Deseq2_Task09

John Reddy Peasari

4/27/2021

Setting the current working directory

```
dir <- getwd()
setwd(dir)
```

Loading required packages

Loading the expression data and the meta data

```
gene_count <- read.csv("../Task09-Transcriptomic_Analysis/mayo.path_aging.con.salmon.gene.counts.csv",r
un <- gene_count[!duplicated(gene_count$X),] ## Removing the records that contain same row names
write.csv(un,"gene_count_unique_trail.csv", row.names = FALSE) ## Saving it to a new csv file
```

```
gene_count_unique <- read.csv("gene_count_unique_trail.csv",row.names = 1) ## Loading the refined csv c
phenotype <- read.csv("../Task09-Transcriptomic_Analysis/mayo.path_aging.con.phenotype.csv")

### Converting Diagnosis, AgeAtDeath, and Sex to a factor values for the Deseq2
phenotype$Diagnosis <- as.factor(phenotype$Diagnosis)
phenotype$AgeAtDeath <- as.factor(phenotype$AgeAtDeath)
phenotype$Sex <- as.factor(phenotype$Sex)

head(gene_count_unique)[1:6] ## Displaying the head of count file
```

```
##            X12306_TCX X12308_TCX X18198_TCX X18199_TCX X18200_TCX X18201_TCX
## 5S_rRNA         0.000      0.000      0.000      0.000      0.000      0.000
## 7SK            65.265     25.000     22.653     22.000     25.529     23.000
## A1BG          313.568    202.025    220.096    302.769    163.877    401.159
## A1BG-AS1      168.680    383.478    133.707    462.377    102.781    409.766
## A1CF            1.000      3.000      1.000      1.000      0.000      1.000
## A2M         14450.625   5840.377   6024.068   8653.515   8900.739   6063.940
```

```
head(phenotype) ## Displaying the head of count file
```

```
##         UID         Tissue        Diagnosis Sex  AgeAtDeath
## 1 12306_TCX TemporalCortex Pathologic_Aging   F          67
## 2 12308_TCX TemporalCortex Pathologic_Aging   M          72
## 3 18198_TCX TemporalCortex Pathologic_Aging   F          81
## 4 18199_TCX TemporalCortex Pathologic_Aging   F          75
## 5 18200_TCX TemporalCortex Pathologic_Aging   M          88
## 6 18201_TCX TemporalCortex Pathologic_Aging   F 90_or_above
```

Checking if number of rows in the meta-data equals number of columns in the count data

For the deseq2 to run, it should be true

```
ncol(gene_count_unique)
```

```
## [1] 61
```

```
nrow(phenotype)
```

```
## [1] 61
```

Design formula

It is an important analysis to do to identify the major sources of variation in the dataset. One can control those variances inside the dese2 model statistics using the design formula. It was evident from the task9 instructions that the major covariate or major varaition was " Diagnosis “. Howerever, we are controlling the other covariates or varations such as” Sex " and " AgeAtDeath “. We are looking for the gene expression difference between the”Diagnosis" levels. Therefore, we are looking for genes that are diffeentially expressed according to the " Diagnosis" factor.

Create a DeSeqDataSet by specifying the gene expression count values datafame, meta-data dataframe and the design formula.

```
dds <- DESeqDataSetFromMatrix(round(gene_count_unique), phenotype, formula(~ Sex + AgeAtDeath + Diagnos
```

```
## converting counts to integer mode
```

```
assay(dds)[1:3,1:2] ## Displaying first 3 rows and 3 columns of the DESeqDaset count
```

```
##         X12306_TCX X12308_TCX
## 5S_rRNA          0          0
## 7SK             65         25
## A1BG           314        202
```

```r
colData(dds)[1:3,1:5] ## Displaying first 3 rows and 3 columns
```

```
## DataFrame with 3 rows and 5 columns
##                    UID        Tissue        Diagnosis      Sex AgeAtDeath
##            <character>   <character>         <factor> <factor>   <factor>
## X12306_TCX   12306_TCX TemporalCortex Pathologic_Aging        F         67
## X12308_TCX   12308_TCX TemporalCortex Pathologic_Aging        M         72
## X18198_TCX   18198_TCX TemporalCortex Pathologic_Aging        F         81
```

Looking for the levels in each factor of the meta-data file

```r
dds$Diagnosis
```

```
##  [1] Pathologic_Aging Pathologic_Aging Pathologic_Aging Pathologic_Aging
##  [5] Pathologic_Aging Pathologic_Aging Pathologic_Aging Pathologic_Aging
##  [9] Pathologic_Aging Pathologic_Aging Pathologic_Aging Pathologic_Aging
## [13] Pathologic_Aging Pathologic_Aging Pathologic_Aging Pathologic_Aging
## [17] Pathologic_Aging Pathologic_Aging Pathologic_Aging Pathologic_Aging
## [21] Pathologic_Aging Pathologic_Aging Pathologic_Aging Pathologic_Aging
## [25] Pathologic_Aging Pathologic_Aging Pathologic_Aging Pathologic_Aging
## [29] Pathologic_Aging Pathologic_Aging Pathologic_Aging Pathologic_Aging
## [33] Pathologic_Aging Pathologic_Aging Pathologic_Aging Pathologic_Aging
## [37] Pathologic_Aging Pathologic_Aging Pathologic_Aging Pathologic_Aging
## [41] Pathologic_Aging Control          Control          Control
## [45] Control          Control          Control          Control
## [49] Control          Control          Control          Control
## [53] Control          Control          Control          Control
## [57] Control          Control          Control          Control
## [61] Control
## Levels: Control Pathologic_Aging
```

Here, deseq2 assumes the the first factor in the levels as a refrence level or it uses chronological order. So our refrence level was Control

```r
dds$Sex
```

```
##  [1] F M F F M F F F F F F F F M M F M F F F F M M F F M M F F M M M F M M F M F
## [39] M F F F M F F M F F F F F M M F M F F F F M F
## Levels: F M
```

```r
dds$AgeAtDeath
```

```
##  [1] 67          72          81          75          88          90_or_above
##  [7] 90_or_above 90_or_above 90_or_above 90_or_above 90_or_above 79
## [13] 90_or_above 75          82          90_or_above 81          90_or_above
## [19] 83          90_or_above 79          85          90_or_above 90_or_above
## [25] 89          90_or_above 90_or_above 87          77          90_or_above
## [31] 90_or_above 83          90_or_above 81          77          87
## [37] 90_or_above 90_or_above 81          90_or_above 90_or_above 87
## [43] 89          90_or_above 88          90_or_above 90_or_above 90_or_above
## [49] 90_or_above 88          89          90_or_above 89          87
## [55] 88          90_or_above 86          90_or_above 87          90_or_above
## [61] 90_or_above
## Levels: 67 72 75 77 79 81 82 83 85 86 87 88 89 90_or_above
```

```
dds <- DESeq(dds)
```

```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing

## 12 rows did not converge in beta, labelled in mcols(object)$betaConv. Use larger maxit argument with

## -- replacing outliers and refitting for 19946 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)

## estimating dispersions

## fitting model and testing

## 11 rows did not converge in beta, labelled in mcols(object)$betaConv. Use larger maxit argument with
```

```
resultsNames(dds)
```

```
##  [1] "Intercept"
##  [2] "Sex_M_vs_F"
##  [3] "AgeAtDeath_72_vs_67"
##  [4] "AgeAtDeath_75_vs_67"
##  [5] "AgeAtDeath_77_vs_67"
##  [6] "AgeAtDeath_79_vs_67"
##  [7] "AgeAtDeath_81_vs_67"
##  [8] "AgeAtDeath_82_vs_67"
##  [9] "AgeAtDeath_83_vs_67"
## [10] "AgeAtDeath_85_vs_67"
## [11] "AgeAtDeath_86_vs_67"
## [12] "AgeAtDeath_87_vs_67"
## [13] "AgeAtDeath_88_vs_67"
## [14] "AgeAtDeath_89_vs_67"
## [15] "AgeAtDeath_90_or_above_vs_67"
## [16] "Diagnosis_Pathologic_Aging_vs_Control"
```

```r
results <- results(dds)
summary(results)
```

```
##
## out of 42982 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 1730, 4%
## LFC < 0 (down)     : 2921, 6.8%
## outliers [1]       : 33, 0.077%
## low counts [2]     : 15585, 36%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

Summary above provides information about the percentage of genes (both Up and Down regulated) that are differentially expressed. It was observed that around 4% genes i.e., 1730 genes are up regulated and around 7% genes are down regulated.

We can examine the counts and normalized counts for a single gene with the smallest p value. Showing the most significant gene in all the samples

```r


res = results(dds, contrast=c("Diagnosis","Pathologic_Aging","Control"))
write.table(res, file = "washu.csv", sep = ",",row.names = TRUE, col.names = NA) ## Writing DEGs to a n

ix = which.min(res$padj) # most significant
res <- res[order(res$padj),] # sort
head(res)
```

contrast() was used to get results for all the possible combnations for a condition.

```
## log2 fold change (MLE): Diagnosis Pathologic_Aging vs Control
## Wald test p-value: Diagnosis Pathologic Aging vs Control
## DataFrame with 6 rows and 6 columns
##                 baseMean log2FoldChange     lfcSE      stat      pvalue
##                <numeric>      <numeric> <numeric> <numeric>   <numeric>
## TMSB4XP8        136.6263      -1.691667 0.1527550 -11.07438 1.67031e-28
## SPCS2P4          54.4505      -2.103844 0.2232522  -9.42362 4.35803e-21
## YBX1P10          10.9911      -3.056021 0.3231879  -9.45586 3.20366e-21
## TMSB4XP4         62.1449      -1.750289 0.1917165  -9.12957 6.87743e-20
## DYNLL2         15250.5133      -0.713101 0.0880173  -8.10183 5.41368e-16
## RP11-265N6.2     31.7652      -1.631336 0.2198902  -7.41887 1.18128e-13
##                     padj
##                <numeric>
## TMSB4XP8     4.57081e-24
## SPCS2P4      3.97525e-17
## YBX1P10      3.97525e-17
## TMSB4XP4     4.70502e-16
## DYNLL2       2.96291e-12
## RP11-265N6.2 4.61796e-10
```

```r
kable(res[1:5,-(3:4)])
```

	baseMean	log2FoldChange	pvalue	padj
TMSB4XP8	136.62632	-1.6916670	0	0
SPCS2P4	54.45052	-2.1038440	0	0
YBX1P10	10.99107	-3.0560210	0	0
TMSB4XP4	62.14492	-1.7502886	0	0
DYNLL2	15250.51334	-0.7131012	0	0

```r
pdf("top_gene.pdf")
barplot(assay(dds)[ix,],las=2, main=rownames(dds)[ ix  ]) ## Barplot of the most significant gene
dev.off()
```

```
## pdf
##   2
```

```r
res1 = results(dds, contrast=c("Diagnosis","Pathologic_Aging","Control")) ## Creating another copy
```

Extracting the table of differential genes

```r
diff_genes = res %>%
  as.data.frame() %>%
  rownames_to_column("genes")
head(diff_genes)
```

```
##           genes    baseMean log2FoldChange      lfcSE       stat       pvalue
## 1      TMSB4XP8   136.62632     -1.6916670 0.15275500 -11.074380 1.670314e-28
## 2       SPCS2P4    54.45052     -2.1038440 0.22325224  -9.423619 4.358031e-21
## 3       YBX1P10    10.99107     -3.0560210 0.32318792  -9.455864 3.203657e-21
## 4      TMSB4XP4    62.14492     -1.7502886 0.19171652  -9.129566 6.877434e-20
## 5        DYNLL2 15250.51334     -0.7131012 0.08801726  -8.101834 5.413684e-16
## 6 RP11-265N6.2    31.76518     -1.6313358 0.21989020  -7.418865 1.181279e-13
##           padj
## 1 4.570814e-24
## 2 3.975251e-17
## 3 3.975251e-17
## 4 4.705025e-16
## 5 2.962909e-12
## 6 4.617958e-10
```

MA and dispersion plot

```r
pdf("Ma_plot.pdf")
plotMA(dds, alpha = 0.05,main=paste0('Diagnosis: Pathologic_Aging vs Control'))
topGene <- rownames(res)[which.min(res$padj)]
with(res[topGene, ], {
  points(baseMean, log2FoldChange, col="red", cex=2, lwd=2)
  text(baseMean, log2FoldChange, topGene, pos=2, col="red")
})
```

In the MA plot each datapoint represents a gene.In the above MA plot, highlighted points in blue are the genes whcih has an adjusted p-values less then 0.05. MA plots provides informations about the relationship between the expression chage between the conditions.All the genes that successfully pass the significance threshold are coloured in blue.Points will be coloured red if the adjusted p__value is less than 0.05. The x-axis is the average expression values between all the samples and the y-axis is the log2 fold change of the normalized counts between Pathologic__Aging and the Control. The top most differentially expressed gene was labelled and circled in a red colour.

```r
pdf("dispersion_plot.pdf")
plotDispEsts(dds, ylim = c(1e-6, 1e1))
dev.off()
```

```
## pdf
##   2
```

The above curve is used to visualize differentially expressed genes. Each dot each a gene with associated mean expression level.If you see a cloud or different shapes, then you might want to explore your data more to see if you have contamination or outlier samples. Note how much shrinkage you get across the whole range of means in the plotDispEsts() plot for any experiment with low degrees of freedom.

Data transformations

```r
 resLFC <- lfcShrink(dds = dds,
                  res = res1,
                  type = "normal",
                  coef = 2)
```

```
## using 'normal' for LFC shrinkage, the Normal prior from Love et al (2014).
##
## Note that type='apeglm' and type='ashr' have shown to have less bias than type='normal'.
## See ?lfcShrink for more details on shrinkage type, and the DESeq2 vignette.
## Reference: https://doi.org/10.1093/bioinformatics/bty895
```

Shrinkage of the effect size (LFC estimates) is used to visualize and rank the genes. We are here just shrinking the log2 fold change values to remove the noise associated with the fold change values that comes from genes with a low count values. We can build a volcano plot using the shrinkage values efficiently.

```r
rld <- rlog(dds, blind = FALSE)
```

```
## rlog() may take a long time with 50 or more samples,
## vst() is a much faster transformation
```

It was evident that the deseq2 already transforms the data initially, but for a other visualizaion plots one must still normalize the counta data. Extracting the rlog transformed values into an object called rld. The rlog transformation was estimated by fitting a general linear model (i.e., a GLM) to each gene for each sample. The rlog creates a matricx with an interncept and coefficient for each gene for each sample.

```r
vsd <- vst(dds, blind = FALSE)
```

Variance stabilizaing transformation (VST)is much faster to rlog transforamtion. VST looks at the trend between variance and mean in the count data and tries to find a strict transforamtion fit to the data.

PCA plots

Sample-sample to distance is visualized by a PCA.

```r
pdf("PCA_diagnosis.pdf")
pca_diagnosis <- plotPCA(vsd, intgroup = c("Diagnosis"), returnData = TRUE)
plotPCA(vsd, intgroup = c("Diagnosis"))
dev.off()
```

```
## pdf
##   2
```

In the PCA plot x-axis is the direction that seperates the data points most. The values of the samples written in this direction are termed as PC1 and the y-axis is the direction that seperates the data points second most. The values of these samples are written as PC2. The pecent variance was displayed on both the axis. By default the function uses top 500 most variable genes. The first dimension was seperating Pathologic_Aging samples from the Control samlples. It can be concluded that the once cannot expect the differences from this dataset. It looks there was not much variance observed between the conditions because the datapoints were not clustered. Instead, they are scattered apart from one another. The reason could be attributed to the imbalanced samples in Pathologic_Aging and the Control samples. It is also evident that the data doesn't contain outliers.

```r
pdf("PCA_diagnosis_sex.pdf")
pca_sex <- plotPCA(vsd, intgroup = c("Sex"), returnData = TRUE)
plotPCA(vsd, intgroup = c("Diagnosis","Sex"))
dev.off()
```

```
## pdf
##   2
```

Conclusion: The same conclusion appears to this PCA from the first PCA plot. It looks like there isn't the signal that one might expecting from this dataset.

Heatmaps of the count matrix

Heatmap is a great visualization plot to analyze gene count values. These heatmaps are generated by transforming the data accross the samples.

```r
pdf("heatmap1.pdf")
ntd <- normTransform(dds)
select <- order(rowMeans(counts(dds,normalized=TRUE)),
                decreasing=TRUE)[1:20]
df <- as.data.frame(colData(dds)[,c("Diagnosis","Sex")])
pheatmap(assay(ntd)[select,], cluster_rows=FALSE,
         cluster_cols=FALSE, annotation_col=df)
dev.off()
```

```
## pdf
##   3
```

Conclusion: It was observed that the samples are clustered by the condition. However the patterns found were not that strong. Here, Diagnosis and Sex factors are labelled in colour. It was observed that at the top of the plot a few genes are seperating the Pathologic_Aging diagnosis from others. It was also evident that both diagnosis conditions not clusered into sex type and show little variation from each other.

Conclusion: The heatmap gives similarities and dissimilarities between the samples and the conditions. Good correlations were observed at the bottom of the heatmap for a few genes accross all the combinations of the design conditions. High expression was observed for the bottom two genes.

Getting all the gene names in a data frame for further use

```r
## Use the dataframe all_genes to get the names of the genes
all_genes <- as.data.frame(diff_genes$genes)
names(all_genes) <- c("gene")
head(all_genes)
```

```
##             gene
## 1     TMSB4XP8
## 2      SPCS2P4
## 3      YBX1P10
## 4     TMSB4XP4
## 5       DYNLL2
## 6 RP11-265N6.2
```

```r
pdf("heatmap2.pdf")
betas <- coef(dds)
topGenes <- head(order(res1$padj),10)
mat <- betas[topGenes, -c(1,2)]
thr <- 2
mat[mat < -thr] <- -thr
```

```
mat[mat > thr] <- thr
pheatmap(mat, breaks=seq(from=-thr, to=thr, length=101),
         cluster_col=FALSE)
dev.off()
```

```
## pdf
##   3
```

Heatmap with user specifc gene list

```
heatmap1 <- function(top_genes) {
df <- as.data.frame(colData(dds)[,c("Diagnosis","Sex")])
pheatmap(assay(vsd)[top_genes,], annotation_col=df,cluster_rows=FALSE,
         cluster_cols=FALSE)
}
```

Here the user can specify a list of genes to plot as heatmap

```
top_genes <- diff_genes$genes[1:10] ### Getting only top 10 DEG

select_genes <- c("TMSB4XP8","SPCS2P4",'YBX1P10')  ## Make your own list of genes and call the function
pdf("heatmap3.pdf")
heatmap1(top_genes) ## Calling the function with a list of genes names
dev.off()
```

```
## pdf
##   3
```

Conclusion: It was observed that the gene DYNLL2 highly expressed in the control condition than pathologic_aging condition. A high correlation score was observed for the gene DYNLL2 in both Diagnosis conditions accross the two gender types.

Using DESeq2 plotCounts() to plot expression of a single gene

```
plot_single_gene <- function(name) {
data <- plotCounts(dds, gene=name, intgroup="Diagnosis", returnData=TRUE)
ggplot(data, aes(x = Diagnosis, y = count, color = Diagnosis)) +
  geom_point(position=position_jitter(w = 0.1,h = 0)) +
  geom_text_repel(aes(label = rownames(data))) +
  #theme_bw() +
  ggtitle(name) +
  theme(plot.title = element_text(hjust = 0.5))
}
```

Change the name of your desired gene name to plot the expression plot for that specific gene

Plotting all the genes at once is very useful. However, one would like to visualize and analize gene expression changes for a specific gene of interest.

```
pdf("count_plot.pdf")
topGene <- rownames(res)[which.min(res$padj)] ## To get the top gene name
plot_single_gene("DYNLL2") ## Change the gene name here
```

```
## Warning: ggrepel: 27 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

```
dev.off()
```

```
## pdf
##   2
```

Conclusion: The count values were grouped by the Diagnosis factor.Looks like count are more in the control condition than the Pathologic_Aging, which was also evident about this gene from the above heatmap. Here, the user can select any gene of interest.

Valcano Plot

Most commonly used plot to get the gloval view of what's goind was through a valcano plot.

```
diff_genes <- diff_genes %>%
                 mutate(threshold_OE = padj < 0.05 & abs(log2FoldChange) >= 0.58)
head(diff_genes)
```

```
##          genes    baseMean log2FoldChange      lfcSE       stat       pvalue
## 1     TMSB4XP8   136.62632     -1.6916670 0.15275500 -11.074380 1.670314e-28
## 2      SPCS2P4    54.45052     -2.1038440 0.22325224  -9.423619 4.358031e-21
## 3      YBX1P10    10.99107     -3.0560210 0.32318792  -9.455864 3.203657e-21
## 4     TMSB4XP4    62.14492     -1.7502886 0.19171652  -9.129566 6.877434e-20
## 5       DYNLL2 15250.51334     -0.7131012 0.08801726  -8.101834 5.413684e-16
## 6 RP11-265N6.2    31.76518     -1.6313358 0.21989020  -7.418865 1.181279e-13
##           padj threshold_OE
## 1 4.570814e-24         TRUE
## 2 3.975251e-17         TRUE
## 3 3.975251e-17         TRUE
## 4 4.705025e-16         TRUE
## 5 2.962909e-12         TRUE
## 6 4.617958e-10         TRUE
```

```
# diff_genes <- diff_genes %>% arrange(padj) %>% mutate(genelabels = "")
# diff_genes$genelabels[1:10] <- diff_genes$genes[1:10]
```

```
volcano_plot <- function(data) {

  ggplot(data,aes(x=log2FoldChange, y=-log10(padj))) +
        geom_point(aes(colour=threshold_OE)) +
        geom_text_repel(aes(label=genelabels)) +
        ggtitle("Volcano Plot") +
        xlab("log2 fold change") +
        ylab("-log10 adjusted p-value") +
        xlim(-10,10) +
        theme(legend.position = "none",
              plot.title = element_text(size = rel(1.5), hjust = 0.5),
              axis.title = element_text(size = rel(1.25)))
  }
```

Here, choose your oown gene names to display on the volcano plot.

To generate a valcano plot, we need a column to tell whether or not a particular gene is differentially expressed based on the adjusted p-value. In order to do that, obtain logical vector where TRUE values denote padj values < 0.05 and fold change > 1.5 vice versa. Once can aslo locate top 20 DE genes on the volcano plot but just having a column names genelabels that corresponds to whether that gene is significant or not.

```
diff_genes_copy <- diff_genes
diff_genes_copy <- diff_genes %>% arrange(padj) %>% mutate(genelabels = "")
# diff_genes_copy$genelabels[1:10] <- diff_genes_copy$genes[1:10]  ## or once can add their own gene na

diff_genes_copy$genelabels[1:2] <- c("TMSB4XP8","DYNLL2") ## Make sure you use only one type to choose
head(diff_genes_copy)
```

```
##          genes    baseMean log2FoldChange      lfcSE       stat       pvalue
## 1     TMSB4XP8   136.62632     -1.6916670 0.15275500 -11.074380 1.670314e-28
## 2      SPCS2P4    54.45052     -2.1038440 0.22325224  -9.423619 4.358031e-21
## 3      YBX1P10    10.99107     -3.0560210 0.32318792  -9.455864 3.203657e-21
## 4     TMSB4XP4    62.14492     -1.7502886 0.19171652  -9.129566 6.877434e-20
## 5       DYNLL2 15250.51334     -0.7131012 0.08801726  -8.101834 5.413684e-16
## 6 RP11-265N6.2    31.76518     -1.6313358 0.21989020  -7.418865 1.181279e-13
##           padj threshold_OE genelabels
## 1 4.570814e-24         TRUE   TMSB4XP8
## 2 3.975251e-17         TRUE     DYNLL2
## 3 3.975251e-17         TRUE
## 4 4.705025e-16         TRUE
## 5 2.962909e-12         TRUE
## 6 4.617958e-10         TRUE
```

```
pdf("valcano_plot.pdf")
volcano_plot(diff_genes_copy) ## Passing two gene names to the function to generate volcano plot
```

```
## Warning: Removed 28496 rows containing missing values (geom_point).
```

```
## Warning: Removed 28496 rows containing missing values (geom_text_repel).
```

```r
dev.off()
```

```
## pdf
##   2
```

Conslusion: It was evident from the above volcano plot that the TMSB4XP8 gene was the top most differentially expressed genes. Genes on left are down regulated and genes to the right are up regulated and genes coloured in red colour are not signficant genes.