Homework 4

Charlee Cobb

Data source

The data was downloaded from NCBI GEO (GSE124548). The study looks at the affect of a drug (Lumacaftor/Ivacaftor) to treat cystic fibrosis (CF). This drug has been approved for individuals that are homozygous for CFTR (Cystic fibrosis transmembrane conductance regulator) mutation. Clinical studies have observed that there is a large variation in the response thus the researchers are looking at RNA expressions to help identify the cause of this variation. Blood samples were taken from : - 20 healthy patients (do not have CF) - 20 patients that CF (before treatment) - 20 patients that have CF (after treatment - these are paired with those before treatment)

I have extracted the raw read counts from the original matrix and provided it with the homework. The healthy patients have "HC" in their names, CF patients before treatment are labeled "Base" and after treatment are labeled "V2".

For the homework we will compare the HC with Base to do a simple unpaired differentially expression.

Step 1 (5pts)

Load the file **GSE124548.raw.txt** and create a new dataframe with just the columns with the raw counts for healthy (HC) and CF patients before treatment (Base) and call it **readcount**. Use the *third* column (EntrezID) in the original file as the rownames of readcount.

```
gse124548 <- read.delim("GSE124548.raw.fixed.txt")

#subset gse124548 and keep only the columns with "HC" and BASE, change rownames
geneIDs <- gse124548$EntrezID
col_names <- colnames(gse124548)
new_col_names <- c("Raw_10_HC_Auto_066_237", "Raw_11_Orkambi_006_Base", "Raw_13_HC_Auto_068_239", "Raw_

readcount <- as.data.frame(gse124548[ ,new_col_names])
row.names(readcount) <- geneIDs
```

Step 2 (5pts)

Create a dataframe, called **expgroup**, with one column, labeled **condition**, that correctly assigns which column is healthy and which is CF. Use the column names of readcount as rownames of expgroup.

```
expgroup <- data.frame(condition = c(1:40))
row.names(expgroup) <- new_col_names

#HC == healthy
#base == CF
```

```
expgroup[grep("HC", row.names(expgroup)), 1] = "healthy"
expgroup[grep("Base", row.names(expgroup)), 1] = "CF"
```

Step 3 (5pts)

Load the Deseq2 package (install if necessary) and create a Counts Dataset Matrix using the command **DESeqDataSetFromMatrix()** and call it **cds**.

```
library(DESeq2)
```

```
## Loading required package: S4Vectors

## Loading required package: stats4

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which.max, which.min

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

## The following object is masked from 'package:grDevices':
##
##     windows

## Loading required package: GenomicRanges
```

```
## Loading required package: GenomeInfoDb

## Loading required package: SummarizedExperiment

## Loading required package: MatrixGenerics

## Loading required package: matrixStats

##
## Attaching package: 'MatrixGenerics'

## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

##
## Attaching package: 'Biobase'

## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians

## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians
```

```
cds <- DESeqDataSetFromMatrix(countData = readcount, colData = expgroup, design = ~ condition)
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

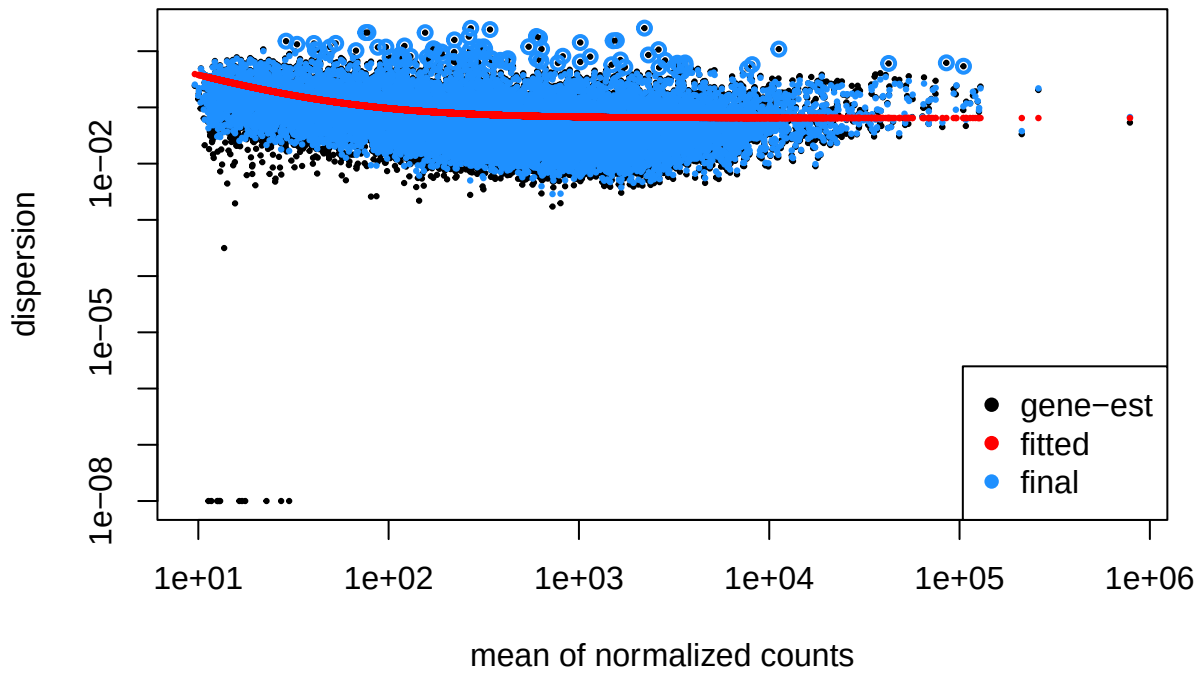
Step 4 (5pts)

Use the functions **estimateSizeFactors** and **estimateDispersions** so that Deseq2 can correct for size of library and estimates the dispersion. Plot the dispersion using **plotDispEsts**. What does the graph tell you? - The Dispersion Estimates graph shows the estimated dispersion value for the gene's expression strength between our two samples (healthy and cf). The goal in creating this graph is to visulize how much variation is in the dataset. In the graph below, the read line is the expected dipersion value given an expression strength. We can see in this graph that as the mean of normalized counts increase, the dispersion slightly decreases. According to DESeq2 documentation, this indcates that our data is a good fit for DESeq2 analysis as it doesn't show signs of contamination or other extreme outling data points.

```
library(DESeq2)
cds_factors <- estimateSizeFactors(cds)
cds_dispersion <- estimateDispersions(cds_factors)
```

```
## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates
```

```
plotDispEsts(cds_dispersion)
```



Step 5 (5pts)

Perform the Differential expression and obtain the results using **DESeq** and **results** functions.

```
library(DESeq2)
diffexp_cds <- DESeq(cds)
```

```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing

## -- replacing outliers and refitting for 128 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)

## estimating dispersions

## fitting model and testing
```

```
diffexp_res_cds <- results(diffexp_cds)
diffexp_res_cds
```

```
## log2 fold change (MLE): condition healthy vs CF
## Wald test p-value: condition healthy vs CF
## DataFrame with 12946 rows and 6 columns
##         baseMean log2FoldChange     lfcSE      stat      pvalue        padj
##        <numeric>      <numeric> <numeric> <numeric>   <numeric>   <numeric>
## 1        81.1684      0.0401876 0.1185343  0.339038  0.73458121   0.8162950
## 503538   50.7995      0.0338344 0.1097210  0.308367  0.75780272   0.8345990
## 144571   97.8569     -0.4733766 0.3265152 -1.449784  0.14711863   0.2592701
## 8086    125.5683      0.1852751 0.1368028  1.354322  0.17563363   0.2949479
## 65985   166.8755     -0.1533419 0.0560134 -2.737595  0.00618903   0.0235518
## ...          ...            ...       ...       ...         ...         ...
## 79364    1284.81      -0.212286 0.0816856  -2.59882 9.35437e-03 3.25542e-02
## 79699    2593.21      -0.325019 0.0974538  -3.33511 8.52662e-04 4.80146e-03
## 7791     1291.19      -0.957212 0.1590951  -6.01660 1.78115e-09 1.15253e-07
## 23140    2799.48      -0.393193 0.1204569  -3.26418 1.09781e-03 5.85106e-03
## 26009    1523.41       0.148691 0.0641000   2.31967 2.03586e-02 5.93876e-02
```

Step 6 (5pts)

How many genes have an adjusted p-value of less than 0.05 and log2FoldChange greater than 1 or less than -1 ? Save this list of genes as **diffexpgenes** - 209 genes have an adjusted p-value of less than 0.05 and log2FoldChange greater than 1 or less than -1.

```
library(DESeq2)
#find number of genes
res <- sum(diffexp_res_cds$padj < 0.05 & diffexp_res_cds$log2FoldChange > 1 | diffexp_res_cds$padj < 0.
res
```

```
## [1] 209
```

```
#create dataframe where pvalue is < 0.05 and log2FoldChange > 1 or log2FoldChange < -1
res_genes <- diffexp_res_cds[diffexp_res_cds$padj < 0.05 & diffexp_res_cds$log2FoldChange > 1 | diffexp

diffexpgenes <- row.names(res_genes)
```

Step 7 (5pts)

Get the normalized values of the counts data in cds using the counts() function with option normalized=T and call this normvalues.

```
library(DESeq2)
#needed to use the estimateSizeFactors(cds) object because the cds object wouldn't run with
#the normalized=TRUE option
normvalues <- counts(cds_factors, normalized=TRUE)
```

Step 8 (5pts)

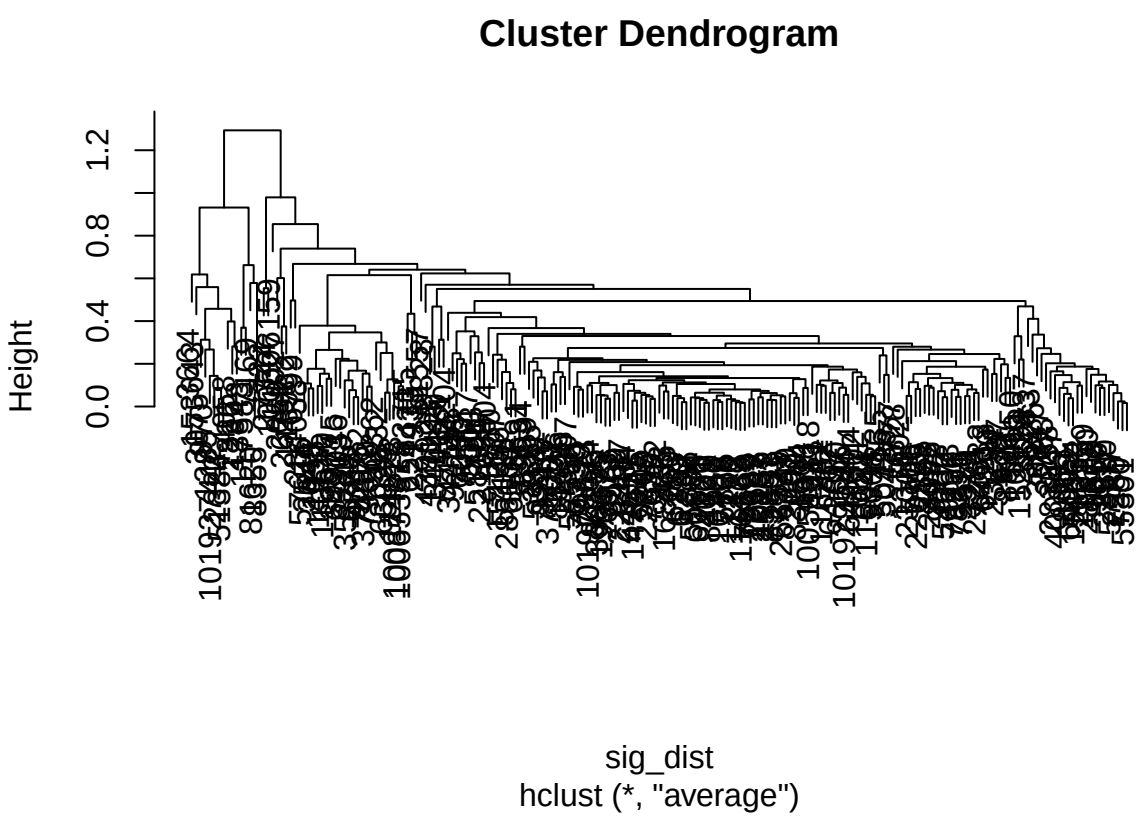
Create a new matrix or dataframe that contains the expression values from normvalues for just the diffexpgenes and call it diffexpvalues.

```
#subset normvalues to have only the diffexpgenes genes
diffexpvalues <- normvalues[diffexpgenes, ]
```

Step 9 (10pts)

Cluster the differentially expressed genes using hierarchical clustering and use the cutree function to create 8 groups. How many genes are in each group? - Group 1 has 185 genes, Group 2 has 2 genes, Group 3 has 2 genes, Group 4 has 11 genes, Group 5 has 3 genes, Group 6 has 3 genes, group 7 has 2 genes, and Group 8 has 1 gene. While the distribution size is inconsistent, this is expected when clustering in large group sizes.

```
library(cluster)
#create an hclust object with the diffexpvalues matirx
sig_dist = as.dist(1 - cor(t(diffexpvalues)))
sig_hclust = hclust(sig_dist, method="average")
plot(sig_hclust)
```




```
sig_hclust_8 = cutree(sig_hclust, k=8)
head(sig_hclust_8)
```

```
## 154664   2180   8728  30817 222487    133
##      1      1      1      1      1      1
```

```
#get genes from each group
sig_hclust_g1= diffexpvalues[names(which(sig_hclust_8==1)),]
sig_hclust_g2= diffexpvalues[names(which(sig_hclust_8==2)),]
sig_hclust_g3= diffexpvalues[names(which(sig_hclust_8==3)),]
sig_hclust_g4= diffexpvalues[names(which(sig_hclust_8==4)),]
sig_hclust_g5= diffexpvalues[names(which(sig_hclust_8==5)),]
sig_hclust_g6= diffexpvalues[names(which(sig_hclust_8==6)),]
sig_hclust_g7= diffexpvalues[names(which(sig_hclust_8==7)),]
sig_hclust_g8= diffexpvalues[names(which(sig_hclust_8==8)),]

nrow(sig_hclust_g1)
```

```
## [1] 185
```

```
nrow(sig_hclust_g2)
```

```
## [1] 2
```

```
nrow(sig_hclust_g3)
```

```
## [1] 2
```

```
nrow(sig_hclust_g4)
```

```
## [1] 11
```

```
nrow(sig_hclust_g5)
```

```
## [1] 3
```

```
nrow(sig_hclust_g6)
```

```
## [1] 3
```

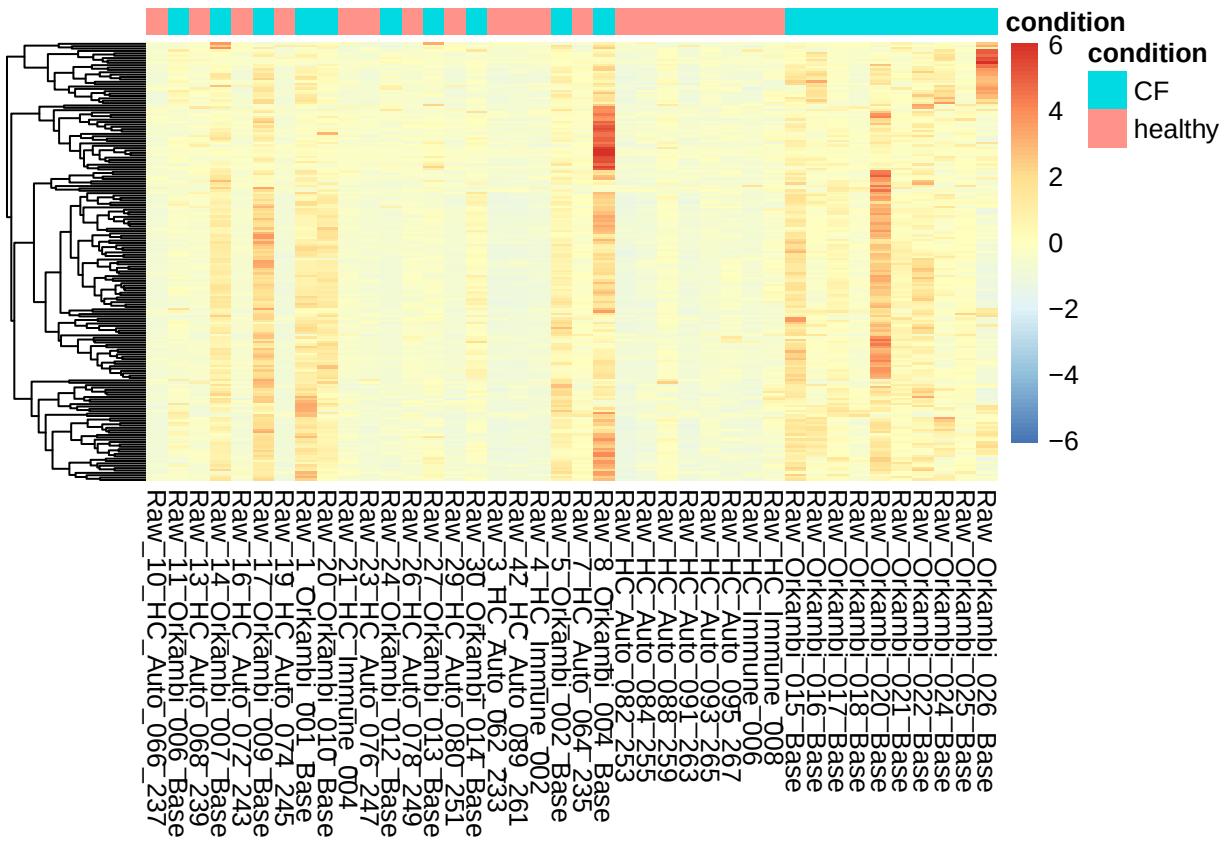
```
nrow(sig_hclust_g7)
```

```
## [1] 2
```

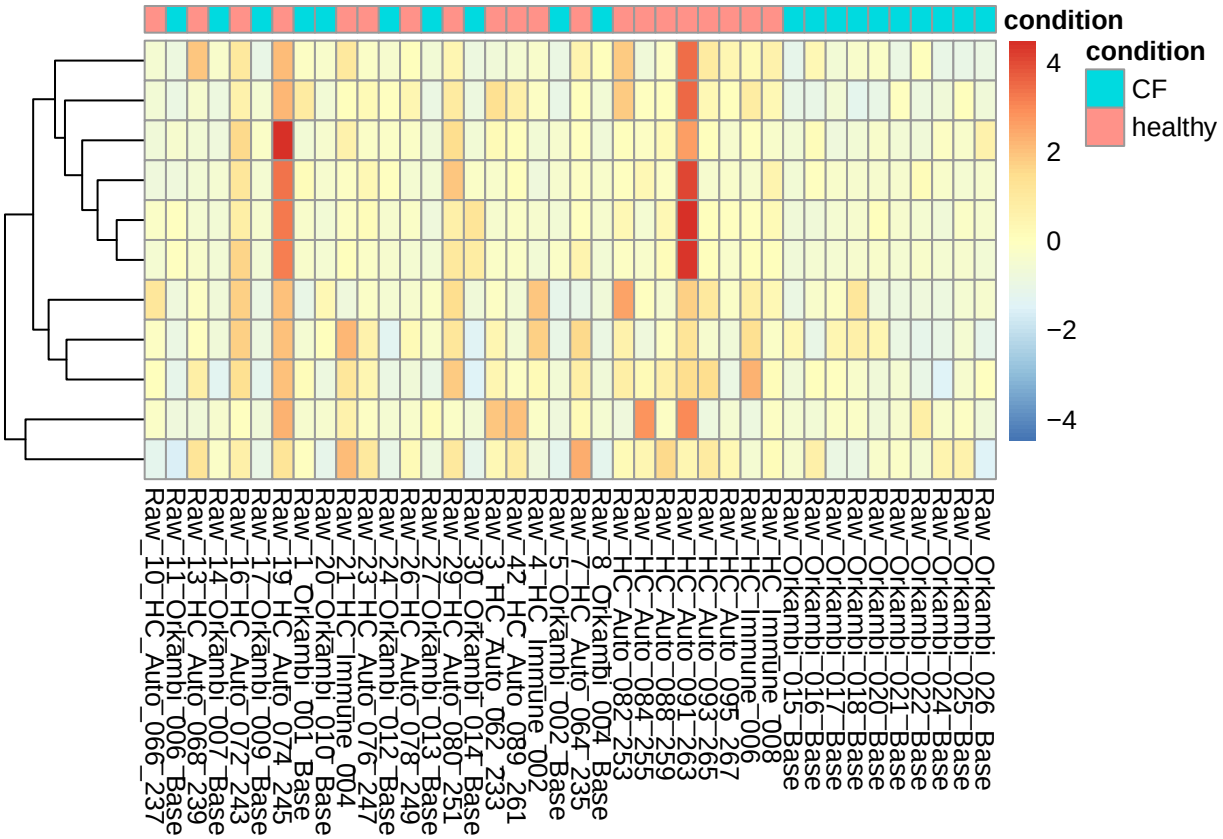
```
nrow(sig_hclust_g8)
```

```
## NULL
```

```
#showing pheat map of group 1
library(pheatmap)
pheatmap(sig_hclust_g1,annotation_col = expgroup,
         scale="row", cluster_cols = F, show_rownames = F)
```




```
#showing pheat map of group 4
pheatmap(sig_hclust_g4,annotation_col = expgroup,
         scale="row", cluster_cols = F, show_rownames = F)
```

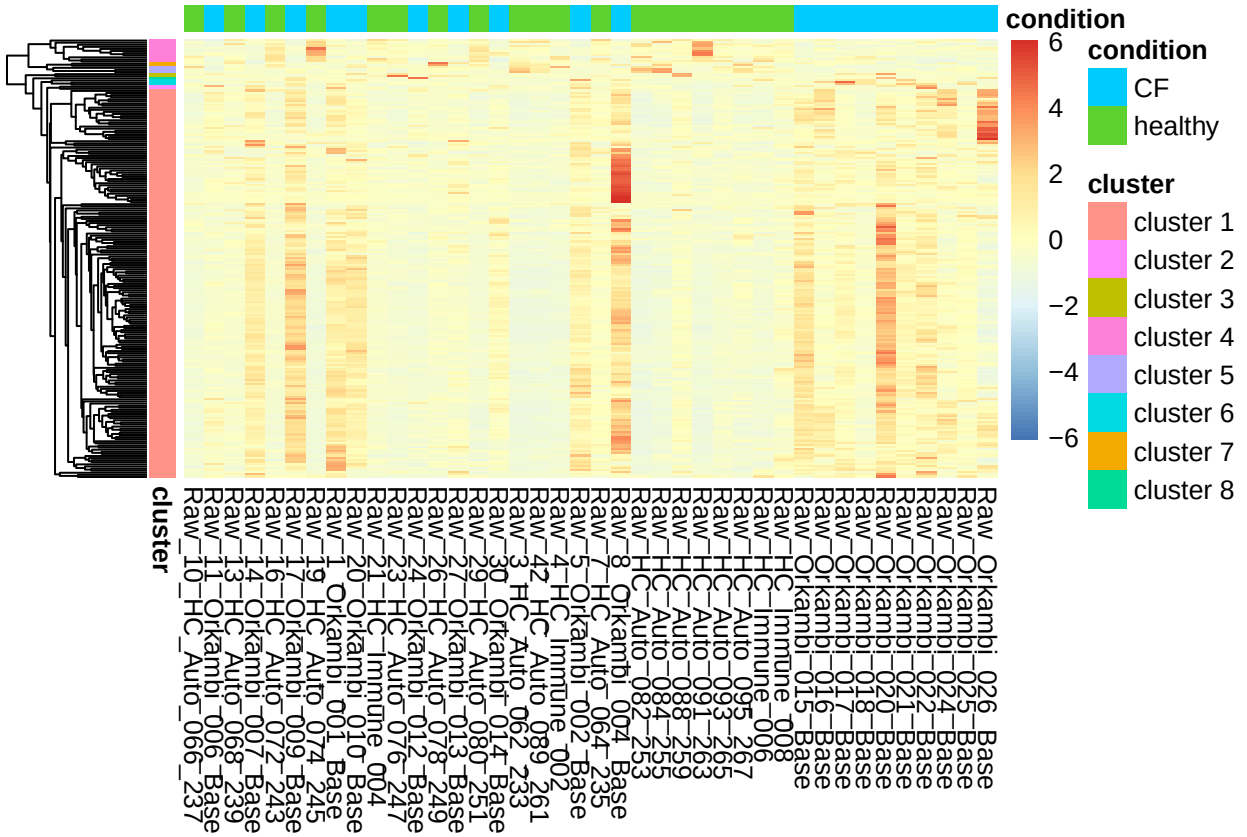


Step 10 (10pts)

Create a heatmap of all the genes using the **pheatmap** package. Some options you should pay attention to are: - scale (scaling gene values shows expression pattern changes rather than magnitude) - cluster_rows (define how the rows should be clustered) - annotation_col (color code the different samples based on a category- health or cf) - annotation_row (color code the different genes based on the cluster they belong to)

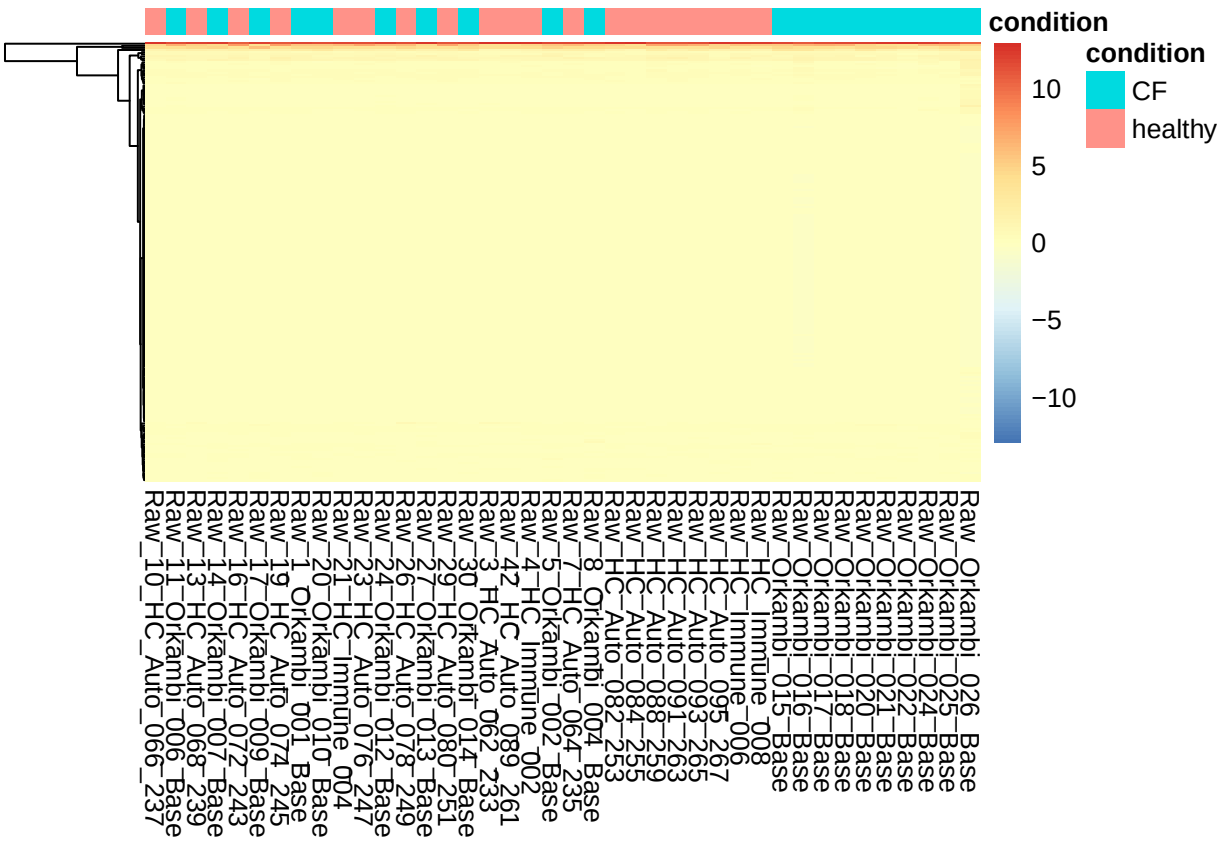
```
library(pheatmap)
#defining a dataframe to use as row annotation for cluster

#expgroup[grep("HC", row.names(expgroup)), 1] = "healthy"

my_gene_col <- data.frame(cluster = ifelse(test = sig_hclust_8 == 1, yes = "cluster 1",
                                           no = ifelse(test=sig_hclust_8 == 2, yes = "cluster 2",
                                                       no = ifelse(test = sig_hclust_8 == 3, yes = "clus
                                                                   no = ifelse(test = sig_hclust_8 == 4
                                                                               no = ifelse(test = sig_h
                                                                                           no = ifelse(
```

```
#heatmap with normalized diff expressed genes, ungrouped, added annotation of
#'healthy' or 'CF', added cluster annotation
pheatmap(diffexpvalues, annotation_col = expgroup, annotation_row = my_gene_col,
         scale="row", cluster_cols = F, show_rownames = F, clustering_method = "average")
```

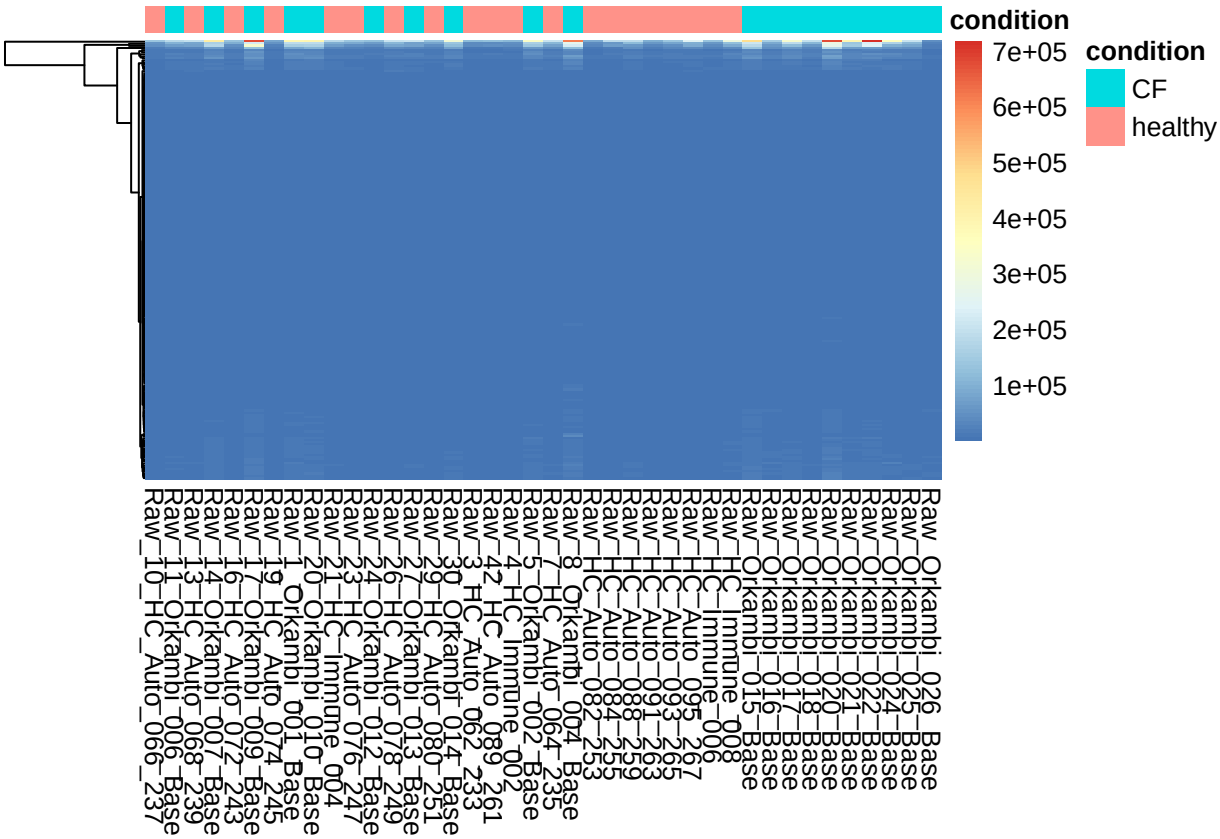


Below are examples of heatmaps with different adjustments of scale, and row clustering

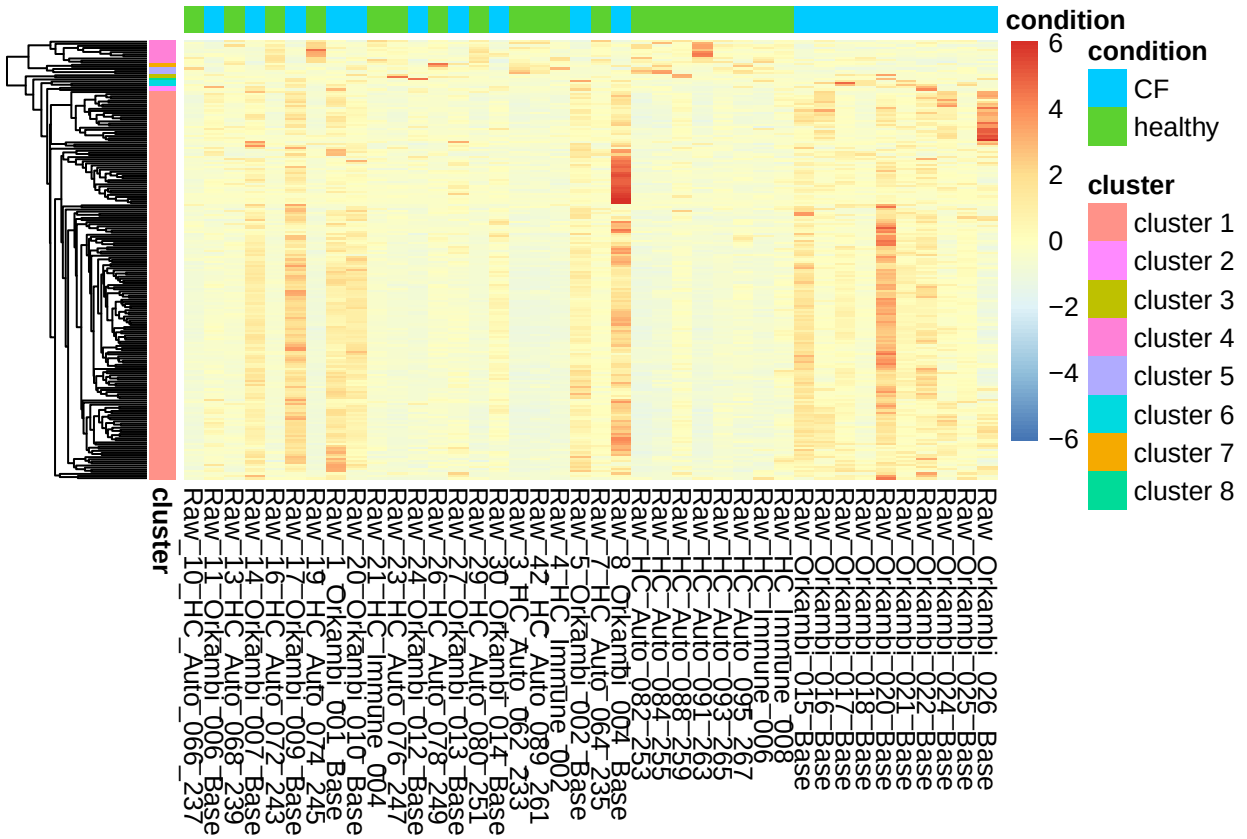
```
#adjusting scale to 'column' with normalized diff expressed genes, ungrouped, added annotation of
#'healthy' or 'CF', added cluster annotation
pheatmap(diffexpvalues,annotation_col = expgroup,
         scale="column", cluster_cols = F, show_rownames = F)
```



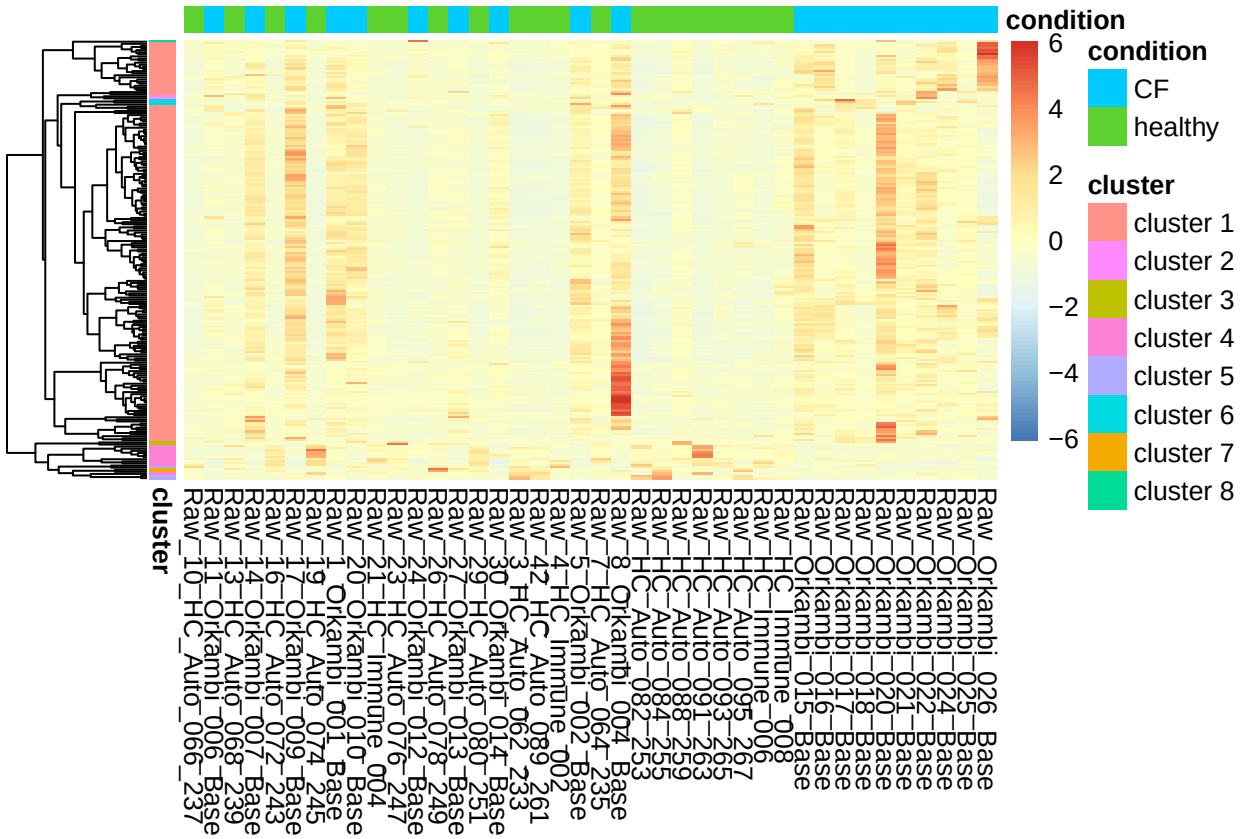

```
#adjusting scale to 'none' with normalized diff expressed genes, ungrouped, added annotation of
#'healthy' or 'CF', added cluster annotation
pheatmap(diffexpvalues,annotation_col = expgroup,
         scale="none", cluster_cols = F, show_rownames = F)
```




```
#clustering rows by euclidean distance with normalized diff expressed genes, ungrouped, added annotatio
#'healthy' or 'CF', added cluster annotation
pheatmap(diffexpvalues, annotation_col = expgroup, annotation_row = my_gene_col,
         scale="row", cluster_cols = F, clustering_method = "average",
         cluster_rows = T, clustering_distance_rows = "euclidean", show_rownames = F)
```




```
#clustering rows by correlation distance with normalized diff expressed genes, ungrouped, added annotat
#'healthy' or 'CF', added cluster annotation
pheatmap(diffexpvalues, annotation_col = expgroup, annotation_row = my_gene_col,
         scale="row", cluster_cols = F,
         cluster_rows = T, clustering_distance_rows = "correlation", show_rownames = F)
```



Step 11 (10pts)

Use the GOstats package to determing which GO-terms are enriched in **diffexpgenes**. To do this you will need to install the following packages from Bioconductor: - Note: I downloaded the necessary libraries in the console of Rstudio and removed the code chunk

Now create a new **GOHyperGParams** object using the **new()** function and save it as variable called **params**. The **geneIds** is **diffexpgenes**. These should be the EntrezIDs we made into rownames in the beginning. The **universeGeneIds** is the rownames of **readcount**. The annotation is **org.Hs.eg** and the ontology we are going to use is **BP** with a pvaluecutoff of 0.001 and our testDirection is **over**.

```
library(GOstats)
```

```
## Loading required package: Category
```

```
## Loading required package: AnnotationDbi
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following object is masked from 'package:S4Vectors':
##
##     expand
```

```
## Loading required package: graph
```

```
##
```

```
##
## Attaching package: 'GOstats'
```

```
## The following object is masked from 'package:AnnotationDbi':
##
##     makeGOGraph
```

```
library(GO.db)
library(Category)
library(org.Hs.eg.db)
```

```
##
```

```
params <- new("GOHyperGParams",
  geneIds=diffexpgenes,
  universeGeneIds=rownames(readcount),
  annotation="org.Hs.eg",
  ontology="BP",
  pvalueCutoff=0.001,
  conditional=TRUE,
  testDirection="over")


overRepresented <- hyperGTest(params)
sum_rep <- summary(overRepresented)[,c(1,2,5,6,7)]

#sort the column function to show genes with highest count number
#show top 10 highest gene counts
overRepresented_Count <- sum_rep[order(-sum_rep$Count), ]
head(overRepresented_Count, n=10)
```

```
##        GOBPID       Pvalue Count Size
## 12 GO:0023052 4.596912e-05    89 3798
## 23 GO:0007154 2.385101e-04    87 3843
## 22 GO:0048513 2.338567e-04    56 2166
## 9  GO:0051239 2.935438e-05    51 1767
## 1  GO:0006952 1.002952e-08    45 1115
## 27 GO:0007166 3.258688e-04    39 1423
## 7  GO:0044419 2.213205e-05    37 1104
## 3  GO:0043207 2.691732e-06    36  965
## 31 GO:0007155 4.808253e-04    29  913
## 15 GO:0050776 9.314806e-05    25  663
##                                                                    Term
## 12                                                            signaling
## 23                                                 cell communication
## 22                                          animal organ development
## 9                  regulation of multicellular organismal process
## 1                                                   defense response
```

```
## 27                                    cell surface receptor signaling pathway
## 7  biological process involved in interspecies interaction between organisms
## 3                                       response to external biotic stimulus
## 31                                                             cell adhesion
## 15                                             regulation of immune response
```

Step 12 (5pts)

What conclusions can you make about your analysis? - Based on the GO Terms, it seems that a majority of immune response and cell defense pathways are being expressed differently between the 'healthy' and 'cf' patients. The increased defense immune response may also be a result of infection from CF patients as mucus in the longs is retaining harmful bacteria and viruses. Signaling and cell communication are the two highest counts of genes, and this could be due to because the fact that there is a change in cell state or cell activity that requires more or less RNA to be produced. It would be necessary to see the variation between treated patients to pinpoint how the expression of immune response is changed and expression in cellular communication is changed.