06: GSEA

Mariavittoria Pizzinga, Eneko Villanueva,Rayner Queiroz, Manasa Ramakrishna, Tom Smith

November 19, 2019

Contents

1. Introduction 1
2. Reading in data 2
3. Gene sets of interest 2
    3a. Home-made gene sets 2
    3b. Gene sets based on GO terms 3
4. Assessing hits 4
    4a. Specific pathways 4
    4b. Plotting enrichment plots for specific pathways 4

1. Introduction

Since we don't have any unifying functional themes for the proteins in our analysis, we use Gene Set Enrichment Analysis (GSEA) to work out if there are any genesets with which are data aligns. The goal of GSEA is to determine whether members of a gene set S (in our case proteins with p-value < 0.05), tend to occur toward the top (or bottom) of the list L, in which case the gene set is correlated with the phenotypic class distinction. In our case, this list L could be things like "Amino acyl transferase genes", "Unfolded protein response genes" and so on.

As a process, we would first rank our list of DE genes either by fold change or by p-value or by log odds score (B) and then pick a gene set we are interested in comparing it to eg: AAtransferases. We start at the top of our ranked list. If the protein at the top of our list is in the AAtransferase list, then a positive number gets added to the running total score. Then we move to the next protein and if that one is also in the AAtransferase list, the score goes up, else the score goes down. Hence you see the craggy peaks in the line graphs depicted below. Each vertical bars that cuts the flat line at the top represents a protein/gene from our list that belongs to the gene set of interest.

GSEA then provides an enrichment score which reflects the extent to which our dataset is represented at the start (top) or end (bottom) of the list L. If we see majority of our genes are

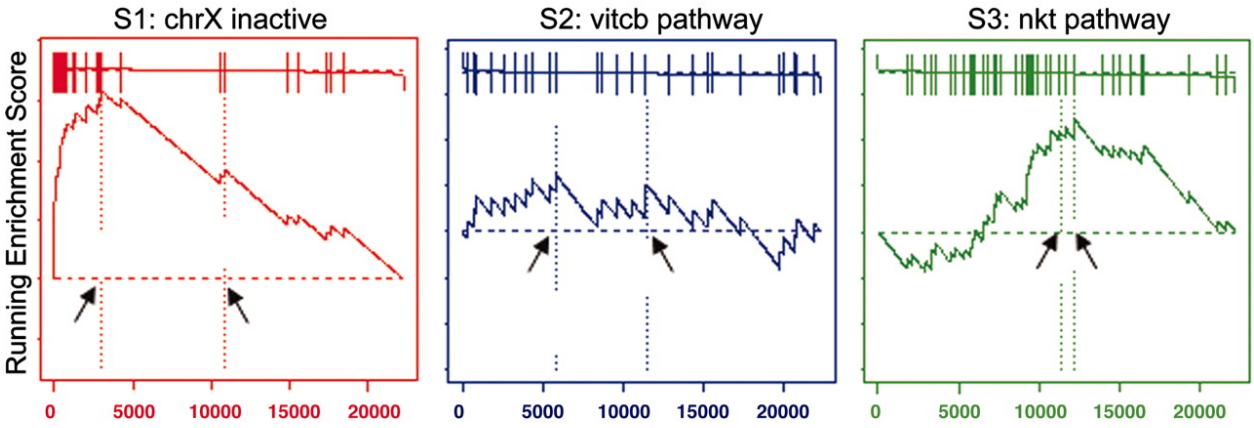

Figure 1: Potential outcomes from GSEA

1. at the top of a list, then the score is high and we can say that our list is significantly enriched for that term (S1 above)
2. at the bottom of a list, then our data is significantly depleted for that term
3. scattered randomly, then the score is generally low and we have no significant enrichment (S2 above)
4. in the middle of the list but enriched, then the score is lower than at when at either end of the list and may not be significant (S3 above)

The enrichment score is then normalised and a significance level for the enrichment score is derived using permutation testing. This significance level is then corrected for multiple-hypotheses. Overall we get an enrichment score(ES), a normalised enrichment score (NES), a pvalue (pva), an adjusted pvalue (padj) which we can use to interpret the data.

2. Reading in data

The first step is to read in data from the previous step of analysis. In our case, we have the results of a differential analysis between Ctrl and 100uM and Ctrl and 400uM Arsenite treated cells. We have approximately 200 proteins that are DE in each comparison. We also open up the full mapping of human proteins to go terms which is saved in the RDS "../shared_files/h_sapiens_go_full.rds".

```
Ctrl.100uM <- readRDS("../results/Ctrl.100uM.rds")
Ctrl.400uM <- readRDS("../results/Ctrl.400uM.rds")
# Ctrl.400uM[,45:52] %>%
# tibble::rownames_to_column() %>%
# arrange(desc(logFC)) Ctrl.400uM[,45:52] %>%
# arrange(desc(logFC)) %>% tail() head(Ctrl.400uM)

human_go <- readRDS("../shared_files/h_sapiens_go_full.rds")
```

3. Gene sets of interest

3a. Home-made gene sets

In the first pass of GSEA analysis, we set up some manual gene sets which we'd like to assess ur data against. These include translation initiation, elongation, tRNA aminoacyltransferases etc...

Additionally, we take the list of proteins in our study and map every one of them to their entire GO term repertoire. Having done that, we then reverse the mapping to be centred around each GO terms i.e for each GO term, we get a list of all proteins that are annotated with it. Since this bit of code is time consuming, we save it as an RDS and call on it later as the analysis progresses.

```
# Gene sets of interest
translation_init_activity <- human_go %>% filter(GO.ID ==
    "GO:0003743") %>% pull(UNIPROTKB)
translation_elong_activity <- human_go %>% filter(GO.ID ==
    "GO:0003746") %>% pull(UNIPROTKB)
translation_term_activity <- human_go %>% filter(GO.ID ==
    "GO:0008079") %>% pull(UNIPROTKB)
tRNA_AA <- human_go %>% filter(GO.ID == "GO:0004812") %>%
    pull(UNIPROTKB)
translocon <- human_go %>% filter(GO.ID == "GO:0006616") %>%
    pull(UNIPROTKB)

# GO terms of interest (gotoi)
gotoi <- list(translation_init_activity, translation_elong_activity,
```

```
    translation_term_activity, tRNA_AA, translocon)

names(gotoi) <- c("GO_0003743_Initiation", "GO:0003746:Elongation",
    "GO:0008079:Translation", "GO:0004812:tRNA-AA",
    "GO:0006616:Translocon")
print(gotoi)

# Gene set for each GO term The set of GO terms is
# same for both Ctrl.400uM and Ctrl.100uM) as it is
# the same set of proteins that were analysed using
# TMT

all_go_terms <- human_go %>% filter(UNIPROTKB %in%
    rownames(Ctrl.400uM)) %>% pull(TERM) %>% unique()

all_go <- vector("list", length = length(all_go_terms))
names(all_go) <- all_go_terms

for (x in all_go_terms) {
    all_go[[x]] <- human_go %>% filter(TERM == x) %>%
        pull(UNIPROTKB)
}

print(head(all_go, 1))
saveRDS(all_go, "../results/all_go.rds")

# Plot enrichment for our own defined genesets
lapply(gotoi, function(x) plotEnrichment(x, ranks))
```

3b. Gene sets based on GO terms

Using 'all_go' which contains and extensive protein map onto GO terms, we perform a GSEA and display the green line graphs for the top 10 hits. We start by ranking out set of proteins based on logFC in either the 100uM or 40-uM arsenite comparison. Plots can be found under "../plots/*Ranked-enrichment-data.pdf"

```
all_go = readRDS("../results/all_go.rds")

# Ranking the 100uM dataset using logFC
ranks.100 <- rev(sort(Ctrl.100uM$logFC))
names(ranks.100) <- rownames(Ctrl.100uM)
head(ranks.100)
```

```
##    Q12830    Q12904    P62241    Q15366    P62277    P55209
## 1.5070948 1.3799502 1.3006645 1.1763754 1.1061871 0.9889491
```

```
# Ranking the 100uM dataset using logFC
ranks.400 <- rev(sort(Ctrl.400uM$logFC))
names(ranks.400) <- rownames(Ctrl.400uM)
head(ranks.400)
```

```
##   P23396   Q9Y314   Q12904   Q16637   P38432   O15479
## 1.569130 1.482512 1.372430 1.265899 1.250935 1.237015
```

```
# Plots ranked data as well as enrichment
fgseaRes.100 = plotRanksEnrich(all_go, ranks.100, "Ctrl-vs-100uM-Arsenite")
```

```
fgseaRes.400 = plotRanksEnrich(all_go, ranks.400, "Ctrl-vs-400uM-Arsenite")
```

4. Assessing hits

Here we are trying to look at the ranking+enrichment for our terms of interest. We start by looking at which terms are represented in our data

4a. Specific pathways

We are looking at specific pathways related to the ribosome, endoplasmic reticulum and translation to see if there are any pathways enriched for in our gene set. There are some that are significant before correcting for multiple hypothesis testing such as endoplasmic reticulum unfolded protein response (p = 0.01152738), regulation of translation (p = 0.006944444) and tRNA aminoacylation for protein translation (p = 0.029619182).

```
fgseaRes.100 %>% arrange(pval) %>% filter(grepl("Ribosome",
    pathway, ignore.case = TRUE))
fgseaRes.100 %>% arrange(pval) %>% filter(grepl("Endoplasmic",
    pathway, ignore.case = TRUE))
fgseaRes.100 %>% arrange(pval) %>% filter(grepl("Translation",
    pathway, ignore.case = TRUE))

fgseaRes.400 %>% arrange(pval) %>% filter(grepl("Ribosome",
    pathway, ignore.case = TRUE))
fgseaRes.400 %>% arrange(pval) %>% filter(grepl("Endoplasmic",
    pathway, ignore.case = TRUE))
fgseaRes.400 %>% arrange(pval) %>% filter(grepl("Translation",
    pathway, ignore.case = TRUE))
```

4b. Plotting enrichment plots for specific pathways

We draw a volcano plot and an enrichment plot showing the terms enriched and the genes that contribute to the enrichment based on the pathways of interest from above.

In the volcano plots, turquoise represents non-significant proteins and purple represents significant ones after arsenite treatment compared to controls. Asterisk shapes indicate which of these proteins are in the category of interest eg : 'structural constituent of ribosome'. Black labels indicate those proteins that are significantly differentially expressed upon arsenite treatment AND in our pathway of interest.

In the GSEA plot, along the x-axis, we plot our ranked list of proteins in descending order of log fold-change in expression vs Ctrl. There are 896 such proteins. Then we mark with vertical bars all the locations of proteins that belong to our geneset of interest. Finally, we follow the GSEA method to draw the green line where the enrichment score is a running sum and goes up or down depending on whether a protein is in the gene set of interest or not. Genesets with early or late peaks are most relevant. This varies in the 100uM vs 400uM treated cells and plots can be found under "../plots/*features-of-interest-volcano-random-walks.pdf"

```
# 100uM
pdf("../plots/Ctrl-vs-100uM-Arsenite_GSEA-features-of-interest-volcano-random-walks.pdf",
    paper = "a4r", width = 14, height = 8)
plot_foi_trends(Ctrl.100uM, all_go, "regulation of translation",
    ranks.100)
```

```
## [1] "regulation of translation"
```

```
plot_foi_trends(Ctrl.100uM, all_go, "tRNA aminoacylation for protein translation",
    ranks.100)
```

```
## [1] "tRNA aminoacylation for protein translation"
```

```
plot_foi_trends(Ctrl.100uM, all_go, "structural constituent of ribosome",
    ranks.100)
```

```
## [1] "structural constituent of ribosome"
```

```
plot_foi_trends(Ctrl.100uM, all_go, "endoplasmic reticulum unfolded protein response",
    ranks.100)
```

```
## [1] "endoplasmic reticulum unfolded protein response"
```

```
dev.off()
```

```
## pdf
##   2
```

```
# 400uM
pdf("../plots/Ctrl-vs-400uM-Arsenite_GSEA-features-of-interest-volcano-random-walks.pdf",
    paper = "a4r", width = 14, height = 8)
plot_foi_trends(Ctrl.400uM, all_go, "regulation of translation",
    ranks.400)
```

```
## [1] "regulation of translation"
```

```
plot_foi_trends(Ctrl.400uM, all_go, "tRNA aminoacylation for protein translation",
    ranks.400)
```

```
## [1] "tRNA aminoacylation for protein translation"
```

```
plot_foi_trends(Ctrl.400uM, all_go, "structural constituent of ribosome",
    ranks.400)
```

```
## [1] "structural constituent of ribosome"
```

```
plot_foi_trends(Ctrl.400uM, all_go, "endoplasmic reticulum unfolded protein response",
    ranks.400)
```

```
## [1] "endoplasmic reticulum unfolded protein response"
```

```
dev.off()
```

```
## pdf
##   2
```

```
# Example plot
plot_foi_trends(Ctrl.400uM, all_go, "endoplasmic reticulum unfolded protein response",
    ranks.400)
```

```
## [1] "endoplasmic reticulum unfolded protein response"
```

4c. Overall enriched/depleted genesets To get a feel for the overall

```
gseaTab(all_go, fgseaRes.100, ranks.100, "Ctrl-vs-100uM-Arsenite")
```

```
## pdf
##   2
```

```
gseaTab(all_go, fgseaRes.400, ranks.400, "Ctrl-vs-400uM-Arsenite")
```

```
## pdf
##   2
```