Introduction to Bulk RNAseq data analysis

Gene Set Testing for RNA-seq - Solutions

Contents

Exercise 1 - pathview

Load the required packages and data for Day 11 if you have not already done so.

```
library(msigdbr)
library(clusterProfiler)
library(pathview)
library(tidyverse)

shrink.d11 <- readRDS("RObjects/Shrunk_Results.d11.rds")
```

1. Use `pathview` to export a figure for "mmu04659"or "mmu04658", 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) %>%
  pull(logFC, Entrez)

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/sawle01/Documents/training/Bulk_RNAseq_Course_Base/Markdowns
```

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

mmu04659.pathview.png:

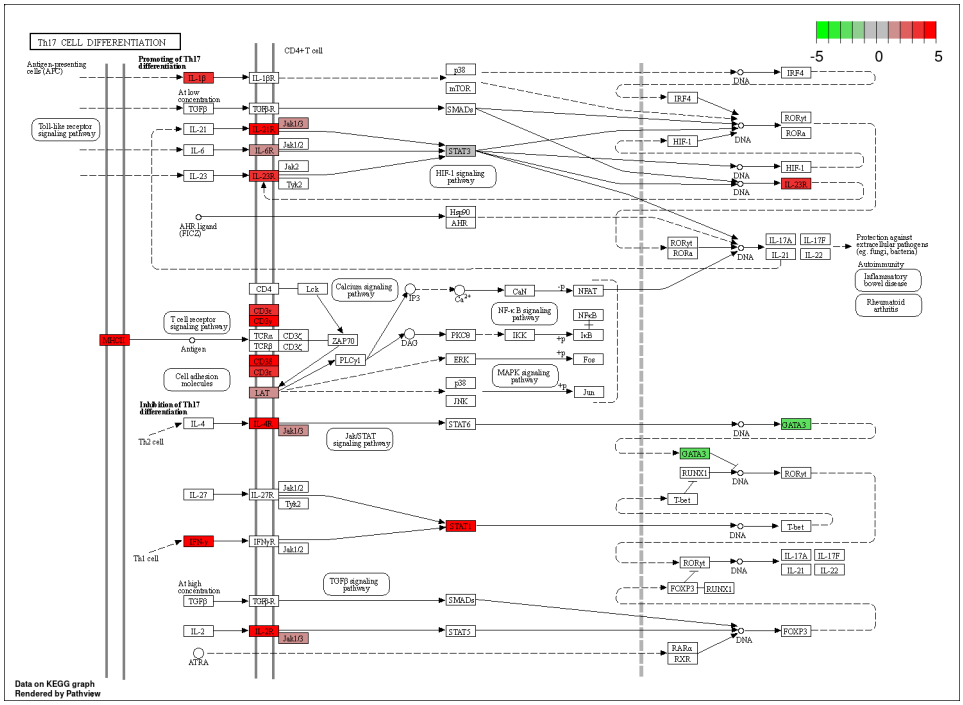


Figure 1: mmu04659 - Th17 cell differentiation

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
   - 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.

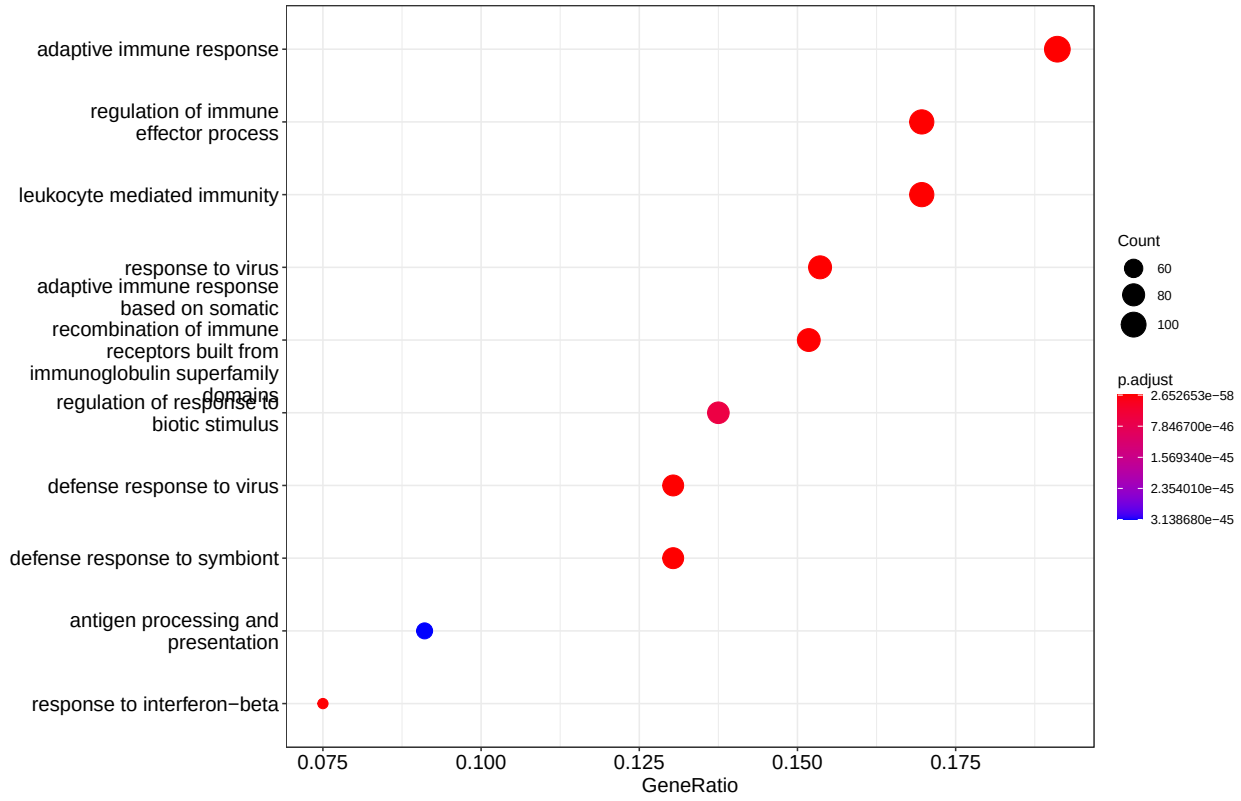
   - As we are using Ensembl IDs, you'll need to set the `keyType` parameter in the `enrichGO` command to indicate this.
   - Only test terms in the "Biological Processes" ontology

```
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)
```

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

```
dotplot(ego, font.size = 14)
```



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.

First load the pathway details if you have not already done so.

```
library(msigdbr)
term2gene <- msigdbr(species = "Mus musculus", category = "H") %>%
  select(gs_name, entrez_gene)
term2name <- msigdbr(species = "Mus musculus", category = "H") %>%
```

```
  select(gs_name, gs_description) %>%
  distinct()
```

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

```
# rank genes
rankedGenes.e11 <- shrink.d11 %>%
  drop_na(Entrez, pvalue, logFC) %>%
  mutate(rank = -log10(pvalue) * sign(logFC)) %>%
  arrange(desc(rank)) %>%
  pull(rank, Entrez)
```

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

```
# conduct analysis:
gseaRes.e11 <- GSEA(rankedGenes.e11,
                TERM2GENE = term2gene,
                TERM2NAME = term2name,
                pvalueCutoff = 1.00,
                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...
```

View the results:

```
as_tibble(gseaRes.e11) %>%
  arrange(desc(abs(NES))) %>%
  top_n(10, wt=-p.adjust) %>%
  select(-core_enrichment) %>%
  mutate(across(c("enrichmentScore", "NES"), round, digits=3)) %>%
  mutate(across(c("pvalue", "p.adjust", "qvalues"), scales::scientific))
```

3. Conduct the same analysis for the day 33 Infected vs Uninfected contrast.

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

# rank genes
```

```
rankedGenes.e33 <- shrink.d33 %>%
  drop_na(Entrez, pvalue, logFC) %>%
  mutate(rank = -log10(pvalue) * sign(logFC)) %>%
  arrange(desc(rank)) %>%
  pull(rank,Entrez)

# perform analysis
gseaRes.e33 <- GSEA(rankedGenes.e33,
                TERM2GENE = term2gene,
                TERM2NAME = term2name,
                pvalueCutoff = 1.00,
                minGSSize = 15,
                maxGSSize = 500)
```

```
## preparing geneSet collections...

## GSEA analysis...

## Warning in fgseaMultilevel(...): There were 3 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...
```

View the results:

```
as_tibble(gseaRes.e33) %>%
  arrange(desc(abs(NES))) %>%
  top_n(10, wt=-p.adjust) %>%
  select(-core_enrichment) %>%
  mutate(across(c("enrichmentScore", "NES"), round, digits=3)) %>%
  mutate(across(c("pvalue", "p.adjust", "qvalues"), scales::scientific))
```