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))
```

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

## Info: Working in directory /Users/baller01/MyProjectsSvn/SvnRepoForTraining/BioinfoCore/FernandesM/20

## 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 commmand `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
    - 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.

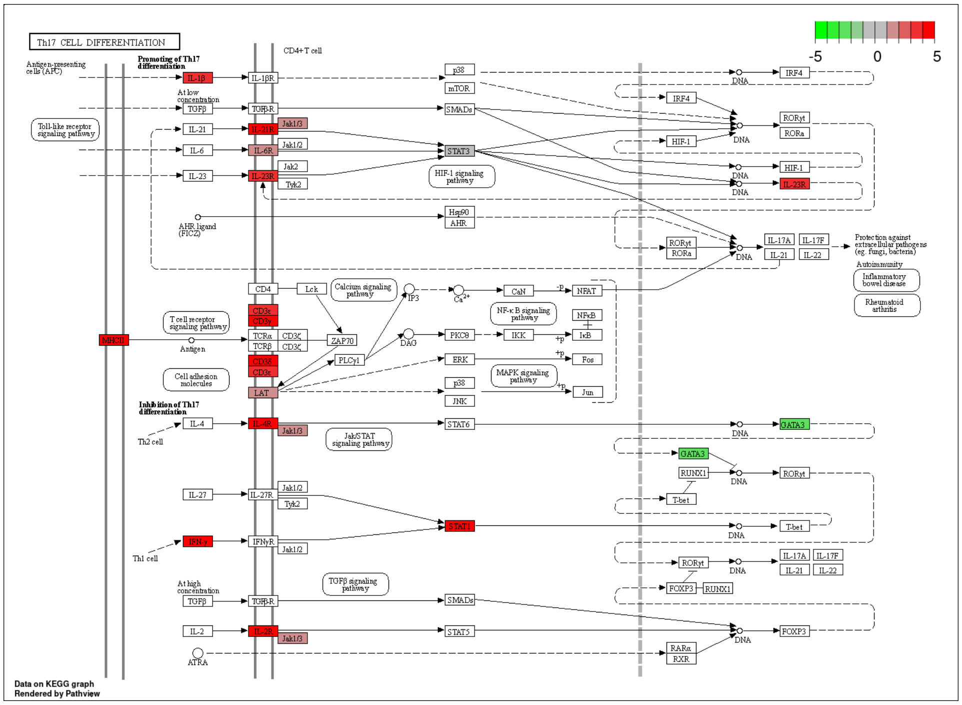
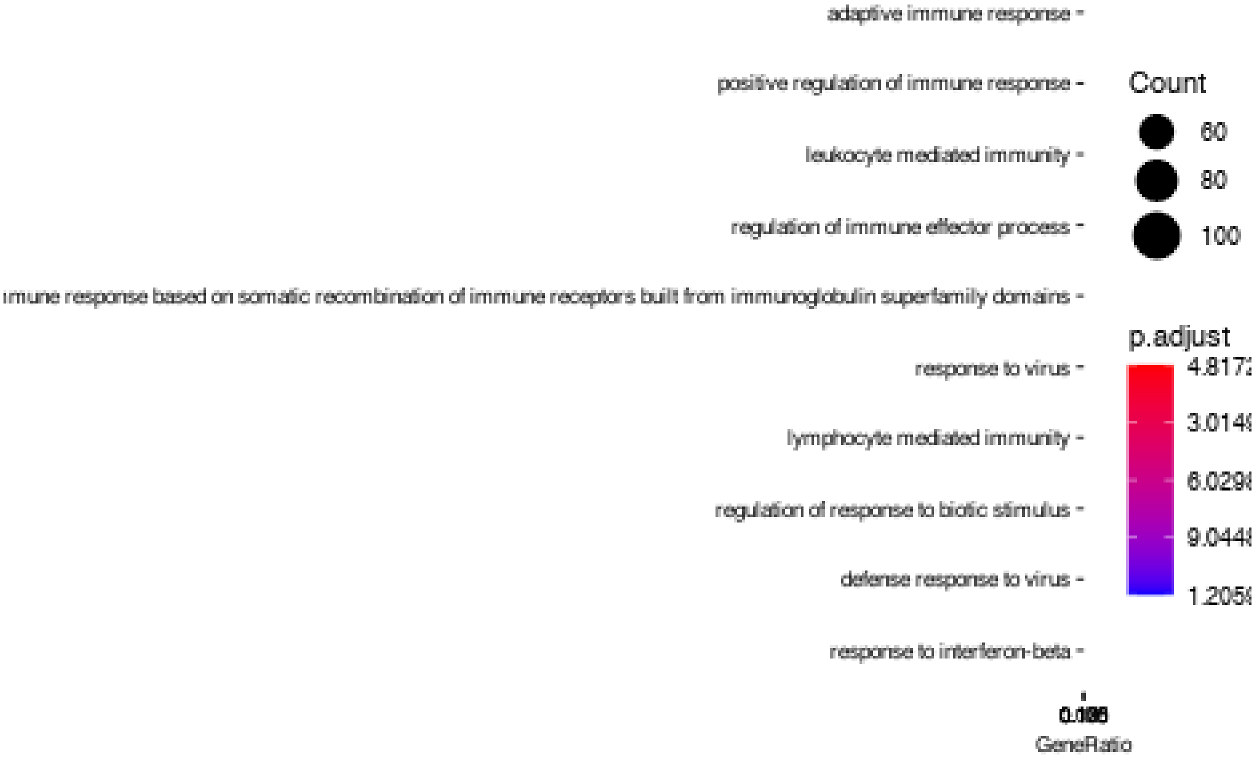


Figure 1: mmu04659 - Th17 cell differentiation

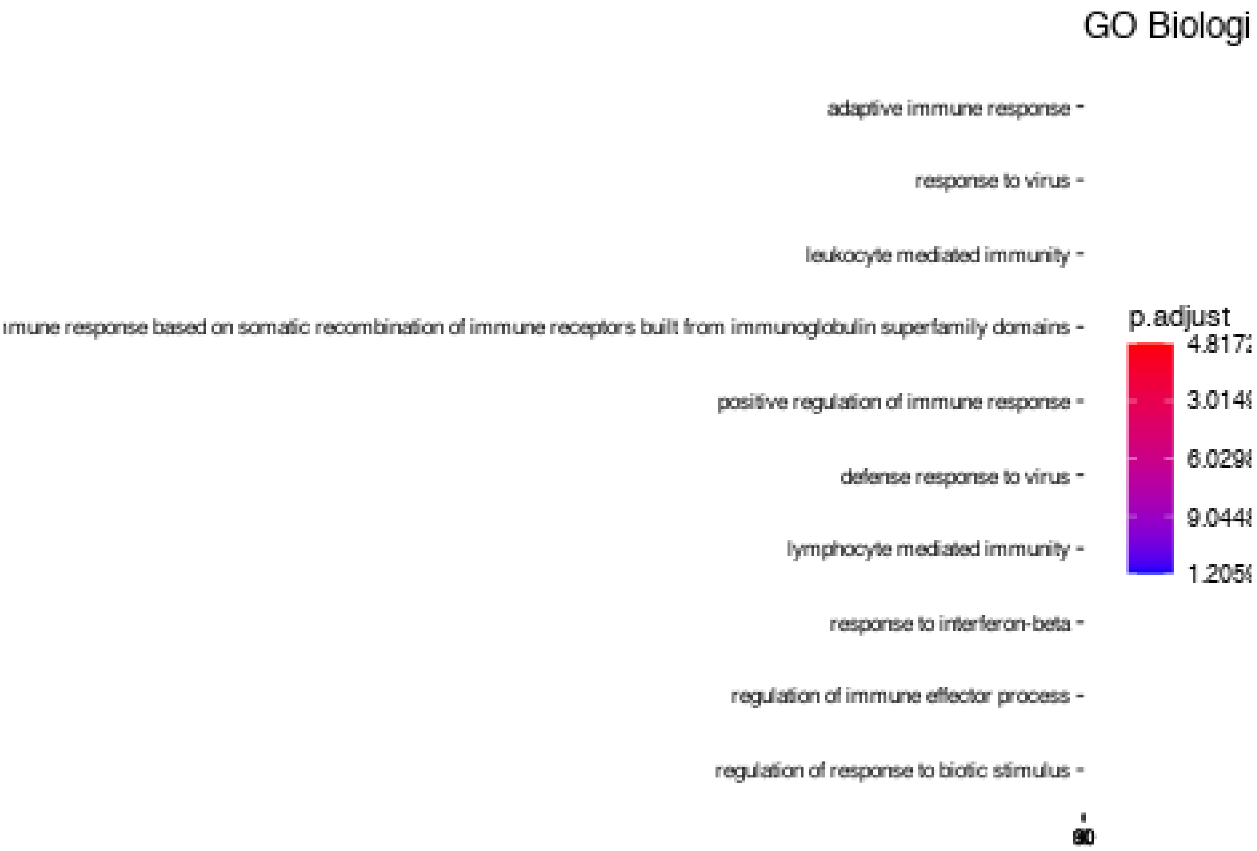
- As we are using Ensembl IDs, you'll need to set the `keyType` parameter in the `enrichGO` command to indicate this.
- Only test terms in the "Biological Processes" ontology

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

```
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,
        )
```



```r
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 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...
```

```
# 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'))
```

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(...): 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'))
```