Bladder cohort - R Notebook

Adam Cankaya
acankaya2017@fau.edu
Based on tutorial found at https://www.costalab.org/wp-content/uploads/2020/11/R_class_D3.htm

First install BiocManager, edgeR, and TCGAbiolinks

```r
if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

BiocManager::install("edgeR")
```

```
## Bioconductor version 3.19 (BiocManager 1.30.23), R 4.4.0 (2024-04-24 ucrt)
```

```
## Warning: package(s) not installed when version(s) same as or greater than current; use
##   ‘force = TRUE‘ to re-install: ’edgeR’
```

```
## Installation paths not writeable, unable to update packages
##   path: C:/Program Files/R/R-4.4.0/library
##   packages:
##     KernSmooth, survival
```

```r
BiocManager::install("TCGAbiolinks")
```

```
## Bioconductor version 3.19 (BiocManager 1.30.23), R 4.4.0 (2024-04-24 ucrt)
```

```
## Warning: package(s) not installed when version(s) same as or greater than current; use
##   ‘force = TRUE‘ to re-install: ’TCGAbiolinks’
```

```
## Installation paths not writeable, unable to update packages
##   path: C:/Program Files/R/R-4.4.0/library
##   packages:
##     KernSmooth, survival
```

```r
BiocManager::install("genefilter")
```

```
## Bioconductor version 3.19 (BiocManager 1.30.23), R 4.4.0 (2024-04-24 ucrt)
```

```
## Warning: package(s) not installed when version(s) same as or greater than current; use
##   ‘force = TRUE‘ to re-install: ’genefilter’
## Installation paths not writeable, unable to update packages
##   path: C:/Program Files/R/R-4.4.0/library
##   packages:
##     KernSmooth, survival
```

Step 1 - Load packages, download data from TCGA, and prepare it for DEGList

```
library("TCGAbiolinks")
library("limma")
library("edgeR")
library("glmnet")
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```
library("factoextra")
```

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

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library("FactoMineR")
library("caret")
```

```
## Loading required package: lattice
```

```
library("SummarizedExperiment")
```

```
## Loading required package: MatrixGenerics
```

```
## Loading required package: matrixStats
```

```
##
## Attaching package: 'MatrixGenerics'
```

```
## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars

## Loading required package: GenomicRanges

## Loading required package: stats4

## Loading required package: BiocGenerics

##
## Attaching package: ’BiocGenerics’

## The following object is masked from ’package:limma’:
##
##     plotMA

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

## The following objects are masked from ’package:base’:
##
##     anyDuplicated, aperm, 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, table,
##     tapply, union, unique, unsplit, which.max, which.min

## Loading required package: S4Vectors

##
## Attaching package: ’S4Vectors’

## The following objects are masked from ’package:Matrix’:
##
##     expand, unname

## The following object is masked from ’package:utils’:
##
##     findMatches

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

## Loading required package: IRanges
```

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

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

## Loading required package: GenomeInfoDb

## Loading required package: Biobase

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

##
## Attaching package: 'Biobase'

## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians

## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians
```

```r
library("gplots")
```

```
##
## Attaching package: 'gplots'

## The following object is masked from 'package:IRanges':
##
##     space

## The following object is masked from 'package:S4Vectors':
##
##     space

## The following object is masked from 'package:stats':
##
##     lowess
```

```r
library("survival")
```

```
##
## Attaching package: 'survival'

## The following object is masked from 'package:caret':
##
##     cluster
```

```r
library("survminer")
```

```
## Loading required package: ggpubr

##
## Attaching package: 'survminer'

## The following object is masked from 'package:survival':
##
##     myeloma
```

```r
library("RColorBrewer")
library("gProfileR")
library("genefilter")
```

```
##
## Attaching package: 'genefilter'

## The following objects are masked from 'package:MatrixGenerics':
##
##     rowSds, rowVars

## The following objects are masked from 'package:matrixStats':
##
##     rowSds, rowVars
```

```r
setwd('C:/Adam/R/')  # make sure it already exists

# Before we perform a GDC query let's look at the TCGA-BLCA data
# As of June 2024 we should see a case count of 412
TCGAbiolinks:::getProjectSummary("TCGA-BLCA")
```

```
## $file_count
## [1] 23394
##
## $data_categories
##    file_count case_count                data_category
## 1        6729        412  Simple Nucleotide Variation
## 2        4285        412             Sequencing Reads
## 3        1760        412                  Biospecimen
## 4         994        412                     Clinical
## 5        4478        412      Copy Number Variation
## 6        1736        412     Transcriptome Profiling
## 7        1320        412              DNA Methylation
## 8         343        343          Proteome Profiling
## 9          26         12 Somatic Structural Variation
## 10       1723        406         Structural Variation
##
## $case_count
## [1] 412
##
## $file_size
## [1] 4.082979e+14
```

```r
# Download TCGA-BLCA data from GDC
# We want the complete RNA sequencing and raw gene count data
# So we run a query of the Transcriptome Profiling category and RNA-Seq experimental type
# We use the STAR - Counts workflow type because it contains the raw gene counts we need
# We ignore other sample types besides tumor and normal
# The original paper by Wang uses the HTSeq-counts workflow, but this is a legacy version of
#  the new STAR - COUNTS workflow type
query_TCGA = GDCquery(
  project = "TCGA-BLCA",
  data.category = "Transcriptome Profiling",
  data.type="Gene Expression Quantification",
  experimental.strategy = "RNA-Seq",
  workflow.type = "STAR - Counts",
  sample.type = c("Primary Tumor", "Solid Tissue Normal"))
```

```
## --------------------------------------

## o GDCquery: Searching in GDC database

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

## Genome of reference: hg38

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

## oo Accessing GDC. This might take a while...

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

## ooo Project: TCGA-BLCA

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

## oo Filtering results

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

## ooo By experimental.strategy

## ooo By data.type

## ooo By workflow.type

## ooo By sample.type

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

## oo Checking data
## ----------------

## ooo Checking if there are duplicated cases

## ooo Checking if there are results for the query

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

## o Preparing output

## -------------------
```

```
# Run the query and format it as a table
# The results are a table with 431 rows (because some patients have multiple cases each)
# There are 29 columns with meta data about each case such as sample_type (tumor vs normal)
lihc_res = getResults(query_TCGA)

# We can create a summary table shows there are 412 tumor and 19 normal (412+19=431)
summary(factor(lihc_res$sample_type))
```

```
##       Primary Tumor Solid Tissue Normal
##                 412                  19
```

```
# Go ahead and download all the data from GDC to our working directory
GDCdownload(query = query_TCGA)
```

```
## Downloading data for project TCGA-BLCA

## Of the 431 files for download 431 already exist.

## All samples have been already downloaded
```

```
# Now load the RNA-Seq data from the files into R workspace
tcga_data = GDCprepare(query_TCGA)
```

```
## |                                                    |   0%                           |

## Starting to add information to samples

##  => Add clinical information to samples

##  => Adding TCGA molecular information from marker papers

##  => Information will have prefix 'paper_'

## blca subtype information from:doi:10.1016/j.cell.2017.09.007
## Available assays in SummarizedExperiment :
##   => unstranded
##   => stranded_first
##   => stranded_second
##   => tpm_unstrand
##   => fpkm_unstrand
##   => fpkm_uq_unstrand
```

```r
# This data object has 60660 rows and 431 columns
# This indicates there are 60660 different genes found throughout all the cases
# The object contains both clincal and expression data
dim(tcga_data)
```

```
## [1] 60660   431
```

```r
# We can access the data in the object like this which verifies 412 tumor and 19 normal
table(tcga_data@colData$definition)
```

```
## 
## Primary solid Tumor Solid Tissue Normal 
##                 412                  19
```

```r
# Or see the gender data of 117 female and 314 male
table(tcga_data@colData$gender)
```

```
## 
## female   male 
##    117    314
```

```r
# To preview the raw gene counts let's look at the expression levels of the first
#  6 genes in the first 10 samples...
head(assay(tcga_data)[,1:10])
```

```
##                    TCGA-CU-A3KJ-01A-11R-A21D-07 TCGA-K4-A3WU-01B-11R-A23N-07
## ENSG00000000003.15                         3679                        28986
## ENSG00000000005.6                             0                           21
## ENSG00000000419.13                         4190                         2917
## ENSG00000000457.14                          850                         1910
## ENSG00000000460.17                         1196                         1495
## ENSG00000000938.13                          353                          905
##                    TCGA-DK-A3IU-01A-11R-A20F-07 TCGA-GV-A40G-01A-11R-A23N-07
## ENSG00000000003.15                          951                        12697
## ENSG00000000005.6                             1                            3
## ENSG00000000419.13                         2976                         3565
## ENSG00000000457.14                          705                         1049
## ENSG00000000460.17                          655                          448
## ENSG00000000938.13                         2282                          243
##                    TCGA-DK-A3IN-01A-11R-A20F-07 TCGA-SY-A9G0-01A-12R-A38B-07
## ENSG00000000003.15                         5761                         1717
## ENSG00000000005.6                             0                            4
## ENSG00000000419.13                         1441                         1525
## ENSG00000000457.14                          746                          444
## ENSG00000000460.17                         1369                          261
## ENSG00000000938.13                          412                          238
##                    TCGA-XF-A9SU-01A-31R-A39I-07 TCGA-UY-A78O-01A-12R-A33J-07
## ENSG00000000003.15                         3954                        11311
## ENSG00000000005.6                             2                            1
## ENSG00000000419.13                         1645                         2983
## ENSG00000000457.14                          167                          469
## ENSG00000000460.17                          180                          333
## ENSG00000000938.13                          257                          187
##                    TCGA-GD-A2C5-01A-12R-A180-07 TCGA-G2-A2EK-01A-22R-A18C-07
## ENSG00000000003.15                        11308                        19633
## ENSG00000000005.6                             2                            0
## ENSG00000000419.13                         2783                         3331
## ENSG00000000457.14                         1338                         1128
## ENSG00000000460.17                         1067                          400
## ENSG00000000938.13                          317                          357
```

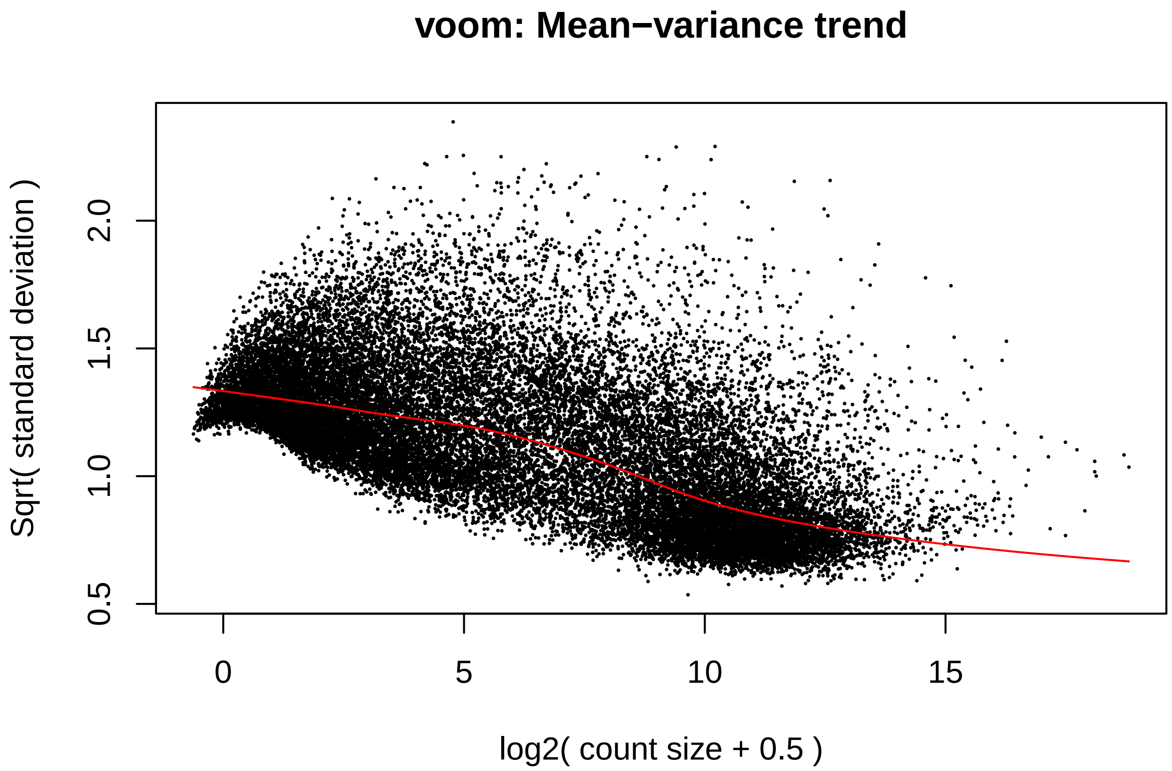
```
# And let's look at the various names of the first 6 genes...
head(rowData(tcga_data))
```

```
## DataFrame with 6 rows and 10 columns
##                      source     type     score     phase            gene_id
##                    <factor> <factor> <numeric> <integer>        <character>
## ENSG00000000003.15   HAVANA     gene        NA        NA ENSG00000000003.15
## ENSG00000000005.6    HAVANA     gene        NA        NA  ENSG00000000005.6
## ENSG00000000419.13   HAVANA     gene        NA        NA ENSG00000000419.13
## ENSG00000000457.14   HAVANA     gene        NA        NA ENSG00000000457.14
## ENSG00000000460.17   HAVANA     gene        NA        NA ENSG00000000460.17
## ENSG00000000938.13   HAVANA     gene        NA        NA ENSG00000000938.13
##                         gene_type   gene_name       level     hgnc_id
##                       <character> <character> <character> <character>
## ENSG00000000003.15 protein_coding      TSPAN6           2  HGNC:11858
## ENSG00000000005.6  protein_coding        TNMD           2  HGNC:17757
## ENSG00000000419.13 protein_coding        DPM1           2   HGNC:3005
## ENSG00000000457.14 protein_coding       SCYL3           2  HGNC:19285
## ENSG00000000460.17 protein_coding    C1orf112           2  HGNC:25565
## ENSG00000000938.13 protein_coding         FGR           2   HGNC:3697
##                             havana_gene
##                             <character>
## ENSG00000000003.15 OTTHUMG00000022002.2
## ENSG00000000005.6  OTTHUMG00000022001.2
## ENSG00000000419.13 OTTHUMG00000032742.2
## ENSG00000000457.14 OTTHUMG00000035941.6
## ENSG00000000460.17 OTTHUMG00000035821.9
## ENSG00000000938.13 OTTHUMG00000003516.3
```

Step 2 - Generate DEGList, filter low counts, and normalize data

```
# Before we can perform DEG analysis we need to normalize the data
# Let's create a limma pipeline to do this...
```

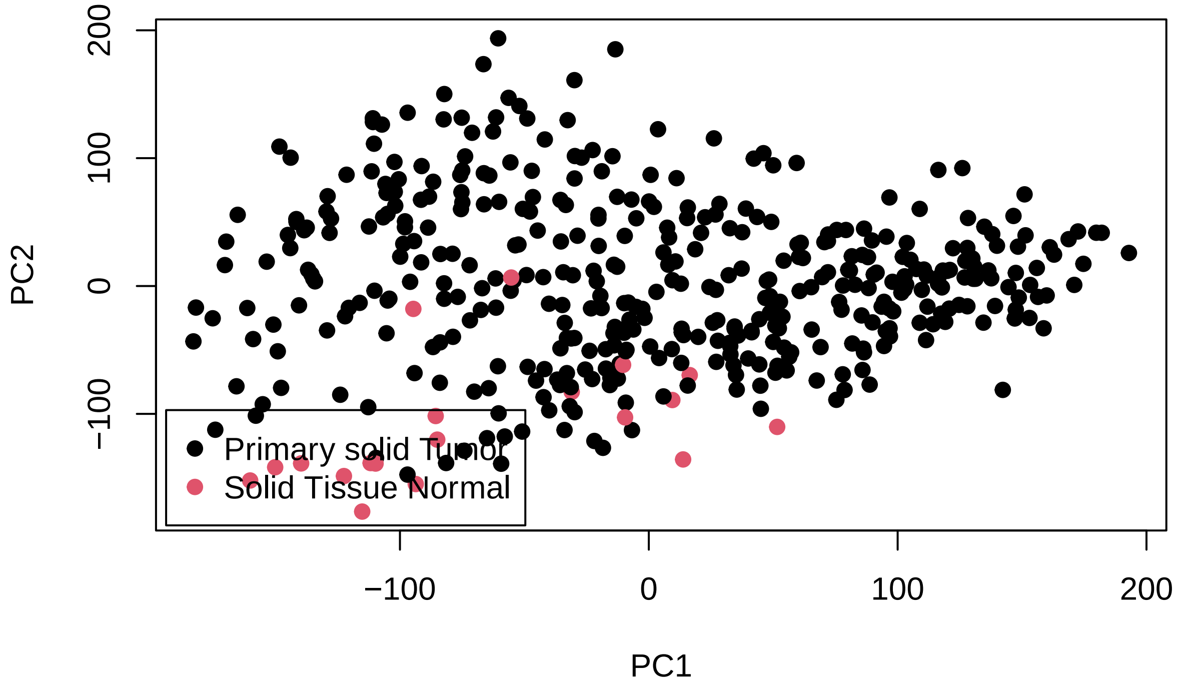
```r
# The pipeline function will take in three input parameters:
#   tcga_data - the data object we created in Step 1
#   condition_variable - the variable by which we will group patients (tumor vs normal)
#   reference_group - indicates which of the condition variable
#    values is the reference group (no tumors)
# The pipeline will return a list of three objects:
#   voom - the TMM normalized data returned by running voom
#   eBayes - the fitted model returned by running eBayes
#   topTable - a simple table which contains the top 100 differentially expressed genes
#    sorted by p.value
limma_pipeline = function(
  tcga_data,
  condition_variable,
  reference_group=NULL){

  # Create a design matrix
  # The factor is the category classifier for the data (tumor vs normal)
  #  limma requires it to be a factor object
  design_factor = colData(tcga_data)[, condition_variable, drop=T] # definition
  group = factor(design_factor) # Solid Normal Tissue

  # otherwise just pick the first class as the reference class
  if (!is.null(reference_group)) {
    group = relevel(group, ref=reference_group)
  }

  # make the design matrix
  design = model.matrix(~ group)

  # generate the DGEList object using the input...
  #  counts is the raw gene counts (numericla matrix - rows as genes, columns as cases)
  #  samples is the clinical data (data frame)
  #  genes is the annotation information (data frame - gene id and names)
  # the DGEList object returned is a transformed version of tcga_data
  dge = DGEList(counts=assay(tcga_data),
                samples=colData(tcga_data),
                genes=as.data.frame(rowData(tcga_data)))

  # filtering - by default genes with less than 10 reads are removed
  keep = filterByExpr(dge,design) # genes which meet are left after filtering
  dge = dge[keep,,keep.lib.sizes=FALSE] # filter the DGEList object, only keep the genes we want
  rm(keep) # remove this object from memory because we are done with it

  # Normalization (TMM followed by voom)
  # normalizing - minimize batch effects and variation with the TMM normalization
  # TMM - trimmed mean of M-values
  # use the voom method to convert the data to have a similar variance as arrays
  #  (TODO what is this?)
  dge = calcNormFactors(dge)
  v = voom(dge, design, plot=TRUE)

  # Fit model to data given design
  #  fits a series of linear models, one to each probe
  #  then pass it to eBayes to rank the differential expression
  fit = lmFit(v, design)
  fit = eBayes(fit)

  # Show top genes
  topGenes = topTable(fit, coef=ncol(design), number=100, sort.by="p")
  print(topGenes)

  return(
    list(
      voomObj=v, # normalized data
      fit=fit, # fitted model and statistics
      topGenes=topGenes # the 100 most differentially expressed genes
    )
  )
}

# TODO only run the pipeline if we didn't already run it before and save the data to a local file
#  tcga_data = readRDS(file = "tcga_data.RDS")
#  saveRDS(object = tcga_data,
#    file = "tcga_data.RDS",
#    compress = FALSE)

# Run the pipeline on the tcga_data from step 1 and normal tissue as the reference
#  "definition" is the column name for the tissue type (tumor vs normal)
#  "Solid Tissue Normal" is our baseline/control/reference class value
# The limma_res object returned is a list of 3 objects - voomObj, fit, topGenes
limma_res = limma_pipeline(
  tcga_data=tcga_data,
  condition_variable="definition",
  reference_group="Solid Tissue Normal"
)
```



```
##                      source type score phase            gene_id
## ENSG00000164530.15 HAVANA gene    NA    NA ENSG00000164530.15
## ENSG00000168079.17 HAVANA gene    NA    NA ENSG00000168079.17
## ENSG00000163815.6  HAVANA gene    NA    NA  ENSG00000163815.6
## ENSG00000153446.16 HAVANA gene    NA    NA ENSG00000153446.16
## ENSG00000196616.14 HAVANA gene    NA    NA ENSG00000196616.14
## ENSG00000168309.18 HAVANA gene    NA    NA ENSG00000168309.18
## ENSG00000108924.14 HAVANA gene    NA    NA ENSG00000108924.14
## ENSG00000018625.15 HAVANA gene    NA    NA ENSG00000018625.15
## ENSG00000224958.6  HAVANA gene    NA    NA  ENSG00000224958.6
## ENSG00000197766.8  HAVANA gene    NA    NA  ENSG00000197766.8
## ENSG00000126218.12 HAVANA gene    NA    NA ENSG00000126218.12
## ENSG00000168477.19 HAVANA gene    NA    NA ENSG00000168477.19
## ENSG00000068976.14 HAVANA gene    NA    NA ENSG00000068976.14
## ENSG00000123560.14 HAVANA gene    NA    NA ENSG00000123560.14
## ENSG00000034971.17 HAVANA gene    NA    NA ENSG00000034971.17
## ENSG00000241158.7  HAVANA gene    NA    NA  ENSG00000241158.7
## ENSG00000168497.5  HAVANA gene    NA    NA  ENSG00000168497.5
## ENSG00000004776.13 HAVANA gene    NA    NA ENSG00000004776.13
## ENSG00000077943.8  HAVANA gene    NA    NA  ENSG00000077943.8
## ENSG00000154330.13 HAVANA gene    NA    NA ENSG00000154330.13
## ENSG00000106809.11 HAVANA gene    NA    NA ENSG00000106809.11
## ENSG00000119147.10 HAVANA gene    NA    NA ENSG00000119147.10
## ENSG00000181856.15 HAVANA gene    NA    NA ENSG00000181856.15
## ENSG00000232855.7  HAVANA gene    NA    NA  ENSG00000232855.7
## ENSG00000144218.19 HAVANA gene    NA    NA ENSG00000144218.19
## ENSG00000108018.15 HAVANA gene    NA    NA ENSG00000108018.15
## ENSG00000167281.19 HAVANA gene    NA    NA ENSG00000167281.19
## ENSG00000141052.18 HAVANA gene    NA    NA ENSG00000141052.18
## ENSG00000119508.18 HAVANA gene    NA    NA ENSG00000119508.18
## ENSG00000101605.13 HAVANA gene    NA    NA ENSG00000101605.13
## ENSG00000171368.12 HAVANA gene    NA    NA ENSG00000171368.12
## ENSG00000163145.13 HAVANA gene    NA    NA ENSG00000163145.13
## ENSG00000179915.24 HAVANA gene    NA    NA ENSG00000179915.24
## ENSG00000125851.10 HAVANA gene    NA    NA ENSG00000125851.10
## ENSG00000112936.19 HAVANA gene    NA    NA ENSG00000112936.19
## ENSG00000136546.16 HAVANA gene    NA    NA ENSG00000136546.16
## ENSG00000182253.15 HAVANA gene    NA    NA ENSG00000182253.15
## ENSG00000205221.12 HAVANA gene    NA    NA ENSG00000205221.12
## ENSG00000179388.9  HAVANA gene    NA    NA  ENSG00000179388.9
## ENSG00000123358.20 HAVANA gene    NA    NA ENSG00000123358.20
## ENSG00000189129.14 HAVANA gene    NA    NA ENSG00000189129.14
## ENSG00000118526.7  HAVANA gene    NA    NA  ENSG00000118526.7
## ENSG00000123243.15 HAVANA gene    NA    NA ENSG00000123243.15
## ENSG00000225398.3  HAVANA gene    NA    NA  ENSG00000225398.3
## ENSG00000268926.3  HAVANA gene    NA    NA  ENSG00000268926.3
## ENSG00000149294.17 HAVANA gene    NA    NA ENSG00000149294.17
## ENSG00000172403.11 HAVANA gene    NA    NA ENSG00000172403.11
## ENSG00000127528.6  HAVANA gene    NA    NA  ENSG00000127528.6
## ENSG00000172348.15 HAVANA gene    NA    NA ENSG00000172348.15
## ENSG00000004799.8  HAVANA gene    NA    NA  ENSG00000004799.8
## ENSG00000206579.9  HAVANA gene    NA    NA  ENSG00000206579.9
## ENSG00000172260.15 HAVANA gene    NA    NA ENSG00000172260.15
## ENSG00000181072.11 HAVANA gene    NA    NA ENSG00000181072.11
## ENSG00000231943.9  HAVANA gene    NA    NA  ENSG00000231943.9
## ENSG00000065325.13 HAVANA gene    NA    NA ENSG00000065325.13
## ENSG00000186642.16 HAVANA gene    NA    NA ENSG00000186642.16
## ENSG00000111452.13 HAVANA gene    NA    NA ENSG00000111452.13
## ENSG00000268388.6  HAVANA gene    NA    NA  ENSG00000268388.6
## ENSG00000181234.9  HAVANA gene    NA    NA  ENSG00000181234.9
## ENSG00000173175.15 HAVANA gene    NA    NA ENSG00000173175.15
## ENSG00000163431.13 HAVANA gene    NA    NA ENSG00000163431.13
## ENSG00000156218.13 HAVANA gene    NA    NA ENSG00000156218.13
## ENSG00000176533.13 HAVANA gene    NA    NA ENSG00000176533.13
## ENSG00000103241.7  HAVANA gene    NA    NA  ENSG00000103241.7
## ENSG00000140538.16 HAVANA gene    NA    NA ENSG00000140538.16
## ENSG00000141338.14 HAVANA gene    NA    NA ENSG00000141338.14
## ENSG00000121671.12 HAVANA gene    NA    NA ENSG00000121671.12
## ENSG00000280429.1  HAVANA gene    NA    NA  ENSG00000280429.1
## ENSG00000149090.12 HAVANA gene    NA    NA ENSG00000149090.12
## ENSG00000070193.5  HAVANA gene    NA    NA  ENSG00000070193.5
## ENSG00000144655.15 HAVANA gene    NA    NA ENSG00000144655.15
## ENSG00000118407.15 HAVANA gene    NA    NA ENSG00000118407.15
## ENSG00000149451.18 HAVANA gene    NA    NA ENSG00000149451.18
## ENSG00000254510.2  HAVANA gene    NA    NA  ENSG00000254510.2
## ENSG00000147588.7  HAVANA gene    NA    NA  ENSG00000147588.7
## ENSG00000166091.21 HAVANA gene    NA    NA ENSG00000166091.21
## ENSG00000108405.4  HAVANA gene    NA    NA  ENSG00000108405.4
## ENSG00000154175.18 HAVANA gene    NA    NA ENSG00000154175.18
## ENSG00000135472.9  HAVANA gene    NA    NA  ENSG00000135472.9
## ENSG00000153234.15 HAVANA gene    NA    NA ENSG00000153234.15
## ENSG00000267505.1  HAVANA gene    NA    NA  ENSG00000267505.1
## ENSG00000174576.10 HAVANA gene    NA    NA ENSG00000174576.10
## ENSG00000100307.13 HAVANA gene    NA    NA ENSG00000100307.13
## ENSG00000198932.13 HAVANA gene    NA    NA ENSG00000198932.13
## ENSG00000154721.15 HAVANA gene    NA    NA ENSG00000154721.15
## ENSG00000077157.22 HAVANA gene    NA    NA ENSG00000077157.22
## ENSG00000153823.19 HAVANA gene    NA    NA ENSG00000153823.19
## ENSG00000022267.19 HAVANA gene    NA    NA ENSG00000022267.19
## ENSG00000154734.16 HAVANA gene    NA    NA ENSG00000154734.16
## ENSG00000059915.17 HAVANA gene    NA    NA ENSG00000059915.17
## ENSG00000151892.15 HAVANA gene    NA    NA ENSG00000151892.15
## ENSG00000143171.13 HAVANA gene    NA    NA ENSG00000143171.13
## ENSG00000132840.10 HAVANA gene    NA    NA ENSG00000132840.10
## ENSG00000133392.18 HAVANA gene    NA    NA ENSG00000133392.18
## ENSG00000138356.14 HAVANA gene    NA    NA ENSG00000138356.14
## ENSG00000164736.6  HAVANA gene    NA    NA  ENSG00000164736.6
## ENSG00000241684.6  HAVANA gene    NA    NA  ENSG00000241684.6
## ENSG00000108381.11 HAVANA gene    NA    NA ENSG00000108381.11
## ENSG00000188729.6  HAVANA gene    NA    NA  ENSG00000188729.6
## ENSG00000179796.12 HAVANA gene    NA    NA ENSG00000179796.12
##                                 gene_type   gene_name level    hgnc_id
## ENSG00000164530.15         protein_coding        PI16     2 HGNC:21245
## ENSG00000168079.17         protein_coding      SCARA5     2 HGNC:28701
## ENSG00000163815.6          protein_coding      CLEC3B     2 HGNC:11891
## ENSG00000153446.16         protein_coding    C16orf89     1 HGNC:28687
## ENSG00000196616.14         protein_coding       ADH1B     2   HGNC:250
## ENSG00000168309.18         protein_coding     FAM107A     1 HGNC:30827
## ENSG00000108924.14         protein_coding         HLF     1  HGNC:4977
## ENSG00000018625.15         protein_coding      ATP1A2     2   HGNC:800
## ENSG00000224958.6                  lncRNA    PGM5-AS1     1 HGNC:44181
## ENSG00000197766.8          protein_coding         CFD     2  HGNC:2771
## ENSG00000126218.12         protein_coding         F10     1  HGNC:3528
## ENSG00000168477.19         protein_coding        TNXB     1 HGNC:11976
## ENSG00000068976.14         protein_coding        PYGM     2  HGNC:9726
## ENSG00000123560.14         protein_coding        PLP1     2  HGNC:9086
## ENSG00000034971.17         protein_coding        MYOC     2  HGNC:7610
## ENSG00000241158.7                  lncRNA ADAMTS9-AS1     2 HGNC:40625
## ENSG00000168497.5          protein_coding      CAVIN2     2 HGNC:10690
## ENSG00000004776.13         protein_coding       HSPB6     2 HGNC:26511
## ENSG00000077943.8          protein_coding       ITGA8     2  HGNC:6144
## ENSG00000154330.13         protein_coding        PGM5     1  HGNC:8908
## ENSG00000106809.11         protein_coding         OGN     2  HGNC:8126
## ENSG00000119147.10         protein_coding       ECRG4     1 HGNC:24642
## ENSG00000181856.15         protein_coding      SLC2A4     2 HGNC:11009
## ENSG00000232855.7                  lncRNA  AF165147.1     2       <NA>
## ENSG00000144218.19         protein_coding        AFF3     2  HGNC:6473
## ENSG00000108018.15         protein_coding      SORCS1     2 HGNC:16697
## ENSG00000167281.19         protein_coding      RBFOX3     1 HGNC:27097
## ENSG00000141052.18         protein_coding       MYOCD     2 HGNC:16067
## ENSG00000119508.18         protein_coding       NR4A3     2  HGNC:7982
## ENSG00000101605.13         protein_coding       MYOM1     2  HGNC:7613
## ENSG00000171368.12         protein_coding        TPPP     2 HGNC:24164
## ENSG00000163145.13         protein_coding     C1QTNF7     1 HGNC:14342
## ENSG00000179915.24         protein_coding       NRXN1     1  HGNC:8008
## ENSG00000125851.10         protein_coding       PCSK2     2  HGNC:8744
## ENSG00000112936.19         protein_coding          C7     1  HGNC:1346
## ENSG00000136546.16         protein_coding       SCN7A     2 HGNC:10594
## ENSG00000182253.15         protein_coding        SYNM     1 HGNC:24466
## ENSG00000205221.12         protein_coding         VIT     1 HGNC:12697
## ENSG00000179388.9          protein_coding        EGR3     1  HGNC:3240
## ENSG00000123358.20         protein_coding       NR4A1     1  HGNC:7980
## ENSG00000189129.14         protein_coding       PLAC9     2 HGNC:19255
## ENSG00000118526.7          protein_coding       TCF21     2 HGNC:11632
## ENSG00000123243.15         protein_coding       ITIH5     1 HGNC:21449
## ENSG00000225398.3  unprocessed_pseudogene      PGM5P4     2 HGNC:49605
## ENSG00000268926.3                  lncRNA  AL354861.3     2       <NA>
## ENSG00000149294.17         protein_coding       NCAM1     1  HGNC:7656
## ENSG00000172403.11         protein_coding      SYNPO2     1 HGNC:17732
## ENSG00000127528.6          protein_coding        KLF2     2  HGNC:6347
## ENSG00000172348.15         protein_coding       RCAN2     2  HGNC:3041
## ENSG00000004799.8          protein_coding        PDK4     2  HGNC:8812
## ENSG00000206579.9          protein_coding        XKR4     2 HGNC:29394
## ENSG00000172260.15         protein_coding       NEGR1     2 HGNC:17302
## ENSG00000181072.11         protein_coding       CHRM2     2  HGNC:1951
## ENSG00000231943.9                  lncRNA  PGM5P4-AS1     2 HGNC:51195
## ENSG00000065325.13         protein_coding       GLP2R     2  HGNC:4325
## ENSG00000186642.16         protein_coding       PDE2A     2  HGNC:8777
## ENSG00000111452.13         protein_coding      ADGRD1     1 HGNC:19893
## ENSG00000268388.6                  lncRNA      FENDRR     2 HGNC:43894
## ENSG00000181234.9          protein_coding    TMEM132C     2 HGNC:25436
## ENSG00000173175.15         protein_coding       ADCY5     1   HGNC:236
## ENSG00000163431.13         protein_coding       LMOD1     2  HGNC:6647
## ENSG00000156218.13         protein_coding    ADAMTSL3     2 HGNC:14633
## ENSG00000176533.13         protein_coding        GNG7     2  HGNC:4410
## ENSG00000103241.7          protein_coding       FOXF1     1  HGNC:3809
## ENSG00000140538.16         protein_coding       NTRK3     1  HGNC:8033
## ENSG00000141338.14         protein_coding       ABCA8     2    HGNC:38
## ENSG00000121671.12         protein_coding        CRY2     2  HGNC:2385
## ENSG00000280429.1                     TEC  AF001548.3     2       <NA>
## ENSG00000149090.12         protein_coding       PAMR1     1 HGNC:24554
## ENSG00000070193.5          protein_coding       FGF10     2  HGNC:3666
## ENSG00000144655.15         protein_coding      CSRNP1     2 HGNC:14300
## ENSG00000118407.15         protein_coding      FILIP1     2 HGNC:21015
## ENSG00000149451.18         protein_coding      ADAM33     2 HGNC:15478
## ENSG00000254510.2                  lncRNA  AP001107.5     2       <NA>
## ENSG00000147588.7          protein_coding        PMP2     1  HGNC:9117
## ENSG00000166091.21         protein_coding       CMTM5     2 HGNC:19176
## ENSG00000108405.4          protein_coding       P2RX1     2  HGNC:8533
## ENSG00000154175.18         protein_coding      ABI3BP     2 HGNC:17265
## ENSG00000135472.9          protein_coding       FAIM2     2 HGNC:17067
## ENSG00000153234.15         protein_coding       NR4A2     1  HGNC:7981
## ENSG00000267505.1                  lncRNA  AC005180.2     2       <NA>
## ENSG00000174576.10         protein_coding       NPAS4     2 HGNC:18983
## ENSG00000100307.13         protein_coding        CBX7     2  HGNC:1557
## ENSG00000198932.13         protein_coding     GPRASP1     2 HGNC:24834
## ENSG00000154721.15         protein_coding        JAM2     2 HGNC:14686
## ENSG00000077157.22         protein_coding    PPP1R12B     1  HGNC:7619
## ENSG00000153823.19         protein_coding        PID1     2 HGNC:26084
## ENSG00000022267.19         protein_coding        FHL1     2  HGNC:3702
## ENSG00000154734.16         protein_coding     ADAMTS1     2   HGNC:217
## ENSG00000059915.17         protein_coding         PSD     2  HGNC:9507
## ENSG00000151892.15         protein_coding       GFRA1     2  HGNC:4243
## ENSG00000143171.13         protein_coding        RXRG     2 HGNC:10479
## ENSG00000132840.10         protein_coding       BHMT2     1  HGNC:1048
## ENSG00000133392.18         protein_coding       MYH11     1  HGNC:7569
## ENSG00000138356.14         protein_coding        AOX1     1   HGNC:553
## ENSG00000164736.6          protein_coding       SOX17     2 HGNC:18122
## ENSG00000241684.6                  lncRNA ADAMTS9-AS2     2 HGNC:42435
## ENSG00000108381.11         protein_coding        ASPA     2   HGNC:756
## ENSG00000188729.6          protein_coding        OSTN     2 HGNC:29961
## ENSG00000179796.12         protein_coding      LRRC3B     2 HGNC:28105
##                              havana_gene     logFC      AveExpr         t
## ENSG00000164530.15  OTTHUMG00000014611.4 -8.626179 -0.973925495 -27.03235
## ENSG00000168079.17  OTTHUMG00000132172.4 -6.963857 -0.156400551 -21.66294
## ENSG00000163815.6   OTTHUMG00000133087.3 -4.876303 -0.185071830 -21.27967
## ENSG00000153446.16  OTTHUMG00000159314.4 -6.216739 -1.183845033 -21.15637
## ENSG00000196616.14  OTTHUMG00000161413.4 -7.621807  0.001506268 -19.53526
## ENSG00000168309.18  OTTHUMG00000159159.5 -5.042092  1.304653118 -19.53313
## ENSG00000108924.14  OTTHUMG00000177840.3 -5.369232 -0.325488435 -19.45081
## ENSG00000018625.15  OTTHUMG00000024080.4 -6.564195 -0.544764663 -19.21638
## ENSG00000224958.6   OTTHUMG00000019962.6 -7.679657 -3.290932892 -18.73245
## ENSG00000197766.8   OTTHUMG00000181840.6 -4.677230  3.650846614 -18.33332
## ENSG00000126218.12  OTTHUMG00000017374.8 -4.503594 -0.459572869 -18.29138
## ENSG00000168477.19 OTTHUMG00000031088.12 -4.968998  2.168624352 -18.06128
## ENSG00000068976.14  OTTHUMG00000066835.3 -4.532689  0.072910209 -17.98268
## ENSG00000123560.14  OTTHUMG00000022111.6 -6.574817 -2.625770160 -17.79452
## ENSG00000034971.17  OTTHUMG00000034789.4 -6.498489 -3.685419186 -17.51631
## ENSG00000241158.7   OTTHUMG00000158723.8 -5.827560 -1.836861554 -17.23980
## ENSG00000168497.5   OTTHUMG00000154309.3 -4.320994  2.528618766 -17.23891
## ENSG00000004776.13  OTTHUMG00000048122.5 -5.729677  0.216725607 -17.09048
## ENSG00000077943.8   OTTHUMG00000017733.2 -3.896196  2.434404137 -17.05508
## ENSG00000154330.13  OTTHUMG00000019966.5 -5.913704  2.252661747 -16.88277
## ENSG00000106809.11  OTTHUMG00000020224.4 -6.636379 -0.812230294 -16.85679
## ENSG00000119147.10  OTTHUMG00000130921.4 -6.310495 -1.943277386 -16.82963
## ENSG00000181856.15  OTTHUMG00000102181.6 -4.952430 -0.317989379 -16.76719
## ENSG00000232855.7   OTTHUMG00000078747.6 -4.858875 -0.699000016 -16.66335
## ENSG00000144218.19 OTTHUMG00000153011.13 -4.740952 -0.040013910 -16.65630
## ENSG00000108018.15  OTTHUMG00000019018.5 -6.049695 -2.759462352 -16.60717
## ENSG00000167281.19 OTTHUMG00000150183.10 -5.956345 -1.410364551 -16.57846
## ENSG00000141052.18  OTTHUMG00000058767.6 -5.362071  0.519849699 -16.53639
## ENSG00000119508.18  OTTHUMG00000021030.3 -4.647134  2.389304230 -16.52689
## ENSG00000101605.13  OTTHUMG00000178209.7 -3.822224  0.920676137 -16.19462
## ENSG00000171368.12  OTTHUMG00000131011.5 -4.033155  1.445003094 -16.16513
## ENSG00000163145.13  OTTHUMG00000097095.5 -4.507519  0.037173958 -16.13653
## ENSG00000179915.24 OTTHUMG00000129263.23 -6.237905 -3.167047767 -16.10341
## ENSG00000125851.10  OTTHUMG00000031941.6 -6.466017 -3.797395189 -15.96300
## ENSG00000112936.19  OTTHUMG00000150340.4 -6.498061  1.176456881 -15.87290
## ENSG00000136546.16  OTTHUMG00000154078.6 -6.150101 -2.264897249 -15.81937
## ENSG00000182253.15  OTTHUMG00000171887.6 -5.240493  3.859004452 -15.68876
## ENSG00000205221.12  OTTHUMG00000152149.3 -6.532017 -2.932556957 -15.67676
## ENSG00000179388.9   OTTHUMG00000097825.3 -4.273992  2.204104102 -15.59435
## ENSG00000123358.20  OTTHUMG00000150393.8 -4.418795  5.478898911 -15.58365
## ENSG00000189129.14  OTTHUMG00000018596.3 -3.682754  1.303610464 -15.57245
## ENSG00000118526.7   OTTHUMG00000015608.2 -4.241260  0.026899594 -15.35148
## ENSG00000123243.15  OTTHUMG00000017635.7 -3.963833  3.690212567 -15.27199
## ENSG00000225398.3   OTTHUMG00000047819.3 -5.166785 -4.709966015 -15.22305
## ENSG00000268926.3   OTTHUMG00000187223.2 -5.045298 -4.527673373 -15.21993
## ENSG00000149294.17  OTTHUMG00000167196.8 -5.277470  0.742991061 -15.14274
## ENSG00000172403.11  OTTHUMG00000161165.7 -5.167767  4.440104709 -15.12016
## ENSG00000127528.6   OTTHUMG00000182330.1 -3.269929  3.564517862 -15.08928
## ENSG00000172348.15  OTTHUMG00000014782.2 -3.352853  3.007426971 -15.03846
## ENSG00000004799.8   OTTHUMG00000153977.3 -4.378008  3.725075760 -15.03381
## ENSG00000206579.9   OTTHUMG00000164288.3 -5.453541 -3.795037379 -15.01275
## ENSG00000172260.15  OTTHUMG00000009698.6 -4.657299  1.108165818 -14.89647
## ENSG00000181072.11  OTTHUMG00000155658.3 -6.340924 -1.789217412 -14.83018
## ENSG00000231943.9   OTTHUMG00000047830.7 -4.473422 -5.101604554 -14.80784
## ENSG00000065325.13 OTTHUMG00000130269.12 -5.409062 -1.980473354 -14.80492
## ENSG00000186642.16  OTTHUMG00000102045.6 -2.944570  2.336269499 -14.78984
## ENSG00000111452.13  OTTHUMG00000168339.5 -5.125893  0.262488107 -14.75977
## ENSG00000268388.6   OTTHUMG00000183870.4 -4.680187  1.822081042 -14.72162
## ENSG00000181234.9   OTTHUMG00000163736.4 -6.128057 -3.227731577 -14.68565
## ENSG00000173175.15  OTTHUMG00000159517.6 -4.323428  1.743721620 -14.59378
## ENSG00000163431.13  OTTHUMG00000035802.3 -4.933308  3.539199852 -14.58777
## ENSG00000156218.13  OTTHUMG00000147363.4 -4.725293 -0.338709187 -14.57113
## ENSG00000176533.13  OTTHUMG00000180435.3 -3.378803  1.326409088 -14.52900
## ENSG00000103241.7   OTTHUMG00000137651.5 -3.580024  3.015106923 -14.47241
## ENSG00000140538.16 OTTHUMG00000148677.12 -5.043862 -1.560919484 -14.46698
## ENSG00000141338.14  OTTHUMG00000180192.5 -4.556774  0.579598926 -14.39071
## ENSG00000121671.12  OTTHUMG00000153225.5 -1.676210  4.661172734 -14.33210
## ENSG00000280429.1   OTTHUMG00000177388.1 -4.553553 -4.868570328 -14.33071
## ENSG00000149090.12  OTTHUMG00000166328.4 -3.164357  2.669330699 -14.30631
## ENSG00000070193.5   OTTHUMG00000131153.5 -5.535710 -1.652197610 -14.27331
## ENSG00000144655.15  OTTHUMG00000131293.3 -2.758220  5.120375107 -14.25475
## ENSG00000118407.15  OTTHUMG00000015056.4 -3.770371  0.413755019 -14.22960
## ENSG00000149451.18  OTTHUMG00000031758.4 -4.176481  1.395455703 -14.22354
## ENSG00000254510.2   OTTHUMG00000166924.2 -5.031673 -2.432220083 -14.21994
## ENSG00000147588.7   OTTHUMG00000164600.2 -5.370378 -3.933873367 -14.16483
## ENSG00000166091.21  OTTHUMG00000028751.9 -4.581960 -4.037912126 -14.14605
## ENSG00000108405.4   OTTHUMG00000177673.2 -4.918751  0.625778571 -14.13839
## ENSG00000154175.18  OTTHUMG00000159094.7 -3.965052  2.526864714 -14.12700
## ENSG00000135472.9   OTTHUMG00000169808.5 -4.928698 -0.980590147 -14.09656
## ENSG00000153234.15 OTTHUMG00000131950.10 -3.128064  3.662202986 -14.07752
## ENSG00000267505.1   OTTHUMG00000179859.1 -5.260080 -2.230528864 -14.06555
## ENSG00000174576.10  OTTHUMG00000167045.2 -4.892362 -3.802365251 -14.04037
## ENSG00000100307.13  OTTHUMG00000150418.4 -2.436388  4.288751912 -14.03972
## ENSG00000198932.13  OTTHUMG00000022061.6 -2.906324  2.046697962 -14.00398
## ENSG00000154721.15  OTTHUMG00000078441.4 -3.080590  2.100043596 -13.98343
## ENSG00000077157.22  OTTHUMG00000041393.7 -3.376847  5.429915744 -13.97479
## ENSG00000153823.19  OTTHUMG00000133191.5 -3.702208  1.063116175 -13.94636
## ENSG00000022267.19 OTTHUMG00000022504.13 -4.719986  4.067977088 -13.94317
## ENSG00000154734.16  OTTHUMG00000078688.5 -3.801545  5.366583655 -13.91267
## ENSG00000059915.17  OTTHUMG00000018954.5 -4.016501  1.971407677 -13.90296
## ENSG00000151892.15  OTTHUMG00000019097.4 -4.956486 -0.231853445 -13.85131
## ENSG00000143171.13  OTTHUMG00000034626.5 -4.934314 -3.815693813 -13.82263
## ENSG00000132840.10  OTTHUMG00000108158.5 -4.363937 -0.427305459 -13.78691
## ENSG00000133392.18  OTTHUMG00000129935.9 -6.303887  6.131469203 -13.78683
## ENSG00000138356.14  OTTHUMG00000154536.6 -4.729822  0.121843854 -13.75385
## ENSG00000164736.6   OTTHUMG00000164377.3 -3.004320  0.926057591 -13.75308
## ENSG00000241684.6   OTTHUMG00000158725.3 -4.521205 -2.370643238 -13.71818
## ENSG00000108381.11  OTTHUMG00000090655.5 -4.654585 -0.766601359 -13.69808
## ENSG00000188729.6   OTTHUMG00000156190.2 -4.919253 -5.006323614 -13.66123
## ENSG00000179796.12  OTTHUMG00000130572.8 -5.309622 -4.036877767 -13.65030
##                          P.Value    adj.P.Val         B
## ENSG00000164530.15 4.954238e-95 1.391497e-90 205.92819
## ENSG00000168079.17 4.725979e-71 6.636929e-67 151.09498
## ENSG00000163815.6  2.570364e-69 2.406460e-65 146.93834
## ENSG00000153446.16 9.299827e-69 6.530106e-65 145.69469
## ENSG00000196616.14 2.043719e-61 9.781097e-58 129.08962
## ENSG00000168309.18 2.089457e-61 9.781097e-58 129.04287
## ENSG00000108924.14 4.924192e-61 1.975797e-57 128.11082
## ENSG00000018625.15 5.649209e-60 1.983367e-56 125.75012
## ENSG00000224958.6  8.633305e-58 2.694263e-54 120.61735
## ENSG00000197766.8  5.413403e-56 1.520463e-52 116.68510
## ENSG00000126218.12 8.357028e-56 2.133853e-52 116.07362
## ENSG00000168477.19 9.032864e-55 2.114217e-51 113.88867
## ENSG00000068976.14 2.034802e-54 4.396268e-51 112.99480
## ENSG00000123560.14 1.419089e-53 2.846997e-50 110.96054
## ENSG00000034971.17 2.492947e-52 4.667959e-49 107.93586
## ENSG00000241158.7  4.271769e-51 7.122271e-48 105.35347
## ENSG00000168497.5  4.310841e-51 7.122271e-48 105.47318
## ENSG00000004776.13 1.974452e-50 3.080912e-47 103.95234
## ENSG00000077943.8  2.837509e-50 4.194585e-47 103.60044
## ENSG00000154330.13 1.653731e-49 2.322417e-46 101.84152
## ENSG00000106809.11 2.156537e-49 2.884317e-46 101.57511
## ENSG00000119147.10 2.846298e-49 3.633817e-46 101.23272
## ENSG00000181856.15 5.384321e-49 6.575192e-46 100.63314
## ENSG00000232855.7  1.552573e-48 1.816963e-45  99.55047
## ENSG00000144218.19 1.668366e-48 1.874376e-45  99.51620
## ENSG00000108018.15 2.752143e-48 2.973056e-45  98.84869
## ENSG00000167281.19 3.686560e-48 3.834979e-45  98.71913
## ENSG00000141052.18 5.657282e-48 5.674860e-45  98.33953
## ENSG00000119508.18 6.231064e-48 6.034893e-45  98.24536
## ENSG00000101605.13 1.817198e-46 1.701322e-43  94.87918
## ENSG00000171368.12 2.449670e-46 2.219480e-43  94.59902
## ENSG00000163145.13 3.271758e-46 2.871684e-43  94.27832
## ENSG00000179915.24 4.573869e-46 3.892917e-43  93.78166
## ENSG00000125851.10 1.889414e-45 1.560823e-42  92.31794
## ENSG00000112936.19 4.687207e-45 3.761417e-42  91.66320
## ENSG00000136546.16 8.037414e-45 6.270746e-42  91.05686
## ENSG00000182253.15 2.989435e-44 2.269304e-41  89.77912
## ENSG00000205221.12 3.372401e-44 2.492648e-41  89.61079
## ENSG00000179388.9  7.711893e-44 5.553947e-41  88.89281
## ENSG00000123358.20 8.585490e-44 6.028516e-41  88.70233
## ENSG00000189129.14 9.605668e-44 6.580351e-41  88.67102
## ENSG00000118526.7  8.764576e-43 5.861206e-40  86.45335
## ENSG00000123243.15 1.937467e-42 1.265526e-39  85.66526
## ENSG00000225398.3  3.155168e-42 2.014073e-39  84.71324
## ENSG00000268926.3  3.254971e-42 2.031608e-39  84.68921
## ENSG00000149294.17 7.017736e-42 4.284938e-39  84.42183
## ENSG00000172403.11 8.784264e-42 5.249439e-39  84.11056
## ENSG00000127528.6  1.193926e-41 6.986207e-39  83.87734
## ENSG00000172348.15 1.977768e-41 1.133665e-38  83.38507
## ENSG00000004799.8  2.070994e-41 1.163360e-38  83.30143
## ENSG00000206579.9  2.552472e-41 1.405711e-38  82.83539
## ENSG00000172260.15 8.077351e-41 4.362857e-38  82.00000
## ENSG00000181072.11 1.555839e-40 8.245068e-38  81.34010
## ENSG00000231943.9  1.940168e-40 1.009139e-37  80.56397
## ENSG00000065325.13 1.996968e-40 1.019797e-37  81.03533
## ENSG00000186642.16 2.317581e-40 1.162391e-37  80.95658
## ENSG00000111452.13 3.118328e-40 1.536570e-37  80.66292
## ENSG00000268388.6  4.542849e-40 2.199914e-37  80.27684
## ENSG00000181234.9  6.475745e-40 3.082784e-37  79.79977
## ENSG00000173175.15 1.599248e-39 7.486345e-37  79.03470
## ENSG00000163431.13 1.696449e-39 7.811173e-37  78.91497
## ENSG00000156218.13 1.997857e-39 9.050614e-37  78.81319
## ENSG00000176533.13 3.021808e-39 1.347199e-36  78.41120
## ENSG00000103241.7  5.264474e-39 2.310364e-36  77.83902
## ENSG00000140538.16 5.552112e-39 2.399110e-36  77.74565
## ENSG00000141338.14 1.171808e-38 4.986752e-36  77.07029
## ENSG00000121671.12 2.078232e-38 8.701752e-36  76.47474
## ENSG00000280429.1  2.106737e-38 8.701752e-36  76.02272
## ENSG00000149090.12 2.673487e-38 1.088264e-35  76.24306
## ENSG00000070193.5  3.689708e-38 1.480469e-35  75.90937
## ENSG00000144655.15 4.421820e-38 1.749235e-35  75.68220
## ENSG00000118407.15 5.650606e-38 2.204286e-35  75.50582
## ENSG00000149451.18 5.994038e-38 2.306227e-35  75.44901
## ENSG00000254510.2  6.207831e-38 2.356207e-35  75.30219
## ENSG00000147588.7  1.061682e-37 3.975930e-35  74.63470
## ENSG00000166091.21 1.274435e-37 4.709875e-35  74.35592
## ENSG00000108405.4  1.373071e-37 5.008501e-35  74.62851
## ENSG00000154175.18 1.533884e-37 5.523357e-35  74.49682
## ENSG00000135472.9  2.061676e-37 7.329909e-35  74.21131
## ENSG00000153234.15 2.480501e-37 8.708728e-35  74.00818
## ENSG00000267505.1  2.785952e-37 9.660374e-35  73.86633
## ENSG00000174576.10 3.557135e-37 1.211259e-34  73.41505
## ENSG00000100307.13 3.579396e-37 1.211259e-34  73.64058
## ENSG00000198932.13 5.061557e-37 1.692428e-34  73.33977
## ENSG00000154721.15 6.176324e-37 2.040875e-34  73.14104
## ENSG00000077157.22 6.715316e-37 2.193175e-34  72.94676
## ENSG00000153823.19 8.842500e-37 2.854705e-34  72.78923
## ENSG00000022267.19 9.119307e-37 2.910613e-34  72.65367
## ENSG00000154734.16 1.224781e-36 3.865217e-34  72.34100
## ENSG00000059915.17 1.345380e-36 4.198633e-34  72.35488
## ENSG00000151892.15 2.215282e-36 6.837432e-34  71.87998
## ENSG00000143171.13 2.921406e-36 8.918863e-34  71.35281
## ENSG00000132840.10 4.122091e-36 1.232580e-33  71.25646
## ENSG00000133392.18 4.125130e-36 1.232580e-33  71.02944
## ENSG00000138356.14 5.667234e-36 1.670438e-33  70.95026
## ENSG00000164736.6  5.709477e-36 1.670438e-33  70.93414
## ENSG00000241684.6  7.986913e-36 2.312664e-33  70.48350
## ENSG00000108381.11 9.689537e-36 2.777041e-33  70.40586
## ENSG00000188729.6  1.380413e-35 3.916330e-33  69.71280
## ENSG00000179796.12 1.533106e-35 4.306036e-33  69.73624
```

Step 3 - Visualize

```
# make a function to generate a scatter plot to show a separation of tumor vs normal points
plot_PCA = function(voomObj, condition_variable){
  # create a factor
  group = factor(voomObj$targets[, condition_variable])
  # perform a principal component analysis
  pca = prcomp(t(voomObj$E))
  # Take PC1 and PC2 for the plot
  plot(pca$x[,1:2],col=group, pch=19)
  # include a legend for points
  legend("bottomleft", inset=.01, levels(group), pch=19, col=1:length(levels(group)))
  return(pca)
}

# call the plot function with the voom object and the defintion column
res_pca = plot_PCA(limma_res$voomObj, "definition")
```



Step 4 - Classification model training, testing, and evaluation

```
# TODO need to redo this whole step using WGCNA
# use the expression data that has been normalized
# Transpose and make it into a matrix object
d_mat = as.matrix(t(limma_res$voomObj$E))

# and the clinical feature to distinguish cases ("definition")
# Make it a factor
d_resp = as.factor(limma_res$voomObj$targets$definition)

# Divide data into training and testing set
# 75% of samples for training and 25% for testing

# Set (random-number-generator) seed so that results are consistent between runs
set.seed(42)

# create a vector of booleans to subset the cases
train_ids = createDataPartition(d_resp, p=0.75, list=FALSE)

# x is the matrix with normalized expression data
# y is the vector with the response variable (tumor vs normal)
x_train = d_mat[train_ids, ]
x_test  = d_mat[-train_ids, ]

y_train = d_resp[train_ids]
y_test  = d_resp[-train_ids]

# do an elastic net model - a generalized linear model that
#  combines lasso and ridge regression, it selects the genes or groups of genes
#  that best predict the condition and uses these to build the model
#  that is then used for classification

# Train model on training dataset using cross-validation
#  alpha can be between 0 (ridge regression) and 1 (lasso)
# the res object here is an object that holds the model coeffiecients and the
#  mean error found during training
res = cv.glmnet(
  x = x_train,
  y = y_train,
  alpha = 0.5,
  family = "binomial")

# Test/Make prediction on test dataset
y_pred = predict(res, newx=x_test, type="class", s="lambda.min")

# confusion matrix shows the TP, TN, FP, and FN
confusion_matrix = table(y_pred, y_test)

# Evaluation statistics
print(confusion_matrix)
```

```
##                       y_test
## y_pred                Primary solid Tumor Solid Tissue Normal
##   Primary solid Tumor                 103                   1
##   Solid Tissue Normal                   0                   3
```

```
print(paste0("Sensitivity: ",sensitivity(confusion_matrix)))
```

```
## [1] "Sensitivity: 1"
```

```
print(paste0("Specificity: ",specificity(confusion_matrix)))
```

```
## [1] "Specificity: 0.75"
```

```
print(paste0("Precision: ",precision(confusion_matrix)))
```

```
## [1] "Precision: 0.990384615384615"
```

```
# now we can look at the genes that most contribute for the prediction
res_coef = coef(res, s="lambda.min") # the "coef" function returns a sparse matrix

# ignore zero value coefficients
res_coef = res_coef[res_coef[,1] != 0,]

# remove first coefficient as this is the intercept, a variable of the model itself
res_coef = res_coef[-1]

relevant_genes = names(res_coef) # get names of the (non-zero) variables.
length(relevant_genes) # number of selected genes
```

```
## [1] 83
```

```
# get the Ensembl gene names
head(relevant_genes) # few select genes
```

```
## [1] "ENSG00000034971.17" "ENSG00000078804.13" "ENSG00000081181.8"
## [4] "ENSG00000086991.13" "ENSG00000101057.16" "ENSG00000102683.8"
```

```
# get the common gene names
# TODO fix this
head(limma_res$voomObj$genes)
```

```
##                    source type score phase            gene_id      gene_type
## ENSG00000000003.15 HAVANA gene    NA    NA ENSG00000000003.15 protein_coding
## ENSG00000000005.6  HAVANA gene    NA    NA  ENSG00000000005.6 protein_coding
## ENSG00000000419.13 HAVANA gene    NA    NA ENSG00000000419.13 protein_coding
## ENSG00000000457.14 HAVANA gene    NA    NA ENSG00000000457.14 protein_coding
## ENSG00000000460.17 HAVANA gene    NA    NA ENSG00000000460.17 protein_coding
## ENSG00000000938.13 HAVANA gene    NA    NA ENSG00000000938.13 protein_coding
##                    gene_name level    hgnc_id          havana_gene
## ENSG00000000003.15    TSPAN6     2 HGNC:11858 OTTHUMG00000022002.2
## ENSG00000000005.6       TNMD     2 HGNC:17757 OTTHUMG00000022001.2
## ENSG00000000419.13      DPM1     2  HGNC:3005 OTTHUMG00000032742.2
## ENSG00000000457.14     SCYL3     2 HGNC:19285 OTTHUMG00000035941.6
## ENSG00000000460.17  C1orf112     2 HGNC:25565 OTTHUMG00000035821.9
## ENSG00000000938.13       FGR     2  HGNC:3697 OTTHUMG00000003516.3
```

```
relevant_gene_names = limma_res$voomObj$genes[relevant_genes,"external_gene_name"]
head(relevant_gene_names) # few select genes (with readable names now)
```

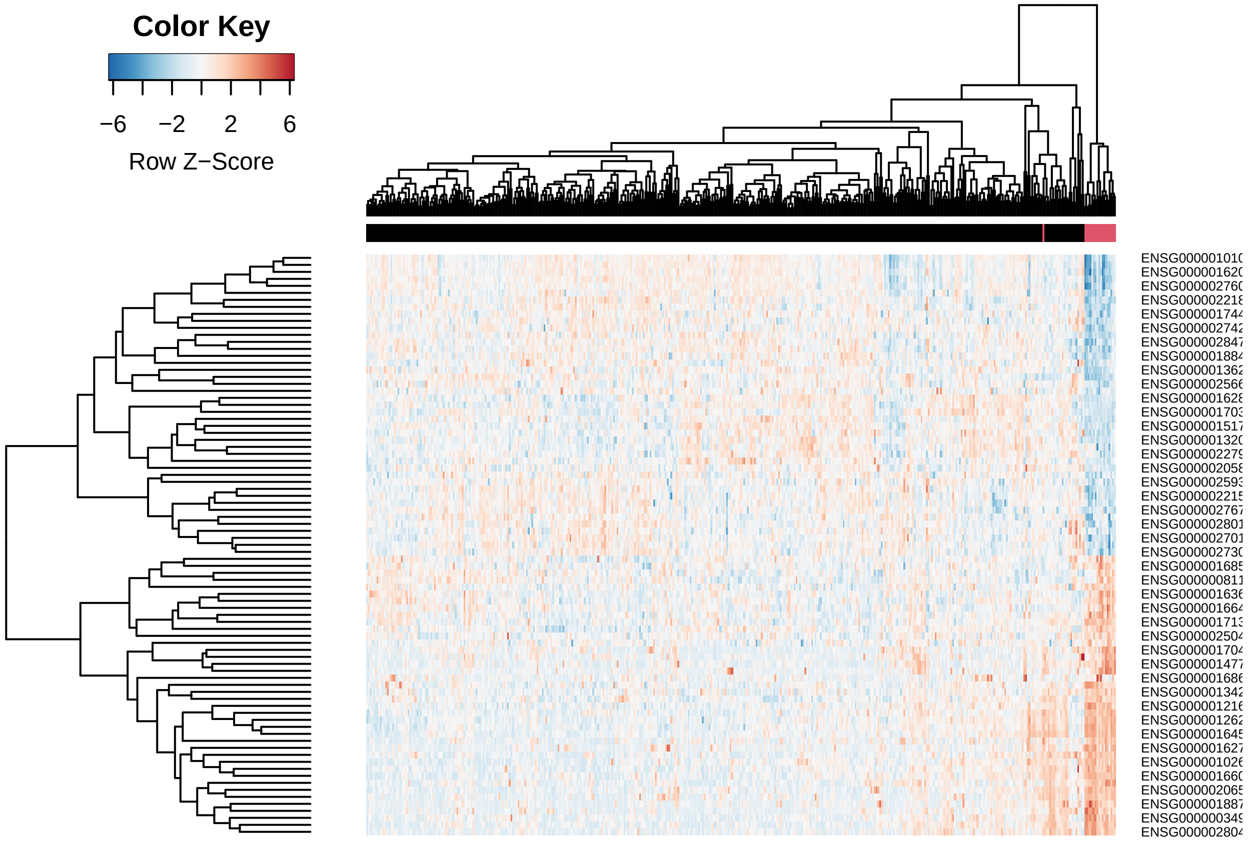
```
## NULL
```

```
# did elastic net find the same genes originally found by the limma pipeline?
#  "Of note, we do not expect a high overlap between genes selected by limma and Elastic net.
#   The reason for this is the fact Elastic Net criteria bias the selection of genes,
#   which are not highly correlated against each other, while not such bias is
#   present in limma."
print(intersect(limma_res$topGenes$ensembl_gene_id, relevant_genes))
```

```
## NULL
```

Step 5 - Hierarchical clustering

```
# we are only considering the elastic net results to cluster genes together
# genes in green are original limma results
# genes in red are normal tissue from the elastic net results
# genes in black are tumor tissue from the elastic net results

 # define the color palette for the plot
hmcol = colorRampPalette(rev(brewer.pal(9, "RdBu")))(256)

# perform complete linkage clustering
clust = function(x) hclust(x, method="complete")
# use the inverse of correlation as distance.
dist = function(x) as.dist((1-cor(t(x)))/2)

# Show green color for genes that also show up in DE analysis
colorLimmaGenes = ifelse(
  # Given a vector of boolean values
  (relevant_genes %in% limma_res$topGenes$ensembl_gene_id),
  "green", # if true, return green for that value
  "white" # if false, return white for that value
)

# As you've seen a good looking heatmap involves a lot of parameters
gene_heatmap = heatmap.2(
  t(d_mat[,relevant_genes]),
  scale="row",          # scale the values for each gene (row)
  density.info="none",  # turns off density plot inside color legend
  trace="none",         # turns off trace lines inside the heat map
  col=hmcol,            # define the color map
  labRow=relevant_gene_names, # use gene names instead of ensembl annotation
  RowSideColors=colorLimmaGenes,
  labCol=FALSE,         # Not showing column labels
  ColSideColors=as.character(as.numeric(d_resp)), # Show colors for each response class
  dendrogram="both",    # Show dendrograms for both axis
  hclust = clust,       # Define hierarchical clustering method
  distfun = dist,          # Using correlation coefficient for distance function
  cexRow=.6,               # Resize row labels
  margins=c(1,5)           # Define margin spaces
)
```


Color Key
Row Z−Score
ENSG000001010
ENSG000001620
ENSG000002760
ENSG000002218
ENSG000001744
ENSG000002742
ENSG000002847
ENSG000001884
ENSG000001362
ENSG000002566
ENSG000001628
ENSG000001703
ENSG000001517
ENSG000001320
ENSG000002279
ENSG000002058
ENSG000002593
ENSG000002215
ENSG000002767
ENSG000002801
ENSG000002701
ENSG000002730
ENSG000001685
ENSG000000811
ENSG000001636
ENSG000001664
ENSG000001713
ENSG000002504
ENSG000001704
ENSG000001477
ENSG000001686
ENSG000001342
ENSG000001216
ENSG000001262
ENSG000001645
ENSG000001627
ENSG000001026
ENSG000001660
ENSG000002065
ENSG000001887
ENSG000000349
ENSG000002804