EGCG blood sample gene expression Analysis

Janis Corona

12/16/2019

This Rmarkdown document analyzes the gene expression data from blood samples of overweight people before being treated with EGCG pills twice a day for 70 days and after. There were some that were treated additionally with flavanoids, fish oil, and vitamin C in the quenterin samples. This data was pulled from the NCBI GEO gene expression data and combined with the HGNC names from BioMart of Ensembl using the Affymetrix human genome 1.0 version 1 ID names.Files for this series are found at NCBI: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE74560

Read in all the data tables.

```
pre_egcg <- read.csv('pre_placebo_EGCG.csv',sep=',', header=TRUE,
                     na.strings=c('',' '), skip=1)
pre_egcg_quer <- read.csv('pre_vitC_flavanoids_fishOil_EGCG.csv',
                          sep=',', header=TRUE, na.strings=c('',' '), skip=1)
post_egcg <- read.csv('post_placebo_EGCG.csv', sep=',', header=TRUE,
                      na.strings=c('',' '), skip=1)
post_egcg_quer <- read.csv('post_vitaminC_flavanoids_fishOil_EGCG.csv',
                     sep=',', header=TRUE, na.strings=c('',' '), skip=1)
hgnc <- read.csv('HGNC_bioMart_affy_1_v1_IDs.csv', sep=',',
                 header=TRUE, na.strings=c('',' '))
```

Combine the HGNC names to each of the four tables and make the table row names.

```
HGNC <- hgnc[,4:5]
HGNC <- HGNC[!duplicated(HGNC$Affy_HuGene_1_v1_probe_ID),]
colnames(HGNC)[2] <- 'Affy'

post_egcg2 <- merge(HGNC,post_egcg, by.x='Affy',
                    by.y='ID_REF')

post_egcg_quer2 <- merge(HGNC, post_egcg_quer,
                         by.x='Affy',by.y='ID_REF')

pre_egcg2 <- merge(HGNC, pre_egcg, by.x='Affy',
                   by.y='ID_REF')

pre_egcg_quer2 <- merge(HGNC, pre_egcg_quer,
                        by.x='Affy',
                        by.y='ID_REF')
```

There are duplicate HGNC names because of the different Affymetrix IDs, this can be fixed by grouping by genes and getting the mean of those genes and creating a table for each

gene by means using the dplyr package.This is time consuming to run because it groups all the genes then it gets the means for each group by sample and outputs that table. It takes 20-30 minutes for each table grouping by genes and then taking the means of each gene per sample.

```
library(dplyr)
```

Remove the Affymetrix ID from the four tables because it isn't needed anymore.

```
post_eq <- post_egcg_quer2[,-1]
post_eg <- post_egcg2[,-1]
pre_eq <- pre_egcg_quer2[,-1]
pre_eg <- pre_egcg2[,-1]
```

EGCG only pre-treatment samples

```
samples <- colnames(pre_eg)[2:8]
Samples <- as.vector(samples)
Pre_E <- pre_eg %>% group_by(HGNC.Symbol) %>%
  summarise_at(vars(Samples), mean, na.rm=TRUE)
```

EGCG+quenterin pre-treatment samples

```
samples1 <- colnames(pre_eq)[2:8]
Samples1 <- as.vector(samples1)
Pre_Q <- pre_eq %>% group_by(HGNC.Symbol) %>%
  summarise_at(vars(Samples1), mean, na.rm=TRUE)
```

EGCG samples and EGCG+flavanoids+fish oil+vitamin C can be combined into a pretreatment table.

```
pre <- merge(Pre_E, Pre_Q, by.x='HGNC.Symbol', by.y='HGNC.Symbol')
row.names(pre) <- pre$HGNC.Symbol
PRE <- pre[,-1]
```

Post EGCG and Quercetin

```
samples2 <- colnames(post_eq)[2:8]
Samples2 <- as.vector(samples2)
Post_Q <- post_eq %>% group_by(HGNC.Symbol) %>%
  summarise_at(vars(Samples2), mean, na.rm=TRUE)
```

Post EGCG only

```
samples3 <- colnames(post_eg)[2:8]
Samples3 <- as.vector(samples3)
Post_E <- post_eg %>% group_by(HGNC.Symbol) %>%
  summarise_at(vars(Samples3), mean, na.rm=TRUE)
```

Give the post-treatment samples row names and remove the HGNC fields

```
Post_E <- as.data.frame(Post_E)
Post_Q <- as.data.frame(Post_Q)
```

```
row.names(Post_E) <- Post_E$HGNC.Symbol
row.names(Post_Q) <- Post_Q$HGNC.Symbol
Post_E <- Post_E[,-1]
Post_Q <- Post_Q[,-1]
```

Write the new tables to csv files

```
write.csv(PRE, 'pre_combined.csv', row.names=TRUE)
write.csv(Post_E, 'Post_EGCG_combine.csv', row.names=TRUE)
write.csv(Post_Q, 'Post_EGCG_Quer_combine.csv', row.names=TRUE)
```

To save time, the above has been commented out, and the files can be opened to run the rest of the code.

```
PRE <- read.csv('pre_combined.csv', sep=',', header=TRUE,
                na.strings=c('',' '), row.names=1)
Post_E <- read.csv('Post_EGCG_combine.csv', sep=',', header=TRUE,
                   na.strings=c('',' '), row.names=1)
Post_Q <- read.csv('Post_EGCG_Quer_combine.csv', sep=',', header=TRUE,
                   na.strings=c('',' '), row.names=1)
```

Get the means of the pre-treatment samples with rowMeans() per gene and the post-treatment samples of EGCG and EGCG+Quercentin, then attach to each of those tables as the mean of those samples of gene expression values.

```
PRE$PRE_Means <- rowMeans(PRE)
Post_Q$Post_Q_Means <- rowMeans(Post_Q)
Post_E$Post_E_Means <- rowMeans(Post_E)

colnames(PRE)[1:14] <- paste('pre_',sep='', colnames(PRE)[1:14])

colnames(Post_E)[1:7] <- paste('post_EG_', sep='',colnames(Post_E)[1:7])

colnames(Post_Q)[1:7] <- paste('post_EQ_', sep='', colnames(Post_Q)[1:7])
```

Combine all tables into one with the means of each type of sample at the start of the columns

```
All <- cbind(PRE[,15],Post_E[,8],Post_Q[,8],PRE[,1:14], Post_E[1:7],
             Post_Q[1:7])
colnames(All)[1:3] <-
c('Pre_Means','Post_EGCG_Means','Post_EGCG_Quercentin_Means')
```

Now for the differential expression in Means using dplyr. Here the overweight is the initial value minus the value after treatment type.

```
DE <- All %>% mutate(DE_EGCG=All$Pre_Means-All$Post_EGCG_Means)
DE2 <- DE %>% mutate(DE_Quercentin=DE$Pre_Means-
All$Post_EGCG_Quercentin_Means)

DE3 <- DE2[,c(32,33,1:31)] #table with both differential values
row.names(DE3) <- row.names(All)
```

Get the over and under expressed genes after treatment compared to before treatment of EGCG and EGCG+quercentin in overweight females.The negative DE values means the treatment made the gene over expressed, since the initial values minus treatment values would be negative only if treatment values are greater than initial values.These two tables are the top20 up regulated genes after treatment with either EGCG or EGCG+Quercentin

```
DE_20up_egcg <- DE3[order(DE3$DE_EGCG, decreasing=FALSE)[0:20],]
DE_20up_egcg_quer <- DE3[order(DE3$DE_Quercentin,decreasing=FALSE)[0:20],]
```

The top 20 up regulated genes after treatment with only EGCG

```
row.names(DE_20up_egcg)
```

```
##  [1] "TMEM176A"  "HBZ"       "TMEM176B"  "FAU"       "HBZP1"     "RNY3P16"
##  [7] "IGKV1-5"   "RNA5SP63"  "TREML4"    "CYP4F35P"  "ALOX15"    "IGKV3-
11"
## [13] "IGKV2-30"  "ZFP57"     "RNA5SP449" "IGKV1-33"  "CAV1"      "RNU6-
893P"
## [19] "POU2AF1"   "IGKV2-24"
```

The top 20 up regulated genes after treatment with EGCG and Quercenting

```
row.names(DE_20up_egcg_quer)
```

```
##  [1] "KARSP3"     "GSTM1"      "RNU4-51P"   "RNU6-1024P" "COX7B"
##  [6] "OCLN"       "CTNNAL1"    "RNA5SP30"   "C4BPA"      "CD274"
## [11] "LINC01356"  "RNU6-522P"  "HMGB3P24"   "BBOF1"      "RPL36AP26"
## [16] "SLPI"       "RNU6-943P"  "SLC3A1"     "RPS7"       "RNU6-853P"
```

Now look at the top 20 down regulated genes after treatment with EGCG or EGCG and Quercentin

```
DE_20down_egcg <- DE3[order(DE3$DE_EGCG, decreasing=TRUE)[0:20],]
DE_20down_egcg_quer <- DE3[order(DE3$DE_Quercentin,decreasing=TRUE)[0:20],]
```

The top 20 genes down regulated in overweight females after treatment with only EGCG are:

```
row.names(DE_20down_egcg)
```

```
##  [1] "RNU4-2"    "TMTC1"     "SLFN14"    "SLC14A1"   "GSTM1"
"SCARNA10"
##  [7] "RNF182"    "LINC00189" "XK"        "TSPAN7"    "SFRP2"     "ORM1"
## [13] "LRP1"      "RPPH1"     "ALDH5A1"   "LGALS2"    "IFIT1B"
"RNA5SP242"
## [19] "UBE2O"     "OSBP2"
```

The top 20 genes down regulated in overweight females after treatment with EGCG and Quercentin are:

```
row.names(DE_20down_egcg_quer)
```

```
##  [1] "HBZ"         "RNU6-1124P" "RNU6-921P"  "IGKV2-26"   "IGKV1D-27"
##  [6] "ALOX15"      "TMEM176A"   "RNU6-1315P" "KIAA1324"   "IL1RL1"
## [11] "SNORA38B"    "HLA-DOA"    "IGHV3-47"   "VTRNA1-1"   "SMPD3"
## [16] "TMEM176B"    "SNORA23"    "RNU5A-1"    "FOLR3"      "RN7SL218P"
```

Now for the fold change in genes to see which genes changed the most in samples after treatment with EGCG or EGCG and Quercentin using the ratio of treatment type value/initial value.

```
FC <- DE3 %>% mutate(FC_egcg = DE3$Post_EGCG_Means/DE3$Pre_Means)
FC <- FC[,c(34,1:33)]
row.names(FC) <- row.names(DE3)

FC2 <- FC %>% mutate(FC_egcg_quer =
FC$Post_EGCG_Quercentin_Means/FC$Pre_Means)
FC2 <- FC2[,c(35,1:34)]
row.names(FC2) <- row.names(FC)

FC3 <- round(FC2,3)
write.csv(FC3,'foldChange_EGCG.csv', row.names=TRUE)
```

The top 20 genes that changed the most have a higher fold change value and are:

```
FC_top20_egcg <- FC2[order(FC2$FC_egcg, decreasing=TRUE)[0:20],]
FC_top20_egcg_quer <- FC2[order(FC2$FC_egcg_quer, decreasing=TRUE)[0:20],]
```

The top 20 genes with the most fold change in EGCG treatment only are:

```
row.names(FC_top20_egcg)
```

```
##  [1] "HBZ"         "FAU"         "RNA5SP449"  "RNY3P16"    "MTCO2P29"
##  [6] "TMEM176A"    "TMEM176B"    "PSG1"       "CYP4F35P"   "RN7SKP198"
## [11] "RNA5SP63"    "NPY5R"       "MIR548AD"   "RN7SL452P"  "GPR87"
## [16] "RNU6-1024P"  "BUD31P2"     "RNU6-893P"  "ZNF578"     "RNU6-263P"
```

The top 20 genes with the most fold change in EGCG and Quercentin treatment are:

```
row.names(FC_top20_egcg_quer)
```

```
##  [1] "KARSP3"      "RNU4-51P"    "RNU6-1244P" "RNU6-403P"  "RNU6-1024P"
##  [6] "COX7B"       "ZNF847P"     "RNU6-853P"  "RNU6-943P"  "RNA5SP30"
## [11] "OR5M10"      "RNY1P5"      "DDX18P6"    "HMGB3P24"   "LINC01356"
## [16] "MTCO1P11"    "RPS7"        "RNU6-98P"   "FAU"        "GSTM1"
```

Create a data table of the statistical values only

```
Statistical <- FC2[,1:7] #all genes, not filtered for top 20
```

Create a table of all the samples with all the genes that are in some top 20 category from the statistical values produced earlier.

```
Fold1 <- FC_top20_egcg[,c(2,8:35)]
colnames(Fold1)[1] <- 'change'
Fold1$value <- rep('Fold Change EGCG',20)

Fold2 <- FC_top20_egcg_quer[,c(1,8:35)]
colnames(Fold2)[1] <- 'change'
Fold2$value <- rep('Fold Change EGCG+Quercentin',20)

DE_down_1 <- DE_20down_egcg[,c(1,6:33)]
colnames(DE_down_1)[1] <- 'change'
DE_down_1$value <- rep('Down Regulated EGCG',20)

DE_down_2 <- DE_20down_egcg_quer[,c(2,6:33)]
colnames(DE_down_2)[1] <- 'change'
DE_down_2$value <- rep('Down Regulated EGCG+Quercentin',20)

DE_up_1 <- DE_20up_egcg[,c(1,6:33)]
colnames(DE_up_1)[1] <- 'change'
DE_up_1$value <- rep('Up Regulated EGCG',20)

DE_up_2 <- DE_20up_egcg_quer[,c(2,6:33)]
colnames(DE_up_2)[1] <- 'change'
DE_up_2$value <- rep('Up Regulated EGCG+Quercentin',20)

Top20 <- rbind(Fold1, Fold2, DE_down_1, DE_down_2, DE_up_1, DE_up_2)
Top20 <- Top20[,c(30,1:29)]
```

Write the previous table out to csv with the change value and corresponding type measure each of the genes in a top 20 category produced.

```
write.csv(Top20, 'Top20_all.csv', row.names=TRUE)
```

Now make the table of top genes (120) and the samples transposed with a new field for type of sample, so that it can be used to run machine learning on and predict the sample type the observation was from as in pre treatment, post treatment with EGCG or post treatment with EGCG and a combination of Vitatmin C, Niacin, and Quercentin fish oil.

```
Top20_1 <- Top20[, -c(1:2)]

ML_Top20 <- t(Top20_1)

pre <- as.data.frame(rep('pre-treatment',14))
post_eg <- as.data.frame(rep('EGCG treatment',7))
post_eq <- as.data.frame(rep('EGCG and Quercentin treatment', 7))

colnames(pre) <- 'Type'
colnames(post_eg) <- 'Type'
colnames(post_eq) <- 'Type'
```

```
Type <- rbind(pre, post_eg, post_eq)

genes <- Top20[,1:2]
write.csv(genes, 'genes_120_top20.csv', row.names=TRUE)

ML_ready <- cbind(Type, ML_Top20)

write.csv(ML_ready, 'ML_ready.csv', row.names=TRUE)
```

Some gene summaries were obtained from NCBI gene and from Genecard.org, but some summaries only state the gene is a pseudogene and some have no results. Read in this table that the summaries were added in Excel from the sites above and keep only the complete cases.

```
geneSummaries <- read.csv('genes_120_top20_summaries.csv', sep=',',
                          header=TRUE, na.strings=c('',' '))

geneSumm <- geneSummaries[complete.cases(geneSummaries$summary),]
```

Combine the up and fold change top genes as they are both the genes that were up regulated the most when building those tables, the fold change for genes down regulated wasn't made. Then make the down regulated genes it's own table

```
u <- geneSumm[grep('Up',geneSumm$value),]
f <- geneSumm[grep('Fold',geneSumm$value),]

up <- rbind(u,f)#the negative values for up reg. can be confusing, but it is
how much less the healthy genes are less than this diseased gene

down <- geneSumm[grep('Down', geneSumm$value),]
```

Remove the titles of 'GeneCards Summary' and 'Gene Ontology (GO) annotations' and 'UniProtKB/Swiss-Prot Summary' and 'PubMed' so that the actual summary can be made for the up and down regulated genes separately.

```
up$summary <- gsub('UniProtKB/Swiss-Prot Summary','',up$summary)
up$summary <- gsub('GeneCards Summary','',up$summary)
up$summary <- gsub('Gene Ontology','',up$summary)
up$summary <- gsub('(GO)','',up$summary)
up$summary <- gsub('annotations','',up$summary)
up$summary <- gsub('PubMed','',up$summary)

down$summary <- gsub('UniProtKB/Swiss-Prot Summary','',down$summary)
down$summary <- gsub('GeneCards Summary','',down$summary)
down$summary <- gsub('Gene Ontology','',down$summary)
down$summary <- gsub('(GO)','',down$summary)
down$summary <- gsub('annotations','',down$summary)
down$summary <- gsub('PubMed','',down$summary)
```

Perform Natural Language Processing on the Up regulated gene summaries and produce some word clouds for visualizations to get an idea of the gene functions that are associated with green tea effects on overweight females.

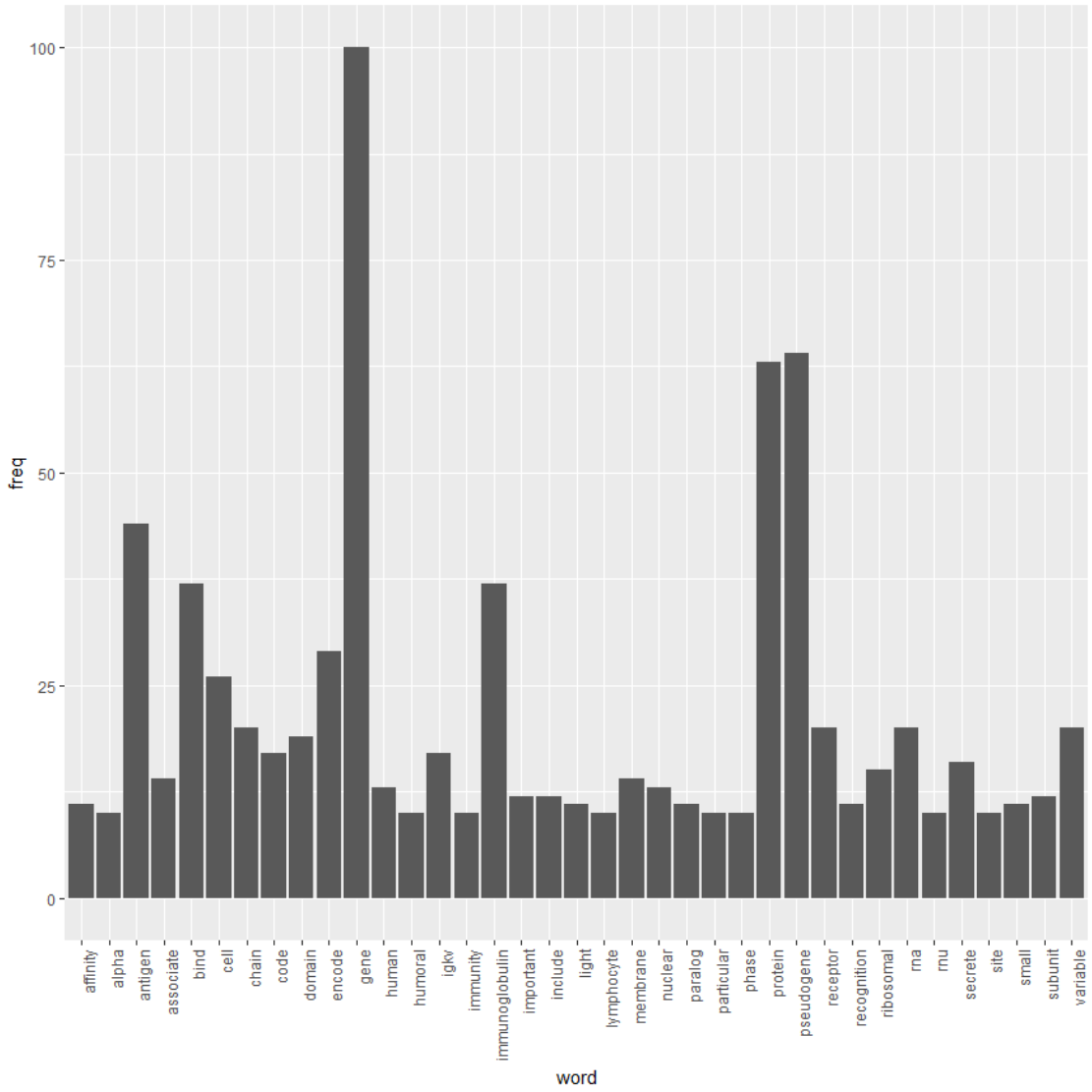
```
library(tm)
library(SnowballC)
library(wordcloud)
library(ggplot2)
library(textstem)

lemma <- lemmatize_strings(up$summary, dictionary=lexicon::hash_lemmas)

Lemma <- as.data.frame(lemma)
Lemma <- cbind(Lemma, up)

colnames(Lemma)[1] <- 'lemmatized_summary'

write.csv(Lemma, 'Lemmatized_upreg.csv', row.names=FALSE)

dir.create('./upreg20-Lemma')

ea <- as.character(Lemma$lemmatized_summary)
setwd('./upreg20-Lemma')

for (j in 1:length(ea)){
  write(ea[j], paste(paste('up',j, sep='.'), '.txt', sep=''))
}
setwd('../')

EGCG <- Corpus(DirSource("upreg20-Lemma"))

EGCG

## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 59

EGCG <- tm_map(EGCG, removePunctuation)
EGCG <- tm_map(EGCG, removeNumbers)
EGCG <- tm_map(EGCG, tolower)
EGCG <- tm_map(EGCG, removeWords, stopwords("english"))
EGCG <- tm_map(EGCG, stripWhitespace)

dtmEGCG <- DocumentTermMatrix(EGCG)
dtmEGCG

## <<DocumentTermMatrix (documents: 59, terms: 622)>>
## Non-/sparse entries: 1388/35310
## Sparsity           : 96%
## Maximal term length: 19
## Weighting          : term frequency (tf)
```

```
freq <- colSums(as.matrix(dtmEGCG))

FREQ <- data.frame(freq)
ord <- order(freq, decreasing=TRUE)

freq[head(ord, 25)]
```

```
##           gene     pseudogene        protein        antigen           bind
##            100             64             63             44             37
## immunoglobulin         encode           cell            rna          chain
##             37             29             26             20             20
##       receptor       variable         domain           code           igkv
##             20             20             19             17             17
##        secrete      ribosomal      associate       membrane        nuclear
##             16             15             14             14             13
##          human      important        include        subunit        paralog
##             13             12             12             12             11
```

Up regulated and increasing fold change genes

```
wf <- data.frame(word=names(freq), freq=freq)
p <- ggplot(subset(wf, freq>9), aes(word, freq))
p <- p + geom_bar(stat= 'identity')
p <- p + theme(axis.text.x=element_text(angle=90, hjust=1))
p
```



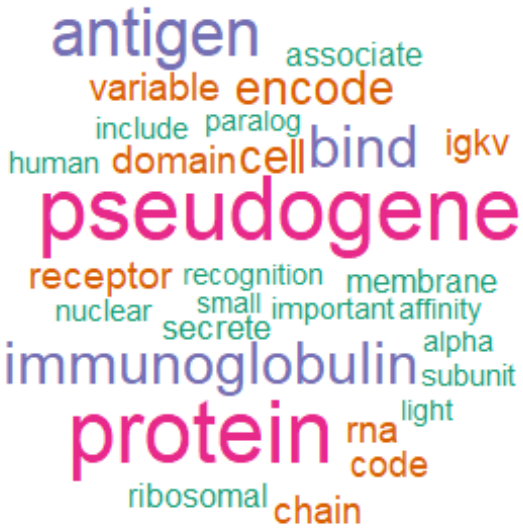
Up regulated and increasing fold change

```
wordcloud(names(freq), freq, min.freq=9,colors=brewer.pal(3,'Dark2'))
```

```
## Warning in wordcloud(names(freq), freq, min.freq = 9, colors =
brewer.pal(3, :
## protein could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(freq), freq, min.freq = 9, colors =
brewer.pal(3, :
## gene could not be fit on page. It will not be plotted.
```

Up regulated and increasing fold change

```
wordcloud(names(freq), freq, max.words=30,colors=brewer.pal(6,'Dark2'))

## Warning in wordcloud(names(freq), freq, max.words = 30, colors =
brewer.pal(6, :
## gene could not be fit on page. It will not be plotted.
```



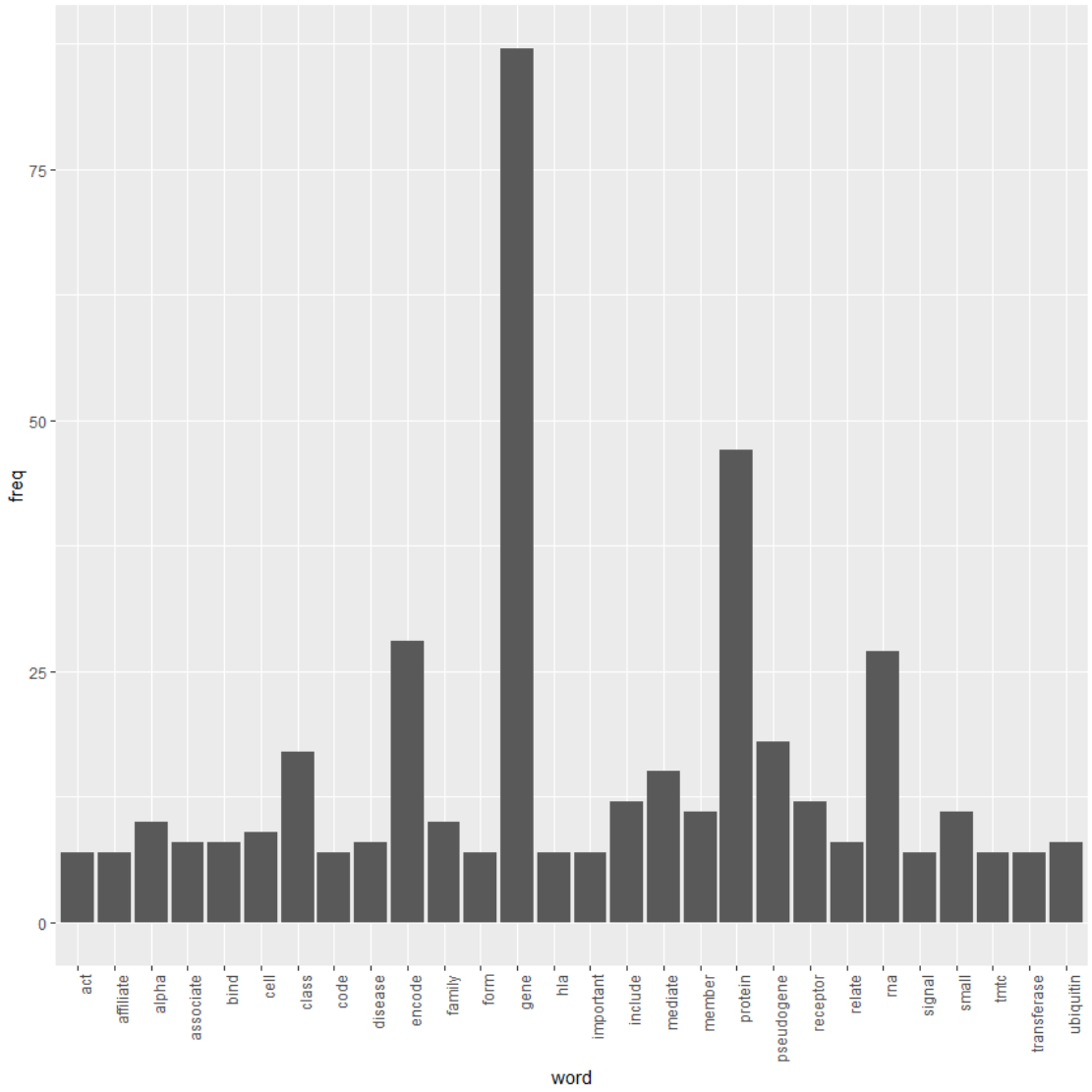
Now for the down regulated available summaries for the top 20 down regulated genes

```
lemma <- lemmatize_strings(down$summary, dictionary=lexicon::hash_lemmas)

Lemma <- as.data.frame(lemma)
Lemma <- cbind(Lemma, down)

colnames(Lemma)[1] <- 'lemmatized_summary'

write.csv(Lemma, 'Lemmatized_downreg.csv', row.names=FALSE)

dir.create('./downreg-Lemma')

ea <- as.character(Lemma$lemmatized_summary)
setwd('./downreg-Lemma')

for (j in 1:length(ea)){
  write(ea[j], paste(paste('down',j, sep='.'), '.txt', sep=''))
}
setwd('../')

EGCG <- Corpus(DirSource("downreg-Lemma"))

EGCG

## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 38

EGCG <- tm_map(EGCG, removePunctuation)
EGCG <- tm_map(EGCG, removeNumbers)
EGCG <- tm_map(EGCG, tolower)
EGCG <- tm_map(EGCG, removeWords, stopwords("english"))
EGCG <- tm_map(EGCG, stripWhitespace)

dtmEGCG <- DocumentTermMatrix(EGCG)
dtmEGCG

## <<DocumentTermMatrix (documents: 38, terms: 556)>>
## Non-/sparse entries: 980/20148
## Sparsity           : 95%
## Maximal term length: 24
## Weighting          : term frequency (tf)

freq <- colSums(as.matrix(dtmEGCG))

FREQ <- data.frame(freq)
ord <- order(freq, decreasing=TRUE)

freq[head(ord, 25)]
```

```
##        gene     protein      encode         rna  pseudogene       class
##          87          47          28          27          18          17
##     mediate     include    receptor      member       small       alpha
##          15          12          12          11          11          10
##      family        cell   associate        bind     disease      relate
##          10           9           8           8           8           8
##   ubiquitin        code   important         act        tmtc transferase
##           8           7           7           7           7           7
##      signal
##           7
```

Down regulated

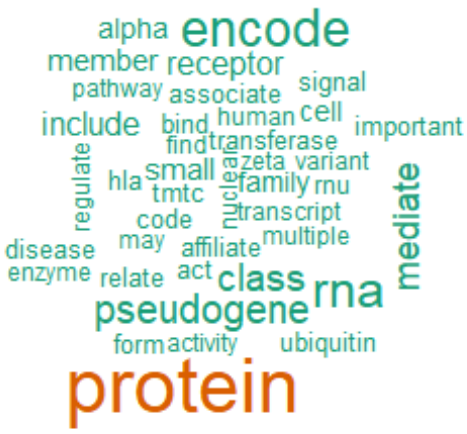
```
wf <- data.frame(word=names(freq), freq=freq)
p <- ggplot(subset(wf, freq>6), aes(word, freq))
p <- p + geom_bar(stat= 'identity')
p <- p + theme(axis.text.x=element_text(angle=90, hjust=1))
p
```



Down regulated

```
wordcloud(names(freq), freq, min.freq=6,colors=brewer.pal(3,'Dark2'))
```

```
## Warning in wordcloud(names(freq), freq, min.freq = 6, colors =
brewer.pal(3, :
## gene could not be fit on page. It will not be plotted.
```



Down regulated

```
wordcloud(names(freq), freq, max.words=30,colors=brewer.pal(6,'Dark2'))
```