dge_18082023.R

hp

2023-08-18

```
#DGE analysis for SB-positive and negative-cell lines
BiocManager::install('tximport')
```

```
## Bioconductor version 3.16 (BiocManager 1.30.22), R 4.2.3 (2023-03-15 ucrt)

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

## Installation paths not writeable, unable to update packages
##   path: C:/Program Files/R/R-4.2.3/library
##   packages:
##     class, KernSmooth, lattice, MASS, Matrix, mgcv, nlme, nnet, spatial,
##     survival

## Old packages: 'Rcpp', 'rlang'
```

```
library("tximport")
library("RColorBrewer")
library("DESeq2")
```

```
## Loading required package: S4Vectors

## Loading required package: stats4

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

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

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

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

## 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: GenomicRanges

## Loading required package: GenomeInfoDb

## Loading required package: 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: 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
```

```
library("readr")
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
```

```
library("ggrepel")
```

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

```
library("ggplot2")
library("circlize")
```

```
## ========================================
## circlize version 0.4.15
## CRAN page: https://cran.r-project.org/package=circlize
## Github page: https://github.com/jokergoo/circlize
## Documentation: https://jokergoo.github.io/circlize_book/book/
##
## If you use it in published research, please cite:
## Gu, Z. circlize implements and enhances circular visualization
```

```
##   in R. Bioinformatics 2014.
##
## This message can be suppressed by:
##   suppressPackageStartupMessages(library(circlize))
## ========================================
```

```
library("ComplexHeatmap")
```

```
## Loading required package: grid
```

```
## ========================================
## ComplexHeatmap version 2.14.0
## Bioconductor page: http://bioconductor.org/packages/ComplexHeatmap/
## Github page: https://github.com/jokergoo/ComplexHeatmap
## Documentation: http://jokergoo.github.io/ComplexHeatmap-reference
##
## If you use it in published research, please cite either one:
## - Gu, Z. Complex Heatmap Visualization. iMeta 2022.
## - Gu, Z. Complex heatmaps reveal patterns and correlations in multidimensional
##     genomic data. Bioinformatics 2016.
##
##
## The new InteractiveComplexHeatmap package can directly export static
## complex heatmaps into an interactive Shiny app with zero effort. Have a try!
##
## This message can be suppressed by:
##   suppressPackageStartupMessages(library(ComplexHeatmap))
## ========================================
```

```
setwd("E:/MSR BLY/lab work/202190 Autophagy")
#read model information
model<- read.csv("Model.csv")
profiles <- read.csv("OmicsProfiles.csv")
profiles_rna <-profiles[which(profiles$Datatype=="rna"),]

#make a cell line and results table
sampleDat <- data.frame(sample=c("A549", "AGS", "CACO2", "DU145", "HCT116",
                                 "HEK293T", "HT29", "HUH7", "MCF10A",
                                 "MCF7", "PC3", "SHSY5Y", "SW480", "WM1617",
                                 "WM793", "NCIH460", "MDAMB231", "U87MG",
                                 "HACAT", "LN229"),
                        status=as.factor(c("pos", "pos", "pos", "pos", "pos", "neg",
                                 "pos", "neg", "pos", "neg", "neg", "neg", "pos",
                                 "neg", "neg", "pos", "pos", "pos", "neg", "pos")))

#read expected counts
expected_counts <- read_csv("OmicsExpressionGenesExpectedCountProfile.csv")
```

```
## New names:
## * `` -> `...1`
```

```
## Rows: 1466 Columns: 54344
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr     (1): ...1
## dbl (54343): TSPAN6 (ENSG00000000003), TNMD (ENSG00000000005), DPM1 (ENSG000...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
rsem<- as.data.frame(expected_counts)

#matching cell line names
model_idx <- profiles$ModelID[match(rsem[, 1], profiles$ProfileID)]
rsem$cell_line<- model$StrippedCellLineName[match(model_idx, model$ModelID)]

#match cell lines
sampleDat$idxs <- match(sampleDat$sample, rsem$cell_line)
sampleDat1 <- na.omit(sampleDat)

rsem_sb90<- t.data.frame(rsem[sampleDat1$idxs,])
colnames(rsem_sb90)<-rsem_sb90[54345,]

#remove the profileID and cell line name and then round off the data
#removing genes if they have zero counts in 5 or more samples
rsem_sb90<- rsem_sb90[-c(1, 54345),]
rsem_sb90R <- apply(rsem_sb90, 2, function(x){
  round(as.numeric(x), digits=0)
})

genes_keep<- sapply(1:nrow(rsem_sb90R), function(i){
  if (length(which(rsem_sb90R[i,]==0))<10){
    return(i)
  } else
  return(NA)
})

rsem_sb90R<- cbind.data.frame(rownames(rsem_sb90), rsem_sb90R)
rsem_sb90R<-rsem_sb90R[na.omit(genes_keep),]

#DESeq2
dds <- DESeqDataSetFromMatrix(countData=rsem_sb90R,
                              colData=sampleDat1[,1:2],
                              design=~status, tidy = TRUE)
```

```
## converting counts to integer mode
```

```
dds<- DESeq(dds)
```

```
## estimating size factors
## estimating dispersions
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
## fitting model and testing
```

```
## -- replacing outliers and refitting for 2882 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)
## estimating dispersions
## fitting model and testing
```

```
#results
res <- results(dds)
head(results(dds, tidy=TRUE))
```

```
##                          row    baseMean log2FoldChange     lfcSE       stat
## 1   TSPAN6 (ENSG00000000003) 3095.157757      0.2253646 0.5692665  0.3958859
## 2     DPM1 (ENSG00000000419) 5541.449721      0.2753800 0.3762919  0.7318254
## 3    SCYL3 (ENSG00000000457)  850.887066     -0.2453928 0.3080977 -0.7964771
## 4 C1orf112 (ENSG00000000460) 1737.409192      0.1853798 0.3659587  0.5065595
## 5      FGR (ENSG00000000938)    1.425909     -0.1313007 1.1063573 -0.1186783
## 6      CFH (ENSG00000000971) 3220.913764     -2.9707086 1.6726522 -1.7760468
##      pvalue      padj
## 1 0.6921892 0.9560545
## 2 0.4642751 0.9111463
## 3 0.4257548 0.8997988
## 4 0.6124639 0.9448242
## 5 0.9055302 0.9915056
## 6        NA        NA
```

```
#summary
summary(res)
```

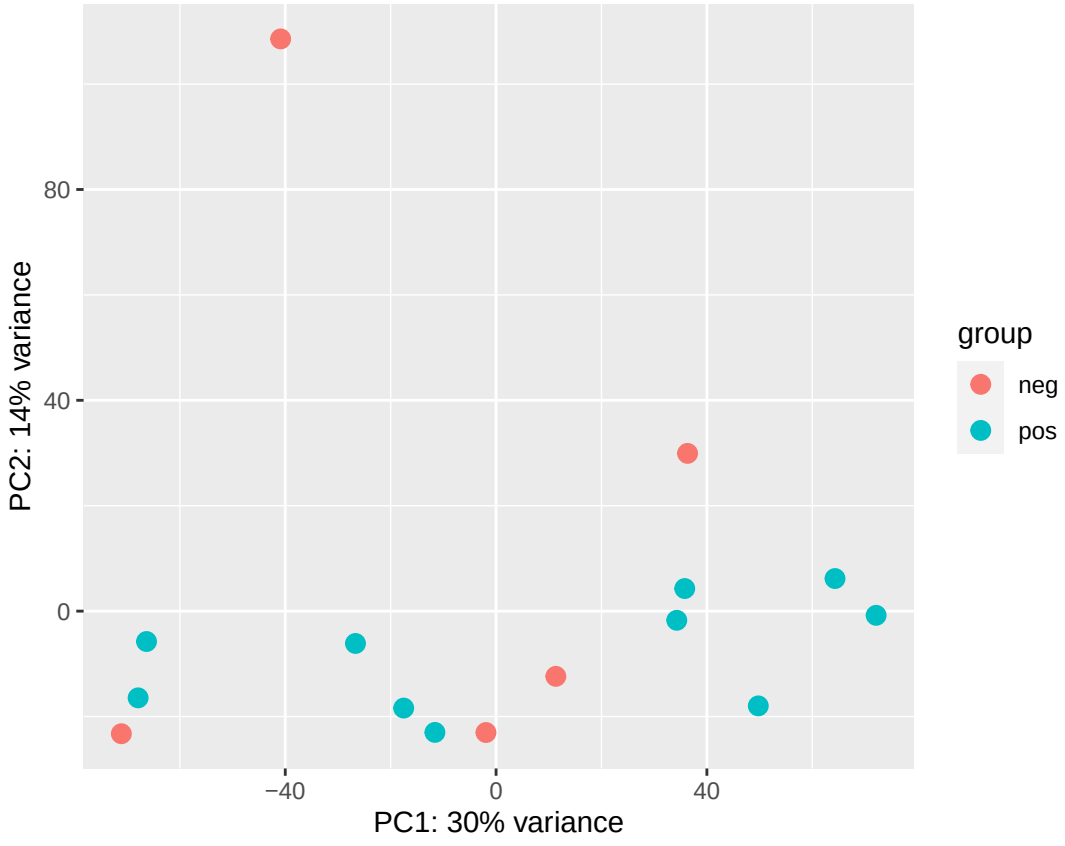
```
##
## out of 29234 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 153, 0.52%
## LFC < 0 (down)     : 337, 1.2%
## outliers [1]       : 1931, 6.6%
## low counts [2]     : 1701, 5.8%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
res <- res[order(res$padj),]
head(res)
```

```
## log2 fold change (MLE): status pos vs neg
## Wald test p-value: status pos vs neg
## DataFrame with 6 rows and 6 columns
##                            baseMean log2FoldChange     lfcSE      stat
##                           <numeric>      <numeric> <numeric> <numeric>
## AEBP1 (ENSG00000106624)    379.2162       -6.75786  0.911766  -7.41184
## UBE2QL1 (ENSG00000215218)  218.2916       -7.20511  1.025729  -7.02438
## RELN (ENSG00000189056)    2153.8840       -7.94102  1.211411  -6.55518
## DDX43 (ENSG00000080007)     86.6075       -7.62860  1.198074  -6.36739
## LAMP5 (ENSG00000125869)    225.0429       -8.85721  1.429637  -6.19542
```

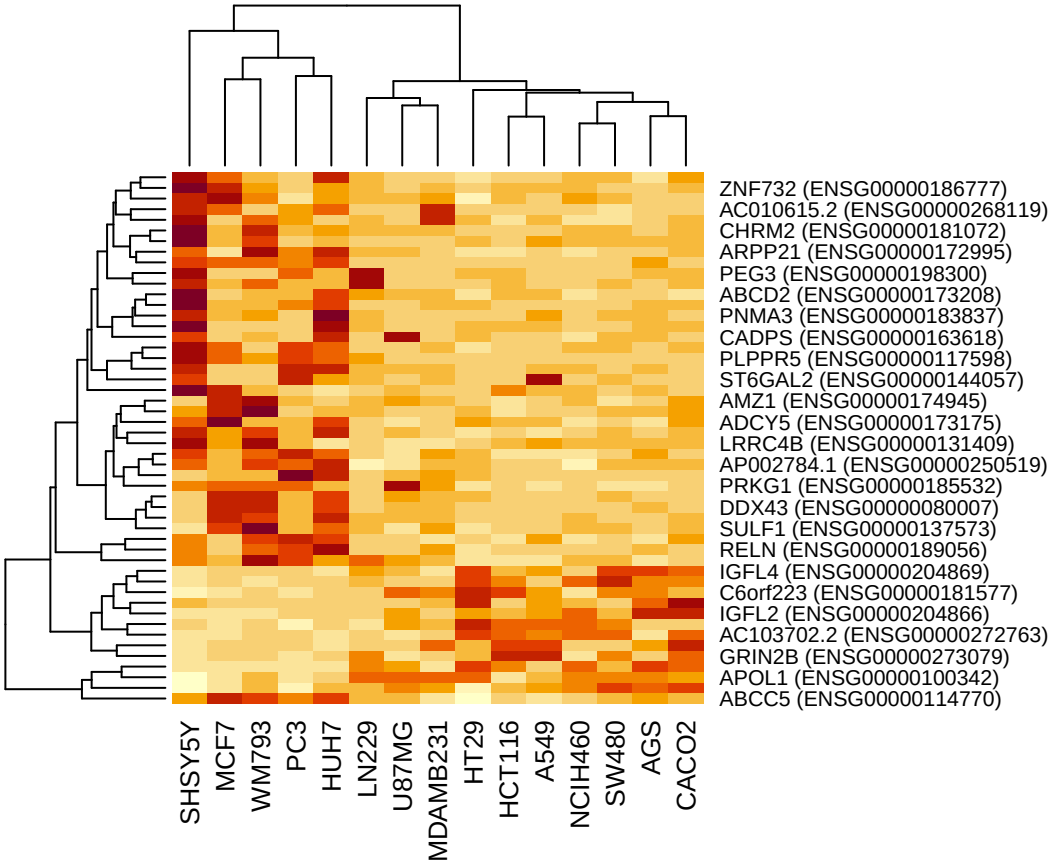
```
## INHBA (ENSG00000122641)   5280.6457       -7.32380  1.191064  -6.14896
##                                pvalue        padj
##                             <numeric>   <numeric>
## AEBP1 (ENSG00000106624)   1.24560e-13 3.18898e-09
## UBE2QL1 (ENSG00000215218) 2.15020e-12 2.75247e-08
## RELN (ENSG00000189056)    5.55746e-11 4.74274e-07
## DDX43 (ENSG00000080007)   1.92272e-10 1.23064e-06
## LAMP5 (ENSG00000125869)   5.81285e-10 2.97641e-06
## INHBA (ENSG00000122641)   7.79939e-10 3.32800e-06
```

```
write.table(res, file="18082023_dge_sb90_filter=10.txt", sep="\t")

#vst
vsdata <- vst(dds, blind=FALSE)
plotPCA(vsdata, intgroup="status")
```

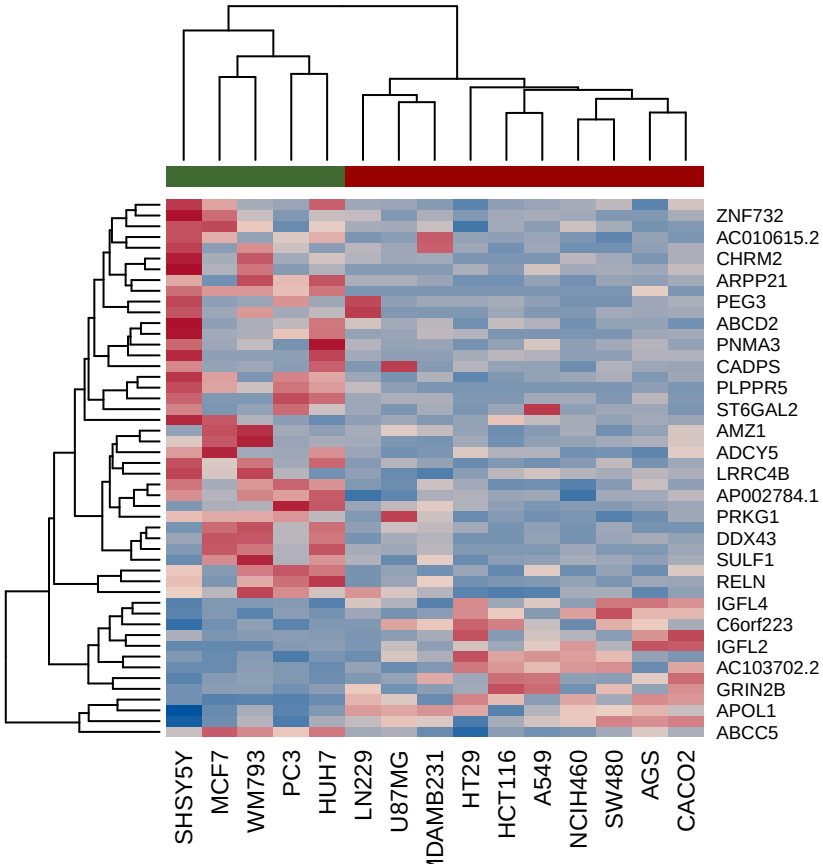



```
genes<- sapply(res@rownames, function(x){strsplit(x, split=" ")[[1]][1]})
write.table(genes, file="18072023_genes.txt", row.names = FALSE)

vst_matrix<- vsdata@assays@data@listData[[1]]

genes_pos<- vector()
all_genes<- c(res@rownames[1:50])

for (gene in all_genes){
```

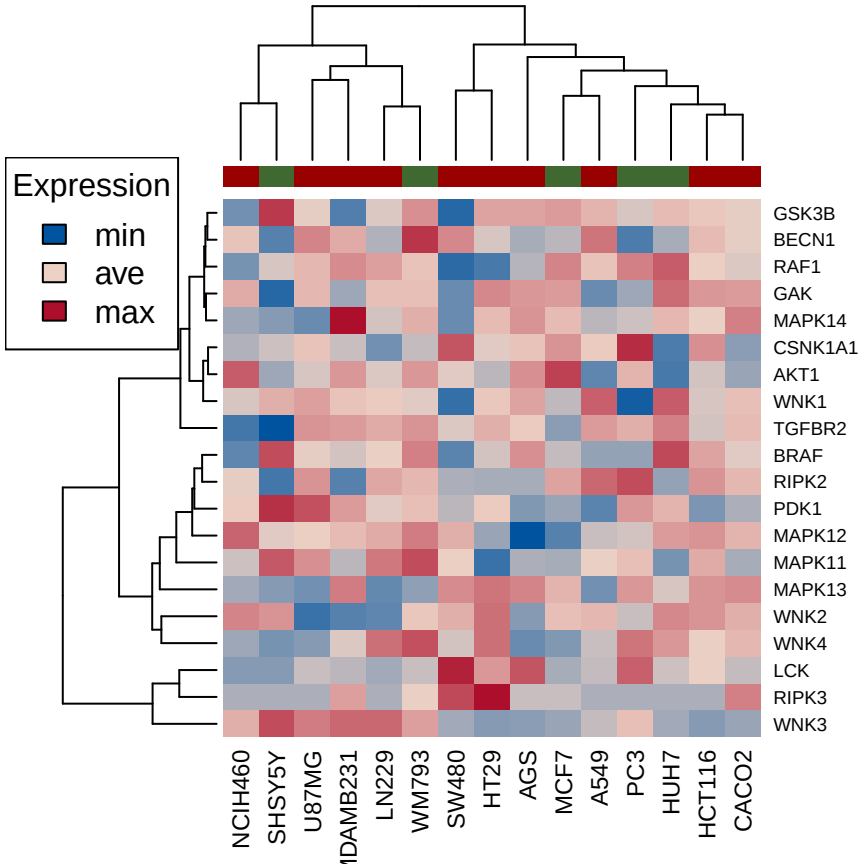
```
  pos<-which(rownames(vst_matrix)==gene)
  genes_pos <- c(genes_pos, pos)
}
genes_matrix<- vst_matrix[genes_pos,]

heatmap(genes_matrix[1:50,])
```




```
#colors for heatmap
col_fun = colorRamp2(c(2.5, 10, 15), c("#00549f", "#edd1c5", "#ac0e2b"))
col_vac<- sapply(sampleDat1$status, function(x){
  if(x=="pos"){status="#990000"}
  else if(x=="neg"){status="#406931"}
})

col_heatmap <- c("#00549f", "#edd1c5", "#ac0e2b")
pal <- colorRampPalette(col_heatmap)(100)
rownames(genes_matrix) <- sapply(rownames(genes_matrix), function(x){unlist(strsplit(x, split=" \\("))[
heatmap(genes_matrix[1:50,], col=pal, ColSideColors=col_vac, cexRow = 0.9, cexCol = 1.15)
```




```
targets <- c("MAPK11", "MAPK14", "MAPK13", "MAPK12", "WNK1", "WNK2", "WNK3", "WNK4",
             "RIPK2", "GAK", "BRAF", "RAF1", "GSK3B", "LCK", "PDK1", "TGFBR2", "CSNK1A1",
             "RIPK3", "AKT1", "BECN1")

rownames(vst_matrix)<- sapply(rownames(vst_matrix), function(x){unlist(strsplit(x, split=" \\("))[1]})
heatmap(vst_matrix[match(targets, rownames(vst_matrix)),], col=pal, ColSideColors=col_vac, cexRow = 0.9
legend(x="topleft", legend=c("min", "ave", "max"), fill=colorRampPalette(col_heatmap)(3), trace=TRUE, t
```

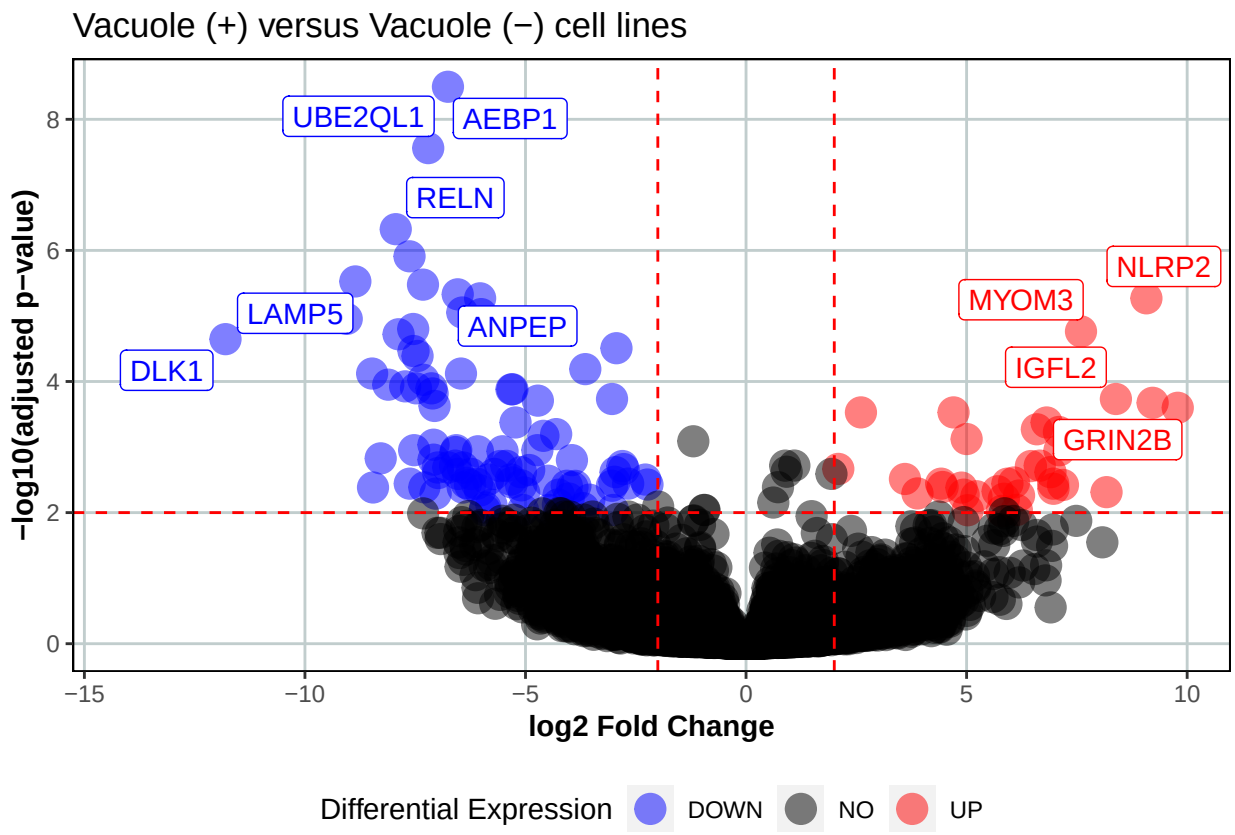



```
##   xchar= 0.02852,0.02852,0.02852 ; (yextra, ychar)= 0,0,0, 0.07519,0.07519,0.07519
##   rect2(0,1, w=0.171, h=0.3759, ...)
```

```
#a better volcano plot!
de <- read.delim("18082023_dge_sb90_filter=10.txt", row.names = 1)
de$gene<-sapply(rownames(de), function(x){unlist(strsplit(x, split=" \\("))[1]})
de$diffexpressed <- "NO"
# if log2Foldchange > 2 and pvalue < 0.05, set as "UP"
de$diffexpressed[de$log2FoldChange > 2 & de$padj < 0.01] <- "UP"
# if log2Foldchange < -2 and pvalue < 0.01, set as "DOWN"
de$diffexpressed[de$log2FoldChange < -2 & de$padj < 0.01] <- "DOWN"
de$delabel <- NA
de$delabel[de$diffexpressed != "NO"] <- de$gene[which(de$diffexpressed != "NO")]
de_labels <- match(c("AEBP1", "RELN", "ANPEP", "LAMP5", "DLK1", "UBE2QL1",
                     "IGFL2", "GRIN2B", "MYOM3", "NLRP2"), de$delabel)

#adding colors for DEGs
mycolors <- c("blue", "red", "black")
names(mycolors) <- c("DOWN", "UP", "NO")
de_lab<- de[de_labels,]
ggplot(data=de, aes(x=log2FoldChange, y=-log10(padj),
                    col=diffexpressed)) +
  geom_point(alpha=0.5, size=5)+
  geom_vline(xintercept=c(-2, 2), col="red", linetype=2) +
  geom_hline(yintercept=-log10(0.01), col="red", linetype=2)+
```

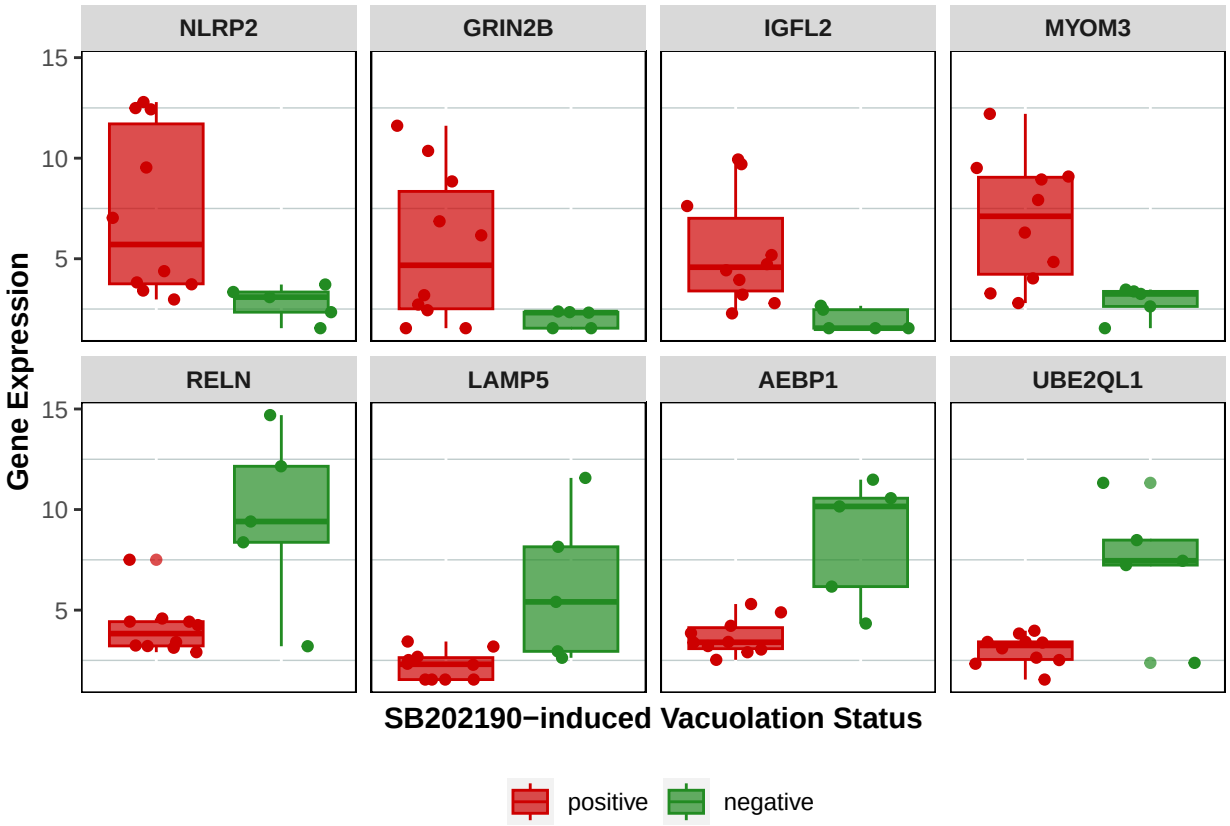
```
  theme(plot.background = element_blank(),
        axis.title = element_text(face="bold"),
        panel.background = element_blank(),
        panel.border = element_rect(fill=NA),
        legend.position = "bottom",
        panel.grid.major = element_line(colour = "azure3"))+
  geom_label_repel(data=de_lab, label=de_lab$delabel, show.legend = FALSE)+
  labs(x="log2 Fold Change",
       y="-log10(adjusted p-value)",
       title = "Vacuole (+) versus Vacuole (-) cell lines")+
  scale_colour_manual(name="Differential Expression", values = mycolors)
```

```
## Warning: Removed 3632 rows containing missing values (`geom_point()`).
```




```
#boxplot
status_vac<- sapply(sampleDat1$status, function(x){
  if(x=="pos"){status="positive"}
  else if(x=="neg"){status="negative"}
})
genes<- c("NLRP2", "GRIN2B", "IGFL2", "MYOM3", "RELN", "LAMP5", "AEBP1", "UBE2QL1")
genes_plot<- genes_matrix[match(genes, row.names(genes_matrix)),]
colnames(genes_plot)<- status_vac
cols=c("positive" = "red3", "negative" = "forestgreen")
```

```
plot_dat<- reshape2::melt(genes_plot)
plot_exp <- ggplot(data=plot_dat, aes(x=Var2, y=value, fill=Var2, colour=Var2))+
  geom_boxplot(alpha=0.7)+geom_jitter(show.legend = FALSE)+
  theme(plot.background = element_blank(),
        axis.title = element_text(face="bold"),
        panel.background = element_blank(),
        panel.border = element_rect(fill=NA),
        legend.position="bottom",
        legend.title=element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        strip.text.x = element_text(face="bold"),
        panel.grid.minor = element_line(colour = "azure3")
        )+
  labs(y="Gene Expression",
       x="SB202190-induced Vacuolation Status")+
  scale_colour_manual(values=cols)+
  scale_fill_manual(values=cols)+facet_wrap(~ Var1, ncol=4, nrow=2)
plot(plot_exp)
```




```
print(sessionInfo())
```

```
## R version 4.2.3 (2023-03-15 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
```

```
## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## attached base packages:
## [1] grid      stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] ComplexHeatmap_2.14.0       circlize_0.4.15
##  [3] ggrepel_0.9.3               ggplot2_3.4.3
##  [5] gplots_3.1.3                readr_2.1.4
##  [7] DESeq2_1.38.3               SummarizedExperiment_1.28.0
##  [9] Biobase_2.58.0              MatrixGenerics_1.10.0
## [11] matrixStats_1.0.0           GenomicRanges_1.50.2
## [13] GenomeInfoDb_1.34.9         IRanges_2.32.0
## [15] S4Vectors_0.36.2            BiocGenerics_0.44.0
## [17] RColorBrewer_1.1-3          tximport_1.26.1
##
## loaded via a namespace (and not attached):
##  [1] bitops_1.0-7           bit64_4.0.5            doParallel_1.0.17
##  [4] httr_1.4.7             tools_4.2.3            utf8_1.2.3
##  [7] R6_2.5.1               KernSmooth_2.23-20     DBI_1.1.3
## [10] colorspace_2.1-0       GetoptLong_1.0.5       withr_2.5.0
## [13] tidyselect_1.2.0       bit_4.0.5              compiler_4.2.3
## [16] cli_3.6.1              DelayedArray_0.24.0    labeling_0.4.2
## [19] caTools_1.18.2         scales_1.2.1           stringr_1.5.0
## [22] digest_0.6.33          rmarkdown_2.24         XVector_0.38.0
## [25] pkgconfig_2.0.3        htmltools_0.5.6        fastmap_1.1.1
## [28] rlang_1.1.0            GlobalOptions_0.1.2    rstudioapi_0.15.0
## [31] RSQLite_2.3.1          farver_2.1.1           shape_1.4.6
## [34] generics_0.1.3         BiocParallel_1.32.6    gtools_3.9.4
## [37] vroom_1.6.3            dplyr_1.1.2            RCurl_1.98-1.12
## [40] magrittr_2.0.3         GenomeInfoDbData_1.2.9 Matrix_1.6-1
## [43] Rcpp_1.0.10            munsell_0.5.0          fansi_1.0.4
## [46] lifecycle_1.0.3        stringi_1.7.12         yaml_2.3.7
## [49] zlibbioc_1.44.0        plyr_1.8.8             blob_1.2.4
## [52] parallel_4.2.3         crayon_1.5.2           lattice_0.20-45
## [55] Biostrings_2.66.0      annotate_1.76.0        hms_1.1.3
## [58] KEGGREST_1.38.0        locfit_1.5-9.8         knitr_1.43
## [61] pillar_1.9.0           rjson_0.2.21           reshape2_1.4.4
## [64] geneplotter_1.76.0     codetools_0.2-19       XML_3.99-0.14
## [67] glue_1.6.2             evaluate_0.21          BiocManager_1.30.22
## [70] png_0.1-8              vctrs_0.6.3            tzdb_0.4.0
## [73] foreach_1.5.2          gtable_0.3.3           clue_0.3-64
## [76] cachem_1.0.8           xfun_0.40              xtable_1.8-4
## [79] tibble_3.2.1           iterators_1.0.14       AnnotationDbi_1.60.2
```

```
## [82] memoise_2.0.1          cluster_2.1.4
```