

class15

Rui Huang (PID: A15606522)

11/16/2021

```
counts <- read.csv("airway_scaledcounts.csv", row.names=1)
metadata <- read.csv("airway_metadata.csv")
```

```
head(counts)
```

```
##          SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516
## ENSG00000000003     723      486      904      445     1170
## ENSG00000000005      0       0       0       0       0
## ENSG00000000419    467      523      616      371      582
## ENSG00000000457    347      258      364      237      318
## ENSG00000000460     96       81       73       66      118
## ENSG00000000938     0       0       1       0       2
##          SRR1039517 SRR1039520 SRR1039521
## ENSG00000000003    1097      806      604
## ENSG00000000005      0       0       0
## ENSG00000000419    781      417      509
## ENSG00000000457    447      330      324
## ENSG00000000460     94      102       74
## ENSG00000000938     0       0       0
```

```
head(metadata)
```

```
##      id   dex celltype geo_id
## 1 SRR1039508 control N61311 GSM1275862
## 2 SRR1039509 treated N61311 GSM1275863
## 3 SRR1039512 control N052611 GSM1275866
## 4 SRR1039513 treated N052611 GSM1275867
## 5 SRR1039516 control N080611 GSM1275870
## 6 SRR1039517 treated N080611 GSM1275871
```

#Q1: 38694 genes.

```
nrow(counts)
```

```
## [1] 38694
```

#Q2: we have 4 cell lines.

```

sum(metadata$dex == "control")

## [1] 4

control inds <- metadata$dex == "control"
control counts <- counts[, control inds]
head(control counts)

##          SRR1039508 SRR1039512 SRR1039516 SRR1039520
## ENSG00000000003     723      904     1170      806
## ENSG00000000005      0       0       0       0
## ENSG00000000419    467      616      582      417
## ENSG00000000457    347      364      318      330
## ENSG00000000460     96       73      118      102
## ENSG00000000938      0       1       2       0

control means <- rowMeans(control counts)

treated inds <- metadata$dex == "treated"
treated counts <- counts[, treated inds]
head(treated counts)

##          SRR1039509 SRR1039513 SRR1039517 SRR1039521
## ENSG00000000003    486      445     1097      604
## ENSG00000000005      0       0       0       0
## ENSG00000000419    523      371      781      509
## ENSG00000000457    258      237      447      324
## ENSG00000000460     81       66      94       74
## ENSG00000000938      0       0       0       0

treated means <- rowMeans(treated counts)

control <- metadata[metadata[, "dex"] == "control", ]
control counts <- counts[, control$id]
control mean <- rowSums(control counts) / 4
head(control mean)

## ENSG00000000003 ENSG00000000005 ENSG00000000419 ENSG00000000457 ENSG00000000460
##      900.75          0.00      520.50      339.75      97.25
## ENSG00000000938
##      0.75

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

control <- metadata %>% filter(dex=="control")
control.counts <- counts %>% select(control$id)
control.mean <- rowSums(control.counts)/4
head(control.mean)

## ENSG0000000003 ENSG0000000005 ENSG00000000419 ENSG00000000457 ENSG00000000460
##           900.75          0.00        520.50        339.75        97.25
## ENSG00000000938
##           0.75

#Q3: replace "4" with "nrow(control)" to divide the rowSum with the correct number to calculate the mean
# if there is more samples

control <- metadata[metadata[, "dex"]=="control",]
control.counts <- counts[, control$id]
control.mean <- rowSums( control.counts )/nrow(control)
head(control.mean)

## ENSG0000000003 ENSG0000000005 ENSG00000000419 ENSG00000000457 ENSG00000000460
##           900.75          0.00        520.50        339.75        97.25
## ENSG00000000938
##           0.75

#Q4:

treated <- metadata[metadata[, "dex"]=="treated",]
treated.counts <- counts[, treated$id]
treated.mean <- rowSums( treated.counts )/nrow(treated)
head(treated.mean)

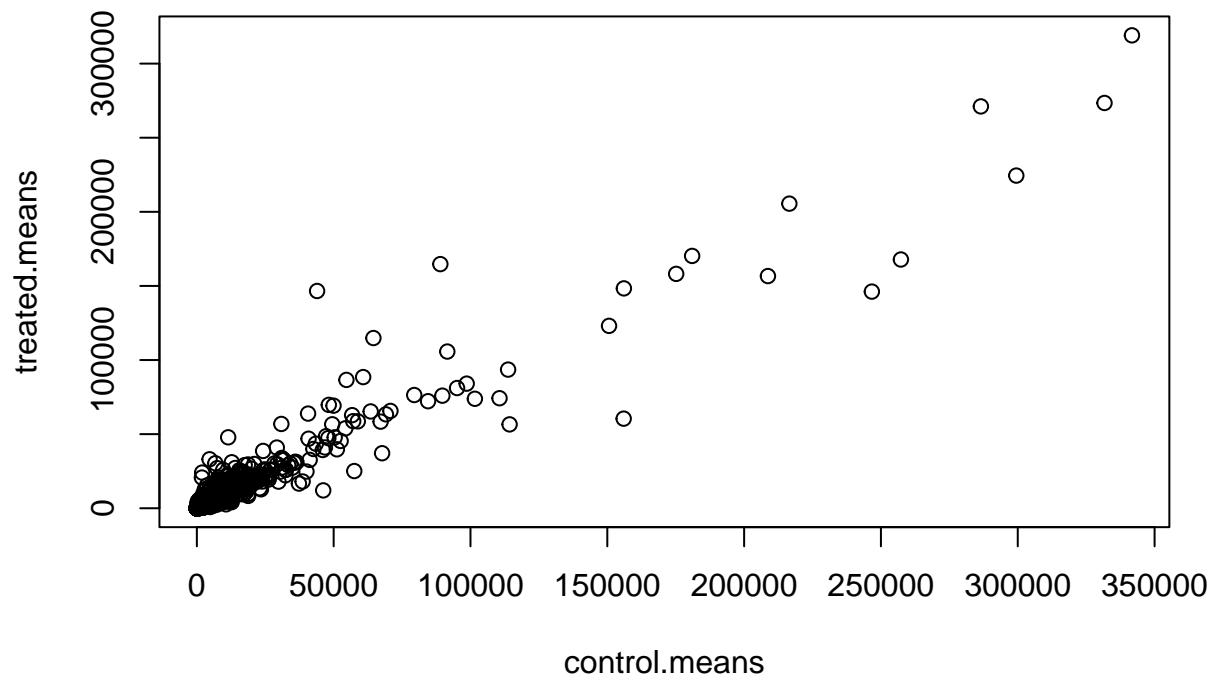
## ENSG0000000003 ENSG0000000005 ENSG00000000419 ENSG00000000457 ENSG00000000460
##           658.00          0.00        546.00        316.50        78.75
## ENSG00000000938
##           0.00

meancounts <- data.frame(control.means, treated.means)

#Q5:

plot(meancounts)

```

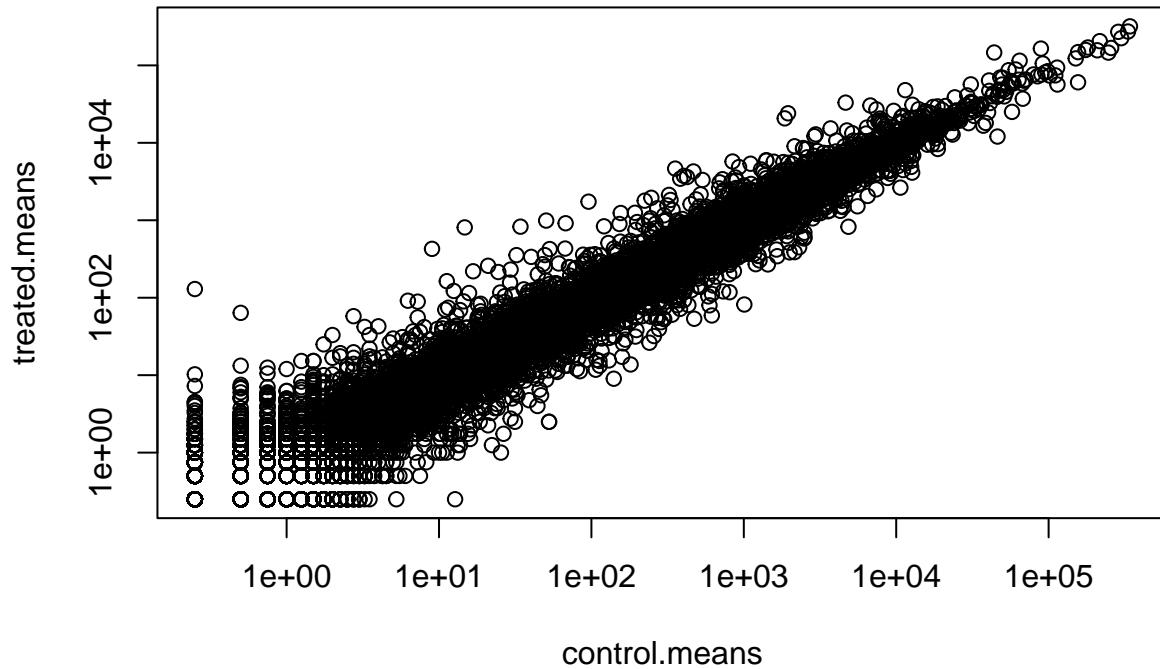


#Q6: argument log

```
plot(meancounts, log="xy")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 15032 x values <= 0 omitted  
## from logarithmic plot
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 15281 y values <= 0 omitted  
## from logarithmic plot
```



#0 value for no change, “+” for increase, “-” are for decrease. work with log2(fold-change), and log2(fold-change) for meancounts df

```
meancounts$log2fc <- log2(meancounts[, "treated.means"] / meancounts[, "control.means"])
head(meancounts)
```

```
##                  control.means treated.means      log2fc
## ENSG000000000003       900.75     658.00 -0.45303916
## ENSG000000000005        0.00      0.00       NaN
## ENSG00000000419       520.50     546.00  0.06900279
## ENSG00000000457       339.75     316.50 -0.10226805
## ENSG00000000460        97.25      78.75 -0.30441833
## ENSG00000000938        0.75      0.00       -Inf
```

```
zero.vals <- which(meancounts[, 1:2] == 0, arr.ind=TRUE)
to.rm <- unique(zero.vals[, 1])
mycounts <- meancounts[-to.rm,]
head(mycounts)
```

```
##                  control.means treated.means      log2fc
## ENSG000000000003       900.75     658.00 -0.45303916
## ENSG00000000419       520.50     546.00  0.06900279
## ENSG00000000457       339.75     316.50 -0.10226805
## ENSG00000000460        97.25      78.75 -0.30441833
## ENSG00000000971      5219.00    6687.50  0.35769358
## ENSG00000001036      2327.00    1785.75 -0.38194109
```

#Q7: arr.ind=T in which() will return the dimensional array indices of the zero counts genes. Taking the first column in the unique() function is to eliminate the duplicates, so that we can focus on the rows without counting any row twice when we type meancounts[-to.rm,]

#Q8:250

#Q9:367

```
up.ind <- mycounts$log2fc > 2  
down.ind <- mycounts$log2fc < (-2)  
sum(up.ind)
```

```
## [1] 250
```

```
sum(down.ind)
```

```
## [1] 367
```

#Q10: No, because we need DESeq2 set a p-value to filter out the non-statistically significant fold changes. Thus, this outcome is not very precise since it is solely based on fold changes regardless if they are significant or not.

```
head(meancounts[,1:2] == 0)
```

```
## control.means treated.means  
## ENSG00000000003 FALSE FALSE  
## ENSG00000000005 TRUE TRUE  
## ENSG00000000419 FALSE FALSE  
## ENSG00000000457 FALSE FALSE  
## ENSG00000000460 FALSE FALSE  
## ENSG00000000938 FALSE TRUE
```

```
zero.vals <- which(meancounts[,1:2] == 0, arr.ind = T)  
head(zero.vals)
```

```
##      row col  
## ENSG00000000005   2   1  
## ENSG00000004848  65   1  
## ENSG00000004948  70   1  
## ENSG00000005001  73   1  
## ENSG00000006059 121   1  
## ENSG00000006071 123   1
```

```
to.rm <- unique(zero.vals[, "row"])  
head(sort(to.rm))
```

```
## [1] 2 6 65 70 73 81
```

```
mycounts <- meancounts[-to.rm,]  
head(mycounts)
```

```

##           control.means treated.means      log2fc
## ENSG00000000003      900.75      658.00 -0.45303916
## ENSG00000000419      520.50      546.00  0.06900279
## ENSG00000000457      339.75      316.50 -0.10226805
## ENSG00000000460      97.25       78.75 -0.30441833
## ENSG00000000971     5219.00     6687.50  0.35769358
## ENSG00000001036     2327.00     1785.75 -0.38194109

nrow(mycounts)

## [1] 21817

sum(mycounts$log2fc)

## [1] -3442.214

library(DESeq2)

## Loading required package: S4Vectors

## Loading required package: stats4

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

## 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.max, which.min

##
## Attaching package: 'S4Vectors'

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

```

```

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

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

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

## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb

## Loading required package: SummarizedExperiment

## Loading required package: MatrixGenerics

## Loading required package: matrixStats

##
## Attaching package: 'matrixStats'

## The following object is masked from 'package:dplyr':
##
##     count

##
## 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

citation("DESeq2")

```

```

##
## Love, M.I., Huber, W., Anders, S. Moderated estimation of fold change
## and dispersion for RNA-seq data with DESeq2 Genome Biology 15(12):550
## (2014)
##
## A BibTeX entry for LaTeX users is
##
## @Article{,
##   title = {Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2},
##   author = {Michael I. Love and Wolfgang Huber and Simon Anders},
##   year = {2014},
##   journal = {Genome Biology},
##   doi = {10.1186/s13059-014-0550-8},
##   volume = {15},
##   issue = {12},
##   pages = {550},
## }

```

fist set up input for Deseq

```

dds <- DESeqDataSetFromMatrix(countData=counts,
                               colData=metadata,
                               design=~dex)

```

```

## converting counts to integer mode

```

```

## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors

```

```

dds

## class: DESeqDataSet
## dim: 38694 8
## metadata(1): version
## assays(1): counts
## rownames(38694): ENSG00000000003 ENSG00000000005 ... ENSG00000283120
##   ENSG00000283123
## rowData names(0):
## colnames(8): SRR1039508 SRR1039509 ... SRR1039520 SRR1039521
## colData names(4): id dex celltype geo_id

```

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

```

## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing

```

```
res <- results(dds)
res
```

```

## log2 fold change (MLE): dex treated vs control
## Wald test p-value: dex treated vs control
## DataFrame with 38694 rows and 6 columns
##           baseMean log2FoldChange      lfcSE       stat     pvalue
##             <numeric>      <numeric> <numeric> <numeric> <numeric>
## ENSG00000000003    747.1942     -0.3507030  0.168246 -2.084470  0.0371175
## ENSG00000000005     0.0000        NA         NA         NA         NA
## ENSG00000000419    520.1342     0.2061078  0.101059  2.039475  0.0414026
## ENSG00000000457    322.6648     0.0245269  0.145145  0.168982  0.8658106
## ENSG00000000460    87.6826     -0.1471420  0.257007 -0.572521  0.5669691
## ...
##           ...          ...          ...          ...          ...
## ENSG00000283115    0.000000        NA         NA         NA         NA
## ENSG00000283116    0.000000        NA         NA         NA         NA
## ENSG00000283119    0.000000        NA         NA         NA         NA
## ENSG00000283120    0.974916     -0.668258   1.69456  -0.394354  0.693319
## ENSG00000283123    0.000000        NA         NA         NA         NA
##           padj
##             <numeric>
## ENSG00000000003    0.163035
## ENSG00000000005     NA
## ENSG00000000419    0.176032
## ENSG00000000457    0.961694

```

```

## ENSG00000000460 0.815849
## ...
## ENSG00000283115 NA
## ENSG00000283116 NA
## ENSG00000283119 NA
## ENSG00000283120 NA
## ENSG00000283123 NA

summary(res)

##
## out of 25258 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)      : 1563, 6.2%
## LFC < 0 (down)    : 1188, 4.7%
## outliers [1]       : 142, 0.56%
## low counts [2]     : 9971, 39%
## (mean count < 10)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

res05 <- results(dds, alpha=0.05)
summary(res05)

##
## out of 25258 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)      : 1236, 4.9%
## LFC < 0 (down)    : 933, 3.7%
## outliers [1]       : 142, 0.56%
## low counts [2]     : 9033, 36%
## (mean count < 6)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

library(AnnotationDbi)

## Warning: package 'AnnotationDbi' was built under R version 4.1.2

##
## Attaching package: 'AnnotationDbi'

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

library(org.Hs.eg.db)

##

```

```

columns(org.Hs.eg.db)

## [1] "ACCCNUM"      "ALIAS"        "ENSEMBL"       "ENSEMLPROT"    "ENSEMLTRANS"
## [6] "ENTREZID"     "ENZYME"       "EVIDENCE"      "EVIDENCEALL"   "GENENAME"
## [11] "GENETYPE"     "GO"           "GOALL"         "IPI"          "MAP"
## [16] "OMIM"          "ONTOLOGY"     "ONTOLOGYALL"  "PATH"         "PFAM"
## [21] "PMID"          "PROSITE"      "REFSEQ"        "SYMBOL"       "UCSCKG"
## [26] "UNIPROT"

res$symbol <- mapIds(org.Hs.eg.db,
                      keys=row.names(res),
                      keytype="ENSEMBL",
                      column="SYMBOL",
                      multiVals="first")

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

head(res)

## log2 fold change (MLE): dex treated vs control
## Wald test p-value: dex treated vs control
## DataFrame with 6 rows and 7 columns
##           baseMean log2FoldChange      lfcSE      stat      pvalue
##           <numeric>      <numeric> <numeric> <numeric> <numeric>
## ENSG000000000003 747.194195 -0.3507030  0.168246 -2.084470 0.0371175
## ENSG000000000005  0.000000      NA        NA        NA        NA
## ENSG000000000419 520.134160  0.2061078  0.101059  2.039475 0.0414026
## ENSG000000000457 322.664844  0.0245269  0.145145  0.168982 0.8658106
## ENSG000000000460  87.682625 -0.1471420  0.257007 -0.572521 0.5669691
## ENSG000000000938  0.319167 -1.7322890  3.493601 -0.495846 0.6200029
##           padj      symbol
##           <numeric> <character>
## ENSG000000000003  0.163035  TSPAN6
## ENSG000000000005   NA        TNMD
## ENSG000000000419  0.176032  DPM1
## ENSG000000000457  0.961694  SCYL3
## ENSG000000000460  0.815849  C1orf112
## ENSG000000000938   NA        FGR

#Q11:

res$entrez <- mapIds(org.Hs.eg.db,
                      keys=row.names(res),
                      column="ENTREZID",
                      keytype="ENSEMBL",
                      multiVals="first")

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

```

```

res$uniprot <- mapIds(org.Hs.eg.db,
                      keys=row.names(res),
                      column="UNIPROT",
                      keytype="ENSEMBL",
                      multiVals="first")

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

res$genename <- mapIds(org.Hs.eg.db,
                      keys=row.names(res),
                      column="GENENAME",
                      keytype="ENSEMBL",
                      multiVals="first")

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

head(res)

## log2 fold change (MLE): dex treated vs control
## Wald test p-value: dex treated vs control
## DataFrame with 6 rows and 10 columns
##           baseMean log2FoldChange      lfcSE      stat     pvalue
##           <numeric>      <numeric> <numeric> <numeric> <numeric>
## ENSG00000000003 747.194195 -0.3507030  0.168246 -2.084470 0.0371175
## ENSG00000000005  0.000000    NA        NA        NA        NA
## ENSG00000000419 520.134160  0.2061078  0.101059  2.039475 0.0414026
## ENSG00000000457 322.664844  0.0245269  0.145145  0.168982 0.8658106
## ENSG00000000460 87.682625 -0.1471420  0.257007 -0.572521 0.5669691
## ENSG00000000938 0.319167 -1.7322890  3.493601 -0.495846 0.6200029
##          padj      symbol      entrez      uniprot
##          <numeric> <character> <character> <character>
## ENSG00000000003 0.163035   TSPAN6      7105 AOA024RCI0
## ENSG00000000005  NA        TNMD       64102 Q9H2S6
## ENSG00000000419 0.176032   DPM1       8813 060762
## ENSG00000000457 0.961694   SCYL3      57147 Q8IZE3
## ENSG00000000460 0.815849   C1orf112    55732 AOA024R922
## ENSG00000000938  NA        FGR        2268 P09769
##          genename
##          <character>
## ENSG00000000003  tetraspanin 6
## ENSG00000000005  tenomodulin
## ENSG00000000419  dolichyl-phosphate m..
## ENSG00000000457  SCY1 like pseudokina..
## ENSG00000000460  chromosome 1 open re..
## ENSG00000000938  FGR proto-oncogene, ..

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

## log2 fold change (MLE): dex treated vs control
## Wald test p-value: dex treated vs control

```

```

## DataFrame with 6 rows and 10 columns
##           baseMean log2FoldChange      lfcSE      stat     pvalue
##          <numeric>      <numeric> <numeric> <numeric>    <numeric>
## ENSG00000152583   954.771       4.36836  0.2371268   18.4220 8.74490e-76
## ENSG00000179094   743.253       2.86389  0.1755693   16.3120 8.10784e-60
## ENSG00000116584   2277.913      -1.03470 0.0650984  -15.8944 6.92855e-57
## ENSG00000189221   2383.754       3.34154  0.2124058   15.7319 9.14433e-56
## ENSG00000120129   3440.704       2.96521  0.2036951   14.5571 5.26424e-48
## ENSG00000148175  13493.920      1.42717  0.1003890   14.2164 7.25128e-46
##           padj      symbol      entrez      uniprot
##          <numeric> <character> <character> <character>
## ENSG00000152583 1.32441e-71    SPARCL1      8404  AOA024RDE1
## ENSG00000179094 6.13966e-56    PER1        5187  Q15534
## ENSG00000116584 3.49776e-53    ARHGEF2      9181  Q92974
## ENSG00000189221 3.46227e-52    MAOA        4128  P21397
## ENSG00000120129 1.59454e-44    DUSP1        1843  B4DU40
## ENSG00000148175 1.83034e-42    STOM        2040  F8VSL7
##           genename
##          <character>
## ENSG00000152583      SPARC like 1
## ENSG00000179094      period circadian reg..
## ENSG00000116584      Rho/Rac guanine nucl..
## ENSG00000189221      monoamine oxidase A
## ENSG00000120129      dual specificity pho..
## ENSG00000148175      stomatin

```

```
write.csv(res[ord,], "deseq_results.csv")
```

write out whole results dataset (including genes that don't change significantly).

```
write.csv(res, file="allmyresults.csv")
```

```
summary(res05)
```

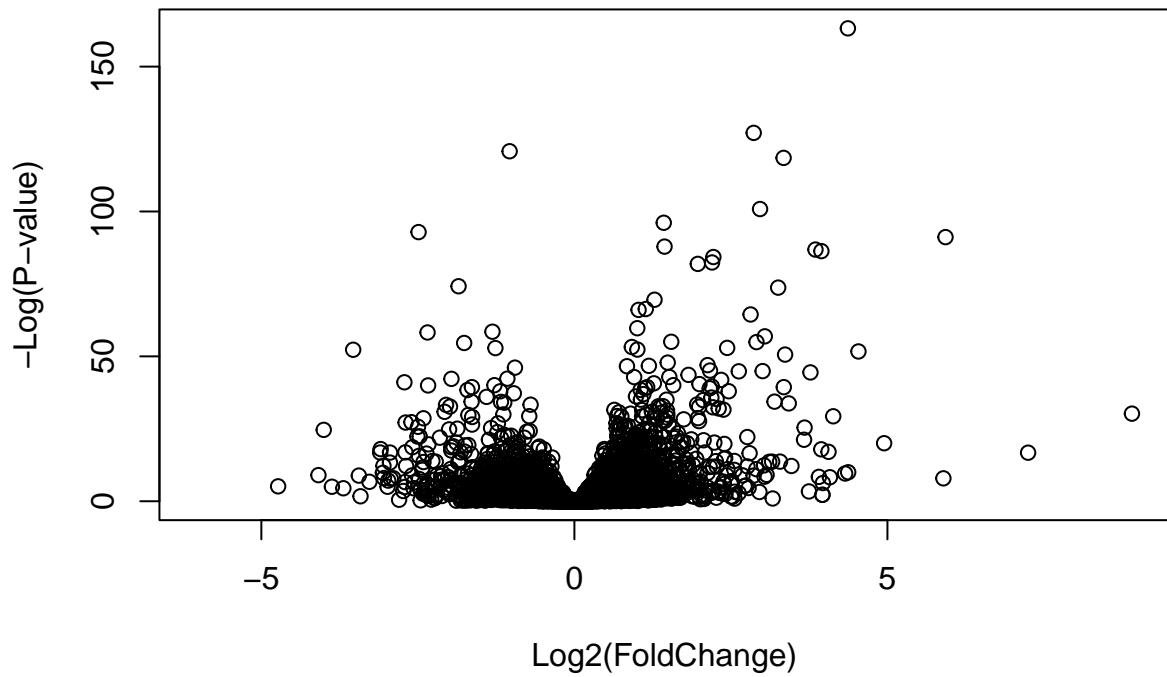
```

## 
## out of 25258 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)      : 1236, 4.9%
## LFC < 0 (down)    : 933, 3.7%
## outliers [1]       : 142, 0.56%
## low counts [2]     : 9033, 36%
## (mean count < 6)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

```

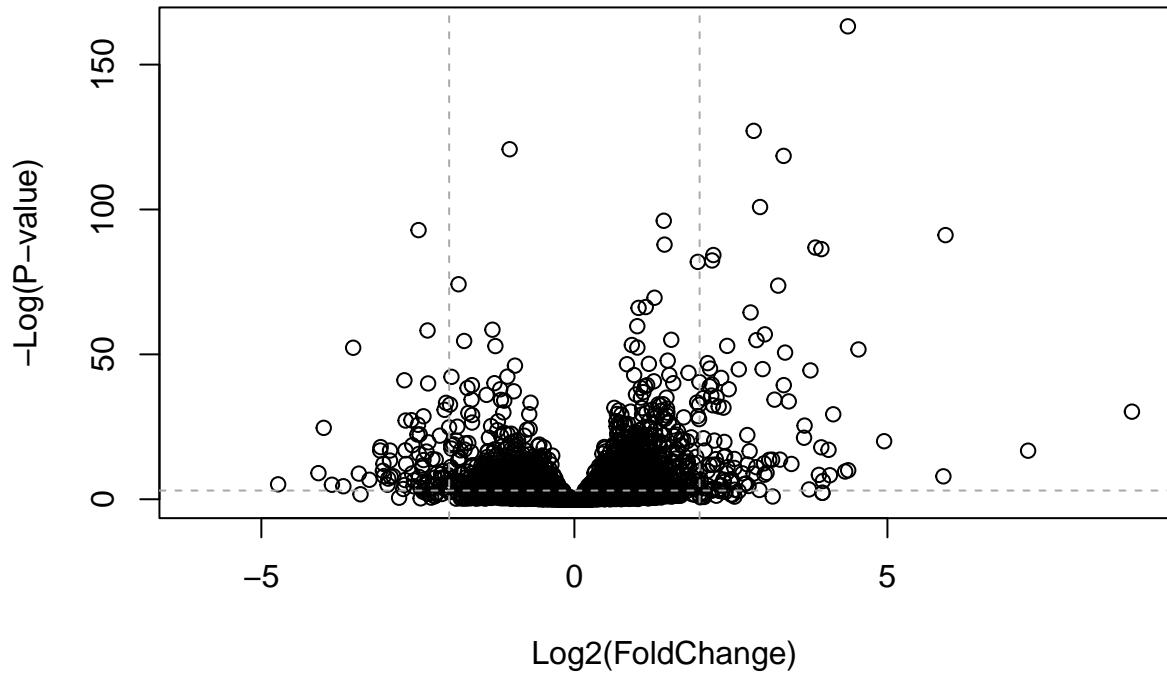
volcano plot

```
plot( res$log2FoldChange, -log(res$padj), ylab="-Log(P-value)", xlab="Log2(FoldChange)")
```



flip pvalue axis by putting a minus sign on it then we will have classic volcano plot.

```
plot( res$log2FoldChange, -log(res$padj),
  ylab="-Log(P-value)", xlab="Log2(FoldChange)")
abline(v=c(-2,2), col="darkgray", lty=2)
abline(h=-log(0.05), col="darkgray", lty=2)
```



```

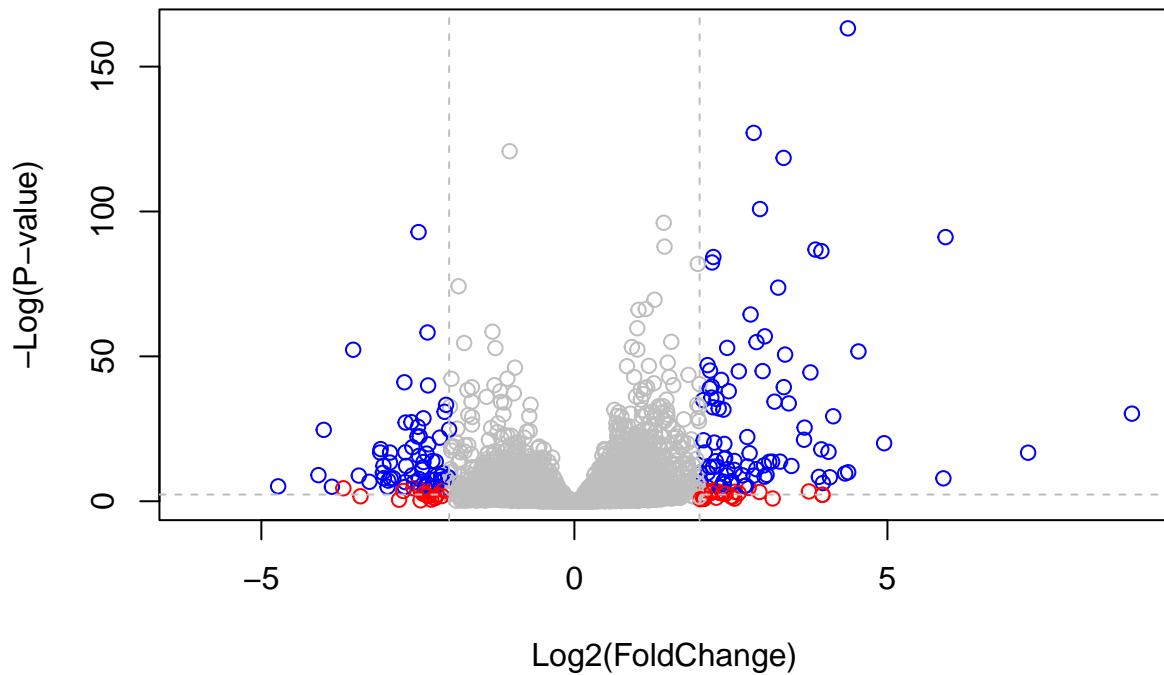
# Setup our custom point color vector
mycols <- rep("gray", nrow(res))
mycols[ abs(res$log2FoldChange) > 2 ] <- "red"

inds <- (res$padj < 0.01) & (abs(res$log2FoldChange) > 2 )
mycols[ inds ] <- "blue"

# Volcano plot with custom colors
plot( res$log2FoldChange, -log(res$padj),
      col=mycols, ylab="-Log(P-value)", xlab="Log2(FoldChange)" )

# Cut-off lines
abline(v=c(-2,2), col="gray", lty=2)
abline(h=-log(0.1), col="gray", lty=2)

```



add annotation data for our genes.

```
columns(org.Hs.eg.db)

## [1] "ACNUM"      "ALIAS"       "ENSEMBL"     "ENSEMLPROT"  "ENSEMLTRANS"
## [6] "ENTREZID"   "ENZYME"     "EVIDENCE"    "EVIDENCEALL" "GENENAME"
## [11] "GENETYPE"   "GO"         "GOALL"       "IPI"        "MAP"
## [16] "OMIM"        "ONTOLOGY"   "ONTOLOGYALL" "PATH"       "PFAM"
## [21] "PMID"        "PROSITE"    "REFSEQ"      "SYMBOL"     "UCSCKG"
## [26] "UNIPROT"

library(EnhancedVolcano)

## Loading required package: ggplot2

## Loading required package: ggrepel

## Registered S3 methods overwritten by 'ggalt':
##   method           from
##   grid.draw.absoluteGrob  ggplot2
##   grobHeight.absoluteGrob ggplot2
##   grobWidth.absoluteGrob ggplot2
##   grobX.absoluteGrob    ggplot2
##   grobY.absoluteGrob    ggplot2
```

```

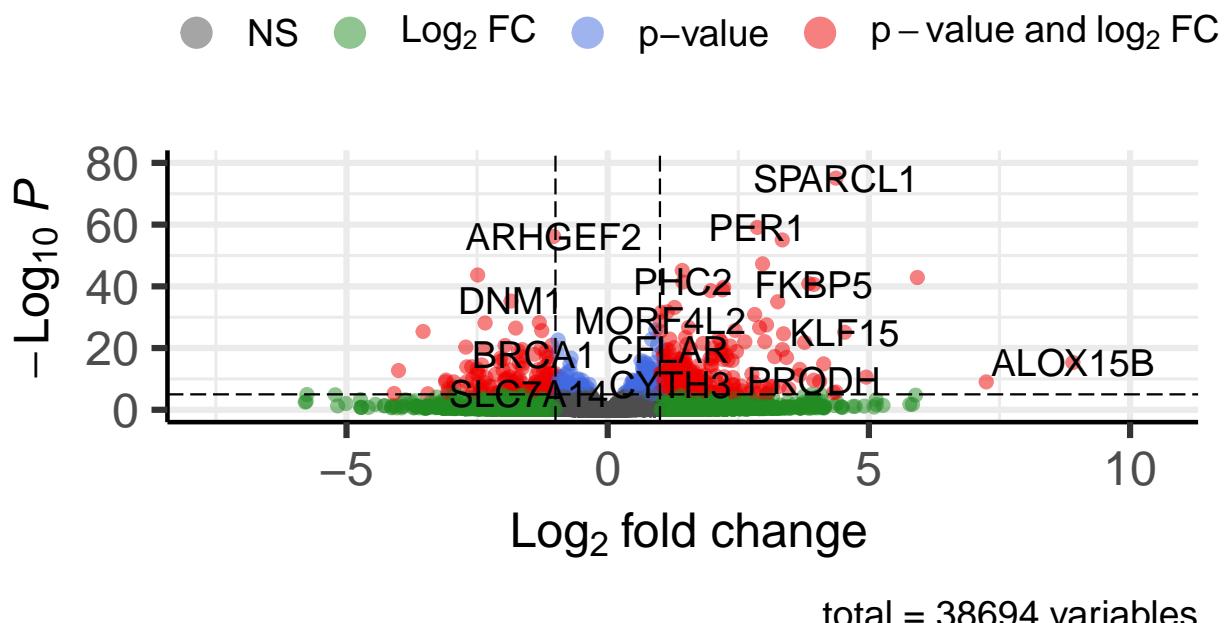
x <- as.data.frame(res)

EnhancedVolcano(x,
  lab = x$symbol,
  x = 'log2FoldChange',
  y = 'pvalue')

```

Volcano plot

EnhancedVolcano



```

res$entrez <- mapIds(org.Hs.eg.db,
  keys=row.names(res),
  keytype="ENSEMBL",
  column="SYMBOL",
  multiVals="first")

```

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

the main **gage()** function requires a named vector of fold changes,

```
library(pathview)
```

```

## #####
## Pathview is an open source software package distributed under GNU General
## Public License version 3 (GPLv3). Details of GPLv3 is available at
## http://www.gnu.org/licenses/gpl-3.0.html. Particullary, users are required to
## formally cite the original Pathview paper (not just mention it) in publications

```

```

## or products. For details, do citation("pathview") within R.
##
## The pathview downloads and uses KEGG data. Non-academic uses may require a KEGG
## license agreement (details at http://www.kegg.jp/kegg/legal.html).
## #####
library(gage)

## 

library(gageData)
data(kegg.sets.hs)

foldchange <- res$log2FoldChange
names(foldchange) = res$entrez

keggres = gage(foldchange, gsets = kegg.sets.hs)

attributes(keggres)

## $names
## [1] "greater" "less"    "stats"

head(keggres$less, 3)

##                                     p.geomean stat.mean p.val q.val
## hsa00232 Caffeine metabolism          NA      NaN   NA   NA
## hsa00983 Drug metabolism - other enzymes  NA      NaN   NA   NA
## hsa01100 Metabolic pathways          NA      NaN   NA   NA
##                                         set.size exp1
## hsa00232 Caffeine metabolism          0     NA
## hsa00983 Drug metabolism - other enzymes  0     NA
## hsa01100 Metabolic pathways          0     NA

pathview(gene.data=foldchange, pathway.id="hsa05310")

## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.

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

## Info: Working in directory /Users/apple/Desktop/bimm143_github/class15

## Info: Writing image file hsa05310.pathview.png

#Q12:

```

```
pathview(gene.data=foldchange, pathway.id="hsa05332")
```

```
## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.
```

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

```
## Info: Working in directory /Users/apple/Desktop/bimm143_github/class15
```

```
## Info: Writing image file hsa05332.pathview.png
```

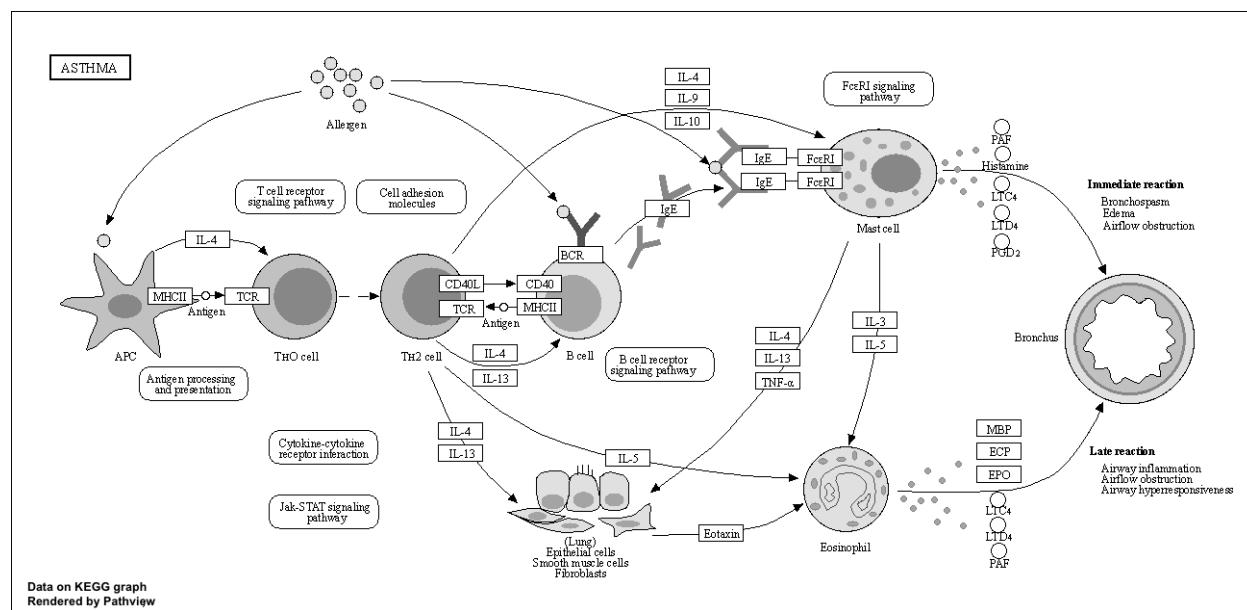
```
pathview(gene.data=foldchange, pathway.id="hsa04940")
```

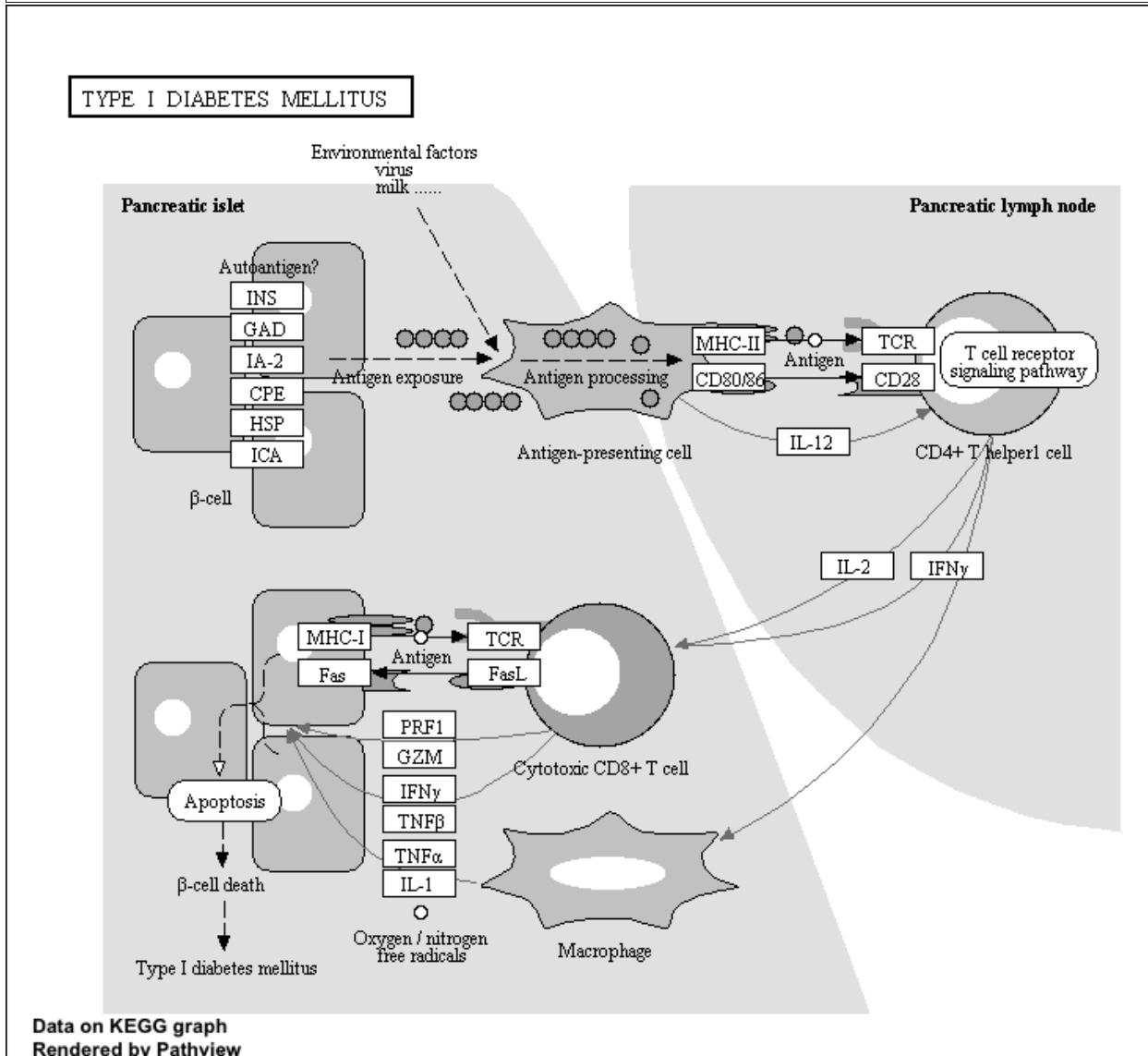
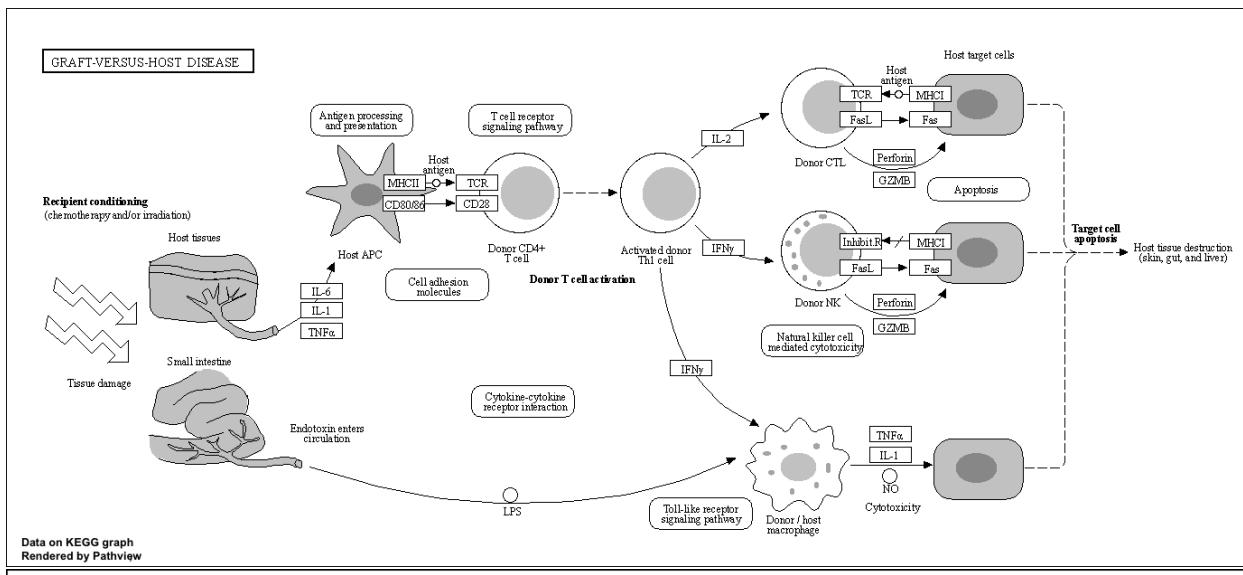
```
## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.
```

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

```
## Info: Working in directory /Users/apple/Desktop/bimm143_github/class15
```

```
## Info: Writing image file hsa04940.pathview.png
```





```
pathview(gene.data=foldchange, pathway.id="hsa05310", kegg.native=FALSE)

## Warning: None of the genes or compounds mapped to the pathway!
## Argument gene.idtype or cpd.idtype may be wrong.

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

## Info: Working in directory /Users/apple/Desktop/bimm143_github/class15

## Info: Writing image file hsa05310.pathview.pdf
```