RNA-seq analysis in R

Gene Set Testing for RNA-seq

Last modified: 24 Mar 2021

Contents

The list of differentially expressed genes is sometimes so long that its interpretation becomes cumbersome and time consuming. It may also be very short while some genes have low p-value yet higher than the given threshold.

A common downstream procedure to combine information across genes is gene set testing. It aims at finding pathways or gene networks the differentially expressed genes play a role in.

Various ways exist to test for enrichment of biological pathways. We will look into over representation and gene set enrichment analyses.

A gene set comprises genes that share a biological function, chromosomal location, or any other relevant criterion.

Over-representation

Method

This method tests whether genes in a pathway are present in a subset of our data in a higher number than expected by chance (explanations derived from the clusterProfiler manual).

Genes in the experiment are split in two ways:

- annotated to the pathway or not
- differentially expressed or not

We can then create a contingency table with:

- rows: genes in pathway or not
- columns: genes differentially expressed or not

A toy example

Let's start with an experiment with:

- 20,000 genes of which 100 are differentially expressed,
- a pathway with 2000 genes,
  - including 20 of the 100 differentially expressed genes.

```
contingencyTable <- data.frame(
  diffExpNo=c(1980, 17920),
  diffExpYes=c(20, 80))
row.names(contingencyTable) <- c("pathwayYes", "pathwayNo")

rowSums(contingencyTable)
```

```
## pathwayYes  pathwayNo
##       2000      18000
```

```
colSums(contingencyTable)
```

```
##  diffExpNo diffExpYes
##      19900        100
```

```
contingencyTable
```

```
##             diffExpNo diffExpYes
## pathwayYes       1980         20
## pathwayNo       17920         80
```

Let's define variables for the analysis of a given pathway:

- **N**: total number of genes in the **background** set, e.g. all genes tested - 20,000 in our example.
  - This background set is sometimes referred to as the **universe**.
- **M**: number of genes within that background that are annotated to the pathway - 2,000 in our example
- **n**: number of differentially expressed genes - 100 in our example
- **k**: number of differentially expressed genes that are annotated to the pathway
- 20 in our example

Significance can then be assessed with the hypergeometric distribution:

$$p = 1 - \sum_{i=0}^{k-1} \frac{\binom{M}{i}\binom{N-M}{n-i}}{\binom{N}{n}}$$

The test above is identical to the one-tailed Fisher's exact test.

```
fisher.test(contingencyTable, alternative = "greater")
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  contingencyTable
## p-value = 0.9992
## alternative hypothesis: true odds ratio is greater than 1
## 95 percent confidence interval:
##  0.2874878       Inf
## sample estimates:
## odds ratio
##  0.4419959
```

To save time and effort there are a number of packages that make applying this test to a large number of gene sets simpler, and which will import gene lists for testing from various sources.

Today we will use `clusterProfiler`.

clusterProfiler

`clusterprofiler` (Yu et al. 2012) supports direct online access of the current KEGG database (KEGG: Kyoto Encyclopedia of Genes and Genomes), rather than relying on R annotation packages. It also provides some nice visualisation options.

We first search the resource for mouse data:

```
library(clusterProfiler)
library(tidyverse)

search_kegg_organism('mouse', by='common_name')
```

```
##    kegg_code scientific_name common_name
## 14       mmu    Mus musculus       mouse
```

We will use the 'mmu' 'kegg_code.'

KEGG enrichment analysis

The input for the KEGG enrichment analysis is the list of gene IDs of significant genes.

We now load the R object keeping the outcome of the differential expression analysis for the d11 contrast.

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

We will only use genes that have:

- an adjusted p-value (FDR) of less than 0.05
- and an absolute fold change greater than 2.

We need to remember to eliminate genes with missing values in the FDR as a result of the independent filtering by DESeq2.

For this tool we need to use Entrez IDs, so we will also need to eliminate genes with a missing Entrez ID (NA values in the 'Entrez' column).

```
## Reading KEGG annotation online:
##
## Reading KEGG annotation online:
```

```
## # A tibble: 10 x 9
##    ID     Description GeneRatio BgRatio   pvalue p.adjust   qvalue geneID  Count
##    <chr>  <chr>       <chr>     <chr>      <dbl>    <dbl>    <dbl> <chr>   <int>
##  1 mmu04~ Antigen pr~ 40/337    90/8914 1.41e-33 3.15e-31 2.24e-31 14991/~    40
##  2 mmu05~ Epstein-Ba~ 56/337    231/89~ 1.73e-30 1.94e-28 1.37e-28 12502/~    56
##  3 mmu05~ Graft-vers~ 32/337    63/8914 2.25e-29 1.68e-27 1.19e-27 14939/~    32
##  4 mmu04~ Type I dia~ 33/337    70/8914 6.90e-29 3.87e-27 2.74e-27 16160/~    33
##  5 mmu04~ Phagosome   48/337    182/89~ 5.98e-28 2.35e-26 1.67e-26 16414/~    48
##  6 mmu05~ Allograft ~ 31/337    63/8914 6.29e-28 2.35e-26 1.67e-26 16160/~    31
##  7 mmu05~ Influenza A 45/337    173/89~ 5.95e-26 1.91e-24 1.35e-24 217069~    45
##  8 mmu04~ Cell adhes~ 45/337    174/89~ 7.77e-26 2.18e-24 1.54e-24 16414/~    45
##  9 mmu05~ Viral myoc~ 32/337    88/8914 1.04e-23 2.59e-22 1.84e-22 16414/~    32
## 10 mmu05~ Leishmania~ 28/337    70/8914 3.25e-22 7.28e-21 5.17e-21 16414/~    28
```

Visualise a pathway in a browser

`clusterProfiler` has a function `browseKegg` to view the KEGG pathway in a browser, highlighting the genes we selected as differentially expressed.

We will show one of the top hits: pathway 'mmu04612' for 'Antigen processing and presentation.'

```
browseKEGG(kk, 'mmu04612')
```

Visualise a pathway as a file

The package `pathview` (Luo et al. 2013) can be used to generate figures of KEGG pathways.

One advantage over the `clusterProfiler` browser method `browseKEGG` is that genes can be coloured according to fold change levels in our data. To do this we need to pass `pathview` a named vector of fold change values (one could in fact colour by any numeric vector, e.g. p-value).

The package plots the KEGG pathway to a `png` file in the working directory.

```
# check working directory
#getwd()

# run pathview
library(pathview)
logFC <- shrink.d11$logFC
names(logFC) <- shrink.d11$Entrez
pathview(gene.data = logFC,
         pathway.id = "mmu04612",
         species = "mmu",
         limit = list(gene=20, cpd=1))
```

mmu04612.pathview.png:

Exercise 1

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

Exercise 2 - GO term enrichment analysis

`clusterProfiler` can also perform over-representation analysis on GO terms using the command `enrichGO`. Check: * the help page for the command `enrichGO` (type `?enrichGO` at the console prompt) * and the instructions in the clusterProfiler book.

1. Run the over-representation analysis for GO terms

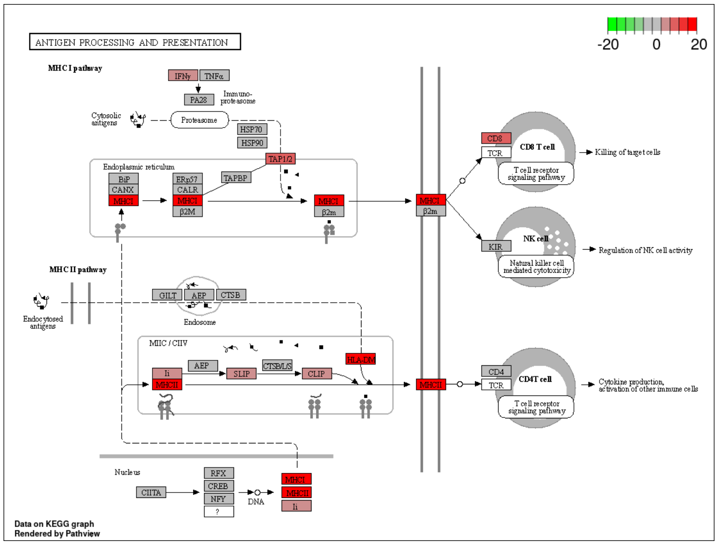


Figure 1: mmu04612 - Antigen processing and presentation

- 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

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

```
# may need devtools::install_github("YuLab-SMU/enrichplot")
# to avoid a 'wrong orderBy parameter' warning.
```

GSEA analysis

Gene Set Enrichment Analysis (GSEA) identifies gene sets that are related to the difference of interest between samples (Subramanian et al. 2005).

The software is distributed by the Broad Institute and is freely available for use by academic and non-profit organisations. The Broad also provide a number of very well curated gene sets for testing against your data - the Molecular Signatures Database (MSigDB). These are collections of human genes. Fortunately, these lists have been translated to mouse equivalents by the Walter+Eliza Hall Institute Bioinformatics service and made available for download. They are now also available from a recent R package msigdbr, which we will use.

Let's load `msigdbr` now.

```
library(msigdbr)
```

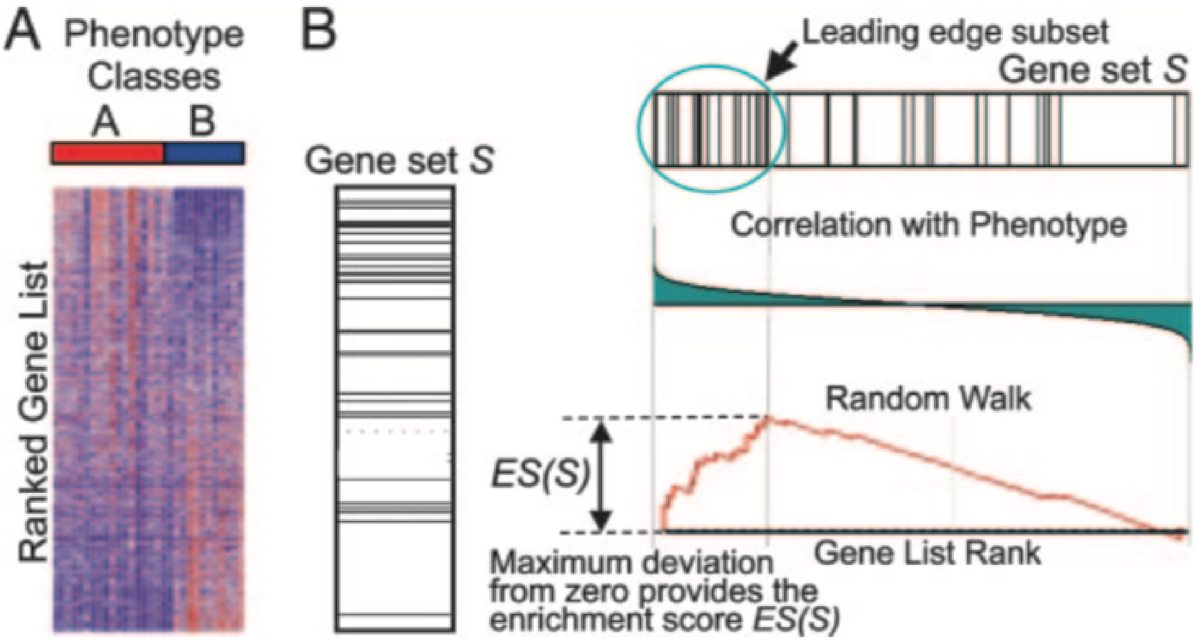
Method

The analysis is performed by:

1. ranking all genes in the data set

2. identifying in the ranked data set the rank positions of all members of the gene set
3. calculating an enrichment score (ES) that represents the difference between the observed rankings and that which would be expected assuming a random rank distribution.

The article describing the original software is available here, while this commentary on GSEA provides a shorter description.



We will use `clusterProfiler`'s `GSEA` package (Yu et al. 2012) that implements the same algorithm in R.

Rank genes

We need to provide `GSEA` with a vector containing values for a given gene mtric, e.g. log(fold change), sorted in decreasing order.

To start with we will simply use a rank based on their fold change.

We must exclude genes with no Entrez ID.

Also, we should use the shrunk LFC values.

```
rankedGenes <- shrink.d11 %>%
  drop_na(Entrez) %>%
  mutate(rank = logFC) %>%
  arrange(-rank) %>%
  pull(rank,Entrez)
```

Load pathways

We will load the MSigDB Hallmark gene set with `msigdbr`, setting the `category` parameter to 'H' for **H**allmark gene set. The object created is a `tibble` with information on each {gene set; gene} pair (one per row). We will only keep the the gene set name, gene Entrez ID and symbol, in mouse.

```
m_H_t2g <- msigdbr(species = "Mus musculus", category = "H") %>%
  dplyr::select(gs_name, entrez_gene, gene_symbol)
```

Conduct analysis

Arguments passed to `GSEA` include:

- ranked genes
- pathways
- gene set minimum size
- gene set maximum size

```
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...
```

Let's look at the top 10 results.

```
# have function to format in scientific notation
format.e1 <- function(x) (sprintf("%.1e", x))
# show table
gseaRes %>%
  arrange(desc(abs(NES))) %>%
  top_n(10, -p.adjust) %>%
  dplyr::select(-core_enrichment) %>%
  dplyr::select(-Description) %>%
  data.frame() %>%
  remove_rownames() %>%
  # format
  mutate(enrichmentScore=formatC(NES, digits = 3)) %>%
  mutate(NES=formatC(NES, digits = 3)) %>%
  # format p-values
  modify_at(
    c("pvalue", "p.adjust", "qvalues"),
    format.e1
  ) %>%
  DT::datatable(options = list(dom = 't'))
```

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

Enrichment score plot

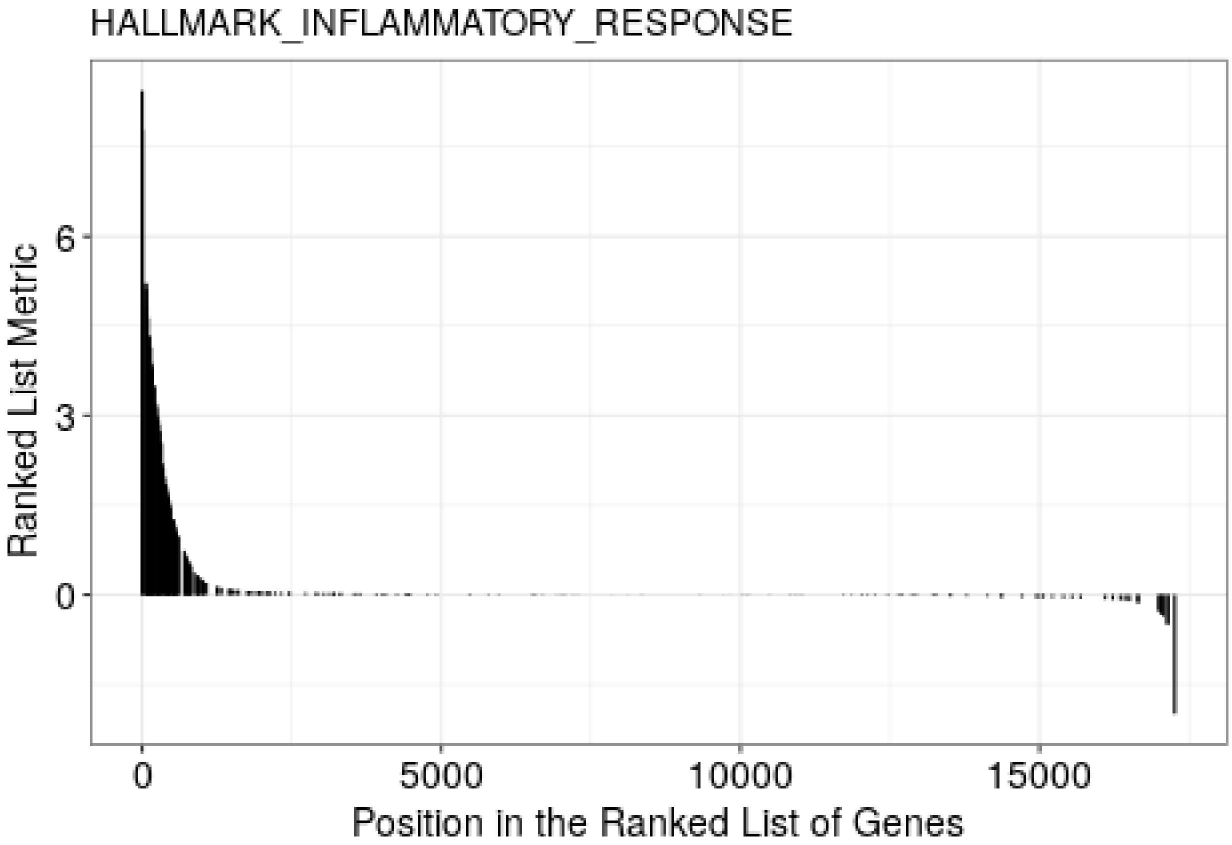
The enrichment score plot displays along the x-axis that represents the decreasing gene rank:

- genes involved in the pathway under scrutiny: one black tick per gene in the pathway (no tick for genes not in the pathway)
- the enrichment score: the green curve shows the difference between the observed rankings and that which would be expected assuming a random rank distribution.

```
# HALLMARK_INFLAMMATORY_RESPONSE is 4th
topx <- match("HALLMARK_INFLAMMATORY_RESPONSE", data.frame(gseaRes)$ID)
```

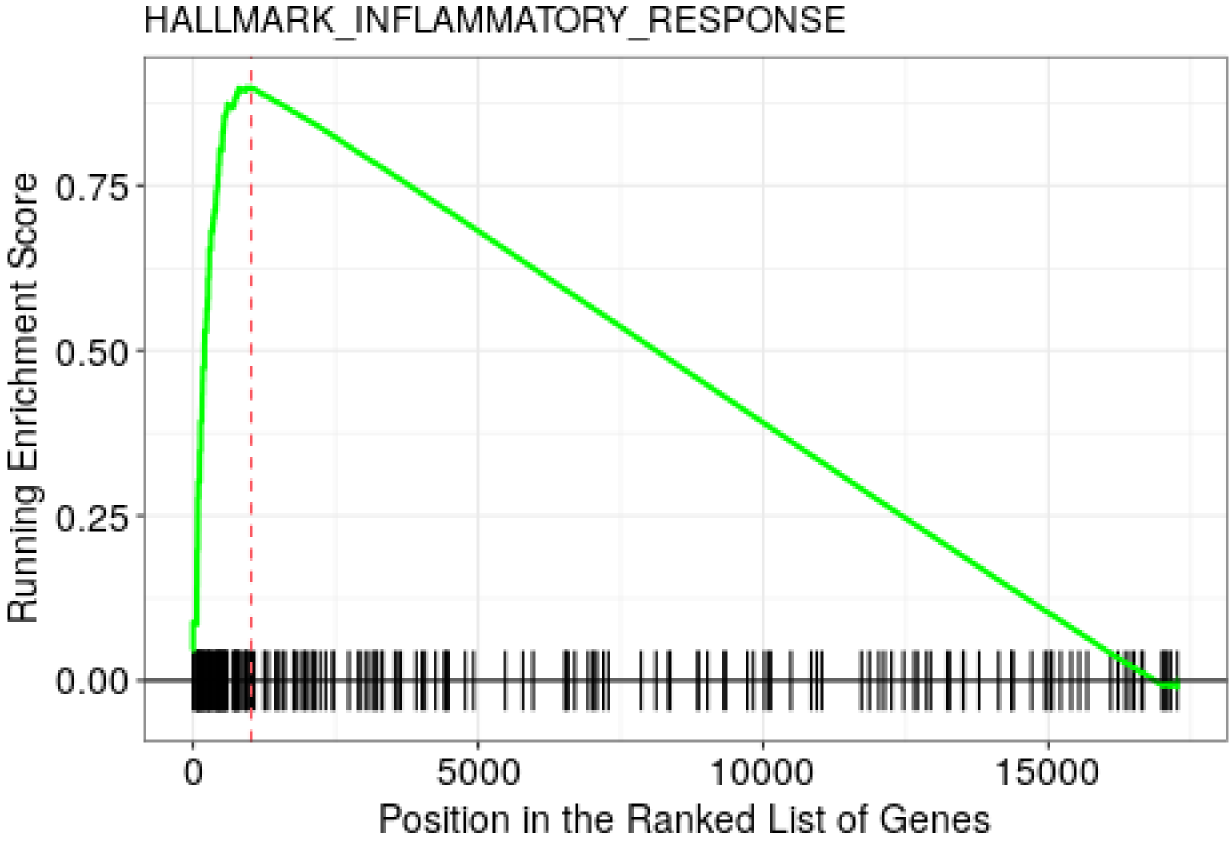
Gene log(fold change):

```
gseaplot(gseaRes, geneSetID = topx, by = "preranked", title = gseaRes$Description[topx])
```



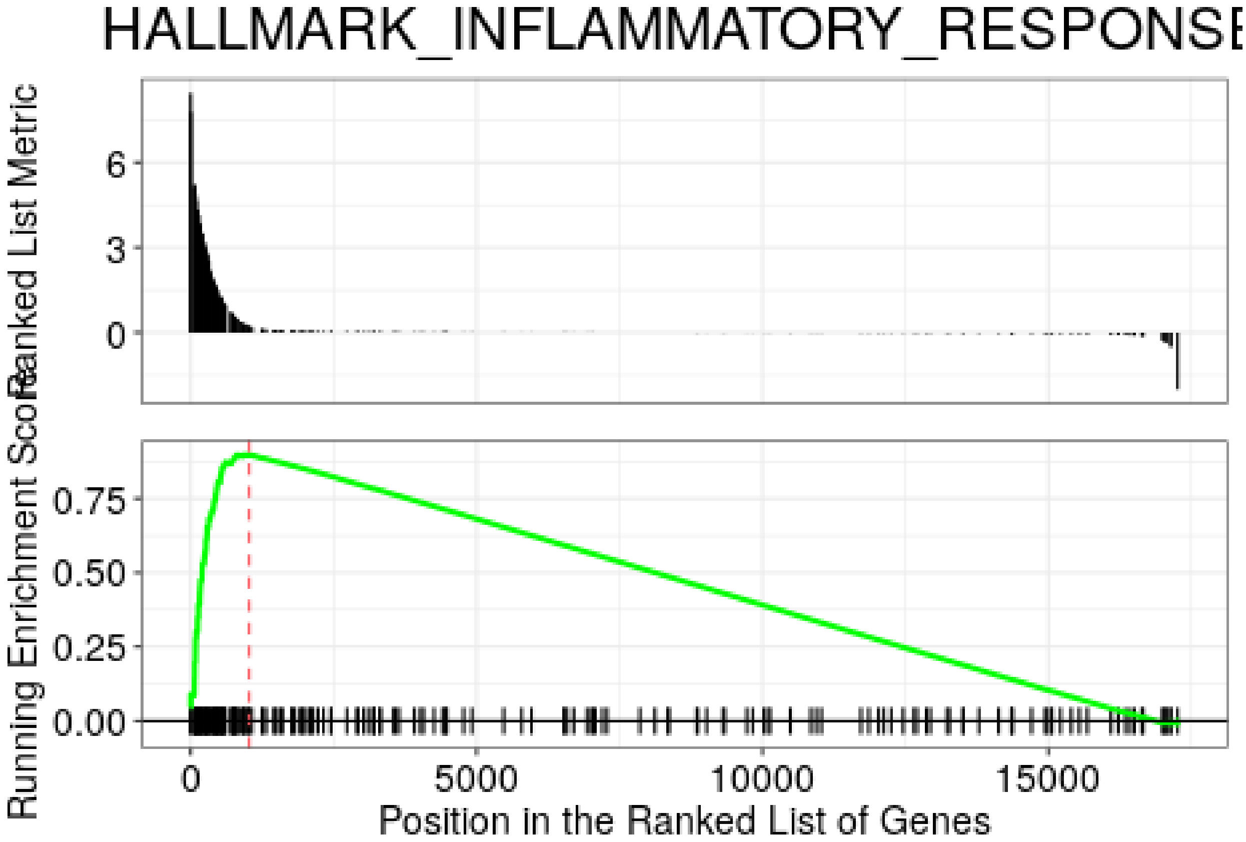
Running score:

```
gseaplot(gseaRes, geneSetID = topx, by = "runningScore", title = gseaRes$Description[topx])
```



Both the log(fold change) and running score:

```
gseaplot(gseaRes, geneSetID = topx, title = gseaRes$Description[topx])
```



Remember to check the GSEA article for the complete explanation.

Exercise 3

Another common way to rank the genes is to order by pvalue while 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 statistical 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. Extended: Do results differ between ranking scheme?
   Extended: Do results differ between d11 and d33, with the significance-based ranking scheme?

References

Luo, Weijun, Brouwer, and Cory. 2013. "Pathview: An r/Bioconductor Package for Pathway-Based Data Integration and Visualization." *Bioinformatics* 29 (14): 1830–31. https://doi.org/10.1093/bioinformatics/btt285.

Subramanian, Aravind, Pablo Tamayo, Vamsi K. Mootha, Sayan Mukherjee, Benjamin L. Ebert, Michael A. Gillette, Amanda Paulovich, et al. 2005. "Gene Set Enrichment Analysis: A Knowledge-Based Approach for Interpreting Genome-Wide Expression Profiles.” *Proceedings of the National Academy of Sciences* 102 (43): 15545–50. https://doi.org/10.1073/pnas.0506580102.

Yu, Guangchuang, Li-Gen Wang, Yanyan Han, and Qing-Yu He. 2012. “clusterProfiler: An r Package for Comparing Biological Themes Among Gene Clusters.” *OMICS: A Journal of Integrative Biology* 16 (5): 284–87. https://doi.org/10.1089/omi.2011.0118.