sSNAPPY: a R/Bioconductor package for single-sample directional pathway perturbation analysis

Wenjun Liu[1], Ville-Petteri Mäkinen[2,3,4,5], Wayne D. Tilley[1], and Stephen M. Pederson[1,6,7]

[1]**Dame Roma Mitchell Cancer Research Laboratories, Adelaide Medical School, Faculty of Health and Medical Sciences, University of Adelaide, Adelaide, Australia**
[2]**Australian Centre for Precision Health, Cancer Research Institute, University of South Australia, Adelaide, Australia**
[3]**Computational and Systems Biology Program, Precision Medicine Theme, South Australian Health and Medical Research Institute, Adelaide, Australia**
[4]**Computational Medicine, Faculty of Medicine, University of Oulu, Oulu, Finland**
[5]**Center for Life Course Health Research, Faculty of Medicine, University of Oulu, Oulu, Finland**
[6]**Black Ochre Data Laboratories, Telethon Kids Institute, Adelaide, Australia**
[7]**John Curtin School of Medical Research, Australian National University, Canberra, Australia**

Abstract When analysing RNA-Seq data, a common outcome is to detect biological pathways with significantly altered activity between the conditions under investigation. The most common strategies test for over-representation within pre-defined gene-sets for genes showing changed expression[1, 2], but without accounting for gene-gene interactions encoded by pathway topologies, and not being able to directly predict the directional change of pathway activity. To address these issues, we have deloped a single-sample pathway perturbation analysis method *sSNAPPY*, now available as an R/Bioconductor package, which leverages pathway topology information to compute pathway perturbation scores, and predicts the potential direction of change across a set of pathways. Here, we demonstrate the use of *sSNAPPY* by applying the method to public scRNA-seq data, derived from ovarian cancer patient tissues collected before and after chemotherapy. Not only were we able to replicate results reported in the original study, but *sSNAPPY* was also able to detect significant perturbation of other biological processes, yielding far greater insight into the response to treatment. *sSNAPPY* represents a novel pathway analysis strategy that takes into consideration of pathway topology to predict impacted biology pathways, both within related samples and across treatment groups. In addition to not relying on the detection of differentially expressed genes, the method and associated R package offer important flexibility and provide powerful visualisation tools.

Keywords

RNA-seq, pathway enrichment, R package, topology, KEGG, scRNA-seq

Corresponding Author wenjun.liu@adelaide.edu.au

R version: R version 4.2.3 (2023-03-15)

Bioconductor version: 3.16

Package: 1.3.4

Introduction

Using pathway enrichment analysis to gain biological insights from gene expression data is a pivotal step in the analysis and interpretation of RNA-seq data, for which numerous methods have been developed (reviewed in [3, 4]). Many existing methods tend to view pathways simply as a collection of gene names, as seen in those relying on the detection of differentially expressed genes and applying over-representation analysis (ORA) strategies, and those scoring all genes using functional class scoring (FCS), such as in Gene Set Enrichment Analysis (GSEA) [1], arguably the most widely-used approach. However, databases such as the Kyoto Encyclopaedia of Genes and Genomes (KEGG)[5] and WikiPathways[6] capture not only which genes are implicated in a certain biological process but also their interactions, activating or inhibitory roles, and their relative importance within the pathway, all of which are overlooked in ORA- and FCS-based approaches.

To fully utilise that additional information, the latest generation of pathway analysis approaches include many which are topology-based such as SPIA[7], DEGraph[8], NetGSA[9] and PRS[10], as well as others which explicitly model inter-gene correlations[11]. Despite differences in the null hypotheses tested across these approaches, overall, they have demonstrated enhanced sensitivity and specificity due to their abilities to take gene-gene interconnections into account[12, 13]. Nevertheless, most topology-based methods focus only on comparing activities of pathways between two treatment groups and cannot be used to score individual samples. However, in heterogenous data where more than one factor may be influencing our observations[14], incorporating scoring within paired samples may be desirable and may be able to reveal more nuanced insights. To address this gap, we present a Single-Sample Network and Pathway Perturbation analysis methodology called *sSNAPPY*, available as an R/Bioconductor package. This article defines how *sSNAPPY* computes changes in gene expression within paired samples, and propagates this through gene-set topologies to predict the perturbation in pathway activities within paired samples, before providing summarised results across an entire dataset which would include more robust levels of biological replication. The practical usage of the *sSNAPPY* R/Bioconductor package will be illustrated through the analysis of a public scRNA-seq dataset using pseudo-bulk strategies.

Methods

Implementation

sSNAPPY is an R package that has been reviewed and published on the open-source bioinformatics software platform Bioconductor with all source code available via GitHub. The methodology itself is topology-based, designed to compute directional, single-sample, pathway perturbation scores in gene expression datasets with a matched-pair, or nested design (eg. samples collected before and after treatment). This allows for the detection of pathway perturbations within all samples from a treatment group, but also within individual samples.

To run *sSNAPPY*, the only required data is a log-transformed expression matrix (e.g. logCPM) with matching sample metadata describing treatment groups and the nested structure. It is assumed that all pre-processing has been performed beforehand, such as the exclusion of low-signal genes or normalisation to minimise technical artefacts like GC-bias. The first step, performed internally by *sSNAPPY*, is to estimate sample-specific log fold-change ($\delta_{ghi} = \mu_{ghi} - \mu_{g0i}$) for a treatment h across all genes g within each sample i, by subtracting expression estimates for the baseline samples $\hat{\mu}_{g0i}$ from those in the treatment group h. Since it has been shown that in RNA-seq data, genes with lower expression tend to have larger variance and larger estimates of change[15], we utilise a gene-level weighting strategy to de-emphasise logFC estimates for low-abundance genes. Gene-level weights w_g are obtained in a treatment-agnostic manner by fitting a loess curve through the relationship between observed gene-level variance ($\widehat{\sigma}_g$) and average signal ($\bar{\mu}_{g..}$), and taking the inverse of the loess-predicted variance as the weight $w_g = a/f(\bar{\mu}_{g..})$, where $f(\bar{\mu}_{g..})$ is the predicted value from the loess curve and the constant a ensures $\sum w_g = 1$. We then use these weighted estimates of logFC for calculation of all pathway perturbation scores.

sSNAPPY was built upon the group-level topology-based scoring algorithm initially proposed in R package SPIA[16] to propagate genes' changes in expression through pathway topologies to compute a perturbation score for each pathway. By modifying the algorithm to incorporate single-sample, weighted estimates of changes in expression we are able to quantify changes in a pathway within a given sample, and then model these across all samples within a treatment group. Thus, we define the single-sample perturbation score (S_{hip}) for a given pathway p, sample i and treatment h:

$$S_{hip} = \sum_{g \in G_p} [S_{ghip} - \delta^*_{ghi}], \text{ where}$$

$$S_{ghip} = \delta^*_{ghi} + \sum_{g' \in U_{gp}} \beta_{gg'p} \frac{S_{g'hip}}{N_{g'p}}$$

where:

- G_p represents the set of genes in pathway p, such that $g \in G_p$
- S_{ghip} is the gene-, treatment- and sample-specific perturbation score for pathway p
- $\delta^*_{ghi} = w_g \delta_{ghi}$ is the weighted logFC of gene g as described above
- U_{gp} is the subset of G_p containing only the genes directly upstream of gene g
- $\beta_{gg'p}$ is the pair-wise gene-gene interactions[16] encoded by the topology matrix for genes g and g'
- N_{gp} is the number of downstream genes from any gene g
- S_{hip} is the accumulated pathway perturbation score for pathway p in treatment h for sample i

The Bioconductor package graphite[17] provides functions that can be used to retrieve pathway topologies from a database and convert topology information to adjacency matrices. In order to streamline this process we have implemented a convenience function, where users only need to provide the name of the desired database to retrieve all topology information in the format required by the scoring algorithm with the correct type of gene identifiers (ie. Entrez ID).

To scale the single-sample pathway perturbation scores (S_{hip}) so they are comparable across pathways and to test for significance of individual scores, null distributions of perturbation scores for each pathway are generated through a sample permutation strategy, retaining the correct correlation structure between genes within a pathway. With each permutation, column names (i.e. sample labels) for the logCPM matrix are randomly shuffled while the rest of the scoring algorithm remains unchanged. We recommend users to perform a minimum of 1000 permutations, requiring at least 8 unique samples. Subsequently, the median and median absolute deviation (MAD) of the permuted perturbation scores will be calculated and used to normalise the raw perturbation scores to robust Z-scores and obtain associated two-sided p-values. Since the method is single sample-based, the permutation strategy remains applicable regardless of experimental design.

Apart from assessing whether a pathway's activity changed significantly within an individual sample, users may also be interested in detecting changes at the group-level, which can be performed by modelling scores with regression models, incorporating Smyth's moderated t-statistics[18] as implemented in *limma*[19]. The single-sample nature of *sSNAPPY*'s pathway perturbation scores is particularly helpful for datasets with complex experimental designs or known confounding factors as these can also be incorporated into the final regression models.

Operation

The package has been tested on all operating systems, requiring R > 4.2.0, and can be installed using BiocManager as follows.

```
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install("sSNAPPY")
```

Use Cases

Data

To demonstrate the application of *sSNAPPY*, we used pre-processed counts from a publicly available scRNA-seq dataset, retrieved from Gene Expression Omnibus (GEO) with accession code GSE165897. This dataset consists of 11 homogeneously treated high-grade serous ovarian cancer (HGSOC) patients, with samples taken before treatment and after chemotherapy[20]. In the original study, cells were classified into epithelial, stromal, and immune cells, and we have chosen to only focuses on epithelial cells as they were what the original study primarily focused on. Since *sSNAPPY* was designed for bulk RNA-seq data, counts of epithelial cells from the same samples were first summed into pseudo-bulk profiles, giving rise to a total of 22 samples. We considered a gene detectable if we observed >1.5 counts per million in >11 samples out of 22, representing all samples from a complete treatment group. 11,101 (33.8%) of the 32,847 annotated genes passed this

selection criteria and were included for downstream analyses. Conditional quantile normalisation[21] was then applied to mitigate potential biases introduced by gene length and GC content. The normalised logCPM matrix of the processed dataset and sample metadata can be downloaded from here.

The following packages are required for this workflow

```
library(sSNAPPY)
library(tidyverse)
library(magrittr)
library(ggplot2)
library(cowplot)
library(kableExtra)
library(AnnotationHub)
library(edgeR)
```

First we can read in the data, setting the treatment column in the metadata to be a factor.

```
logCPM <- read_tsv(here::here("data/logCPM.tsv")) %>%
    column_to_rownames("entrezid")
sample_meta <- read_tsv(here::here("data/sample_meta.tsv"), col_types = "cfccncnc")
head(sample_meta)
```

```
## # A tibble: 6 x 8
##   sample                 treatment       patie~1 anato~2   Age Stage   PFI CRS
##   <chr>                  <fct>           <chr>   <chr>   <dbl> <chr> <dbl> <chr>
## 1 EOC372_treatment-naive treatment-naive EOC372  Perito~    68 IIIC    460 1
## 2 EOC372_post-NACT       post-NACT       EOC372  Perito~    68 IIIC    460 1
## 3 EOC443_post-NACT       post-NACT       EOC443  Omentum    54 IVA     177 3
## 4 EOC443_treatment-naive treatment-naive EOC443  Omentum    54 IVA     177 3
## 5 EOC540_treatment-naive treatment-naive EOC540  Omentum    62 IIIC    126 2
## 6 EOC540_post-NACT       post-NACT       EOC540  Omentum    62 IIIC    126 2
## # ... with abbreviated variable names 1: patient_id, 2: anatomical_location
```

Note that to apply sSNAPPY, the rownames of the logCPM matrix must be in Entrez IDs.

Retrieval of pathway topology

As important step, pathway topology information needs to be retrieved from a chosen database. Using KEGG as an example, the retrieved topology information will be stored as a list where each element corresponds to a pathway and the numbers in the matrices encode gene-gene interaction.

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

Instead of downloading the topology matrices of all pathways, it is also possible to provide a restricted set of keywords for a targeted analysis. Customised weights could be assigned to different types of gene-gene interaction type by providing a named numeric vector.

```
# Only retrieve the topology matrices of metabolism- or signalling-related pathways
gsTopology_sub <- retrieve_topology(database = "kegg",species = "hsapien",
                                    keyword = c("metabolism", "estrogen"))
names(gsTopology_sub)
```

In addition to focusing on one database, users could provide a vector to the `database` parameter to retrieve topology information from multiple database

```
gsTopology_2databases <- retrieve_topology(database = c("kegg", "reactome"),
                                           species = "hsapien")
```

Score single-sample pathway perturbation

To compute the single-sample expression changes (logFC) needed for perturbation scores, each treated sample must have a matching control sample, with the factor defining the paired structure passed to the `weight_ss_fc()` function through the `groupBy` parameter. In our example dataset, pre- and post-treatment samples are matched by patient IDs. Additionally, the sample metadata must include the treatment of all samples. The treatment column must be a factor with the reference level set to be the control treatment.

```
weightedFC <- weight_ss_fc(as.matrix(logCPM), metadata = sample_meta,
 sampleColumn = "sample", groupBy = "patient_id", treatColumn = "treatment")
glimpse(weightedFC)
```

```
## List of 2
##  $ weight: num [1:10098] 9.38e-05 1.19e-04 7.63e-05 1.18e-04 1.21e-04 ...
##  $ logFC : num [1:10098, 1:11] -1.52e-05 -2.39e-05 2.00e-04 -1.53e-04 3.84e-05 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:10098] "ENTREZID:643837" "ENTREZID:26155" "ENTREZID:84069" "ENTREZID:57801" ...
##   .. ..$ : chr [1:11] "EOC372_post-NACT" "EOC443_post-NACT" "EOC540_post-NACT" "EOC3_post-NACT" ...
```

The output of `weight_ss_fc()` is a `list` where one element is a matrix of weighted single-sample logFCs (δ^*_{ghi}), with rows corresponding to genes and columns to samples, and the other element is the vector of gene-wise weights (w_g) used to calculate the weighted logFC (δ^*_{ghi}), as described above.

The matrix of δ^*_{ghi} values are then passed to pathway topologies to compute the gene-wise perturbation scores for all genes included in a pathway, before being summed into a single score for each pathway. Character string `ENTREZID` was added to all row names of the δ^*_{ghi} matrix to align with the format that pathway topologies were retrieved in. These gene-wise perturbation scores can also be used in downstream analysis to identify genes playing the most significant roles in each pathway, as will be demonstrated in the visualisation section below. The pathway-level perturbation scores (S_{hip}) are returned as a data.frame containing sample and gene-set names. In the following steps, we first calculate the gene-level contributions to each pathway (S_{ghip}) using the function `raw_gene_pert()` and then obtain pathway-level summaries using `pathway_pert()`. Pathways with zero perturbatio scorse across all genes and samples will be dropped at this stage.

```
genePertScore <- raw_gene_pert(weightedFC$logFC, gsTopology)
ssPertScore <- pathway_pert(genePertScore)
head(ssPertScore)
```

```
##             sample        score                                        gs_name
## 1 EOC372_post-NACT  0.005134216 kegg.EGFR tyrosine kinase inhibitor resistance
## 2 EOC443_post-NACT -0.001959631 kegg.EGFR tyrosine kinase inhibitor resistance
## 3 EOC540_post-NACT -0.006543325 kegg.EGFR tyrosine kinase inhibitor resistance
## 4   EOC3_post-NACT -0.003840943 kegg.EGFR tyrosine kinase inhibitor resistance
## 5  EOC87_post-NACT -0.003917201 kegg.EGFR tyrosine kinase inhibitor resistance
## 6 EOC136_post-NACT -0.008308897 kegg.EGFR tyrosine kinase inhibitor resistance
```

Sample permutation for normalisation and significance testing

The values obtained from each pathway will vary greatly due to the variability in topologies. To determine the significance of individual scores and transform scores to ensure they are comparable across pathways, sSNAPPY utilises a sample-permutation strategy to simulate the null distributions of perturbation scores. Since sample labels will be permuted randomly, sample metadata is not required by the generate_permuted _scores function, instead, users only need to specify the number of treatment groups in the study, including the control level. Since permutation requires a large amount of computational time and memory, sSNAPPY utilises the BiocParallel backend[22], also allowing for customisation by the user. Paralleled with 8 cores, permutating 1000 times took approximately 30 minutes to complete on a local laptop. Whilst permutations can be performed an a local machine, this strategy also allows for performing the permutation steps on a HPC cluster or similar.

```
set.seed(123)
permutedScore <- generate_permuted_scores(
    as.matrix(logCPM), numOfTreat = 2, NB = 1000,
    gsTopology = gsTopology, weight = weightedFC$weight
)
```

Apart from pathways whose permuted perturbation scores are all zeros, the rest of the pathways' empirical distributions should be approximately normally distributed with means equal to zero. To demonstrate that, we randomly sampled 6 pathways and visualised the permuted perturbation scores as boxplots (Figure 1).

```
set.seed(234)
permutedScore %>%
    keep(~all(.!=0)) %>%
    .[sample(seq_along(.), 6)] %>%
    as.data.frame() %>%
    pivot_longer(
        cols = everything(),
        names_to = "gs_name",
        values_to = "score"
    ) %>%
    mutate(
        gs_name = str_replace_all(gs_name, "\\.", " "),
        gs_name = str_remove_all(gs_name, "kegg ")
    ) %>%
    ggplot(
        aes(gs_name, score, fill = gs_name)
    ) +
    geom_boxplot() +
    scale_x_discrete(labels = function(x) str_wrap(x, width = 10)) +
    scale_fill_discrete(name = "Gene-set Name") +
    labs(
        x = "Gene-set Name",
        y = "Permuted Perturbation Score") +
    theme_bw(
    )
```

[Figure 1 about here.]

Permuted distributions are then used to convert each pathway-level score into a scaled robust Z-score using the function normalise_by_permu. Robust Z-scores can in turn be transformed into two-sided p-values and corrected for multiple testing using any of the available methods, and returning the FDR adjusted values by default. Users can choose to sort the output by p-values, gene-set names or sample names through the `sortBy` parameter. In our example data, no pathways would be considered as significantly perturbed at the single-sample level using an FDR adjustment with $\alpha = 0.05$.

```
normalisedScores <- normalise_by_permu(permutedScore, ssPertScore, sortBy = "pvalue")
head(normalisedScores)
```

```
##               MAD        MEDIAN                                    gs_name
## 388  0.0275353965  1.538283e-04            kegg.Epstein-Barr virus infection
## 417  0.0002129193  7.921798e-07              kegg.Autoimmune thyroid disease
## 421  0.0002129193  7.921798e-07                     kegg.Allograft rejection
## 1142 0.0034661258 -1.673836e-06 kegg.C-type lectin receptor signaling pathway
## 505  0.0098582420 -1.139206e-04               kegg.Th17 cell differentiation
## 1530 0.0036864141  2.507001e-05                      kegg.Autophagy - animal
##                sample         score   robustZ      pvalue adjPvalue
## 388  EOC136_post-NACT -0.0817379496 -2.974055 0.002938926 0.2392677
## 417  EOC136_post-NACT -0.0006237555 -2.933261 0.003354221 0.2392677
## 421  EOC136_post-NACT -0.0006237555 -2.933261 0.003354221 0.2392677
## 1142 EOC349_post-NACT  0.0085037555  2.453872 0.014132724 0.9803990
## 505  EOC153_post-NACT  0.0238610587  2.431973 0.015016823 0.7618825
## 1530 EOC443_post-NACT  0.0087762111  2.373890 0.017601798 1.0000000
```

A key question of interest in this dataset is to identify which biological processes were impacted by chemotherapy across the entire group of patients. Using the sample-level output obtained above, we can explore this by applying t-tests or regression models across all samples. In order to minimise spurious results, Smyth's moderated t-statistics[18] are able to be applied across the complete dataset, however given that we have robust Z-score, a constant variance should be assumed across all pathways. To perform this analysis, we convert the robust Z-scores to a matrix then use the standard limma methodologies setting variance to be constant as values have been normalised using robust Z scores. For our use case here, no design matrix is required and a simple t-test is appropriate.

```
z_matrix <- normalisedScores %>%
  dplyr::select(robustZ, gs_name, sample) %>%
  pivot_wider(names_from = "sample", values_from = "robustZ") %>%
  column_to_rownames("gs_name") %>%
  as.matrix()
z_fits <- lmFit(z_matrix, design = rep(1, ncol(z_matrix))) %>%
  eBayes(trend = FALSE)
```

Pathways with an FDR < 0.05 in the moderated t-test were considered to be significantly perturbed at the group level. 22 out of the 315 tested KEGG pathways passed the selection threshold (Table 1).

```
table1 <- topTable(z_fits, number = Inf) %>%
  rownames_to_column("gs_name") %>%
  dplyr::filter(adj.P.Val < 0.05) %>%
  mutate(
    Direction = ifelse(logFC < 0, "Inhibited", "Activated"),
    gs_name = str_remove_all(gs_name, "kegg.")
  ) %>%
  dplyr::select(
    Pathway = gs_name, Change = logFC,  P.Value, FDR = adj.P.Val, Direction
  )
```

[Table 1 about here.]

For enrichment analysis in the original study[20], unsupervised clustering was performed on all cells labelled to be cancer cells. Identified cancer clusters were labelled by performing pathway enrichment testing on cluster marker genes. Two clusters, associated with proliferative DNA repair signatures and stress-related markers, were found to contain significantly higher numbers of post-chemotherapy cells then pre-treatment ones (Ta in [20]). The representative pathways enriched in the stress-associated cluster were *IL6-mediated signaling events*, *TNF signaling pathway*, and *cellular responses to stress*, characterized by marker genes JUN, FOS, IL6, TNF, CXCR4, SNAI1, VIM, GADD45B, and MCL1. Among the stress-related marker genes reported, 6 of them(CXCR4, FOS, GADD45B, IL6, JUN, and TNF) were implicated in pathways that were considered to be significantly impacted by sSNAPPY. The other post-chemotherapy cell dominated cluster in the original study was enriched for pathways associated with cell proliferation and DNA repair, such as the Cell cycle, DNA repair, Homology directed repair (HDR) through homologous recombination, and Fanconi anaemia pathway. A key gene involved in those pathways was CHEK1, which was also found in significantly perturbed pathways detected by sSNAPPY: the p53 signaling pathway, Cellular senescence and the Human T-cell leukaemia virus 1 infection pathway.

Apart from treating all treated samples as biological replicates, users may wish to perform an analysis incorporating other phenotypic traits which may affect patients' responses to chemotherapy, such as the stages of their cancers. To do that through the moderated t-statistic strategy and extend the above analysis, we simply need to provide an appropriate design matrix in the `lmFit` step, or subset our samples accordingly

Visualise perturbed pathways as networks

In addition to performing the analysis, sSNAPPY provides various visualisation functions to assist in the interpretation of results. Biological pathways are not independent of each other with many genes playing a role across multiple pathways, and as such, visualising pathway analysis results as a network can be a powerful way to intuitively summarise the results, and also to facilitate interpretation of the underlying biology. The `plot_gs_network()` function allows users to easily visualise a set of pathways as a network where edges connect genes to pathway nodes, easily capturing information across multiple pathways. Pathway nodes can be coloured by any of the returned values, such as the magnitude of perturbation (logFC), p-value or additional classifications performed manually (Figure 1). As the returned plot is a ggplot2 [23] object, colour schemes and plotting themes can be easily customised as desired.

In order to First we should obtain a subset of pathways to visualise. In the following we'll inspect the most highly ranked pathways and then for visualisation, 1) create a categorical variable with the pathway status, 2) rename the logFC column to reflect the true meaning of the value and, 3) transform the p-values for simpler visualisation

```
# Extract significantly perturbed pathway. To color the pathways
# by perturbation status, add a column to the data.frame
set.seed(123)
sigPathway <- topTable(z_fits, number = Inf) %>%
    dplyr::filter(adj.P.Val < 0.05) %>%
    rownames_to_column("gs_name") %>%
    mutate(Status = ifelse(AveExpr < 0, "Inhibited", "Activated"))
p1 <- sSNAPPY::plot_gs_network(
    normalisedScores = sigPathway ,
    gsTopology = gsTopology,
    colorBy = "Status"
) + theme_void()
p2 <- sSNAPPY::plot_gs_network(
    normalisedScores = sigPathway,
    gsTopology = gsTopology,
    colorBy  = "P.Value"
) + theme_void()
plot_grid(
    p1, p2,
    labels = c("A", "B")
)
```

[Figure 2 about here.]

By examining the network structure, we can see that many of the highly connected pathways playing a central role in the network may be immune-related. To summarise related pathways and further enable interpretation, we use community detection to group related pathways Widely used in network analysis, community detection is a technique for identifying groups of nodes that are more densely connected than to any other nodes in the network[24]. sSNAPPY's `plot_community()` function is a one-stop shop for applying a community detection algorithm of the user's choice to the network structure and annotating identified communities by the most common pathway category, denoting the main biological processes perturbed in that community. Retrieved directly from the KEGG website, we have curated the most recent categories for KEGG pathways and included this as part of sSNAPPY. Annotation of KEGG pathway communities will be automatically completed by calling the in-built data object. However, analyses involving other pathway databases will require user-provided pathway categories. In the current dataset, the Louvain method was applied to the network of biological pathways and revealed two primary communities, where one was annotated to be endocrine system related and the other one was clearly related to infectious diseases and the immune system (Figure 2).

```
set.seed(123)
sSNAPPY::plot_community(
  normalisedScores = sigPathway,
  gsTopology = gsTopology,
  colorBy = "Status"
) +
  scale_fill_ordinal() +
  scale_x_continuous(expand = expansion(0.3)) +
  scale_y_continuous(expand = expansion(0.3)) +
  theme_void()
```

[Figure 3 about here.]

Inferred directly from the expression matrix, a key advantage of sSNAPPY is that it does not require the prior identification of differentially expressed genes, which is not always possible in clinical datasets. However, knowing which genes are implicated in the perturbation of pathways, particularly those which influence multiple pathways, can provide valuable insights for hypotheses generation about underlying biological mechanisms. Therefore, sSNAPPY presents another visualisation feature called `plot_gs2gene`, which enables the inclusion of select genes from each pathway using network structures. Users can provide a vector of fold-change estimates to visualise genes within pathways, showing their estimated change in expression. As pathways often include hundreds of genes, we recommend to filter for genes most likely to be playing a significant role. In this example dataset, we chose to only include genes within the top 500 when ranking by the magnitude of the mean log fold-change (Figure 3).

```
meanFC <- rowMeans(weightedFC$logFC) / weightedFC$weight
top500 <- rank(1/abs(meanFC)) <= 500
dirFC <- ifelse(meanFC > 0, "Up-Regulated", "Down-Regulated")
```

Since pathway topologies were retrieved in Entrez IDs, by default, genes' Entrez IDs will be used to annotate gene nodes in the plot. However, users can provide a data.frame mapping Entrez IDs to their chosen identifiers through the mapRownameTo parameter. A data.frame converting Entrez IDs to ensemble gene names has been made available as part of the package.

```
# Read in built-in data.frame entrez2name that matches genes'
# Entrez IDs to gene names
load(system.file("extdata", "entrez2name.rda", package = "sSNAPPY"))
head(entrez2name)
```

```
## # A tibble: 6 x 2
##   entrezid           mapTo
##   <chr>              <chr>
## 1 ENTREZID:84771     DDX11L1
## 2 ENTREZID:727856    DDX11L1/DDX11L9/DDX11L10
## 3 ENTREZID:100287102 DDX11L1
## 4 ENTREZID:100287596 DDX11L1/DDX11L9
## 5 ENTREZID:102725121 DDX11L1
## 6 ENTREZID:653635    WASH7P
```

```
entrez2name %>%
    dplyr::filter(str_detect(mapTo, "HLA-DQA2"))
```

```
## # A tibble: 2 x 2
##   entrezid      mapTo
##   <chr>         <chr>
## 1 ENTREZID:3117 HLA-DQA1/HLA-DQA2
## 2 ENTREZID:3118 HLA-DQA2
```

```
# str_subset(names(dirFC), "ENTREZID:3117|ENTREZID:3118")
```

```
# Plot the pathway-gene network for genes with top 500
#magnitudes of mean FCs and label gene nodes by gene names
sSNAPPY::plot_gs2gene(
  normalisedScores = sigPathway ,
  gsTopology = gsTopology,
  colorGsBy = "Status",
  mapEntrezID = entrez2name,
  geneFC = dirFC,
  filterGeneBy = 0,
  GsName_size = 4,
  geneNameSize = 5,
  gsNodeSize = 4
) +
  theme(
    panel.background = element_blank()
  ) +
  scale_fill_manual(values = c("darkred", "lightskyblue"), name = "Pathway") +
  scale_colour_manual(values = c("blue", "red"), name = "Gene\nDirection")
```

```
## Warning: ggrepel: 2 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

[Figure 4 about here.]

Identify hub genes contributing to pathway perturbation

If we would like to further investigate a specific pathway and elucidate the key genes that contributed to its perturbation, such as the activation of the "p53 signalling pathway," we can employ a heatmap to display the gene-level perturbation scores of all the genes within the pathway and annotate each column (ie. each sample) by the direction of pathway perturbation in that sample or any other sample metadata using the plot_gene_contribution function (Figure 5).

```
annotation_df <- ssPertScore %>%
    dplyr::filter(gs_name == "kegg.p53 signaling pathway") %>%
    mutate(
        `pathway-level` = ifelse(score > 0, "Activated", "Inhibited")
    ) %>%
    left_join(
        sample_meta %>%
            dplyr::select(sample, Stage)
    ) %>%
    dplyr::select(sample, `pathway-level`, Stage)
```

```
## Joining with 'by = join_by(sample)'
```

```
plot_gene_contribution(
    genePertMatr = genePertScore$`kegg.p53 signaling pathway`,
    annotation_df =  annotation_df,
    topGene = 20, filterBy = "mean",
    mapEntrezID = entrez2name,
    annotation_colors = list(
        `pathway-level` = c(Activated = "darkred", Inhibited = "lightskyblue"))
)
```

[Figure 5 about here.]

From this heatmap we can easily identify that the hub genes making the biggest contribution to the activation of p53 signalling pathway upon chemotherapy were gene ART and gene ATM. The Ataxia-telangiectasia mutated (ATM) gene is a well-established oncosuppressor[25], mutation of which has been observed in many types of cancers[26]. Also involved in DNA damage repair, the Ataxia telangiectasia and RAD3-related protein kinase (ATR) gene has been shown to be a promising therapeutic target for HGSOC[27].

Discussion

In conclusion, the paper showcased an R/Bioconductor package that offers a novel single-sample pathway perturbation testing approach. sSNAPPY utilizes pathway topology information to compute perturbation scores that predict pathways' potential directions of changes in individual samples. This approach addresses the limitations of current strategies that fail to account for gene-gene interactions encoded by pathway topologies or predict the directionality of pathway activities. By applying sSNAPPY to a public scRNA-seq data collected before and after HGSOC patients were subjected to chemotherapy, we demonstrated its ability to detect significant pathway perturbations of various interesting biological processes beyond what were shown in the original studies. Overall, sSNAPPY presents a promising strategy for single sample-based pathway analysis in RNA-seq data.

Data availability

The dataset analysed in this manuscript are stored in the data directory of this GitHub repository.

Software availability

- Software available from: `https://bioconductor.org/packages/release/bioc/html/sSNAPPY.html`
- Source code available from: `https://github.com/Wenjun-Liu/sSNAPPY`
- Archived source code at time of publication: [DOI (found on right hand side of a Zenodo record)]
- License: MIT

Competing interests

No competing interests were disclosed

Grant information

Any grants that supported the work must be listed here, including the grant number.

Acknowledgements

[1] Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, and Jill P Mesirov. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. U. S. A.*, 102 (43):15545–15550, October 2005.

[2] Matthew D Young, Matthew J Wakefield, Gordon K Smyth, and Alicia Oshlack. Gene ontology analysis for RNA-seq: accounting for selection bias. *Genome Biol.*, 11(2):R14, February 2010.

[3] Farhad Maleki, Katie Ovens, Daniel J Hogan, and Anthony J Kusalik. Gene set analysis: Challenges, opportunities, and future research. *Front. Genet.*, 11:654, June 2020.

[4] Sarah Mubeen, Alpha Tom Kodamullil, Martin Hofmann-Apitius, and Daniel Domingo-Fernández. On the influence of several factors on pathway enrichment analysis. *Brief. Bioinform.*, 23(3), May 2022.

[5] Hiroyuki Ogata, Susumu Goto, Kazushige Sato, Wataru Fujibuchi, Hidemasa Bono, and Minoru Kanehisa. KEGG: Kyoto Encyclopedia of Genes and Genomes.

[6] Marvin Martens, Ammar Ammar, Anders Riutta, Andra Waagmeester, Denise N Slenter, Kristina Hanspers, Ryan A. Miller, Daniela Digles, Elisson N Lopes, Friederike Ehrhart, Lauren J Dupuis, Laurent A Winckers, Susan L Coort, Egon L Willighagen, Chris T Evelo, Alexander R Pico, and Martina Kutmon. WikiPathways: connecting communities. *Nucleic Acids Research*, 49(D1):D613–D621, January 2021. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkaa1024.

[7] Adi Laurentiu Tarca, Sorin Draghici, Purvesh Khatri, Sonia S Hassan, Pooja Mittal, Jung-Sun Kim, Chong Jai Kim, Juan Pedro Kusanovic, and Roberto Romero. A novel signaling pathway impact analysis. *Bioinformatics*, 25(1):75–82, January 2009.

[8] Laurent Jacob, Pierre Neuvial, and Sandrine Dudoit. More power via graph-structured tests for differential expression of gene networks. *The Annals of Applied Statistics*, 6(2):561 – 600, 2012. doi: 10.1214/11-AOAS528. URL `https://doi.org/10.1214/11-AOAS528`.

[9] Jing Ma, Ali Shojaie, and George Michailidis. Network-based pathway enrichment analysis with incomplete network information. *Bioinformatics*, 32(20):3165–3174, 06 2016. ISSN 1367-4803. doi: 10.1093/bioinformatics/btw410. URL `https://doi.org/10.1093/bioinformatics/btw410`.

[10] Maysson Al-Haj Ibrahim, Sabah Jassim, Michael Anthony Cawthorne, and Kenneth Langlands. A topology-based score for pathway enrichment. *Journal of Computational Biology*, 19(5):563–573, 2012. doi: 10.1089/cmb.2011.0182. URL `https://doi.org/10.1089/cmb.2011.0182`. PMID: 22468678.

[11] Di Wu and Gordon K. Smyth. Camera: a competitive gene set test accounting for inter-gene correlation. *Nucleic Acids Research*, 40(17):e133–e133, 05 2012. ISSN 0305-1048. doi: 10.1093/nar/gks461. URL `https://doi.org/10.1093/nar/gks461`.

[12] Tuan-Minh Nguyen, Adib Shafi, Tin Nguyen, and Sorin Draghici. Identifying significantly impacted pathways: a comprehensive review and assessment. *Genome Biol.*, 20(1):203, October 2019.

[13] Jing Ma, Ali Shojaie, and George Michailidis. A comparative study of topology-based pathway enrichment analysis methods. *BMC Bioinformatics*, 20(1):546, December 2019. ISSN 1471-2105. doi: 10.1186/s12859-019-3146-1.

[14] Sonja Hänzelmann, Robert Castelo, and Justin Guinney. GSVA: gene set variation analysis for microarray and RNA-Seq data. *BMC Bioinformatics*, 14(1):7, December 2013. ISSN 1471-2105. doi: 10.1186/1471-2105-14-7.

[15] Charity W Law, Yunshun Chen, Wei Shi, and Gordon K Smyth. voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biol*, 15(2):R29, 2014. ISSN 1465-6906. doi: 10.1186/gb-2014-15-2-r29.

[16] Adi Laurentiu Tarca, Sorin Draghici, Purvesh Khatri, Sonia S. Hassan, Pooja Mittal, Jung-sun Kim, Chong Jai Kim, Juan Pedro Kusanovic, and Roberto Romero. A novel signaling pathway impact analysis. *Bioinformatics*, 25(1):75–82, January 2009. ISSN 1367-4811, 1367-4803. doi: 10.1093/bioinformatics/btn577.

[17] Gabriele Sales, Enrica Calura, Duccio Cavalieri, and Chiara Romualdi. graphite - a Bioconductor package to convert pathway topology to gene network. *BMC Bioinformatics*, 13(1):20, December 2012. ISSN 1471-2105. doi: 10.1186/1471-2105-13-20.

[18] Gordon K Smyth. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3(1), 2004. doi: doi:10.2202/1544-6115.1027. URL `https://doi.org/10.2202/1544-6115.1027`.

[19] Matthew E Ritchie, Belinda Phipson, Di Wu, Yifang Hu, Charity W Law, Wei Shi, and Gordon K Smyth. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research*, 43(7):e47, 2015. doi: 10.1093/nar/gkv007.

[20] Kaiyang Zhang, Erdogan Pekcan Erkan, Sanaz Jamalzadeh, Jun Dai, Noora Andersson, Katja Kaipio, Tarja Lamminen, Naziha Mansuri, Kaisa Huhtinen, Olli Carpén, Sakari Hietanen, Jaana Oikkonen, Johanna Hynninen, Anni Virtanen, Antti Häkkinen, Sampsa Hautaniemi, and Anna Vähärautio. Longitudinal single-cell RNA-seq analysis reveals stress-promoted chemoresistance in metastatic ovarian cancer. *Sci. Adv.*, 8(8):eabm1831, February 2022. ISSN 2375-2548. doi: 10.1126/sciadv.abm1831.

[21] K. D. Hansen, R. A. Irizarry, and Z. Wu. Removing technical variability in RNA-seq data using conditional quantile normalization. *Biostatistics*, 13(2):204–216, April 2012. ISSN 1465-4644, 1468-4357. doi: 10.1093/biostatistics/kxr054.

[22] Martin Morgan, Jiefei Wang, Valerie Obenchain, Michel Lang, Ryan Thompson, and Nitesh Turaga. *BiocParallel: Bioconductor facilities for parallel evaluation*, 2022. URL `https://github.com/Bioconductor/BiocParallel`. R package version 1.32.5.

[23] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer New York, New York, NY, 2009. ISBN 978-0-387-98140-6. doi: 10.1007/978-0-387-98141-3. URL `https://link.springer.com/10.1007/978-0-387-98141-3`.

[24] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(2):026113, February 2004. ISSN 1539-3755, 1550-2376. doi: 10.1103/PhysRevE.69.026113.

[25] Masoumeh Moslemi, Yousef Moradi, Hojat Dehghanbanadaki, Hamed Afkhami, Mansoor Khaledi, Najmeh Sedighimehr, Javad Fathi, and Ehsan Sohrabi. The association between ATM variants and risk of breast cancer: a systematic review and meta-analysis. *BMC Cancer*, 21(1):27, December 2021. ISSN 1471-2407. doi: 10.1186/s12885-020-07749-6.

[26] Michael Choi, Thomas Kipps, and Razelle Kurzrock. ATM Mutations in Cancer: Therapeutic Implications. *Molecular Cancer Therapeutics*, 15(8):1781–1791, August 2016. ISSN 1535-7163, 1538-8514. doi: 10.1158/1535-7163.MCT-15-0945.

[27] Siyu Li, Tao Wang, Xichang Fei, and Mingjun Zhang. ATR Inhibitors in Platinum-Resistant Ovarian Cancer. *Cancers*, 14(23):5902, November 2022. ISSN 2072-6694. doi: 10.3390/cancers14235902.

List of Figures

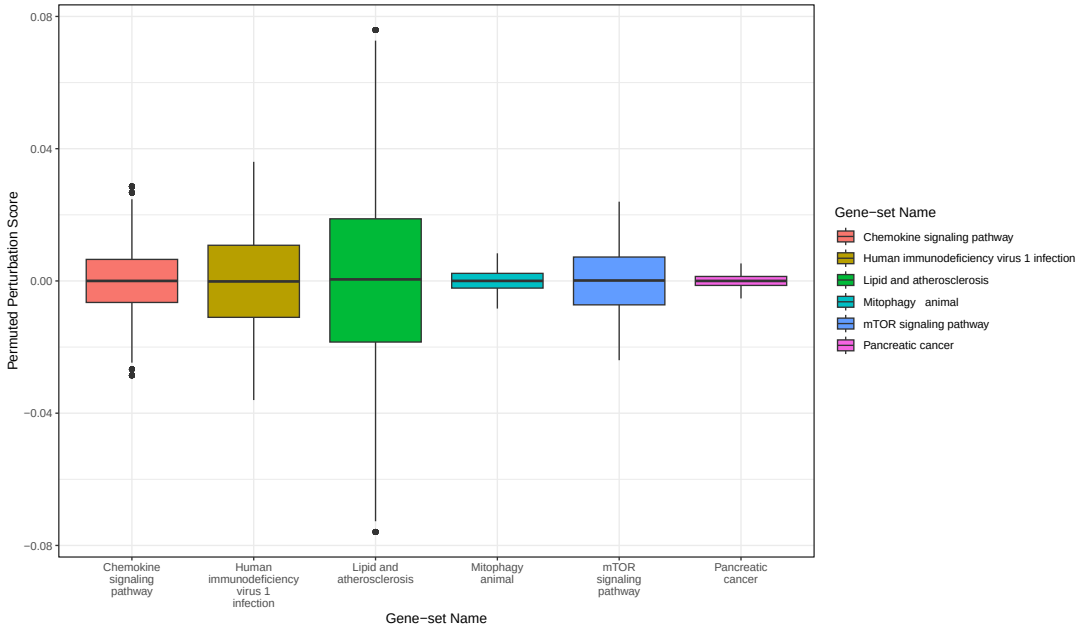

Figure 1. Permuted perturbation scores of six randomly selected pathways. All sampled empirical distributions are approximately normally distributed with a mean of zero.

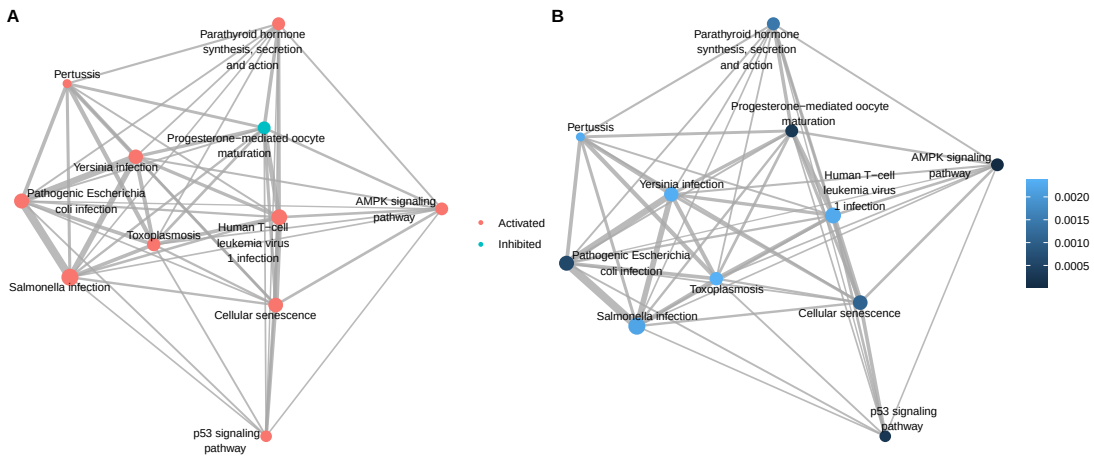

Figure 2. Significantly perturbed KEGG pathways identified among post-chemotherapy samples using sSNAPPY, colored by (A) pathways' predicted directions of changes and (B) pathways' -log10(p-values). Pathways with a FDR < 0.05 in the moderated t-test were included.

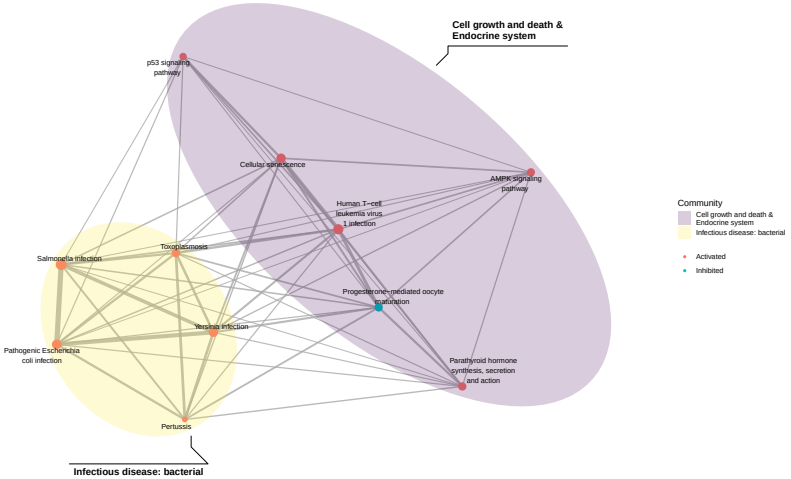

Figure 3. Significantly perturbed KEGG pathways identified among post-chemotherapy samples using sSNAPPY, colored by community structures detected through the louvain algorithm. The two main biological processes perturbed by the chemo-therapy were endocrine- and immune-related.

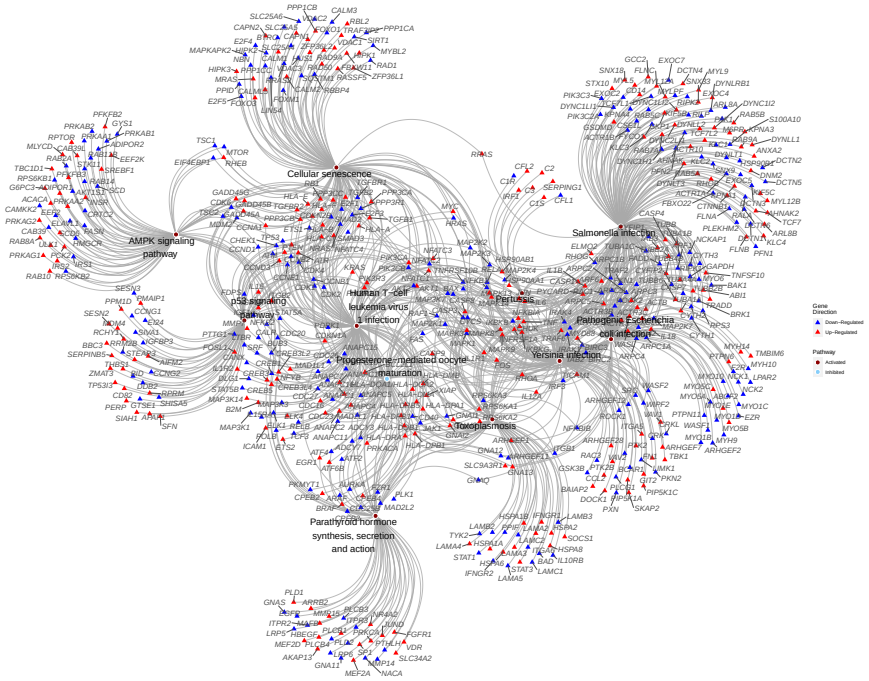

Figure 4. Significantly perturbed KEGG pathways identified among post-chemotherapy samples using sSNAPPY and asssiociated pathway gens with top 500 magnitudes of fold-changes. Both pathways and genes were colored by their directions of changes. Only genes involced in at least 2 pathways were included.

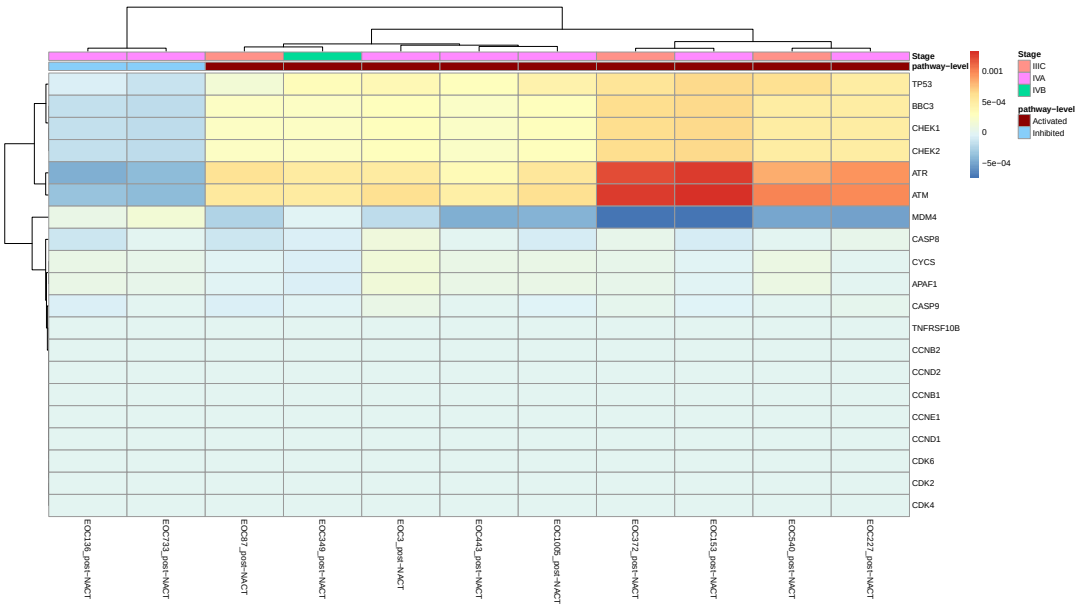

Figure 5. Gene-level perturbation scores of all genes included in the "p53 signalling pathway" in each sample where columns of samples were annotated by pathway-level perturbation and the stages of cancers. Genes ATR and ATM were identified to be the key driverd of the activation of p53 signalling pathway.

List of Tables

Table 1. Significantly impacted KEGG pathways identified among post-chemotherapy samples using sSNAPPY

Pathway	Change	P.Value	FDR	Direction
AMPK signaling pathway	0.88	1.2e-05	0.0026	Activated
p53 signaling pathway	0.88	2.4e-04	0.0190	Activated
Progesterone-mediated oocyte maturation	-0.77	2.7e-04	0.0190	Inhibited
Pathogenic Escherichia coli infection	0.66	5.7e-04	0.0303	Activated
Cellular senescence	0.68	1.1e-03	0.0464	Activated
Parathyroid hormone synthesis, secretion and action	0.63	1.4e-03	0.0464	Activated
Salmonella infection	0.59	2.2e-03	0.0464	Activated
Human T-cell leukemia virus 1 infection	0.62	2.3e-03	0.0464	Activated
Pertussis	0.59	2.3e-03	0.0464	Activated
Yersinia infection	0.62	2.3e-03	0.0464	Activated
Toxoplasmosis	0.63	2.4e-03	0.0464	Activated