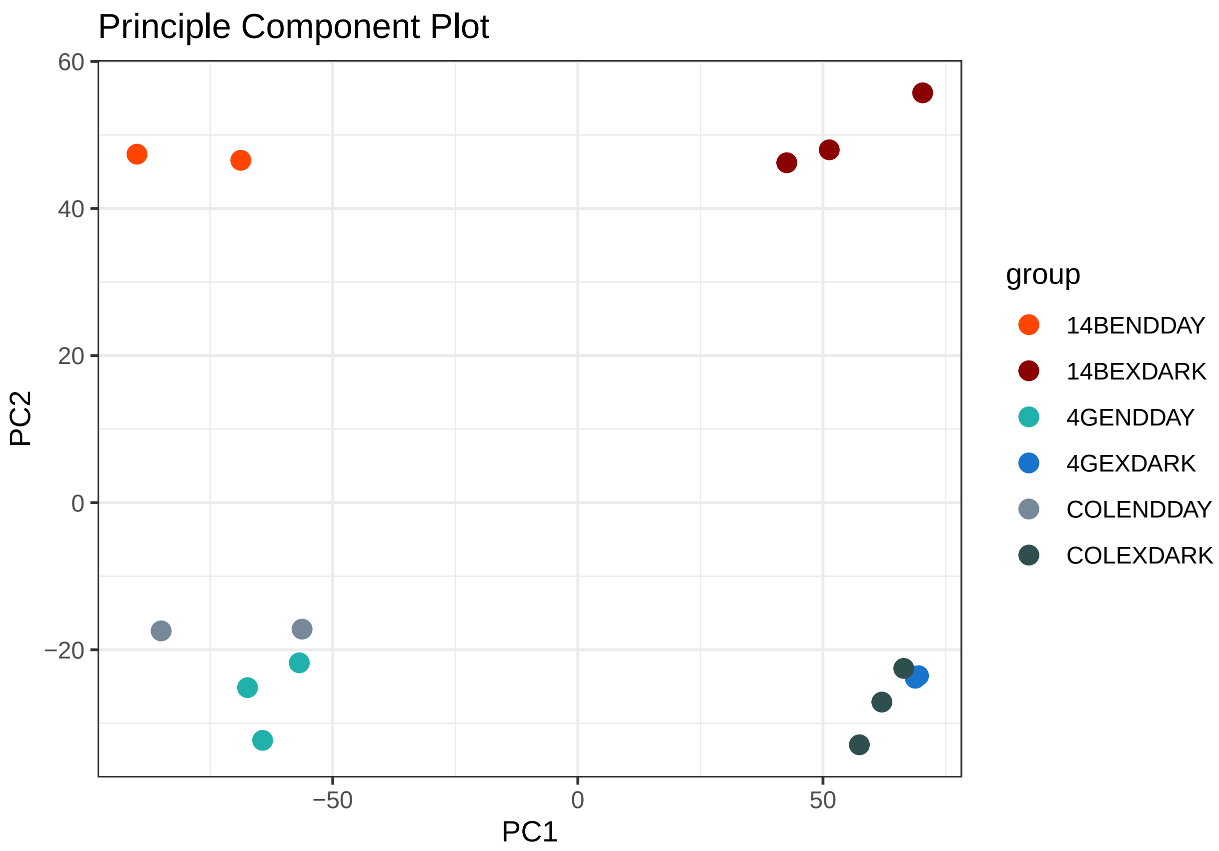
```


normCounts = counts(DESeqDataSet, normalized=TRUE)
lgNorm = log2(normCounts+1)
```

Using DESeq, the gene counts were normalized, and then log-transformed with an offset of 1.

3.0: Principle Component Analysis

```


pca = prcomp(t(lgNorm))

pcaData = data.frame(pca$x[,1:2])

pcaData$group = sampleAnnotation[rownames(pcaData),"group"]
pcaData$sample = rownames(pcaData)

pcaGG = ggplot(pcaData, aes(x=PC1, y=PC2, color=group, label=sample)) + geom_point(size=3) +
                scale_color_manual(values=groupColors) + ggtitle('Principle Component Plot')
pcaGG
```

This chunk contains a principle component analysis, and a plot representing it. Additionally, the minimum and maximum PC1 and PC2 values and the genes corresponding to them are dis-



covered.

```
#PC1 Of Least Value
min1 = head(sort(pca$x[,1]), n=1)
#PC1 of Most Value
max1 = tail(sort(pca$x[,1]), n=1)


#PC2 Of Least Value
min2 = head(sort(pca$x[,2]), n=1)
#PC2 of Most Value
max2 = tail(sort(pca$x[,2]), n=1)

#Lowest PC1 Gene
min1
```

```
## 14BENDDAY4
##  -89.92308
```

```
#Highest PC1 Gene
max1
```

```
## 14BEXDARK2
##   70.33168
```

```
#Lowest PC2 Gene
min2
```

```
## COLEXDARK4
##  -32.92678
```

```
#Highest PC2 Gene
max2
```

```
## 14BEXDARK2
##   55.73175
```

As shown by the min# and max# variables, as well as the PC Plot, the greatest distance between two samples in the PC1 direction exists between samples 14BENDDAY4and 14BEXDARK2, while that in the PC2 direction exists between COLEXDARK4 and 14BEXDARK2.

4.0: Extracting Genes of Interest

```
#Finds IDs of all genes within gene ID group of interest
primIDs = goAssociations %>%
  filter(gene_ontology_primary_id == "GO:0071495")

#Grabs names/symbols/descriptions of all of these genes
genesOfInterest = geneNamesAndDescriptions %>%
  filter(gene %in% primIDs$gene)

#Checks that each gene appears only once
#genesOfInterest %>% count(gene)

#Outputs this dataset to a .tsv file
write.table(
  genesOfInterest,
  "genesOfInterest.tsv",
  sep = "\t",
  row.names = FALSE,
  quote = FALSE
)
```

This chunk uses dplyr to filter one gene group from a dataset of many. Then these genes are matched with those in a dataset containing gene names, symbols, and descriptions. Finally, this dataset of the names, symbols, and descriptions of only the genes of interest is exported to a .tsv file, "genesOfInterest.tsv", provided in this report.

5.0: Filtering Normalized Counts

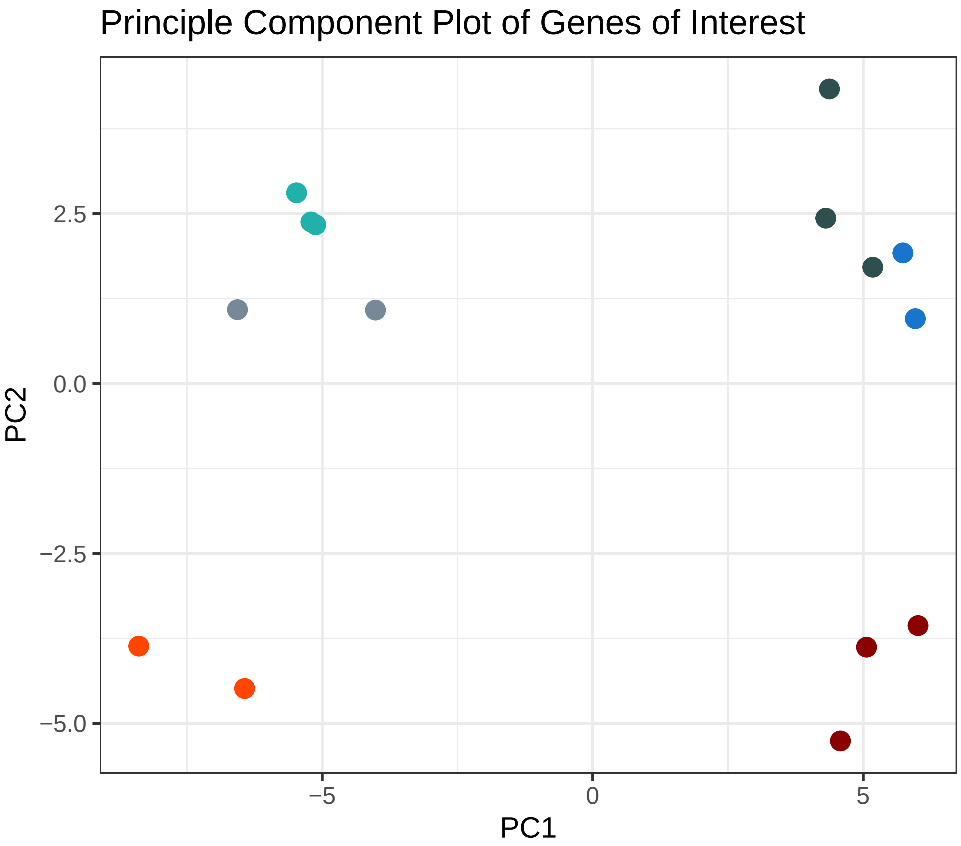
```
#Converts lgNorm to dataframe.
normFrame = lgNorm %>% as.data.frame()

#Moves rownames to column for filtering
lgGo = rownames_to_column(normFrame, var = "geneName")

lgGo = lgGo %>%
  filter(geneName %in% primIDs$gene)

lgGo = column_to_rownames(lgGo, var = "geneName")
```

This chunk simply filters the normalized counts dataframe for just the genes of interest defined in part 4.0.

6.0: Principle Component Analysis of Genes of Interest

```
pca2 = prcomp(t(lgGo))

pcaData2 = data.frame(pca2$x[,1:2])

pcaData2$group = sampleAnnotation[rownames(pcaData2),"group"]
pcaData2$sample = colnames(lgGo)

pcaGG2 = ggplot(pcaData2, aes(x=PC1, y=PC2, color=group, label=sample)) + geom_point(size=3) +
                scale_color_manual(values=groupColors) + ggtitle('Principle Component Plot of Genes of
pcaGG2
```

This chunk computes another Principle Component Analysis, but this time only for the genes within one gene group of interest.

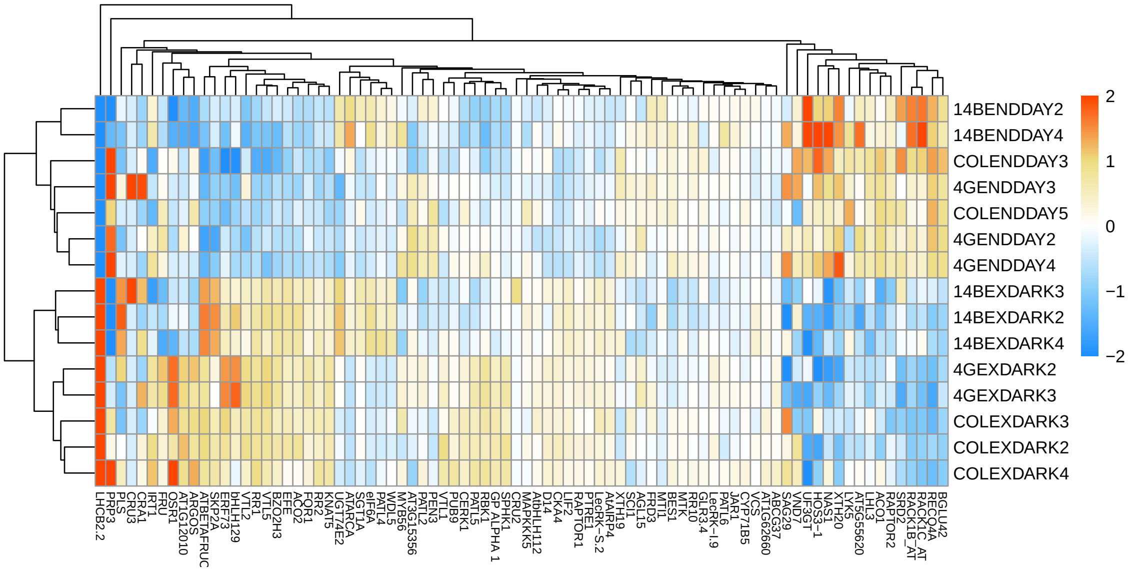


Compared to the first PC Plot of all the gene samples, this plot shows genes much more similar to one another. Instead of a plot with an x axis ranging from near -100 to 80, we see an x axis ranging from about -10 to 7.5. Additionally, the y axis is a much smaller range of -6 to 5, compared to the first y axis of -40 to 60. Like the principle components themselves, the samples have changed drastically in relation to them. Interestingly enough, the relative PC2 values seem to be reversed, with those that initially scored high for PC2 are no scoring low, and visa versa. While changed, the relative PC1 values are much more conserved. Again, however, all PC values have had their magnitudes drastically reduced.

7.0: Generating Gene Heatmap

```
heatData = lgGo - rowMeans(lgGo)
heatData[heatData > 2] = 2
heatData[heatData < -2] = -2
heatMap = pheatmap(t(heatData), color = heatPalette, clustering_method = "average", labels_col = geneNa
```

This chunk creates a heatmap displaying each of the distances between each sample's gene's count and the mean gene count accross all samples for each given gene.

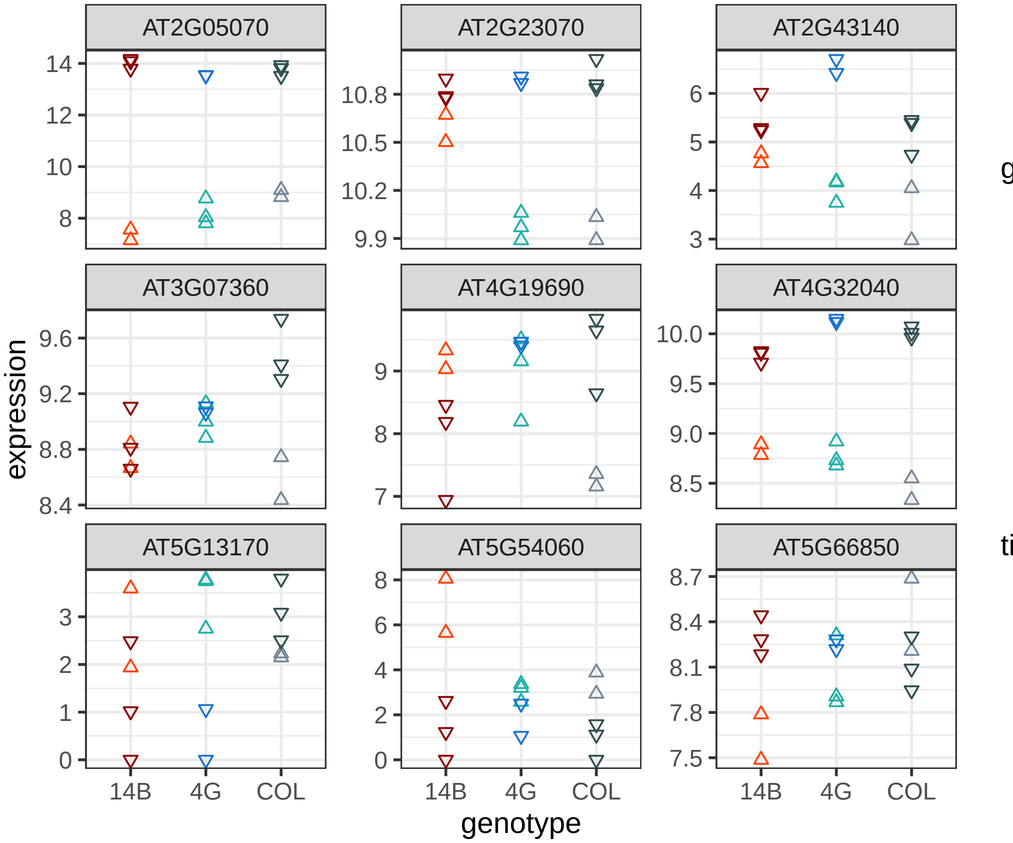


The two largest clusters of samples are separated by the day and dark aspects of the samples. Interestingly enough, two genes are split from all of the other genes, and these are represented either extremely strongly, or extremely weakly, for dark and day samples respectively.

8.0: Stripchart of Highest Probability Interaction Terms

```
#Grab the genes in the DESeq matrix that are part of my genes of interest

#Grab the nine genes with the lowest pvalues for the genotype:time interaction term.
nineGenes = head(nonAdj %>% rownames_to_column(var = "geneName") %>% filter(geneName %in% primIDs$gene)

nineGenesCounts = lgGo %>% rownames_to_column(var = "geneName") %>% filter(geneName %in% nineGenes$geneN

stripChart = stripchart321g(nineGenesCounts, sampleAnnotation)

stripChart
```

This chunk creates a stripchart for the nine genes of interest with the lowest p-value, and therefore highest likelihood, of having a significant interaction term between genotype and time, showing the strength of gene expression for each genotype and sample, distinguished by the



time attribute of each sample.

Many of the genes plotted here appear to be have distinguishable expression levels on the basis of time. However, not all genotypes are affected in the same way by the difference in time across all genes. For example, in the gene AT2G23070, the genotype 14B shows very little difference between the time groups, but the genotypes 4G and COL have relatively distant expression levels based on time group. An even greater example of this observation occurs in gene AT5G13170. In this gene, the genotypes 14B and 4G show a high expression of ENDDAY samples, while the COL genotype presents a higher expression of EXDARK samples than ENDDAY.