BCB420 Assignment 1 Sabbir Hossain

##Background information here, paraphased from the paper. This was a really interesting paper, but I will save that discussion for the later part of this assignment. A prior transcriptome meta-analysis found significantly decreased levels of corticotropin-releasing hormone (CRH) mRNA in corticolimbic brain areas in MDD patients, indicating that cortical CRH-expressing (CRH+) cells are impaired in MDD. Although rodent studies reveal that cortical CRH is predominantly expressed in GABAergic interneurons, little is known about the characteristics of CRH+ cells in the human cerebral cortex and their relationship to MDD. Human volunteers without brain illnesses had their subgenual anterior cingulate cortex (sgACC) identified for CRH and markers of excitatory (SLC17A7), inhibitory (GAD1), and other interneuron subpopulations using fluorescent in situ hybridization (FISH) (PVALB, SST, VIP). Changes in CRH+ cell density and cellular CRH expression (n = 6/group) were investigated in MDD patients. RNA-sequencing was done on sgACC CRH+ interneurons from comparison and MDD participants (n = 6/group) to see if there were any variations between the two groups. In mice with TrkB function suppressed, the effect of decreased BDNF on CRH expression was investigated. GABAergic cells made up 80 percent of CRH+ cells, whereas glutamatergic cells made up 17.5 percent. VIP (52%) and SST (7%), as well as PVALB, were co-expressed by CRH+ GABAergic interneurons (7 percent). MDD patients had lower CRH mRNA levels in GABAergic interneurons than control participants, despite no differences in cell density. The transcriptome profile of CRH+ interneurons suggests decreased excitability and less GABA release and reuptake. Further research revealed that these molecular alterations are not caused by altered glucocorticoid feedback, but rather occur downstream of a common neurotrophic function modulator.

Here is a direct link to the query for this dataset. (GSE193417 (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE193417)).

1. Inquire and then download the required files from GEO.

Check to see if any packages that have been run above are missing or not installed properly.

```
suppressWarnings({if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
if (!requireNamespace("GEOmetadb", quietly = TRUE))
    BiocManager::install("GEOmetadb")
if (!requireNamespace("limma", quietly = TRUE))
    BiocManager::install("limma")
if (!requireNamespace("org.Hs.eg.db", quietly = TRUE))
    BiocManager::install("org.Hs.eg.db")
if (!requireNamespace("edgeR", quietly = TRUE))
    BiocManager::install("edgeR")
if (!requireNamespace("tidyverse", quietly = TRUE))
    install.packages("tidyverse")
if (!requireNamespace("ggplot2", quietly = TRUE))
    install.packages("ggplot2")
if (!requireNamespace("org.Hs.eg.db", quietly = TRUE))
    BiocManager::install("org.Hs.eg.db")
if (!requireNamespace("AnnotationDbi", quietly = TRUE))
    BiocManager::install("AnnotationDbi")
if (!requireNamespace("readxl", quietly = TRUE))
    BiocManager::install("readxl")
if (!requireNamespace("RColorBrewer", quietly = TRUE))
    install.packages("RColorBrewer")
if (!requireNamespace("dplyr", quietly = TRUE))
    install.packages("dplyr")})
```

```
## Setting options('download.file.method.GEOquery'='auto')
```

```
## Setting options('GEOquery.inmemory.gpl'=FALSE)
```

```
##
```

Load in the respective libraries for this assignment.

```
library(Biobase)
```

```
## 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, 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
```

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

```
library(BiocManager)
library(GEOmetadb)
```

```
## Loading required package: GEOquery
```

```
## Loading required package: RSQLite
```

```
library(edgeR)
```

```
## Loading required package: limma
```

```
##
## Attaching package: 'limma'
```

```
## The following object is masked from 'package:BiocGenerics':
##
##     plotMA
```

```
library(biomaRt)
library(magrittr)
library(GEOquery)
library(RSQLite)
library(limma)
library(org.Hs.eg.db)
```

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

```
## Loading required package: stats4
```

```
## Loading required package: IRanges
```

```
## Loading required package: S4Vectors
```

```
##
## Attaching package: 'S4Vectors'
```

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

```
library(umap)
```

Download the GEO metadata, or load it from the disk if it is already has been downloaded before. Local copy is fine as well.

```
if (!file.exists("GSE193417.rds")) {
  GSE193417 <- getGEO("GSE193417", GSEMatrix =TRUE, getGPL= FALSE)
  if (length(GSE193417) > 1) idx <- grep("GPL16791", attr(GSE193417, "names")) else idx <- 1
  GSE193417s <- GSE193417[[idx]]
  saveRDS(GSE193417, "GSE193417.rds")
} else {
  GSE193417 <- readRDS("GSE193417.rds")
}
```

Check to see if we have the dataset. If you do get an error you will need to delete all the files that were downloaded so far, and try again. Packages do not need to be removed.

```
# Checkpoint ...
if (! exists("GSE193417")) {
  stop("PANIC: GSE193417 was not loaded. Or properly formatted.
       Clear everything and run code again from the start. Can't continue.")
}
```

Check to see that there are the correct number of samples in this series. There should be 12 samples in this series at the time of writing this comment. You can double check by running the next code block and looking for the number of series.

```
length(Biobase::sampleNames(GSE193417))
```

```
## [1] 12
```

```
Biobase::sampleNames(GSE193417) #GEO sequence names of all the samples used in this experiment.
```

```
##  [1] "GSM5799928" "GSM5799929" "GSM5799930" "GSM5799931" "GSM5799932"
##  [6] "GSM5799933" "GSM5799934" "GSM5799935" "GSM5799936" "GSM5799937"
## [11] "GSM5799938" "GSM5799939"
```

Slight Testing of the values. Making sure that everything is indeed okay.

```
GSE193417
```

```
## $GSE193417_series_matrix.txt.gz
## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 0 features, 12 samples
##   element names: exprs
## protocolData: none
## phenoData
##   sampleNames: GSM5799928 GSM5799929 ... GSM5799939 (12 total)
##   varLabels: title geo_accession ... tissue:ch1 (65 total)
##   varMetadata: labelDescription
## featureData: none
## experimentData: use 'experimentData(object)'
## Annotation: GPL15520
```

Download the supplementary files if there are any into the current working directory.

```
if (!dir.exists('GSE193417')){
  gsefiles = getGEOSuppFiles('GSE193417')
  (fnames <- rownames(gsefiles))
} else {
  gsefiles = getGEOSuppFiles('GSE193417', fetch_files = FALSE)
  (fnames <- paste(getwd(), 'GSE193417', gsefiles$fname, sep = "/"))
}
```

```
## [1] "/home/rstudio/Assignment1/GSE193417/GSE193417_Raw_count_matrix_CRH.csv.gz"
## [2] "/home/rstudio/Assignment1/GSE193417/GSE193417_Sample_metadata.csv.gz"
```

```
#Will give you the names of all the files that were downloaded using the getGEOSuppFiles.
#This is to just show the raw table that was acquired from getGEOSuppFiles.
```

2. Assess, Explore and Clean Sample File(s) and their metadata, then Map to HUGO symbols

This is primarily straightforward, and is based on the lecture notes with some slight adjustments to account for taking in all the files. We only want the file at index 1 since it is the only file available right now. This may change in the future. There is a lot of metadata associated with these file types and are very helpful.

```
#Obtain platform and experiment data
gse <- getGEO("GSE193417", GSEMatrix = FALSE)
```

```
## Reading file....
```

```
## Parsing....
```

```
## Found 13 entities...
```

```
## GPL15520 (1 of 14 entities)
```

```
## GSM5799928 (2 of 14 entities)
```

```
## GSM5799929 (3 of 14 entities)
```

```
## GSM5799930 (4 of 14 entities)
```

```
## GSM5799931 (5 of 14 entities)
```

```
## GSM5799932 (6 of 14 entities)
```

```
## GSM5799933 (7 of 14 entities)
```

```
## GSM5799934 (8 of 14 entities)
```

```
## GSM5799935 (9 of 14 entities)
```

```
## GSM5799936 (10 of 14 entities)
```

```
## GSM5799937 (11 of 14 entities)
```

```
## GSM5799938 (12 of 14 entities)
```

```
## GSM5799939 (13 of 14 entities)
```

```
gse_metadt <- Meta(gse)
gpl_metadt <- Meta(getGEO(names(GPLList(gse)[1])))
```

Platform information - GPL15520 Illumina MiSeq (Homo sapiens)

Platform Title: Cell-Specific Transcriptomic Analysis of CRH+ cells Identifies Unique Cellular Responses to Chronic Stress

Submission data: Jan 10 2022

Last update data: Feb 22 2022

Organisms: (taxid:)

Summary: Rationale: A previous transcriptome meta-analysis revealed significantly lower levels of corticotropin-releasing hormone (CRH) mRNA in corticolimbic brain regions in major depressive disorder (MDD) subjects. Rodent studies show that cortical CRH is mostly expressed in GABAergic neurons; however, the characteristic features of CRH+ cells in human brain cortex and their association with MDD are largely unknown., Methods: Subgenual anterior cingulate cortex (sgACC) of human subjects without brain disorders were labeled using fluorescent in situ hybridization (FISH) for CRH and markers of excitatory (SLC17A7), inhibitory (GAD1) neurons, as well as markers of other interneuron subpopulations (PVALB, SST, VIP). MDD-associated changes in CRH+ cell density and cellular CRH expression (n=6/group) were analyzed. RNA-sequencing was performed on sgACC CRH+ neurons from comparison and MDD subjects (n=6/group), and analyzed for group differences., Results: About 80% of CRH+ cells were GABAergic and 17.5% were glutamatergic. CRH+ GABAergic neurons co-expressed VIP (52%), SST (7%), or PVALB (7%). MDD subjects displayed lower CRH mRNA levels in GABAergic neurons relative to comparison subjects without changes in cell density. CRH+ neurons show transcriptomic profile suggesting lower excitability and less GABA release and reuptake. Further analyses suggested that these molecular changes are not mediated by altered glucocorticoid feedback and potentially occur downstream for a common modulator of neurotrophic function., Summary: CRH+ cells in human sgACC are a heterogeneous population of GABAergic neurons, although largely co-expressing VIP. MDD is associated with reduced markers of inhibitory function of CRH+ neurons.

Number of GEO datasets that use this technology: 0

Number of GEO samples that use this technology: 12

Here is some housekeeping to make sure the information we have downloaded so far is correct and can be processed.

```
mddInMe = read.csv(fnames[1],header=TRUE, check.names = FALSE)
colnames(mddInMe)[1] <- "EntrezID" #Will make adding Ensemble IDs really easy later on.
head(mddInMe, 5) #Displays the head
```

	EntrezID <chr>	CRH-Hu1001.bam <int>	CRH-Hu1031.bam <int>	CRH-Hu1047.bam <int>	CRH-Hu1086.bam <int>	CRH-Hu513.bam <int>
1	ENSG00000000003	0	111	8	0	128
2	ENSG00000000005	0	0	0	0	42
3	ENSG00000000419	139	151	0	1	43
4	ENSG00000000457	0	489	194	353	378
5	ENSG00000000460	0	33	51	30	156

5 rows | 1-7 of 14 columns

```
colnames(mddInMe)
```

```
##  [1] "EntrezID"       "CRH-Hu1001.bam" "CRH-Hu1031.bam" "CRH-Hu1047.bam"
##  [5] "CRH-Hu1086.bam" "CRH-Hu513.bam"  "CRH-Hu600.bam"  "CRH-Hu615.bam"
##  [9] "CRH-Hu789.bam"  "CRH-Hu809.bam"  "CRH-Hu852.bam"  "CRH-Hu863.bam"
## [13] "CRH-Hu943.bam"
```

```
length(colnames(mddInMe))
```

```
## [1] 13
```

```
dim(mddInMe) #Tells you how many rows and columns there are respectively.
```

```
## [1] 19961    13
```

```
length(unique(mddInMe$EntrezID))#This tells me how many unique gene Ids exist in this dataset, in this case they are all uni
que.However we do not know if all the datasets are valid.
```

```
## [1] 19961
```

Checking to see that all the gene names for each of the rows are unique. This number should be the same as the number of rows in the dataset. Furthermore, even though at this point they MIGHT be unique, as in there could be typos, or any other underlying issues, we need to remove them and make sure the data set is workable and clean to use. Since we will also need to normalize it and account for any outliers in future exercises of this assignment The more invalid data we remove now the better it is for us Remove any null or NA values that would cause any problems with the dataset. Just as a precaution. It really isn't necessary in our case because the values are all unique as we established earlier but its good to do, because again we don't know the "state" of uniqueness. NA values, typos, incorrect formats are no good for us.

```
na.omit(mddInMe)
```

	EntrezID <chr>	CRH-Hu1001.bam <int>	CRH-Hu1031.bam <int>	CRH-Hu1047.bam <int>	CRH-Hu1086.bam <int>	CRH-Hu513.bam <int>
1	ENSG00000000003	0	111	8	0	128
2	ENSG00000000005	0	0	0	0	42
3	ENSG00000000419	139	151	0	1	43
4	ENSG00000000457	0	489	194	353	378
5	ENSG00000000460	0	33	51	30	156
6	ENSG00000000938	0	17	0	0	0
7	ENSG00000000971	0	0	142	2	16
8	ENSG00000001036	316	24	180	455	1
9	ENSG00000001084	176	192	218	361	202
10	ENSG00000001167	372	460	172	470	751

1-10 of 10,000 rows | 1-7 of 14 columns Previous **1** 2 3 4 5 6 … 1000 Next

4. Clean as required

It is recommended to remove features with insufficient readings as we started learning in lecture 4, according to the 'edgeR' protocol. There always exists a group of at least one which satisfies the different sample sequences in this experiment.This will tell us how many EntrezIDs we can expect to map at most based on the sample size cut off we introduced with respect to the guidelines used by edgeR and in lecture.

```
filtMddInMe = sum(rowSums(mddInMe > 1) >= 6)
```

This will tell us how many EntrezIDs we can expect to map at most based on the sample size cut off we introduced with respect to the guidelines used by edgeR and in lecture.

```
filtMddInMe <- (mddInMe[rowSums(mddInMe > 1) >= 6, ])
```

```
samples <- data.frame(lapply(colnames(mddInMe)[2:13], function(x) {
  splt = unlist(strsplit(x, split = "."))
  c(splt[1], unlist(strsplit(splt[2], split = "-"))[c(0,1)])
}))
colnames(samples) <- colnames(mddInMe)[2:13]
rownames(samples) <- c("happy", "sad")

samples <- data.frame(t(samples))
```

Assign all the genes in the table an HUGO symbol based on the Ensemble ID for the gene respectively. We will remove all the respective values for duplicate genes, genes with too few. Paraphasing from the paper; "Sequencing data was analyzed as previously described (34). In short, HiSat2 (35) and Genomic-Alignments were used to align 2 100 bp paired-end reads to the GRCh38 human reference genome (ftp.ensembl.org/pub/release-86/fasta/homo sapiens/dna/) (36). After matching genes to exons, noise was reduced by deleting low-expressing genes with fewer

than ten reads and not found in more than two-thirds of the samples. 79.4% and 14.4% of total reads acquired per participant (115,354,986 on average) were aligned to genome and exon, respectively. This study looked at 15,472 genes in total."

```
if(!exists('ensembl')){
  ensembl <- useMart(biomart = "ensembl", dataset="hsapiens_gene_ensembl")
}
if(!exists('geneIDs')){
  geneIDs <- getBM(attributes = c('ensembl_gene_id', 'hgnc_symbol'),
                                filters = 'ensembl_gene_id',
                                values = mddInMe$EntrezID,
                                mart = ensembl)
}
dim(geneIDs)
```

```
## [1] 19915     2
```

We see that there are a few genes missing that have not been mapped. Recall that when we first called dim(mddInMe) we had 19961;rows i.e gene names and 13;columns i.e sample names return as results. This suggests that only 46 genes are un-mapped.However we know that it not true as checking the sample counts of too few for a few genes demonstrated that there are still a number of values that need to be removed. Which we will fix now.

```
# unmappedsegs genes
unmappedsegsNums = nrow(filtMddInMe) - nrow(geneIDs)
#The unmapped segments and portions will be removed.
unmappedsegs <- dplyr::anti_join(filtMddInMe[1], geneIDs[1], by = c("EntrezID" = "ensembl_gene_id"))
#Remove some repitions
FinalGeneFilter <- dplyr::inner_join(geneIDs, filtMddInMe, by = c("ensembl_gene_id" = "EntrezID"))
tempRepeats <- data.frame(table(geneIDs$ensembl_gene_id))
#And then again.
FinalGeneFilter <- FinalGeneFilter[!(FinalGeneFilter$hgnc_symbol=="STRA6LP" | FinalGeneFilter$hgnc_symbol=="LINC00856"),]
#And again. The aim is to remove any of the hgnc symbols that mapped to more than one Ensemble IDs. Essentially the Ensemble
IDs should be treated as some sort of function. For every 1x, there is exactly 1y to pair with it. Lock and key so to speak.
FinalGeneFilter <- FinalGeneFilter[!(FinalGeneFilter$hgnc_symbol=="POLR2J3" | FinalGeneFilter$hgnc_symbol=="TBCE"),]
goodHGNCboy <- data.frame(table(geneIDs$hgnc_symbol))
FinalGeneFilter <- FinalGeneFilter[!(FinalGeneFilter$hgnc_symbol ==""), ]


keep = rowSums(FinalGeneFilter[2:13] >1) >= 6
FinalGeneFilter <- FinalGeneFilter[keep,]
dim(FinalGeneFilter)
```

```
## [1] 15349    14
```

Which is a lot less than what we had initially.

```
#Check for more duplicates after sorting the first time.
smmryGeneCts <- sort(table(FinalGeneFilter$hgnc_symbol), decreasing = TRUE)
smmryGeneCts[which(smmryGeneCts > 1)]
```

```
## named integer(0)
```

```
#Going through the first time, there were some repititions missing and we had to go back to remove them manually.
```

The finally percentage of removals due to duplications, NA values etc.

```
mddnumbers = length(unique(mddInMe$EntrezID))
goodmddnumbers = length(unique(FinalGeneFilter$ensembl_gene_id))
percentageleft = goodmddnumbers/mddnumbers
percentageremoved <- (1 - percentageleft) * 100
cat(paste("The total amount of genes removed so far is:", percentageremoved))
```

```
## The total amount of genes removed so far is: 23.1050548569711
```

Convert to Matrix and then to Vector, this will be useful later when we are subsetting across various files and values of the data sctructure.

```
# Create a matrix
matrixdata <- as.matrix(mddInMe)[, 2:13]

# Create a vector
vectordata <- as.vector(mddInMe)[, 2:13]

matrixdata2 <- as.matrix(FinalGeneFilter)[, 3:14]
vectordata2 <- as.vector(FinalGeneFilter)[, 3:14]
```

Edge Case calculation represented as a variable for the final normalized values.

```
d <- edgeR::cpm(FinalGeneFilter[,3:14])
```

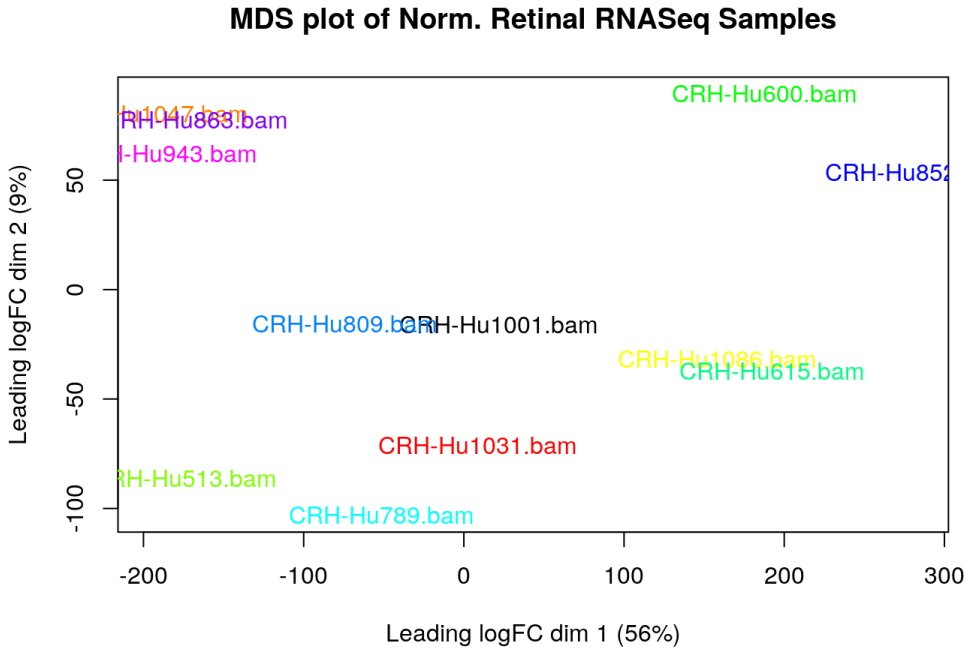
Manually doing this because I had a really frustrating time regarding the subsetting.

```
samples <- data.frame(
  lapply(colnames(mddInMe), FUN=function(x){
    unlist(strsplit(x, split="\\_"))[c(2,3)]
  }
))
colnames(samples) <- ordered(colnames(mddInMe))
rownames(samples) <- c("sample_number", "group")

#Not going to lie this could have been done with a forloop or even subsetting. I wanted to challenged myself and understand
the different aspects of subsetting and value accession.
#It was quite probably the worst thing I have ever put myself through.
samples[2, 2:13] = 1
samples[1, 2] = 'CRH-Hu1001 - 2'
samples[1, 3] = 'CRH-Hu1031 - 1'
samples[1, 4] = 'CRH-Hu1047 - 1'
samples[1, 5] = 'CRH-Hu1086 - 1'
samples[1, 6] = 'CRH-Hu513 - 2'
samples[1, 7] = 'CRH-Hu600 - 2'
samples[1, 8] = 'CRH-Hu615 - 1'
samples[1, 9] = 'CRH-Hu789 - 1'
samples[1, 10] = 'CRH-Hu809 - 2'
samples[1, 11] = 'CRH-Hu852 - 1'
samples[1, 12] = 'CRH-Hu863 - 2'
samples[1, 13] = 'CRH-Hu943 - 2'
samples[2, 2] = 2
samples[2, 13] = 2
samples[2, 12] = 2
samples[2, 10] = 2
samples[2, 6:7] = 2
```

##MDS plot

```
plotMDS(d,
        labels=samples$sample_number,
        col = rainbow(length(levels(factor(samples[1,])))), alpha=1)[factor(samples[1,])], main="MDS plot of Norm. Retinal RN
ASeq Samples")
```

```
## Warning in xtfrm.data.frame(x): cannot xtfrm data frames

## Warning in xtfrm.data.frame(x): cannot xtfrm data frames
```



The multidimensional scaling points, which have already been accounted for with correct log base due to the “edge” fixing are a means of showing how the NR as well as the R groups of any experiment interact with one another as well as with respect to other samples from the same organism, different organism, varying age etc. It has a multitude of ideas, here we can see there is over a half fold difference between the two values, the first being the control group for this study, those whom are not affected by Major Depressive disorder (MDD) and those individuals that are. Although predominantly co-expressing VIP, CRH+ cells in human sgACC are a diverse population of GABAergic interneurons. Our findings imply that MDD is linked to decreased inhibitory function indicators in sgACC CRH+ interneurons, and they add to the growing body of evidence for altered GABAergic function in the cortex in MDD. From the results we can see there is a large degree of variability relative from controls and non-controls but also those who are suffering from MDD or not tend to cluster around the same area of the plot. This does state that there are genes that are responsible for the response that individuals and the researchers took note of. One thing that would be great to see for a plot like this is obviously a lot more samples, as 6 samples per group seems very little, which is something that the researchers mentioned in the paper. To add, controls are denoted with a “-1” after their sample name whilst the affected are denoted with a “-2” after their sample name. The output that denotes the control on non control can be found above this plot as well. I do believe though that we got a good distribution of the samples.

The box plots for the edge cased non-normalized values can be found below, respectively. These are primary used to show the distribution of a gene segment, clusters, distribution of a sgene segment/gene in question. My gene segment for non-normalized was absolutely disheartening to see, this is because there were so few values on the graph and most of the boxes for the plot were considered to be outlines for their very drastic difference compared to some others. Since we have already removed values of cpm that are required based on the smallest sample size of our experiment, in this case it is 6, we can for the most part assume that the normalized values are a bit more acceptable. I did two box plots on purpose, because I wanted to compare the difference between the values that get removed and those that do not. It really was interesting to see that the values, especially the normalized values would change the appearance of the box plot so drastically. This is probably due to the fact that the values where cpm = 0 6 or more time for a given gene were removed. As you can see the box plot for the normalized values have a lot more data. ##Box plot for non-normalized values

```
data2plot <- log2(edgeR::cpm(mddInMe[,2:13]))
boxplot(data2plot, xlab = "Samples", ylab = "log2 CPM", las = 2, cex = 0.5, cex.lab = 0.5, cex.axis = 0.5, main = "RNASeq Sa
mples")
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z$group
## == : Outlier (-Inf) in boxplot 2 is not drawn
```

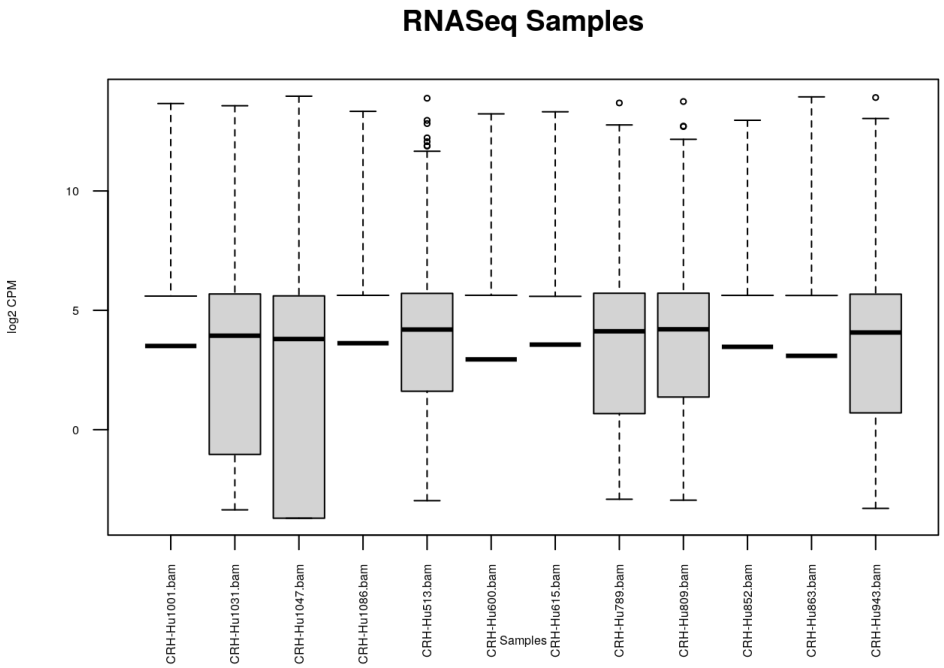
```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z$group
## == : Outlier (-Inf) in boxplot 3 is not drawn
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z$group
## == : Outlier (-Inf) in boxplot 5 is not drawn
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z$group
## == : Outlier (-Inf) in boxplot 8 is not drawn
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z$group
## == : Outlier (-Inf) in boxplot 9 is not drawn
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z$group
## == : Outlier (-Inf) in boxplot 12 is not drawn
```



##Box plot for normalized values

```
data2plot <- log2(edgeR::cpm(FinalGeneFilter[,3:14]))
boxplot(data2plot, xlab = "Samples", ylab = "log2 CPM", las = 2, cex = 0.5, cex.lab = 0.5, cex.axis = 0.5, main = "RNASeq Sa
mples")
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z$group
## == : Outlier (-Inf) in boxplot 1 is not drawn
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z$group
## == : Outlier (-Inf) in boxplot 2 is not drawn
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z$group
## == : Outlier (-Inf) in boxplot 3 is not drawn
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z$group
## == : Outlier (-Inf) in boxplot 4 is not drawn
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z$group
## == : Outlier (-Inf) in boxplot 5 is not drawn
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z$group
## == : Outlier (-Inf) in boxplot 6 is not drawn
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z$group
## == : Outlier (-Inf) in boxplot 7 is not drawn
```

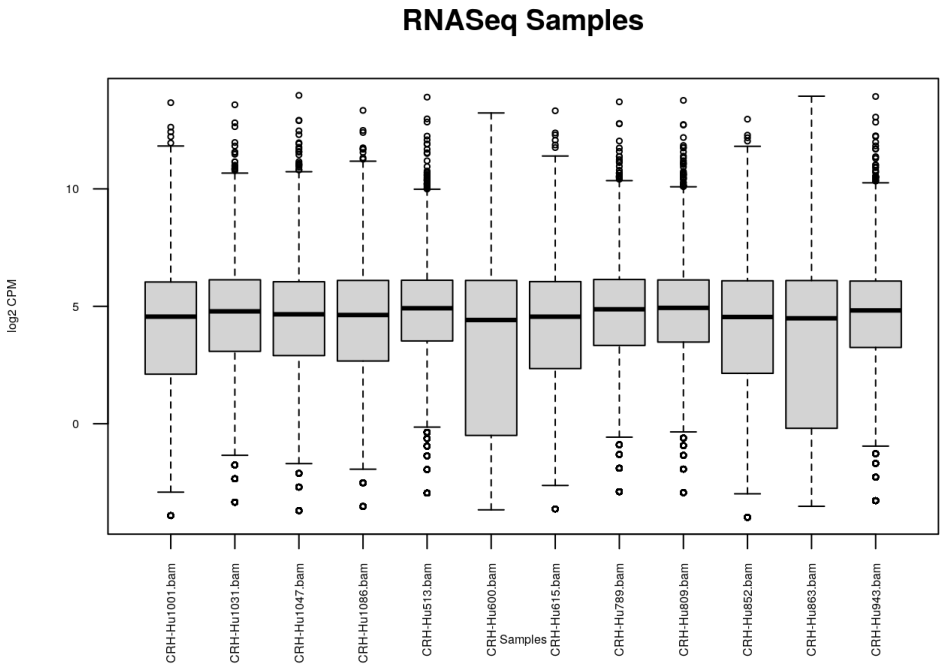
```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z$group
## == : Outlier (-Inf) in boxplot 8 is not drawn
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z$group
## == : Outlier (-Inf) in boxplot 9 is not drawn
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z$group
## == : Outlier (-Inf) in boxplot 10 is not drawn
```

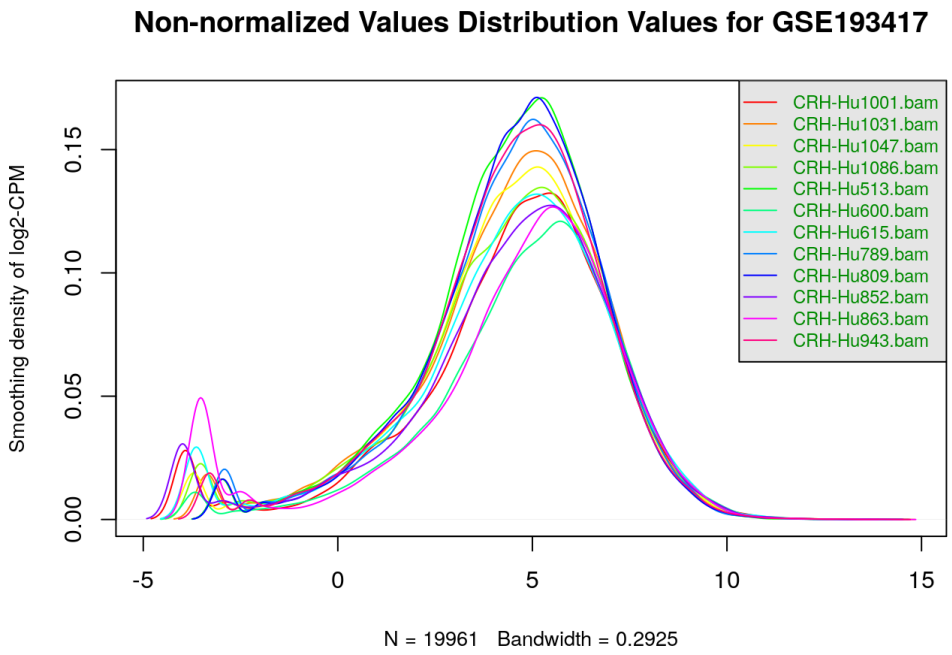
```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z$group
## == : Outlier (-Inf) in boxplot 11 is not drawn
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z$group
## == : Outlier (-Inf) in boxplot 12 is not drawn
```



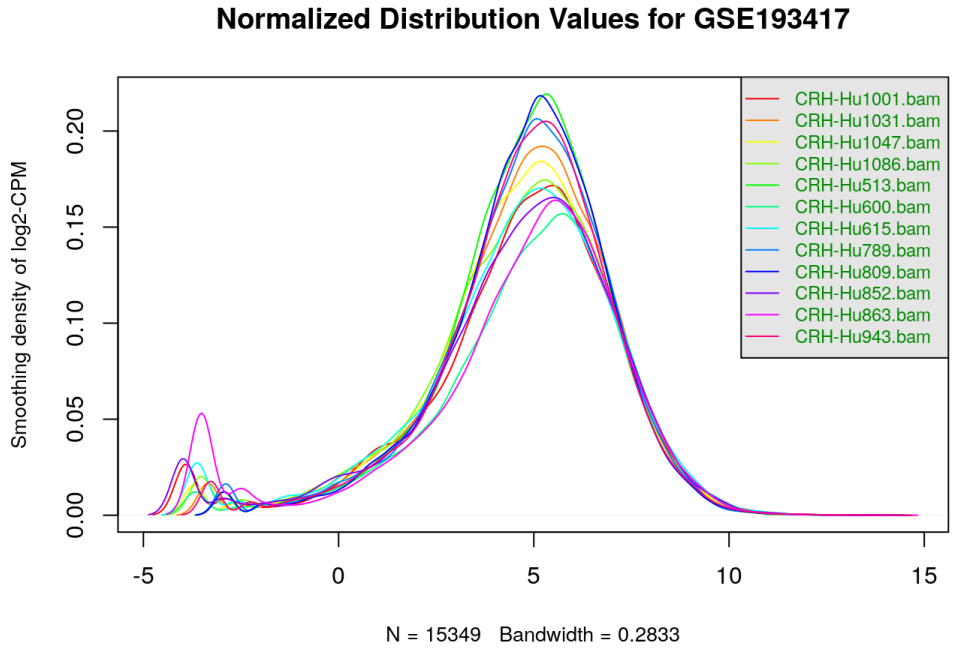
The density curves are another way for us to analyze our data before and after the normalization process, they work really well with larger data sets because of the fact that they are proportional to the size of the data set, which I think is wonderful news. Assuming that fact that the research behind this continues onward, the density curves across samples maps vs. genes will more than likely stay the same. While the extreme ends of the curve are more than likely to be affect by outlines and data that do not match with the average, like for most cases, we can still deduce that this experience and overall those who do suffer from MDD will more than likely find someone where they can comfortably align themselves to. From the analytic side of things, the more information we can add to this curve, the more we can predict and model. Further more, for this data set and these samples specifically, the higher curves for the youngest and oldest of the group. Another thing to note is that the curves have relatively the same shape, peaks and even axis, while there are a few points where the difference in the variable values mapped against the curve fall short of the non-normalized graph, the normalized graph lack extra values, repetitions, are the most likely culprit as they are counting for double what they are normally supposed to count for, so obviously that will affect the graph in larger peaks, and a higher density. Which I do expect considered if you you twice as much of something and someone else has exactly half at the very leas or none at all, then tangibly they will have more of said thing. The same ambiguous concept applies here for these graphs, samples and data. ##Density Curve for Non-normalized Distribution Values

```
ctsDenct <- apply(log2(edgeR::cpm(mddInMe[,2:13])), 2, density)

xlim <- 0; ylim <- 0
for (i in 1:length(ctsDenct)) {
  xlim <- range(c(xlim, ctsDenct[[i]]$x));
  ylim <- range(c(ylim, ctsDenct[[i]]$y))
}
cols <- rainbow(length(ctsDenct))
ltys <- rep(1, length(ctsDenct))

plot(ctsDenct[[1]], xlim = xlim, ylim = ylim, type = "n", ylab = "Smoothing density of log2-CPM", main = "Non-normalized Val
ues Distribution Values for GSE193417", cex.lab = 0.85)
for (i in 1:length(ctsDenct))
  lines(ctsDenct[[i]], col = cols[i], lty = ltys[i])

legend("topright", colnames(data2plot), col=cols, lty=ltys, cex=0.75, border="blue", text.col = "green4", merge = TRUE, bg =
"gray90")
```



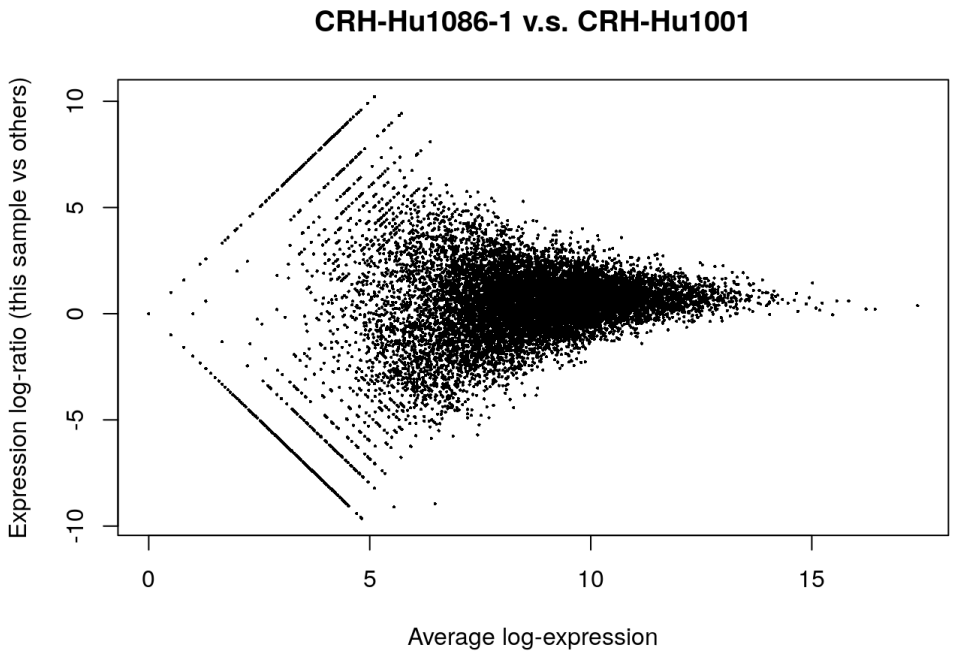
##Density Curve for Normalized Distribution Values

```
ctsDenct <- apply(log2(edgeR::cpm(FinalGeneFilter[,3:14])), 2, density)

xlim <- 0; ylim <- 0
for (i in 1:length(ctsDenct)) {
  xlim <- range(c(xlim, ctsDenct[[i]]$x));
  ylim <- range(c(ylim, ctsDenct[[i]]$y))
}
cols <- rainbow(length(ctsDenct))
ltys <- rep(1, length(ctsDenct))

plot(ctsDenct[[1]], xlim = xlim, ylim = ylim, type = "n", ylab = "Smoothing density of log2-CPM", main = "Normalized Distrib
ution Values for GSE193417", cex.lab = 0.85)
for (i in 1:length(ctsDenct))
  lines(ctsDenct[[i]], col = cols[i], lty = ltys[i])

legend("topright", colnames(data2plot), col=cols, lty=ltys, cex=0.75, border="blue", text.col = "green4", merge = TRUE, bg =
"gray90")
```

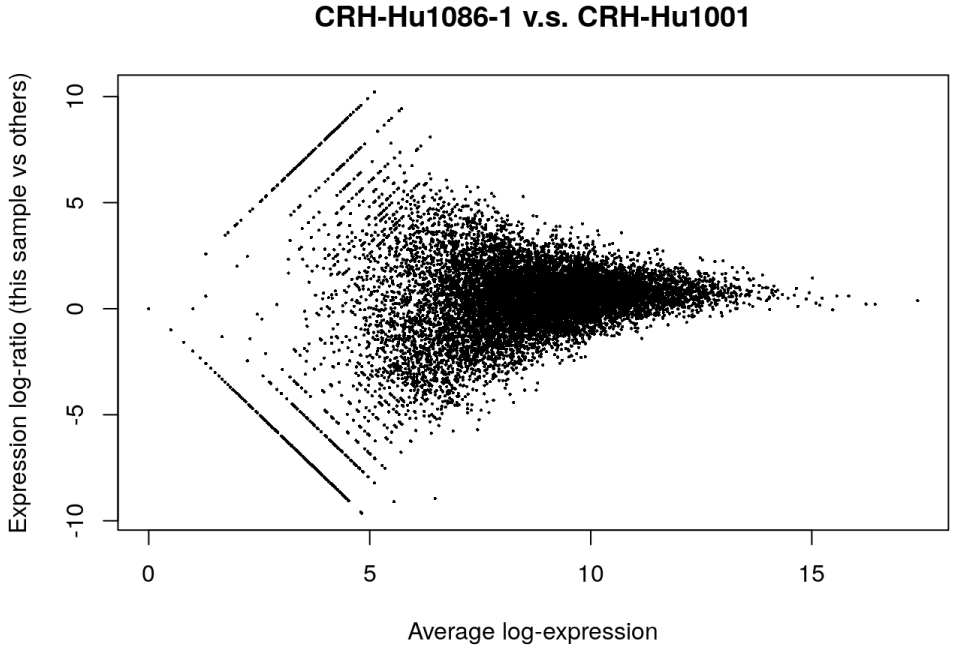


Simple log ration expression of a sample within the data set mapped against other sample within the data set. I did this for both normalized and non normalized values. The distribution of the samples is really well put together. It looks really pretty, and I am amazed that it looks this good. I am not sure why this is the case for these two, I understand that I compared the same two before and after normalization so that is why in terms of looks they appear identical because as we have already concluded that these are really great for getting the overall vibe for a set amount of data and being able to predict it based on the desnity/adjustments etc. because of normalization and distribution. Though I am wondering why they look like a kite, I have seen some really pretty graphs before, but I think this is the best one yet. Possibly due to the fact that I made it. ##Expression Log Ratios

```
limma::plotMA(log2(mddInMe[,c(2,13)]), main = "CRH-Hu1086-1 v.s. CRH-Hu1001")
```




```
limma::plotMA(log2(FinalGeneFilter[,c(3,14)]), main = "CRH-Hu1086-1 v.s. CRH-Hu1001")
```



6. Interpretation

What are the control and test conditions of the dataset? Controls = 'CRH-Hu1031 - 1', 'CRH-Hu1047 - 1', CRH-Hu1086 - 1 'CRH-Hu615 - 1', 'CRH-Hu789 - 1', 'CRH-Hu852 - 1' Test Conditions = 'CRH-Hu1001 - 2', 'CRH-Hu513 - 2', 'CRH-Hu600 - 2', 'CRH-Hu809 - 2', 'CRH-Hu863 - 2', 'CRH-Hu943 - 2' What these mean is there are individuals who are the controls, that are still able to express high amounts of CRH+ cells , denoted byt a "-1" whereas those that suffer from MDD are unable to express any CRH+ or have very little expression. Subgenual anterior cingulate cortex (sgACC) of human subjects without brain disorders were labeled using fluorescent in situ hybridization (FISH), as well as for any other receptors and cells that made you feel positive and that was measured. Those who suffered from MDD were first thought to have this effect due to unbalanced glucocorticoid feedback however it has been recently proven and this article is further the research on this, that neurotrophic function caused degradtion in these cells making someone more susceptible to MDD. The participants ranged from being very happy with their life (not limited to just the controls), to unfortunately eventually ending their own life (the test conditions).

Why is the dataset of interest to you?' As someone who struggle with mental health issues, I am always interested in these topics. It can only make us more aware and better for the long term once we are able to understand how to take care of our mind, and not just our physical bodies too. Broken bones almost always heal, but broken souls do not. Or so I've heard.

Were there expression values that were not unique for specific genes? How did you handle these? I removed any genes that were not unique, or non-availbe. I don't think I had any that were duplicated in terms of expression, but I had a lot of the same hgnc symbols.

Were there expression values that could not be mapped to current HUGO symbols? No, I was luck in the sense that they all mapped t HUGO symbols. They did however have duplicates.

How many outliers were removed? See above for the specific amount. 23% approximately.

How did you handle replicates? Removed them in the filtered dataset but kept them in my orignal data set that was loaded.

What is the final coverage of your dataset? Approximately 15349. Which is slightly lower than the paper's but I suppose they did not remove any outlines.

##Final Dataset With everything removed and accounted for, normalized etc.

```
FinalGeneFilter <- data.frame(FinalGeneFilter)
```

7. Citations

1. R programming for data science: https://bookdown.org/rdpeng/rprogdatascience/data-analysis-case-study-changes-in-fine-particle-air-pollution-in-the-u-s-.html (https://bookdown.org/rdpeng/rprogdatascience/data-analysis-case-study-changes-in-fine-particle-air-pollution-in-the-u-s-.html)
2. Lecture modules: https://q.utoronto.ca/courses/248455/modules (https://q.utoronto.ca/courses/248455/modules)
3. Oh, Hyunjung & Newton, Dwight & Lewis, David & Sibille, Etienne. (2022). Lower Levels of GABAergic Function Markers in Corticotropin-

Releasing Hormone-Expressing Neurons in the sgACC of Human Subjects With Depression. Frontiers in Psychiatry. 13. 10.3389/fpsyt.2022.827972.

```
citation("GEOmetadb")
```

```
##
## Please cite the following if utilizing the GEOmetadb software:
##
##   Zhu Y, Davis S, Stephens R, Meltzer PS, Chen Y. GEOmetadb: powerful
##   alternative search engine for the Gene Expression Omnibus.
##   Bioinformatics. 2008 Dec 1;24(23):2798-800. doi:
##   10.1093/bioinformatics/btn520. Epub 2008 Oct 7. PubMed PMID:
##   18842599; PubMed Central PMCID: PMC2639278.
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     author = {Yuelin Zhu and Sean Davis and Robert Stephens and Paul S. Meltzer and Yidong Chen},
##     title = {GEOmetadb: powerful alternative search engine for the Gene Expression Omnibus.},
##     journal = {Bioinformatics (Oxford, England)},
##     year = {2008},
##     month = {Dec},
##     day = {01},
##     volume = {24},
##     number = {23},
##     pages = {2798--2800},
##     abstract = {The NCBI Gene Expression Omnibus (GEO) represents the largest public repository of microarray data. Howev
er, finding data in GEO can be challenging. We have developed GEOmetadb in an attempt to make querying the GEO metadata both
easier and more powerful. All GEO metadata records as well as the relationships between them are parsed and stored in a loca
l MySQL database. A powerful, flexible web search interface with several convenient utilities provides query capabilities no
t available via NCBI tools. In addition, a Bioconductor package, GEOmetadb that utilizes a SQLite export of the entire GEOme
tadb database is also available, rendering the entire GEO database accessible with full power of SQL-based queries from with
in R.},
##     issn = {1367-4811},
##     doi = {10.1093/bioinformatics/btn520},
##     url = {http://www.ncbi.nlm.nih.gov/pubmed/18842599},
##     language = {eng},
##   }
```

```
citation("GEOquery")
```

```
##
## Please cite the following if utilizing the GEOquery software:
##
##   Davis, S. and Meltzer, P. S. GEOquery: a bridge between the Gene
##   Expression Omnibus (GEO) and BioConductor. Bioinformatics, 2007, 14,
##   1846-1847
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     author = {Sean Davis and Paul Meltzer},
##     title = {GEOquery: a bridge between the Gene Expression Omnibus (GEO) and BioConductor},
##     journal = {Bioinformatics},
##     year = {2007},
##     volume = {14},
##     pages = {1846--1847},
##   }
```

```
citation("biomaRt")
```

```
##
## To cite the biomaRt package in publications use:
##
##   Mapping identifiers for the integration of genomic datasets with the
##   R/Bioconductor package biomaRt. Steffen Durinck, Paul T. Spellman,
##   Ewan Birney and Wolfgang Huber, Nature Protocols 4, 1184-1191 (2009).
##
##   BioMart and Bioconductor: a powerful link between biological
##   databases and microarray data analysis. Steffen Durinck, Yves Moreau,
##   Arek Kasprzyk, Sean Davis, Bart De Moor, Alvis Brazma and Wolfgang
##   Huber, Bioinformatics 21, 3439-3440 (2005).
##
## To see these entries in BibTeX format, use 'print(<citation>,
## bibtex=TRUE)', 'toBibtex(.)', or set
## 'options(citation.bibtex.max=999)'.
```

```
citation("edgeR")
```

```
##
## See Section 1.2 in the User's Guide for more detail about how to cite
## the different edgeR pipelines.
##
##   Robinson MD, McCarthy DJ and Smyth GK (2010). edgeR: a Bioconductor
##   package for differential expression analysis of digital gene
##   expression data. Bioinformatics 26, 139-140
##
##   McCarthy DJ, Chen Y and Smyth GK (2012). Differential expression
##   analysis of multifactor RNA-Seq experiments with respect to
##   biological variation. Nucleic Acids Research 40, 4288-4297
##
##   Chen Y, Lun ATL, Smyth GK (2016). From reads to genes to pathways:
##   differential expression analysis of RNA-Seq experiments using
##   Rsubread and the edgeR quasi-likelihood pipeline. F1000Research 5,
##   1438
##
## To see these entries in BibTeX format, use 'print(<citation>,
## bibtex=TRUE)', 'toBibtex(.)', or set
## 'options(citation.bibtex.max=999)'.
```