code_nafld_final_Lynn

Initialization

Packages import

First, we import packages and datasets

```
# Install Bioconductor packages
#if (!require("BiocManager", quietly = TRUE))
    #install.packages("BiocManager")
#BiocManager::install(version = "3.17")

## Package to download data from acession numbers
# BiocManager::install("Biobase")
# BiocManager::install("GEOquery")
library(Biobase)
```

```
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

Welcome to Bioconductor

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

```
library(GEOquery)
```

```
Setting options('download.file.method.GEOquery'='auto')

Setting options('GEOquery.inmemory.gpl'=FALSE)
```

```r
## Packages for DESeq2
# Install DESeq2 package
# BiocManager::install("DESeq2")
library(DESeq2)
```

```
Warning: package 'DESeq2' was built under R version 4.3.1

Loading required package: S4Vectors

Warning: package 'S4Vectors' was built under R version 4.3.1

Loading required package: stats4


Attaching package: 'S4Vectors'

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

Warning: package 'IRanges' was built under R version 4.3.1


Attaching package: 'IRanges'

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

    windows

Loading required package: GenomicRanges

Warning: package 'GenomicRanges' was built under R version 4.3.1

Loading required package: GenomeInfoDb

Warning: package 'GenomeInfoDb' was built under R version 4.3.1

Loading required package: SummarizedExperiment

Loading required package: MatrixGenerics

Warning: package 'MatrixGenerics' was built under R version 4.3.1

Loading required package: matrixStats

Warning: package 'matrixStats' was built under R version 4.3.2
```

```
Attaching package: 'matrixStats'

The following objects are masked from 'package:Biobase':

    anyMissing, rowMedians


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

The following object is masked from 'package:Biobase':

    rowMedians
```

```
# package for logistic regression + basic calculation
# install.packages("glmnet", repos = "http://cran.us.r-project.org")
```

```
library(glmnet)
```

```
Warning: package 'glmnet' was built under R version 4.3.3

Loading required package: Matrix

Warning: package 'Matrix' was built under R version 4.3.2


Attaching package: 'Matrix'

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

    expand

Loaded glmnet 4.1-8
```

```
library(dplyr)
```

```
Warning: package 'dplyr' was built under R version 4.3.2
```

```

Attaching package: 'dplyr'
```

```
The following object is masked from 'package:matrixStats':

    count
```

```
The following objects are masked from 'package:GenomicRanges':

    intersect, setdiff, union
```

```
The following object is masked from 'package:GenomeInfoDb':

    intersect
```

```
The following objects are masked from 'package:IRanges':

    collapse, desc, intersect, setdiff, slice, union
```

```
The following objects are masked from 'package:S4Vectors':

    first, intersect, rename, setdiff, setequal, union
```

```
The following object is masked from 'package:Biobase':

    combine
```

```
The following objects are masked from 'package:BiocGenerics':

    combine, intersect, setdiff, union
```

```
The following objects are masked from 'package:stats':

    filter, lag
```

```
The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
Warning: package 'ggplot2' was built under R version 4.3.2
```

```
## Packages for GOSeq (pathway examination)
#BiocManager::install("goseq")
#BiocManager::install("clusterProfiler")
#BiocManager::install("AnnotationDbi")
#BiocManager::install("org.Hs.eg.db")
```

```r
#BiocManager::install("DOSE")
# BiocManager::install("hgu133a.db")
library(fgsea)
library(clusterProfiler)
```

```
Warning: package 'clusterProfiler' was built under R version 4.3.1
```

```
Registered S3 methods overwritten by 'treeio':
  method               from
  MRCA.phylo           tidytree
  MRCA.treedata        tidytree
  Nnode.treedata       tidytree
  Ntip.treedata        tidytree
  ancestor.phylo       tidytree
  ancestor.treedata    tidytree
  child.phylo          tidytree
  child.treedata       tidytree
  full_join.phylo      tidytree
  full_join.treedata   tidytree
  groupClade.phylo     tidytree
  groupClade.treedata tidytree
  groupOTU.phylo       tidytree
  groupOTU.treedata    tidytree
  inner_join.phylo     tidytree
  inner_join.treedata tidytree
  is.rooted.treedata   tidytree
  nodeid.phylo         tidytree
  nodeid.treedata      tidytree
  nodelab.phylo        tidytree
  nodelab.treedata     tidytree
  offspring.phylo      tidytree
  offspring.treedata   tidytree
  parent.phylo         tidytree
  parent.treedata      tidytree
  root.treedata        tidytree
  rootnode.phylo       tidytree
  sibling.phylo        tidytree
```

```
clusterProfiler v4.8.3  For help: https://yulab-smu.top/biomedical-knowledge-
mining-book/

If you use clusterProfiler in published research, please cite:
T Wu, E Hu, S Xu, M Chen, P Guo, Z Dai, T Feng, L Zhou, W Tang, L Zhan, X Fu,
S Liu, X Bo, and G Yu. clusterProfiler 4.0: A universal enrichment tool for
interpreting omics data. The Innovation. 2021, 2(3):100141


Attaching package: 'clusterProfiler'
```

```
The following object is masked from 'package:IRanges':

    slice

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

    rename

The following object is masked from 'package:stats':

    filter
```

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

```
Loading required package: AnnotationDbi

Warning: package 'AnnotationDbi' was built under R version 4.3.1


Attaching package: 'AnnotationDbi'

The following object is masked from 'package:clusterProfiler':

    select

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

    select
```

```
library(AnnotationDbi)
library(DOSE)
```

```
Warning: package 'DOSE' was built under R version 4.3.1

DOSE v3.26.2  For help: https://yulab-smu.top/biomedical-knowledge-mining-
book/

If you use DOSE in published research, please cite:
Guangchuang Yu, Li-Gen Wang, Guang-Rong Yan, Qing-Yu He. DOSE: an
R/Bioconductor package for Disease Ontology Semantic and Enrichment analysis.
Bioinformatics 2015, 31(4):608-609
```

```
library(hgu133a.db)
```

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

```
Warning: package 'GOplot' was built under R version 4.3.3
```

```
Loading required package: ggdendro

Warning: package 'ggdendro' was built under R version 4.3.3

Loading required package: gridExtra

Warning: package 'gridExtra' was built under R version 4.3.1


Attaching package: 'gridExtra'

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

    combine

The following object is masked from 'package:Biobase':

    combine

The following object is masked from 'package:BiocGenerics':

    combine

Loading required package: RColorBrewer
```

```
#BiocManager::install("illuminaHumanv4.db")
library("illuminaHumanv4.db")
```

Data import

```
#US_1 - Arendt et al., 2015
data_u <- getGEO("GSE89632")

Found 1 file(s)

GSE89632_series_matrix.txt.gz

data_u <- data_u$GSE89632_series_matrix.txt.gz


#US_2
data_u2 <- getGEO("GSE163211")

Found 1 file(s)

GSE163211_series_matrix.txt.gz

data_u2 <- data_u2$GSE163211_series_matrix.txt.gz
```

Preprocessing

Get to know the data

```
#US_1
# dimension of data
dim(data_u)

Features  Samples
   29377       63

# check for zeros expression values and/or NA, just in case
zeros <- apply(exprs(data_u), 1, function(x) sum(x==0))
data_u <- data_u[zeros!=63,]
dim(data_u)

Features  Samples
   29377       63

#US_2
# dimension of data
dim(data_u2)

Features  Samples
     800      318

# check for zeros expression values and/or NA, just in case
zeros <- apply(exprs(data_u2), 1, function(x) sum(x==0))
data_u2 <- data_u2[zeros!=318,]
dim(data_u2)

Features  Samples
     800      318
```

All datasets are cleaned and do not have zero expression values.

In the analysis, we will focus on gene expression values and its relationship with each other and phenotype characteristics.

Andrent et al (USA data 1)

Differentially expressed analysis

Get values for progression of NAFLD

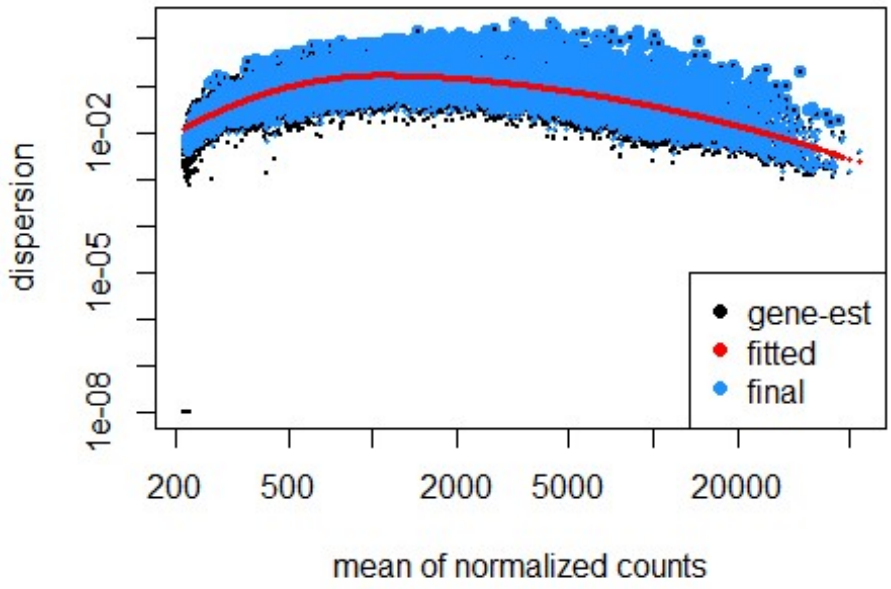
```
# Values for diabetes
data_uu <- data_u[, !is.na(data_u@phenoData@data[["characteristics_ch1.22"]])
& data_u@phenoData@data[["characteristics_ch1.22"]] == "diabetes: yes" |
```

```r
data_u@phenoData@data[["characteristics_ch1.22"]] == "diabetes: no" ]

data_uu$diabetes <- ifelse(data_uu@phenoData@data[["characteristics_ch1.22"]]
== "diabetes: no", '0', '1')

# Values for progression of NAFLD
data_uu$stages <- gsub("[^:]+: (.*)", "\\1", data_uu$characteristics_ch1.1)

data_uu$stages <- ifelse(data_uu$stages == 'HC', 1,
                  ifelse(data_uu$stages == 'NASH', 2,
                         ifelse(data_uu$stages == 'SS', 3, data_uu$stages)))




# get the back-transformed data into a new variable
data_u_expr = data_uu@assayData$exprs
data_u_expr = round((2^data_u_expr-1),0)

# Fit DESeq2
dds_u <- DESeqDataSetFromMatrix(countData = data_u_expr, colData =
pData(data_uu), design=~stages+diabetes)

converting counts to integer mode

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

dds_u <- DESeq(dds_u)

estimating size factors

estimating dispersions

gene-wise dispersion estimates

mean-dispersion relationship

-- note: fitType='parametric', but the dispersion trend was not well captured
by the
   function: y = a/x + b, and a local regression fit was automatically
substituted.
   specify fitType='local' or 'mean' to avoid this message next time.

final dispersion estimates

fitting model and testing

-- replacing outliers and refitting for 451 genes
-- DESeq argument 'minReplicatesForReplace' = 7
-- original counts are preserved in counts(dds)
```

```
estimating dispersions

fitting model and testing
```

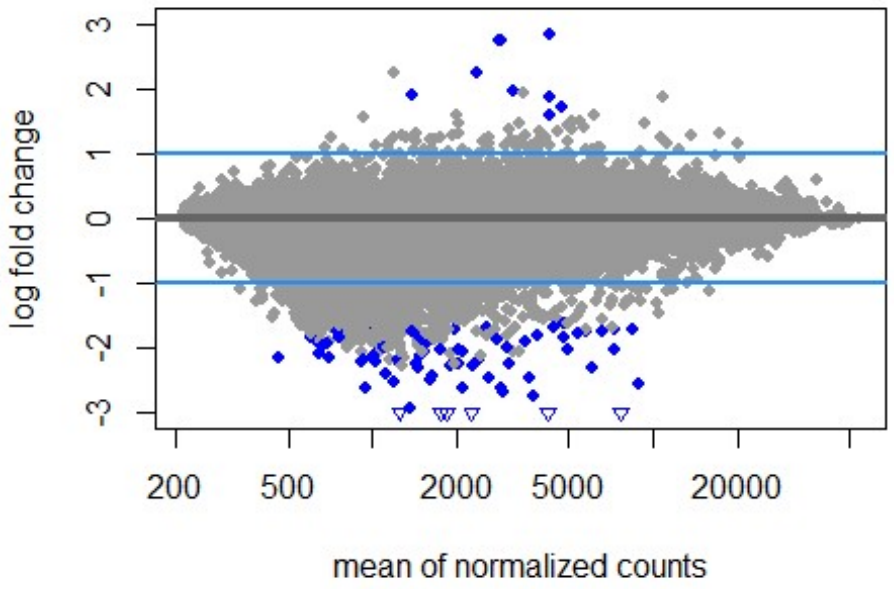
Plot dispersal

```
plotDispEsts(dds_u)
```



MA plot

```
resApeT <- lfcShrink(dds_u, coef=2, type="apeglm", lfcThreshold=1)

using 'apeglm' for LFC shrinkage. If used in published research, please cite:
    Zhu, A., Ibrahim, J.G., Love, M.I. (2018) Heavy-tailed prior
distributions for
    sequence count data: removing the noise and preserving large differences.
    Bioinformatics. https://doi.org/10.1093/bioinformatics/bty895

computing FSOS 'false sign or small' s-values (T=1)

plotMA(resApeT, ylim=c(-3,3), cex=.8)

thresholding s-values on alpha=0.005 to color points

abline(h=c(-1,1), col="dodgerblue", lwd=2)
```



Compare healthy vs simple steatosis

```
# get the result and summary
head(dds_u)
```

```
class: DESeqDataSet
dim: 6 29
metadata(1): version
assays(6): counts mu ... replaceCounts replaceCooks
rownames(6): ILMN_1343291 ILMN_1651209 ... ILMN_1651235 ILMN_1651236
rowData names(31): baseMean baseVar ... maxCooks replace
colnames(29): GSM2385720 GSM2385723 ... GSM2385771 GSM2385773
colData names(86): title geo_accession ... sizeFactor replaceable
```

```
dds.results_u <- results(dds_u, contrast = c('stages', "2", "1"))
summary(dds.results_u, alpha = 0.05) # p-value = 0.05
```

```


out of 29377 with nonzero total read count
adjusted p-value < 0.05
LFC > 0 (up)       : 1031, 3.5%
LFC < 0 (down)     : 2587, 8.8%
outliers [1]       : 292, 0.99%
low counts [2]     : 2278, 7.8%
(mean count < 280)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results
```

```r
# Only get differential expressed genes (p-val <= 0.05 and log2FC outside of
[-2,2])
dds.results_u.h_ss <- dds.results_u[!is.na(dds.results_u$padj) &
dds.results_u$padj <= 0.05 & dds.results_u$log2FoldChange > 2 |
dds.results_u$log2FoldChange < -2 ,]

head(dds.results_u.h_ss)
```

```
log2 fold change (MLE): stages 2 vs 1
Wald test p-value: stages 2 vs 1
DataFrame with 6 rows and 6 columns
              baseMean log2FoldChange     lfcSE      stat      pvalue
             <numeric>      <numeric> <numeric> <numeric>   <numeric>
ILMN_1651498  3177.259       -2.35824  0.568095  -4.15113 3.30844e-05
ILMN_1651838  2099.424       -2.25276  0.458222  -4.91631 8.81914e-07
ILMN_1652287   653.217       -2.00275  0.449868  -4.45186          NA
ILMN_1652464   994.163       -2.05476  0.509094  -4.03611 5.43439e-05
ILMN_1652866   961.951       -2.14865  0.555177  -3.87020 1.08744e-04
ILMN_1653447  1787.750       -2.34395  0.684971  -3.42197 6.21691e-04
                    padj
               <numeric>
ILMN_1651498 0.001636334
ILMN_1651838 0.000115324
ILMN_1652287          NA
ILMN_1652464 0.002255105
ILMN_1652866 0.003676049
ILMN_1653447 0.011703411
```

```r
# Overview of result
#upregulated
sum(dds.results_u.h_ss$log2FoldChange > 0)
```

```
[1] 10
```

```r
#downregulated
sum(dds.results_u.h_ss$log2FoldChange < 0)
```

```
[1] 155
```

```r
# Plot

plot(dds.results_u$log2FoldChange, -log10(dds.results_u$padj), col =
c("gray","pink4", "blue")[(dds.results_u$padj < 0.05 &
abs(dds.results_u$log2FoldChange) > 2) + 1 ],  xlab = "Changes of gene
expression from Healthy to SS", ylab = "Significant level, FDR", cex = 0.8,
pch = 20)
title("GSE89632 - Healthy versus SS")
abline(v = c(-2, 2), col = "green")
abline(h = -log10(0.05), col = "green")
```

Compare healthy versus NASH

```
# get the result and summary
head(dds_u)
```

```
class: DESeqDataSet
dim: 6 29
metadata(1): version
assays(6): counts mu ... replaceCounts replaceCooks
rownames(6): ILMN_1343291 ILMN_1651209 ... ILMN_1651235 ILMN_1651236
rowData names(31): baseMean baseVar ... maxCooks replace
colnames(29): GSM2385720 GSM2385723 ... GSM2385771 GSM2385773
colData names(86): title geo_accession ... sizeFactor replaceable
```

```
dds.results_u_h_nash <- results(dds_u, contrast = c('stages', "3", "1"))
summary(dds.results_u_h_nash, alpha = 0.05) # p-value = 0.05
```

```


out of 29377 with nonzero total read count
adjusted p-value < 0.05
LFC > 0 (up)       : 227, 0.77%
LFC < 0 (down)     : 484, 1.6%
outliers [1]       : 292, 0.99%
low counts [2]     : 6259, 21%
(mean count < 414)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results
```

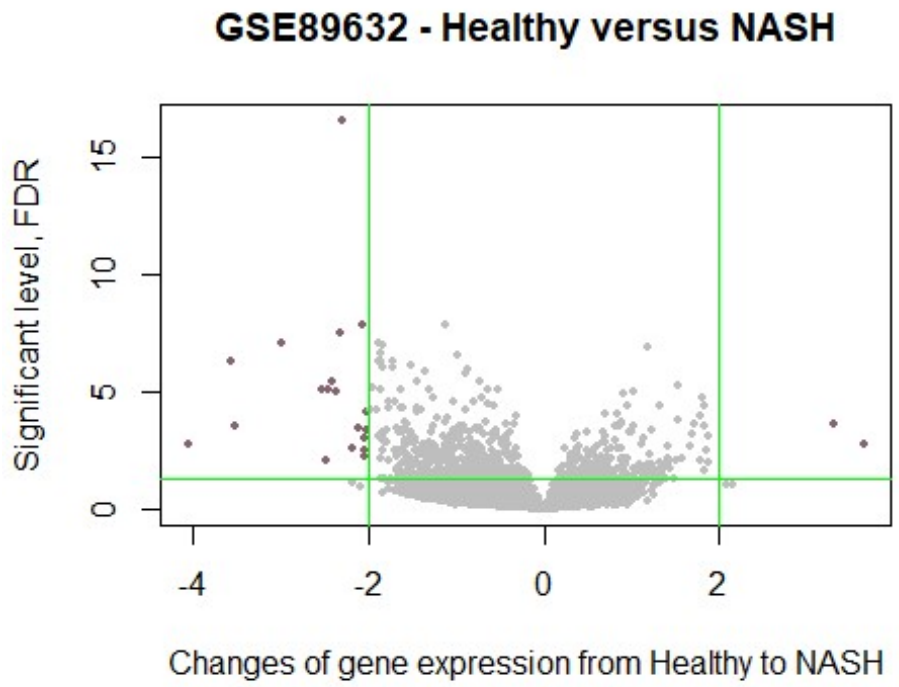
```r
# Only get differential expressed genes (p-val <= 0.05 and log2FC outside of
[-2,2])
dds.results_u.h_nash <- dds.results_u[!is.na(dds.results_u_h_nash$padj) &
dds.results_u_h_nash$padj <= 0.05 & dds.results_u_h_nash$log2FoldChange > 2 |
dds.results_u_h_nash$log2FoldChange < -2 ,]

# Overview of the result
#upregulated
sum(dds.results_u.h_nash$log2FoldChange > 0)
```

```
[1] 2
```

```r
#downregulated
sum(dds.results_u.h_nash$log2FoldChange < 0)
```

```
[1] 23
```

```r
# Plot
plot(dds.results_u_h_nash$log2FoldChange, -log10(dds.results_u_h_nash$padj),
col = c("gray","pink4", "blue")[(dds.results_u_h_nash$padj < 0.05 &
abs(dds.results_u_h_nash$log2FoldChange) > 2) + 1 ], xlab = "Changes of gene
expression from Healthy to NASH", ylab = "Significant level, FDR", cex = 0.8,
pch = 20)
title("GSE89632 - Healthy versus NASH")
abline(v = c(-2, 2), col = "green")
abline(h = -log10(0.05), col = "green")
```



Compare SS versus NASH

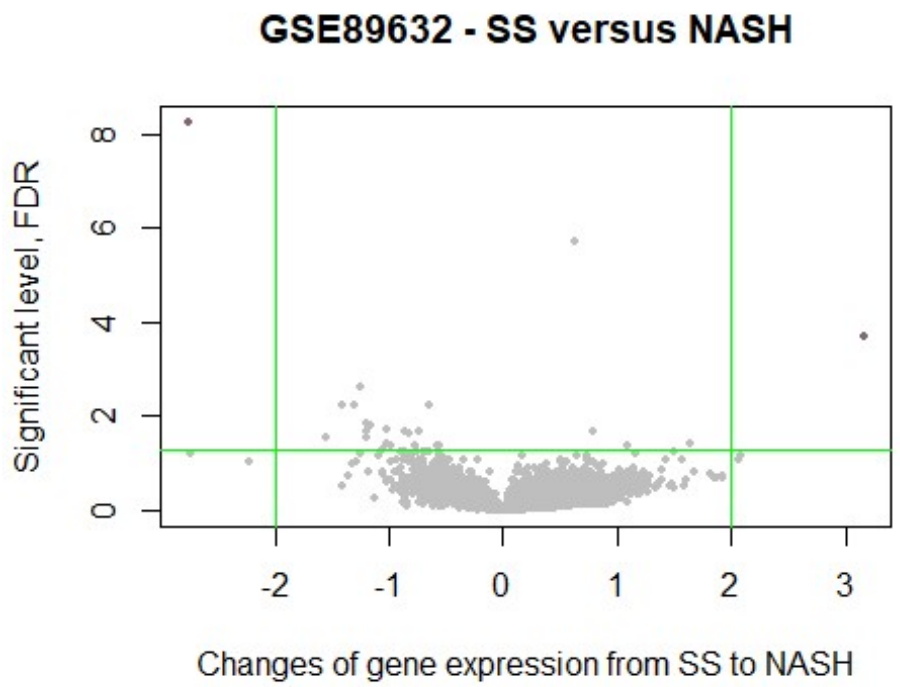
```
# get the result and summary of SS and NASH
dds.results_u_ss_nash <- results(dds_u, contrast = c('stages', "3", "2"))

# Only get differential expressed genes (p-val <= 0.05 and log2FC outside of
[-2,2])
dds.results_u.ss_nash <- dds.results_u[!is.na(dds.results_u_ss_nash$padj) &
dds.results_u_ss_nash$padj <= 0.05 & dds.results_u_ss_nash$log2FoldChange > 2
| dds.results_u_ss_nash$log2FoldChange < -2 ,]

# Overview of the result
#upregulated
sum(dds.results_u.ss_nash$log2FoldChange > 0)
```

```
[1] 2
```

```
#downregulated
sum(dds.results_u.ss_nash$log2FoldChange < 0)
```

```
[1] 2
```

```
# Plot
plot(dds.results_u_ss_nash$log2FoldChange, -
log10(dds.results_u_ss_nash$padj), col = c("gray","pink4",
"blue")[(dds.results_u_ss_nash$padj < 0.05 &
abs(dds.results_u_ss_nash$log2FoldChange) > 2) + 1 ], xlab = "Changes of gene
expression from SS to NASH", ylab = "Significant level, FDR", cex = 0.8, pch
= 20)
title("GSE89632 - SS versus NASH")
abline(v = c(-2, 2), col = "green")
abline(h = -log10(0.05), col = "green")
```



We have 2 upregulated and 2 downregulated genes that are differentially expressed in this sub-analysis.

Gene Identification Analysis

We want to find which molecular function pathway associated with those differentially expressed genes.

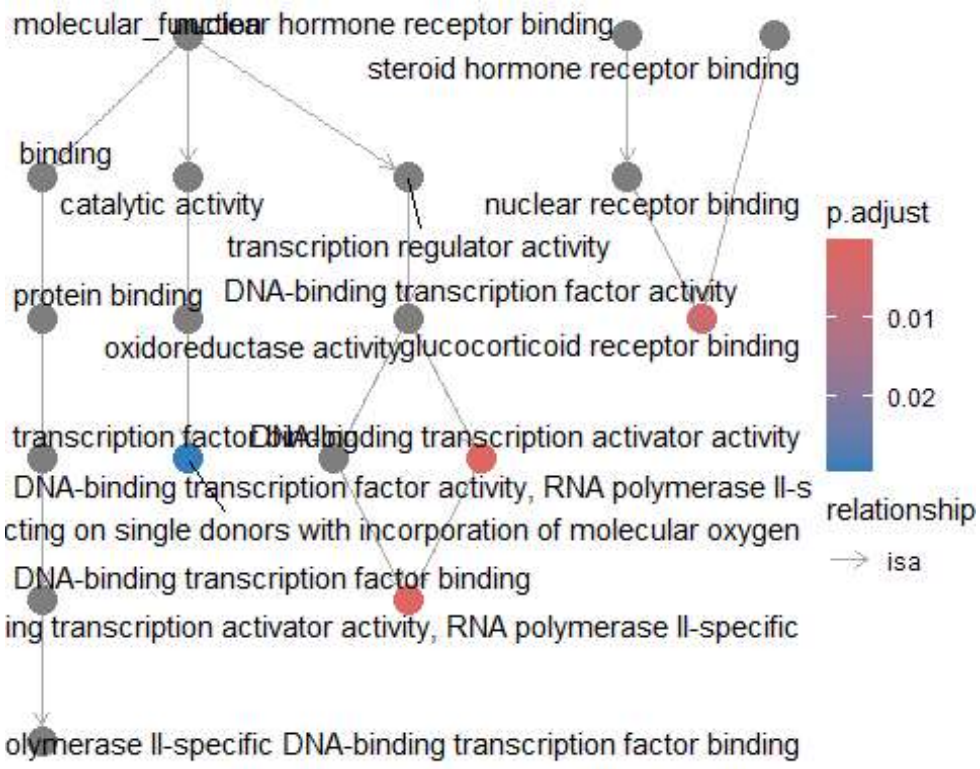
1. Healthy versus SS

```
# FDR <= 0.05 and foldchange outside abs 2
sigs_FC_u <- dds.results_u[!is.na(dds.results_u$padj) & dds.results_u$padj <
0.05 &( dds.results_u$log2FoldChange > 2 | dds.results_u$log2FoldChange < -2)
,]

sigs_FC_u$log2FoldChange < 0
```

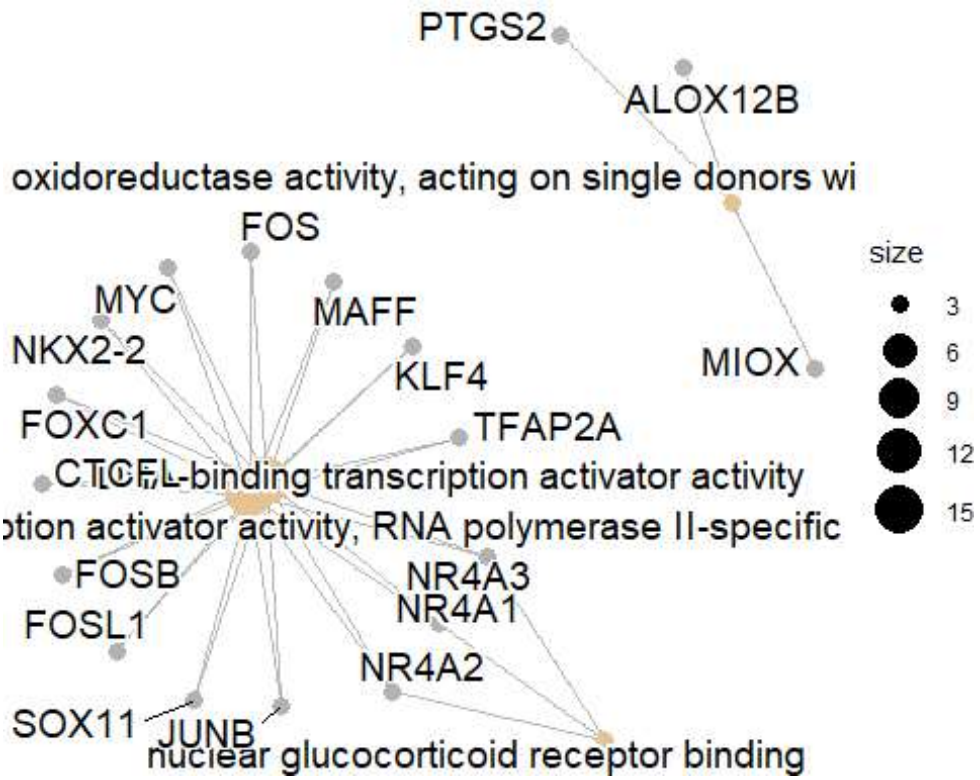
```
  [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [13] FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
 [25]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
 [37]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [49]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
 [61]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [73]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [85]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [97]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[109]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
```

```
[121]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[133]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

```
genes_sigs_FC_u <- rownames(sigs_FC_u)


# Gene identification
genes_h_ss_1 <- data.frame(Gene=unlist(mget(x = genes_sigs_FC_u,envir =
illuminaHumanv4SYMBOL)))

go_h_ss_1 <- enrichGO(gene = genes_h_ss_1$Gene, OrgDb = "org.Hs.eg.db",
keyType = "SYMBOL", ont = "MF")

# export the supplemental table A1
write.csv(go_h_ss_1@result,"C:\\Users\\nklin\\Downloads\\spring 24\\DA
401\\table_A1_1_h_ss.csv", row.names = FALSE)

goplot(go_h_ss_1)
```




```
cnetplot(go_h_ss_1, showCategory = 12)
```

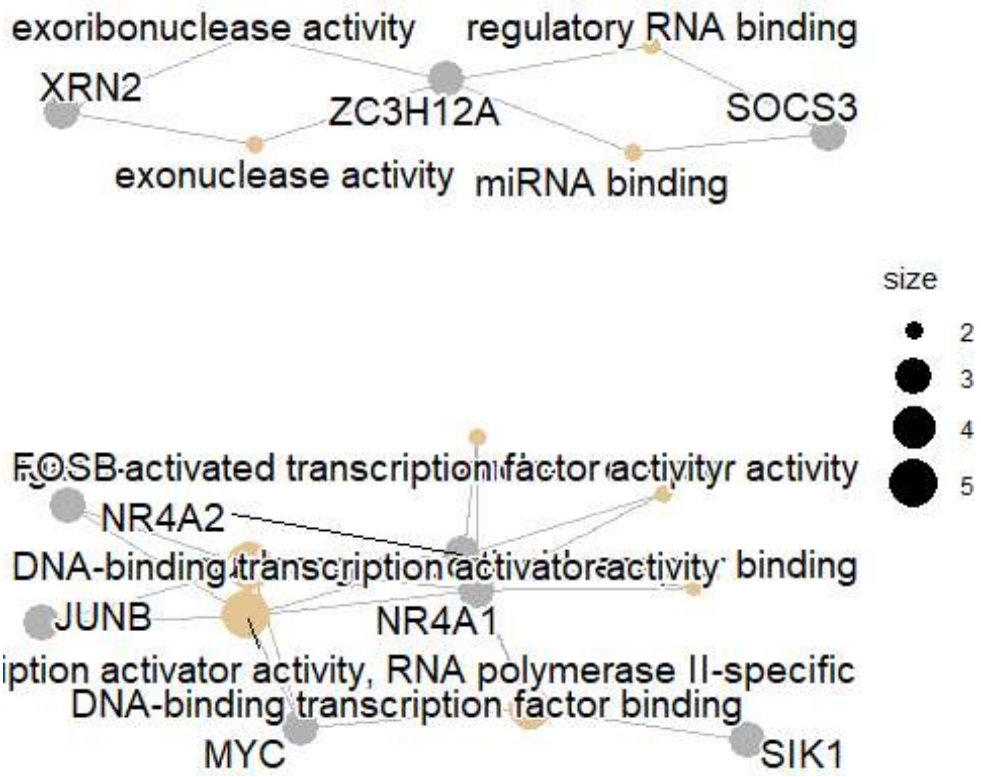


Bind gene names and log2fc

```
h_ss_1_genes <- merge(data.frame(dds.results_u), genes_h_ss_1, by = 0)

# export the supplemental table A.1.2
write.csv(h_ss_1_genes,"C:\\Users\\nklin\\Downloads\\spring 24\\DA
401\\table_A1_1_2_h_ss.csv", row.names = FALSE)
```

2. Healthy versus NASH

```
# FDR <= 0.05 and foldchange outside abs 2
sigs_FC_u_h_nash <- dds.results_u_h_nash[!is.na(dds.results_u_h_nash$padj) &
dds.results_u_h_nash$padj <= 0.05 &( dds.results_u_h_nash$log2FoldChange > 2
| dds.results_u_h_nash$log2FoldChange < -2) ,]
genes_sigs_FC_u_h_nash <- rownames(sigs_FC_u_h_nash)


# Gene identification
genes_h_nash_1 <- data.frame(Gene=unlist(mget(x =
genes_sigs_FC_u_h_nash,envir = illuminaHumanv4SYMBOL)))

# Gene identification

go_h_nash_1 <- enrichGO(gene = genes_h_nash_1$Gene, OrgDb = "org.Hs.eg.db",
keyType = "SYMBOL", ont = "MF")

cnetplot(go_h_nash_1, showCategory = 12)
```



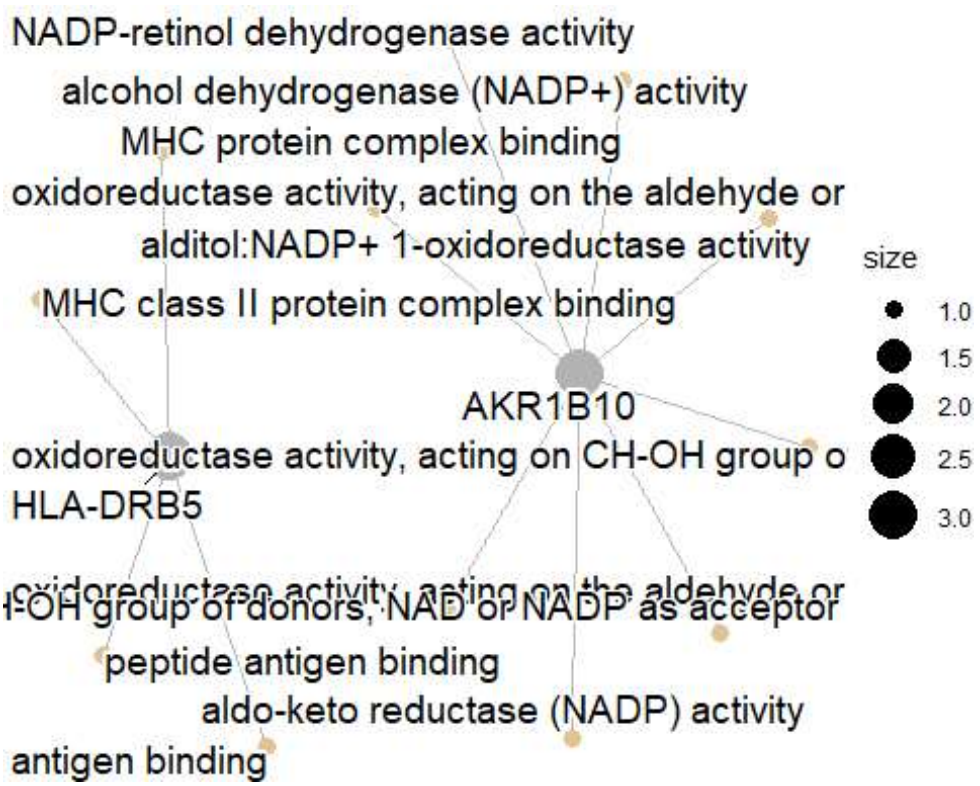
Export table A2

```
write.csv(go_h_nash_1@result,"C:\\Users\\nklin\\Downloads\\spring 24\\DA
401\\table_A2_1_h_nash.csv", row.names = FALSE)

h_nash_1_genes <- merge(data.frame(sigs_FC_u_h_nash), genes_h_nash_1, by = 0)

# export the supplemental table A.1.2
write.csv(h_nash_1_genes,"C:\\Users\\nklin\\Downloads\\spring 24\\DA
401\\table_A1_1_2_h_nash.csv", row.names = FALSE)
```

3. SS vs NASH

```
# FDR <= 0.05 and foldchange outside abs 2
sigs_FC_u_ss_nash <- dds.results_u_ss_nash[!is.na(dds.results_u_ss_nash$padj)
& dds.results_u_ss_nash$padj < 0.05 &( dds.results_u_ss_nash$log2FoldChange >
2 | dds.results_u_ss_nash$log2FoldChange < -2) ,]
genes_sigs_FC_u_ss_nash <- rownames(sigs_FC_u_ss_nash)



# Gene identification
genes_ss_nash_1 <- data.frame(Gene=unlist(mget(x =
genes_sigs_FC_u_ss_nash,envir = illuminaHumanv4SYMBOL)))

# Gene identification

go_ss_nash_1 <- enrichGO(gene = genes_ss_nash_1$Gene, OrgDb = "org.Hs.eg.db",
keyType = "SYMBOL", ont = "MF")
```

```
cnetplot(go_ss_nash_1, showCategory = 12)
```




```
ss_nash_1_genes <- merge(data.frame(sigs_FC_u_ss_nash), genes_ss_nash_1, by =
0)

# export the supplemental table A.1.2
write.csv(h_nash_1_genes,"C:\\Users\\nklin\\Downloads\\spring 24\\DA
401\\table_A1_1_2_h_nash.csv", row.names = FALSE)
```

4. Similar genes between these analyses for this dataset

```
sigs_h_nafld <- intersect(genes_sigs_FC_u_h_nash, genes_sigs_FC_u)
sigs_h_nafld_0 <- rbind(sigs_FC_u[row.names(sigs_FC_u) %in% sigs_h_nafld, ],
sigs_FC_u_h_nash[row.names(sigs_FC_u_h_nash) %in% sigs_h_nafld, ] )

genes_h_nafld <- data.frame(Gene=unlist(mget(x =
unique(rownames(sigs_h_nafld_0)),envir = illuminaHumanv4SYMBOL)))
genes_h_nafld <- as.data.frame(na.omit(genes_h_nafld))


genes_h_nafld_1 <- merge(as.data.frame(sigs_FC_u[row.names(sigs_FC_u) %in%
sigs_h_nafld, ]), genes_h_nafld, by = "row.names")

genes_h_nafld_2 <-
merge(as.data.frame(sigs_FC_u_h_nash[row.names(sigs_FC_u_h_nash) %in%
sigs_h_nafld, ]), genes_h_nafld, by = "row.names")
```

```
genes_h_nafld_all <- rbind(genes_h_nafld_1, genes_h_nafld_2)
```

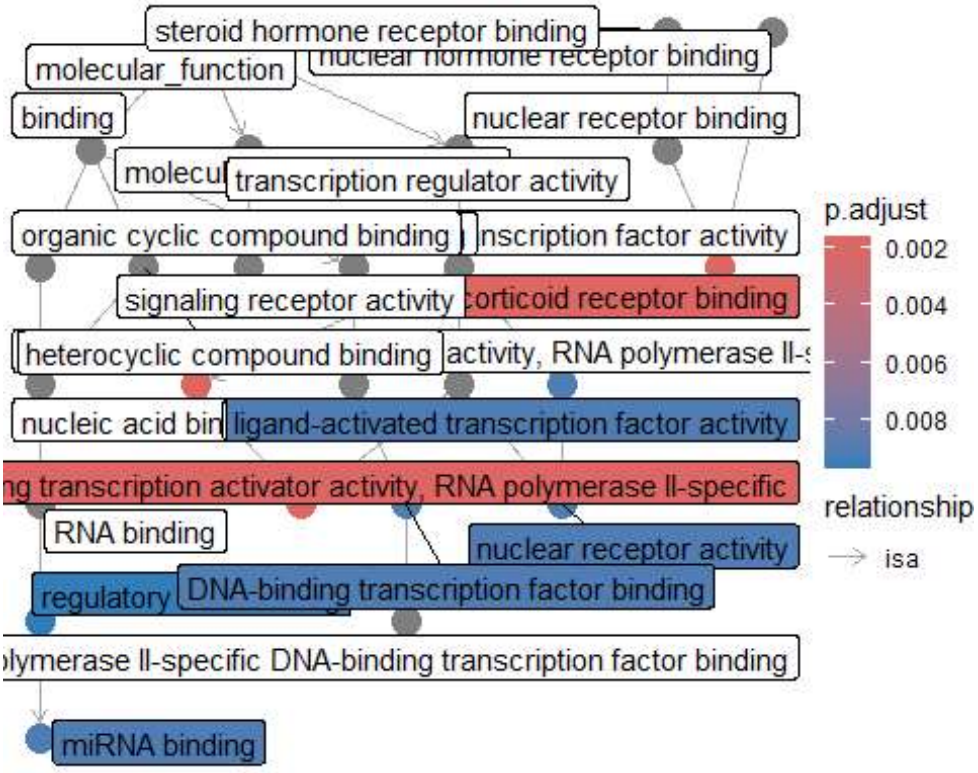
Molecular function pathway of these similar genes

```
# run goseq, find molecular function pathways
GO_results_sigs <- enrichGO(gene = unique(genes_h_nafld$Gene), OrgDb =
"org.Hs.eg.db",  keyType = "SYMBOL", ont = "MF")

# check results
dfGO_sigs <- data.frame(GO_results_sigs@result) # transform the result file
to dataframe
head(dfGO_sigs)
```

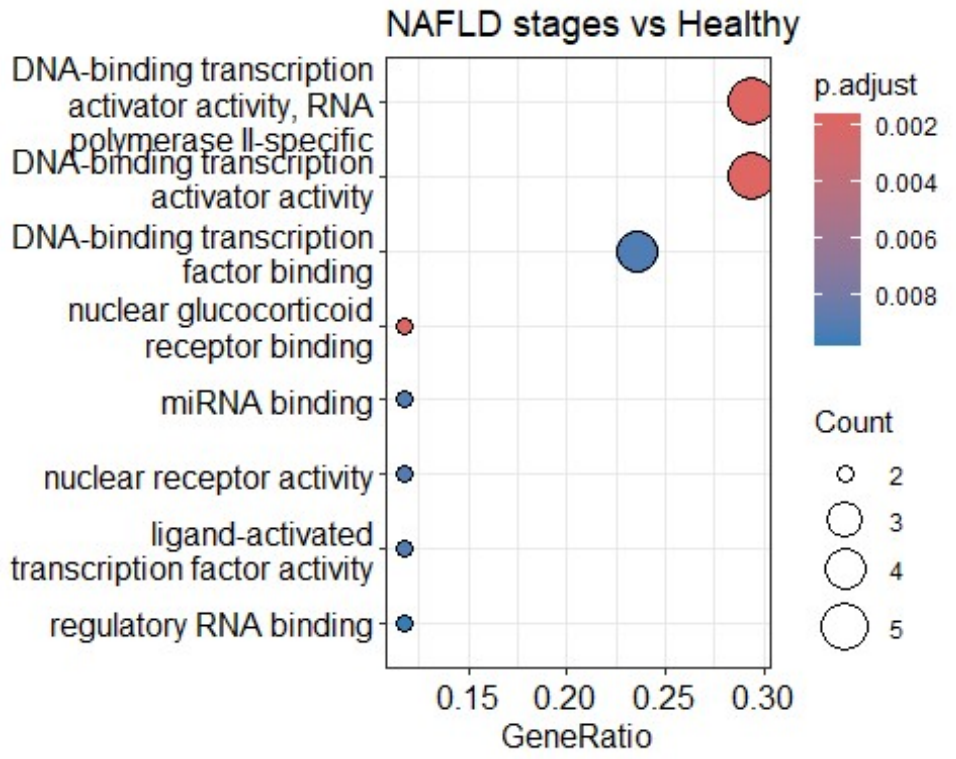
```
                   ID
GO:0001228 GO:0001228
GO:0001216 GO:0001216
GO:0035259 GO:0035259
GO:0035198 GO:0035198
GO:0004879 GO:0004879
GO:0098531 GO:0098531

Description
GO:0001228 DNA-binding transcription activator activity, RNA polymerase II-
specific
GO:0001216                              DNA-binding transcription activator
activity
GO:0035259                                   nuclear glucocorticoid receptor
binding
GO:0035198                                                          miRNA
binding
GO:0004879                                                 nuclear receptor
activity
GO:0098531                            ligand-activated transcription factor
activity
           GeneRatio   BgRatio       pvalue    p.adjust       qvalue
GO:0001228      5/17 468/18369 5.046270e-05 0.001625130 0.0009649895
GO:0001216      5/17 472/18369 5.254974e-05 0.001625130 0.0009649895
GO:0035259      2/17  13/18369 6.250500e-05 0.001625130 0.0009649895
GO:0035198      2/17  37/18369 5.267705e-04 0.009171301 0.0054458468
GO:0004879      2/17  46/18369 8.146316e-04 0.009171301 0.0054458468
GO:0098531      2/17  46/18369 8.146316e-04 0.009171301 0.0054458468
                           geneID Count
GO:0001228 NR4A1/MYC/FOSB/JUNB/NR4A2     5
GO:0001216 NR4A1/MYC/FOSB/JUNB/NR4A2     5
GO:0035259               NR4A1/NR4A2     2
GO:0035198             ZC3H12A/SOCS3     2
GO:0004879               NR4A1/NR4A2     2
GO:0098531               NR4A1/NR4A2     2
```

Draw the plot

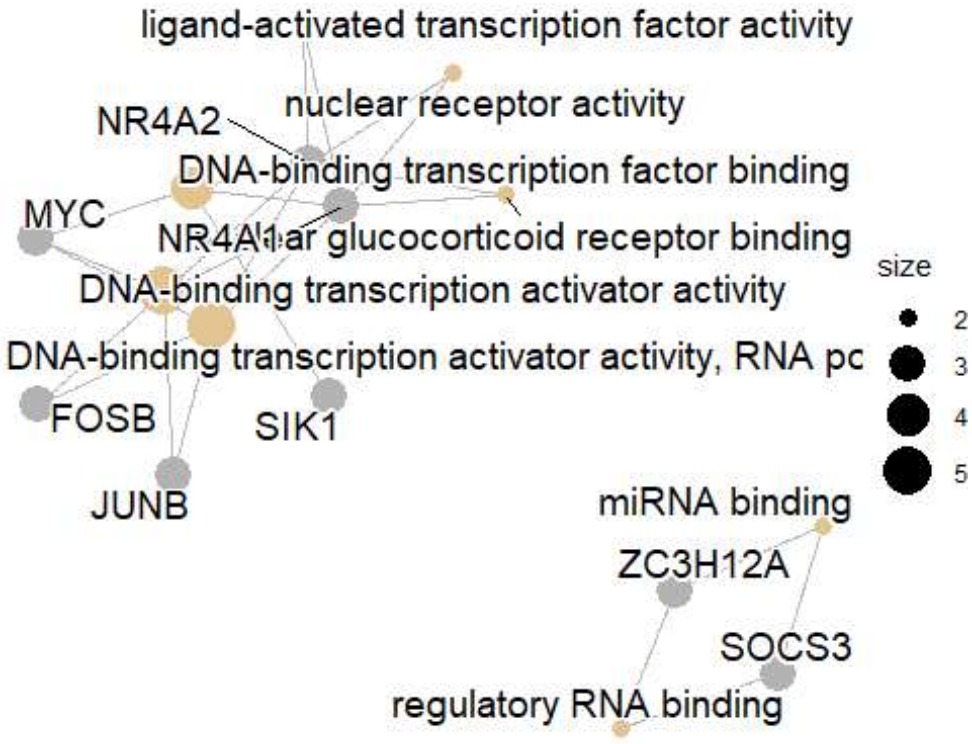
```
goplot(GO_results_sigs, geom = 'label')
```




```
dotplot(GO_results_sigs, showCategory = 20) + ggtitle("NAFLD stages vs
Healthy")
```




```
cnetplot(GO_results_sigs, showCategory = 12)
```



Subudhi et al

1. Get data to analyze

```
# Values for diabetes
data_u2u <- data_u2[,
!is.na(data_u2@phenoData@data[["characteristics_ch1.5"]]) &
data_u2@phenoData@data[["characteristics_ch1.5"]] == "diabetes: Yes" |
data_u2@phenoData@data[["characteristics_ch1.5"]] == "diabetes: No" ]

data_u2u$diabetes <-
ifelse(data_u2u@phenoData@data[["characteristics_ch1.5"]] == "diabetes: No",
0, 1)
```

We still also have to back transform the expression values of this dataset - but the method is different compared to the previous one. We will also re-code values in the stage column for easy interpretation

backtransform the expression values

```
exprs_data_u2 <- round(data_u2u@assayData$exprs,0)

data_u2u$stages <- ifelse(data_u2u$`nafld stage:ch1` == "Normal", '1',
                          ifelse(data_u2u$`nafld stage:ch1` == "Steatosis",
'2', '3'))
```

2. Fit DeSEQ2

```
# Fit DESeq2
dds_u2 <- DESeqDataSetFromMatrix(countData = exprs_data_u2, colData =
pData(data_u2u), design=~stages+diabetes)

converting counts to integer mode

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

  the design formula contains one or more numeric variables with integer
values,
  specifying a model with increasing fold change for higher values.
  did you mean for this to be a factor? if so, first convert
  this variable to a factor using the factor() function

dds_u2 <- DESeq(dds_u2)

estimating size factors

estimating dispersions

gene-wise dispersion estimates
```

```
mean-dispersion relationship

final dispersion estimates

fitting model and testing

-- replacing outliers and refitting for 61 genes
-- DESeq argument 'minReplicatesForReplace' = 7
-- original counts are preserved in counts(dds)

estimating dispersions

fitting model and testing
```

Healthy vs ss

```
# head(dds_u2)



# normal vs ss
dds.results_u2_h_ss <- results(dds_u2, contrast = c('stages', '2', '1'))
summary(dds.results_u2_h_ss, alpha = 0.05) # p-value = 0.05


out of 800 with nonzero total read count
adjusted p-value < 0.05
LFC > 0 (up)       : 7, 0.88%
LFC < 0 (down)     : 0, 0%
outliers [1]       : 0, 0%
low counts [2]     : 16, 2%
(mean count < 9)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results

# Only get differential expressed genes (p-val <= 0.05) - we did not consider
log2FC, though
dds_u2_h_ss <- dds.results_u2_h_ss[!is.na(dds.results_u2_h_ss$padj) &
dds.results_u2_h_ss$padj <= 0.05 & dds.results_u2_h_ss$log2FoldChange > .5 |
dds.results_u2_h_ss$log2FoldChange < -.5 ,]



head(dds_u2_h_ss)

log2 fold change (MLE): stages 2 vs 1
Wald test p-value: stages 2 vs 1
DataFrame with 4 rows and 6 columns
        baseMean log2FoldChange     lfcSE      stat      pvalue       padj
       <numeric>      <numeric> <numeric> <numeric>   <numeric>  <numeric>
CTSG     23.4556       0.605882  0.138166   4.38517 1.15898e-05 0.00302879
EGR1    294.0780      -0.848085  0.255523  -3.31902 9.03330e-04 0.07709536
IGFBP1 1834.4655      -0.528878  0.260498  -2.03026 4.23304e-02 0.35727053
TPSAB1  167.0480       0.589854  0.132370   4.45610 8.34655e-06 0.00302879
```

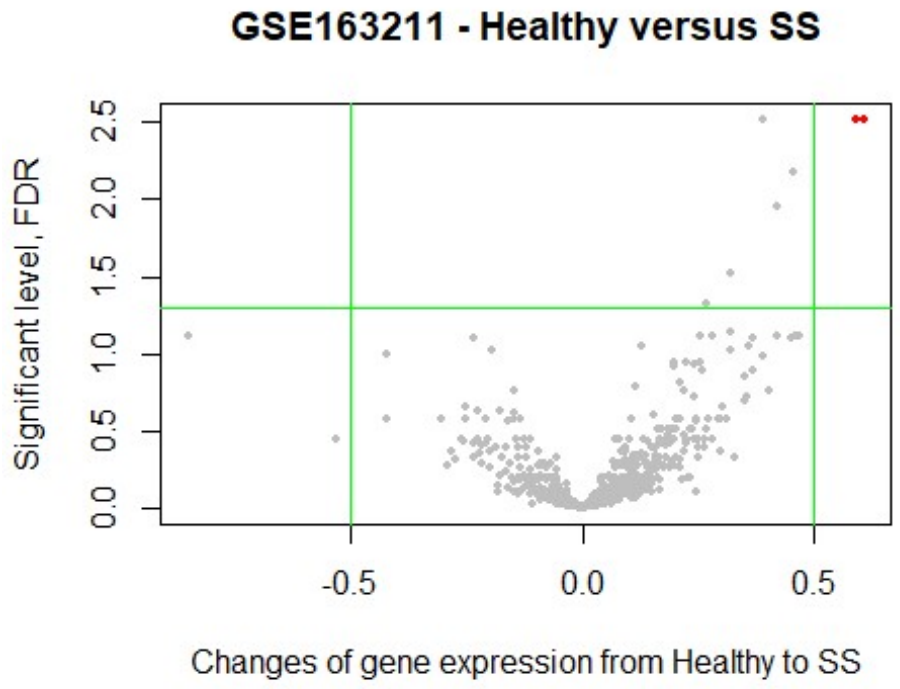
```
# Overview of result
#upregulated
sum(dds_u2_h_ss$log2FoldChange > 0)
```

```
[1] 2
```

```
#downregulated
sum(dds_u2_h_ss$log2FoldChange < 0)
```

```
[1] 2
```

```
# Plot

plot(dds.results_u2_h_ss$log2FoldChange, -log10(dds.results_u2_h_ss$padj),
col = c("gray","red", "blue")[(dds.results_u2_h_ss$padj < 0.05 &
abs(dds.results_u2_h_ss$log2FoldChange) > 0.5) + 1 ], xlab = "Changes of gene
expression from Healthy to SS", ylab = "Significant level, FDR", cex = 0.8,
pch = 20)
title("GSE163211 - Healthy versus SS")
abline(v = c(-0.5, .5), col = "green")
abline(h = -log10(0.05), col = "green")
```



Healthy vs nash

```
# normal vs ss
dds.results_u2_h_nash <- results(dds_u2, contrast = c('stages', '3', '1'))
summary(dds.results_u2_h_nash, alpha = 0.05) # p-value = 0.05
```

```
out of 800 with nonzero total read count
adjusted p-value < 0.05
LFC > 0 (up)       : 82, 10%
LFC < 0 (down)     : 41, 5.1%
outliers [1]       : 0, 0%
low counts [2]     : 124, 16%
(mean count < 17)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results
```

```
# Only get differential expressed genes (p-val <= 0.05) - we did not consider
log2FC, though
dds_u2_h_nash <- dds.results_u2_h_nash[!is.na(dds.results_u2_h_nash$padj) &
dds.results_u2_h_nash$padj <= 0.05 & dds.results_u2_h_nash$log2FoldChange >
.5 | dds.results_u2_h_nash$log2FoldChange < -.5 ,]

head(dds_u2_h_ss)
```

```
log2 fold change (MLE): stages 2 vs 1
Wald test p-value: stages 2 vs 1
DataFrame with 4 rows and 6 columns
        baseMean log2FoldChange     lfcSE      stat      pvalue       padj
       <numeric>      <numeric> <numeric> <numeric>   <numeric>  <numeric>
CTSG     23.4556       0.605882  0.138166   4.38517 1.15898e-05 0.00302879
EGR1    294.0780      -0.848085  0.255523  -3.31902 9.03330e-04 0.07709536
IGFBP1 1834.4655      -0.528878  0.260498  -2.03026 4.23304e-02 0.35727053
TPSAB1  167.0480       0.589854  0.132370   4.45610 8.34655e-06 0.00302879
```

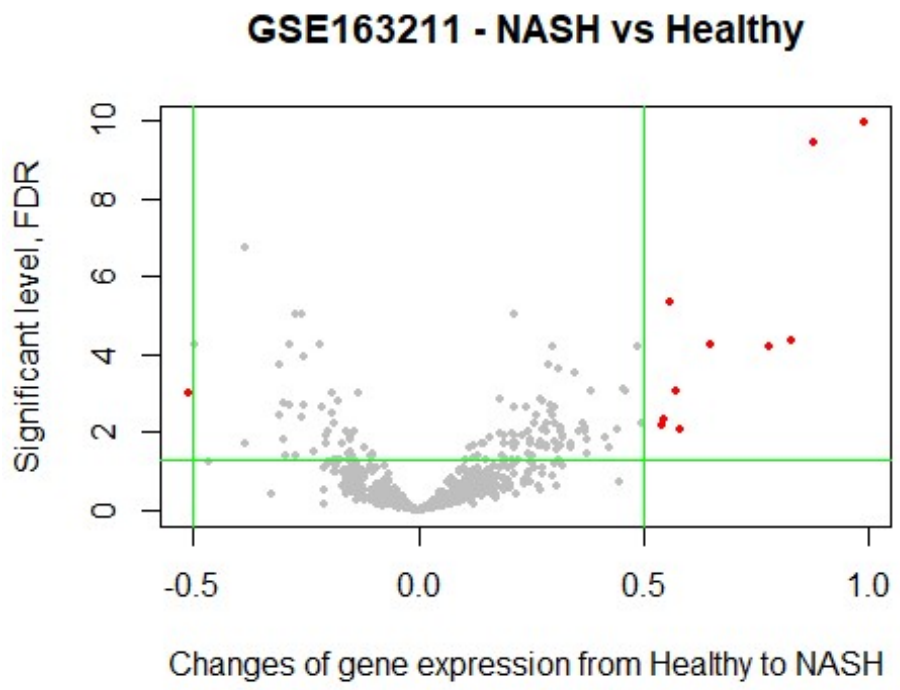
```
# Overview of result
#upregulated
sum(dds_u2_h_nash$log2FoldChange > 0)
```

```
[1] 10
```

```
#downregulated
sum(dds_u2_h_nash$log2FoldChange < 0)
```

```
[1] 1
```

```
# Plot

plot(dds.results_u2_h_nash$log2FoldChange, -
log10(dds.results_u2_h_nash$padj), col = c("gray","red",
"blue")[(dds.results_u2_h_nash$padj < 0.05 &
abs(dds.results_u2_h_nash$log2FoldChange) > 0.5) + 1 ], xlab = "Changes of
gene expression from Healthy to NASH", ylab = "Significant level, FDR", cex =
0.8, pch = 20)
abline(v = c(-0.5, .5), col = "green")
abline(h = -log10(0.05), col = "green")
title("GSE163211 - NASH vs Healthy")
```



SS vs NASH

```
# ss vs nash
dds.results_u2_ss_nash <- results(dds_u2, contrast = c('stages', '3', '2'))
summary(dds.results_u2_ss_nash, alpha = 0.05) # p-value = 0.05


out of 800 with nonzero total read count
adjusted p-value < 0.05
LFC > 0 (up)       : 83, 10%
LFC < 0 (down)     : 89, 11%
outliers [1]       : 0, 0%
low counts [2]     : 0, 0%
(mean count < 7)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results

# Only get differential expressed genes (p-val <= 0.05) - we did not consider
log2FC, though
dds_u2_ss_nash <- dds.results_u2_ss_nash[!is.na(dds.results_u2_ss_nash$padj)
& dds.results_u2_ss_nash$padj <= 0.05 & dds.results_u2_ss_nash$log2FoldChange
> .5 | dds.results_u2_ss_nash$log2FoldChange < -.5  ,]

head(dds_u2_ss_nash)

log2 fold change (MLE): stages 3 vs 2
Wald test p-value: stages 3 vs 2
DataFrame with 6 rows and 6 columns
```

```
        baseMean log2FoldChange     lfcSE      stat      pvalue        padj
       <numeric>      <numeric> <numeric> <numeric>   <numeric>   <numeric>
COL1A1  214.7989       0.600397  0.152336   3.94128 8.10496e-05 1.90705e-03
CXCL9   138.6186       0.563011  0.113319   4.96838 6.75151e-07 9.00201e-05
EGR1    294.0780       1.294711  0.220830   5.86294 4.54750e-09 7.27600e-07
FGF21    99.9763       0.565766  0.149281   3.78994 1.50686e-04 2.97174e-03
JUN     110.9270       0.836661  0.121620   6.87928 6.01544e-12 2.40618e-09
KLF6    319.2813       0.776580  0.131251   5.91676 3.28348e-09 6.56695e-07
```

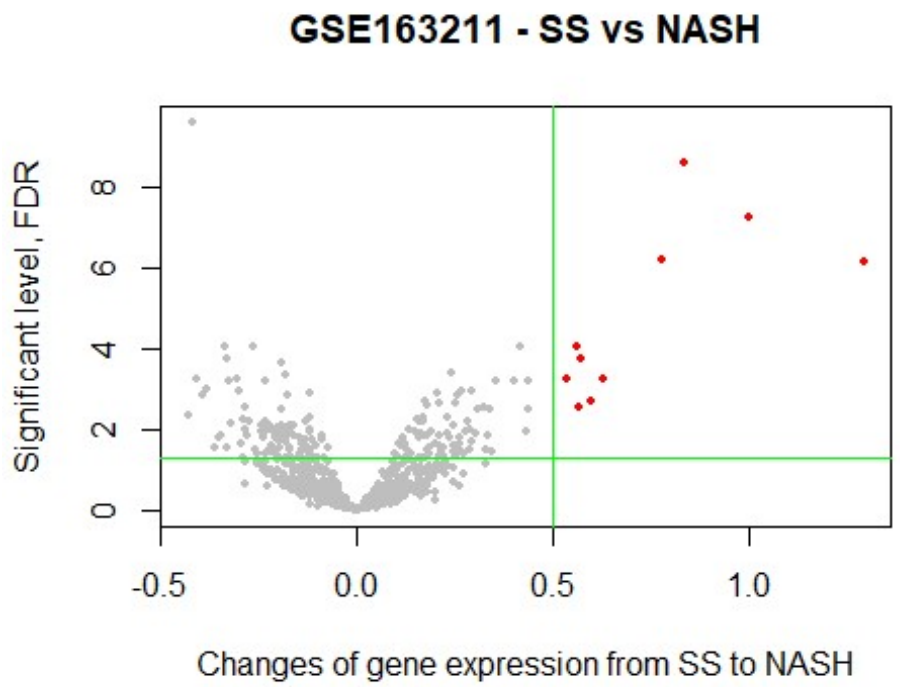
```
# Overview of result
#upregulated
sum(dds_u2_ss_nash$log2FoldChange > 0)
```

```
[1] 10
```

```
#downregulated
sum(dds_u2_ss_nash$log2FoldChange < 0)
```

```
[1] 0
```

```
# Plot

plot(dds.results_u2_ss_nash$log2FoldChange, -
log10(dds.results_u2_ss_nash$padj), col = c("gray","red",
"blue")[(dds.results_u2_ss_nash$padj < 0.05 &
abs(dds.results_u2_ss_nash$log2FoldChange) > 0.5) + 1 ], xlab = "Changes of
gene expression from SS to NASH", ylab = "Significant level, FDR", cex = 0.8,
pch = 20)
abline(v = c(0.5), col = "green")
abline(h = -log10(0.05), col = "green")
title("GSE163211 - SS vs NASH")
```



2. Comparison

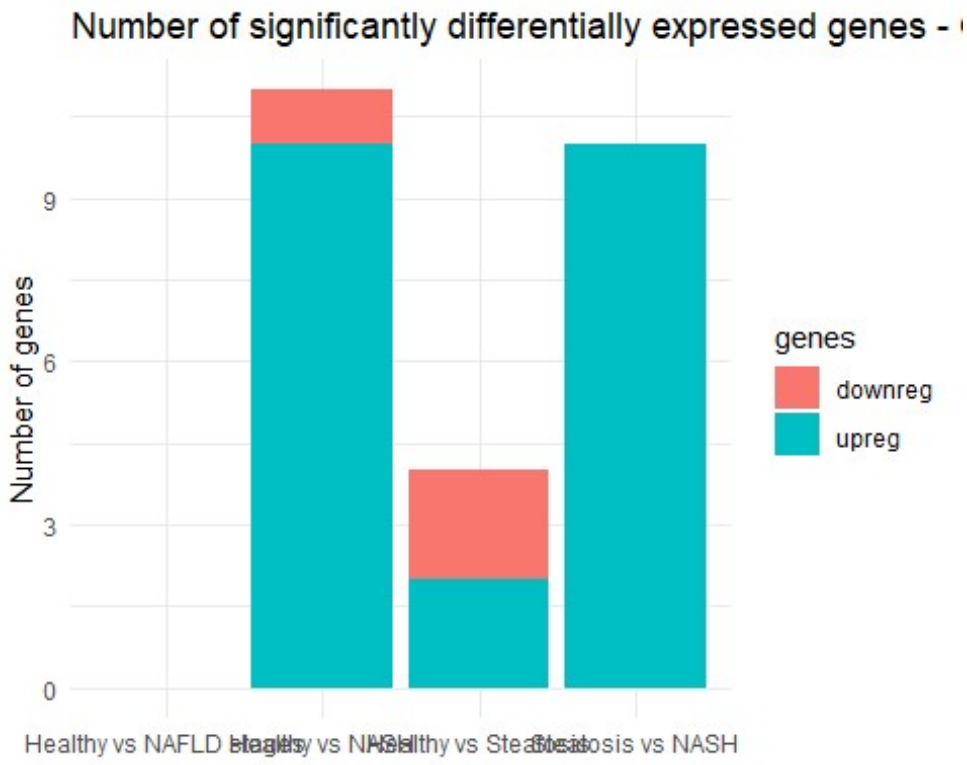
```
sigs_h_nafld_2 <- intersect(rownames(dds_u2_h_ss), rownames(dds_u2_h_nash))
sigs_h_nafld_2 <- rbind(dds.results_u2_h_ss[row.names(dds.results_u2_h_ss)
%in% sigs_h_nafld_2, ],

dds.results_u2_h_nash[row.names(dds.results_u2_h_nash) %in% sigs_h_nafld_2, ]
)

genes_h_nafld <- data.frame(Gene=unlist(mget(x =
unique(rownames(sigs_h_nafld)),envir = illuminaHumanv4SYMBOL)))
```

2.4 All genes together

```
deg_sum <- data.frame(
  category = c("Healthy vs Steatosis", "Healthy vs NASH", "Steatosis vs
NASH", "Healthy vs NAFLD stages"),
  count = c(4, 11, 10, 0),
  upreg = c(sum(dds_u2_h_ss$log2FoldChange > 0 ),
sum(dds_u2_h_nash$log2FoldChange > 0), sum(dds_u2_ss_nash$log2FoldChange >0),
0 ),
  downreg = c(sum(dds_u2_h_ss$log2FoldChange < 0 ),
sum(dds_u2_h_nash$log2FoldChange  < 0), sum(dds_u2_ss_nash$log2FoldChange
<0), 0 )
  )

library(tidyr)
```

```
Warning: package 'tidyr' was built under R version 4.3.2


Attaching package: 'tidyr'

The following objects are masked from 'package:Matrix':

    expand, pack, unpack

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

    expand
```

```r
df_long <- pivot_longer(deg_sum, cols = c(downreg,upreg), names_to = "genes",
values_to = "value")

# Create stacked bar chart
ggplot(df_long, aes(x = category, y = value, fill = genes)) +
  geom_bar(stat = "identity", position = "stack") +
  labs(title = "Number of significantly differentially expressed genes -
GSE163211",
       x = NULL,
       y = "Number of genes") +
  theme_minimal()
```




```r
  geom_text(aes(label = value),
            color = "black", size =4)
```

```
mapping: label = ~value
geom_text: parse = FALSE, check_overlap = FALSE, size.unit = mm, na.rm =
FALSE
stat_identity: na.rm = FALSE
position_identity
```

3. Gene identification

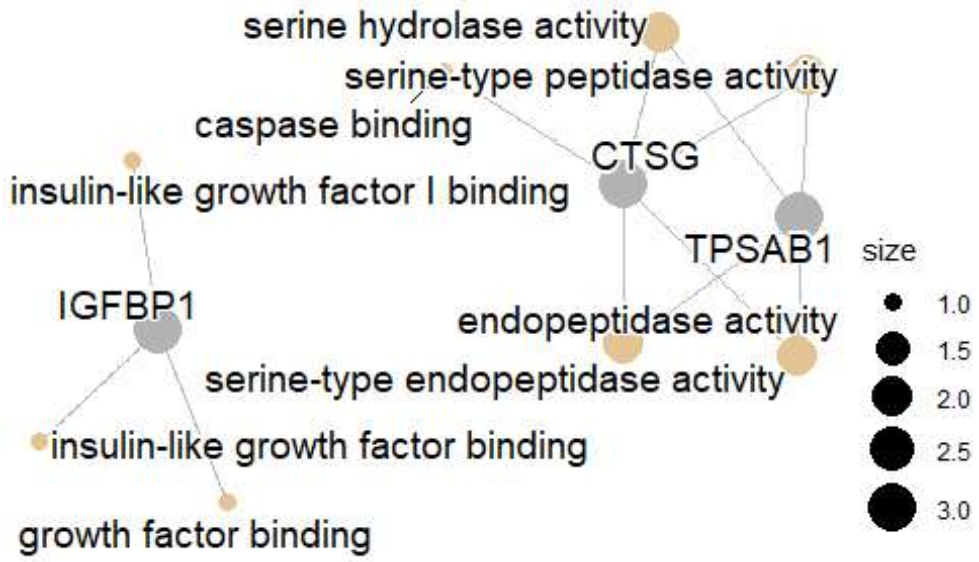
For Healthy and SS

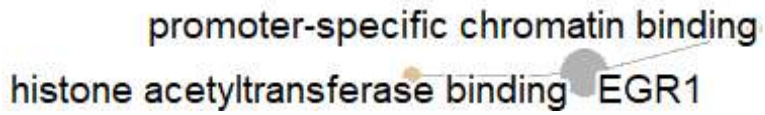
cluster of genes

```
mf_u2_h_ss <- enrichGO(gene  = rownames(dds_u2_h_ss), keyType = 'SYMBOL',
OrgDb = org.Hs.eg.db, ont = "MF")

# check results
mf_u2_h_ss.df <- data.frame(mf_u2_h_ss@result) # transform the result file to
dataframe
head(mf_u2_h_ss.df)
```

```
                   ID                          Description GeneRatio
BgRatio
GO:0004252 GO:0004252   serine-type endopeptidase activity       2/4
170/18369
GO:0008236 GO:0008236       serine-type peptidase activity       2/4
190/18369
GO:0017171 GO:0017171            serine hydrolase activity       2/4
194/18369
GO:0031994 GO:0031994 insulin-like growth factor I binding       1/4
13/18369
GO:0089720 GO:0089720                     caspase binding       1/4
14/18369
GO:0004175 GO:0004175               endopeptidase activity       2/4
428/18369
                pvalue    p.adjust      qvalue      geneID Count
GO:0004252 0.0005046946 0.003720649 0.000921523 CTSG/TPSAB1     2
GO:0008236 0.0006299039 0.003720649 0.000921523 CTSG/TPSAB1     2
GO:0017171 0.0006565851 0.003720649 0.000921523 CTSG/TPSAB1     2
GO:0031994 0.0028280833 0.008925875 0.002210743      IGFBP1     1
GO:0089720 0.0030453794 0.008925875 0.002210743        CTSG     1
GO:0004175 0.0031503089 0.008925875 0.002210743 CTSG/TPSAB1     2
```

```
cnetplot(mf_u2_h_ss,showCategory=10)
```





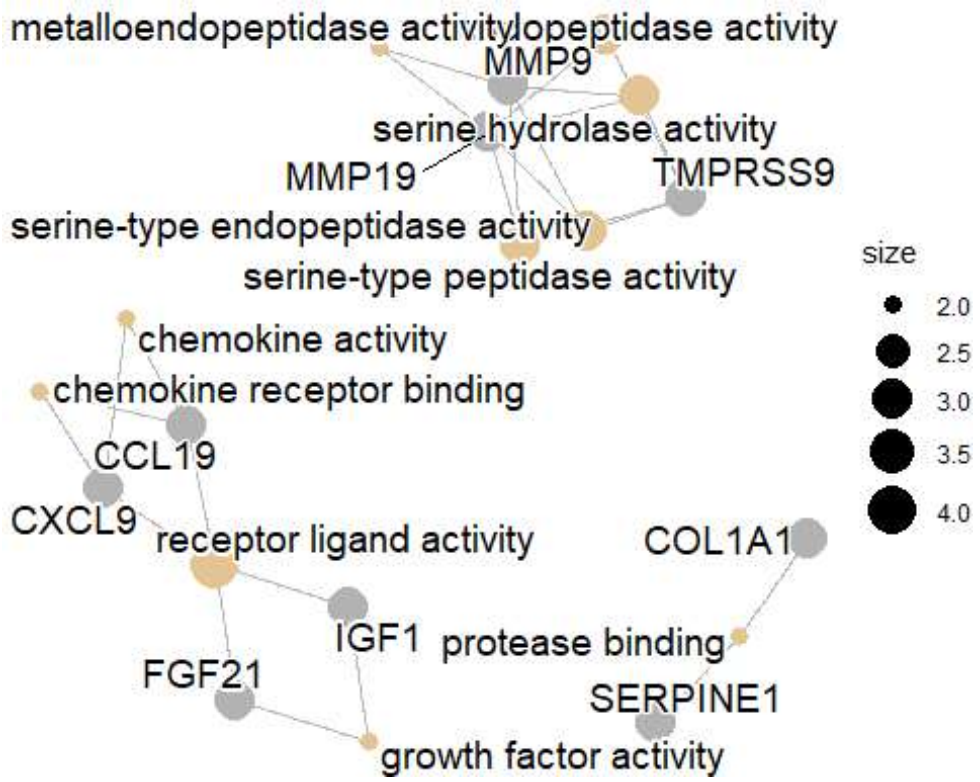

```
h_ss_2_genes <- merge(data.frame(dds_u2_h_ss), mf_u2_h_ss, by = 0)

# export the supplemental table A.1.2
write.csv(h_ss_1_genes,"C:\\Users\\nklin\\Downloads\\spring 24\\DA
401\\table_A1_1_2_h_ss.csv", row.names = FALSE)
```

From healthy to nash

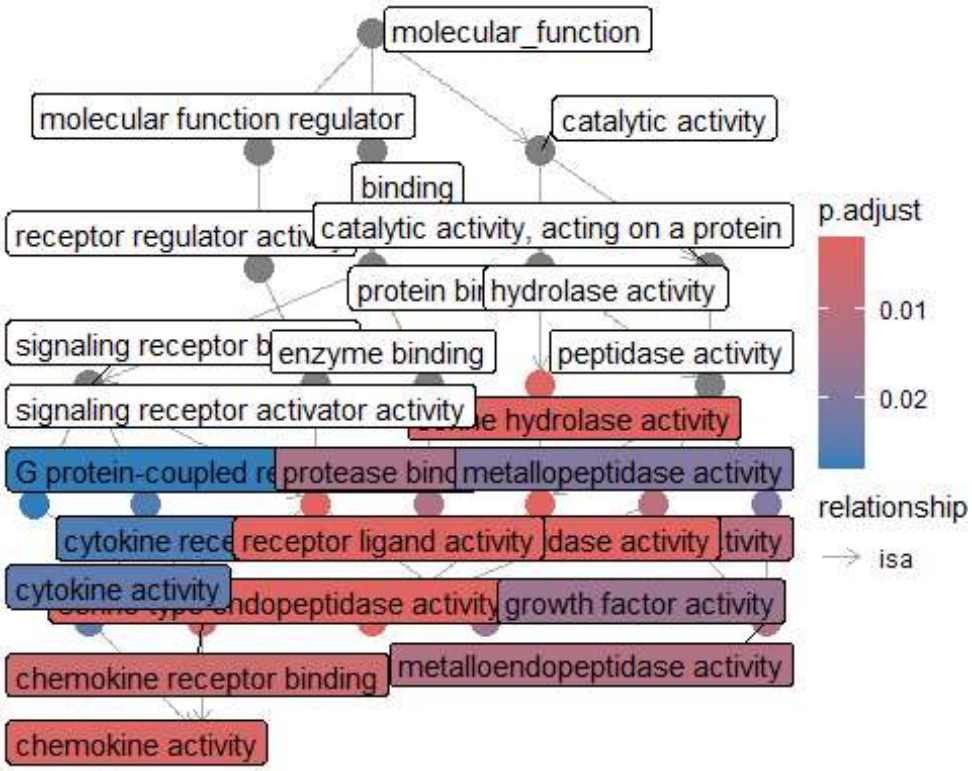
```
mf_u2_h_nash <- enrichGO(gene  = rownames(dds_u2_h_nash), keyType = 'SYMBOL',
OrgDb = org.Hs.eg.db, ont = "MF")

# check results
mf_u2_h_nash.df <- data.frame(mf_u2_h_nash@result) # transform the result
file to dataframe
head(mf_u2_h_nash.df)
```

```
                   ID                         Description GeneRatio   BgRatio
GO:0004252 GO:0004252 serine-type endopeptidase activity      3/11 170/18369
GO:0048018 GO:0048018           receptor ligand activity      4/11 497/18369
GO:0008236 GO:0008236     serine-type peptidase activity      3/11 190/18369
GO:0017171 GO:0017171          serine hydrolase activity      3/11 194/18369
GO:0008009 GO:0008009                 chemokine activity      2/11  49/18369
GO:0042379 GO:0042379         chemokine receptor binding      2/11  74/18369
                  pvalue    p.adjust       qvalue                 geneID Count
GO:0004252 0.0001216754 0.001842991 0.0007097528     MMP19/MMP9/TMPRSS9     3
GO:0048018 0.0001501969 0.001842991 0.0007097528 CCL19/CXCL9/FGF21/IGF1     4
GO:0008236 0.0001690761 0.001842991 0.0007097528     MMP19/MMP9/TMPRSS9     3
```

```
GO:0017171 0.0001798040 0.001842991 0.0007097528     MMP19/MMP9/TMPRSS9     3
GO:0008009 0.0003775579 0.003095975 0.0011922881            CCL19/CXCL9     2
GO:0042379 0.0008601090 0.005877411 0.0022634447            CCL19/CXCL9     2

# View(dfGO_sigs.CN)


cnetplot(mf_u2_h_nash,showCategory=10)
```




```
goplot(mf_u2_h_nash, geom = 'label')
```



From SS to NASH

```
mf_u2_ss_nash <- enrichGO(gene  = rownames(dds_u2_ss_nash), keyType =
'SYMBOL', OrgDb = org.Hs.eg.db, ont = "MF")

# check results
mf_u2_ss_nash.df <- data.frame(mf_u2_ss_nash@result) # transform the result
file to dataframe
head(mf_u2_ss_nash.df)

                    ID
GO:0004252 GO:0004252
GO:0008236 GO:0008236
GO:0017171 GO:0017171
GO:0004175 GO:0004175
GO:0001228 GO:0001228
GO:0001216 GO:0001216

Description
GO:0004252                                          serine-type endopeptidase
activity
GO:0008236                                              serine-type peptidase
activity
GO:0017171                                                   serine hydrolase
activity
GO:0004175                                                      endopeptidase
activity
```

```
GO:0001228 DNA-binding transcription activator activity, RNA polymerase II-
specific
GO:0001216                              DNA-binding transcription activator
activity
           GeneRatio   BgRatio       pvalue    p.adjust       qvalue
GO:0004252      3/10 170/18369 8.909705e-05 0.001976865 0.0009710918
GO:0008236      3/10 190/18369 1.239075e-04 0.001976865 0.0009710918
GO:0017171      3/10 194/18369 1.317910e-04 0.001976865 0.0009710918
GO:0004175      3/10 428/18369 1.334246e-03 0.012223222 0.0060043897
GO:0001228      3/10 468/18369 1.725375e-03 0.012223222 0.0060043897
GO:0001216      3/10 472/18369 1.768046e-03 0.012223222 0.0060043897
                       geneID Count
GO:0004252 MMP19/MMP9/TMPRSS9     3
GO:0008236 MMP19/MMP9/TMPRSS9     3
GO:0017171 MMP19/MMP9/TMPRSS9     3
GO:0004175 MMP19/MMP9/TMPRSS9     3
GO:0001228      EGR1/JUN/KLF6     3
GO:0001216      EGR1/JUN/KLF6     3

# View(dfGO_sigs.CN)


cnetplot(mf_u2_ss_nash,showCategory=10)
```