GSE107968

Natalia

12 May 2019

```
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)
```

```
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#2//


#Gene expression profiles of CD34+ cells from patients with
#myelodysplastic syndrome CAA or AML:

es <- getGEO("GSE107968", AnnotGPL = TRUE, parseCharacteristics = FALSE)[[1]]
```

```
## Found 1 file(s)
```

```
## GSE107968_series_matrix.txt.gz
```

```
## Parsed with column specification:
## cols(
##   ID_REF = col_character(),
##   GSM2884491 = col_double(),
##   GSM2884492 = col_double(),
##   GSM2884493 = col_double(),
##   GSM2884494 = col_double(),
##   GSM2884495 = col_double(),
##   GSM2884496 = col_double(),
##   GSM2884497 = col_double(),
##   GSM2884498 = col_double(),
##   GSM2884499 = col_double()
## )
```

```
## File stored at:
```

```
## C:\Users\Public\Documents\iSkysoft\CreatorTemp\Rtmp4ILb51/GPL570.annot.gz
```

```
## Warning: 62 parsing failures.
##   row             col           expected    actual         file
```

```
## 54614 Platform_SPOTID 1/0/T/F/TRUE/FALSE --Control literal data
## 54615 Platform_SPOTID 1/0/T/F/TRUE/FALSE --Control literal data
## 54616 Platform_SPOTID 1/0/T/F/TRUE/FALSE --Control literal data
## 54617 Platform_SPOTID 1/0/T/F/TRUE/FALSE --Control literal data
## 54618 Platform_SPOTID 1/0/T/F/TRUE/FALSE --Control literal data
## ..... ............... .................. ......... ............
## See problems(...) for more details.
```

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

```
## Formal class 'MIAME' [package "Biobase"] with 13 slots
##   ..@ name             : chr "ai,ping,jiang"
##   ..@ lab              : chr ""
##   ..@ contact          : chr "aiping_jiang@shbio.com"
##   ..@ title            : chr "Gene expression profiles of CD34+ cells from patients with myelodysplas
##   ..@ abstract         : chr "To identifying candidate genes which may assist in furthering our knowl
##   ..@ url              : chr "https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE107968"
##   ..@ pubMedIds        : chr ""
##   ..@ samples          : list()
##   ..@ hybridizations   : list()
##   ..@ normControls     : list()
##   ..@ preprocessing    : list()
##   ..@ other            :List of 23
##   .. ..$ contact_address        : chr "No.151, Libing Rd., Zhangjiang Hi-tech Park, Pudong"
##   .. ..$ contact_city           : chr "Shanghai"
##   .. ..$ contact_country        : chr "China"
##   .. ..$ contact_email          : chr "aiping_jiang@shbio.com"
##   .. ..$ contact_institute      : chr "Shanghai Biotechnology Corporation"
##   .. ..$ contact_name           : chr "ai,ping,jiang"
##   .. ..$ contact_phone          : chr "17621760133"
##   .. ..$ contact_zip/postal_code: chr "201203"
##   .. ..$ contributor            : chr "Xiaofei,,Qi\nZixing,,Chen"
##   .. ..$ geo_accession          : chr "GSE107968"
##   .. ..$ last_update_date       : chr "Mar 25 2019"
##   .. ..$ overall_design         : chr "CD34+ cell samples isolated for microarray analysis from eight
##   .. ..$ platform_id            : chr "GPL570"
##   .. ..$ platform_taxid         : chr "9606"
##   .. ..$ relation               : chr "BioProject: https://www.ncbi.nlm.nih.gov/bioproject/PRJNA42205
##   .. ..$ sample_id              : chr "GSM2884491 GSM2884492 GSM2884493 GSM2884494 GSM2884495 GSM2884
##   .. ..$ sample_taxid           : chr "9606"
##   .. ..$ status                 : chr "Public on Dec 13 2017"
##   .. ..$ submission_date        : chr "Dec 12 2017"
##   .. ..$ summary                : chr "To identifying candidate genes which may assist in furthering
##   .. ..$ supplementary_file     : chr "ftp://ftp.ncbi.nlm.nih.gov/geo/series/GSE107nnn/GSE107968/supp
##   .. ..$ title                  : chr "Gene expression profiles of CD34+ cells from patients with myo
##   .. ..$ 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
```

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

```
## 'data.frame':    9 obs. of  38 variables:
##  $ title                  : Factor w/ 9 levels "AML1","AML2",..: 1 2 6 7 4 5 8 3 9
```

```
##  $ geo_accession          : chr  "GSM2884491" "GSM2884492" "GSM2884493" "GSM2884494" ...
##  $ status                 : Factor w/ 1 level "Public on Dec 13 2017": 1 1 1 1 1 1 1 1 1 1
##  $ submission_date        : Factor w/ 1 level "Dec 12 2017": 1 1 1 1 1 1 1 1 1 1
##  $ last_update_date       : Factor w/ 1 level "Jan 23 2018": 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/ 6 levels "AML sample","CAA sample",..: 1 1 4 4 3 3 5 2 6
##  $ organism_ch1           : Factor w/ 1 level "Homo sapiens": 1 1 1 1 1 1 1 1 1 1
##  $ characteristics_ch1    : Factor w/ 2 levels "subject status: myelodysplastic syndrome (MDS) patie
##  $ characteristics_ch1.1  : Factor w/ 6 levels "mds subtype: acute myeloid leukemia (AML)",..: 1 1 5
##  $ characteristics_ch1.2  : Factor w/ 2 levels "cell type: CD34+ cell",..: 2 2 2 2 2 2 2 2 2 1
##  $ characteristics_ch1.3  : Factor w/ 2 levels "","cell type: CD34+ cell": 2 2 2 2 2 2 2 2 2 1
##  $ treatment_protocol_ch1 : Factor w/ 1 level "Heparinized bone marrow samples were obtained by aspi
##  $ growth_protocol_ch1    : Factor w/ 1 level "CD34+ cell samples isolated for microarray analysis fi
##  $ 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 "Trizol extraction of total RNA was performed accordin
##  $ 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 according to the stan
##  $ taxid_ch1              : Factor w/ 1 level "9606": 1 1 1 1 1 1 1 1 1 1
##  $ hyb_protocol           : Factor w/ 1 level "Following fragmentation, 10 ug of cRNA were hybridize
##  $ scan_protocol          : Factor w/ 1 level "GeneChips were scanned using the Hewlett-Packard Gene
##  $ data_processing        : Factor w/ 1 level "The data were analyzed with Microarray Suite version 5
##  $ platform_id            : Factor w/ 1 level "GPL570": 1 1 1 1 1 1 1 1 1 1
##  $ contact_name           : Factor w/ 1 level "ai,ping,jiang": 1 1 1 1 1 1 1 1 1 1
##  $ contact_email          : Factor w/ 1 level "aiping_jiang@shbio.com": 1 1 1 1 1 1 1 1 1 1
##  $ contact_phone          : Factor w/ 1 level "17621760133": 1 1 1 1 1 1 1 1 1 1
##  $ contact_institute      : Factor w/ 1 level "Shanghai Biotechnology Corporation": 1 1 1 1 1 1 1 1
##  $ contact_address        : Factor w/ 1 level "No.151, Libing Rd., Zhangjiang Hi-tech Park, Pudong":
##  $ contact_city           : Factor w/ 1 level "Shanghai": 1 1 1 1 1 1 1 1 1 1
##  $ contact_zip/postal_code: Factor w/ 1 level "201203": 1 1 1 1 1 1 1 1 1 1
##  $ contact_country        : Factor w/ 1 level "China": 1 1 1 1 1 1 1 1 1 1
##  $ supplementary_file     : Factor w/ 9 levels "ftp://ftp.ncbi.nlm.nih.gov/geo/samples/GSM2884nnn/GSM
##  $ data_row_count         : Factor w/ 1 level "54675": 1 1 1 1 1 1 1 1 1 1
##  $ cell type:ch1          : chr  "CD34+ cell" "CD34+ cell" "CD34+ cell" "CD34+ cell" ...
##  $ mds subtype:ch1        : chr  "acute myeloid leukemia (AML)" "acute myeloid leukemia (AML)" "refra
##  $ subject status:ch1     : chr  "myelodysplastic syndrome (MDS) patient" "myelodysplastic syndrome
##  $ tissue:ch1             : chr  "bone marrow" "bone marrow" "bone marrow" "bone marrow" ...
```

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

```
##                   ID
## 1007_s_at 1007_s_at
## 1053_at     1053_at
## 117_at       117_at
## 121_at       121_at
## 1255_g_at 1255_g_at
## 1294_at     1294_at
##                                                          Gene title
## 1007_s_at microRNA 4640///discoidin domain receptor tyrosine kinase 1
## 1053_at                                 replication factor C subunit 2
## 117_at                   heat shock protein family A (Hsp70) member 6
## 121_at                                                 paired box 8
## 1255_g_at                              guanylate cyclase activator 1A
## 1294_at   microRNA 5193///ubiquitin like modifier activating enzyme 7
##              Gene symbol          Gene ID UniGene title UniGene symbol
```

```
## 1007_s_at MIR4640///DDR1  100616237///780
## 1053_at             RFC2             5982
## 117_at             HSPA6             3310
## 121_at              PAX8             7849
## 1255_g_at         GUCA1A             2978
## 1294_at   MIR5193///UBA7 100847079///7318
##           UniGene ID
## 1007_s_at
## 1053_at
## 117_at
## 121_at
## 1255_g_at
## 1294_at
##                                                                                                  Nucleotide Title
## 1007_s_at                                                Human receptor tyrosine kinase DDR gene, complete cds
## 1053_at                                Human replication factor C, 40-kDa subunit (A1) mRNA, complete cds
## 117_at                                                        Human heat-shock protein HSP70B' gene
## 121_at                                                                                 H.sapiens Pax8 mRNA
## 1255_g_at Homo sapiens guanylate cyclase activating protein (GCAP) gene exons 1-4, complete cds
## 1294_at          Homo sapiens ubiquitin-activating enzyme E1 related protein mRNA, complete cds
##               GI GenBank Accession Platform_CLONEID Platform_ORF
## 1007_s_at 1753221            U48705               NA           NA
## 1053_at   1590810            M87338               NA           NA
## 117_at      35221            X51757               NA           NA
## 121_at      38425            X69699               NA           NA
## 1255_g_at  623404            L36861               NA           NA
## 1294_at    520832            L13852               NA           NA
##           Platform_SPOTID Chromosome location
## 1007_s_at              NA              6p21.3
## 1053_at                NA             7q11.23
## 117_at                 NA                1q23
## 121_at                 NA                2q13
## 1255_g_at              NA              6p21.1
## 1294_at                NA                3p21
##
## 1007_s_at                          Chromosome 6, NC_000006.12 (30890883..30890972)///Chromosome 6, NC
## 1053_at                                                                  Chromosome 7, NC_000007.14 (
## 117_at                                                                             Chromosome 1, NC_0
## 121_at                                                                Chromosome 2, NC_000002.12 (11
## 1255_g_at                                                                            Chromosome 6, NC
## 1294_at   Chromosome 3, NC_000003.12 (49806137..49806245, complement)///Chromosome 3, NC_000003.12 (
##
## 1007_s_at
## 1053_at
## 117_at
## 121_at    DNA binding///DNA binding///RNA polymerase II core promoter proximal region sequence-speci
## 1255_g_at
## 1294_at
##
## 1007_s_at
## 1053_at
## 117_at
## 121_at    anatomical structure morphogenesis///branching involved in ureteric bud morphogenesis///cel
## 1255_g_at
```

```
## 1294_at
##
## 1007_s_at basolateral plasma membrane///extracellular exosome///extracellular space///integral compor
## 1053_at                                                                                  Ctf18 RFC-li
## 117_at                                          colocalizes_with COP9 signalosome///blood micropartic
## 121_at
## 1255_g_at                                                                          photoreceptor di
## 1294_at
##
## 1007_s_at                          GO:0005524///GO:0005518///GO:0005518///GO:0046872///GO:0005515//
## 1053_at                         GO:0005524///contributes_to GO:0003689///GO:0019899///GO:0005515///c
## 117_at                                            GO:0005524///GO:0042623///GO:0019899///GO:0031072//
## 121_at    GO:0003677///GO:0003677///GO:0000978///GO:0000979///GO:0005515///GO:0004996///GO:0003700//
## 1255_g_at                                                                          GO:0005509//
## 1294_at                                           GO:0005524///GO:0019782///GO:0005515///GO:0004839//
##
## 1007_s_at
## 1053_at
## 117_at
## 121_at    GO:0009653///GO:0001658///GO:0071371///GO:0007417///GO:0042472///GO:0001822///GO:0003337//
## 1255_g_at
## 1294_at
##                                                                                        GO:Component
## 1007_s_at                  GO:0016323///GO:0070062///GO:0005615///GO:0005887///GO:0005886///GO:00432
## 1053_at                                                                  GO:0031390///GO:0005663///GO:00056
## 117_at    colocalizes_with GO:0008180///GO:0072562///GO:0005814///GO:0005737///GO:0005829///GO:00700
## 121_at                                                                   GO:0005654///GO:0005654///GO:00056
## 1255_g_at                                                                GO:0097381///GO:0001917///GO:00058
## 1294_at                                                 GO:0005829///GO:0005829///GO:0005654///GO:00056
```

```
es$`subject status:ch1`
```

```
## [1] "myelodysplastic syndrome (MDS) patient"
## [2] "myelodysplastic syndrome (MDS) patient"
## [3] "myelodysplastic syndrome (MDS) patient"
## [4] "myelodysplastic syndrome (MDS) patient"
## [5] "myelodysplastic syndrome (MDS) patient"
## [6] "myelodysplastic syndrome (MDS) patient"
## [7] "myelodysplastic syndrome (MDS) patient"
## [8] "myelodysplastic syndrome (MDS) patient"
## [9] "normal; healthy"
```

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

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

```
## [1] "myelodysplastic syndrome (MDS) patient"
## [2] "myelodysplastic syndrome (MDS) patient"
## [3] "myelodysplastic syndrome (MDS) patient"
## [4] "myelodysplastic syndrome (MDS) patient"
## [5] "myelodysplastic syndrome (MDS) patient"
## [6] "myelodysplastic syndrome (MDS) patient"
## [7] "myelodysplastic syndrome (MDS) patient"
## [8] "myelodysplastic syndrome (MDS) patient"
## [9] "normal; healthy"
```

```
#Remove "white spaces" and change with "_":

es$condition[1:8] <- gsub("(MDS)", "MDS", "myelodysplastic syndrome MDS patient")
es$condition[1:8] <- gsub(" ", "_", "myelodysplastic syndrome MDS patient")

es$condition[9] <- gsub("; ", "_", "normal; healthy")
es$condition
```

```
## [1] "myelodysplastic_syndrome_MDS_patient"
## [2] "myelodysplastic_syndrome_MDS_patient"
## [3] "myelodysplastic_syndrome_MDS_patient"
## [4] "myelodysplastic_syndrome_MDS_patient"
## [5] "myelodysplastic_syndrome_MDS_patient"
## [6] "myelodysplastic_syndrome_MDS_patient"
## [7] "myelodysplastic_syndrome_MDS_patient"
## [8] "myelodysplastic_syndrome_MDS_patient"
## [9] "normal_healthy"
```

```
#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))
```

```
##    GSM2884491       GSM2884492        GSM2884493        GSM2884494
##  Min.   : 1.881   Min.   : 0.4459   Min.   : 0.1268   Min.   : 1.905
##  1st Qu.: 6.846   1st Qu.: 7.0005   1st Qu.: 6.6513   1st Qu.: 6.995
##  Median : 8.423   Median : 8.5459   Median : 8.2325   Median : 8.553
##  Mean   : 8.397   Mean   : 8.4898   Mean   : 8.2723   Mean   : 8.500
##  3rd Qu.: 9.837   3rd Qu.: 9.7892   3rd Qu.: 9.9579   3rd Qu.: 9.984
##  Max.   :16.026   Max.   :16.1748   Max.   :15.5379   Max.   :16.307
##    GSM2884495       GSM2884496        GSM2884497         GSM2884498
##  Min.   :-2.041   Min.   : 1.460   Min.   : 0.06263   Min.   : 1.646
##  1st Qu.: 6.730   1st Qu.: 7.109   1st Qu.: 6.05711   1st Qu.: 6.992
##  Median : 8.375   Median : 8.574   Median : 7.76907   Median : 8.540
##  Mean   : 8.329   Mean   : 8.508   Mean   : 7.93503   Mean   : 8.441
##  3rd Qu.:10.002   3rd Qu.: 9.844   3rd Qu.: 9.98225   3rd Qu.: 9.736
##  Max.   :16.394   Max.   :16.746   Max.   :15.51544   Max.   :17.063
##    GSM2884499
##  Min.   :-0.2341
```

```
##  1st Qu.: 6.3396
##  Median : 8.0858
##  Mean   : 8.1349
##  3rd Qu.:10.0617
##  Max.   :15.5244
```

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

```
## Warning in is.data.frame(object): NaNs produced
```

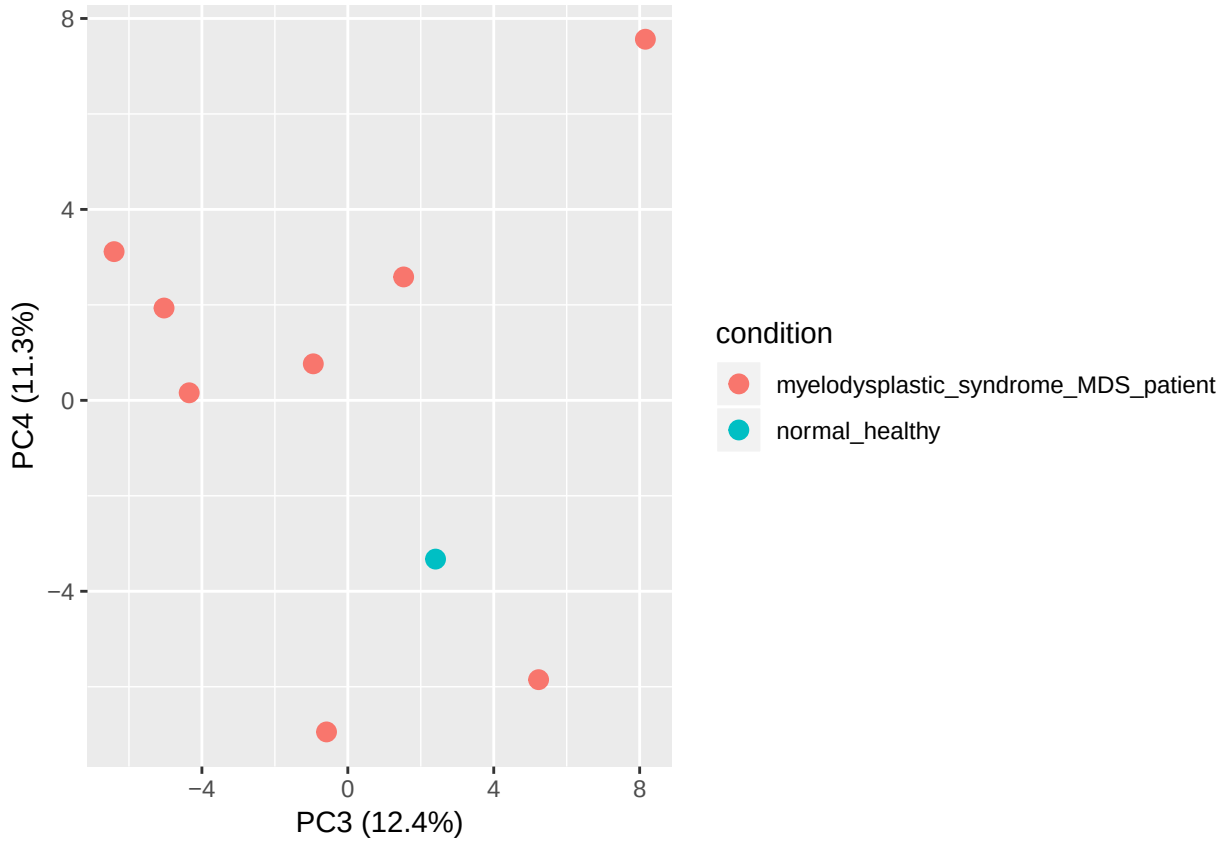
```
summary(exprs(es.qnorm))
```

```
##    GSM2884491       GSM2884492       GSM2884493       GSM2884494
##  Min.   :0.6852   Min.   :0.6852   Min.   :0.6852   Min.   :0.6852
##  1st Qu.:2.9522   1st Qu.:2.9522   1st Qu.:2.9522   1st Qu.:2.9522
##  Median :3.2235   Median :3.2235   Median :3.2235   Median :3.2235
##  Mean   :3.1732   Mean   :3.1732   Mean   :3.1732   Mean   :3.1732
##  3rd Qu.:3.4476   3rd Qu.:3.4476   3rd Qu.:3.4476   3rd Qu.:3.4476
##  Max.   :4.0989   Max.   :4.0989   Max.   :4.0989   Max.   :4.0989
##
##    GSM2884495       GSM2884496       GSM2884497       GSM2884498
##  Min.   :0.6852   Min.   :0.6852   Min.   :0.6852   Min.   :0.6852
##  1st Qu.:2.9522   1st Qu.:2.9522   1st Qu.:2.9522   1st Qu.:2.9522
##  Median :3.2235   Median :3.2235   Median :3.2235   Median :3.2235
##  Mean   :3.1732   Mean   :3.1732   Mean   :3.1732   Mean   :3.1732
##  3rd Qu.:3.4476   3rd Qu.:3.4476   3rd Qu.:3.4476   3rd Qu.:3.4476
##  Max.   :4.0989   Max.   :4.0989   Max.   :4.0989   Max.   :4.0989
##  NA's   :1
##    GSM2884499
##  Min.   :0.6852
##  1st Qu.:2.9522
##  Median :3.2235
##  Mean   :3.1732
##  3rd Qu.:3.4476
##  Max.   :4.0989
##
```

```
#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), ]
```

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

pcaPlot(es.qnorm.top12K,3,4) + aes(color = condition)
```

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




```
#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
```

```
##            conditionmyelodysplastic_syndrome_MDS_patient
## GSM2884491                                             1
## GSM2884492                                             1
## GSM2884493                                             1
## GSM2884494                                             1
## GSM2884495                                             1
## GSM2884496                                             1
## GSM2884497                                             1
## GSM2884498                                             1
## GSM2884499                                             0
##            conditionnormal_healthy
## GSM2884491                       0
## GSM2884492                       0
## GSM2884493                       0
## GSM2884494                       0
## GSM2884495                       0
## GSM2884496                       0
## GSM2884497                       0
## GSM2884498                       0
## GSM2884499                       1
## attr(,"assign")
```

```
## [1] 1 1
## attr(,"contrasts")
## attr(,"contrasts")$condition
## [1] "contr.treatment"
```

```
#we have 2 conditions:
im <- data.frame(es.design)
colnames(im) <- c("conditionmyelodysplastic_syndrome_MDS_patient",
                  "conditionnormal_healthy")

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

fit2 <- contrasts.fit(fit, makeContrasts(conditionmyelodysplastic_syndrome_MDS_patient, conditionnormal_

fit2 <- eBayes(fit2)

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

```
##        entrez symbol conditionmyelodysplastic_syndrome_MDS_patient
## ND4       ND4   4538                                        4.082279
## RPS4X   RPS4X   6191                                        4.062212
## EEF1A1 EEF1A1   1915                                        4.065197
## RPL41   RPL41   6171                                        4.057263
## RPL39   RPL39   6170                                        4.055973
## RPS10   RPS10   6204                                        4.048377
##        conditionnormal_healthy  AveExpr        F      P.Value    adj.P.Val
## ND4                   4.042815 4.077894 30559.67 4.400821e-20 2.071011e-17
## RPS4X                 4.058776 4.061831 30500.62 4.444667e-20 2.071011e-17
## EEF1A1                4.039510 4.062343 30303.53 4.594862e-20 2.071011e-17
## RPL41                 4.046776 4.056098 30296.44 4.600377e-20 2.071011e-17
## RPL39                 4.057178 4.056107 30165.86 4.703371e-20 2.071011e-17
## RPS10                 4.055852 4.049207 30150.65 4.715544e-20 2.071011e-17
```

```
# 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.
```

```
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 == "ND4"]
```

```
##     rn entrez symbol conditionmyelodysplastic_syndrome_MDS_patient
## 1: ND4    ND4   4538                                        4.082279
##    conditionnormal_healthy  AveExpr        F      P.Value    adj.P.Val
## 1:                4.042815 4.077894 30559.67 4.400821e-20 2.071011e-17
```

```
#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)
```

```
##          ND4        RPS4X       EEF1A1        RPL41        RPL39
## 4.400821e-20 4.444667e-20 4.594862e-20 4.600377e-20 4.703371e-20
##        RPS10         COX1        RPS16         RPS7       RPL23A
## 4.715544e-20 4.920691e-20 4.930346e-20 4.958315e-20 4.994745e-20
##       RPL37A        HUWE1       RPL10A        RPS29        RPL30
## 5.004110e-20 5.170684e-20 5.277888e-20 5.606331e-20 5.650615e-20
##          B2M         ATP6         RPL7          UBB        RPLP1
## 5.687095e-20 5.716898e-20 5.854754e-20 6.002305e-20 6.589212e-20
```

```
# 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"
```

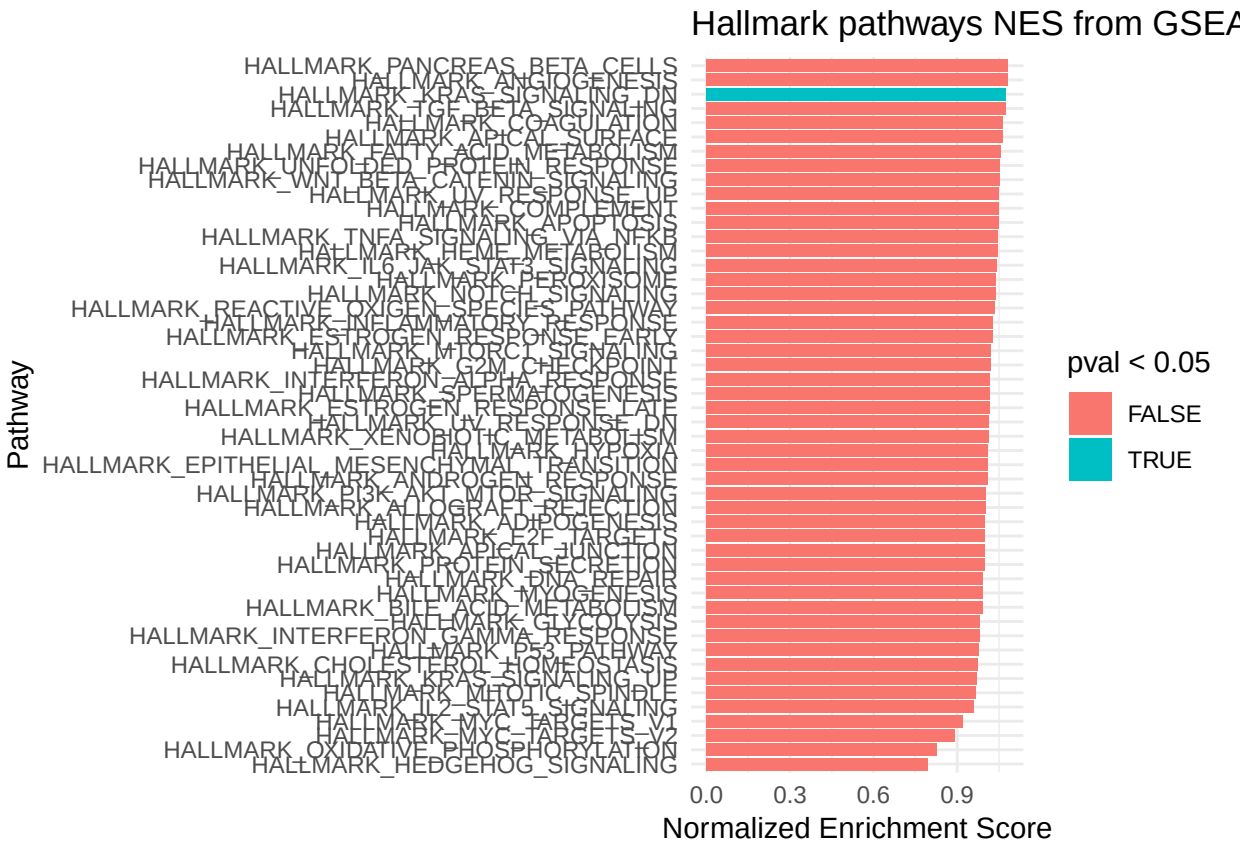
```
# filter the list to include only hallmark pathways:

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 = 10)

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_ADIPOGENESIS,HALLMARK_ALLOGRAFT_REJECTION,
#HALLMARK_ANDROGEN_RESPONSE etc.
# pathway are activated!
# Let's look at all pathways involving the following genes
#that were mentioned in the initial work (paper):
# NB! no paper published was found with this results.

# We are going to search the entire pathway list for any pathway
#that contains these genes that were discussing in the paper on
#the same subject (GSE61853):
#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, EIF3A, RPL31.

#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', 'CLO23', '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] 13172 16458
## [1] 7302
##   [1]   251   258   300   602   605   606   643   675   683   687   809
##  [12]   872   937  1051  1303  1394  1424  1441  1472  1490  1492  1494
##  [23]  1532  1597  1656  1674  1819  2825  2870  3117  3163  3906  3909
##  [34]  3930  7258  7375  8145  8147  8151  8164  8236  8242  8252  8293
##  [45]  8341  8475  8478  8488  8492  8498  8506  8520  8531  8596  8600
##  [56]  8644  8690  8746  8777  8792  8803  8853  8878  8890  9036  9054
##  [67]  9077  9205  9207  9235  9282  9284  9288  9426  9437  9448  9466
##  [78]  9544  9548  9554  9563  9570  9579  9583  9587  9718  9772  9773
##  [89]  9807  9815  9817  9819  9831  9834  9843  9884  9926  9938  9941
## [100]  9947 10012 10044 10056 10060 10066 10144 10229 10233 10254 10292
## [111] 10365 10472 10475 10495 10554 10558 10566 10582 10620 10638 10646
## [122] 10657 10719 10752 10754 10768 10818 10851 10863 10966 11004 11052
## [133] 11070 11092 11094 11139 11153 11168 11177 11187 11209 11212 11221
## [144] 11234 11240 11244 11250 11276 11280 11296 11318 11371 11375 11378
## [155] 11388 11438 11443 11450 11541 11569 11626 11628 11631 11634 11645
## [166] 11647 11652 11654 11680 11697 11754 11764 11812 11816 11834 11901
## [177] 11903 11905 11941 11977 11981 12030 12031 12032 12116 12131 12147
## [188] 12160 12162 12164 12166 12177 12186 12188 12190 12193 12211 12223
```

```
## [199] 12245 12247 12268 12315 12376 12434 12446 12463 12498 12561 12566
## [210] 12625 12652 12679 12683 12686 12689 12724 12742 12784 12786 12802
## [221] 12832 12970 13085 13133 13134 13186 13235 13369 13370 13409 13457
## [232] 13467 13480 13492 13847 13857 13863 14050 14114 14119 14192 14327
## [243] 14355 14362 14368 14474 14495 14512 14571 14583 14622 14629 14692
## [254] 14699 14702 14708 14727 14789 14814 15076 15240 15771 15777 15958
## [265] 16096 16117 16119 16648 16724 16940 17038 17260 17462 17577 17580
## [276] 17597 17619 17706
## [1] 16715 16737 17545 17569
## integer(0)
## [1] 7986
##  [1]  2561  2927  3614  7814  8399  8643  8662 11069 13088 17565
##  [1]  1650  6803  6823  6826  7203 13088 13306 14181 15097 15098 16677
## [12] 17059 17207 17380 17475 17552 17565 17724
## integer(0)
## integer(0)
##   [1]   282   576  1200  1619  1620  1835  1873  2252  2277  2278  2326
##  [12]  2389  2390  2394  2396  2398  2495  2763  2795  2805  2825  2853
##  [23]  2870  3009  3010  3011  3012  3022  3023  3117  3142  3163  3198
##  [34]  3207  3208  3372  3374  3389  3396  3404  3405  3459  3460  3530
##  [45]  3737  3738  3745  3873  3874  3905  3906  3907  3909  3912  3913
##  [56]  3927  3928  3930  3972  3998  4013  4019  4021  4089  4090  4114
##  [67]  4179  4193  4196  4208  4209  4215  4305  4354  4356  4358  4655
##  [78]  4656  4658  4676  4699  4705  4772  4778  4779  4780  4782  4811
##  [89]  4841  4927  4932  4933  4966  5120  5235  5337  5495  5604  5605
## [100]  5607  5608  5698  5735  5737  5742  5743  5750  5751  5893  5931
## [111]  5955  5960  6019  6088  6122  6126  6209  6236  6326  6448  6449
## [122]  6452  6479  6493  6543  6546  6553  6694  6695  6697  6699  6707
## [133]  6708  6712  6713  6718  6722  6727  6728  6734  6769  6926  6927
## [144]  6982  6993  7031  7037  7039  7050  7121  7160  7244  7245  7258
## [155]  7273  7303  7572  7576  7584  7802  7804  7870  7871  7950  7951
## [166]  8019  8020  8023  8058  8154  8158  8160  8406  8477  8927  9003
## [177]  9527  9540  9553  9900  9945 10404 11220 11223 11228 11237 11303
## [188] 11320 11330 11372 11747 12476 12926 12928 12930 13094 13096 13101
## [199] 13202 13203 13205 13304 13566 13572 13578 13579 13591 13637 13639
## [210] 13646 13648 13724 13734 13744 13761 14071 14298 14446 14554 14584
## [221] 14621 14692 14702 14708 14727 14768 15083 15085 15187 15401 15742
## [232] 16515 16584 16593 16611 16613 16615 16723 16849 17135 17579
## [1] 16715 16737 17545 17569
## integer(0)
## [1]   372   469  2118 13375 15482 17460
##  [1]   290   497  1393  1651  2371  2863  2889  2934  3458  3864  4495
## [12]  4678  4794  4934  5213  5675  5984  6115  6142  6193  6229  6579
## [23]  6744  6801  6802  7040  7265  7268  7368  7370  7514  7551  8006
## [34]  9275  9277 11582 11760 11766 13306 13476 14021 15456 15461 15838
## [45] 15840 15873 15909 15974 16002 16186 16486 16686 16689 17161 17166
## [56] 17264
## integer(0)
##  [1]  1650  6803  6823  6826  7203 13088 13306 14181 15097 15098 16677
## [12] 17059 17207 17380 17475 17552 17565 17724
## integer(0)
## [1] 14561 14663
## integer(0)
## integer(0)
```

```
## integer(0)
## integer(0)
## [1] 14657 15830
## [1] 14549
## integer(0)
## [1]  4016  5306  6308 16333 16334
## integer(0)
## [1] 14549
## integer(0)
## [1] 4823 4928
## integer(0)
## integer(0)
## [1] 2085 2086 2137 2178 3140 3509 3617 7932
## integer(0)
## integer(0)
##  [1]  1650  6803  6823  6826  7203 13088 13306 14181 15097 15098 16677
## [12] 17059 17207 17380 17475 17552 17565 17724
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## integer(0)
##  [1]   285   742  1520  1602  2458  2934  8235  9014  9634  9722 10219
## [12] 11105 12895 13931 16708
## integer(0)
## [1]  5053 14289
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## integer(0)
##  [1] 1464 1489 1898 2051 2085 2086 2137 2144 2178 3140 3509 3617 7932
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## [1] 13577 13686 16060
```

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

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

```r
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)
```

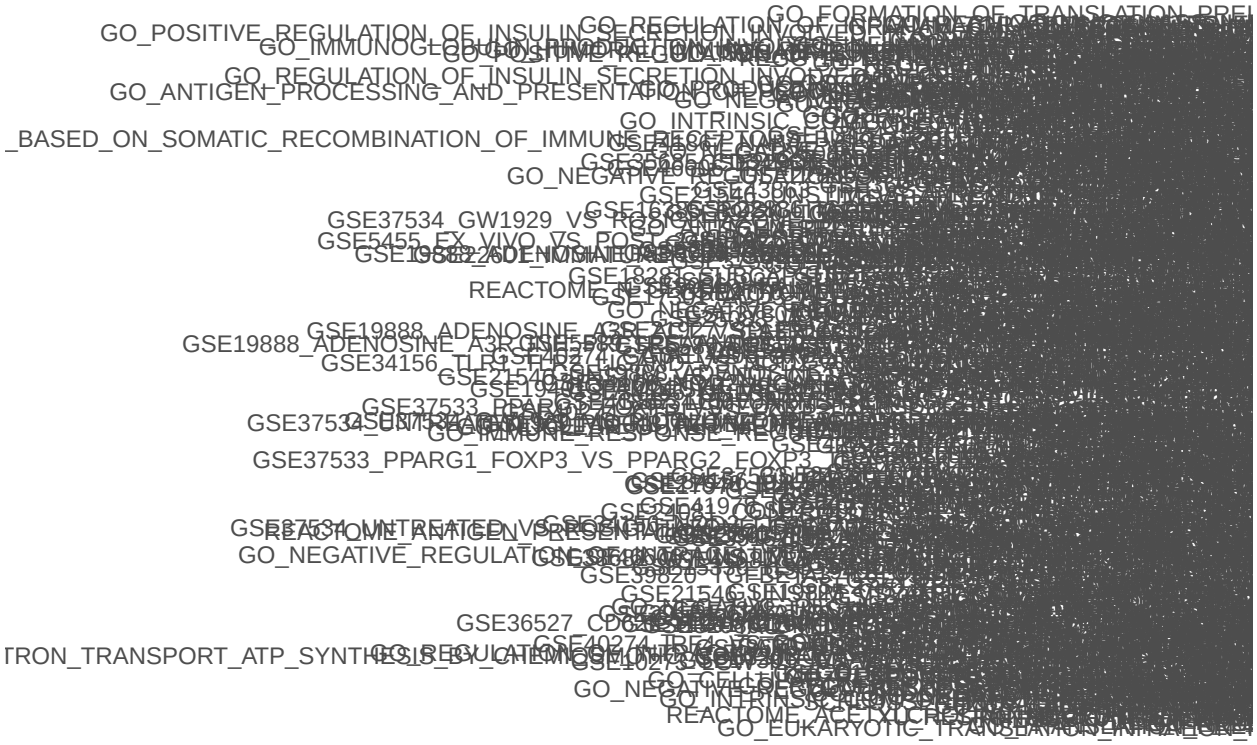
```
## $pathway
## [1] "BAELDE_DIABETIC_NEPHROPATHY_DN"
## [2] "BAKKER_FOXO3_TARGETS_DN"
## [3] "BASSO_CD40_SIGNALING_UP"
## [4] "BAUS_TFF2_TARGETS_DN"
## [5] "BCAT_BILD_ET_AL_UP"
## [6] "BENNETT_SYSTEMIC_LUPUS_ERYTHEMATOSUS"
##
## $pval
## [1] 0.20079920 0.85514486 0.23076923 0.47839506 0.45554446 0.08691309
##
## $padj
## [1] 0.8934790 0.9943545 0.8967701 0.9278521 0.9202918 0.8900129
##
## $ES
## [1] 0.9765394 0.8991007 0.9713721 0.8755697 0.9313908 0.9853872
##
## $NES
## [1] 1.0233827 0.9563561 1.0388982 1.0545692 1.0269060 1.0878654
##
## $nMoreExtreme
## [1] 200 855 230 464 455  86
##
## $size
## [1] 335 110  87   7  27  26
##
## $leadingEdge
## $leadingEdge[[1]]
##  [1] "CHI3L1" "NDN"    "CD200"  "MSH2"   "CALD1"  "AXL"    "TUBB2A"
##  [8] "CITED2" "IFI35"  "WT1"
##
## $leadingEdge[[2]]
##  [1] "IFI35"   "TACSTD2" "CFD"     "MKI67"   "ALKBH7"  "ETV5"    "TNRC18"
##  [8] "ANKZF1"  "HSPB8"   "LSM10"
##
## $leadingEdge[[3]]
## [1] "HLA-DRB4" "HLA-DQB1" "NCF2"     "KCNN4"    "CCR7"     "LGALS1"
## [7] "CCL4"
##
```

```
## $leadingEdge[[4]]
## [1] "BAG2"  "PRDX2" "BEX4"
##
## $leadingEdge[[5]]
## [1] "CAMKMT" "STXBP2" "CAPS"   "FHDC1"  "CALR"
##
## $leadingEdge[[6]]
## [1] "CAMP"  "IFIT3" "IFI35"
```

```
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=30, height = 100 )
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()
```

```
## Warning: Removed 8 rows containing missing values (position_stack).
```



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

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

# Show in a table for all pathways:

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

Show 10 entries Search:

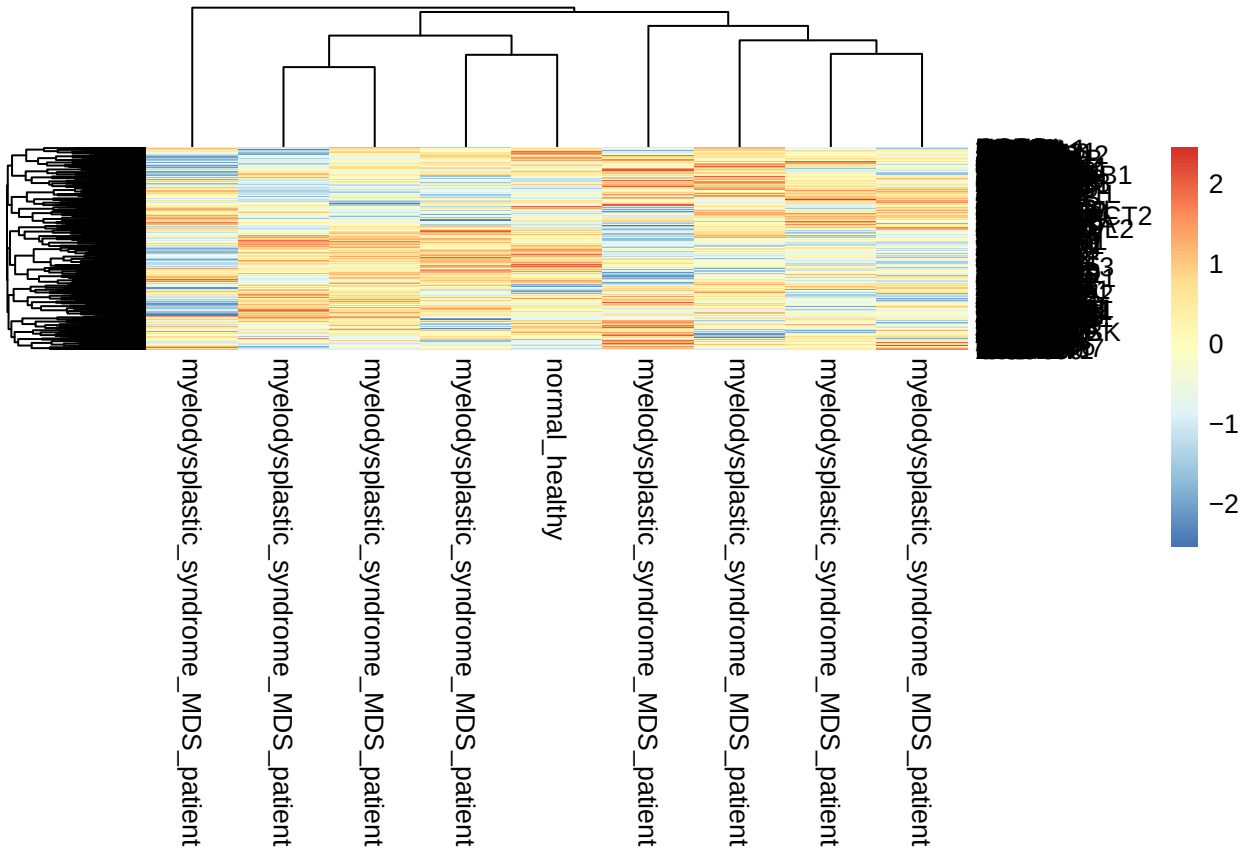
	pathway	pval	padj	NES	size
1	HALLMARK_KRAS_SIGNALING_DN	0.035964035964036	0.638250638250638	1.07568976759648	62
2	HALLMARK_COAGULATION	0.0869130869130869	0.638250638250638	1.06323573701849	78
3	HALLMARK_FATTY_ACID_METABOLISM	0.0539460539460539	0.638250638250638	1.05734780892545	124
4	HALLMARK_UNFOLDED_PROTEIN_RESPONSE	0.110889110889111	0.638250638250638	1.05388265796502	99
5	HALLMARK_UV_RESPONSE_UP	0.109890109890911	0.638250638250638	1.05055637681384	117
6	HALLMARK_COMPLEMENT	0.101898101898102	0.638250638250638	1.0502680396657	134
7	HALLMARK_APOPTOSIS	0.114885114885115	0.638250638250638	1.04892141152634	132
8	HALLMARK_TNFA_SIGNALING_VIA_NFKB	0.0829170829170829	0.638250638250638	1.04728366276719	168
9	HALLMARK_HEME_METABOLISM	0.0789210789210789	0.638250638250638	1.04666681638303	178
10	HALLMARK_TGF_BETA_SIGNALING	0.131868131868132	0.659340659340659	1.07360166952198	44

Showing 1 to 10 of 50 entries Previous 1 2 3 4 5 Next

```
# 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=10, height=10)
```

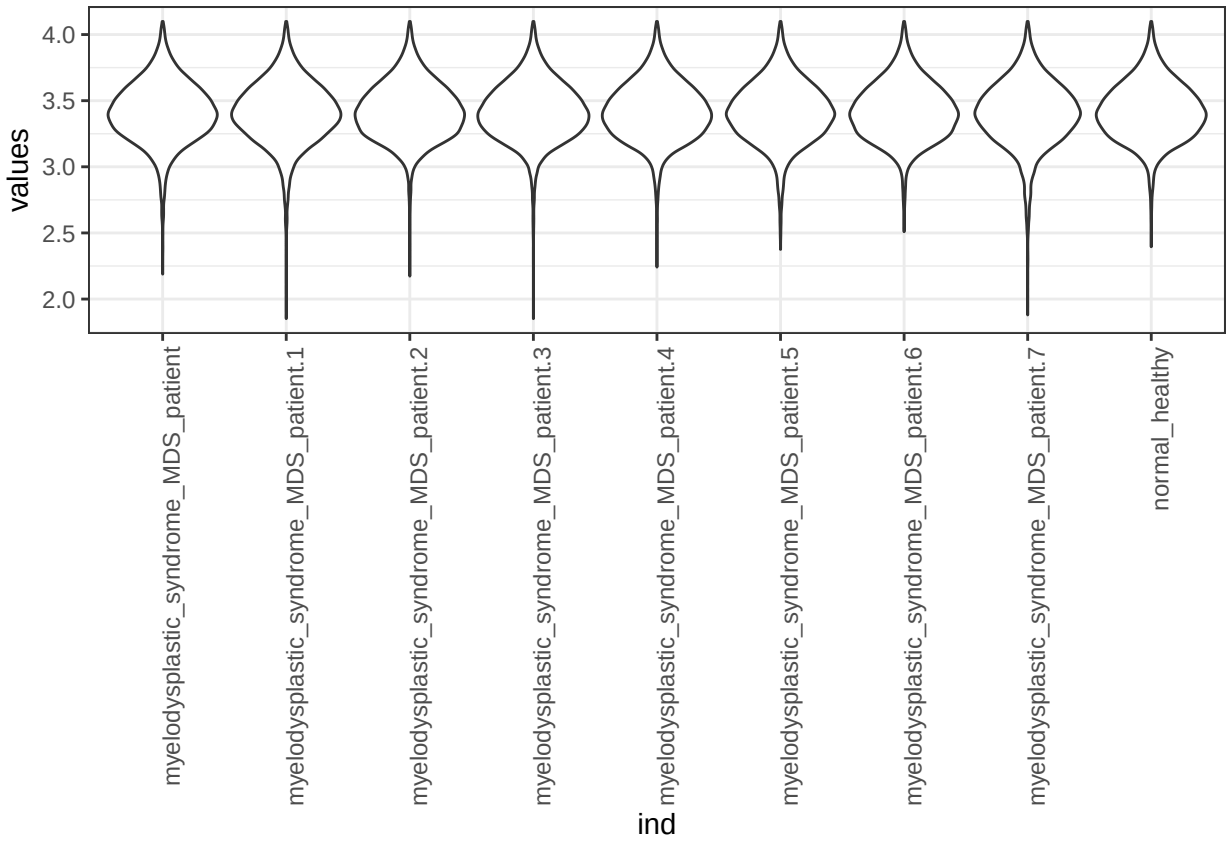
```
#draw heatmap allowing larger margins and adjusting row label font size
pheatmap(h)
```




```
#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 = 5, width = 5)


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
4.0
3.5
3.0
2.5
2.0
myelodysplastic_syndrome_MDS_patient
myelodysplastic_syndrome_MDS_patient.1
myelodysplastic_syndrome_MDS_patient.2
myelodysplastic_syndrome_MDS_patient.3
myelodysplastic_syndrome_MDS_patient.4
myelodysplastic_syndrome_MDS_patient.5
myelodysplastic_syndrome_MDS_patient.6
myelodysplastic_syndrome_MDS_patient.7
normal_healthy
ind

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