GSE61853

Natalia

12 May 2019

```r
set.seed(42)

# read the dataset into R
library(GEOquery)
```

```
## Loading required package: Biobase

## Loading required package: BiocGenerics

## Loading required package: parallel

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB

## 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,
##     colMeans, colnames, colSums, dirname, do.call, duplicated,
##     eval, evalq, Filter, Find, get, grep, grepl, intersect,
##     is.unsorted, lapply, lengths, Map, mapply, match, mget, order,
##     paste, pmax, pmax.int, pmin, pmin.int, Position, rank, rbind,
##     Reduce, rowMeans, rownames, rowSums, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which, 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")'.

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

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

```r
library(limma)
```

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

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

```r
#library(org.Mm.eg.db)
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 object is masked from 'package:base':
##
##     expand.grid

##
## Attaching package: 'IRanges'

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

##
```

```r
# for collapseBy:

source("C://Users//Natalia//Desktop//ITMO//SystemBiology//RNAseq_analysis//RNAseq_analysis//dataset#1//


#Gene expression analysis of bone marrow mesenchymal stromal cells from myelodysplastic syndrome (MDS)

es <- getGEO("GSE61853", AnnotGPL = TRUE, parseCharacteristics = FALSE)[[1]]
```

```
## Found 1 file(s)

## GSE61853_series_matrix.txt.gz

## Parsed with column specification:
## cols(
##   ID_REF = col_character(),
##   GSM1515746 = col_double(),
##   GSM1515747 = col_double(),
##   GSM1515748 = col_double(),
##   GSM1515749 = col_double(),
##   GSM1515750 = col_double(),
##   GSM1515751 = col_double(),
##   GSM1515752 = col_double(),
##   GSM1515753 = col_double(),
##   GSM1515754 = col_double(),
##   GSM1515755 = col_double(),
##   GSM1515756 = col_double(),
##   GSM1515757 = col_double(),
##   GSM1515758 = col_double(),
##   GSM1515759 = col_double()
## )

## File stored at:
```

```
## C:\Users\Public\Documents\iSkysoft\CreatorTemp\RtmpCy8mAD/GPL10558.annot.gz
```

```
## Warning: 13 parsing failures.
##   row     col               expected          actual         file
## 29686 Gene ID no trailing characters ///283507       literal data
## 29841 Gene ID no trailing characters ///2074         literal data
## 30401 Gene ID no trailing characters ///27185///7257 literal data
## 31184 Gene ID no trailing characters ///57592        literal data
## 31271 Gene ID no trailing characters ///3199         literal data
## ..... ....... ...................... ............... ............
## See problems(...) for more details.
```

```
str(experimentData(es))
```

```
## Formal class 'MIAME' [package "Biobase"] with 13 slots
##   ..@ name             : chr "Seungwoo,,Hwang"
##   ..@ lab              : chr ""
##   ..@ contact          : chr "swhwang10@yahoo.com"
##   ..@ title            : chr "Gene expression analysis of bone marrow mesenchymal stromal cells from
##   ..@ abstract         : chr "Myelodysplastic syndrome (MDS) is a group of heterogeneous clonal stem
##   ..@ url              : chr "https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE61853"
##   ..@ pubMedIds        : chr "25803272"
##   ..@ samples          : list()
##   ..@ hybridizations   : list()
##   ..@ normControls     : list()
##   ..@ preprocessing    : list()
##   ..@ other            :List of 25
##   .. ..$ contact_address        : chr "52 Eoeun-dong Yuseong-gu"
##   .. ..$ contact_city           : chr "Daejeon"
##   .. ..$ contact_country        : chr "South Korea"
##   .. ..$ contact_department     : chr "Korean Bioinformation Center"
##   .. ..$ contact_email          : chr "swhwang10@yahoo.com"
##   .. ..$ contact_institute      : chr "Korea Research Institute of Bioscience and Biotechnology"
##   .. ..$ contact_name           : chr "Seungwoo,,Hwang"
##   .. ..$ contact_phone          : chr "82-42-879-8544"
##   .. ..$ contact_zip/postal_code: chr "305-806"
##   .. ..$ contributor            : chr "Miyoung,,Kim\nDong,S,Lee"
##   .. ..$ geo_accession          : chr "GSE61853"
##   .. ..$ last_update_date       : chr "Aug 13 2018"
##   .. ..$ overall_design         : chr "The present study include seven adults referred for staging o
##   .. ..$ platform_id            : chr "GPL10558"
##   .. ..$ platform_taxid         : chr "9606"
##   .. ..$ pubmed_id              : chr "25803272"
##   .. ..$ relation               : chr "BioProject: https://www.ncbi.nlm.nih.gov/bioproject/PRJNA2625
##   .. ..$ sample_id              : chr "GSM1515746 GSM1515747 GSM1515748 GSM1515749 GSM1515750 GSM151
##   .. ..$ sample_taxid           : chr "9606"
##   .. ..$ status                 : chr "Public on Mar 25 2015"
##   .. ..$ submission_date        : chr "Sep 29 2014"
##   .. ..$ summary                : chr "Myelodysplastic syndrome (MDS) is a group of heterogeneous cl
##   .. ..$ supplementary_file     : chr "ftp://ftp.ncbi.nlm.nih.gov/geo/series/GSE61nnn/GSE61853/suppl
##   .. ..$ title                  : chr "Gene expression analysis of bone marrow mesenchymal stromal c
##   .. ..$ type                   : chr "Expression profiling by array"
##   ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slot
##   .. .. ..@ .Data:List of 2
##   .. .. .. ..$ : int [1:3] 1 0 0
##   .. .. .. ..$ : int [1:3] 1 1 0
```

```r
str(pData(es))
```

```
## 'data.frame':    14 obs. of  34 variables:
##  $ title                  : Factor w/ 14 levels "BM_MSCs-Control-rep1",..: 1 2 3 4 5 6 7 12 13 14 ..
##  $ geo_accession          : chr  "GSM1515746" "GSM1515747" "GSM1515748" "GSM1515749" ...
##  $ status                 : Factor w/ 1 level "Public on Mar 25 2015": 1 1 1 1 1 1 1 1 1 1 ...
##  $ submission_date        : Factor w/ 1 level "Sep 29 2014": 1 1 1 1 1 1 1 1 1 1 ...
##  $ last_update_date       : Factor w/ 1 level "Mar 25 2015": 1 1 1 1 1 1 1 1 1 1 ...
##  $ type                   : Factor w/ 1 level "RNA": 1 1 1 1 1 1 1 1 1 1 ...
##  $ channel_count          : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ source_name_ch1        : Factor w/ 3 levels "Bone marrow mesenchymal stromal cells (BM MSCs), Con
##  $ organism_ch1           : Factor w/ 1 level "Homo sapiens": 1 1 1 1 1 1 1 1 1 1 ...
##  $ characteristics_ch1    : Factor w/ 1 level "cell type: bone marrow mesenchymal stromal cells": 1 1
##  $ characteristics_ch1.1  : Factor w/ 4 levels "diagnosis: Lymphoma with no evidence of BM involvemer
##  $ molecule_ch1           : Factor w/ 1 level "total RNA": 1 1 1 1 1 1 1 1 1 1 ...
##  $ extract_protocol_ch1   : Factor w/ 1 level "Total RNA was extracted with Trizol and purified with
##  $ label_ch1              : Factor w/ 1 level "biotin": 1 1 1 1 1 1 1 1 1 1 ...
##  $ label_protocol_ch1     : Factor w/ 1 level "Biotinylated cRNA were prepared with the Ambion Illum:
##  $ taxid_ch1              : Factor w/ 1 level "9606": 1 1 1 1 1 1 1 1 1 1 ...
##  $ hyb_protocol           : Factor w/ 1 level "Standard Illumina hybridization protocol": 1 1 1 1 1 1
##  $ scan_protocol          : Factor w/ 1 level "Standard Illumina scanning protocol": 1 1 1 1 1 1 1 1
##  $ description            : Factor w/ 3 levels "Bone marrow mesenchymal stromal cells (BM MSCs) from
##  $ data_processing        : Factor w/ 1 level "The data were normalised using quantile normalisation
##  $ platform_id            : Factor w/ 1 level "GPL10558": 1 1 1 1 1 1 1 1 1 1 ...
##  $ contact_name           : Factor w/ 1 level "Seungwoo,,Hwang": 1 1 1 1 1 1 1 1 1 1 ...
##  $ contact_email          : Factor w/ 1 level "swhwang10@yahoo.com": 1 1 1 1 1 1 1 1 1 1 ...
##  $ contact_phone          : Factor w/ 1 level "82-42-879-8544": 1 1 1 1 1 1 1 1 1 1 ...
##  $ contact_department     : Factor w/ 1 level "Korean Bioinformation Center": 1 1 1 1 1 1 1 1 1 1 ..
##  $ contact_institute      : Factor w/ 1 level "Korea Research Institute of Bioscience and Biotechnol
##  $ contact_address        : Factor w/ 1 level "52 Eoeun-dong Yuseong-gu": 1 1 1 1 1 1 1 1 1 1 ...
##  $ contact_city           : Factor w/ 1 level "Daejeon": 1 1 1 1 1 1 1 1 1 1 ...
##  $ contact_zip/postal_code: Factor w/ 1 level "305-806": 1 1 1 1 1 1 1 1 1 1 ...
##  $ contact_country        : Factor w/ 1 level "South Korea": 1 1 1 1 1 1 1 1 1 1 ...
##  $ supplementary_file     : Factor w/ 1 level "NONE": 1 1 1 1 1 1 1 1 1 1 ...
##  $ data_row_count         : Factor w/ 1 level "47323": 1 1 1 1 1 1 1 1 1 1 ...
##  $ cell type:ch1          : chr  "bone marrow mesenchymal stromal cells" "bone marrow mesenchymal st:
##  $ diagnosis:ch1          : chr  "Lymphoma with no evidence of BM involvement" "Lymphoma with no evic
```

```r
head(fData(es))
```

```
##                        ID
## ILMN_1343291 ILMN_1343291
## ILMN_1343295 ILMN_1343295
## ILMN_1651199 ILMN_1651199
## ILMN_1651209 ILMN_1651209
## ILMN_1651210 ILMN_1651210
## ILMN_1651221 ILMN_1651221
##                                                Gene title
## ILMN_1343291 eukaryotic translation elongation factor 1 alpha 1
## ILMN_1343295           glyceraldehyde-3-phosphate dehydrogenase
## ILMN_1651199
## ILMN_1651209                   solute carrier family 35 member E2
## ILMN_1651210                      dual specificity phosphatase 22
## ILMN_1651221
##             Gene symbol Gene ID UniGene title UniGene symbol UniGene ID
## ILMN_1343291      EEF1A1    1915
## ILMN_1343295       GAPDH    2597
## ILMN_1651199                  NA
## ILMN_1651209     SLC35E2    9906
## ILMN_1651210      DUSP22   56940
## ILMN_1651221                  NA
##                                                                                       Nucleotide Titl
## ILMN_1343291            Homo sapiens eukaryotic translation elongation factor 1 alpha 1 (EEF1A1), mR
## ILMN_1343295 Homo sapiens glyceraldehyde-3-phosphate dehydrogenase (GAPDH), transcript variant 1, mR
## ILMN_1651199
## ILMN_1651209     Homo sapiens solute carrier family 35 member E2 (SLC35E2), transcript variant 1, mR
## ILMN_1651210         Homo sapiens dual specificity phosphatase 22 (DUSP22), transcript variant 2, mR
## ILMN_1651221
##                     GI GenBank Accession Platform_CLONEID Platform_ORF
## ILMN_1343291  83367078         NM_001402               NA           NA
## ILMN_1343295 576583510         NM_002046               NA           NA
## ILMN_1651199        NA                                 NA           NA
## ILMN_1651209 315139027         NM_182838               NA           NA
## ILMN_1651210 557440873         NM_020185               NA           NA
## ILMN_1651221        NA                                 NA           NA
##             Platform_SPOTID Chromosome location
## ILMN_1343291              NA              6q14.1
## ILMN_1343295              NA               12p13
## ILMN_1651199              NA
## ILMN_1651209              NA             1p36.33
## ILMN_1651210              NA              6p25.3
## ILMN_1651221              NA
##                                                    Chromosome annotation
## ILMN_1343291 Chromosome 6, NC_000006.12 (73515750..73521032, complement)
## ILMN_1343295             Chromosome 12, NC_000012.12 (6534405..6538375)
## ILMN_1651199
## ILMN_1651209   Chromosome 1, NC_000001.11 (1724838..1745999, complement)
## ILMN_1651210                Chromosome 6, NC_000006.12 (292057..351355)
## ILMN_1651221
##
## ILMN_1343291
## ILMN_1343295 NAD binding///NADP binding///glyceraldehyde-3-phosphate dehydrogenase (NAD+) (phosphoryl
## ILMN_1651199
## ILMN_1651209
## ILMN_1651210
## ILMN_1651221
##
## ILMN_1343291
## ILMN_1343295
## ILMN_1651199
## ILMN_1651209
## ILMN_1651210 apoptotic process///cell proliferation///inactivation of MAPK activity///multicellular
## ILMN_1651221
##
## ILMN_1343291                                                                                cortical
## ILMN_1343295 GAIT complex///cytoplasm///cytoplasm///cytosol///cytosol///cytosol///extracellular exoso
## ILMN_1651199
## ILMN_1651209
## ILMN_1651210
## ILMN_1651221
##
## ILMN_1343291                           GO:0005525///GO:0003924///GO:0044822///GO:0005515///GO:001990
## ILMN_1343295 GO:0051287///GO:0050661///GO:0004365///GO:0004365///GO:0004365///GO:0042802///GO:000801
## ILMN_1651199
## ILMN_1651209
## ILMN_1651210
## ILMN_1651221
##
## ILMN_1343291
## ILMN_1343295                                        GO:0061621///GO:0071346///GO:0006094///GO:000022
## ILMN_1651199
## ILMN_1651209
## ILMN_1651210 GO:0006915///GO:0008283///GO:0000188///GO:0007275///GO:0050868///GO:0002710///GO:005086
## ILMN_1651221
##
## ILMN_1343291                                                                  GO:0030864///GO:000573
## ILMN_1343295 GO:0097452///GO:0005737///GO:0005737///GO:0005829///GO:0005829///GO:0005829///GO:007006
## ILMN_1651199
## ILMN_1651209
## ILMN_1651210
## ILMN_1651221
##                                                Platform_SEQUENCE
## ILMN_1343291 TGTGTTGAGAGCTTCTCAGACTATCCACCTTTGGGTCGCTTTGCTGTTCG
## ILMN_1343295 CTTCAACAGCGACACCCACTCCTCCACCTTTGACGCTGGGGCTGGCATTG
## ILMN_1651199 ATGCGAGGCCCCAGGGTTCGGCCCCGCAGCGCCGCTGAGTCCAAGGACCG
## ILMN_1651209 TCACGGCGTACGCCCTCATGGGGAAAATCTCCCCGGTGACTTTCAGGTCC
## ILMN_1651210 TGTGGACATGAGAGTTAGTTCTGTTTTGCCTGCACGGTGGGAGCGGCGTA
## ILMN_1651221 GCCGCCCCCTGCTTCACGGAGCCTGGTCCCATCAACCGCCGAAGGGCTGA
```

```
es$`diagnosis:ch1`
```

```
##  [1] "Lymphoma with no evidence of BM involvement"
##  [2] "Lymphoma with no evidence of BM involvement"
##  [3] "Lymphoma with no evidence of BM involvement"
##  [4] "Lymphoma with no evidence of BM involvement"
##  [5] "Lymphoma with no evidence of BM involvement"
##  [6] "Lymphoma with no evidence of BM involvement"
##  [7] "Lymphoma with no evidence of BM involvement"
##  [8] "MDS (RCMD)"
##  [9] "MDS (RCMD)"
## [10] "MDS (RCMD)"
## [11] "MDS (RAEB-1)"
## [12] "MDS (RAEB-1)"
## [13] "MDS (RAEB-1)"
## [14] "MDS (RAEB-2)"
```

```
#The condition is the "diagnosis:ch1" in this dataset:

es$condition <- gsub("\\+", "_", es$`diagnosis:ch1`)
es$condition
```

```
##  [1] "Lymphoma with no evidence of BM involvement"
##  [2] "Lymphoma with no evidence of BM involvement"
##  [3] "Lymphoma with no evidence of BM involvement"
##  [4] "Lymphoma with no evidence of BM involvement"
##  [5] "Lymphoma with no evidence of BM involvement"
##  [6] "Lymphoma with no evidence of BM involvement"
##  [7] "Lymphoma with no evidence of BM involvement"
##  [8] "MDS (RCMD)"
##  [9] "MDS (RCMD)"
## [10] "MDS (RCMD)"
## [11] "MDS (RAEB-1)"
## [12] "MDS (RAEB-1)"
## [13] "MDS (RAEB-1)"
## [14] "MDS (RAEB-2)"
```

```
#Remove "white spaces" and change with "_":
es$condition <- c("Lymphoma_with_no_evidence_of_BM_involvement", "Lymphoma_with_no_evidence_of_BM_involv
                  "Lymphoma_with_no_evidence_of_BM_involvement", "Lymphoma_with_no_evidence_of_BM_involv
                  "Lymphoma_with_no_evidence_of_BM_involvement", "Lymphoma_with_no_evidence_of_BM_involv
                  "Lymphoma_with_no_evidence_of_BM_involvement", "MDS_RCMD","MDS_RCMD","MDS_RCMD","MDS_
                  "MDS_RAEB_1","MDS_RAEB_1","MDS_RAEB_2")
es$condition
```

```
##  [1] "Lymphoma_with_no_evidence_of_BM_involvement"
##  [2] "Lymphoma_with_no_evidence_of_BM_involvement"
##  [3] "Lymphoma_with_no_evidence_of_BM_involvement"
##  [4] "Lymphoma_with_no_evidence_of_BM_involvement"
##  [5] "Lymphoma_with_no_evidence_of_BM_involvement"
##  [6] "Lymphoma_with_no_evidence_of_BM_involvement"
##  [7] "Lymphoma_with_no_evidence_of_BM_involvement"
##  [8] "MDS_RCMD"
##  [9] "MDS_RCMD"
## [10] "MDS_RCMD"
## [11] "MDS_RAEB_1"
## [12] "MDS_RAEB_1"
## [13] "MDS_RAEB_1"
## [14] "MDS_RAEB_2"
```

```
#Then we collapse the dataset with gene ID as in phantasus:

es <- collapseBy(es, fData(es)$`Gene symbol`, FUN=median)
es <- es[!grepl("///", rownames(es)), ]
es <- es[rownames(es) != "", ]
```

```
# there is a lot of garbage there.
# Annotate the symbols with human database entries:

fData(es) <- data.frame(row.names = rownames(es))
fData(es)$entrez <- row.names(fData(es))

fData(es)$symbol <- mapIds(org.Hs.eg.db, keys=fData(es)$entrez,
                           keytype="SYMBOL", column="ENTREZID" )
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
#To normalize the data:

es.qnorm <- es
summary(exprs(es.qnorm))
```

```
##    GSM1515746         GSM1515747         GSM1515748
##  Min.   :   74.36   Min.   :   77.43   Min.   :   72.49
##  1st Qu.:  156.33   1st Qu.:  157.17   1st Qu.:  156.79
##  Median :  207.08   Median :  208.67   Median :  207.01
##  Mean   :  848.73   Mean   :  868.10   Mean   :  850.16
##  3rd Qu.:  538.25   3rd Qu.:  547.00   3rd Qu.:  535.69
##  Max.   :54301.16   Max.   :54301.16   Max.   :54301.16
##    GSM1515749         GSM1515750         GSM1515751
##  Min.   :   66.39   Min.   :   78.78   Min.   :   72.49
##  1st Qu.:  157.29   1st Qu.:  156.84   1st Qu.:  155.96
##  Median :  206.31   Median :  206.08   Median :  204.77
##  Mean   :  869.38   Mean   :  872.05   Mean   :  887.72
##  3rd Qu.:  534.18   3rd Qu.:  540.33   3rd Qu.:  550.80
##  Max.   :54301.16   Max.   :54301.16   Max.   :54301.16
##    GSM1515752         GSM1515753         GSM1515754
##  Min.   :   72.49   Min.   :   66.39   Min.   :   76.36
##  1st Qu.:  156.11   1st Qu.:  155.58   1st Qu.:  157.62
##  Median :  204.85   Median :  202.60   Median :  207.99
##  Mean   :  872.93   Mean   :  871.65   Mean   :  864.45
##  3rd Qu.:  542.65   3rd Qu.:  542.79   3rd Qu.:  549.61
##  Max.   :54301.16   Max.   :52330.27   Max.   :54301.16
##    GSM1515755         GSM1515756         GSM1515757
##  Min.   :   66.39   Min.   :   76.36   Min.   :   66.39
##  1st Qu.:  157.24   1st Qu.:  155.67   1st Qu.:  157.03
##  Median :  205.12   Median :  202.25   Median :  208.42
##  Mean   :  841.01   Mean   :  871.26   Mean   :  891.78
##  3rd Qu.:  526.35   3rd Qu.:  545.36   3rd Qu.:  553.05
##  Max.   :52330.27   Max.   :54301.16   Max.   :54301.16
##    GSM1515758         GSM1515759
##  Min.   :   76.36   Min.   :   76.36
##  1st Qu.:  156.86   1st Qu.:  154.79
##  Median :  205.45   Median :  200.21
##  Mean   :  874.73   Mean   :  875.93
##  3rd Qu.:  544.31   3rd Qu.:  538.93
##  Max.   :54301.16   Max.   :54301.16
```

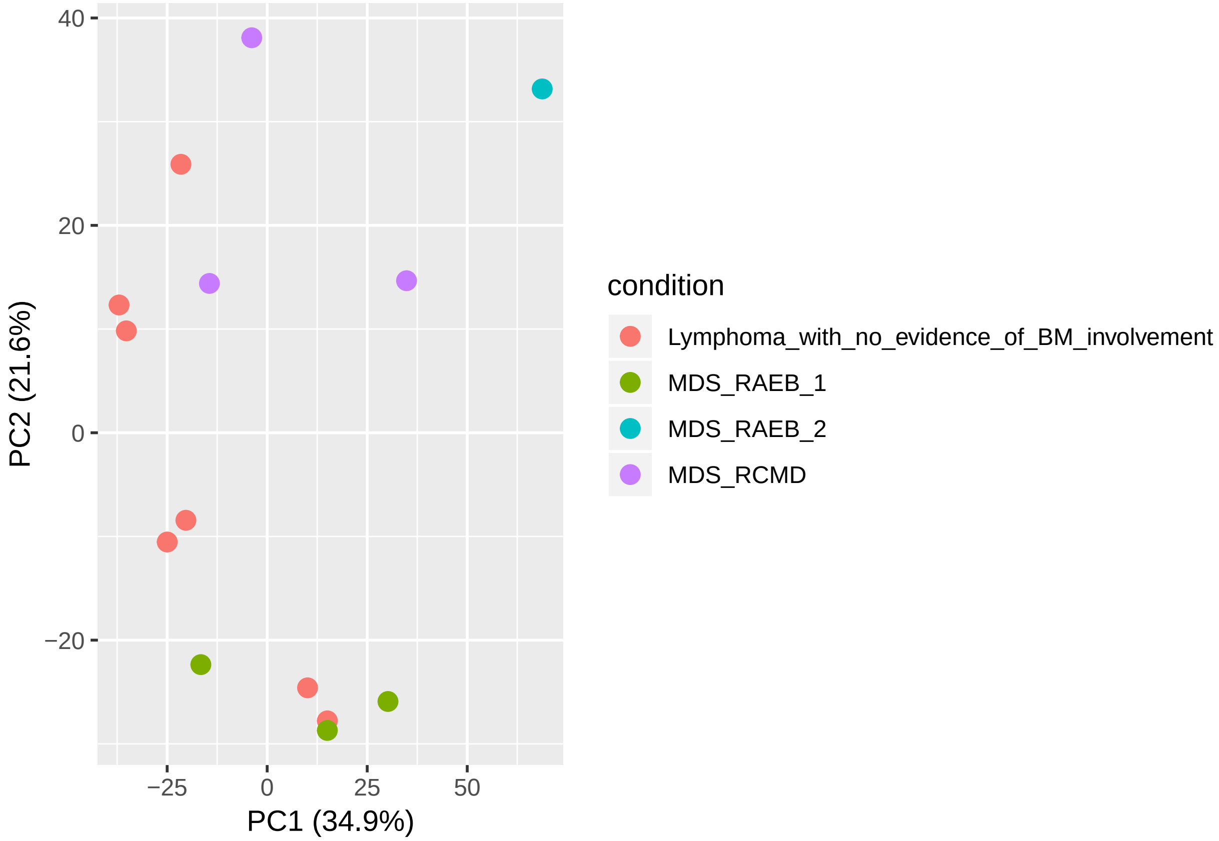
```r
exprs(es.qnorm) <- normalizeBetweenArrays(log2(exprs(es.qnorm)+1), method="quantile")
summary(exprs(es.qnorm))
```

```
##    GSM1515746       GSM1515747       GSM1515748       GSM1515749
##  Min.   : 6.203   Min.   : 6.203   Min.   : 6.203   Min.   : 6.203
##  1st Qu.: 7.299   1st Qu.: 7.299   1st Qu.: 7.299   1st Qu.: 7.299
##  Median : 7.690   Median : 7.690   Median : 7.690   Median : 7.690
##  Mean   : 8.354   Mean   : 8.354   Mean   : 8.354   Mean   : 8.354
##  3rd Qu.: 9.085   3rd Qu.: 9.085   3rd Qu.: 9.085   3rd Qu.: 9.085
##  Max.   :15.721   Max.   :15.721   Max.   :15.721   Max.   :15.721
##    GSM1515750       GSM1515751       GSM1515752       GSM1515753
##  Min.   : 6.203   Min.   : 6.203   Min.   : 6.203   Min.   : 6.203
##  1st Qu.: 7.299   1st Qu.: 7.299   1st Qu.: 7.299   1st Qu.: 7.299
##  Median : 7.690   Median : 7.690   Median : 7.690   Median : 7.690
##  Mean   : 8.354   Mean   : 8.354   Mean   : 8.354   Mean   : 8.354
##  3rd Qu.: 9.085   3rd Qu.: 9.085   3rd Qu.: 9.085   3rd Qu.: 9.085
##  Max.   :15.721   Max.   :15.721   Max.   :15.721   Max.   :15.721
##    GSM1515754       GSM1515755       GSM1515756       GSM1515757
##  Min.   : 6.203   Min.   : 6.203   Min.   : 6.203   Min.   : 6.203
##  1st Qu.: 7.299   1st Qu.: 7.299   1st Qu.: 7.299   1st Qu.: 7.299
##  Median : 7.690   Median : 7.690   Median : 7.690   Median : 7.690
##  Mean   : 8.354   Mean   : 8.354   Mean   : 8.354   Mean   : 8.354
##  3rd Qu.: 9.085   3rd Qu.: 9.085   3rd Qu.: 9.085   3rd Qu.: 9.085
##  Max.   :15.721   Max.   :15.721   Max.   :15.721   Max.   :15.721
##    GSM1515758       GSM1515759
##  Min.   : 6.203   Min.   : 6.203
##  1st Qu.: 7.299   1st Qu.: 7.299
##  Median : 7.690   Median : 7.690
##  Mean   : 8.354   Mean   : 8.354
##  3rd Qu.: 9.085   3rd Qu.: 9.085
##  Max.   :15.721   Max.   :15.721
```

```r
#To get get first 12000 entries:

es.qnorm.top12K <- es.qnorm
es.qnorm.top12K <- es.qnorm.top12K[head(order(apply(exprs(es.qnorm.top12K), 1, mean),
                                             decreasing = TRUE), 12000), ]
```

```r
#Have a look at the data  - make pca plot:

pcaPlot(es.qnorm.top12K,1,2) + aes(color = condition)
```

```
## Loading required package: ggplot2
```



```r
#To make a design matrix that will be used to make a model for given data:

es.design <- model.matrix(~0+condition, data=pData(es.qnorm.top12K))
es.design
```

```
##            conditionLymphoma_with_no_evidence_of_BM_involvement
## GSM1515746                                                    1
## GSM1515747                                                    1
## GSM1515748                                                    1
## GSM1515749                                                    1
## GSM1515750                                                    1
## GSM1515751                                                    1
## GSM1515752                                                    1
## GSM1515753                                                    0
## GSM1515754                                                    0
## GSM1515755                                                    0
## GSM1515756                                                    0
## GSM1515757                                                    0
## GSM1515758                                                    0
## GSM1515759                                                    0
##            conditionMDS_RAEB_1 conditionMDS_RAEB_2 conditionMDS_RCMD
## GSM1515746                   0                   0                 0
## GSM1515747                   0                   0                 0
## GSM1515748                   0                   0                 0
## GSM1515749                   0                   0                 0
## GSM1515750                   0                   0                 0
## GSM1515751                   0                   0                 0
## GSM1515752                   0                   0                 0
## GSM1515753                   0                   0                 1
## GSM1515754                   0                   0                 1
## GSM1515755                   0                   0                 1
## GSM1515756                   1                   0                 0
## GSM1515757                   1                   0                 0
## GSM1515758                   1                   0                 0
## GSM1515759                   0                   1                 0
## attr(,"assign")
## [1] 1 1 1 1
## attr(,"contrasts")
## attr(,"contrasts")$condition
## [1] "contr.treatment"
```

```r
#we have 4 conditions:
im <- data.frame(es.design)
colnames(im) <- c("conditionLymphoma_with_no_evidence_of_BM_involvement",
      "conditionMDS_RAEB_1", "conditionMDS_RAEB_2", "conditionMDS_RCMD")

rm(es.design)
es.design <- as.matrix(im)

#On the base of this matrix, we fit our data:
fit <- lmFit(es.qnorm.top12K, es.design)

#Also we make bayisian model for the data called fit2:
#NB! we need to choose contrast names which specify the sample groups to compare!
```

```r
# we need to specify the condion of interest and level to compare:

fit2 <- contrasts.fit(fit, makeContrasts(conditionLymphoma_with_no_evidence_of_BM_involvement, conditio
                                        levels=es.design))

fit2 <- eBayes(fit2)

#To do Bonferonni-hochback correction:
de <- topTable(fit2, adjust.method="BH", number=Inf)
head(de)
```

```
##            entrez symbol
## LAIR1       LAIR1   3903
## CLUAP1     CLUAP1  23059
## TMSB4X     TMSB4X   7114
## ITIH5       ITIH5  80760
## SLC7A5P2 SLC7A5P2 387254
## ANXA2P2   ANXA2P2    304
##          conditionLymphoma_with_no_evidence_of_BM_involvement
## LAIR1                                                15.55748
## CLUAP1                                               15.46982
## TMSB4X                                               15.58231
## ITIH5                                                15.28855
## SLC7A5P2                                             15.43726
## ANXA2P2                                              15.37897
##          conditionMDS_RAEB_1 conditionMDS_RAEB_2 conditionMDS_RCMD
## LAIR1               15.55440            15.63158          15.66142
## CLUAP1              15.44190            15.57380          15.51625
## TMSB4X              15.67929            15.72110          15.67083
## ITIH5               15.34486            15.47126          15.41513
## SLC7A5P2            15.41020            15.48561          15.56994
## ANXA2P2             15.43585            15.54890          15.47088
##           AveExpr        F      P.Value    adj.P.Val
## LAIR1    15.58438 46096.29 1.239895e-26 6.629047e-23
## CLUAP1   15.48121 44497.24 1.556615e-26 6.629047e-23
## TMSB4X   15.63197 42914.06 1.965888e-26 6.629047e-23
## ITIH5    15.34079 39909.42 3.138211e-26 6.629047e-23
## SLC7A5P2 15.46335 39265.21 3.485144e-26 6.629047e-23
## ANXA2P2  15.42299 39064.67 3.602046e-26 6.629047e-23
```

```r
# Here, we have a matrix that contains the enriched genes, we take the
#top genes and submit to database (msigdbr) to get the enriched pathways.
#We first target the hallmark pathways, which are well studied and
#then we target all the pathways. We try to find out what special pathways
#are involved in our normal versus condition. This will further give us insight
#into the comparision.
```

```r
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following object is masked from 'package:IRanges':
##
##     shift
## The following objects are masked from 'package:S4Vectors':
##
##     first, second
```

```
de <- as.data.table(de, keep.rownames=TRUE)
de[entrez == "LAIR1"]
```

```
##       rn entrez symbol
## 1: LAIR1  LAIR1   3903
##    conditionLymphoma_with_no_evidence_of_BM_involvement
## 1:                                             15.55748
##    conditionMDS_RAEB_1 conditionMDS_RAEB_2 conditionMDS_RCMD  AveExpr
## 1:             15.5544            15.63158          15.66142 15.58438
##           F      P.Value    adj.P.Val
## 1: 46096.29 1.239895e-26 6.629047e-23
```

```
#BioConductor: install fgsea:

library(fgsea)
```

```
## Loading required package: Rcpp
```

```
library(tibble)
library(Rcpp)
```

```
# To make a new matrix de2 which will store information about pathways:

de2 <- data.frame(de$entrez, de$P.Value)
colnames(de2) <- c('ENTREZ', 'stat')
```

```
# To get the rank of genes from top differentially expressed to non significant:

ranks <- deframe(de2)
head(ranks, 20)
```

```
##        LAIR1       CLUAP1       TMSB4X        ITIH5     SLC7A5P2
## 1.239895e-26 1.556615e-26 1.965888e-26 3.138211e-26 3.485144e-26
##      ANXA2P2         TPT1          F2R     FAM177A1       RRP7BP
## 3.602046e-26 3.866944e-26 4.834091e-26 5.963492e-26 8.455124e-26
##        RPS27       ZNF674         MYL6       EEF1A1         GNAS
## 1.963045e-25 2.113102e-25 2.255822e-25 4.231092e-25 4.886316e-25
##        ROCK2        PRR13        UBA52       RPL18A         MSH3
## 5.515444e-25 5.983581e-25 6.104485e-25 7.703442e-25 8.154177e-25
```

```
# Load the pathways into a named list:

library(msigdbr)
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

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

```
##     select
```

```
## The following objects are masked from 'package:IRanges':
##
##     collapse, desc, intersect, setdiff, slice, union
```

```
## The following objects are masked from 'package:S4Vectors':
##
##     first, intersect, rename, setdiff, setequal, union
```

```
## The following object is masked from 'package:Biobase':
##
##     combine
```

```
## The following objects are masked from 'package:BiocGenerics':
##
##     combine, intersect, setdiff, union
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
m_df <- msigdbr(species = "Homo sapiens")

# View(m_df):
pathways <- split(m_df$human_gene_symbol, m_df$gs_name)
head(pathways)
```

```
## $AAACCAC_MIR140
##   [1] "ABCC4"     "ACTN4"     "ACVR1"     "ADAM9"     "ADAMTS5"
##   [6] "AGER"      "ANK2"      "API5"      "BACH1"     "BAZ2B"
##  [11] "BCL11A"    "BCL2L2"    "BCL9"      "C15orf29"  "C1orf21"
##  [16] "C3orf58"   "C7orf60"   "CACNA1C"   "CEBPA"     "CHD4"
##  [21] "CIT"       "COL23A1"   "CSK"       "CSNK1G3"   "CTCF"
##  [26] "CUL3"      "DAZL"      "DBNDD2"    "DCUN1D4"   "DDX3X"
##  [31] "DDX3Y"     "DHX57"     "DPP4"      "DSCAM"     "DTNA"
##  [36] "E2F3"      "EHD1"      "EPHB1"     "ERC2"      "ETV3"
##  [41] "EYA2"      "FAM123A"   "FAM175B"   "FAM178A"   "GABARAP"
##  [46] "GALNTL1"   "GDF6"      "GIT1"      "GYS1"      "HDAC4"
##  [51] "HNRNPH3"   "HSPA13"    "IGFBP5"    "KCND2"     "KIAA1370"
##  [56] "LOC440742" "LOXL3"     "LRRC4"     "LRRC8E"    "MAP3K8"
##  [61] "MDGA2"     "MEX3C"     "MGAT1"     "MMD"       "NAV3"
##  [66] "NKIRAS2"   "NR3C1"     "NUTF2"     "OGT"       "OSTM1"
##  [71] "PDGFRA"    "PFN1"      "PHF20L1"   "PHYHIP"    "PITX2"
##  [76] "PPP1CC"    "PRIMA1"    "R3HDM1"    "REEP1"     "RNF19A"
##  [81] "RTKN2"     "SENP1"     "SIAH1"     "SLC25A13"  "SLC38A2"
##  [86] "SLC41A2"   "SLMAP"     "SNX2"      "SOX4"      "SRR"
##  [91] "STAG1"     "STRADB"    "SYT6"      "TAF9B"     "TBX3"
##  [96] "TP53INP2"  "TSHZ1"     "TSPAN2"    "TSSK2"     "TTYH2"
## [101] "UBASH3B"   "USP6"      "VEGFA"     "WHSC1L1"   "WNT1"
## [106] "YES1"      "ZBED4"     "ZBTB10"    "ZNF182"    "ZNF608"
## [111] "ZNF654"
##
## $AAAGACA_MIR511
##   [1] "ABCG8"     "ACE"       "ADAMTSL3"  "ADGRF5"    "ADSS"
##   [6] "AGBL3"     "ALCAM"     "ANKZF1"    "AQP6"      "ARHGEF17"
##  [11] "ATL2"      "ATP2B2"    "ATRX"      "BCL11A"    "BTG1"
##  [16] "BUB3"      "BZRAP1"    "C11orf51"  "C18orf34"  "C1orf21"
##  [21] "C1QL2"     "C21orf59"  "C2orf71"   "C5orf41"   "C6orf106"
##  [26] "C7orf23"   "C7orf42"   "CALM1"     "CAMK2N1"   "CAMTA1"
##  [31] "CAPRIN1"   "CCND1"     "CCNT2"     "CDH2"      "CDK14"
##  [36] "CDK19"     "CELF1"     "CELF6"     "CEP350"    "CLK2"
##  [41] "CLTC"      "CNOT4"     "CORIN"     "CREM"      "CRIM1"
##  [46] "DCTN4"     "DDX3X"     "DDX3Y"     "DEDD"      "DNAJB12"
##  [51] "DNAJC13"   "DSC1"      "DUSP6"     "DYRK1B"    "E2F3"
##  [56] "EDEM3"     "EFR3A"     "EIF2C1"    "EIF2C2"    "EIF2C4"
##  [61] "ELAVL3"    "EMILIN2"   "EML4"      "ENPP1"     "ENPP4"
##  [66] "EPHA4"     "ESRRG"     "EYA1"      "EYA4"      "FAM117A"
##  [71] "FAM60A"    "FGF13"     "FIP1L1"    "FMR1"      "FN1"
##  [76] "FNDC1"     "FNDC5"     "FOXK2"     "FOXN3"     "GAD2"
##  [81] "GEMIN2"    "GFAP"      "GJA1"      "GLRA2"     "GPR116"
##  [86] "HAS2"      "HCN4"      "HLF"       "HLTF"      "HOXA13"
##  [91] "IGF2BP1"   "IGF2BP3"   "KCNE1"     "KCNMA1"    "KHDRBS2"
##  [96] "KIAA1429"  "KLF9"      "KLHL18"    "KLHL24"    "LATS1"
## [101] "LINC00483" "LMCD1"     "LPP"       "LRCH4"     "LUC7L3"
## [106] "MAP3K2"    "MAP4K4"    "MAPK1IP1L" "MBD2"      "MBD6"
## [111] "MDGA2"     "METAP2"    "MIB1"      "MINK1"     "MRPL21"
## [116] "MSTN"      "MTAP"      "MYCBP"     "MYO19"     "NACC1"
## [121] "NEUROD6"   "NHLH2"     "NLK"       "NR4A2"     "NRXN3"
## [126] "NTRK2"     "NXPH1"     "ONECUT2"   "PAX8"      "PCDH10"
## [131] "PCDH17"    "PELI1"     "PHLPP1"    "PIK3R3"    "PMEPA1"
## [136] "POGK"      "POU4F2"    "PPARGC1A"  "PRELP"     "PRPF4B"
## [141] "PSMA1"     "PSMD10"    "QKI"       "RAB22A"    "RAB2A"
## [146] "RBM15B"    "RBM26"     "RECK"      "REV3L"     "RGL1"
## [151] "RHOJ"      "RHOT1"     "RNF19A"    "ROBO2"     "RPS6KB1"
## [156] "RPS6KL1"   "SATB2"     "SCN4B"     "SEMA3F"    "SEMA6D"
## [161] "SEPP1"     "SLC22A17"  "SLC25A26"  "SLC6A6"    "SLITRK1"
## [166] "SMARCE1"   "SOCS2"     "SORCS3"    "SOST"      "SOX12"
## [171] "SPTBN4"    "SPTLC2"    "SRGAP3"    "SS18"      "ST18"
## [176] "SYT11"     "T"         "TAF5"      "THOC5"     "TIAL1"
## [181] "TMEM196"   "TNRC6A"    "TNRC6B"    "TOB1"      "TRAPPC3"
## [186] "TRAPPC8"   "TRIM2"     "TRIM24"    "TXNL1"     "UBE2H"
## [191] "VANGL2"    "VAV3"      "VKORC1L1"  "VMP1"      "WNT16"
## [196] "YTHDF2"    "YY1"       "ZADH2"     "ZCCHC24"   "ZDHHC21"
## [201] "ZNF319"    "ZNF654"    "ZNF706"
##
## $AAAGGAT_MIR501
##   [1] "ACACA"    "ACADSB"   "ADCYAP1"  "ADIPOR2"  "ALS2"     "AMMECR1"
##   [7] "APOLD1"   "ATP6V1H"  "BCL6"     "BCLAF1"   "C8orf82"  "CA6"
##  [13] "CACHD1"   "CAMTA1"   "CCDC140"  "CD164"    "CELF2"    "CELSR2"
##  [19] "CHODL"    "CLK1"     "CLK2"     "CTDSP1"   "CTDSPL2"  "CUL1"
##  [25] "CUX2"     "DCX"      "DNAJB12"  "ELAVL4"   "ERRFI1"   "FAM179B"
##  [31] "GIF"      "GRAMD4"   "GRB10"    "H2AFX"    "HAS2"     "HES5"
##  [37] "HOXB8"    "JUN"      "KCND2"    "KCNRG"    "KIAA2022" "KIF1C"
##  [43] "KIF2A"    "KLHL14"   "KRR1"     "LARP1"    "LEPROTL1" "LPGAT1"
##  [49] "LPIN1"    "LRRC1"    "MAP2K1"   "MAP3K8"   "MCU"      "MEF2C"
##  [55] "MYB"      "MYCL1"    "MYLK"     "NFASC"    "NFIL3"    "NFIX"
##  [61] "NPR3"     "NR2F2"    "NR4A3"    "PCDH19"   "PDK1"     "PHC1"
##  [67] "PHF16"    "PHF6"     "PIK3AP1"  "PITX2"    "PLP1"     "PLXNB1"
##  [73] "PNN"      "PPP1CB"   "PPP2R5E"  "PPP6R3"   "PRKCE"    "PURA"
##  [79] "QKI"      "RAB22A"   "RABGEF1"  "RASL10B"  "RCN1"     "RDX"
##  [85] "RET"      "RGL1"     "RNF11"    "ROBO2"    "RPGRIP1L" "RSBN1"
##  [91] "SATB2"    "SCN3A"    "SENP3"    "SEPHS1"   "SGPP1"    "SLC25A3"
##  [97] "SLC35B3"  "SLITRK5"  "SMC1A"    "SMEK1"    "SNAP29"   "SOX11"
## [103] "SOX4"     "SPOPL"    "SRR"      "SRSF2"    "SYNC"     "SYNJ1"
## [109] "SYT7"     "TAF5L"    "TAPT1"    "TNNI2"    "TOMM70A"  "TRIM39"
## [115] "UBAP1"    "UBE2Q1"   "UBE4B"    "USP12"    "VDAC2"    "WDFY3"
## [121] "WIPF2"    "WT1-AS"   "ZC3H7A"   "ZIC4"     "ZMYM5"    "ZNF238"
##
## $AAAGGGA_MIR204_MIR211
##   [1] "ADAMTS9"  "ADCY6"    "AKAP1"    "ALPL"     "ANGPT1"   "ANKRD13A"
##   [7] "ANXA11"   "AP1S1"    "AP1S3"    "AP2A2"    "AP3M1"    "APH1A"
##  [13] "ARAP2"    "ARCN1"    "ARGLU1"   "ARHGAP29" "ARL8B"    "ATF2"
##  [19] "ATP2B1"   "AUP1"     "BAZ2A"    "BCL11B"   "BCL2"     "BCL9"
##  [25] "BCL9L"    "BRD4"     "BRPF3"    "BUD31"    "C16orf72" "C17orf48"
##  [31] "C1orf144" "C21orf63" "CAPRIN1"  "CCNT2"    "CCPG1"    "CDC25B"
##  [37] "CDC42"    "CDH2"     "CELSR3"   "CHD5"     "CHN2"     "CHP"
##  [43] "CLIP1"    "CORO1C"   "COX5A"    "CPD"      "CPNE8"    "CREB5"
##  [49] "CRKL"     "CTDNEP1"  "DAG1"     "DCAF5"    "DCUN1D3"  "DENND5A"
##  [55] "DHH"      "DLG5"     "DMTF1"    "DNAJC13"  "DNM2"     "DTX1"
##  [61] "DVL3"     "DYRK1A"   "EDEM1"    "EEF1E1"   "EFNB3"    "EIF2C4"
##  [67] "ELAVL3"   "ELF2"     "ELL2"     "ELMOD3"   "ELOVL6"   "EPHA7"
##  [73] "EPHB6"    "ESR1"     "ESRRG"    "EZR"      "FAM117B"  "FAM120C"
##  [79] "FAM122B"  "FAM160A2" "FAM175B"  "FARP1"    "FBN2"     "FBXW7"
##  [85] "FJX1"     "FNIP1"    "FRAS1"    "FREM1"    "FRY"      "GABRB3"
##  [91] "GAPVD1"   "GGA2"     "GLIS3"    "GPM6A"    "GRM1"     "HIC2"
##  [97] "HMGA2"    "HOOK3"    "HOXC8"    "HS2ST1"   "IGF2R"    "ING4"
## [103] "ITPR1"    "JPH3"     "KCNA3"    "KCTD1"    "KDM2A"    "KHDRBS1"
## [109] "KHDRBS3"  "KITLG"    "KLF12"    "KLHL13"   "LATS1"    "LRRC8D"
## [115] "MALL"     "MAML3"    "MAP1LC3B" "MAP3K3"   "MBNL1"    "MED13L"
## [121] "METAP1"   "MIR600HG" "MLL"      "MLLT3"    "MMGT1"    "MON2"
## [127] "MRPL35"   "MRPL52"   "MYO10"    "NAA15"    "NBEA"     "NCOA7"
## [133] "NEUROG1"  "NOVA1"    "NPTX1"    "NR3C1"    "NR4A2"    "NRBF2"
## [139] "NTRK2"    "P4HB"     "PCDH9"    "PHF13"    "PID1"     "PLAG1"
## [145] "POU3F2"   "PPARGC1A" "PPP3R1"   "PRDM2"    "PRPF38B"  "PRRX1"
## [151] "RAB10"    "RAB14"    "RAB1A"    "RAP2C"    "REEP1"    "RERE"
## [157] "RHOBTB3"  "RHOT1"    "RICTOR"   "RPS6KA3"  "RPS6KA5"  "RPS6KC1"
## [163] "RSPO3"    "RTKN2"    "RUNX2"    "SATB2"    "SCRT2"    "SEC24D"
## [169] "SEC61A2"  "SERINC3"  "SETD8"    "SF3B1"    "SGCZ"     "SGIP1"
## [175] "SHC1"     "SIN3A"    "SIRT1"    "SLC17A7"  "SLC22A2"  "SLC37A3"
## [181] "SLITRK4"  "SLTM"     "SMOC1"    "SOCS6"    "SOX11"    "SOX4"
## [187] "SPOP"     "SPRED1"   "SPRYD7"   "SSRP1"    "ST7"      "STXBP5"
## [193] "SUMO2"    "SUMO4"    "TAF5"     "TCF12"    "TCF7L1"   "TGFBR2"
## [199] "TMEM30A"  "TMOD3"    "TNRC6B"   "TP53INP1" "TRIAP1"   "TRIP12"
## [205] "TRPC5"    "TTYH1"    "UBE2R2"   "UHRF2"    "USP6"     "WEE1"
## [211] "WNT4"     "WSB1"     "XRN1"     "YTHDF3"   "YWHAG"    "ZCCHC14"
## [217] "ZCCHC24"  "ZDHHC17"  "ZFC3H1"   "ZFP91"    "ZFYVE20"  "ZNF282"
## [223] "ZNF335"   "ZNF423"
##
## $AAANWWTGC_UNKNOWN
##   [1] "ACTB"      "ADHFE1"    "AFF4"      "ANK2"      "ANK3"
##   [6] "APP"       "ASPA"      "ATOH7"     "ATP1B1"    "ATP2B4"
##  [11] "ATXN7L1"   "BCL11A"    "BCL6"      "BNC2"      "C11orf87"
##  [16] "C17orf85"  "CACNA1D"   "CACNG3"    "CALM1"     "CD14"
##  [21] "CDC42EP3"  "CDC42EP5"  "CDH13"     "CDK2AP1"   "CEPT1"
##  [26] "CHD2"      "CITED2"    "CNTFR"     "DAB1"      "DCAF11"
##  [31] "DCHS2"     "DDIT3"     "DIS3L"     "DLG2"      "DLGAP4"
##  [36] "DMD"       "DNAJB5"    "DPYSL5"    "DRD3"      "DSCAM"
##  [41] "DSEL"      "DSTN"      "DTX3L"     "DUSP1"     "DYNC1I2"
##  [46] "EBF1"      "EFNA5"     "EGFLAM"    "EIF4EBP2"  "ELAVL4"
##  [51] "ELF4"      "EPHA7"     "EPHB2"     "ESR1"      "FBXW7"
##  [56] "FGF7"      "FGFR2"     "FLJ45983"  "FN1"       "FOXN3"
##  [61] "FOXP1"     "FOXP2"     "FTHL17"    "FZD7"      "GANAB"
##  [66] "GATA3"     "GLRA2"     "GPC3"      "GPC6"      "GPR21"
##  [71] "GPRIN3"    "GRHL3"     "GRIN2B"    "GTF2E2"    "HEPACAM"
##  [76] "HHEX"      "HOXA2"     "HOXA3"     "HOXB2"     "HOXB6"
##  [81] "HOXC4"     "IGF2BP1"   "INHBA"     "ITM2C"     "KANK1"
##  [86] "KCNJ13"    "KLF12"     "KLF14"     "KRTAP8-1"  "LEAP2"
##  [91] "LECT1"     "LIPG"      "LOC148872" "LOX"       "LOXL4"
##  [96] "LRRC3B"    "LRRN1"     "LSAMP"     "LUC7L3"    "MAML3"
## [101] "MAN2A2"    "MAP3K4"    "MAPK3"     "MBNL1"     "MEF2C"
## [106] "MEIS1"     "MGLL"      "MID1"      "MLLT6"     "MMP3"
## [111] "MPZL3"     "MRPL24"    "MRPS18B"   "MYCL1"     "MYH2"
## [116] "MYLK"      "NEK6"      "NEUROG1"   "NFE2L2"    "NNAT"
## [121] "NR2F2"     "NRAS"      "NTN1"      "NTRK3"     "OLFM1"
## [126] "OLIG2"     "OMG"       "OTX2"      "PATZ1"     "PAX1"
## [131] "PAX6"      "PCSK1"     "PCTP"      "PDGFRB"    "PHF15"
## [136] "PHOX2B"    "PHTF1"     "PIK3R3"    "POU2F1"    "POU4F1"
## [141] "PPARGC1A"  "PPFIA2"    "PPP1R10"   "PPP2R2A"   "PPP3CC"
## [146] "PRDM16"    "PRIMA1"    "PRKRIR"    "PRPF4B"    "RAB10"
## [151] "RBMX"      "RORA"      "RRS1"      "RSPO2"     "S100PBP"
## [156] "SALL3"     "SAMD12"    "SATB2"     "SEMA6C"    "SESN2"
## [161] "SFRP2"     "SGCD"      "SHC3"      "SIX5"      "SKIL"
## [166] "SKP2"      "SLMAP"     "SNCAIP"    "SNX25"     "SORT1"
## [171] "SOX13"     "SOX4"      "SOX5"      "SPAG9"     "SPARCL1"
## [176] "SSBP3"     "STEAP2"    "TBC1D8B"   "TFAP4"     "TFDP2"
## [181] "TGIF1"     "THBS2"     "TLE4"      "TLK1"      "TLX3"
## [186] "TRAM1"     "TRPM3"     "TSC22D4"   "ZFPM1"     "ZHX3"
## [191] "ZNF462"    "ZNF827"    "ZW10"
##
## $AAAYRNCTG_UNKNOWN
##   [1] "ABT1"      "ACVR1"     "ADAM12"    "ADD3"      "AGGF1"
##   [6] "ANKRD12"   "ANKRD28"   "AP4S1"     "APBB2"     "APOBR"
##  [11] "AQP2"      "ARHGAP44"  "ARID1A"    "ARID4A"    "ARPC2"
##  [16] "ARSG"      "ARX"       "ASB4"      "ASPH"      "ATOH8"
##  [21] "ATP1A2"    "ATP5L"     "ATPIF1"    "AXDND1"    "B4GALT6"
##  [26] "BAI3"      "BAMBI"     "BCL2L1"    "BCL9"      "BMPR1B"
##  [31] "BMX"       "BRSK2"     "BTBD3"     "BUB3"      "C11orf84"
##  [36] "C11orf92"  "C12orf65"  "C13orf30"  "C14orf1"   "C15orf26"
##  [41] "C17orf28"  "C20orf197" "C3orf19"   "C6orf138"  "CA3"
##  [46] "CACNA2D3"  "CACNB2"    "CAPN1"     "CAPZA1"    "CASQ2"
##  [51] "CBX2"      "CCNJ"      "CCNY"      "CDC23"     "CDH2"
##  [56] "CER1"      "CHRM1"     "CITED2"    "CLDN5"     "CLTC"
##  [61] "CMKLR1"    "CNTLN"     "CNTN1"     "COCH"      "COL12A1"
##  [66] "COL1A2"    "COL4A5"    "COL4A6"    "COLEC10"   "CRAT"
##  [71] "CRH"       "CRKL"      "CRYGD"     "CRYGS"     "CSNK1A1"
##  [76] "CSRNP3"    "CSTF3"     "CYBRD1"    "DAAM1"     "DBNDD2"
##  [81] "DCAKD"     "DDAH2"     "DDX4"      "DEF6"      "DENND4A"
##  [86] "DGKB"      "DHH"       "DHRS4"     "DHRS4L2"   "DIDO1"
##  [91] "DMD"       "DMRT1"     "DNAJA2"    "DNAJB3"    "DNAJB4"
##  [96] "DSCAML1"   "DUSP4"     "DYNC1I1"   "DYRK1A"    "EDA"
## [101] "EFNA1"     "EGFLAM"    "EIF5"      "EMX2"      "EPC1"
## [106] "EPHA7"     "ERBB4"     "ERRFI1"    "ESRP2"     "ESRRB"
## [111] "ESRRG"     "EYA1"      "FAM49A"    "FAM83F"    "FCER1A"
## [116] "FGD4"      "FGF10"     "FGF12"     "FGFR1"     "FGFR1OP2"
## [121] "FIZ1"      "FKRP"      "FMNL3"     "FNDC9"     "FOXA1"
## [126] "FOXG1"     "FOXO4"     "FOXP2"     "FSIP2"     "FST"
## [131] "GABRA3"    "GDNF"      "GFI1"      "GGNBP2"    "GJB4"
## [136] "GLDN"      "GNAQ"      "GPR85"     "GPRC5D"    "GRIN2B"
## [141] "H3F3A"     "HDAC8"     "HESX1"     "HEXIM2"    "HGF"
## [146] "HIC2"      "HIP1R"     "HN1"       "HOXA10"    "HOXA5"
## [151] "HOXB8"     "HPSE2"     "HSD3B7"    "ICAM4"     "ID1"
## [156] "IGF1"      "IL1RAPL1"  "INHBC"     "IP6K2"     "ITGA10"
## [161] "ITGA8"     "JPH1"      "KANK2"     "KCNIP2"    "KCNK5"
## [166] "KCNN3"     "KCNQ1DN"   "KIAA0182"  "KITLG"     "KLF5"
## [171] "KLHDC10"   "KLHL20"    "KLHL3"     "LARS2"     "LENG9"
## [176] "LHFP"      "LHX9"      "LMO7"      "LOC151534" "LRP5"
## [181] "LRRC4"     "LRRN4CL"   "LTBP1"     "MAML1"     "MANF"
## [186] "MAP2"      "MAP3K5"    "MAP6"      "MEIS1"     "MGAT1"
## [191] "MGAT4A"    "MID1"      "MLL"       "MOAP1"     "MPP6"
## [196] "MPPED2"    "MRPL13"    "MTA2"      "MTBP"      "MYF6"
## [201] "MYH1"      "MYH10"     "MYO18A"    "NAGLU"     "NAPB"
## [206] "NAV2"      "NAV3"      "NCDN"      "NDNF"      "NDST4"
## [211] "NDUFS4"    "NEK1"      "NEK2"      "NFATC4"    "NFYB"
## [216] "NMI"       "NMT1"      "NR2F1"     "NRG1"      "NTRK2"
## [221] "NUP54"     "NXPH4"     "OMA1"      "OMG"       "OR2L13"
## [226] "OTX2"      "PACRG"     "PAPD5"     "PARK2"     "PART1"
## [231] "PCDH17"    "PCDH18"    "PCF11"     "PCYT1B"    "PDGFB"
## [236] "PDGFRA"    "PDLIM2"    "PDS5B"     "PDZRN4"    "PFN2"
## [241] "PHC2"      "PHEX"      "PHF1"      "PHF15"     "PHF6"
## [246] "PHOX2B"    "PLAGL2"    "PLEC"      "PLEKHM1"   "PLP2"
## [251] "PMCH"      "PMCHL1"    "PODXL2"    "POFUT1"    "POU2AF1"
## [256] "POU4F1"    "PPAP2B"    "PPP1R9B"   "PPP2R3A"   "PPP2R4"
## [261] "PPP2R5E"   "PPP3CA"    "PRELP"     "PRKCG"     "PRKCQ"
## [266] "PROK2"     "PTH1R"     "PXN"       "R3HDM1"    "RAB30"
## [271] "RAB5B"     "RAB5C"     "RAPGEF4"   "RBMS3"     "RGS17"
## [276] "RNF146"    "ROBO4"     "ROR1"      "RPLP0"     "RTN1"
## [281] "RUFY3"     "S1PR2"     "SCN3B"     "SCN5A"     "SCN8A"
## [286] "SCOC"      "SDCBP"     "SEMA6D"    "SEPT7"     "SESN3"
## [291] "SGCD"      "SH2D6"     "SHC3"      "SHCBP1L"   "SIPA1"
## [296] "SIRPA"     "SLC26A6"   "SLC4A1"    "SLC6A1"    "SMARCA2"
## [301] "SNX9"      "SORBS2"    "SOX12"     "SOX21"     "SOX30"
## [306] "SOX5"      "SPOCK2"    "SPTLC2"    "SRGAP2"    "SRSF8"
## [311] "SSBP2"     "ST7L"      "STAC3"     "STAG1"     "STAG2"
## [316] "STC2"      "STRN3"     "STRN4"     "TAS1R2"    "TEF"
## [321] "TFAP4"     "TFDP2"     "TM2D3"     "TMEM182"   "TMEM27"
## [326] "TMEM69"    "TMSB4X"    "TMSB4XP1"  "TMSL3"     "TMSL6"
## [331] "TNFAIP8"   "TNS1"      "TNXB"      "TP53INP2"  "TRDN"
## [336] "TREML1"    "TRIM28"    "TRIM68"    "TRIM8"     "TRIML1"
## [341] "TRPS1"     "TSC22D3"   "TSPAN7"    "TSPY26P"   "TSSK3"
## [346] "TTC17"     "TUSC2"     "UBE2W"     "UBXN10"    "USP1"
## [351] "VDR"       "VIP"       "VKORC1L1"  "VWA5A"     "WBP1"
## [356] "WNT2B"     "WT1"       "WT1-AS"    "XRCC1"     "ZADH2"
## [361] "ZBTB11"    "ZFP91"     "ZFPM2"     "ZIC1"      "ZIC4"
## [366] "ZMAT3"     "ZNF238"    "ZNF296"    "ZNF503"    "ZNF521"
## [371] "ZNF524"    "ZNF654"    "ZNF687"    "ZNF710"
```

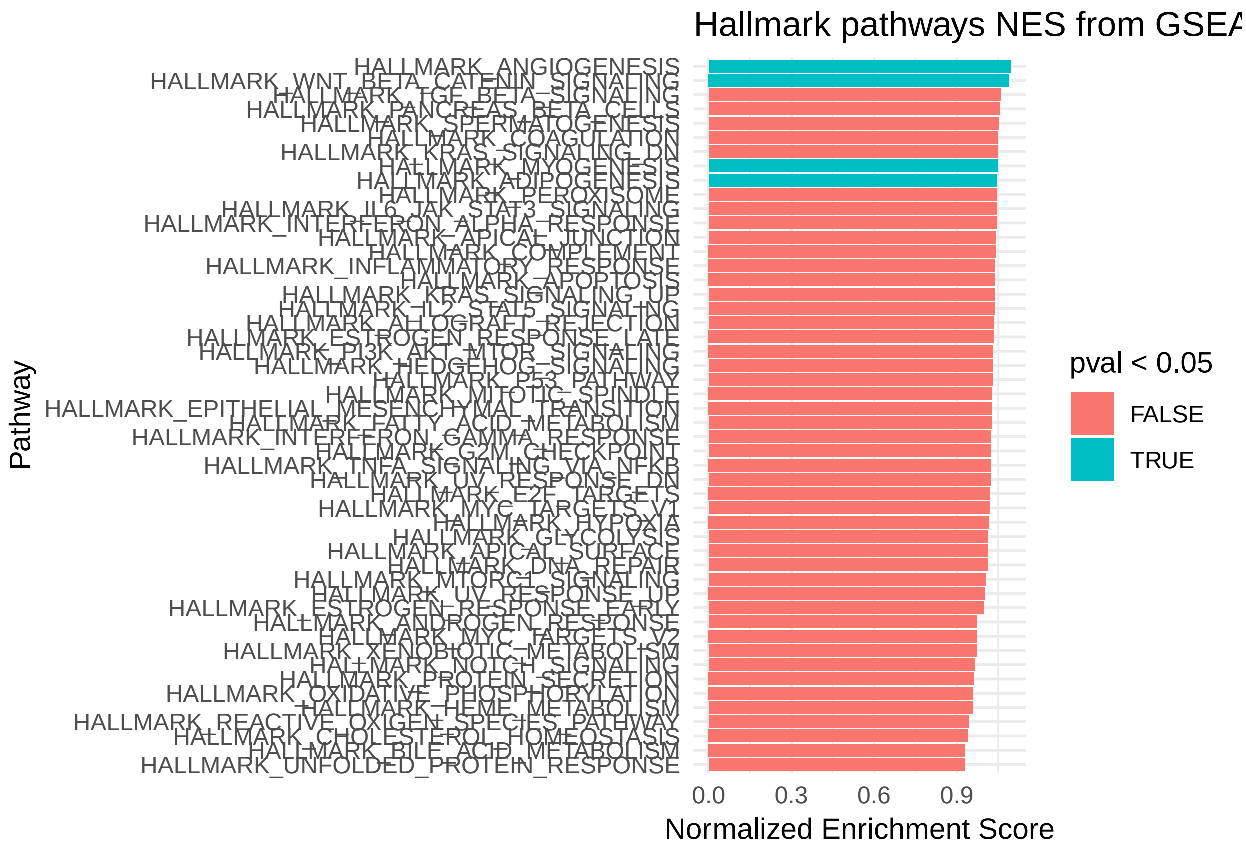
```
# filter the list to include only hallmark pathways:

library(dplyr)
library(data.table)

pathways.hallmark <- m_df[m_df$gs_name %like% "HALLMARK_", ]
pathways.hallmark <- split(pathways.hallmark$human_gene_symbol,
                           pathways.hallmark$gs_name)
```

```
# Show the first few pathways, and within those, show only the first few genes:

pathways.hallmark %>%
  head() %>%
  lapply(head)
```

```
## $HALLMARK_ADIPOGENESIS
## [1] "ABCA1" "ABCB8" "ACAA2" "ACADL" "ACADM" "ACADS"
##
## $HALLMARK_ALLOGRAFT_REJECTION
## [1] "AARS"   "ABCE1"  "ABI1"   "ACHE"   "ACVR2A" "AKT1"
##
## $HALLMARK_ANDROGEN_RESPONSE
## [1] "ABCC4"   "ABHD2"   "ACSL3"   "ACTN1"   "ADAMTS1" "ADRM1"
##
## $HALLMARK_ANGIOGENESIS
## [1] "APOH"   "APP"    "CCND2"  "COL3A1" "COL5A2" "CXCL6"
##
## $HALLMARK_APICAL_JUNCTION
## [1] "ACTA1" "ACTB"  "ACTC1" "ACTG1" "ACTG2" "ACTN1"
##
## $HALLMARK_APICAL_SURFACE
## [1] "ADAM10"  "ADIPOR2" "AFAP1L2" "AIM1"    "AKAP7"   "APP"
```

```
# To run the fgsea algorithm on hallmark.pathways:

fgseaEs <- fgsea(pathways=pathways.hallmark, stats=ranks, nperm=1000)
fgseaEsTidy <- fgseaEs %>%
  as_tibble() %>%
  arrange(desc(NES))  #ggploting for halmark pathways
```

```
# ggplot for hallmark pathways:
library(ggplot2)

 #pdf("fgseaEsTidy.pdf", width = 10, height = 20)

ggplot(fgseaEsTidy, aes(reorder(pathway, NES), NES)) +
  geom_col(aes(fill=pval<0.05)) +
  coord_flip() +
  labs(x="Pathway", y="Normalized Enrichment Score",
       title="Hallmark pathways NES from GSEA") +
  theme_minimal()
```



```
 #dev.off()
```

```
# We have plotted all the significant patways in the hallmark pathways as 'turquoise'
# We can see that:
    # HALLMARK_ANGIOGENESIS, HALLMARK_MYOGENESIS, etc.
# pathway are activated!
# Let's look at all pathways involving the following genes that were mentioned
#in the initial work (paper):
# IRF7, IFITM3, IFI35, IFITM1, IFITM2, MX2, MX1, IFI6, ISG15,AAAS, IFITM3, IFI35,
#HLA-DRB4, IFITM1, IFITM2,  MX2, MX1, IFI6, ISG15,HLA-DRA, CALR, UBE2M, IFI6,YWHAQ,
#AP3S1, YIPF6, VPS4B, CLINT1, STAM, VAMP2,NDUFB5, MPC2,ETFDH, ETFA, NDUFB5,TAF1B, LZTS1, MNAT1,EIF1AX,

# We are going to search the entire pathway list for any pathway
#that contains these genes, this can be done by subsetting and
#appending to a new dataframe of pathways.
```

```
# To make a list of all pathways fgseares.all:

fgseaEs.all <- fgsea(pathways=pathways, stats=ranks, nperm=1000)

item <- data.frame('IRF7', 'IFITM3', 'IFI35', 'IFITM1', 'IFITM2',
                   'MX2', 'MX1', 'IFI6', 'ISG15', 'AAAS', 'HLA-DRB4',
                   'IFITM1', 'IFITM2', 'HLA-DRA', 'CALR', 'UBE2M',
                   'IFI6', 'YWHAQ', 'AP3S1', 'YIPF6', 'VPS4B', 'CLINT1',
                   'STAM', 'VAMP2', 'NDUFB5', 'MPC2', 'ETFDH', 'ETFA',
                   'NDUFB5', 'TAF1B', 'LZTS1', 'MNAT1', 'EIF1AX', 'EIF3A',
                   'RPL31', 'UCRP', 'IFI6', 'IFIT1', 'IN35', 'PAR10',
                   'B1AJZ9', 'FHAD1', 'CE350', 'PTN7', 'PDCD4', 'PLEK2',
                   'ACHB4', 'BAG2', 'FA21A', 'YAP1', 'QCR2', 'ZCH18', 'TXNL1',
                   'MUC24', 'VATH', 'EIF3', 'ZCH18', 'RBX1', 'MUC24', 'TEBP',
                   'CL023', 'RGRF1', 'TXNL1', 'UGDH')

item<- t(item)
rownames(item) <- NULL

entry <- function(){

  x<- for (i in item){
    print(de[entrez == i])

  }

  return(x)

}
```

```
# searching for the genes in pathway and appending the rownumbers
#sink('numbers.csv')

options(max.print=2000)

for(i in item){
  print(grep(i, fgseaEs.all$leadingEdge))
}
```

```
## [1]  3120  7172 16460
## integer(0)
## [1] 16726
##   [1]   108   116   243   251   260   278   303   338   579   581   594
##  [12]   605   606   787   871   947   977  1401  1403  1443  1474  1476
##  [23]  1480  1492  1494  1496  1497  1500  1508  1517  1709  1728  1768
##  [34]  1775  1843  2827  2872  3119  3165  3170  3171  3908  3909  3912
##  [45]  3933  4679  4821  4841  4994  4995  4996  5266  5620  5669  5841
##  [56]  5887  5888  6342  6467  6482  6492  6710  6711  6846  6917  6918
##  [67]  7099  7192  7193  7248  7261  7276  7304  7305  7306  7378  7382
##  [78]  7579  8141  8143  8167  8239  8244  8245  8255  8409  8487  8495
##  [89]  8501  8507  8509  8513  8515  8519  8523  8551  8556  8567  8600
## [100]  8603  8604  8639  8647  8649  8681  8687  8688  8711  8715  8717
## [111]  8725  8726  8777  8977  9414  9469  9528  9774  9775  9776  9785
## [122]  9787  9810  9818  9820  9822  9834  9836  9878  9883  9887  9941
## [133]  9944 10047 10077 10084 10108 10122 10126 10165 10306 10312 10315
## [144] 10317 10465 10492 10527 10557 10559 10648 10649 10651 10668 10675
## [155] 10802 10821 10829 10831 10834 10863 10914 11243 11247 11288 11321
## [166] 11328 11347 11361 11375 11383 11424 11426 11430 11432 11452 11523
## [177] 11527 11629 11631 11634 11648 11654 11655 11657 11706 11714 11749
## [188] 11765 11767 11771 11781 11784 11787 11804 11808 11825 11852 11853
## [199] 11855 11862 11870 12034 12165 12169 12173 12175 12189 12191 12193
## [210] 12196 12214 12373 12488 12595 12601 12629 12648 12682 12686 12689
## [221] 12692 12713 12727 12729 12733 12789 12793 12805 12813 13008 13017
## [232] 13135 13136 13165 13208 13238 13274 13304 13307 13339 13353 13470
## [243] 13493 13495 13584 13849 13859 13954 13961 13998 14022 14054 14063
## [254] 14065 14090 14105 14116 14224 14239 14330 14332 14352 14358 14364
## [265] 14394 14396 14433 14475 14476 14514 14554 14585 14602 14631 14701
## [276] 14719 14734 14795 14806 14810 15070 15077 15085 15087 15093 15096
## [287] 15184 15284 15305 15307 15353 15416 15444 15463 15466 15692 15779
## [298] 15800 15857 15960 16099 16100 16120 16122 16521 16530 16548 16562
## [309] 16592 16717 16739 16740 16763 16773 16892 16941 17007 17035 17037
## [320] 17039 17043 17053 17207 17208 17209 17262 17274 17323 17326 17368
## [331] 17386 17421 17479 17523 17533 17538 17548 17572 17727 17732 17738
## [342] 17777 17779 17781
## integer(0)
##  [1]   174   820   971  1971  2757  3188  3610  3616  3826  4133  7817
## [12] 13260 14267 14818 15814 16546 16619 17508
##   [1]   115   119   204   243   294   300   602   605   606   652   680
##  [12]   683   851   852   864   871   939  1102  1400  1443  1470  1492
##  [23]  1494  1569  1599  1618  1660  1728  1729  1768  2720  2827  2872
##  [34]  3119  3165  3170  3171  3427  3772  3777  3887  3908  3909  3912
##  [45]  3933  4841  4995  4996  5107  5119  5123  5374  6846  7099  7193
##  [56]  7248  7261  7276  7378  7382  7435  8124  8133  8148  8150  8154
##  [67]  8239  8245  8255  8345  8434  8501  8507  8509  8515  8518  8519
##  [78]  8522  8523  8558  8568  8600  8604  8641  8644  8647  8649  8667
##  [89]  8673  8677  8686  8720  8812  8860  8862  8864  8872  8873  8876
## [100]  8878  8881  8893  8957  9005  9100  9115  9125  9135  9172  9204
## [111]  9220  9222  9240  9267  9304  9348  9419  9423  9440  9463  9467
## [122]  9469  9471  9473  9475  9477  9483  9501  9523  9528  9542  9547
## [133]  9551  9562  9582  9586  9590  9603  9614  9658  9679  9696  9775
## [144]  9776  9777  9788  9810  9834  9878  9883  9885  9887  9943  9944
## [155]  9945  9950  9991  9998 10137 10149 10215 10217 10218 10317 10358
## [166] 10383 10437 10478 10485 10512 10737 10800 10805 10834 10967 10974
## [177] 11127 11180 11187 11212 11215 11224 11235 11243 11249 11260 11282
## [188] 11289 11295 11311 11321 11334 11335 11345 11347 11349 11351 11372
## [199] 11403 11446 11452 11481 11565 11590 11592 11629 11631 11634 11637
## [210] 11643 11655 11657 11683 11691 11702 11723 11750 11765 11910 11944
## [221] 11955 11973 12014 12018 12022 12023 12026 12031 12034 12150 12165
## [232] 12169 12173 12175 12179 12189 12191 12193 12196 12210 12214 12218
## [243] 12243 12248 12255 12269 12275 12296 12302 12303 12307 12431 12466
## [254] 12602 12635 12682 12686 12689 12692 12698 12789 12797 12805 12845
## [265] 12956 12961 13008 13091 13136 13137 13138 13167 13189 13202 13208
## [276] 13309 13367 13448 13483 13535 13836 13842 13850 13859 14040 14423
## [287] 14434 14449 14476 14492 14539 14554 14594 14602 14619 14627 14631
## [298] 14644 14701 14704 14719 14742 14806 14816 15070 15284 15305 15455
## [309] 15469 15471 15779 15878 15960 16099 16120 16122 16566 16592 16595
## [320] 16650 16652 16678 16743 16941 17035 17037 17039 17043 17053 17189
## [331] 17262 17326 17368 17413 17479 17556 17568 17727 17732 17777 17779
## [342] 17781
##  [1]   656  1078  1652  2416  4746  4945  6632  6806  6826  6829  7206
## [12] 10057 10071 10335 10399 10944 12535 12539 13091 13309 14184 15099
## [23] 17196 17478 17555
## [1] 13344 13354
## integer(0)
## [1] 8161
##   [1]   108   116   243   251   260   278   303   338   579   581   594
##  [12]   605   606   787   871   947   977  1401  1403  1443  1474  1476
##  [23]  1480  1492  1494  1496  1497  1500  1508  1517  1709  1728  1768
##  [34]  1775  1843  2827  2872  3119  3165  3170  3171  3908  3909  3912
##  [45]  3933  4679  4821  4841  4994  4995  4996  5266  5620  5669  5841
##  [56]  5887  5888  6342  6467  6482  6492  6710  6711  6846  6917  6918
##  [67]  7099  7192  7193  7248  7261  7276  7304  7305  7306  7378  7382
##  [78]  7579  8141  8143  8167  8239  8244  8245  8255  8409  8487  8495
##  [89]  8501  8507  8509  8513  8515  8519  8523  8551  8556  8567  8600
## [100]  8603  8604  8639  8647  8649  8681  8687  8688  8711  8715  8717
## [111]  8725  8726  8777  8977  9414  9469  9528  9774  9775  9776  9785
## [122]  9787  9810  9818  9820  9822  9834  9836  9878  9883  9887  9941
## [133]  9944 10047 10077 10084 10108 10122 10126 10165 10306 10312 10315
## [144] 10317 10465 10492 10527 10557 10559 10648 10649 10651 10668 10675
## [155] 10802 10821 10829 10831 10834 10863 10914 11243 11247 11288 11321
## [166] 11328 11347 11361 11375 11383 11424 11426 11430 11432 11452 11523
## [177] 11527 11629 11631 11634 11648 11654 11655 11657 11706 11714 11749
## [188] 11765 11767 11771 11781 11784 11787 11804 11808 11825 11852 11853
## [199] 11855 11862 11870 12034 12165 12169 12173 12175 12189 12191 12193
## [210] 12196 12214 12373 12488 12595 12601 12629 12648 12682 12686 12689
## [221] 12692 12713 12727 12729 12733 12789 12793 12805 12813 13008 13017
## [232] 13135 13136 13165 13208 13238 13274 13304 13307 13339 13353 13470
## [243] 13493 13495 13584 13849 13859 13954 13961 13998 14022 14054 14063
## [254] 14065 14090 14105 14116 14224 14239 14330 14332 14352 14358 14364
## [265] 14394 14396 14433 14475 14476 14514 14554 14585 14602 14631 14701
## [276] 14719 14734 14795 14806 14810 15070 15077 15085 15087 15093 15096
## [287] 15184 15284 15305 15307 15353 15416 15444 15463 15466 15692 15779
## [298] 15800 15857 15960 16099 16100 16120 16122 16521 16530 16548 16562
## [309] 16592 16717 16739 16740 16763 16773 16892 16941 17007 17035 17037
## [320] 17039 17043 17053 17207 17208 17209 17262 17274 17323 17326 17368
## [331] 17386 17421 17479 17523 17533 17538 17548 17572 17727 17732 17738
## [342] 17777 17779 17781
## integer(0)
##   [1]    56   116   298   301   339   372   383   407   410   469   472
##  [12]   553   558   595   645   653   699   728  1203  1402  1707  1837
##  [23]  2120  2254  2328  2391  2392  2396  2398  2400  2765  2797  2827
##  [34]  2855  2872  3011  3012  3013  3014  3024  3025  3041  3119  3144
##  [45]  3165  3374  3376  3406  3407  3461  3462  3740  3741  3748  3909
##  [56]  3910  3912  3933  3975  3985  4001  4016  4022  4024  4092  4093
##  [67]  4182  4196  4211  4212  4218  4219  4308  4357  4358  4359  4361
##  [78]  4362  5020  5123  5238  5340  5498  5607  5608  5610  5745  5746
##  [89]  5958  6124  6125  6129  6329  6451  6452  6455  6698  6710  6711
## [100]  7261  7306  7575  7579  7587  7794  7805  7873  7874  7953  7954
## [111]  8022  8023  8026  8061  8175  8191  8243  8251  8282  8438  8639
## [122]  8649  9005  9017  9022  9025  9052  9153  9576  9588  9592  9595
## [133]  9690  9698  9755  9993 10041 10043 10070 10072 10078 10087 10121
## [144] 10123 10329 10337 10378 10429 10548 10627 10633 10648 10699 10821
## [155] 10849 10865 10877 10907 10925 10927 10933 10935 10990 11018 11175
## [166] 11209 11298 11321 11454 11468 11718 11746 11754 11776 11792 11806
## [177] 11822 11843 11847 12008 12033 12036 12038 12041 12049 12079 12158
## [188] 12203 12228 12229 12272 12345 12437 12616 12846 12969 13112 13203
## [199] 13208 13211 13334 13339 13369 13372 13378 13495 13569 13575 13581
## [210] 13582 13594 13640 13642 13649 13651 13727 13737 13747 14135 14448
## [221] 14476 14514 14529 14566 14585 14586 14631 14701 14719 14737 14770
## [232] 14795 14806 15189 15286 15353 15484 15531 15578 15579 15670 15744
## [243] 15752 16644 16853 17159 17283 17368 17463 17481 17488 17582
## integer(0)
## integer(0)
##  [1]   656  1078  1652  2416  4746  4945  6632  6806  6826  6829  7206
## [12] 10057 10071 10335 10399 10944 12535 12539 13091 13309 14184 15099
## [23] 17196 17478 17555
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## integer(0)
##  [1]   968  1801  1859  8151  8348  8354  8671  8672  8678  8740  8793
## [12]  8982  8989  9000  9204  9263  9312  9339  9356  9407  9422  9437
## [23]  9454  9654  9855 10142 10143 10144 10158 10203 10208 10262 10270
## [34] 10282 10508 10591 10660 10757 10773 10788 10880 11002 11269 11281
## [45] 11475 11544 11558 11709 11712 11752 11869 11980 12048 12095 12111
## [56] 12139 12141 12423 12561 12564 12636 12784 12921 13778 14282 15448
## [67] 17362
## [1] 15832 16211
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## [1] 7503
## integer(0)
##  [1]   201  1018  1298  2299  2635  3267  3300  3302  3857  4505  5134
## [12]  5180  7451  7469  7482  7883  7896  8048  8404  8664  9095  9790
## [23] 10046 10595 10694 10710 11873 12084 12096 13260 13456 13558 15958
## [34] 16022 16023 16024 16026 16082 16083 16139 16177 16180 16343 16344
## [45] 16345 16346 16347 16348 16349 16466
## [1]   422   569  1510 13103 14792 14897 14899 15035 16710
## [1]   422   493  1684  3511  7935 15581
## integer(0)
## integer(0)
##  [1]   656  1078  1652  2416  4746  4945  6632  6806  6826  6829  7206
## [12] 10057 10071 10335 10399 10944 12535 12539 13091 13309 14184 15099
## [23] 17196 17478 17555
##   [1]    38    89   251   286   605   616   678   724   744   808   823
##  [12]   900   970  1352  1354  1410  1414  1426  1443  1524  1534  1543
##  [23]  1725  1809  1821  1847  2831  2839  2871  2940  3860  3873  3971
##  [34]  4512  4632  4708  4709  4780  4841  4902  4926  4995  4996  6042
##  [45]  6043  6352  6556  6558  6559  6709  6969  7031  7265  7275  7341
##  [56]  8154  8169  8181  8228  8255  8363  8411  8561  8587  8657  8658
##  [67]  8683  8845  8956  8967  8983  9000  9016  9452  9455  9459  9461
##  [78]  9467  9469  9473  9483  9498  9536  9541  9547  9551  9553  9573
##  [89]  9590  9617  9786  9799  9849  9863  9884  9986 10002 10111 10295
## [100] 10352 10390 10395 10467 10529 10567 10575 10577 10585 10703 10724
## [111] 10855 10875 10983 10987 10989 11152 11235 11251 11282 11316 11321
## [122] 11358 11361 11432 11456 11483 11537 11553 11611 11619 11624 11643
## [133] 11669 11708 11754 11872 11945 11950 11957 11972 11980 11990 11999
## [144] 12041 12050 12059 12098 12112 12115 12130 12138 12146 12150 12216
## [155] 12314 12501 12513 12526 12584 12621 12625 12657 12881 12887 12890
## [166] 12894 12947 13156 13161 13175 13321 13372 13462 13483 13535 13763
## [177] 13785 13878 14029 14184 14193 14195 14394 14435 14533 14541 14680
## [188] 14715 15287 15369 15706 15734 15811 15878 16568 16767 16940 16941
## [199] 17139 17140 17207 17382 17484 17533 17556 17568 17630 17779 17798
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## [1]  1750  4804  5842 13800
## integer(0)
## integer(0)
## [1] 1604
## integer(0)
##  [1]  2475  2564  3825  4095  5323  6093  7085  9752 10874 11154 12354
## [12] 15673 16218 16389 16496 16877 17017
## integer(0)
## integer(0)
##   [1]     2    42    56    87   873  1059  1500  1503  2759  2791  2874
##  [12]  3245  3509  3860  4424  5282  5286  5288  5289  6106  6138  6144
##  [23]  7344  7620  7760  8183  8293  8353  8513  8567  8901  8931  8968
##  [34]  9045  9101  9123  9131  9133  9141  9256  9260  9644 10008 10049
##  [45] 10218 10242 10249 10259 10285 10291 10345 10683 10687 10720 10843
##  [56] 10953 11056 11330 11340 11421 11425 11476 11479 11493 11524 11601
##  [67] 11690 11751 11762 11857 12064 12083 12288 12302 12473 12475 12481
##  [78] 12742 12792 12806 12832 12838 12956 13013 13015 13437 13953 14204
##  [89] 14384 14568 14646 14670 15275 15292 15465 15742 15754 15824 16542
## [100] 16556 16986 17125 17158 17228 17728
## integer(0)
## integer(0)
##  [1]   422   493  1684  3511  3619  7934  7935 14419 14453 14813 15581
## [12] 17571
## integer(0)
## [1]  3092 16284
## integer(0)
## integer(0)
## integer(0)
## integer(0)
##   [1]     2    42    56    87   873  1059  1500  1503  2759  2791  2874
##  [12]  3245  3509  3860  4424  5282  5286  5288  5289  6106  6138  6144
##  [23]  7344  7620  7760  8183  8293  8353  8513  8567  8901  8931  8968
##  [34]  9045  9101  9123  9131  9133  9141  9256  9260  9644 10008 10049
##  [45] 10218 10242 10249 10259 10285 10291 10345 10683 10687 10720 10843
##  [56] 10953 11056 11330 11340 11421 11425 11476 11479 11493 11524 11601
##  [67] 11690 11751 11762 11857 12064 12083 12288 12302 12473 12475 12481
##  [78] 12742 12792 12806 12832 12838 12956 13013 13015 13437 13953 14204
##  [89] 14384 14568 14646 14670 15275 15292 15465 15742 15754 15824 16542
## [100] 16556 16986 17125 17158 17228 17728
##  [1]  1812  3661  8584 13580 13689 13723 14579 16062 17180 17462
```

```
#sink()
```

```
# Have to do a lot of cleaning of the data before importing it as csv (to make all values
#in each cell separately inside one column):
# getting only unique values from all numbers, because one gene may overlap with other,
```

```r
#we only want the unique  #row numbers:

new_numbers <- read.csv("C://Users//Natalia//Desktop//ITMO//SystemBiology//RNAseq_analysis//RNAseq_analy

unique_vals <- data.frame(as.integer(unique(unlist(new_numbers))))
colnames(unique_vals) <- c('row_number')

new_unique_vals <- na.omit(unique_vals)

pathways.final <- subset(fgseaEs.all, rownames(fgseaEs.all) %in% new_unique_vals$row_number)

View(pathways.final)
```

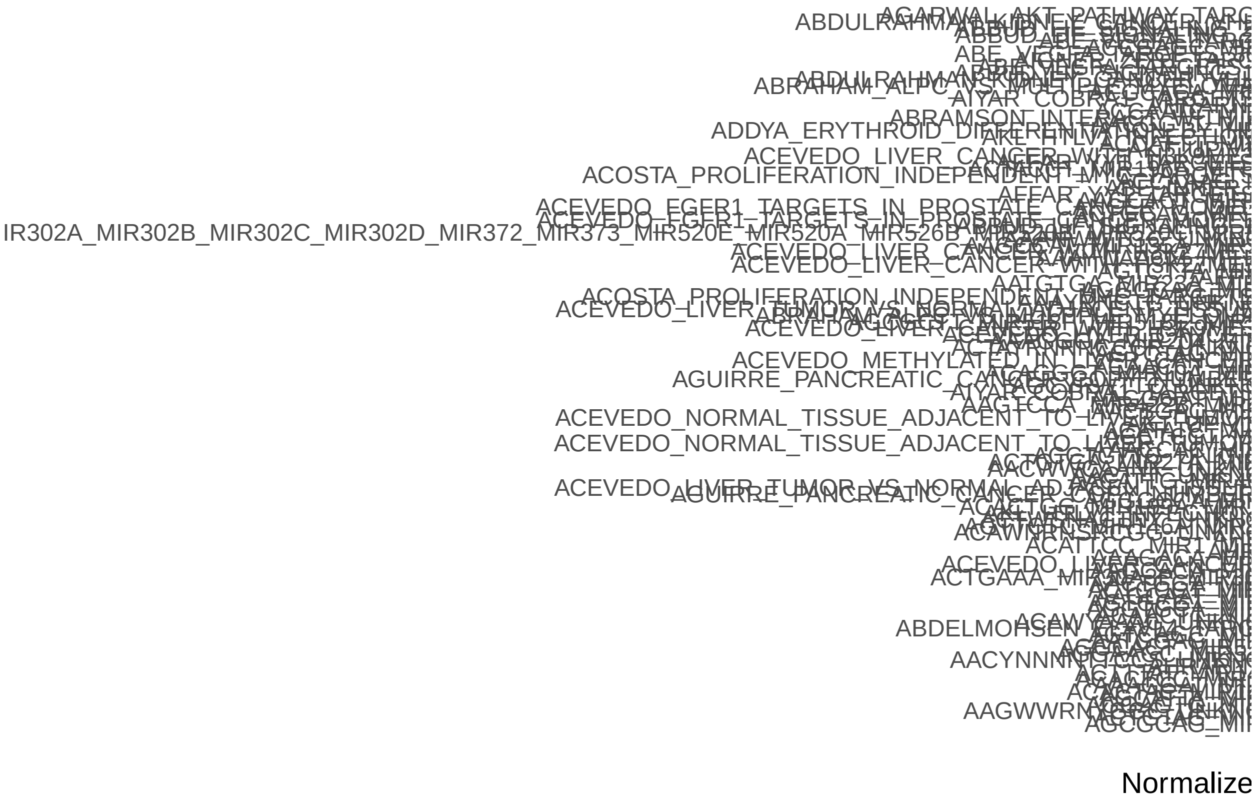
```r
# Show the first few pathways, and within those, show only the first few genes:

pathways.final %>%
  head() %>%
  lapply(head)
```

```
## $pathway
## [1] "AAACCAC_MIR140"        "AAAGACA_MIR511"        "AAAGGAT_MIR501"
## [4] "AAAGGGA_MIR204_MIR211" "AAANWWTGC_UNKNOWN"     "AAAYRNCTG_UNKNOWN"
##
## $pval
## [1] 0.6753247 0.8541459 0.9460539 0.5264735 0.2297702 0.3946054
##
## $padj
## [1] 0.9363187 0.9887099 1.0000000 0.8951302 0.7966539 0.8496807
##
## $ES
## [1] 0.9352824 0.9267576 0.8707464 0.9636525 0.9789520 0.9717770
##
## $NES
## [1] 0.9892724 0.9688885 0.9177629 1.0045162 1.0251431 1.0095832
##
## $nMoreExtreme
## [1] 675 854 946 526 229 394
##
## $size
## [1]  78 132  87 169 124 230
##
## $leadingEdge
## $leadingEdge[[1]]
## [1] "LOXL3"   "ADAMTS5" "IGFBP5"  "MAP3K8"  "PDGFRA"  "PITX2"   "MEX3C"
## [8] "ADAM9"
##
## $leadingEdge[[2]]
##  [1] "FNDC1"   "TXNL1"   "CRIM1"   "CAMK2N1" "GJA1"    "LMCD1"   "ANKZF1"
##  [8] "CLTC"    "NR4A2"   "LUC7L3"  "HAS2"
##
## $leadingEdge[[3]]
## [1] "MAP3K8"   "PITX2"    "NR2F2"    "FAM179B"  "HAS2"     "SYNC"
## [7] "SYT7"     "PLXNB1"   "LEPROTL1"
##
```

```
## $leadingEdge[[4]]
## [1] "MALL"   "ALPL"   "CPNE8"  "SEC24D" "ELOVL6"
##
## $leadingEdge[[5]]
## [1] "MMP3"  "DSEL"  "LOXL4" "HOXA3"
##
## $leadingEdge[[6]]
## [1] "NEK2"    "COL4A5"  "ID1"     "COL12A1" "LRRN4CL"
```

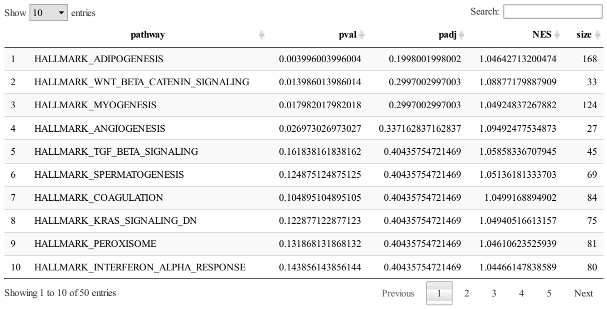
```
final <- data.frame(pathways.final)
# running the fgsea algorithm on final pathways
# Let's look at the plot
```

```
# ggplot for final pathways:

library(ggplot2)

 #pdf('final_pathways.pdf', width=15, height = 120 )
ggplot(final, aes(reorder(pathway, NES), NES)) +
  geom_col(aes(fill=pval<0.05)) +
  coord_flip() +
  labs(x="Pathway", y="Normalized Enrichment Score",
       title="Selected genes from the study") +
  theme_minimal()
```



```
 #dev.off()
```

```
# install.packages('DT')
library(DT)

# Show in a table for all pathways:

fgseaEsTidy %>%
  dplyr::select(-leadingEdge, -ES, -nMoreExtreme) %>%
  arrange(padj) %>%
  DT::datatable()
```



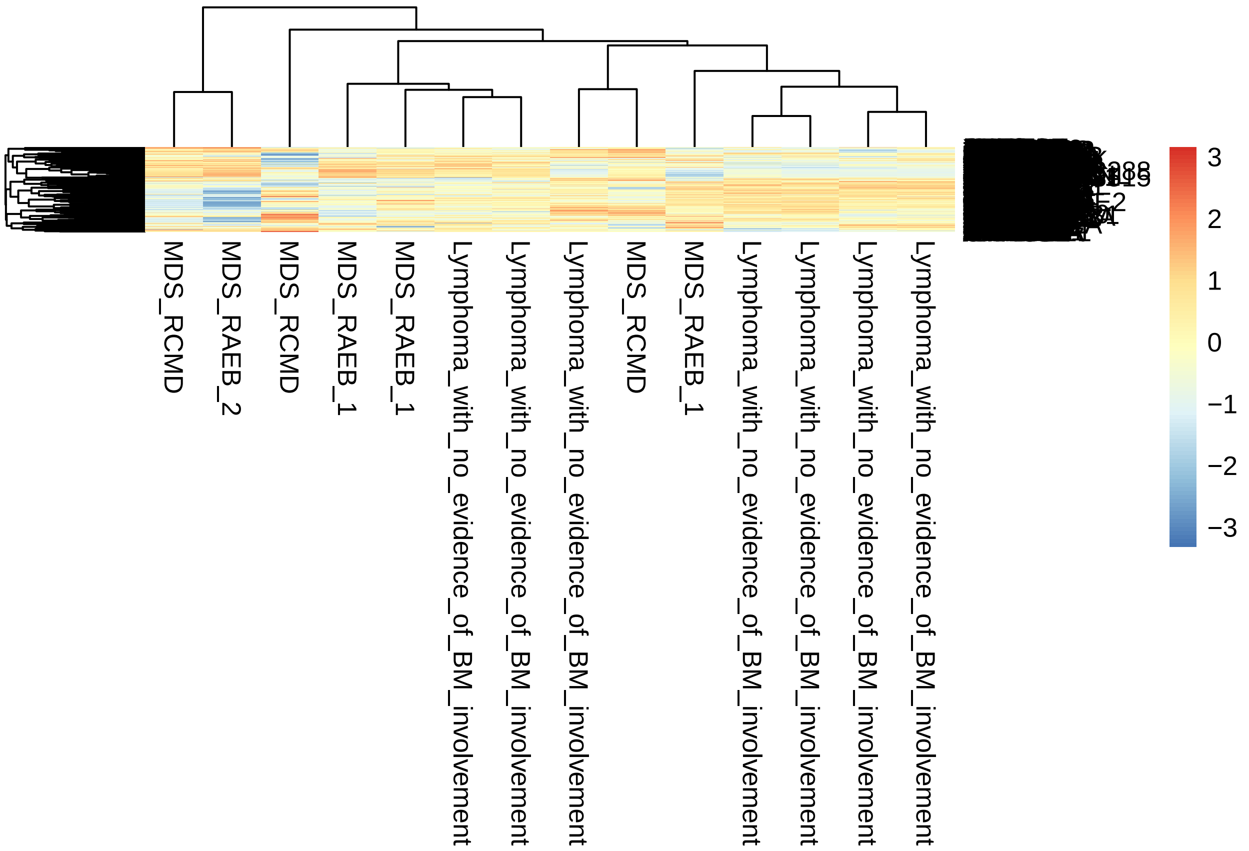
	pathway	pval	padj	NES	size
1	HALLMARK_ADIPOGENESIS	0.003996003996004	0.1998001998002	1.04642713200474	168
2	HALLMARK_WNT_BETA_CATENIN_SIGNALING	0.013986013986014	0.2997002997003	1.08877179887909	33
3	HALLMARK_MYOGENESIS	0.017982017982018	0.2997002997003	1.04924837267882	124
4	HALLMARK_ANGIOGENESIS	0.026973026973027	0.337162837162837	1.09492477534873	27
5	HALLMARK_TGF_BETA_SIGNALING	0.161838161838162	0.40435754721469	1.05858336707945	45
6	HALLMARK_SPERMATOGENESIS	0.124875124875125	0.40435754721469	1.05136181333703	69
7	HALLMARK_COAGULATION	0.104895104895105	0.40435754721469	1.0499168894902	84
8	HALLMARK_KRAS_SIGNALING_DN	0.122877122877123	0.40435754721469	1.04940516613157	75
9	HALLMARK_PEROXISOME	0.131868131868132	0.40435754721469	1.04610623525939	81
10	HALLMARK_INTERFERON_ALPHA_RESPONSE	0.143856143856144	0.40435754721469	1.04466147838589	80

Showing 1 to 10 of 50 entries

```
# heatmap
library(pheatmap)
```

```
#scale rows
xt <-t(as.matrix(es.qnorm.top12K)) # this is a matrix of normalised 12k genes

# To get a heatmap of 1000 genes:

xts <-scale(xt)
xtst <-t(xts)
xtst <- na.omit(xtst)
colnames(xtst) <- es$condition

#only grab top 1000 by p-value:
h <- head(xtst, n = 1000L)

#set layout options - adjust if labels get cut off
#pdf("heatmap.pdf",width=5, height=100)

#draw heatmap allowing larger margins and adjusting row label font size
```

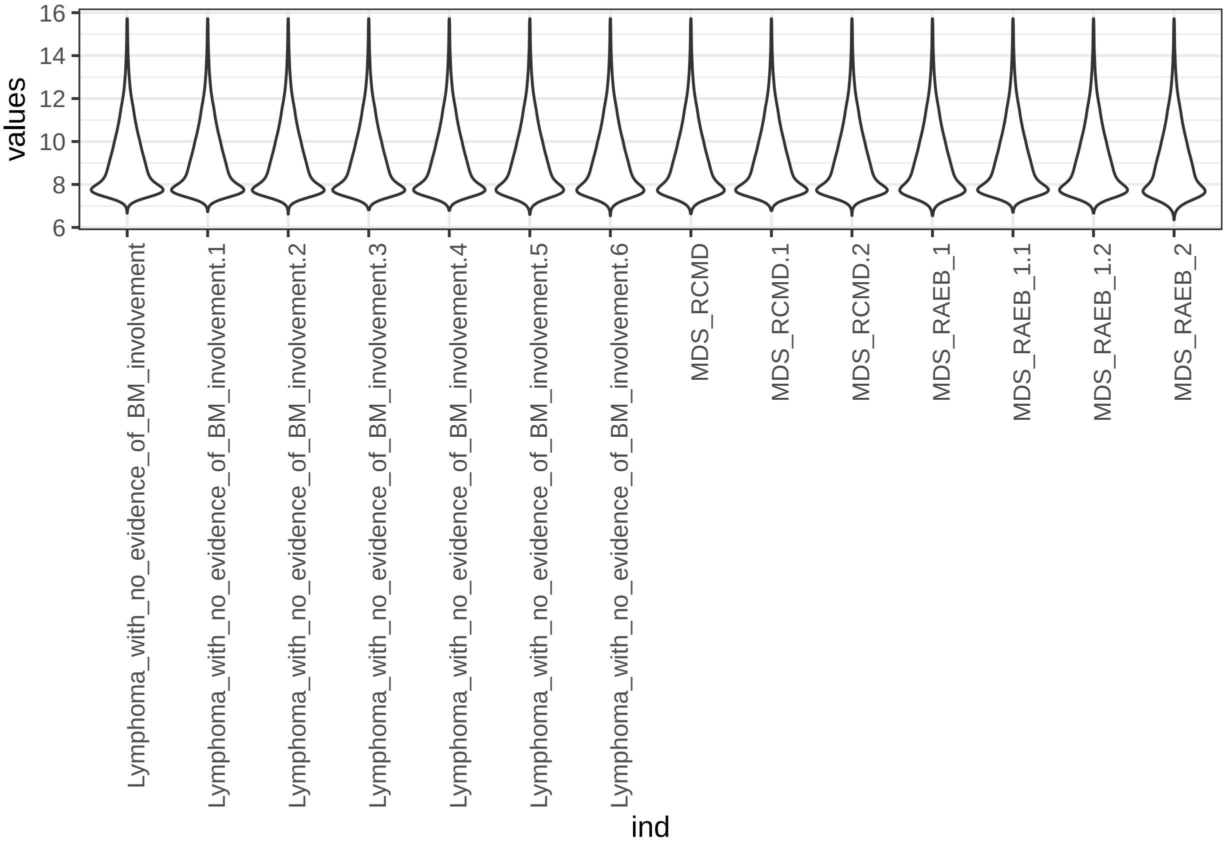
```
pheatmap(h)
```



```
#output plot to file
#dev.off()
```

```
# To make a boxplot of the data:

# install.packages('devtools')
library(devtools)
# devtools::install_github("sinhrks/ggfortify")
library(ggfortify)
```

```
#pdf('box_dataset.pdf', height = 8, width = 10)


gt <- t(xt) # taking xt from the heatmap and transposing it
colnames(gt)<- es$condition # now giving it labels from condition
ggplot(stack(data.frame(gt)), aes(x = ind, y = values)) +
  geom_violin() + theme_bw() + theme(axis.text.x = element_text(angle=90, hjust=1))
```


values
16
14
12
10
8
6
Lymphoma_with_no_evidence_of_BM_involvement
Lymphoma_with_no_evidence_of_BM_involvement.1
Lymphoma_with_no_evidence_of_BM_involvement.2
Lymphoma_with_no_evidence_of_BM_involvement.3
Lymphoma_with_no_evidence_of_BM_involvement.4
Lymphoma_with_no_evidence_of_BM_involvement.5
Lymphoma_with_no_evidence_of_BM_involvement.6
MDS_RCMD
MDS_RCMD.1
MDS_RCMD.2
MDS_RAEB_1
MDS_RAEB_1.1
MDS_RAEB_1.2
MDS_RAEB_2
ind

```
#dev.off()
```