Google Summer of Code 2023

National Resource for Network Biology

Supporting Pathway Commons and ssGSEA in clusterProfiler

(Potential Mentors : Augustin Luna and Guangchuang Yu)

Table of Contents :

1. Personal Background :

- *Name* : **JIGYASA GUPTA**
- *GitHub username* : Jigyasa-G (https://github.com/Jigyasa-G)
- *Email* : jigyasatata@gmail.com
- *Phone* : +91 7004580734
- *LinkedIn* : https://www.linkedin.com/in/jigyasag1121/
- *Location* :
  - Country : **India** (GMT+5:30)
  - City : Bangalore
  - College : Dayananda Sagar College of Engineering
- *Background* : **Computer Science Engineering** (3rd Year) –

  Bachelors of Engineering(B.E. Hons)
- Relevant work experience :
  - R programming for Analyzing Covid-19 Data
  - NPTEL course on NOC: BioInformatics : Algorithms and Applications

2. Relevant Skills :

- What are your programming languages of choice and how do they relate to the project?
  - Languages : **Python , R , C++**
  - I have worked with data ,its cleaning and preprocessing in several projects using Python. I have used R for analyzing the covid-19 data using **Rstudio** and **R shell** to obtain meaningful results. R language will be used for scripting various functions required to achieve project goals and use existing packages and libraries.

- Any prior **experience** with **open source** development?

  - **Hacktoberfest ’2021** :Completed (4+) Pull Requests across various repositories.(1,2,3,4)
  - **Kharagpur Winter of Code ’2021** – Contributed to multiple repositories hosted in the prestigious program carried out by IIT Kharagur.(Link)
  - **GoalScore Project** – Committed significant backend code for the website(Link)

- What do you want to learn or accomplish this summer?

  Being a keen learner of biology and a programming enthusiast I want to bring together the possibility of both this summer. Data is the new oil, but analyzing it puts the oil in the latch for smooth functioning. I intend to **develop** better understanding of the **tools and packages** (clusterProfiler, GSVA, Gene Pattern) and even **code open-source** tools or utilities leveraging existing ones to integrate bioinformatics pipelines and data analysis techniques for studying gene expression data and help in **advancing medical research**.

- Any prior exposure to biology or bioinformatics?

  - Ongoing course on NPTEL Swayam by Govt. of India :
    - NOC: **BioInformatics : Algorithms and Applications**
  - Project on **Analysis of Covid-19 data** using R (Link).
  - Research project on **AI-guided echocardiography** which made me push my limits to learn the biology behind echocardiography and the functioning of the heart.
  - Coursework at School to learn **foundations of biology** - prescribed by NCERT(Grade 12)

3. Project Proposal :

3.1. Original project idea & Mentors :

- Idea : **Supporting ssGSEA and Pathway Commons in clusterProfiler #222**(https://github.com/nrnb/GoogleSummerOfCode/issues/222)

- **Potential Mentors :**
  - Augustin Luna (https://github.com/cannin)
  - Guangchuang Yu (https://github.com/GuangchuangYu)
- I have been in contact with Mr. Augustin Luna on Github and have discussed a couple of doubts regarding drafting of this proposal on single or multiple approaches.

3.2. Project overview :

clusterProfiler provides an impressive universal interface for gene functional annotation and can access data to generate **enrichment** results and provide effective **data interpretation**. Pathway Commons is an aggregated database of molecular interaction pathways , collected from approximately 20 databases.

At present, there exists no direct way to analyze data from **Pathway Commons** leveraging all the functionality of clusterProfiler. The data will be **fetched** just by specifying the **URL** of the database with the help of R scripts. It is proposed to enable support for Pathway Commons database in clusterProfiler to yield advanced and more accurate enrichment results and visualizations of the data.

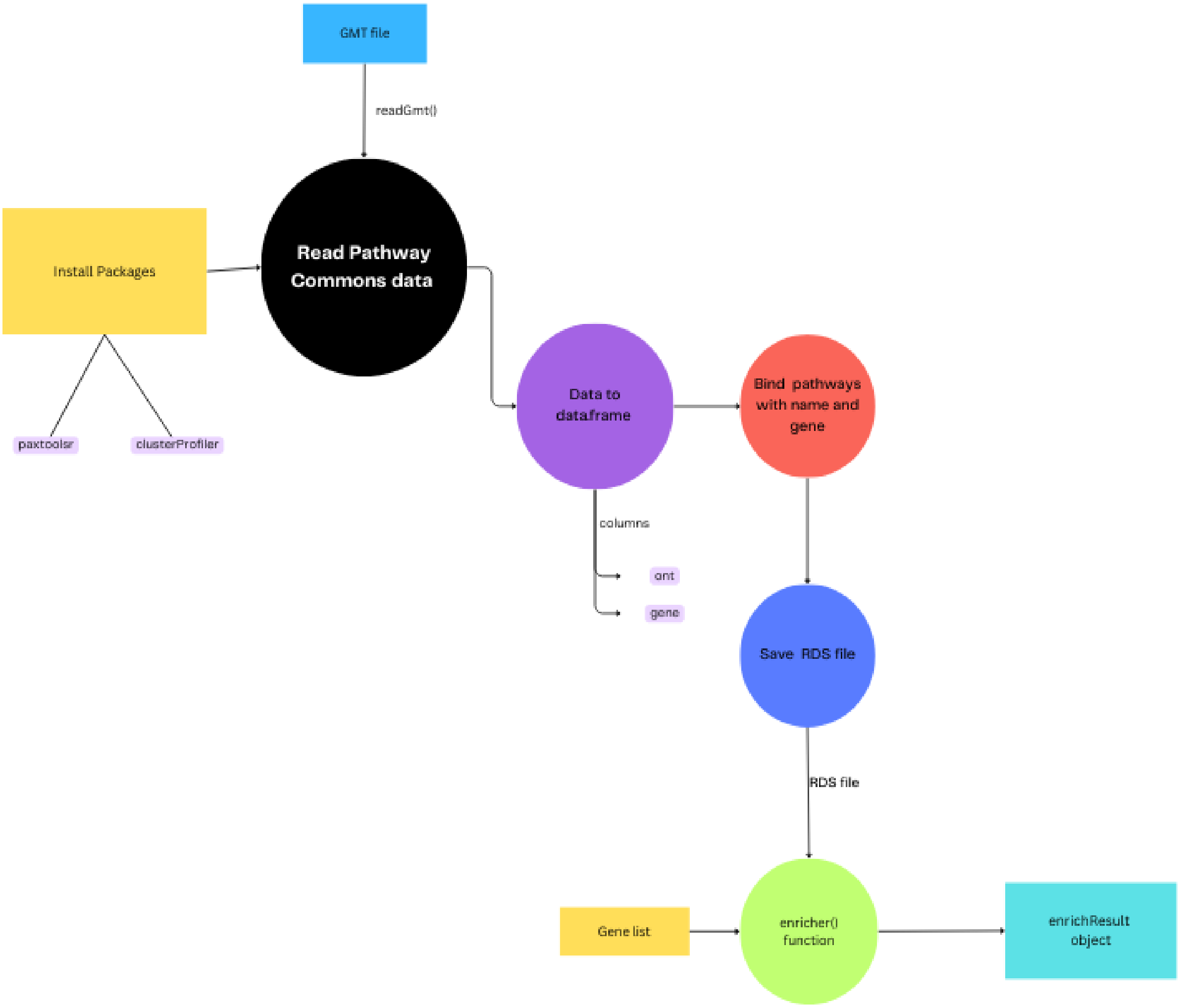
Single sample gene set enrichment analysis is an effective method of data interpretation that uses **permutations of gene sets** to calculate enrichment scores. It has use cases like detecting outliers or tumors in the medical context. This is a variation of GSEA, however, clusterprofiler currently supports only GSEA. To integrate the functionality of ssGSEA, we can use GSVA package with GenePatterns to analyze gene sets in various permutations and get more indicative **enrichment scores** using clusterProfiler.

3.3. Project Goals and details :

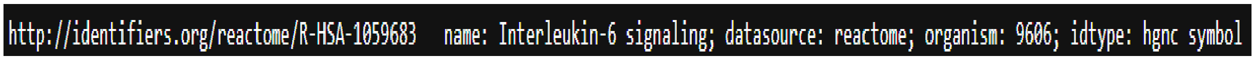
- Supporting Pathway Commons in clusterProfiler
- Supporting ssGSEA in clusterProfiler

3.3.1. Supporting Pathway commons in clusterProfiler :

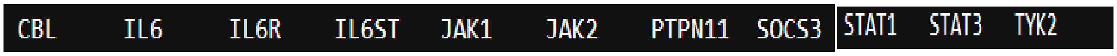
To integrate a database has a purpose to analyze the data, lets perform an **Over-representation analysis**(ORA) of the Pathway Commons data set :
(source : https://www.pathwaycommons.org/archives/PC2/v12/)



1. **Analysis of data** : PathwayCommons12.reactome.hgnc.gmt



```
http://identifiers.org/reactome/R-HSA-1059683   name: Interleukin-6 signaling; datasource: reactome; organism: 9606; idtype: hgnc symbol
```



```
CBL    IL6    IL6R    IL6ST    JAK1    JAK2    PTPN11   SOCS3   STAT1   STAT3   TYK2
```

Taking an example of the Reactome database of Pathways which is represented using identifiers.org format.

- The pathway is named as “**Interleukin-6 signaling**” and the data source is “Reactome”.
- Since, HUGO Gene Nomenclature Committee(HGNC) approves a gene name and its corresponding symbol for every known human gene, the organism here is *Homo sapiens*(human) with NCBI Taxonomy ID of 9606.
- The various genes involved in the signaling of interleukin-6 are identified by their HGNC symbol. They are : CBL, IL6, IL6R, IL6ST, JAK1, JAK2, PTPN11, SOCS3, STAT1, STAT3 and TYK2.
    - CBL, JAK1, JAK2, PTPN11, STAT1, STAT3 and TYK2 encode enzymes or transcription factors involved in the signaling pathway.
    - IL6, IL6R and IL6ST encode proteins that are involved in the binding of interleukin-6 to its receptor and further signaling.
    - SOC3 encodes a protein that is involved in the negative regulation of the interleukin-6 signaling pathway.

2. To better examine the data, we use the **readGmt(data)** function from paxtoolsR package to load the GMT format of Pathway Commons data.

```
pc <- readGmt("PathwayCommons.12.Reactome.GSEA.hgnc.gmt", removePrefix = TRUE)
```

3. Conversion from **GMT format to data frame** is required as the input to the enricher() function should be a data.frame comprising of two columns : “**ont**” and “**gene**”. We use a loop to iterate over each pathway in the `pc` object and use the ‘**rbind()**’ function to bind them together.

```
pcGmt <- data.frame(ont=character(0), gene=character(0))
for(i in 1:length(names(pc)))
{
        x <- names(pc)[i]
        pcGmt <- rbind(pcGmt, data.frame(ont=x, gene=pc[[x]]))
}
```

4. We save the resulting ‘pcGmt’ object as an RDS file to be loaded later and fed to the enricher() function**. RDS**(R Data Serialization) is a format to save R objects.

```
pcGmt <- saveRDS(pcGmt, "pcGmt.rds")
```

5. Finally we come to the part where ORA(Over-Representation Analysis) is performed. To specify the gene list on which to perform ORA, we load a **vector** of **Entrez gene IDs**.

For gene symbols : *CBL, JAK1, JAK2, PTPN11, STAT1, STAT3, TYK2*

```
genes <- c(867, 3716, 3717, 5781, 6772,  6774, 7297)
```

6. We use **enricher()** function from clusterProfiler. It takes two parameters : *genes* (vector of genes specified int the previous step) and *TERM2GENE* and returns an *enrichResult* instance which contains information about the enriched pathways and their statistical significance(like p-value).

```
egmt <- enricher(genes, TERM2GENE=pcGmt)
```

Similarly, we use a ranked gene list for **GSEA(Gene Set Enrichment Analysis)** :

- Load the gene list and use ‘AnnotationDbi’ package to map gene symbols to Entrez gene IDs.

```
genes <- c("CBL", "JAK1", "JAK2", "PTPN11", "STAT1", "STAT3", "TYK2")
ids <- mapIds(org.Hs.eg.db, keys=genes, keytype=”SYMBOL”, column=”ENTREZID”)
symbols <- ids$SYMBOL
```

- *Load Pathway Commons GMT data as in ORA*
- *Perform **GSEA analysis**, here, we calculate gene set enrichment score 1000 times for a randomly permuted gene list using the gsea(). Alternatively, we can use the fgsea() function also.*

```
res <- GSEA(symbols, TERM2GENE = pcGmt, nperm = 1000)
head(res)
```

7. I propose the following set of functions to be used to support Pathway Commons data are :

- **enrichPC** : To support Over-representation analysis(ORA). It takes arguments of gene (vector of entrez gene id) and also any additional parameters. After using the enricher() function, it returns an enrichResult instance.
- **gsePC** : Performs gene set enrichment analysis using GSEA() function. Involves input as ranked gene list, and any other parameters. It returns gseaResult instance as the result.
- **prepare_PC_data** : Used to prepare the Pathway Commons data for analysis. Defines the TERM2GENE and TERM2NAME mapping.
- **get_pc_gmtfile :** Used to retrieve Pathway Commons gmt file.
- **get_pc_data :** This function retrieves the pathway data and returns data in gmt format.

8. Packages required would be :

- **dplyr** : Functions for data manipulation(filtering, summarizing, and joining data frames) can be manipulated using

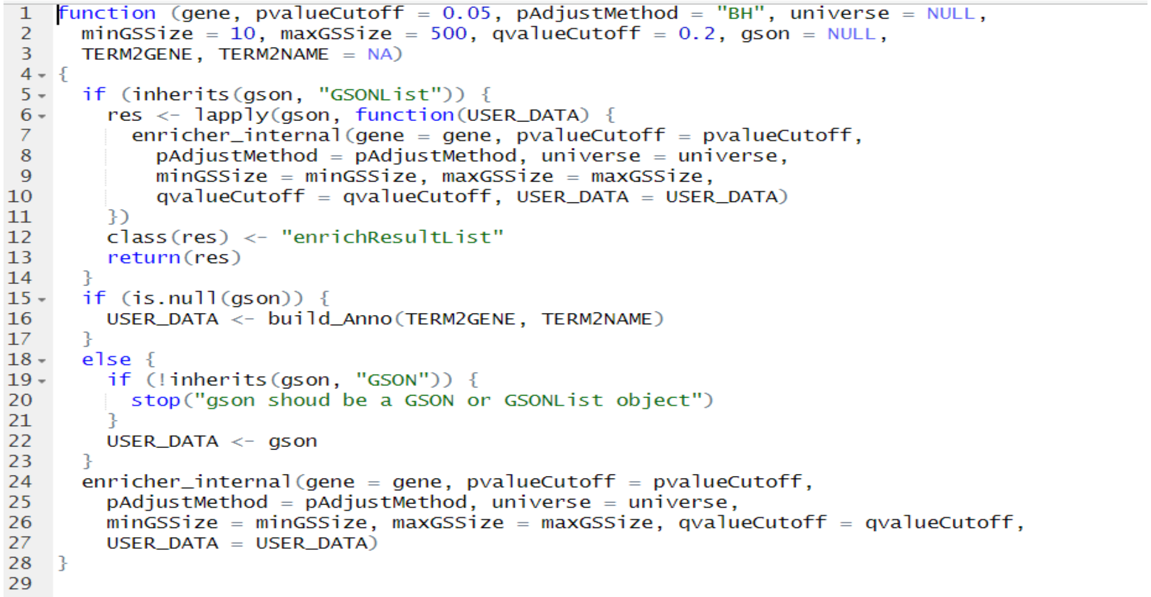
```
library(dplyr)
```

- **magrittr** : Provides a set of pipe operators to chain multiple functions together.

```
library(magrittr)
```

To tweak the usage of functions, the function ***fixInNamespace()*** will be of great significance in modifying an object defined in a package namespace (a special environment that has functions, variables, and other objects). For example, modifying the package code for enricher() function in clusterProfiler package occurs as shown below to change various arguments like minGSSize, qValueCutoff,etc. or any other functions or variables.

fixInNamespace("enricher", "clusterProfiler")



```
function (gene, pvalueCutoff = 0.05, pAdjustMethod = "BH", universe = NULL,
  minGSSize = 10, maxGSSize = 500, qvalueCutoff = 0.2, gson = NULL,
  TERM2GENE, TERM2NAME = NA)
{
  if (inherits(gson, "GSONList")) {
    res <- lapply(gson, function(USER_DATA) {
      enricher_internal(gene = gene, pvalueCutoff = pvalueCutoff,
        pAdjustMethod = pAdjustMethod, universe = universe,
        minGSSize = minGSSize, maxGSSize = maxGSSize,
        qvalueCutoff = qvalueCutoff, USER_DATA = USER_DATA)
    })
    class(res) <- "enrichResultList"
    return(res)
  }
  if (is.null(gson)) {
    USER_DATA <- build_Anno(TERM2GENE, TERM2NAME)
  }
  else {
    if (!inherits(gson, "GSON")) {
      stop("gson shoud be a GSON or GSONList object")
    }
    USER_DATA <- gson
  }
  enricher_internal(gene = gene, pvalueCutoff = pvalueCutoff,
    pAdjustMethod = pAdjustMethod, universe = universe,
    minGSSize = minGSSize, maxGSSize = maxGSSize, qvalueCutoff = qvalueCutoff,
    USER_DATA = USER_DATA)
}
```

Source : RStudio on local machine

3.3.2. Supporting ssGSEA in clusterProfiler :

GSEA:

- stands for Gene Set Enrichment Analysis
- Involves the calculation of a single **enrichment score** for **each gene set.**
- It measures the degree to which the genes in the set are coordinately upregulated and downregulated in a given sample.
- Used to identify patterns in pathways or gene sets that are enriched significantly in differentially expressed genes between two or more sample groups.
- clusterProfiler R package to look for enrichment in KEGG canonical pathways and Gene Ontology.

ES = max(0, ∑(Rank(gene)*Weight(gene))/sqrt(N_genes_in_set))

Where,

Rank(gene) : Rank of the gene in the ranked list based on its differential expression.

Weight(gene) : Weight assigned to the gene – differential expression value multiplied by the sign of its fold change (upregulated genes are positive and downregulated genes are negative).

N_genes_in_set : number of genes in the gene set.

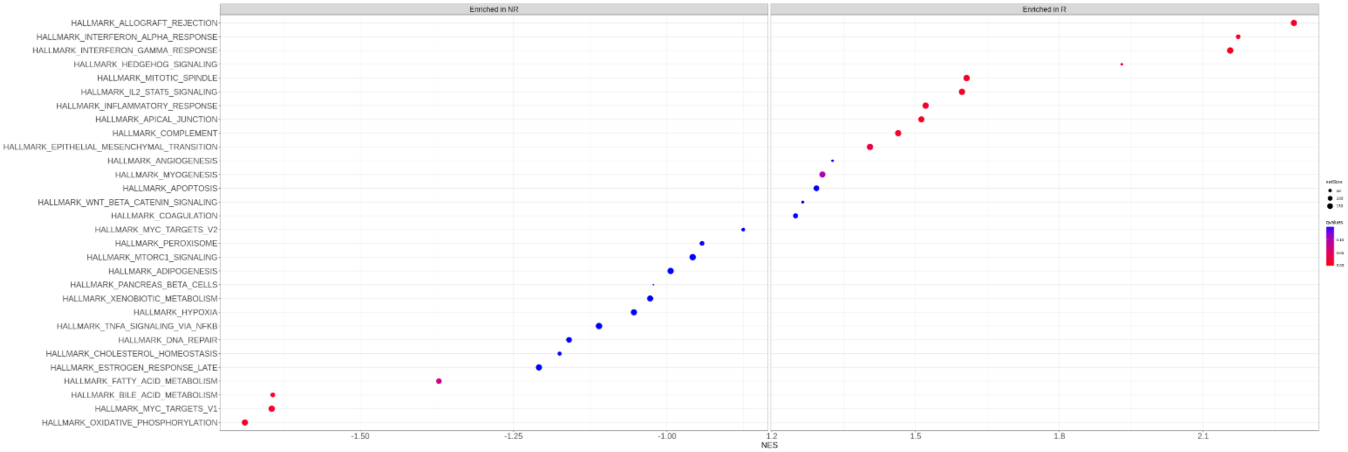


Fig. output for GSEA using the Hallmark genesets.

ssGSEA :

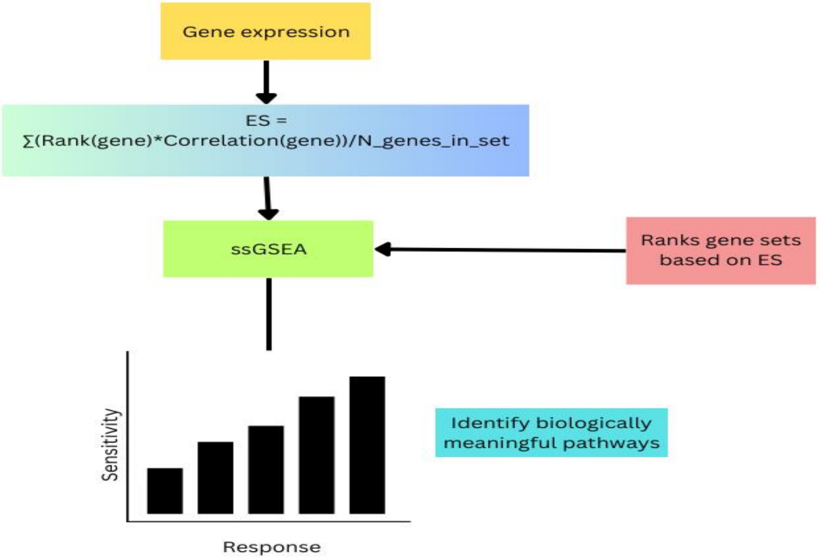
- stands for single-sample Gene Set Enrichment Analysis – variation of GSEA.
- Involves calculation of **enrichment score** for **each sample** to find enriched pathways in the dataset. Suitable for small datasets.
- Reflects the degree to which the genes in the set are upregulated or downregulated relative to other genes in the sample, can identify outliers. The algorithm ranks gene sets based on their scores.
- Used to quantify the activity of a specific pathway or gene set in individual sample, analyzing heterogeneous samples like tumor samples.

ES = ∑ (Rank(gene)*Correlation(gene))/N_genes_in_set

where , *Rank(gene)* : Rank of the gene in the sample based on its expression level.

Correlation(gene) : Correlation coefficient between the expression level of the gene and the phenotype of interest.

N_genes_in_set : Number of genes in the gene set.

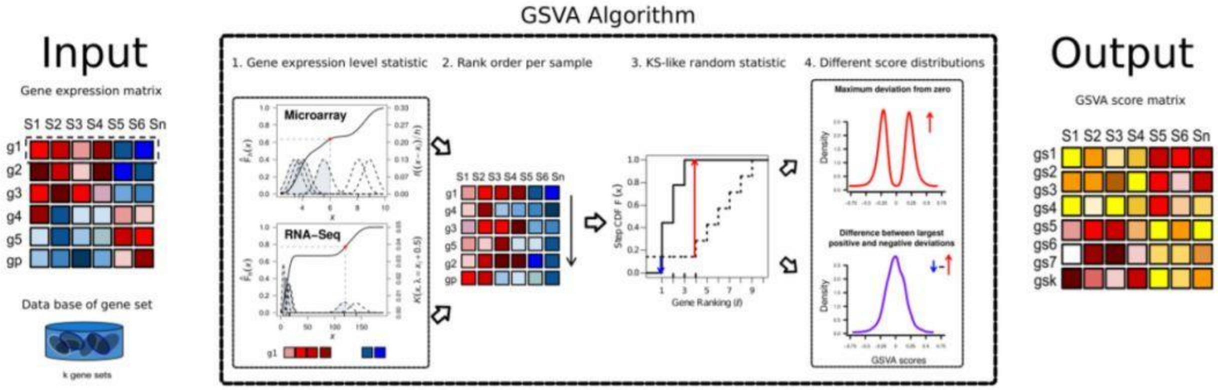

Gene expression
ES =
∑(Rank(gene)*Correlation(gene))/N_genes_in_set
ssGSEA
Ranks gene sets based on ES
Sensitivity
Response
Identify biologically meaningful pathways

GSVA Package :

- Unsupervised , non-parametric computational method used for gene set enrichment analysis which can **detect pathway level changes** in gene expression.
- Transforms gene expression data into **pathway-level score** (ranking the genes according to expression levels and further calculating the running sum of the differences between ranks of genes in pathway and ranks of all other genes in genome)
- Represents **degree of enrichment** for each **sample gene's expression profile** in gene set. Can perform change in coordinate systems to transform data from '*gene by sample matrix'* to '*gene-set by sample matrix'*.

- **Installation** of the development version can be done from the R shell as follows :

```
BiocManager::install(GSVA", version = "devel")
```

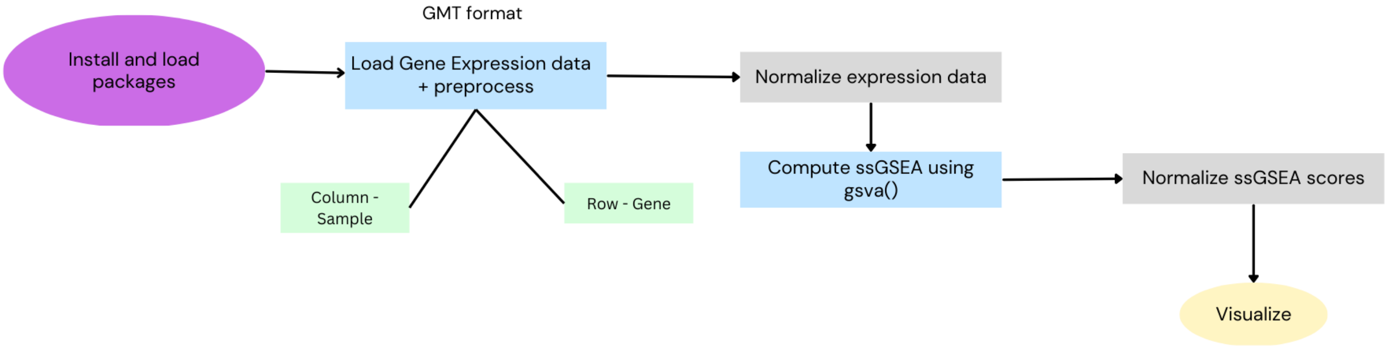


- The methods of our interest in the GSVA package are :
  - `gsva()` (Hänzelmann, Castelo, and Guinney 2013) :
    - Default method of package
    - Non-parametric method – uses empirical cumulative distribution functions (CDFs)+ of gene expression ranks inside and outside the gene set.
    - starts by calculating an expression-level statistic that brings gene expression profiles with different dynamic ranges to a common scale.
  - `ssgsea` (Barbie et al. 2009) : Single sample GSEA (ssGSEA)
    - Non-parametric method
    - Calculates a gene set enrichment score per sample as the normalized difference in empirical CDFs of gene expression ranks inside and outside the gene set.
    - pathway scores are normalized, thus dividing them by the range of calculated values. This normalization step may be switched off using the argument `ssgsea.norm` in the call to the `gsva()` function; see below.

We may use **GenePattern**, which provides software platform for gene and network analysis. It can be downloaded using :

```
install.packages(“full_path_to_gene_pattern_R_package”, repos = NULL)
```

Support for ssGSEA in clusterProfiler can involve the following steps:


GMT format
Install and load packages
Load Gene Expression data + preprocess
Normalize expression data
Column - Sample
Row - Gene
Compute ssGSEA using gsva()
Normalize ssGSEA scores
Visualize

1. Installing latest version of clusterProfiler package in R.

```
install.packages(“GSVA”)
library(GSVA)
```

2. Convert **gene expression data** into **gene expression set** (**GES**) using GSEA base package

library(GSEABase)

eset <- as.geneSet(geneExpressionData)

3. Load gene expression data as a **matrix**. Load GSCs using **read.gmt().** Row correspond to **genes** and columns to **samples**. We normalize the data if required.

4. Perform **ssGSEA analysis** : We compute ssGSEA score after enrichment of gene set in each sample using **gsva()** function.

gene_sets <- read.gmt(“c2.cp.kegg.v7.0.symbols.gmt”)

ssgsea_scores <- gsva(expression_data, gene_stes, method = “ssgsea”, verbose = TRUE)

5*.* Normalise the ssGSEA scores using a function like normalizePathwayScore() which can be scripted in R.

normalized_scores <- normalizePathwayScore(ssgesa_scores)

6. We use the scores to visualize.
7. All the additional functions required will be scripted using R programming.

3.4. Implementation plan and timeline :

Project duration : 12 weeks (Medium)
Project length : 175 hours

- ***Community Bonding period (May 4-28)*** *:*
    - **Week 01: (May 4- May 11) :**
        - Get in touch with mentors
        - Discuss potential tools/packages and finer aspects of the project
        - Ask for suggestions and feedback(if any) from other community
    - **Week 02: (May 12- May 19) :**
        - Go through clusterProfiler documentations and initial project plan
        - Explore, install required packages.
        - Familiarize with the documentations available.
    - **Week 03: (May 20- May 28) :**
        - Understand implementation of Pathway Commons
        - Understanding GSVA package and ssGSEA
        - Preparing for coding period and finalize project plan+schedule.

- ***Coding Phase :***
    - **Week 04: (May 29- June 4) :**
        - Pathway Commons should get annotated, loaded and enriched
    - **Week 05: (June 5- June 12) :**
        - Script underlying functions like enrichPC.
        - ORA made possible for Pathway Commons using clusterProfiler
        - Update Mentors
    - **Week 06: (June 13- June 20):**
        - Code for facilitating functions like gsePC
        - GSEA support for Pathway Commons in clusterProfiler
        - Ask for review of code
    - **Week 07: (June 21- June 28):**
        - Prepare all required functions for complete support of Pathway Commons
        - Document progress and update mentors
    - **Week 08: (June 29- July 5) :**

  - Prepare report for evaluation
  - Test all functions supported.

- **Week 09: (July 6- July 14) Mid-term Evaluation :**
  - Time will be used to optimize the code and speed up visualization
  - Submit Report for mid-evaluations.
- **Week 10: (July 15- July 22) :**
  - Testing functions using GSVA package useful for ssGSEA
  - Scripting necessary functions for ssGSEA
  - Update Mentors
- **Week 11: (July 23- July 31) :**
  - Using GenePattern or any other package with GSVA
- **Week 12: (August 1- August 7) :**
  - Speed up enrichment
  - Optimize code for faster visualizations
- **Week 13: (August 8- August 14) :**
  - Review the scrips, its utility and speed
  - Ask for mentor's views
  - Accommodate any changes
  - Further optimization
- **Week 14: (August 15- August 20) :**
  - Document all the features of Pathway Commons in clusterProfiler.
  - Document for ssGSEA support in clusterProfiler
- **Week 15: (August 21- August 28) : Final week + final mentor evaluation :** Buffer time to compensate for any delays or accommodate any changes required.
- **Week 16: (August 29 – September 5) :**
  - Buffer time for any optimizations
  - Request Mentor for final evaluation
- **After September 5 :**
  - Discuss with mentors about any future enhancements of clusterProfiler
  - Feedback and changes discussed with community members.

3.5. Related Projects :

- A database like Wikipathways supported in clusterProfiler : Link
- GSEA implementation in clusterProfiler : Link
- ssGSEA using GSVA and GenePattern into clusterProfiler : Link

3.6. Identify possible hurdles and questions that will require more research and planning

- I have attempted to implement ssGSEA in clusterProfiler using the GSVA package however, I need to figure out a way to implement **GSVA** to work very **fast** enough with **permutations** of data and calculate its normalized enrichment score.

4. Availability during GsoC :

- Do you have any other time-consuming activities scheduled during the coding period?
  - None whatsoever, I will have a break during summers and can dedicate time solely to this project.
- Do you have a full- or part-time job or internship planned for this summer?
  - None
- How many hours per week do you have available for GSoC?
  - 40 hours / week or 6 hours / day
- Where will you be located during GSoC? Are you traveling during the summer?
  - I will be located in Bangalore, India during the GSOC period and have no plans of travelling anywhere.

5. References :

- Pathway Commons :
  - http://pathwaycommons.org/ , https://www.pathwaycommons.org/archives/PC2/v12/
- clusterProfiler :
  - https://bioconductor.org/packages/release/bioc/html/clusterProfiler.html , https://www.sciencedirect.com/science/article/pii/S2666675821000667
- Vignettes of GSVA package https://bioconductor.org/packages/devel/bioc/vignettes/GSVA/inst/doc/GSVA.html
- About paxtoolsR :
  - https://bioconductor.org/packages/release/bioc/html/paxtoolsr.html, https://bioconductor.org/packages/release/bioc/manuals/paxtoolsr/man/paxtoolsr.pdf, https://www.bioconductor.org/packages/devel/bioc/vignettes/paxtoolsr/inst/doc/using_paxtoolsr.html
- Implementing database for clusterProfiler(WikiPathways) :
  - https://github.com/YuLab-SMU/clusterProfiler/blob/master/R/wikiPathways.R
- Gene set enrichment analysis using clusterProfiler :
  - https://github.com/YuLab-SMU/clusterProfiler/blob/master/R/gseAnalyzer.R ,
- Single sample gene set enrichment Analysis (ssGSEA) https://www.genepattern.org/modules/docs/ssGSEAProjection/4#gsc.tab=0 , https://www.youtube.com/watch?v=bnySyariq5I&t=214s
- Hänzelmann S., Castelo R. and Guinney J. GSVA: gene set variation analysis for microarray and RNA-Seq data. BMC Bioinformatics, 14:7, 2013.
- GenePattern : https://www.genepattern.org/tutorial/archives/333/gp_programmer.pdf
- GSEA vs ssGSEA : https://liulab-dfci.github.io/RIMA/Differential.html