data_3_GSE66486

Sedreh

5/10/2019

```
library(dplyr)
```

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

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

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

```
# BiocManager::install('GEOquery')
# 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:dplyr':
##
##     combine, intersect, setdiff, union
```

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

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

```
# library for mouse annotation
library(org.Hs.eg.db)
```

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

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

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

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

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

```
## The following objects are masked from 'package:dplyr':
##
##     first, rename
```

```
## The following object is masked from 'package:base':
##
##     expand.grid
```

```
##
## Attaching package: 'IRanges'
```

```
## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice
```

```
##
## Attaching package: 'AnnotationDbi'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
##
```

```
# for collapseBy and other functions
source("/home/sedreh/Documents/rnaseq/functions.r")
### load the dataset here
res<- getGEO("GSE66486", AnnotGPL = TRUE)[[1]]
```

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

```
## GSE66486_series_matrix.txt.gz
## Parsed with column specification:
## cols(
##   ID_REF = col_character(),
##   GSM1623213 = col_double(),
##   GSM1623214 = col_double(),
##   GSM1623215 = col_double(),
##   GSM1623216 = col_double(),
##   GSM1623217 = col_double(),
##   GSM1623218 = col_double(),
##   GSM1623219 = col_double(),
##   GSM1623220 = col_double(),
##   GSM1623221 = col_double(),
##   GSM1623222 = col_double(),
##   GSM1623223 = col_double(),
##   GSM1623224 = col_double(),
##   GSM1623225 = col_double(),
##   GSM1623226 = col_double(),
##   GSM1623227 = col_double(),
##   GSM1623228 = col_double(),
##   GSM1623229 = col_double(),
##   GSM1623230 = col_double()
## )
```

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

```
## /tmp/Rtmp7Xdjle/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.
```

```
# GEOquery is working, this is a list of files, I can see all the information
# to access individual list I need to use this format res$data@data
# for example, res@experimentData@title will give us details about the experiment
res@experimentData@title
```

```
## [1] ""
```

```
# this is mouse dataset
res@experimentData@abstract
```

```
## [1] ""
```

```
# every GEO data has these internal identifiers: pData is phenotypeData, fData is featureData
str(experimentData(res))
```

```
## Formal class 'MIAME' [package "Biobase"] with 13 slots
##   ..@ name             : chr ""
##   ..@ lab              : chr ""
##   ..@ contact          : chr ""
##   ..@ title            : chr ""
##   ..@ abstract         : chr ""
##   ..@ url              : chr ""
##   ..@ pubMedIds        : chr ""
##   ..@ samples          : list()
##   ..@ hybridizations   : list()
##   ..@ normControls     : list()
##   ..@ preprocessing    : list()
##   ..@ other            : list()
##   ..@ .__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(res))
```

```
## 'data.frame':	18 obs. of  38 variables:
##  $ title                  : Factor w/ 18 levels "Ctrl1 FLU_16h",..: 3 2 1 6 5 4 9 8 7 12 ...
##  $ geo_accession          : chr  "GSM1623213" "GSM1623214" "GSM1623215" "GSM1623216" ...
##  $ status                 : Factor w/ 1 level "Public on Apr 07 2015": 1 1 1 1 1 1 1 1 1 1 ...
##  $ submission_date        : Factor w/ 1 level "Mar 03 2015": 1 1 1 1 1 1 1 1 1 1 ...
##  $ last_update_date       : Factor w/ 1 level "Apr 07 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/ 9 levels "control_PBMC_FLU_16hr",..: 3 2 1 3 2 1 3 2 1 3 ...
##  $ organism_ch1           : Factor w/ 1 level "Homo sapiens": 1 1 1 1 1 1 1 1 1 1 ...
##  $ characteristics_ch1    : Factor w/ 3 levels "subject status: healthy donor",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ characteristics_ch1.1  : Factor w/ 1 level "cell type: PBMC": 1 1 1 1 1 1 1 1 1 1 ...
##  $ characteristics_ch1.2  : Factor w/ 3 levels "infected with: influenza virus A/CA/4/2009 for 16hrs",..: 3 2
1 3 2 1 3 2 1 3 ...
##  $ treatment_protocol_ch1 : Factor w/ 1 level "0.5x10^6 PBMCs were washed twice with PBS and infected with inf
luenza A/California/4/2009 virus at MOI=2 in RPM"| __truncated__: 1 1 1 1 1 1 1 1 1 1 ...
##  $ growth_protocol_ch1    : Factor w/ 1 level "PBMCs were cultured in RPMI 16-40 medium with 10% fetal bovine
serum.": 1 1 1 1 1 1 1 1 1 1 ...
##  $ 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 "RNA was extracted with Qiagen Rneasy mini kit, followed by DNas
e I treatment and clean-up with QIAGEN RNeasy mi"| __truncated__: 1 1 1 1 1 1 1 1 1 1 ...
##  $ 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 targets were prepared from 250 ng of total RN
A, using the Illumina TotalPrep RNA Amplification kit (Ambion).": 1 1 1 1 1 1 1 1 1 1 ...
##  $ 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 1 1 1 1
...
##  $ scan_protocol          : Factor w/ 1 level "Standard Illumina scanning protocol": 1 1 1 1 1 1 1 1 1 1 ...
##  $ description            : Factor w/ 18 levels "SAMPLE 1","SAMPLE 10",..: 1 11 12 13 14 15 16 17 18 2 ...
##  $ data_processing        : Factor w/ 1 level "The data were normalised using quantile normalisation with in
R. PALO (p<0.05 present in at least one) were use"| __truncated__: 1 1 1 1 1 1 1 1 1 1 ...
##  $ platform_id            : Factor w/ 1 level "GPL10558": 1 1 1 1 1 1 1 1 1 1 ...
##  $ contact_name           : Factor w/ 1 level "Scott,,Presnell": 1 1 1 1 1 1 1 1 1 1 ...
##  $ contact_email          : Factor w/ 1 level "SPresnell@benaroyaresearch.org": 1 1 1 1 1 1 1 1 1 1 ...
##  $ contact_department     : Factor w/ 1 level "Systems Immunology": 1 1 1 1 1 1 1 1 1 1 ...
##  $ contact_institute      : Factor w/ 1 level "Benaroya Research Institute": 1 1 1 1 1 1 1 1 1 1 ...
##  $ contact_address        : Factor w/ 1 level "1201 Ninth Ave.": 1 1 1 1 1 1 1 1 1 1 ...
##  $ contact_city           : Factor w/ 1 level "Seattle": 1 1 1 1 1 1 1 1 1 1 ...
##  $ contact_state          : Factor w/ 1 level "WA": 1 1 1 1 1 1 1 1 1 1 ...
##  $ contact_zip/postal_code: Factor w/ 1 level "98101": 1 1 1 1 1 1 1 1 1 1 ...
##  $ contact_country        : Factor w/ 1 level "USA": 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 "20517": 1 1 1 1 1 1 1 1 1 1 ...
##  $ cell type:ch1          : chr  "PBMC" "PBMC" "PBMC" "PBMC" ...
##  $ infected with:ch1      : chr  "none for 16hrs (uninfected control)" "influenza virus A/CA/4/2009 for 8hrs"
"influenza virus A/CA/4/2009 for 16hrs" "none for 16hrs (uninfected control)" ...
##  $ subject status:ch1     : chr  "healthy donor" "healthy donor" "healthy donor" "healthy donor" ...
```

```
head(fData(res))
```

```
##                          ID
## ILMN_1343291 ILMN_1343291
## ILMN_1343295 ILMN_1343295
## ILMN_1651209 ILMN_1651209
## ILMN_1651228 ILMN_1651228
## ILMN_1651229 ILMN_1651229
## ILMN_1651232 ILMN_1651232
##                                                            Gene title
## ILMN_1343291      eukaryotic translation elongation factor 1 alpha 1
## ILMN_1343295                glyceraldehyde-3-phosphate dehydrogenase
## ILMN_1651209                      solute carrier family 35 member E2
## ILMN_1651228                                   ribosomal protein S28
## ILMN_1651229                                             importin 13
## ILMN_1651232 family with sequence similarity 86, member A pseudogene
##              Gene symbol Gene ID UniGene title UniGene symbol UniGene ID
## ILMN_1343291      EEF1A1    1915
## ILMN_1343295       GAPDH    2597
## ILMN_1651209     SLC35E2    9906
## ILMN_1651228       RPS28    6234
## ILMN_1651229       IPO13    9670
## ILMN_1651232     FAM86FP  653113
##                                                                                             Nucleotide Title
## ILMN_1343291                 Homo sapiens eukaryotic translation elongation factor 1 alpha 1 (EEF1A1), mRNA
## ILMN_1343295      Homo sapiens glyceraldehyde-3-phosphate dehydrogenase (GAPDH), transcript variant 1, mRNA
## ILMN_1651209          Homo sapiens solute carrier family 35 member E2 (SLC35E2), transcript variant 1, mRNA
## ILMN_1651228                                              Homo sapiens ribosomal protein S28 (RPS28), mRNA
## ILMN_1651229                                                        Homo sapiens importin 13 (IPO13), mRNA
## ILMN_1651232 Homo sapiens family with sequence similarity 86 member F, pseudogene (FAM86FP), non-coding RNA
##                     GI GenBank Accession Platform_CLONEID Platform_ORF
## ILMN_1343291  83367078         NM_001402               NA           NA
## ILMN_1343295 576583510         NM_002046               NA           NA
## ILMN_1651209 315139027         NM_182838               NA           NA
## ILMN_1651228  71565158         NM_001031               NA           NA
## ILMN_1651229 224831256         NM_014652               NA           NA
## ILMN_1651232 209954785         NR_024254               NA           NA
##              Platform_SPOTID Chromosome location
## ILMN_1343291              NA              6q14.1
## ILMN_1343295              NA               12p13
## ILMN_1651209              NA             1p36.33
## ILMN_1651228              NA             19p13.2
## ILMN_1651229              NA              1p34.1
## ILMN_1651232              NA            12p13.31
##                                                    Chromosome annotation
## ILMN_1343291 Chromosome 6, NC_000006.12 (73515750..73521032, complement)
## ILMN_1343295             Chromosome 12, NC_000012.12 (6534405..6538375)
## ILMN_1651209  Chromosome 1, NC_000001.11 (1724838..1745999, complement)
## ILMN_1651228             Chromosome 19, NC_000019.10 (8321500..8322396)
## ILMN_1651229            Chromosome 1, NC_000001.11 (43946806..43968022)
## ILMN_1651232 Chromosome 12, NC_000012.12 (8231049..8242946, complement)
##
GO:Function
## ILMN_1343291
GTP binding///GTPase activity///poly(A) RNA binding///protein binding///protein kinase binding///tRNA binding///t
ranslation elongation factor activity
## ILMN_1343295 NAD binding///NADP binding///glyceraldehyde-3-phosphate dehydrogenase (NAD+) (phosphorylating) ac
tivity///glyceraldehyde-3-phosphate dehydrogenase (NAD+) (phosphorylating) activity///glyceraldehyde-3-phosphate
dehydrogenase (NAD+) (phosphorylating) activity///identical protein binding///microtubule binding///peptidyl-cyst
eine S-nitrosylase activity///protein binding
## ILMN_1651209
## ILMN_1651228
poly(A) RNA binding///protein binding///structural constituent of ribosome
## ILMN_1651229
Ran GTPase binding///nuclear localization sequence binding///protein binding///protein transporter activity
## ILMN_1651232
##
GO:Process
## ILMN_1343291
cellular response to epidermal growth factor stimulus///regulation of chaperone-mediated autophagy///regulation o
f transcription, DNA-templated///transcription, DNA-templated///translational elongation
## ILMN_1343295 canonical glycolysis///cellular response to interferon-gamma///gluconeogenesis///microtubule cyto
skeleton organization///negative regulation of translation///negative regulation of translation///neuron apoptoti
c process///peptidyl-cysteine S-trans-nitrosylation///protein stabilization///regulation of macroautophagy
## ILMN_1651209
## ILMN_1651228   SRP-dependent cotranslational protein targeting to membrane///nuclear-transcribed mRNA cataboli
c process, nonsense-mediated decay///rRNA export from nucleus///rRNA processing///rRNA processing///ribosomal sma
ll subunit biogenesis///ribosome biogenesis///translation///translational initiation///viral transcription
## ILMN_1651229
```

```
protein import into nucleus
## ILMN_1651232
##
GO:Component
## ILMN_1343291                                                                                 cortical actin cy
toskeleton///cytoplasm///cytoplasm///cytoplasm///cytoplasmic side of lysosomal membrane///cytosol///cytosol///euk
aryotic translation elongation factor 1 complex///extracellular exosome///extracellular space///membrane///nucleo
lus///nucleus///ruffle membrane
## ILMN_1343295 GAIT complex///cytoplasm///cytoplasm///cytosol///cytosol///cytosol///extracellular exosome///extr
acellular matrix///intracellular membrane-bounded organelle///intracellular ribonucleoprotein complex///lipid par
ticle///membrane///microtubule cytoskeleton///nuclear membrane///nucleus///nucleus///perinuclear region of cytopl
asm///plasma membrane///vesicle
## ILMN_1651209
integral component of membrane
## ILMN_1651228
cytoplasm///cytosol///cytosolic small ribosomal subunit///extracellular exosome///nucleoplasm///small ribosomal s
ubunit
## ILMN_1651229
cytoplasm///nuclear membrane
## ILMN_1651232
##
GO:Function ID
## ILMN_1343291                           GO:0005525///GO:0003924///GO:0044822///GO:0005515///GO:0019901///GO:000
0049///GO:0003746
## ILMN_1343295 GO:0051287///GO:0050661///GO:0004365///GO:0004365///GO:0004365///GO:0042802///GO:0008017///GO:003
5605///GO:0005515
## ILMN_1651209
## ILMN_1651228                                                                               GO:0044822///GO:000
5515///GO:0003735
## ILMN_1651229                                                                  GO:0008536///GO:0008139///GO:000
5515///GO:0008565
## ILMN_1651232
##
GO:Process ID
## ILMN_1343291                                                                  GO:0071364///GO:1904714///GO:000
6355///GO:0006351///GO:0006414
## ILMN_1343295 GO:0061621///GO:0071346///GO:0006094///GO:0000226///GO:0017148///GO:0017148///GO:0051402///GO:003
5606///GO:0050821///GO:0016241
## ILMN_1651209
## ILMN_1651228 GO:0006614///GO:0000184///GO:0006407///GO:0006364///GO:0006364///GO:0042274///GO:0042254///GO:000
6412///GO:0006413///GO:0019083
## ILMN_1651229
GO:0006606
## ILMN_1651232
##
GO:Component ID
## ILMN_1343291                                                                  GO:0030864///GO:0005737///GO:000
5737///GO:0005737///GO:0098574///GO:0005829///GO:0005829///GO:0005853///GO:0070062///GO:0005615///GO:0016020///G
O:0005730///GO:0005634///GO:0032587
## ILMN_1343295 GO:0097452///GO:0005737///GO:0005737///GO:0005829///GO:0005829///GO:0005829///GO:0070062///GO:003
1012///GO:0043231///GO:0030529///GO:0005811///GO:0016020///GO:0015630///GO:0031965///GO:0005634///GO:0005634///G
O:0048471///GO:0005886///GO:0031982
## ILMN_1651209
GO:0016021
## ILMN_1651228
GO:0005737///GO:0005829///GO:0022627///GO:0070062///GO:0005654///GO:0015935
## ILMN_1651229
GO:0005737///GO:0031965
## ILMN_1651232
##                                               Platform_SEQUENCE
## ILMN_1343291 TGTGTTGAGAGCTTCTCAGACTATCCACCTTTGGGTCGCTTTGCTGTTCG
## ILMN_1343295 CTTCAACAGCGACACCCACTCCTCCACCTTTGACGCTGGGGCTGGCATTG
## ILMN_1651209 TCACGGCGTACGCCCTCATGGGGAAAATCTCCCCGGTGACTTTCAGGTCC
## ILMN_1651228 CGCCACACGTAACTGAGATGCTCCTTTAAATAAAGCGTTTGTGTTTCAAG
## ILMN_1651229 ACAAGAGGCGGGTGAAGGAGATGGTGAAGGAGTTCACACTGCTGTGCCGG
## ILMN_1651232 GGGAGGCTCGGTCACAATCTCCGAGAGCACAGCCATCATCTCTCACGGTA
```

```
# this will help us in identifying condition, we need to modify the data according to condition
condition<- res$`subject status:ch1`
res$condition <- gsub("\\+", "_", condition)
res$condition
```

```
##  [1] "healthy donor"
##  [2] "healthy donor"
##  [3] "healthy donor"
##  [4] "healthy donor"
##  [5] "healthy donor"
##  [6] "healthy donor"
##  [7] "healthy donor"
##  [8] "healthy donor"
##  [9] "healthy donor"
## [10] "healthy donor"
## [11] "healthy donor"
## [12] "healthy donor"
## [13] "patients with deficiencies for UNC93B"
## [14] "patients with deficiencies for UNC93B"
## [15] "patients with deficiencies for UNC93B"
## [16] "patients with deficiencies for IRF7"
## [17] "patients with deficiencies for IRF7"
## [18] "patients with deficiencies for IRF7"
```

```
# Now we collapse the dataset with genesymbols, similar to what we did in phantasus
res <- collapseBy(res, fData(res)$`Gene symbol`, FUN=median)
res <- res[!grepl("///", rownames(res)), ]
res <- res[rownames(res) != "", ]
```

```
#clean white spaces
res$condition <- c("healthy_donor","healthy_donor","healthy_donor","healthy_donor","healthy_donor","healthy_dono
r","healthy_donor","healthy_donor","healthy_donor","healthy_donor","healthy_donor","healthy_donor","patients_with
_deficiencies_for_UNC93B","patients_with_deficiencies_for_UNC93B","patients_with_deficiencies_for_UNC93B","patien
ts_with_deficiencies_for_IRF7","patients_with_deficiencies_for_IRF7","patients_with_deficiencies_for_IRF7")
res$condition
```

```
##  [1] "healthy_donor"
##  [2] "healthy_donor"
##  [3] "healthy_donor"
##  [4] "healthy_donor"
##  [5] "healthy_donor"
##  [6] "healthy_donor"
##  [7] "healthy_donor"
##  [8] "healthy_donor"
##  [9] "healthy_donor"
## [10] "healthy_donor"
## [11] "healthy_donor"
## [12] "healthy_donor"
## [13] "patients_with_deficiencies_for_UNC93B"
## [14] "patients_with_deficiencies_for_UNC93B"
## [15] "patients_with_deficiencies_for_UNC93B"
## [16] "patients_with_deficiencies_for_IRF7"
## [17] "patients_with_deficiencies_for_IRF7"
## [18] "patients_with_deficiencies_for_IRF7"
```

```
# let's annotate the symbols with the human database entries

fData(res) <- data.frame(row.names = rownames(res))

fData(res)$entrez <- row.names(fData(res))

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

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

```
# let's normalize this data

res.qnorm <- res

summary(exprs(res.qnorm))
```

```
##    GSM1623213       GSM1623214       GSM1623215       GSM1623216    
##  Min.   : 3.322   Min.   : 3.322   Min.   : 3.322   Min.   : 3.322  
##  1st Qu.: 4.497   1st Qu.: 4.452   1st Qu.: 4.480   1st Qu.: 4.592  
##  Median : 6.451   Median : 6.415   Median : 6.434   Median : 6.486  
##  Mean   : 6.546   Mean   : 6.515   Mean   : 6.538   Mean   : 6.579  
##  3rd Qu.: 8.211   3rd Qu.: 8.184   3rd Qu.: 8.199   3rd Qu.: 8.220  
##  Max.   :13.887   Max.   :13.887   Max.   :13.887   Max.   :13.887  
##    GSM1623217       GSM1623218       GSM1623219       GSM1623220    
##  Min.   : 3.322   Min.   : 3.322   Min.   : 3.322   Min.   : 3.322  
##  1st Qu.: 4.501   1st Qu.: 4.491   1st Qu.: 4.478   1st Qu.: 4.439  
##  Median : 6.422   Median : 6.447   Median : 6.433   Median : 6.403  
##  Mean   : 6.537   Mean   : 6.549   Mean   : 6.527   Mean   : 6.514  
##  3rd Qu.: 8.180   3rd Qu.: 8.194   3rd Qu.: 8.189   3rd Qu.: 8.175  
##  Max.   :13.887   Max.   :13.887   Max.   :13.887   Max.   :13.836  
##    GSM1623221       GSM1623222       GSM1623223       GSM1623224    
##  Min.   : 3.322   Min.   : 3.322   Min.   : 3.322   Min.   : 3.322  
##  1st Qu.: 4.443   1st Qu.: 4.592   1st Qu.: 4.493   1st Qu.: 4.500  
##  Median : 6.417   Median : 6.486   Median : 6.407   Median : 6.436  
##  Mean   : 6.526   Mean   : 6.593   Mean   : 6.524   Mean   : 6.556  
##  3rd Qu.: 8.182   3rd Qu.: 8.230   3rd Qu.: 8.163   3rd Qu.: 8.215  
##  Max.   :13.857   Max.   :13.887   Max.   :13.887   Max.   :13.887  
##    GSM1623225       GSM1623226       GSM1623227       GSM1623228    
##  Min.   : 3.322   Min.   : 3.322   Min.   : 3.322   Min.   : 3.322  
##  1st Qu.: 4.462   1st Qu.: 4.499   1st Qu.: 4.415   1st Qu.: 4.494  
##  Median : 6.416   Median : 6.470   Median : 6.420   Median : 6.468  
##  Mean   : 6.519   Mean   : 6.548   Mean   : 6.520   Mean   : 6.552  
##  3rd Qu.: 8.177   3rd Qu.: 8.197   3rd Qu.: 8.176   3rd Qu.: 8.210  
##  Max.   :13.797   Max.   :13.857   Max.   :13.887   Max.   :13.857  
##    GSM1623229       GSM1623230    
##  Min.   : 3.322   Min.   : 3.322  
##  1st Qu.: 4.494   1st Qu.: 4.503  
##  Median : 6.457   Median : 6.452  
##  Mean   : 6.551   Mean   : 6.556  
##  3rd Qu.: 8.214   3rd Qu.: 8.216  
##  Max.   :13.887   Max.   :13.887
```

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

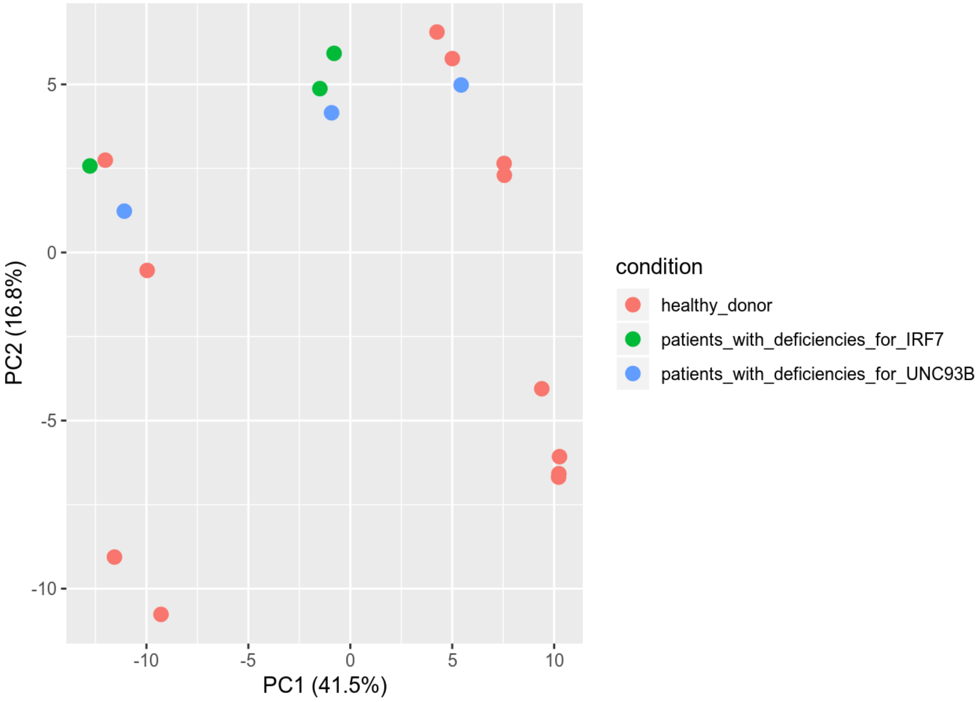
```
##    GSM1623213      GSM1623214      GSM1623215      GSM1623216   
##  Min.   :2.112   Min.   :2.112   Min.   :2.112   Min.   :2.112  
##  1st Qu.:2.457   1st Qu.:2.457   1st Qu.:2.457   1st Qu.:2.457  
##  Median :2.895   Median :2.895   Median :2.895   Median :2.895  
##  Mean   :2.842   Mean   :2.841   Mean   :2.841   Mean   :2.842  
##  3rd Qu.:3.201   3rd Qu.:3.201   3rd Qu.:3.201   3rd Qu.:3.201  
##  Max.   :3.895   Max.   :3.895   Max.   :3.895   Max.   :3.895  
##    GSM1623217      GSM1623218      GSM1623219      GSM1623220   
##  Min.   :2.112   Min.   :2.112   Min.   :2.112   Min.   :2.112  
##  1st Qu.:2.457   1st Qu.:2.457   1st Qu.:2.457   1st Qu.:2.457  
##  Median :2.895   Median :2.895   Median :2.895   Median :2.895  
##  Mean   :2.842   Mean   :2.842   Mean   :2.842   Mean   :2.842  
##  3rd Qu.:3.201   3rd Qu.:3.201   3rd Qu.:3.201   3rd Qu.:3.201  
##  Max.   :3.895   Max.   :3.895   Max.   :3.895   Max.   :3.895  
##    GSM1623221      GSM1623222      GSM1623223      GSM1623224   
##  Min.   :2.112   Min.   :2.112   Min.   :2.112   Min.   :2.112  
##  1st Qu.:2.457   1st Qu.:2.457   1st Qu.:2.457   1st Qu.:2.457  
##  Median :2.895   Median :2.895   Median :2.895   Median :2.895  
##  Mean   :2.842   Mean   :2.842   Mean   :2.842   Mean   :2.841  
##  3rd Qu.:3.201   3rd Qu.:3.201   3rd Qu.:3.201   3rd Qu.:3.201  
##  Max.   :3.895   Max.   :3.895   Max.   :3.895   Max.   :3.895  
##    GSM1623225      GSM1623226      GSM1623227      GSM1623228   
##  Min.   :2.112   Min.   :2.112   Min.   :2.112   Min.   :2.112  
##  1st Qu.:2.457   1st Qu.:2.457   1st Qu.:2.457   1st Qu.:2.457  
##  Median :2.895   Median :2.895   Median :2.895   Median :2.895  
##  Mean   :2.842   Mean   :2.842   Mean   :2.842   Mean   :2.842  
##  3rd Qu.:3.201   3rd Qu.:3.201   3rd Qu.:3.201   3rd Qu.:3.201  
##  Max.   :3.895   Max.   :3.895   Max.   :3.895   Max.   :3.895  
##    GSM1623229      GSM1623230   
##  Min.   :2.112   Min.   :2.112  
##  1st Qu.:2.457   1st Qu.:2.457  
##  Median :2.895   Median :2.895  
##  Mean   :2.842   Mean   :2.842  
##  3rd Qu.:3.201   3rd Qu.:3.201  
##  Max.   :3.895   Max.   :3.895
```

```
res.qnorm.top12K <- res.qnorm
# let's get top 12000 entries
res.qnorm.top12K <-res.qnorm.top12K[head(order(apply(exprs(res.qnorm.top12K), 1, mean), decreasing = TRUE), 12000
), ]
```

```
# Now let's look at the dataset
#pdf('pca_dataset1.pdf')

#also we can make PCA plot from our dataset
pcaPlot(res.qnorm.top12K, 1, 2) + aes(color=condition)
```

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

```
## Registered S3 methods overwritten by 'ggplot2':
##   method         from 
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang
```



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

```
# Now we make a design matrix that will be used to make a model for the given data
res.design <- model.matrix(~0+condition, data=pData(res.qnorm.top12K))
res.design
```

```
##            conditionhealthy_donor
## GSM1623213                      1
## GSM1623214                      1
## GSM1623215                      1
## GSM1623216                      1
## GSM1623217                      1
## GSM1623218                      1
## GSM1623219                      1
## GSM1623220                      1
## GSM1623221                      1
## GSM1623222                      1
## GSM1623223                      1
## GSM1623224                      1
## GSM1623225                      0
## GSM1623226                      0
## GSM1623227                      0
## GSM1623228                      0
## GSM1623229                      0
## GSM1623230                      0
##            conditionpatients_with_deficiencies_for_IRF7
## GSM1623213                                            0
## GSM1623214                                            0
## GSM1623215                                            0
## GSM1623216                                            0
## GSM1623217                                            0
## GSM1623218                                            0
## GSM1623219                                            0
## GSM1623220                                            0
## GSM1623221                                            0
## GSM1623222                                            0
## GSM1623223                                            0
## GSM1623224                                            0
## GSM1623225                                            0
## GSM1623226                                            0
## GSM1623227                                            0
## GSM1623228                                            1
## GSM1623229                                            1
## GSM1623230                                            1
##            conditionpatients_with_deficiencies_for_UNC93B
## GSM1623213                                              0
## GSM1623214                                              0
## GSM1623215                                              0
## GSM1623216                                              0
## GSM1623217                                              0
## GSM1623218                                              0
## GSM1623219                                              0
## GSM1623220                                              0
## GSM1623221                                              0
## GSM1623222                                              0
## GSM1623223                                              0
## GSM1623224                                              0
## GSM1623225                                              1
## GSM1623226                                              1
## GSM1623227                                              1
## GSM1623228                                              0
## GSM1623229                                              0
## GSM1623230                                              0
## attr(,"assign")
## [1] 1 1 1
## attr(,"contrasts")
## attr(,"contrasts")$condition
## [1] "contr.treatment"
```

```
#we have 3 condition:
intermediate <- data.frame (res.design)
colnames(intermediate) <-c("conditionhealthy_donor","conditionpatients_with_deficiencies_for_IRF7", "conditionpat
ients_with_deficiencies_for_UNC93B")

rm(res.design)
res.design <- as.matrix(intermediate)

# based on this matrix we fit our data
fit <- lmFit(res.qnorm.top12K, res.design)
# we will also make a bayisian model for the data called fit2
# this is the tricky part, because we need to choose contrast names

fit2 <- contrasts.fit(fit, makeContrasts(conditionhealthy_donor-conditionpatients_with_deficiencies_for_IRF7, con
ditionhealthy_donor-conditionpatients_with_deficiencies_for_UNC93B,
                                         levels=res.design))

# View(res.design)
fit2 <- eBayes(fit2)

# now let's do a bonferroni-hochback correction
de <- topTable(fit2, adjust.method="BH", number=Inf)
head(de)
```

```
##              entrez symbol
## C9orf135   C9orf135 138255
## HLA-DQB1   HLA-DQB1   3119
## TPSAB1       TPSAB1   7177
## RALGPS2     RALGPS2  55103
## COMMD2       COMMD2  51122
## LINC00347 LINC00347 338864
##           conditionhealthy_donor...conditionpatients_with_deficiencies_for_IRF7
## C9orf135                                                              0.0000000
## HLA-DQB1                                                              1.0889892
## TPSAB1                                                               -0.5321902
## RALGPS2                                                              -0.3627532
## COMMD2                                                                0.2165416
## LINC00347                                                            -0.1834252
##           conditionhealthy_donor...conditionpatients_with_deficiencies_for_UNC93B
## C9orf135                                                             -0.662052476
## HLA-DQB1                                                              0.015570626
## TPSAB1                                                                0.014230798
## RALGPS2                                                              -0.006949756
## COMMD2                                                                0.012925421
## LINC00347                                                             0.002137880
##            AveExpr         F      P.Value    adj.P.Val
## C9orf135  2.222017 320.89471 3.995312e-14 4.794374e-10
## HLA-DQB1  3.016571 207.00479 1.419573e-12 8.517436e-09
## TPSAB1    2.212232 194.39685 2.357785e-12 9.431139e-09
## RALGPS2   2.180358 183.36194 3.774904e-12 1.132471e-08
## COMMD2    3.083436  77.94855 3.101882e-09 7.444517e-06
## LINC00347 2.144028  72.95127 5.121532e-09 1.024306e-05
```

```
# now we can use the big matrix de, to select top differentially expressed genes using p-values
# we can also make pca's, heatmaps etc. But most importantly, we can do pathway analysis
####
# FGSEA
####

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
## The following objects are masked from 'package:dplyr':
## 
##     between, first, last
```

```
de <- as.data.table(de, keep.rownames=TRUE)
de[entrez == "IRF7"]
```

```
##      rn entrez symbol
## 1: IRF7   IRF7   3665
##    conditionhealthy_donor...conditionpatients_with_deficiencies_for_IRF7
## 1:                                                            0.06754679
##    conditionhealthy_donor...conditionpatients_with_deficiencies_for_UNC93B
## 1:                                                              0.03795509
##     AveExpr         F   P.Value adj.P.Val
## 1: 3.609687 0.1702983 0.8448489 0.9367261
```

```
# we can see that de matrix stores information about the gene expression
```

```
# Let's make a new matrix de2 which will store information about pathways
de2 <- data.frame(de$entrez, de$P.Value)
colnames(de2) <- c('ENTREZ', 'stat')

# BiocManager::install('fgsea')
library(fgsea)
```

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

```
library(tibble)

# let's get the rank of genes from top differentially expressed to non significant
ranks <- deframe(de2)
head(ranks, 20)
```

```
##     C9orf135     HLA-DQB1       TPSAB1      RALGPS2       COMMD2 
## 3.995312e-14 1.419573e-12 2.357785e-12 3.774904e-12 3.101882e-09 
##    LINC00347        USMG5         ESR2        STOX1      CAMK2N1 
## 5.121532e-09 2.251907e-08 4.207472e-08 6.000130e-08 2.843293e-07 
##        PRRG2        PTPRM         DKK3         DNLZ         GCAT 
## 3.548328e-07 5.640258e-07 9.900380e-07 1.329012e-06 1.953206e-06 
##       POLR3G       KIF21A          NDP       JCHAIN        FXYD2 
## 2.017197e-06 3.297576e-06 4.759237e-06 5.430128e-06 5.831750e-06
```

```
# Load the pathways into a named list
# BiocManager::install('msigdbr')
library(msigdbr)

m_df <- msigdbr(species = "Homo sapiens")
# View(m_df)
pathways <- split(m_df$human_gene_symbol, m_df$gs_name)
head(pathways)
```

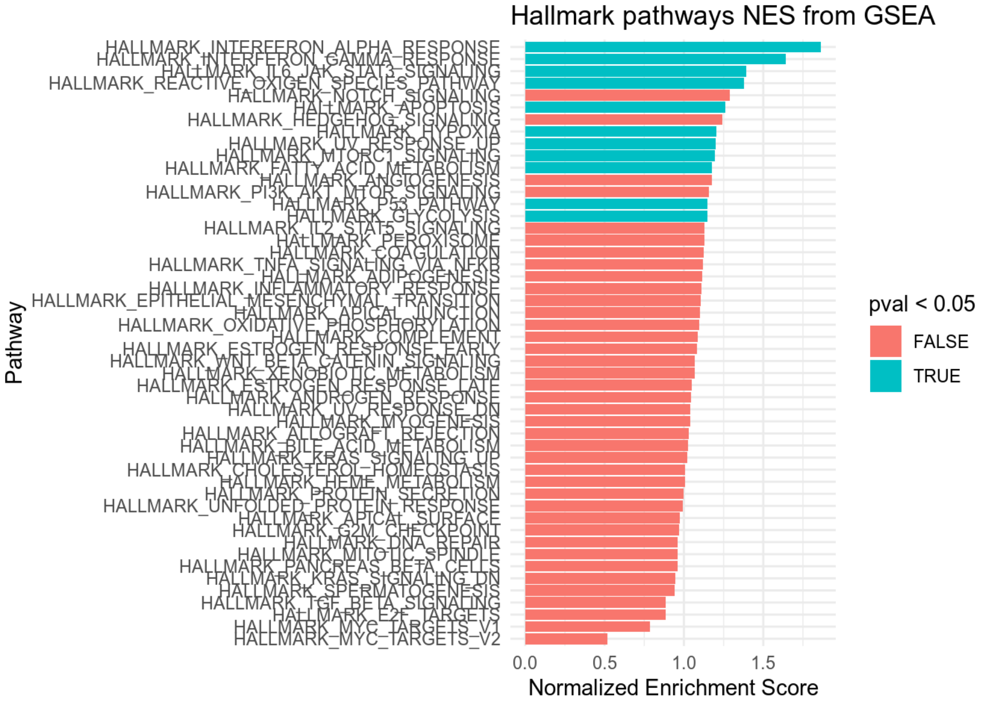
```
## $AAACCAC_MIR140
##   [1] "ABCC4"     "ACTN4"     "ACVR1"     "ADAM9"     "ADAMTS5"
##   [6] "AGER"      "ANK2"      "API5"      "BACH1"     "BAZ2B"
##  [11] "BCL11A"    "BCL2L2"    "BCL9"      "C15orf29"  "C1orf21"
##  [16] "C3orf58"   "C7orf60"   "CACNA1C"   "CEBPA"     "CHD4"
##  [21] "CIT"       "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"
```

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

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"
```

```r
# running the fgsea algorithm on hallmark.pathways

fgseaRes <- fgsea(pathways=pathways.hallmark, stats=ranks, nperm=1000)

fgseaResTidy <- fgseaRes %>%
  as_tibble() %>%
  arrange(desc(NES))
```

```r
# ggplotting for hallmark pathways
library(ggplot2)
#pdf("fgseaResTidy.pdf", width = 10, height = 10)

ggplot(fgseaResTidy, 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()
```

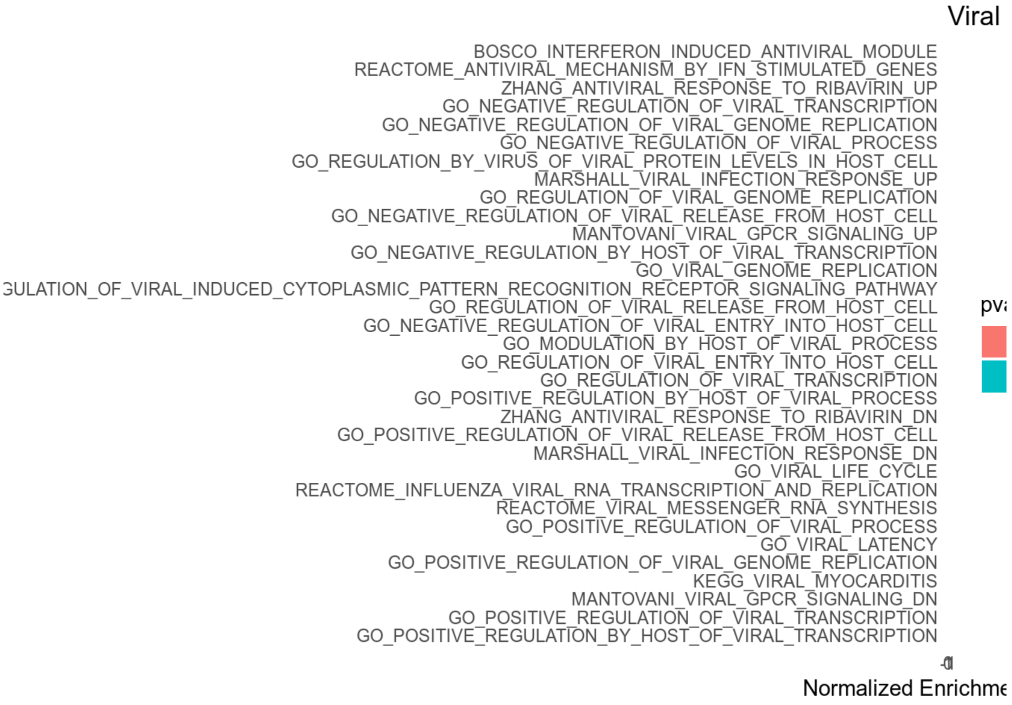

Hallmark pathways NES from GSEA
HALLMARK_INTEREFERON_ALPHA_RESPONSE
HALLMARK_INTERFERON_GAMMA_RESPONSE
HALLMARK_IL6_JAK_STAT3_SIGNALING
HALLMARK_REACTIVE_OXIGEN_SPECIES_PATHWAY
HALLMARK_NOTCH_SIGNALING
HALLMARK_APOPTOSIS
HALLMARK_HEDGEHOG_SIGNALING
HALLMARK_HYPOXIA
HALLMARK_UV_RESPONSE_UP
HALLMARK_MTORC1_SIGNALING
HALLMARK_FATTY_ACID_METABOLISM
HALLMARK_ANGIOGENESIS
HALLMARK_PI3K_AKT_MTOR_SIGNALING
HALLMARK_P53_PATHWAY
HALLMARK_GLYCOLYSIS
HALLMARK_IL2_STAT5_SIGNALING
HALLMARK_PEROXISOME
HALLMARK_COAGULATION
HALLMARK_TNFA_SIGNALING_VIA_NFKB
HALLMARK_ADIPOGENESIS
HALLMARK_INFLAMMATORY_RESPONSE
HALLMARK_EPITHELIAL_MESENCHYMAL_TRANSITION
HALLMARK_APICAL_JUNCTION
HALLMARK_OXIDATIVE_PHOSPHORYLATION
HALLMARK_COMPLEMENT
HALLMARK_ESTROGEN_RESPONSE_EARLY
HALLMARK_WNT_BETA_CATENIN_SIGNALING
HALLMARK_XENOBIOTIC_METABOLISM
HALLMARK_ESTROGEN_RESPONSE_LATE
HALLMARK_ANDROGEN_RESPONSE
HALLMARK_UV_RESPONSE_DN
HALLMARK_MYOGENESIS
HALLMARK_ALLOGRAFT_REJECTION
HALLMARK_BILE_ACID_METABOLISM
HALLMARK_KRAS_SIGNALING_UP
HALLMARK_CHOLESTEROL_HOMEOSTASIS
HALLMARK_HEME_METABOLISM
HALLMARK_PROTEIN_SECRETION
HALLMARK_UNFOLDED_PROTEIN_RESPONSE
HALLMARK_APICAL_SURFACE
HALLMARK_G2M_CHECKPOINT
HALLMARK_DNA_REPAIR
HALLMARK_MITOTIC_SPINDLE
HALLMARK_PANCREAS_BETA_CELLS
HALLMARK_KRAS_SIGNALING_DN
HALLMARK_SPERMATOGENESIS
HALLMARK_TGF_BETA_SIGNALING
HALLMARK_E2F_TARGETS
HALLMARK_MYC_TARGETS_V1
HALLMARK_MYC_TARGETS_V2
Pathway
0.0
0.5
1.0
1.5
Normalized Enrichment Score
pval < 0.05
FALSE
TRUE

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

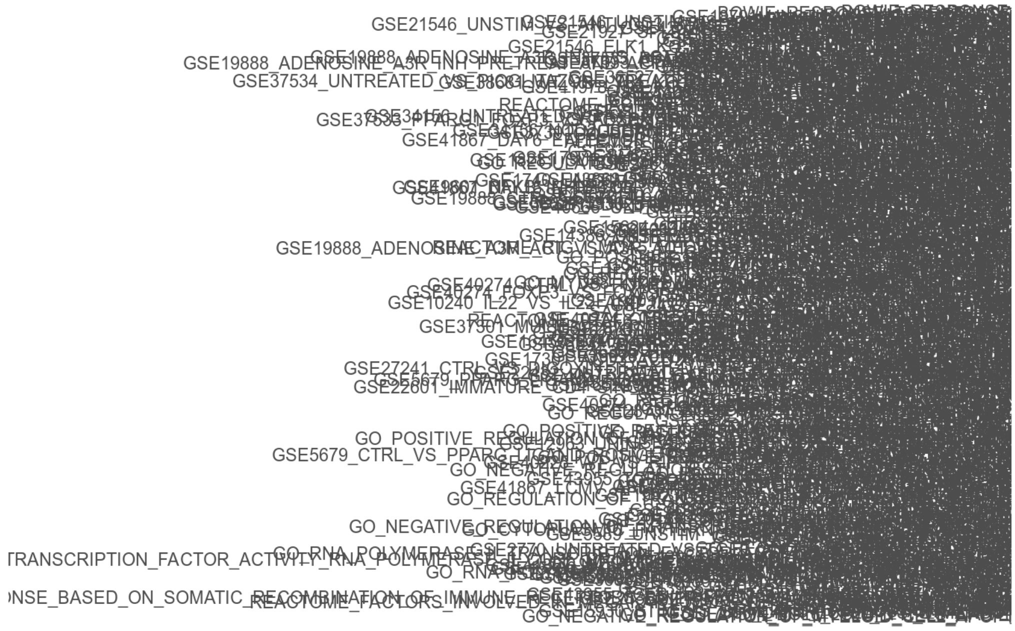
```
# We have just plotted all the significant patways in the hallmark pathways as 'blue'

# Let's look at all viral (Influenza) pathways

pathways.viral <- m_df[m_df$gs_name %like% "VIRAL_", ]
pathways.viral <- split(pathways.viral$human_gene_symbol, pathways.viral$gs_name)

# let's also get every pathway that contains the gene IRF7

#----------------------------------------

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

item <- data.frame('IRF7')

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('new_numbers.txt')
#
# options(max.print=2000)
#
# for(i in item){
#   print(grep(i, fgseaRes.all$leadingEdge))
# }
#
# sink()

# we have to do some cleaning of the data before importing it as csv
# getting only unique values from all numbers, because one gene may overlap with other, we only want the unique r
ow numbers

new_numbers <- read.table("/home/sedreh/Documents/rnaseq/data3/new_numbers.txt", quote="\"", comment.char="")

unique_vals <- data.frame(as.integer(unique(unlist(new_numbers))))

colnames(unique_vals) <- c('row_number')

final.pathways <- subset(fgseaRes.all, rownames(fgseaRes.all) %in% unique_vals$row_number)


#_---------------------------------

# 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"
```

```
# running the fgsea algorithm on viral pathways

fgseaRes_viral <- fgsea(pathways=pathways.viral, stats=ranks, nperm=1000)

fgseaResTidy_viral <- fgseaRes_viral %>%
  as_tibble() %>%
  arrange(desc(NES))


# Let's look at the plot

# ggplotting for viral pathways
library(ggplot2)

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



```
# ggplotting for IRF7 pathways

# pdf('irf7_pathway_enrichment', width = 20, height = 100)
ggplot(final.pathways, aes(reorder(pathway, NES), NES)) +
  geom_col(aes(fill=pval<0.05)) +
  coord_flip() +
  labs(x="Pathway", y="Normalized Enrichment Score",
       title="Pathways involving IRF7 gene") +
  theme_minimal()
```



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

```
# install.packages('DT')
library(DT)
# Show a table for all pathways
fgseaResTidy %>%
  dplyr::select(-leadingEdge, -ES, -nMoreExtreme) %>%
  arrange(padj) %>%
  DT::datatable()
```

Show 10 ▾ entries

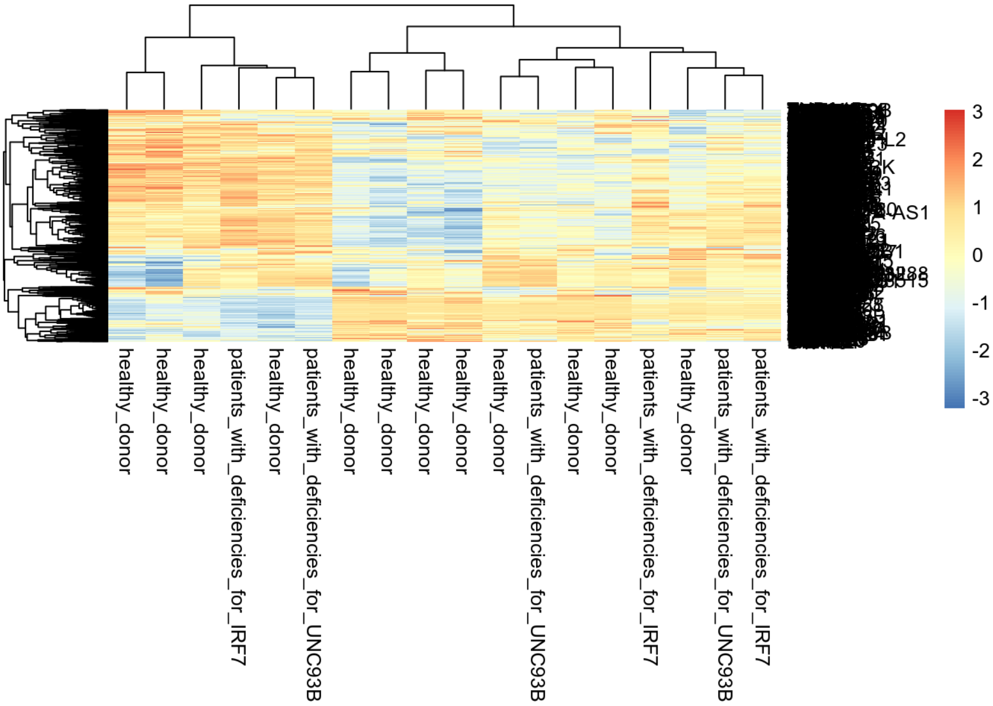
Search:

	pathway	pval	padj	NES	size
1	HALLMARK_INTERFERON_ALPHA_RESPONSE	0.000999000999000999	0.024975024975025	1.86371215126733	92
2	HALLMARK_INTERFERON_GAMMA_RESPONSE	0.000999000999000999	0.024975024975025	1.64052652060071	191
3	HALLMARK_IL6_JAK_STAT3_SIGNALING	0.001998001998002	0.0333000333000333	1.39165357434151	76
4	HALLMARK_APOPTOSIS	0.004995004995005	0.0624375624375624	1.26087391788083	141
5	HALLMARK_REACTIVE_OXIGEN_SPECIES_PATHWAY	0.00899100899100899	0.0832500832500833	1.37764677242465	41
6	HALLMARK_MTORC1_SIGNALING	0.00999000999000999	0.0832500832500833	1.19574061881753	183
7	HALLMARK_HYPOXIA	0.012987012987013	0.0927643784786642	1.20628201438214	154
8	HALLMARK_UV_RESPONSE_UP	0.021978021978022	0.137362637362637	1.1988162499905	126
9	HALLMARK_FATTY_ACID_METABOLISM	0.033966033966034	0.188700188700189	1.17817062299196	132
10	HALLMARK_P53_PATHWAY	0.044955044955045	0.222504767959313	1.14906733901612	167

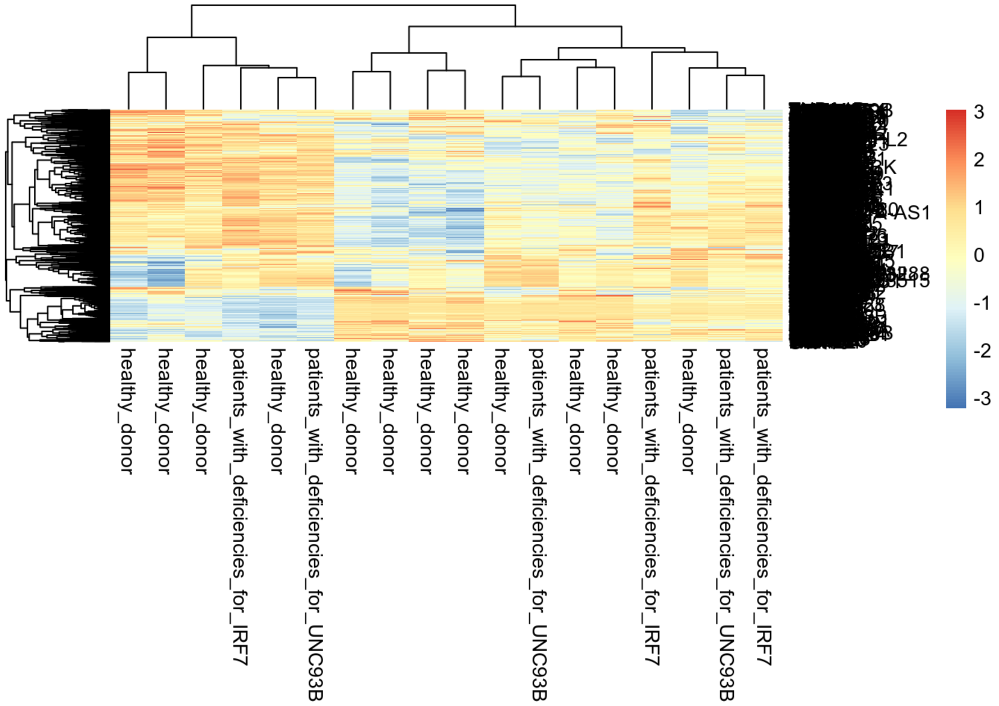
Showing 1 to 10 of 50 entries

Previous 1 2 3 4 5 Next

```
# heatmap
library(pheatmap)
#scale rows
xt<-t(as.matrix(res.qnorm.top12K)) # this is a matrix of normalised 12k genes
xts<-scale(xt)
xtst<-t(xts)
xtst <- na.omit(xtst)
colnames(xtst) <- res$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=20, height=200)

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

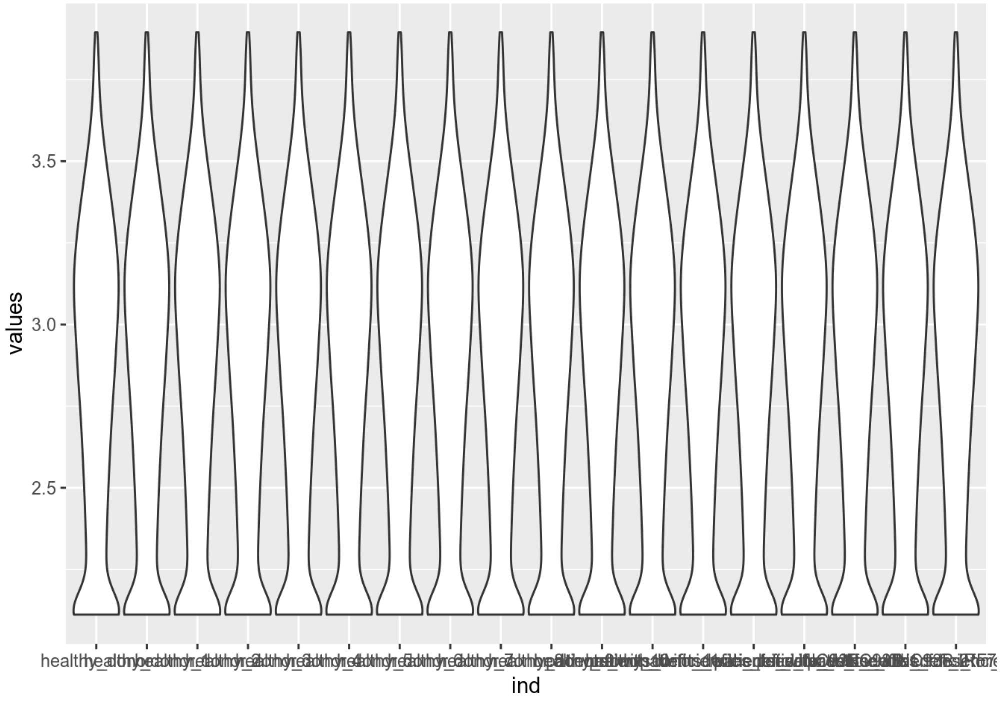


```
#output plot to file
# dev.off()


 # let's get a smaller heatmap
 # pdf('heatmap_small.pdf', width = 20, height = 50)

pheatmap(h)
```



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

```
# let's make a boxplot of the data

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

#pdf('box_dataset.pdf', width = 50)

gt <- t(xt) # taking xt from the heatmap and transposing it
colnames(gt)<- res$condition # now giving it labels from condition
ggplot(stack(data.frame(gt)), aes(x = ind, y = values)) +
  geom_violin()
```



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