Introduction to Bulk RNAseq data analysis

Gene Set Testing for RNA-seq - Solutions

Contents

Exercise 1 - pathview

1. Use `pathview` to export a figure for "mmu04659", but this time only use genes that are statistically significant at FDR < 0.01

```
logFC <- shrink.d11 %>%
  drop_na(FDR, Entrez) %>%
  filter(FDR < 0.01) %>%
  dplyr::select(Entrez, logFC) %>%
  deframe()

pathview(gene.data = logFC,
         pathway.id = "mmu04659",
         species = "mmu",
         limit = list(gene=5, cpd=1))
```

```
## Loading required namespace: org.Mm.eg.db
```

```
##
```

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

```
## Info: Working in directory /home/cri.camres.org/sawle01/Documents/training/Bulk_RNAseq_Course_2021_A
```

```
## Info: Writing image file mmu04659.pathview.png
```

mmu04659.pathview.png:

Exercise 2 - GO term enrichment analysis

`clusterProfiler` can also perform over-representation analysis on GO terms. using the command `enrichGO`. Look at the help page for the command `enrichGO` (`?enrichGO`) and have a look at the instructions in the clusterProfiler book.

1. Run the over-representation analysis for GO terms
   - Use genes that have an adjusted p-value (FDR) of less than 0.01 and an absolute fold change greater than 2.
   - For this analysis you can use Ensembl IDs rather then Entrez

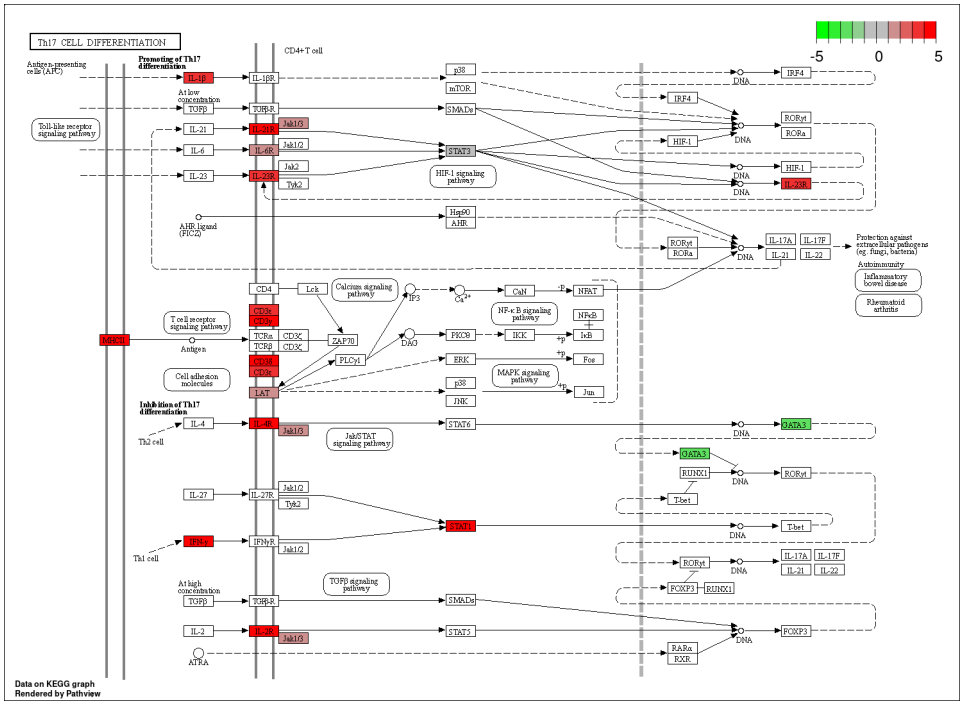


Figure 1: mmu04659 - Th17 cell differentiation

- You'll need to provide the background (`universe`) genes, this should be all the genes in our analysis.
- The mouse database package is called `org.Mm.eg.db`. You'll need to load it using `library` before running the analysis.

- As we are using Ensembl IDs, you'll need to set the `keyType` parameter in the `enrichGO` command to indicate this.
- Only test terms in the "Biological Processes" ontology

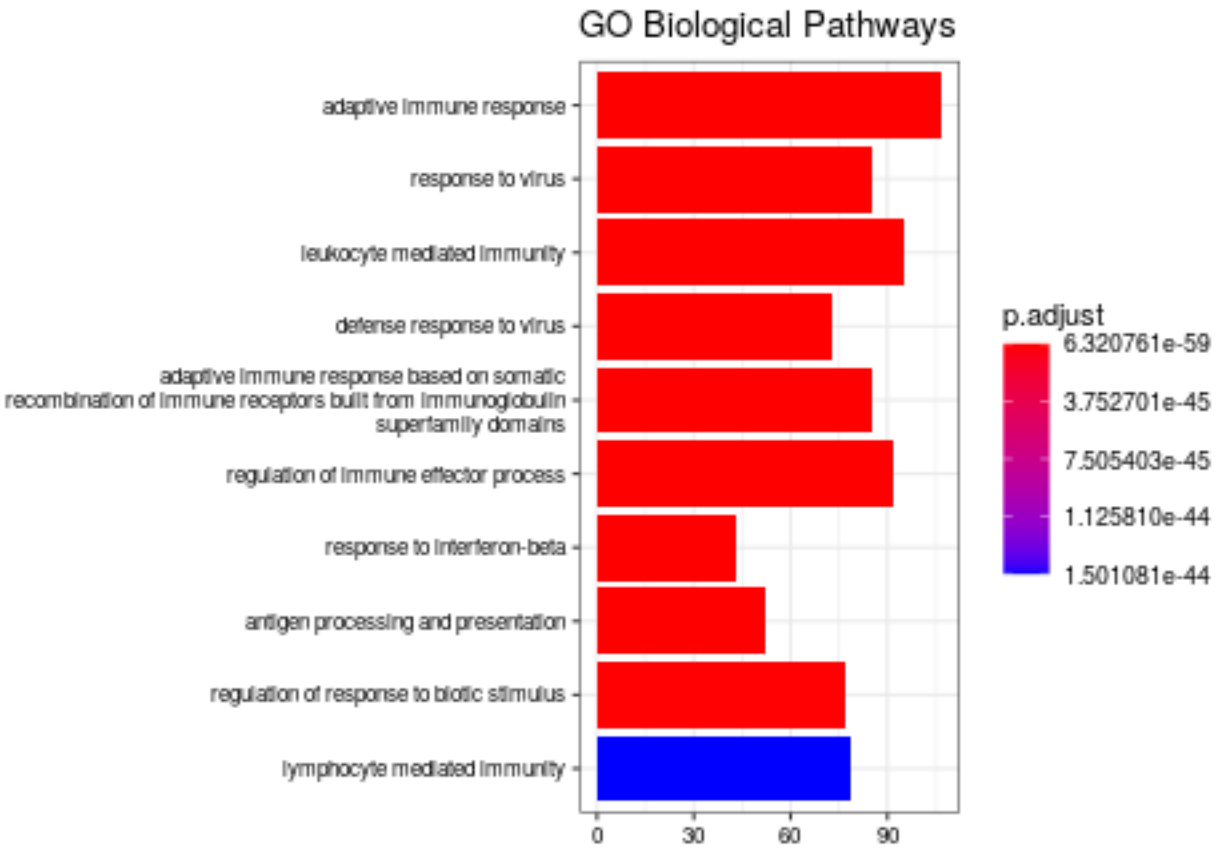
2. Use the `dotplot` function to visualise the results.

```
suppressMessages(library(org.Mm.eg.db))

sigGenes <-  shrink.d11 %>%
    drop_na(FDR) %>%
    filter(FDR < 0.01 & abs(logFC) > 1) %>%
    pull(GeneID)

universe <- shrink.d11$GeneID

ego <- enrichGO(gene          = sigGenes,
                universe      = universe,
                OrgDb         = org.Mm.eg.db,
                keyType       = "ENSEMBL",
                ont           = "BP",
                pvalueCutoff  = 0.01,
                readable      = TRUE)

dotplot(ego,
        font.size = 8,
```

```
        label_format=20
        )
```

```
## wrong orderBy parameter; set to default `orderBy = "x"`
```


```
barplot(ego,
        drop = TRUE,
        showCategory = 10,
        label_format = 20,
        title = "GO Biological Pathways",
        font.size = 8)
```



Exercise 3 - GSEA

Another common way to rank the genes is to order by pvalue, but also, sorting so that upregulated genes are at the start and downregulated at the end - you can do this combining the sign of the fold change and the pvalue.

1. Rank the genes by statisical significance - you will need to create a new ranking value using `-log10({p value}) * sign({Fold Change})`

2. Run `fgsea` using the new ranked genes and the H pathways

3. Conduct the same analysis for the d33 vs control contrast.

Exercise 3 - d11 new rank

```
# 1. Rank the genes by statistical significance - you will need to create
# a new ranking value using `-log10({p value}) * sign({Fold Change})`

# obtain the H(allmarks) catalog for mouse:
m_H_t2g <- msigdbr(species = "Mus musculus", category = "H") %>%
  dplyr::select(gs_name, entrez_gene, gene_symbol)

# rank genes
rankedGenes.e1 <- shrink.d11 %>%
```

```
  drop_na(Entrez, pvalue, logFC) %>%
  # rank genes by strength of significance,
  # keeping the direction of the fold change
  mutate(rank = -log10(pvalue) * sign(logFC)) %>%
  # sort genes by decreasing rank.
  arrange(-rank) %>%
  # keep ranks and Entrez IDs
  pull(rank,Entrez)

# conduct analysis:
gseaRes.e1 <- GSEA(rankedGenes.e1,
                TERM2GENE = m_H_t2g[,c("gs_name", "entrez_gene")],
                #pvalueCutoff = 0.05,
                pvalueCutoff = 1.00, # to retrieve whole output
                minGSSize = 15,
                maxGSSize = 500)
```

```
## preparing geneSet collections...
```

```
## GSEA analysis...
```

```
## Warning in fgseaMultilevel(...): For some pathways, in reality P-values are less
## than 1e-10. You can set the `eps` argument to zero for better estimation.
```

```
## leading edge analysis...
```

```
## done...
```

```
# have function to format in scientific notation
format.e1 <- function(x) (sprintf("%.1e", x))
# format table:
gseaRes.e1 %>%
  # sort in decreasing order of absolute NES
  arrange(desc(abs(NES))) %>%
  # only keep the 10 entries with the lowest p.adjust
  top_n(10, -p.adjust) %>%
  # remove columns 'core_enrichment' and 'Description'
  dplyr::select(-core_enrichment) %>%
  dplyr::select(-Description) %>%
  # convert to data.frame
  data.frame() %>%
  # remove row names
  remove_rownames() %>%
  # format score
  mutate(NES=formatC(NES, digits = 3)) %>%
  mutate(ES=formatC(enrichmentScore, digits = 3)) %>%
  relocate(ES, .before=NES) %>%
  dplyr::select(-enrichmentScore) %>%
  # format p-values
  modify_at(
    c("pvalue", "p.adjust", "qvalues"),
    format.e1
  ) %>%
  # display
  DT::datatable(options = list(dom = 't'))
```

```
## PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is installed, pleas
```

Exercise 3 - d33

With d33 and H catalog:

```
# read d33 data in:
shrink.d33 <- readRDS("RObjects/Shrunk_Results.d33.rds")

# get mouse H(allmarks) catalog
m_H_t2g <- msigdbr(species = "Mus musculus", category = "H") %>%
  dplyr::select(gs_name, entrez_gene, gene_symbol)

# rank genes
rankedGenes.e3 <- shrink.d33 %>%
  drop_na(Entrez, pvalue, logFC) %>%
  mutate(rank = -log10(pvalue) * sign(logFC)) %>%
  arrange(-rank) %>%
  pull(rank,Entrez)

# perform analysis
gseaRes.e3 <- GSEA(rankedGenes.e3,
                TERM2GENE = m_H_t2g[,c("gs_name", "entrez_gene")],
                #pvalueCutoff = 0.05,
                pvalueCutoff = 1.00, # to retrieve whole output
                minGSSize = 15,
                maxGSSize = 500)
```

```
## preparing geneSet collections...
```

```
## GSEA analysis...
```

```
## Warning in fgseaMultilevel(...): There were 2 pathways for which P-values were
## not calculated properly due to unbalanced (positive and negative) gene-level
## statistic values. For such pathways pval, padj, NES, log2err are set to NA. You
## can try to increase the value of the argument nPermSimple (for example set it
## nPermSimple = 10000)
```

```
## Warning in fgseaMultilevel(...): For some pathways, in reality P-values are less
## than 1e-10. You can set the `eps` argument to zero for better estimation.
```

```
## leading edge analysis...
```

```
## done...
```

Check outcome:

```
gseaRes.e3 %>%
  arrange(desc(abs(NES))) %>%
  top_n(10, -p.adjust) %>%
  dplyr::select(-core_enrichment) %>%
  dplyr::select(-Description) %>%
  data.frame() %>%
  remove_rownames() %>%
  # format score
  mutate(NES=formatC(NES, digits = 3)) %>%
  mutate(ES=formatC(enrichmentScore, digits = 3)) %>%
  relocate(ES, .before=NES) %>%
  dplyr::select(-enrichmentScore) %>%
  # format p-values
```

```
  modify_at(
    c("pvalue", "p.adjust", "qvalues"),
    format.e1
  ) %>%
  DT::datatable(options = list(dom = 't'))
```

Extended challenge 3 - compare outcomes for two ranking schemes

Compare to putcomes obtained with the two ranking schemes:

- by logFC only
- by significance strength and direction of change

```
# d11 + logFC-only ranking scheme
rankedGenes <- shrink.d11 %>%
  drop_na(Entrez, pvalue, logFC) %>%
  mutate(rank = logFC) %>%
  arrange(-rank) %>%
  pull(rank, Entrez)
gseaRes <- GSEA(rankedGenes,
                TERM2GENE = m_H_t2g[,1:2],
                #pvalueCutoff = 0.05,
                pvalueCutoff = 1.00, # to retrieve whole output
                minGSSize = 15,
                maxGSSize = 500)
```

```
## preparing geneSet collections...
```

```
## GSEA analysis...
```

```
## Warning in fgseaMultilevel(...): For some pathways, in reality P-values are less
## than 1e-10. You can set the `eps` argument to zero for better estimation.
```

```
## leading edge analysis...
```

```
## done...
```

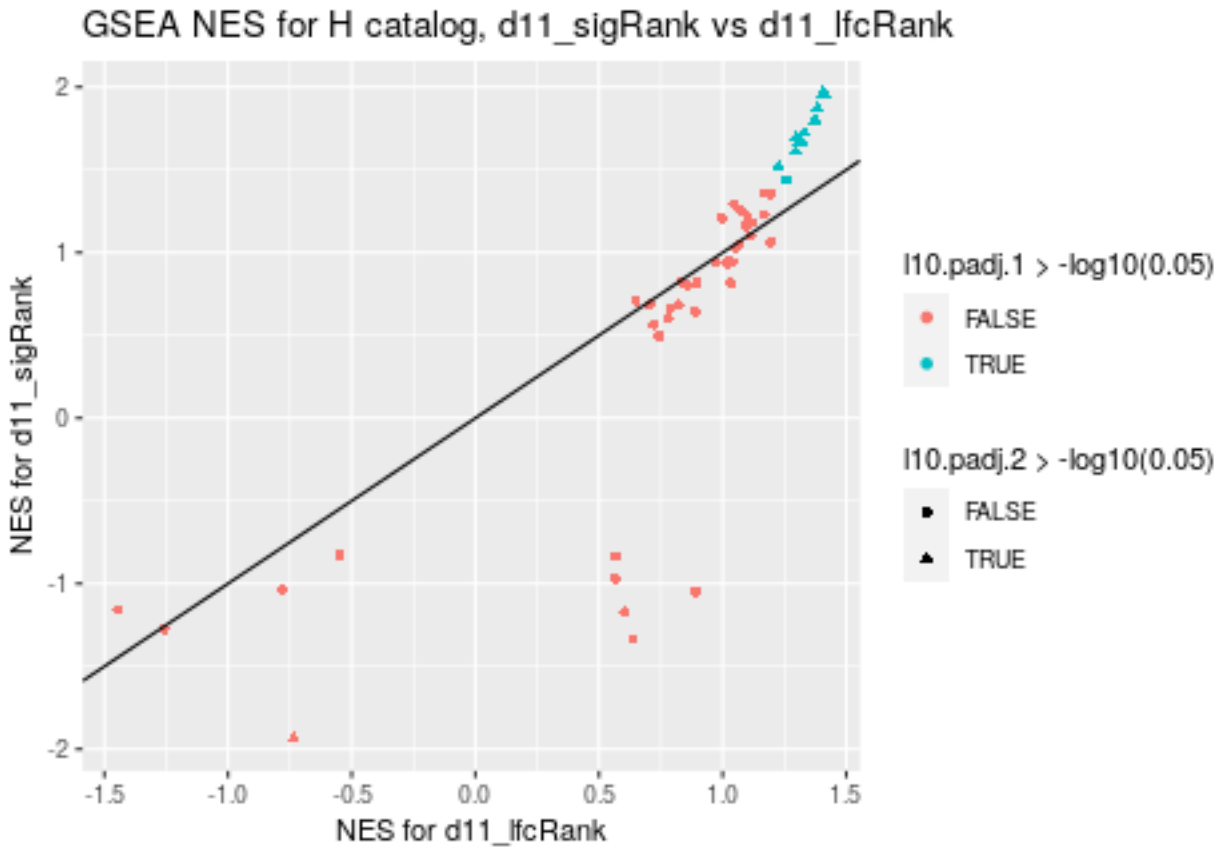
Combine the two sets of results:

```
# store combined data in new data.frame res.df
# only keep "ID", "NES" and "p.adjust"
res.df <- gseaRes %>%
  data.frame() %>%
  # rename NES and p.adjust
  dplyr::rename(NES.1=NES, padj.1=p.adjust) %>%
  # keep "ID", "NES" and "p.adjust"
  dplyr::select(ID, NES.1, padj.1) %>%
  # merge with the d11 + significance strength
  left_join(gseaRes.e1[,c("ID", "NES", "p.adjust")]) %>%
  # rename NES and p.adjust
  dplyr::rename(NES.2=NES, padj.2=p.adjust) %>%
  # compute -log10(p.adjust)
  mutate(l10.padj.1 = -log10(padj.1),
         l10.padj.2 = -log10(padj.2))
```

```
## Joining, by = "ID"
```

Plot NES:

```
p <- res.df %>%
  # skip terms where NES is NA is any data set
  dplyr::filter(!is.na(NES.1) & !is.na(NES.2)) %>%
  # plot NES of 2nd set vs NES of 1st set
  ggplot(aes(x=NES.1,
             y=NES.2,
             # color by sig in 1st set
             col=l10.padj.1>-log10(0.05),
             # shape by sig in 2nd set
             shape=l10.padj.2>-log10(0.05))
         ) +
  # show points
  geom_point() +
  # add 'identity' line
  geom_abline(intercept = 0, slope = 1) +
  # add axes labels and title
  xlab("NES for d11_lfcRank") +
  ylab("NES for d11_sigRank") +
  ggtitle("GSEA NES for H catalog, d11_sigRank vs d11_lfcRank")
p
```



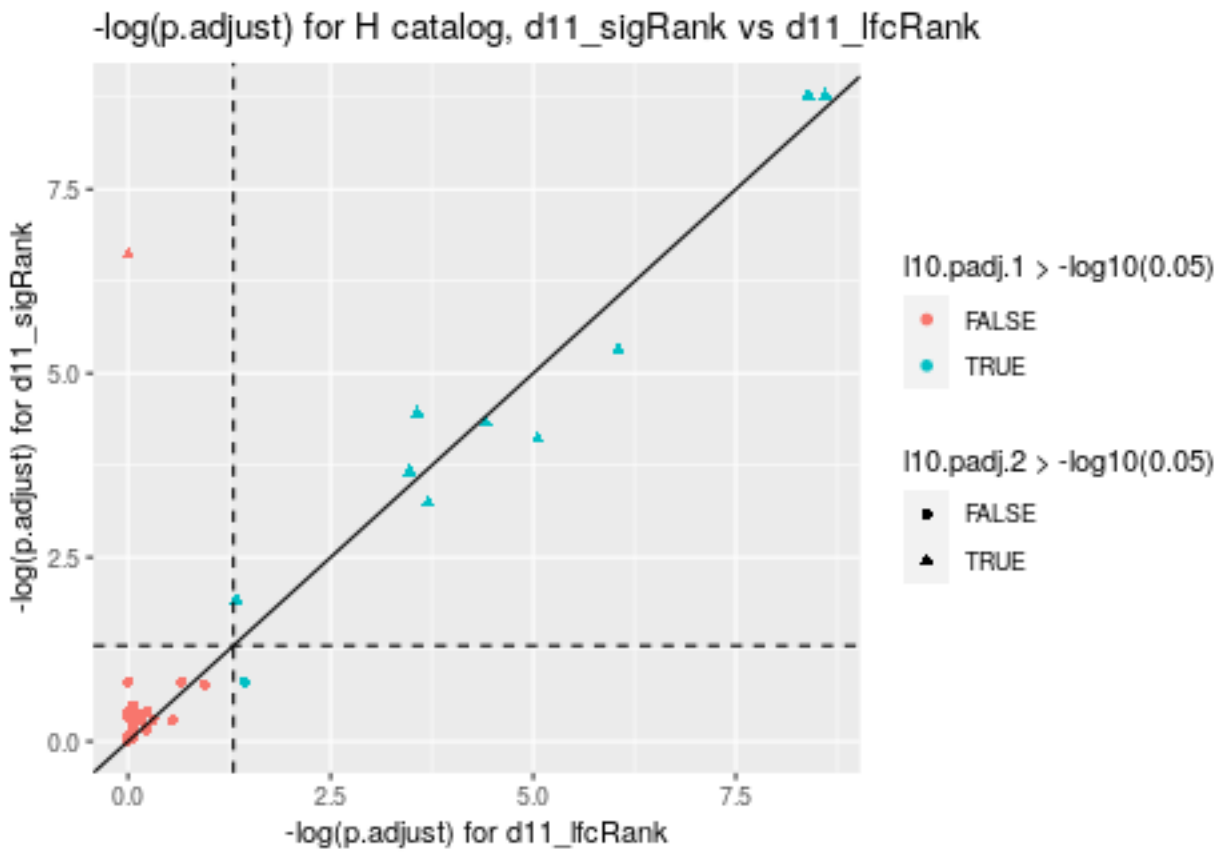
List terms with NES.1 $>$ 0 and NES.2 $<$ 0:

```
res.df %>%
  filter(NES.1 > 0 & NES.2 < 0) %>%
  # format score
```

```
  mutate(NES.1=formatC(NES.1, digits = 3)) %>%
  mutate(NES.2=formatC(NES.2, digits = 3)) %>%
  mutate(l10.padj.1=formatC(l10.padj.1, digits = 3)) %>%
  mutate(l10.padj.2=formatC(l10.padj.2, digits = 3)) %>%
  # format p-values
  # format p-values
  modify_at(
    c("padj.1", "padj.2"),
    format.e1
  ) %>%
  DT::datatable(options = list(dom = 't'))
```

Plot -log10(p.adjust):

```
p <- res.df %>%
  # skip terms where NES is NA is any data set
  dplyr::filter(!is.na(NES.1) & !is.na(NES.2)) %>%
  # plot significance strength of 2nd set vs that of 1st set
  ggplot(aes(x=l10.padj.1,
             y=l10.padj.2,
             # color by sig in 1st set
             col=l10.padj.1>-log10(0.05),
             # shape by sig in 2nd set
             shape=l10.padj.2>-log10(0.05))
         ) +
  # show points
  geom_point() +
  # add 'identity' line
  geom_abline(intercept = 0, slope = 1) +
  # add 5% significance line for set 1
  geom_vline(xintercept = -log10(0.05), linetype = 2) +
  # add 5% significance line for set 2
  geom_hline(yintercept = -log10(0.05), linetype = 2) +
  # add axes labels and title
  xlab("-log(p.adjust) for d11_lfcRank") +
  ylab("-log(p.adjust) for d11_sigRank") +
  ggtitle("-log(p.adjust) for H catalog, d11_sigRank vs d11_lfcRank")
p
```



List terms with whose significance differs between ranking schemes:

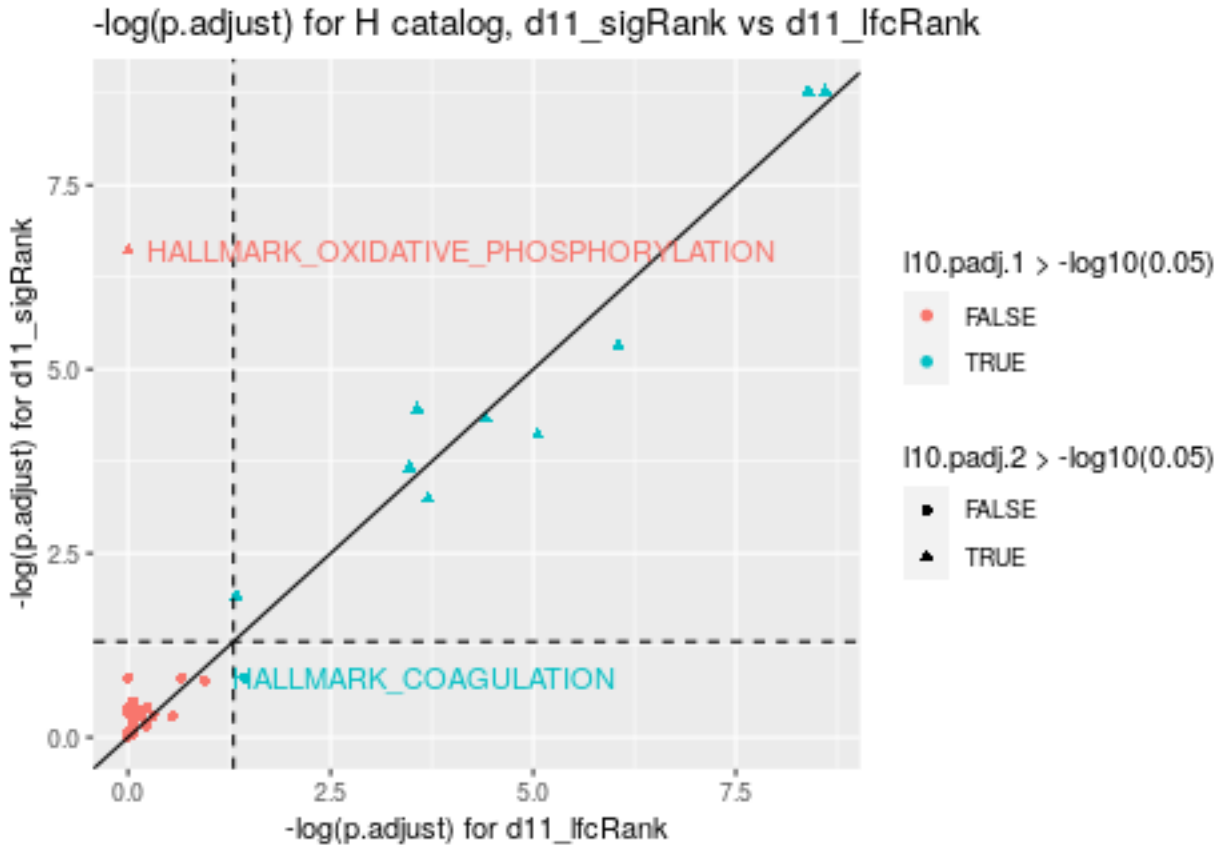
```
diffSig <- res.df %>%
  mutate(sig005.1 = padj.1 < 0.05) %>%
  mutate(sig005.2 = padj.2 < 0.05) %>%
  mutate(sigIsDiff = (sig005.1 | sig005.2) & sig005.1 != sig005.2) %>%
  filter(sigIsDiff)
diffSig %>%
  # format score
  mutate(NES.1=formatC(NES.1, digits = 3)) %>%
  mutate(NES.2=formatC(NES.2, digits = 3)) %>%
  mutate(l10.padj.1=formatC(l10.padj.1, digits = 3)) %>%
  mutate(l10.padj.2=formatC(l10.padj.2, digits = 3)) %>%
  # format p-values
  # format p-values
  modify_at(
    c("padj.1", "padj.2"),
    format.e1
  ) %>%
  DT::datatable(options = list(dom = 't'))
```

```
require(ggrepel)
```

```
## Loading required package: ggrepel
```

```
p + geom_text_repel(data=diffSig,
                    aes(x=l10.padj.1,
```

```
                    y=l10.padj.2,
                    label=ID),
                box.padding = 0.8,
                show.legend = FALSE)
```



Extended challenge 3 - compare outcomes for d11 and d33

Compare results obtained for d11 and d33, with genes ranked by significance and fold change direction:

First get run analysis for d11 with genes ranked by significance and logFC sign:

```
# run analysis for d11 with genes ranked by signinficance and LFC sign
# as for d33
rankedGenes <- shrink.d11 %>%
  drop_na(Entrez, pvalue, logFC) %>%
  mutate(rank = -log10(pvalue) * sign(logFC)) %>%
  arrange(-rank) %>%
  pull(rank,Entrez)
gseaRes <- GSEA(rankedGenes,
                TERM2GENE = m_H_t2g[,1:2],
                pvalueCutoff = 1.00, # to retrieve whole output
                minGSSize = 15,
                maxGSSize = 500)
```

```
## preparing geneSet collections...
```

```
## GSEA analysis...
```

```
## Warning in fgseaMultilevel(...): For some of the pathways the P-values were
## likely overestimated. For such pathways log2err is set to NA.
```

```
## Warning in fgseaMultilevel(...): For some pathways, in reality P-values are less
## than 1e-10. You can set the `eps` argument to zero for better estimation.
```

```
## leading edge analysis...
```

```
## done...
```

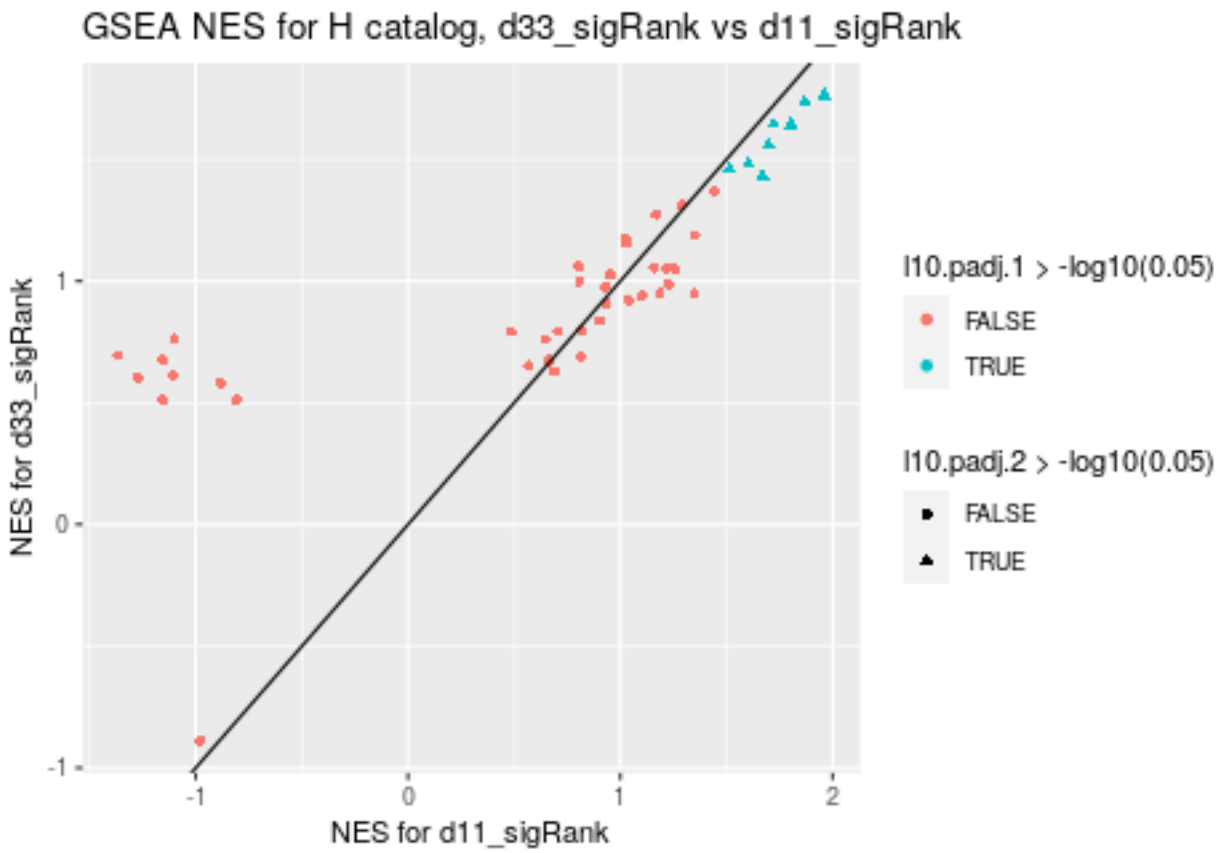
Combine outcomes:

```
res.df <- gseaRes %>%
  data.frame() %>%
  # rename NES and p.adjust
  dplyr::rename(NES.1=NES, padj.1=p.adjust) %>%
  # keep "ID", "NES" and "p.adjust"
  dplyr::select(ID, NES.1, padj.1) %>%
  # merge with the d11 + significance strength
  left_join(gseaRes.e3[,c("ID", "NES", "p.adjust")]) %>%
  # rename NES and p.adjust
  dplyr::rename(NES.2=NES, padj.2=p.adjust) %>%
  # compute -log10(p.adjust)
  mutate(l10.padj.1 = -log10(padj.1),
         l10.padj.2 = -log10(padj.2))
```

```
## Joining, by = "ID"
```

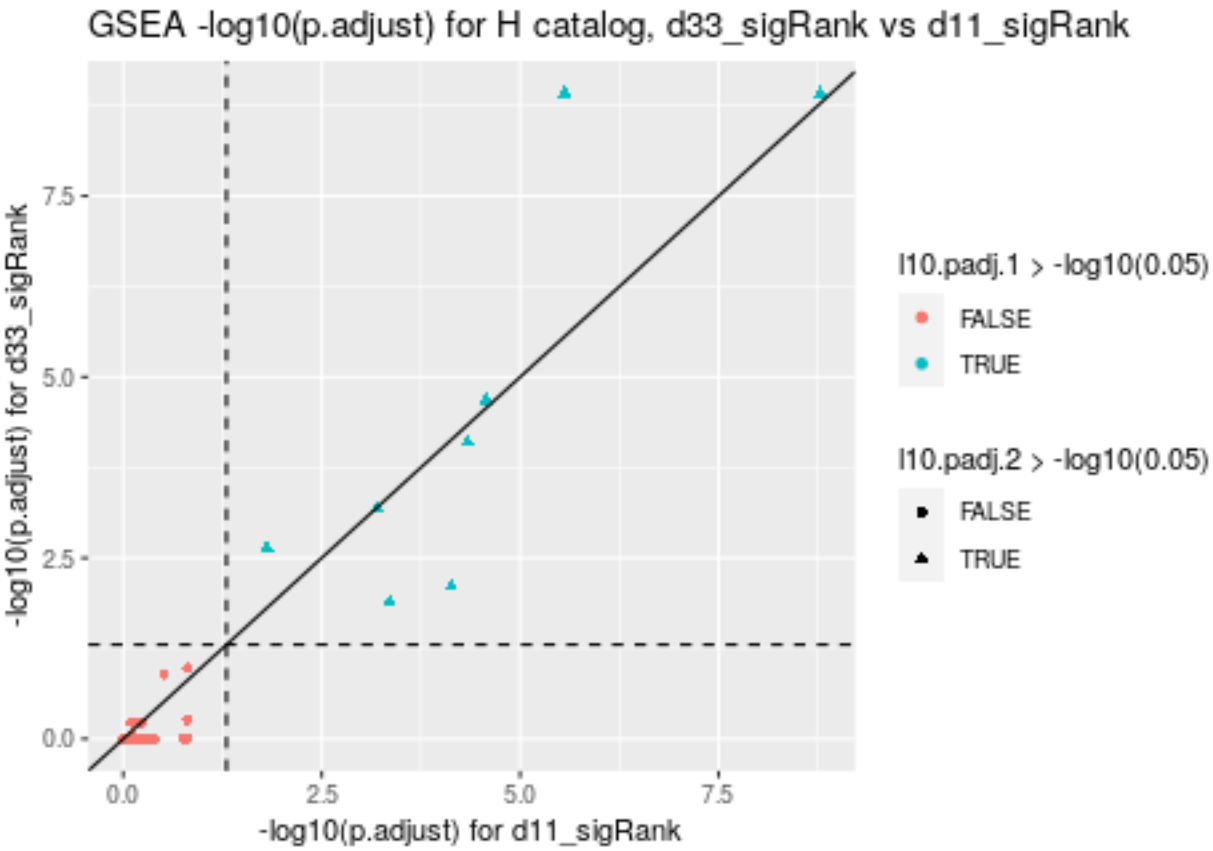
Plot NES:

```
res.df %>%
  dplyr::filter(!is.na(NES.1) & !is.na(NES.2)) %>%
  ggplot(aes(x=NES.1,
             y=NES.2,
             col=l10.padj.1>-log10(0.05),
             shape=l10.padj.2>-log10(0.05))
         ) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1) +
  xlab("NES for d11_sigRank") +
  ylab("NES for d33_sigRank") +
  ggtitle("GSEA NES for H catalog, d33_sigRank vs d11_sigRank")
```



Plot -log10(p.adjust):

```
res.df %>%
  dplyr::filter(!is.na(NES.1) & !is.na(NES.2)) %>%
  ggplot(aes(x=l10.padj.1,
             y=l10.padj.2,
             col=l10.padj.1>-log10(0.05),
             shape=l10.padj.2>-log10(0.05))
         ) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1) +
  geom_hline(yintercept = -log10(0.05), linetype = 2) +
  geom_vline(xintercept = -log10(0.05), linetype = 2) +
  xlab("-log10(p.adjust) for d11_sigRank") +
  ylab("-log10(p.adjust) for d33_sigRank") +
  ggtitle("GSEA -log10(p.adjust) for H catalog, d33_sigRank vs d11_sigRank")
```


GSEA -log10(p.adjust) for H catalog, d33_sigRank vs d11_sigRank
-log10(p.adjust) for d33_sigRank
-log10(p.adjust) for d11_sigRank
0.0
2.5
5.0
7.5
l10.padj.1 > -log10(0.05)
FALSE
TRUE
l10.padj.2 > -log10(0.05)
FALSE
TRUE