Compare *sSNAPPY* against other pathway analysis methods

2023-09-21

In this R Markdown, we compare three existing pathway analysis methods: *SPIA*, *GSEA*,and *fry* against the single-sample pathway analysis method *sSNAPPY*.

Preparation

Firstly, packages required and the example dataset used in the main *sSNAPPY* manuscript are loaded in.

```
library(sSNAPPY)
library(tidyverse)
library(magrittr)
library(ggplot2)
library(patchwork)
library(AnnotationHub)
library(edgeR)
library(patchwork)
library(colorspace)
library(fgsea)
library(DT)
library(UpSetR)
library(graphite)
library(SPIA)
library(pander)
```

```
formatP <- function(p, m = 0.0001){
    out <- rep("", length(p))
    out[p < m] <- sprintf("%.2e", p[p<m])
    out[p >= m] <- sprintf("%.4f", p[p>=m])
    out
}
```

```
readr::local_edition(1)
logCPM <- readr::read_tsv(here::here("data/logCPM.tsv")) %>%
    column_to_rownames("entrezid")
sample_meta <- read_tsv(here::here("data/sample_meta.tsv"), col_types = "cfccncnc")
dge <- readRDS(here::here("data/dge.rds"))
```

Reactome(Gillespie et al. 2021) pathway topology information was retrieved using the `retrieve_topology` functionc from *sSNAPPY*.

```
gsTopology <- retrieve_topology(database = "reactome", species = "hsapiens")
```

Chemotherapy-induced significant pathway perturbation that were detected using *sSNAPPY* on group level were loaded in.

```
sSNAPPY_rs <- read_tsv("data/sSNAPPY_output.tsv")
```

Pathway analysis using other methods

SPIA

Firstly, *sSNAPPY* was compared against an existing topology-based method *SPIA*(Tarca et al. 2009). While the scoring algorithm of *sSNAPPY* was adopted from *SPIA*, there are two fundamental differences between the two methods. *SPIA* relies on the detection of differentially expressed genes (DEGs) and only allows group-level analysis(Tarca et al. 2009). In comparison, *sSNAPPY* does not require any pre-selection of genes and can be used to score pathway perturbation within indiviudal samples.

DE Analysis

To apply *SPIA*, differential expression analysis was firstly performed through *edgeR*(Smyth 2004). Model matrix in the form of `model.matrix(~ 0 + patient_id + treatment_phase, data = dge$samples)` was constructed to nest samples by patients.

```
X <- model.matrix(~ 0 + patient_id + treatment_phase,
                  data = dge$samples %>%
                      mutate(treatment_phase = factor(treatment_phase, levels = c("treatment-naive", "p
) %>%
    set_colnames(str_remove_all(colnames(.), "patient_id|treatment_phase")) %>%
    .[,colSums(.) != 0]
dge <- estimateDisp(dge, design = X, robust = TRUE)
fit <- glmQLFit(dge)
```

```
alpha <- 0.05
topTable <- glmQLFTest(fit, coef = "post-NACT") %>%
      topTags(n = Inf) %>%
      .[["table"]] %>%
      as_tibble() %>%
      mutate(
        location = paste0(seqnames, ":", start, "-", end, ":", strand),
        rankingStat = -sign(logFC)*log10(PValue),
        signedRank = rank(rankingStat),
        DE = FDR < alpha
      ) %>%
      dplyr::select(
        gene_id, gene_name, logCPM, logFC, PValue, FDR,
        location, gene_biotype, entrezid, ave_tx_len, gc_content,
        rankingStat, signedRank, DE
      )
DEGs <- topTable %>%
    dplyr::filter(DE)
```

Using a FDR cut-off of 0.05, 49 DEGs were detected among the 10098 tested genes.

While in the full *SPIA* workflow, both topology-based perturbation analysis and conventional over-representation analysis will be performed. We only compared the results of *sSNAPPY* against the perturbation analysis component of *SPIA*.

```
graphite_reactome <- pathways("hsapiens", "reactome")
graphite_reactome <- convertIdentifiers(graphite_reactome, "ENTREZID")
prepareSPIA(graphite_reactome, "graphite_reactome")
DE_vector <- DEGs$logFC %>%
    set_names(paste("ENTREZID:", DEGs$entrezid, sep = ""))
all_entrez <- dge$genes %>%
    unnest(entrezid) %>%
    drop_na() %>%
    pull(entrezid) %>%
    paste("ENTREZID:", ., sep = "")
spia_res <- runSPIA(de = DE_vector, all = all_entrez, "graphite_reactome")
saveRDS(spia_res, here::here("data/spia_res.rds"))
```

Since *SPIA* only considers pathways with DEGs and a low number of DEGs were detected in this dataset, it is not surprising to observe that none of the Reactome pathway was considered as significantly perturbed by *SPIA* (FDR < 0.05). *SPIA* output was ranked by the perturbation p-value and the 5 most highly ranked pathways are displayed.

```
spia_res %>%
    mutate(Pert_FDR = p.adjust(pPERT, method = "fdr")) %>%
    arrange(pPERT) %>%
    .[1:5,] %>%
    dplyr::select(
        Pathway = Name,
        `Perturbaton Score` = tA,
        PValue = pPERT, FDR = Pert_FDR
    ) %>%
    pander(
        caption = "*Top 5 Reactome pathways with smallest pertubation p-values
        in SPIA output. None of the pathway was considered as significantly
        perturbed using a FDR cut-off of 0.05.*"
    )
```

Table 1: *Top 5 Reactome pathways with smallest pertubation p-values in SPIA output. None of the pathway was considered as significantly perturbed using a FDR cut-off of 0.05.*

Pathway	Perturbaton Score	PValue	FDR
Phosphorylation of CD3 and TCR zeta chains	2.819	0.001	0.129
PD-1 signaling	9.909	0.002	0.129
Interferon alpha/beta signaling	-8.845	0.019	0.6128
Cytosolic sensors of pathogen-associated DNA	-6.581	0.019	0.6128
IKK complex recruitment mediated by RIP1	1.411	0.042	0.7453

GSEA

Following *SPIA*, we also performed a gene-set enrichment analysis (*GSEA*)(Subramanian et al. 2005). Instead of requiring pre-selection of DEGs, *GSEA* requires a ranking for each gene. We calculated the ranking statistic of genes basing on the DE analysis results by `-sign(logFC)*log10(PValue)`. A named vector where the values are the ranking statistic and the names are genes' entrez id was generated.

```
load(system.file("extdata", "entrez2name.rda", package = "sSNAPPY"))
temp <- topTable %>%
        mutate(entrezid = paste("ENTREZID:", entrezid, sep = "")) %>%
        drop_na()
ranked_list <- temp %>%
        pull(rankingStat) %>%
        set_names(temp$entrezid)
```

Since *GSEA* is not a topology-based method, the only pathway information required is genes that are included in each pathway. Therefore, row names of each topology matrix were extracted.

```
reactome_gs <- sapply(gsTopology, rownames)
```

```
gsea <-  fgsea(reactome_gs, ranked_list)
gsea_sig <- gsea %>%
    dplyr::filter(padj < 0.05)
```

Using *GSEA* and a significance cut-off of FDR < 0.05, 240 out of the 1876 tested pathways were considered as significantly enriched, among which 68 pathways were also found to be significantly perturbed by *sSNAPPY*.

While *GSEA* does not account for the pathway topology information, it returns a signed normalised enrichment score (NES) for each pathway. A positive NES indicates that the pathway is more enriched in genes that are highly expressed(Subramanian et al. 2005). However, NES is often over-interpreted to infer the directionality of changes in pathway activity. We compared the directional changes that were predicted by *sSNAPPY* against the sign of NES. Interestingly, the directionality aligned for all pathways that were considered as significantly impacted by both methods.

```
sSNAPPY_dir <- sSNAPPY_rs %>%
    mutate(Direction = ifelse(logFC < 0, "sSNAPPY_Inhibited", "sSNAPPY_Activated")) %>%
    split(.$Direction) %>%
    lapply(pull, gs_name)
gsea_sig_dir  <- gsea_sig %>%
    mutate(Direction = ifelse(NES < 0, "GSEA_Inhibited", "GSEA_Activated")) %>%
    split(.$Direction) %>%
    lapply(pull, pathway)
c(sSNAPPY_dir, gsea_sig_dir) %>%
    fromList() %>%
    upset(
        sets = colnames(.),
        keep.order = TRUE,
        queries = list(
        list(query = intersects,
             params = list("sSNAPPY_Inhibited", "GSEA_Inhibited"),
             color = "blue",
             active = T),
        list(query = intersects,
```

```
            params = list("sSNAPPY_Activated", "GSEA_Activated"),
            color = "red",
            active = T)
        )
    )
```

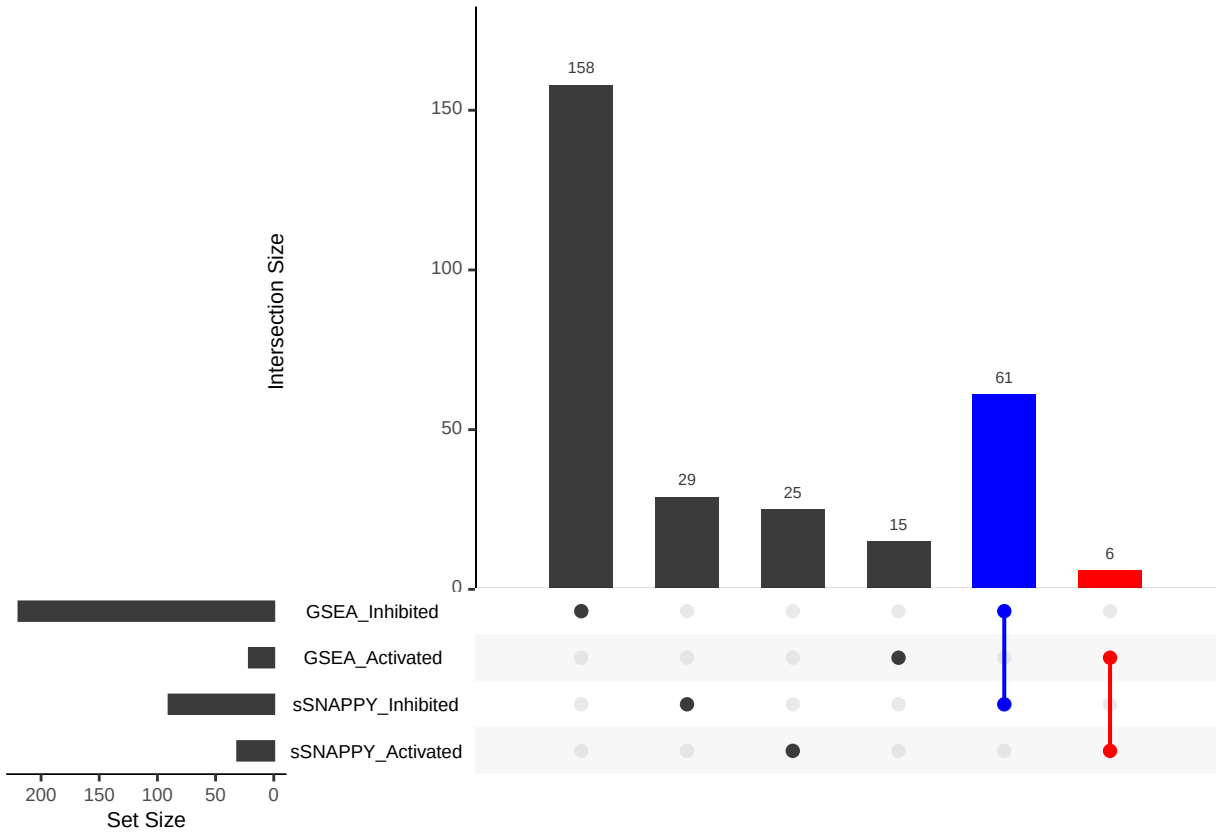

Figure 1: *Overlap between pathways that are considered as significantly impacted by GSEA and sSNAPPY.*

The full result obtained using GSEA was saved as `output/gsea.tsv`.

```
sSNAPPY_sig_path <- sSNAPPY_rs %>%
    dplyr::filter(adj.P.Val < 0.05) %>%
    pull(gs_name)
gsea %>%
    mutate(,
        `Sig in sSNAPPY` = ifelse(
            pathway %in% sSNAPPY_sig_path,
            TRUE, FALSE
        ),
        pathway = str_remove_all(pathway, "reactome.")
    ) %>%
    dplyr::select(
        pathway, padj, NES, `Sig in sSNAPPY`
    ) %>%
    write_tsv(
```

```
        file = here::here("output/gsea.tsv")
    )
```

fry

The other non-topology-based method applied to the example dataset is *fry*, which is a fast version of *roast* (rotation gene set testing)(Wu et al. 2010). Instead of relying on pre-performed DE analysis results, *fry*/*roast* only requires the logCPM matrix and a design matrix as input.

```
fry_res <- logCPM %>%
    set_rownames(paste("ENTREZID:", rownames(.), sep = "")) %>%
    fry(
        index = reactome_gs,
        design  = dge$design,
        contrast = "post-NACT",
        sort = "directional"
    ) %>%
  rownames_to_column("Pathway")
fry_sig <- fry_res %>%
    dplyr::filter(FDR < 0.05)
```

Using the directional version of *fry* and the same statistics threshold of FDR < 0.05, 132 pathways were considered as significantly enriched, 44 of which were also considered to be significantly perturbed by *sSNAPPY*.

fry also returns a 'Direction' column as part of its output, which indicates whether genes in the pathway turn to be more up- or down-regulated. The direction returned by *fry* was compared against the directional prediction made by *sSNAPPY*. While the directionality aligned for most of the pathways that were considered as significant by both methods, some disconcordance arose.

```
fry_sig_dir  <- fry_sig %>%
    mutate(Direction = ifelse(
        Direction == "Down", "fry_Inhibited", "fry_Activated"
    )) %>%
    split(.$Direction) %>%
    lapply(pull, Pathway)
c(sSNAPPY_dir, fry_sig_dir) %>%
    fromList() %>%
    upset(
        sets = colnames(.),
        keep.order = TRUE,
        queries = list(
        list(query = intersects,
             params = list("sSNAPPY_Inhibited", "fry_Inhibited"),
             color = "blue",
             active = T),
        list(query = intersects,
             params = list("sSNAPPY_Activated", "fry_Activated"),
             color = "red",
             active = T)
    )
    )
```

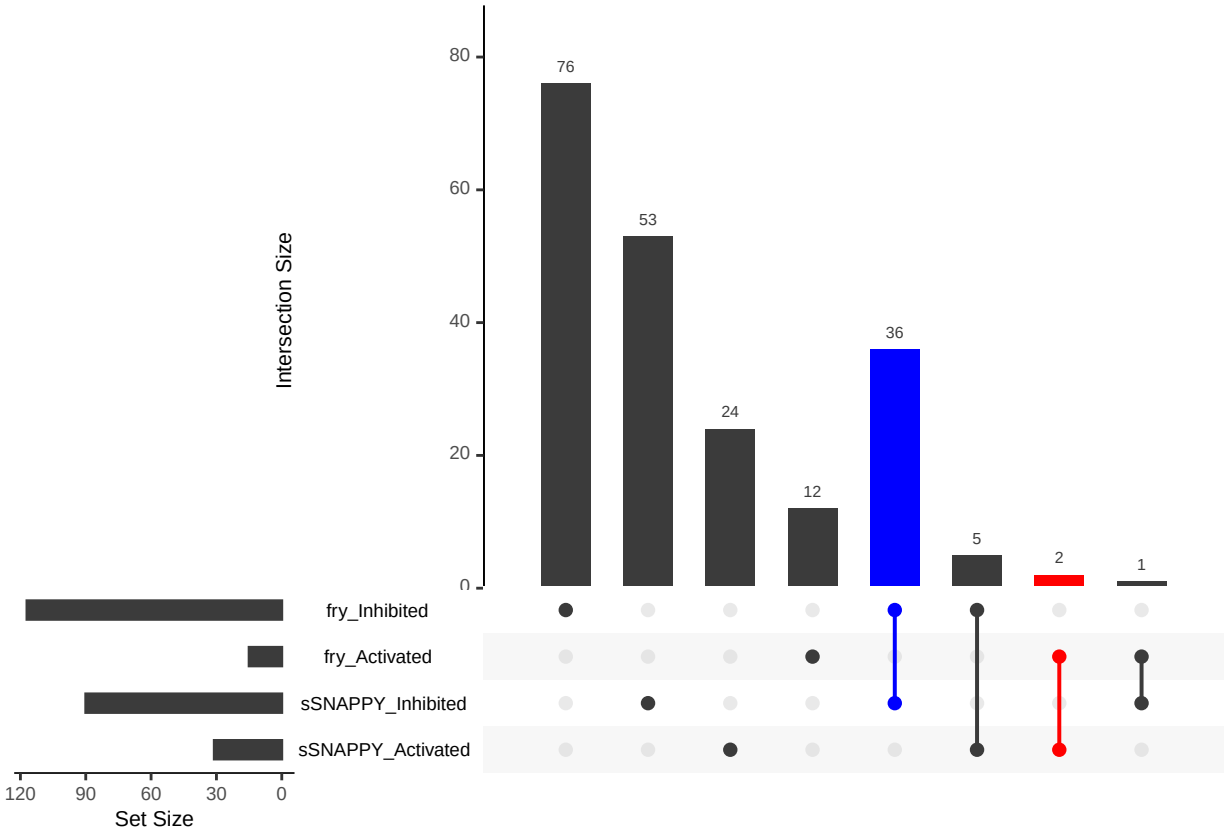

Figure 2: *Overlap between pathways that are considered as significantly impacted by fry and sSNAPPY.*

```
fry_act_sSN_inh <- intersect(sSNAPPY_dir$sSNAPPY_Inhibited, fry_sig_dir$fry_Activated)
```

For example, the pathway **Terminal pathway of complement** was considered as significantly inhibited by *sSNAPPY* but significant activated by *fry*. This pathway contains 8 genes but only gene *CLU* had detectable expression levels in this dataset and was found to be significantly up-regulated in the DE analysis.

```
terminal_gene <- entrez2name %>%
    dplyr::filter(entrezid %in% rownames(gsTopology[[fry_act_sSN_inh]])) %>%
    pull(mapTo)
topTable %>%
    dplyr::filter(
        gene_name %in% terminal_gene
    ) %>%
    dplyr::select(
        Gene = gene_name, logFC, FDR, DE
    ) %>%
    pander(
        caption = "The differential expression analysis output for the gene that is
        involved in the Reactome Terminal pathway of complement pathway."
    )
```

Table 2: The differential expression analysis output for the gene that is involved in the Reactome Terminal pathway of complement pathway.

Gene	logFC	FDR	DE
CLU	1.858	0.04351	TRUE

CLU encodes plasma protein clusterin that is able to inhibit the insertion of complement complexes into cell membranes by binding to them(Chauhan and Moore 2006). Therefore, this gene has a repressor role on the **Terminal pathway of complement** pathway, which is clear in the pathway topology provided by the Reactome database. Inspecting this pathway demonstrated the importance of incorporating pathway topology information to predict change in pathway activity and revealed the strength of *sSNAPPY* over non-topological methods such as *fry*.

The full result obtained using *fry* was saved as `output/fry.tsv`.

```
fry_res %>%
    mutate(
        `Sig in sSNAPPY` = ifelse(
            Pathway %in% sSNAPPY_sig_path,
            TRUE, FALSE
        ),
        Pathway = str_remove_all(Pathway, "reactome.")
    ) %>%
    dplyr::select(
        Pathway, FDR, `Sig in sSNAPPY`
    ) %>%
    write_tsv(
        file = here::here("output/fry.tsv")
    )
```

References

Chauhan, A. K., and T. L. Moore. 2006. "Presence of Plasma Complement Regulatory Proteins Clusterin (Apo j) and Vitronectin (S40) on Circulating Immune Complexes (CIC)." *Clinical & Experimental Immunology* 145 (3): 398–406. https://doi.org/10.1111/j.1365-2249.2006.03135.x.

Gillespie, Marc, Bijay Jassal, Ralf Stephan, Marija Milacic, Karen Rothfels, Andrea Senff-Ribeiro, Johannes Griss, et al. 2021. "The Reactome Pathway Knowledgebase 2022." *Nucleic Acids Research* 50 (D1): D687–92. https://doi.org/10.1093/nar/gkab1028.

Smyth, Gordon K. 2004. "Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments." *Statistical Applications in Genetics and Molecular Biology* 3 (1). https://doi.org/10.2202/1544-6115.1027.

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." *Proc. Natl. Acad. Sci. U. S. A.* 102 (43): 15545–50. https://doi.org/10.1073/pnas.0506580102.

Tarca, Adi Laurentiu, Sorin Draghici, Purvesh Khatri, Sonia S. Hassan, Pooja Mittal, Jung-sun Kim, Chong Jai Kim, Juan Pedro Kusanovic, and Roberto Romero. 2009. "A Novel Signaling Pathway Impact Analysis." *Bioinformatics* 25 (1): 75–82. https://doi.org/10.1093/bioinformatics/btn577.

Wu, D, E Lim, F Vaillant, M-L Asselin-Labat, JE Visvader, and GK Smyth. 2010. "ROAST: Rotation Gene Set Tests for Complex Microarray Experiments." *Bioinformatics* 26: 2176–82. https://doi.org/10.1093/bioinformatics/btq401.