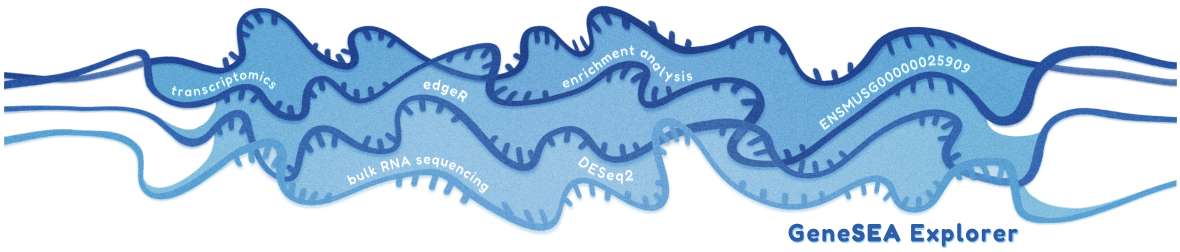
GeneSEA Explorer: A Tool for Exploring the Depths of Gene Expression Data with Shannon Entropy Analysis



User's Guide

Ana M. Gonçalves*[1,2,3], Pedro Macedo[1], Patrício Costa[2,3,4], Nuno S. Osório[2,3]

1. Center for Research and Development in Mathematics and Applications (CIDMA), Department of Mathematics, University of Aveiro, 3810-193 Aveiro, Portugal

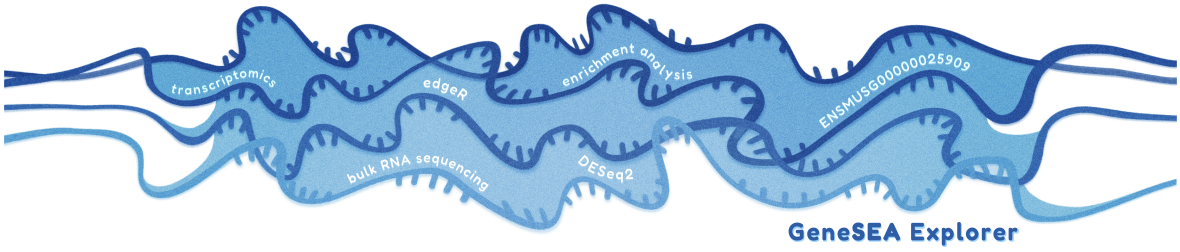
2. Life and Health Sciences Research Institute (ICVS), School of Medicine, University of Minho, Campus Gualtar, 4710-057 Braga, Portugal

3. ICVS (Life and Health Sciences Research Institute)/3B's (Biomaterials, Biodegradables and Biomimetics) Associate Laboratory, 4806-909 Guimarães, Portugal

4. Faculty of Psychology and Education Sciences, University of Porto, Porto, Portugal

First edition: 8 July 2024
Last revised: 18 July 2025

Contents



Introduction

The GeneSEA Explorer is an innovative bioinformatics tool for conducting differential gene expression (DGE) analysis, using Shannon Entropy Aggregation (SEA). Its user-friendly interface enables users to effortlessly explore and analyze diverse DEGs outputs and to conduct functional enrichment analysis. The innovative use of Shannon entropy to aggregate all DEGs outputs provides an informative selection of differentially expressed genes (DEGs), enhancing researchers' understanding of their RNA sequencing (RNA-Seq) data results and minimizing challenges for researchers less confident in this domain.

It is important to note, however, that GeneSEA Explorer is neither more adequate nor more advanced than other existing RNA-Seq analysis platforms. Its design prioritizes functionality, data interpretation, and reproducibility, while requiring minimal knowledge in programming. Additionally, the GeneSEA Explorer was developed to address the challenges that still remain in identifying the DEGs while selecting a suitable normalization method, given that each normalization provides different results.

1.1 Scope

This guide provides an overview of the GeneSEA Explorer, a Shiny application for DGE analyses of read counts arising from RNA-Seq or similar technologies. The code repository can be seen in GeneSEA Explorer: repository. The Shiny application can be applied to any technology that produces read counts for genomic features.

1.2 Workflow

DGE analysis using RNA-Seq is essential to identify genes with varying expression under different conditions, at any given time, providing critical insights into biological and pathological processes. In GeneSEA Explorer, we use DESeq2 and edgeR, R packages that employ a model based on the negative binomial distribution. This model adjusts for sample variations and provides robust statistical inference for identifying DEGs between two groups. The main workflow uses edgeR's quasi-likelihood pipeline (edgeR-quasi) for the DGE analysis. The results are adjusted using the Benjamini-Hochberg method with a false discovery rate (FDR) cutoff selected by the user. After identifying up-regulated and down-regulated DEGs based on positive and negative log2-fold-change values, with a given threshold provided by the user, the application proceeds to perform a comparative analysis between the several normalization methods and the SEA. Lastly, the application provides a section to determine the enriched terms associated with the genes lists.

By only imputing the RNA-Seq count matrix and selecting the variable of interest in the application, not only the users can explore and compare diverse DEGs outcomes across

several normalization methods but also compare these results with the ones provided by the SEA. The application displays analyses commonly reported in the literature, such as volcano and Multidimensional scaling (MDS) plots. The application also provides tables displaying the log2-fold-changes and adjusted p-values, along with a table comparing up-regulated and down-regulated genes of each normalization method. Additionally, there is a section where users can view the calculation of the normalization weights and the ranking of each gene, through the SEA methodology. Additionally, users can also see how many genes are present in both the SEA gene list and the list from the normalization method selected by the user. Finally, the user can perform a functional enrichment analysis with Over-Representation Analysis (ORA) and Gene Set Enrichment Analysis (GSEA).

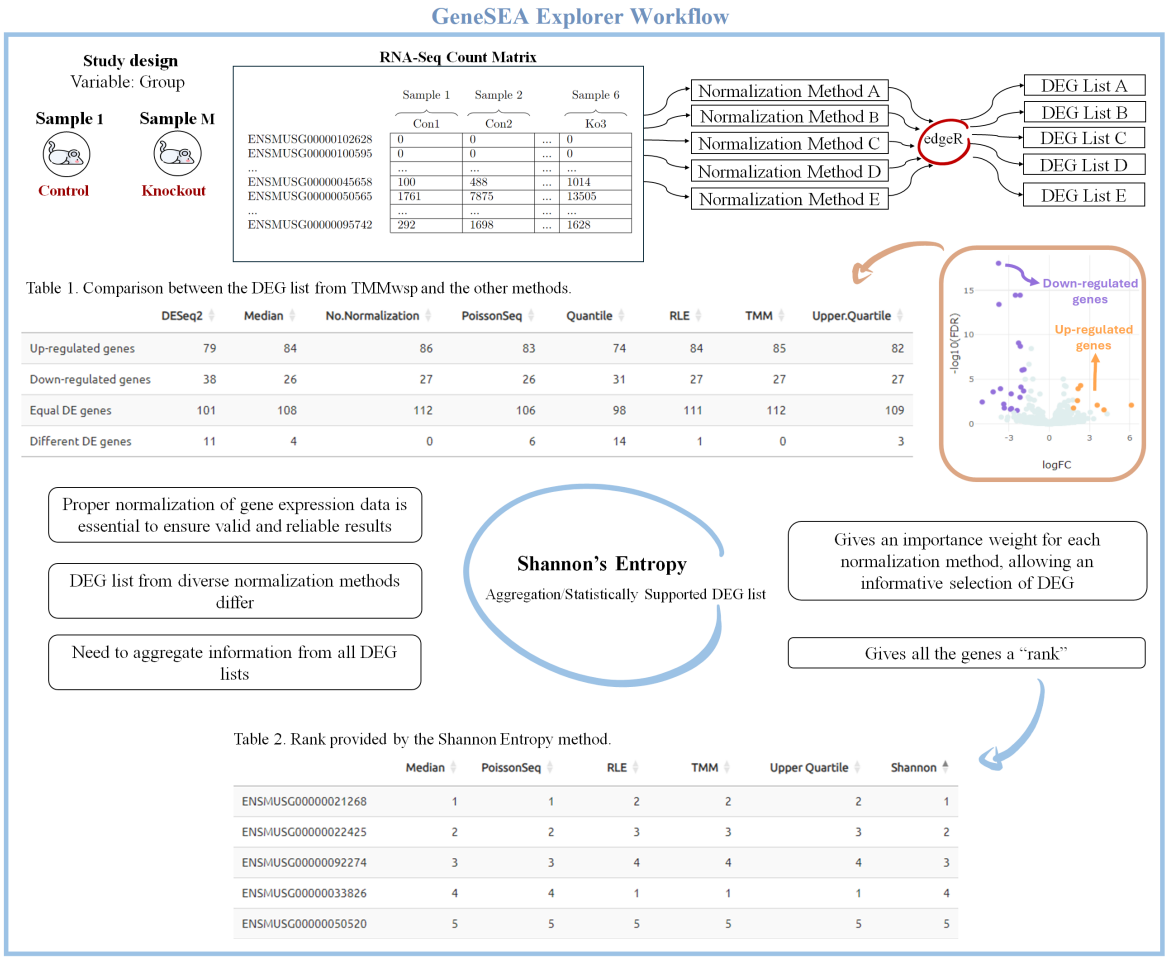


Figure 1: Schematic representation of the GeneSEA Explorer analysis workflow.

Note: The exact duration of each analysis step primarily depends on the user's computational processing power.

Demo dataset

The Gene Expression Omnibus (GEO) [1] is a broad repository of gene expression data generated across multiple platforms (e.g., microarray, bulk RNA-Seq, scRNA-seq) and from multiple organisms that is hosted by the National Institutes of Health (NIH). When publishing new RNA-Seq data, GEO is often chosen by authors for data deposition due to its

inclusiveness and NIH oversight. Therefore, the dataset used in this work (a bulk RNA-Seq dataset, Table 1) was retrieved from this repository, visible in GEO: The SSBP3 co-regulator is required for glucose homeostasis, pancreatic islet architecture, and beta-cell identity.

This example aims to illustrate the effects of selecting a normalization method, not only on the identification, or not, of statistically significant, DEGs, but also on the subsequent insights drawn from the identified genes, which are crucial for researchers.

Table 1: Demo dataset.

(a) RNA-Seq matrix.

	Con1	Con2	···	Ko3
ENSMUSG00000102628	0	0	...	0
ENSMUSG00000100595	0	0	...	0
...	...	...	...	...
ENSMUSG00000045658	100	488	...	1014
ENSMUSG00000050565	1761	7875	...	13505
...	...	...	...	...
ENSMUSG00000095742	292	1698	...	1628

(b) Metadata table.

ID	Group
Con1	Control
Con2	Control
Con3	Control
Ko1	Knockout
Ko2	Knockout
Ko3	Knockout

1.3 Quick Start

Home page

Upon initially launching the application in either a web browser or RStudio, the first page displayed is the Shiny GeneSEA Explorer application's home page (Figure 2). This page provides an overview of the project along with acknowledgments.

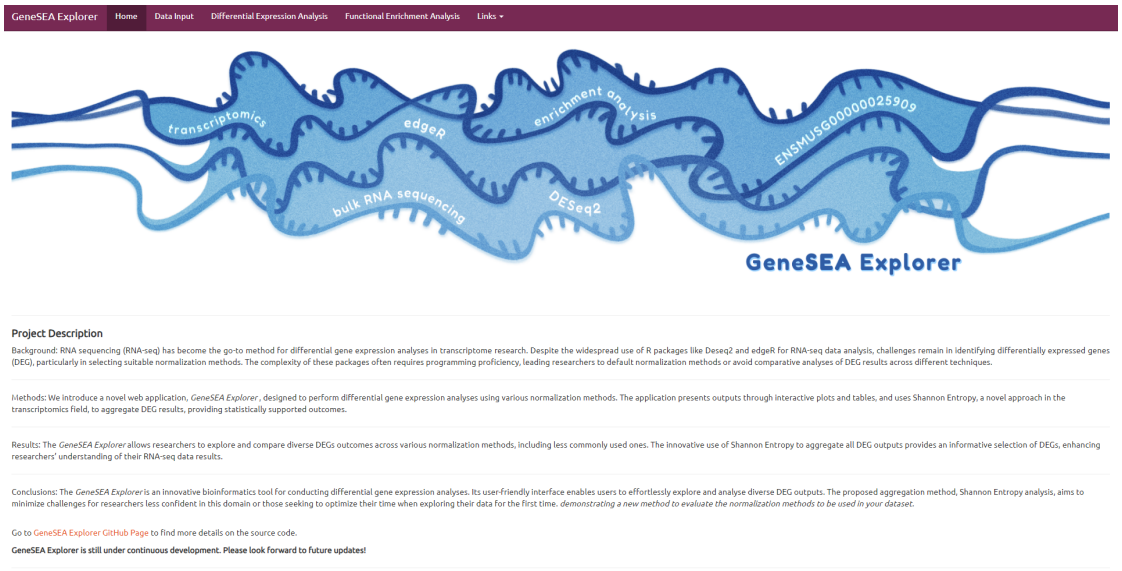


Figure 2: GeneSEA Explorer home page.

Data Input

The section *Data Input*, in a first moment, allows the users to input their RNA-Seq data files in the .txt, .xls, .xlsx, .tsv, and .csv formats. The matrix will consequently appear in the right side of the page. Following the data upload, new options will pop-up, namely the options "Variable to be studied", "Variables to be compared" and "Contrast". The option "Number of replicates" refers to the smaller number of replicates in the data sample. In the case of the demo dataset, both samples groups have three biological replicates.

By selecting the "Demo dataset" button, as illustrated in Figure 3, the fields will automatically be filled with default options.

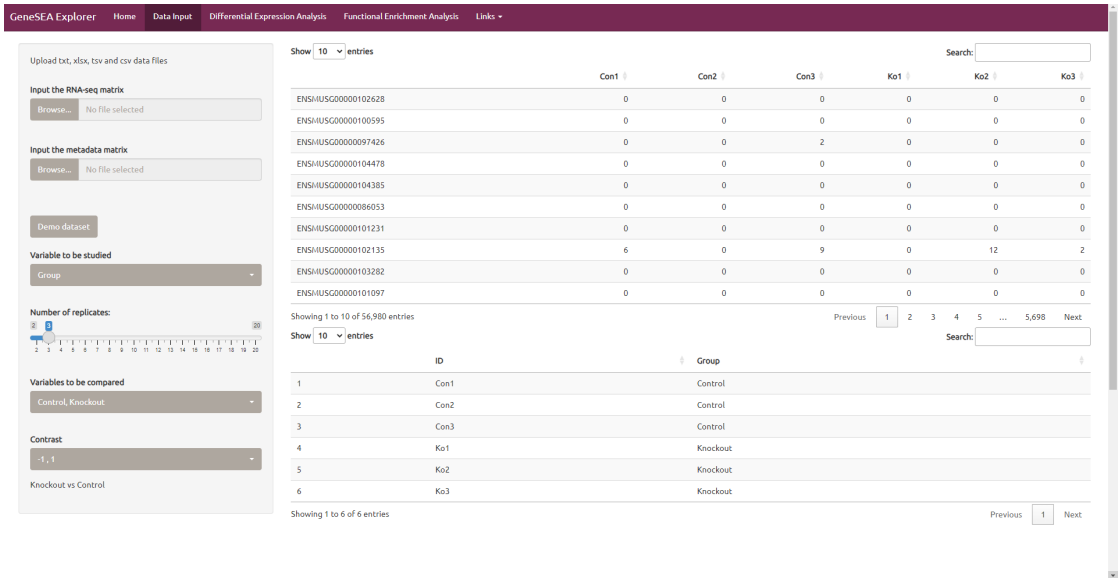


Figure 3: GeneSEA Explorer: *Data Input* section.

Differential Gene Expression Analysis

The section *Differential Gene Expression Analysis* provides three data analysis sections. The first one, the *Volcano Plots* subsection (Figure 4) depicts the DEGs results retrieved from the No Normalization, TMM, TMMwsp, DESeq2, RLE, PoissonSeq, Median, Upper-Quartile, and Quantile normalization methods. The second subsection, the *Results*, depicted in Figure 5, provides the following outputs: a MDS plot, both in the GeneSEA Explorer Shiny web page and in the Glimma web page [2], a volcano plot, a table with the differential expression results and a comparative table which depicts how many DEGs are up- or down-regulated for each normalization method, and how many genes from the DEG list differ between the current normalization method and the others, and a Venn diagram. Lastly, the third subsection, the *Differential Gene Expression Analysis with SEA* (Figure 6), provides the workflow of the Shannon entropy aggregation method while showcasing the normalization weights and genes ranking.

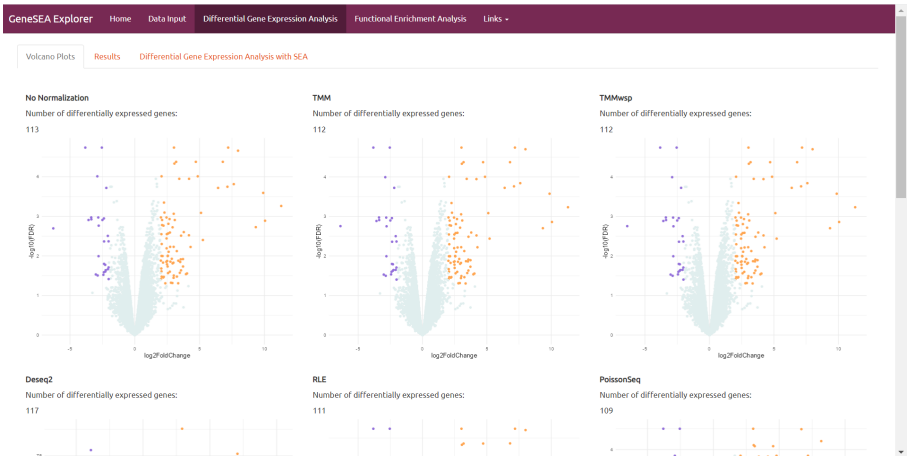


Figure 4: *Volcano Plots*: *Differential Gene Expression Analysis* subsection from the GeneSEA Explorer application.

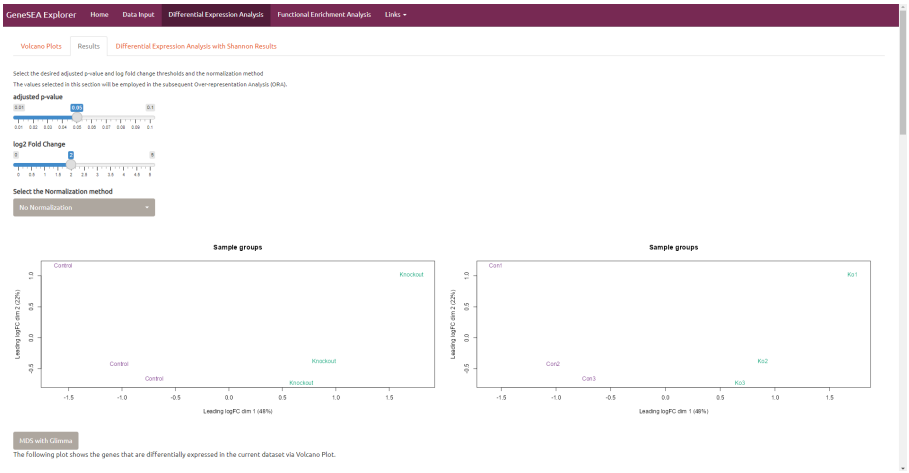


Figure 5: *Results*: *Differential Gene Expression Analysis* subsection from the GeneSEA Explorer application.

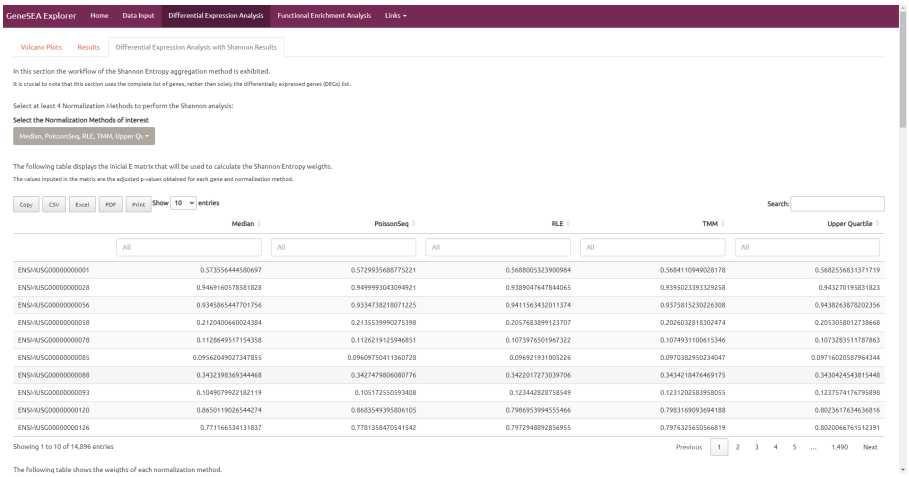


Figure 6: *Differential Expression Analysis with SEA*: *Differential Gene Expression Analysis* subsection from the GeneSEA Explorer application.

Functional Enrichment Analysis

The *Functional Enrichment Analysis* section (Figure 7) provides the user the option to use the Gene Ontology (GO) [3,4] and Kyoto Encyclopedia of Genes and Genomes (KEGG) [5,6] tools to perform an enrichment analysis, through the clusterProfiler package [7]. This section allows the selection of several necessary options to perform the functional enrichment analysis. The results outputs from the current analysis are provided on the right side of the page, where barplots, dotplots, goplots, emapplots, cneplot, heatplots, and upsetplots, retrieved with the ORA and GSEA methods, are shown.

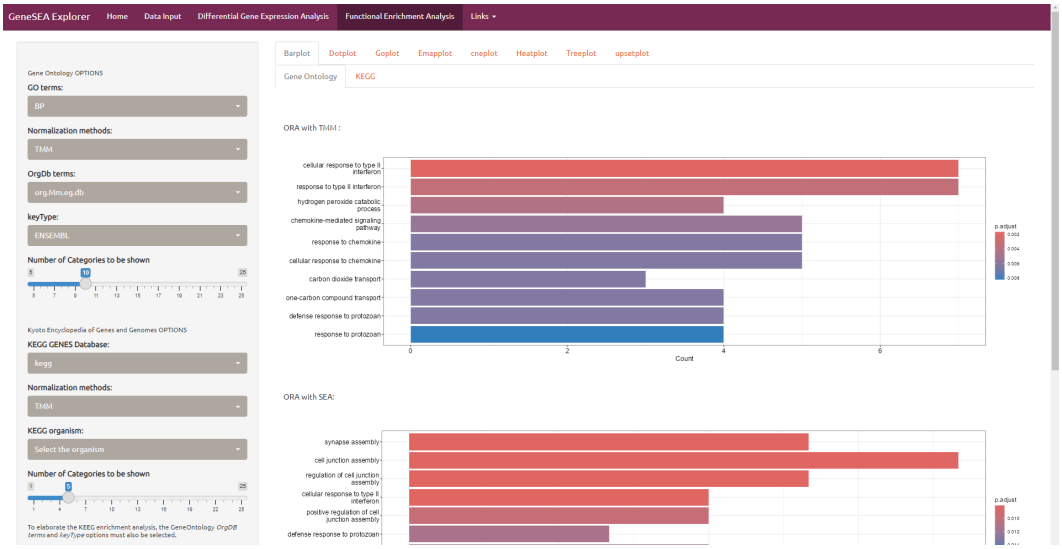
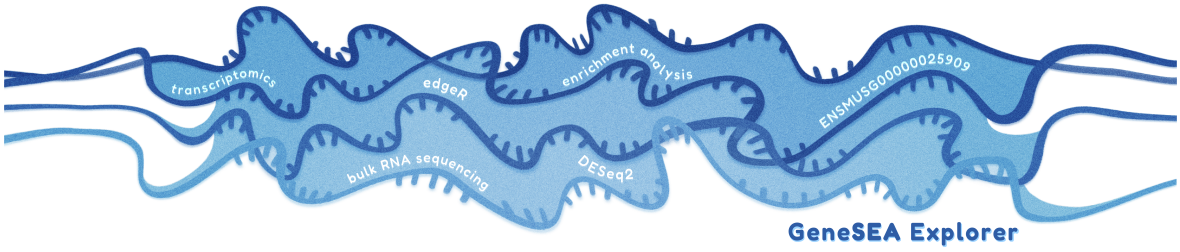


Figure 7: *Functional Enrichment analysis* section page from the GeneSEA Explorer application.



Specific Experimental Designs

2.1 RNA-Seq Read Counts Matrix and Metadata

edgeR works on a table of read counts, with rows corresponding to genes and columns to independent libraries. The counts represent the total number of reads aligning to each gene.

The first step in the analysis will be to input the RNA-Seq read counts matrix into the R session. A metadata file is mandatory to initiate the analysis.

Accepted input formats

GeneSEA Explorer accepts five types of input, each described in detail below.

.csv files

Comma-separated values (.csv) is a text file format that uses commas to separate values, and new lines to separate records. These files are imported in the Shiny application with the function *read.csv* from the R.

```
read.csv(infile$datapath, header = TRUE)
```

.txt files

The text files (.txt) are the most basic text file format, used for generating plain text files with little to no formatting or styling. The .txt files accepted in the Shiny application should be in the format tabular separated by a tab character.

```
read.table(infile$datapath, header = TRUE)
```

If your text files separates the columns with comma or semicolon rather than with a tab separation, their substitution should be performed before importing the data in the Shiny application. To perform those changes, just open the text file and select "Edit" → "Substitute".

.tsv files

The tab separated values (.tsv) is a simple, text-based file format for storing tabular data. These files are stored in columns and rows and separated by a tab character. Therefore, the file is imported in the Shiny application with the function *read.table* from the package *utils*.

```
read.table(infile$datapath, header = TRUE)
```

.xls and .xlsx files

The files in these formats are imported into the application with the function *read_xlsx* from the package *readxl* from the *tidyverse* library. The package supports both the legacy .xls format and the modern xml-based .xlsx format.

```
read_xlsx(infile$datapath)
```

2.2 Errors

There are some errors that might show up in the *Data Input* section where the user can control them.

Error: invalid 'row.names' length

If the following error appears that means that some of your genes names are considered to have symbols within. The R does not behave well with symbols. Therefore, the basic solution is to copy the data from the .csv, .xls and .xlsx to a .txt file and then import the .txt in the GeneSEA Explorer.

Important: deleting the special characters (e.g., ., −, !, ;, #, and others) in the IDs or in the column names manually will probably not solve the error. After removing the special characters, the best following procedure will be to copy the data from the file to a .txt file.

Error: line 1 did not have 19 elements

Note: it could be other number than 19. This only means that it was expected 19 elements in the data per row.

If the following error appears that means that some of your column names might have spaces between the words. That is, instead of having the names in the format "Mod_H3br53_Cntrl" they are in the format "Mod H3br53 Cntrl", as example. These spaces provide false information of the number of columns that exist in your data.

2.3 Design Variable

In the application, under the *Data Input* section, a button following a text labeled "*Variable to be studied*" will appear, allowing the users to select the variable of interest for the analysis. Below this, an additional button will allow users to specify which levels of the selected variable they wish to compare.

In the demo dataset, the variable of interest is "Group" and the comparison is between the ***Control*** and the ***Knockout*** samples.

Note: only one variable can be selected. If the interest is to use a more complex analysis design, go to the GeneSEA Explorer Limitations section.

2.4 Contrasts

In the application, in the *Data Input* section, it will appear a button with the following text "*Contrast*" which will allow the user to select a contrast vector between two options. Taking into consideration the Demo dataset, the variables of interest are ***Control*** and ***Knockout***, therefore, the possible contrast vectors that are shown are:

- `contrast = c(-1,1)`

```
# The contrast argument in this case requests a statistical test of the
null hypothesis that coefficient2-coefficient1 is equal to zero.
```

- `contrast = c(1,-1)`

```
# The contrast argument in this case requests a statistical test of the
null hypothesis that coefficient1-coefficient2 is equal to zero.
```

Note: any information placed to the right of a "#" is designated as a comment and is not executed by the R terminal. This feature is useful for embedding instructions or annotations directly within the code.

The first option means that we are comparing ***Knockout*** vs. ***Control*** while if the second option is selected we will be performing the comparison ***Control*** vs. ***Knockout***. This information is provided to the user in the Shiny application interface, bellow the contrast selection button.

Using as example the TMM normalization method, if the 2^{nd} contrast is selected, the interpretation of the results provided by the contrast will be: 27 genes from the sample ***Control*** are up-regulated while 85 are down-regulated comparatively to the sample ***Knockout***.

2.5 Filtering

Genes with consistently low counts across all the libraries should be removed before downstream analysis for both biological and statistical reasons. From a biological point of view, a gene must reach a certain minimal expression level to be considered biologically important. On the other hand, from a statistical point of view, genes with consistently low counts are unlikely to show significant differential expression due to insufficient statistical information.

As a guideline, genes should have a count of at least 10–15 in some libraries before it is considered to be expressed in the study. Instead of directly filtering based on raw counts, it is preferable to use counts per million (CPM) values, as this accounts for differences in library sizes between samples. Filtering by counts alone can unfairly favor genes in larger libraries [8]. To adjust for this, the filtering threshold metric is set to 10-15/L, with L being the size of the smallest library in the RNA-Seq data.

In the GeneSEA Explorer, the filtering retains genes with CPM values above the specified *cutoffvalue* in at least *input$Replicates* libraries, as illustrated below.

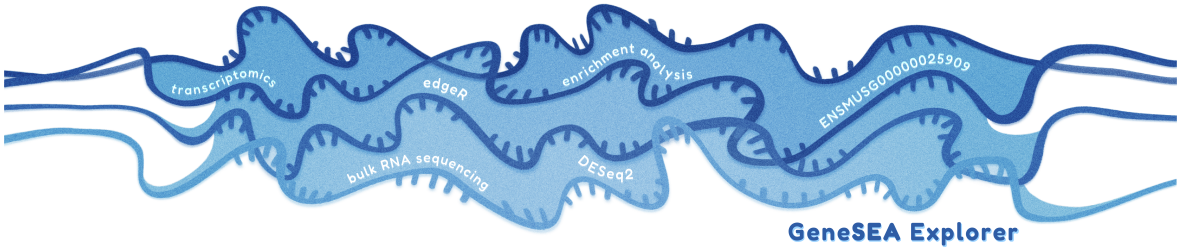
```
cutoffvalue <- reactive ({
L <- as.numeric(min(colSums(expression.matrix)))
threshold <- as.numeric((10)/(L/10^6))
threshold
})

keep = reactive({
rowSums(cpm(expression.matrix) > cutoffvalue()) >= input$Replicates
})

expression.matrix1 <- reactive ({
data <- expression.matrix[keep(),]
data
})
```

In the GeneSEA Explorer, the cutoff value for the expression matrix is set to a CPM of 10/L, where L represents the minimum library size in millions. While the library sizes vary and often exceed 20–25 million, this threshold was chosen for simplicity. The precise cutoff value is not critical, as the downstream DGE analysis is largely insensitive to small variations in this parameter.

The requirement of "*>= input$Replicates*" libraries, where *input$Replicates* is the minimum number of replicas in the dataset samples ensures that a gene will be retained if it is expressed in all defined libraries.

Replicas

The number of replicas in the dataset is provided in the GeneSEA Explorer interface, by the user, in the *Data Input* section.



Specific Analysis

3.1 Normalization Methods

Several studies have compared different normalization approaches for RNA-Seq data. Proper normalization of gene expression data is essential to ensure valid and reliable results from downstream analyses. Normalization is the process that aims to account for the bias and make samples more comparable. The selection of a proper normalization method is a pivotal task for the reliability of the downstream analysis and results [9–12].

Of primary concern, there is no consensus regarding which normalization and statistical methods are the most appropriate for analyzing this data. The lack of standardized analytical methods leads to uncertainties in data interpretation and study reproducibility, especially with studies reporting high false discovery rates.

Different normalization methods address systematic biases in the data differently, and thus choosing an optimal normalization method for a given data set is critical. As the source of systematic bias in the data is usually unknown, an exhaustive comparative evaluation of both non-normalized data and the data normalized through different methods is required to select a suitable normalization method [9, 10] and, consequently, the DEG list.

In opposition to the DESeq2 package, in the edgeR workflow, the RNA-Seq matrix normalization is performed outside of the edgeR functions and, therefore, other normalization methods, such as PoissonSeq [13] and Median [10], can be performed and incorporated in the edgeR workflow. Given this feature, the edgeR workflow was selected for the Shiny application. In the current subsection, the normalization methods used in the GeneSEA Explorer are briefly described.

Quantile

The quantile normalization is a global adjustment method that assumes the statistical distribution of each sample is the same. This normalization method forces the distributions of the samples to be the same by replacing each point of a sample with the mean of the corresponding quantile [12, 14].

The quantile normalization was performed using the *normalize.quantiles* function from the R/Bioconductor package *preprocessCore* [15].

Upper-quartile

The upper-quartile normalization is a method in which the scale factors are calculated from the 75% quantile of the counts for each library, after removing genes that are zero in all

libraries [10, 12]. The scaling factors are then used to adjust the total mapped reads count for each sample.

The upper-quartile normalization was implemented using the the R/Bioconductor edgeR package approach [8, 16].

TMM

The trimmed mean of M values (TMM) normalization, a method proposed in [16], assumes that the majority of genes, common to both samples, are not differentially expressed. Therefore, TMM is commonly used and recommended for most RNA-Seq data where the majority (more than half) of the genes are believed to be not differentially expressed between any pair of the samples.

TMM trims away extreme log-fold-changes to normalize the counts based on the remaining set of non-differentially expressed genes. If many genes are uniquely or highly expressed in one experimental condition, it will affect the accurate quantification of the remaining genes. To adjust for this possibility, TMM calculates scaling factors to adjust library sizes and composition for the normalization of samples within a dataset. To perform this, one sample is chosen as a reference sample. The fold changes and absolute expression levels of other samples within the dataset are then calculated relative to the reference sample [16].

In this work, in the TMM normalization method, the counts per gene were normalized using the TMM approach in the R/Bioconductor edgeR package [8, 16]. The scaling factors for each sample were generated using the *normlibsizes* function.

TMMwsp

The TMMwsp method stands for "TMM with singleton pairing". This is a variant of the TMM normalization method that is intended to perform better for data with a high proportion of zeros. In the TMM method, genes that have zero count in either library are ignored when comparing pairs of libraries. In the TMMwsp method, the positive counts from such genes are reused to increase the number of features by which the libraries are compared. The singleton positive counts are paired up between the libraries in decreasing order of size and then a slightly modified TMM method is applied to the re-ordered libraries [17]. In the new edgeR version (edgeR v4 [18]) a new TMMwsp method has been added to provide more robust behavior for sparse data with many zeros.

The TMMwsp normalization was implemented using the the R/Bioconductor edgeR package approach [8,16]. Like TMM normalization method, the scaling factors were generated using the *normlibsizes* function.

Median

In this normalization method, a scaling factor for a given sample takes the median of the reads of observed samples counts to the geometric mean across samples. Therefore, the counts per gene were median normalized by dividing it by the median of mapped reads for all the samples and multiplying by $1{\times}10^6$ [10, 12].

The median normalization was implemented using the *normalizeMedianValues* function from the R/Bioconductor package *limma* [14].

RLE

The RLE is the scaling factor method proposed by Anders and Huber (2010) [19] implemented in the edgeR package. It is defined as "relative log expression" and it is a library size normalization method. In this method, the scaling factor is calculated as the median of the ratio, for each gene, of its read counts over its geometric mean across all samples. By assuming most genes are not differentially expressed, the median of the ratio for a given sample is used as a correction factor to all read counts to fulfill this hypothesis [20].

The RLE normalization was implemented using the R/Bioconductor edgeR package approach [8, 16].

PoissonSeq

The PoissonSeq normalization method, further detailed in [13], estimates the sequencing depths of experiments using a method based on Poisson goodness-of-fit statistic, providing a scaling factor for each sample. The expression matrix is afterwards normalized with the scaling factors.

The PoissonSeq normalization was performed using the *PS.Est.Depth* function from the R/Bioconductor package *PoissonSeq* [13].

DESeq2

Scaling factor method proposed by Anders and Huber (2010) [19].

The DESeq2 normalization was implemented using the the R/Bioconductor DESeq2 package approach [19, 21].

Given that the RNA-Seq matrix normalization is performed within the *DESeq()* function, it is not possible or recommended to perform other normalization functions within the DESeq2 workflow. Therefore, in opposition to the other normalization methods, this was the only normalization method applied with the DESeq2 workflow.

3.2 Shannon Entropy Aggregation Methodology

Based upon the obtained adjusted p-values acquired with the normalization methods, previously detailed, we proceed to acquire a gene ranking through the SEA by creating an entropy matrix as:

$$E = \begin{array}{c} \begin{array}{cccc} M_1 & M_2 & ... & M_k \\ \downarrow & \downarrow & & \downarrow \end{array} \\ \begin{bmatrix} E_{1,1} & E_{1,2} & ... & E_{1,k} \\ E_{2,1} & E_{2,2} & ... & E_{2,k} \\ A_{3,1} & E_{3,2} & ... & E_{3,k} \\ \vdots & \vdots & & \vdots \\ E_{n,1} & E_{n,2} & ... & E_{n,k} \end{bmatrix} \end{array} \begin{array}{cc} \leftarrow & G_1 \\ \leftarrow & G_2 \\ \leftarrow & G_3 \\ & \vdots \\ \leftarrow & G_n \end{array} \tag{1}$$

where M_l $(l = 1, 2, ..., k)$ represents the normalization methods, the rows, G_j $(j = 1, 2, ..., n)$, denote the genes from the DGE analysis, and $E_{j,l}$ constitutes the adjusted p-values acquired by each one of the genes and normalization methods, through the edgeR workflow, for each l normalization method and for each j gene.

The DEG lists are incorporated into the matrix exactly as they appear in the edgeR output. Additionally, users can select the normalization methods of interest, with a minimum of four methods recommended. For the demo dataset, the normalization methods Median, PoissonSeq, RLE, TMM, and Upper Quartile are pre-selected.

The initial entropy matrix (defined in the *MatrizInicialE()* function in the source code), from the demo dataset, is shown in Figure 8.

	Median	PoissonSeq	RLE	TMM	Upper Quartile
	All	All	All	All	All
ENSMUSG00000000001	0.573556444580697	0.5729935688775221	0.5688005323900984	0.5684110949028178	0.5682556831371719
ENSMUSG00000000028	0.9469160578581828	0.9499993043094921	0.9389047647844065	0.9395023393329258	0.943270195831823
ENSMUSG00000000056	0.9345865447701756	0.9334738218071225	0.9411563432011374	0.9375815230226308	0.9438263878202356
ENSMUSG00000000058	0.2120400660024384	0.2135539990275398	0.2057683899123707	0.2026032818302474	0.2053058012738668
ENSMUSG00000000078	0.1128649517154358	0.1126219125946851	0.1073976501967322	0.1074931100615346	0.1073283511787863
ENSMUSG00000000085	0.09562049027347855	0.09609750411360728	0.096921931005226	0.0970382950234047	0.09716020587964344
ENSMUSG00000000088	0.3432398369344468	0.3427479806080776	0.3422017273039706	0.3434218476469175	0.3430424543815448
ENSMUSG00000000093	0.1049079922182119	0.105172550593408	0.123442828758549	0.1231202583958055	0.1237574176795898
ENSMUSG00000000120	0.8650119026544274	0.8683549395806105	0.7986953994555466	0.7983169093694188	0.8023617634636816
ENSMUSG00000000126	0.771166534131837	0.7781358470541542	0.7972948892856955	0.7976325650566819	0.8020066761512391

Showing 1 to 10 of 14,896 entries Previous 1 2 3 4 5 ... 1,490 Next

Figure 8: *Differential Expression Analysis with SEA*: initial entropy matrix for the SEA.

Upon acquiring the initial matrix, we follow these four steps to determine the degree of importance of the normalization methods and rank the genes [22, 23]:

1. compute the entropy normalization as $\overline{E}_{j,l} = \frac{E_{j,l}}{\sum_{j=1}^{n} E_{j,l}}$.

```
Ejl <- reactive({ apply(TakeTheDeletedToNA(), 2,
 function(col) col / sum(col, na.rm = T)) })
```

2. compute entropy, e_l, as $e_l = -e_0 \sum_{j=1}^{n} \overline{E}_{j,l} \ln \overline{E}_{j,l}$, where e_0 is the Shannon entropy constant, usually considered as $e_0 = (\ln n)^{-1}$.

```
e0 <- reactive({ (log2(nrow(TakeTheDeletedToNA())))^-1 })

m1 <- reactive({
     data <- Ejl()*log2(Ejl())
     data <- as.matrix(data)
     data })

el <- reactive({
     data <- -e0() * colSums(m1(), na.rm = T)
     data <- matrix(data)
     data })
```

3. set d_l as the degree of diversification with $d_l = 1 - e_l$, for $l = 1, 2, ..., k$.

```
dl <- reactive({
      data <- 1 - el()
      data <- matrix(data)
      data })
```

4. compute the degree of importance of the normalization methods, M_l, with

$$W_l = \frac{d_l}{\sum_{l=1}^{k} d_l}.$$

```
wl <- reactive ({
      data <- dl()/sum(dl(), na.rm = T)
      data <- matrix(data)
      data })
```

The degree of importance for each selected normalization method is displayed in the GeneSEA Explorer interface, as shown in Figure 9.

Median	PoissonSeq	RLE	TMM	Upper Quartile
0.1988280328633069	0.1988445276698477	0.200741500023304	0.2008066645084733	0.2007792749350681

Figure 9: *Differential Expression Analysis with SEA*: degree of importance of the selected normalization methods.

After calculating W_l, for $l = 1, 2, ..., k$, we obtain the following importance index, which combines the importance scores (provides a weighted-sum of the importance scores) of all of the considered normalization methods, regarding the values of W_l, and is suitable to provide, for a $j = 1, 2, ..., n$, a full gene ranking by:

$$\beta_j = \sum_{l=1}^{k} W_l \; E_{j,l}. \tag{2}$$

```
Bj <- reactive ({
      data <- rowSums(c(wl())*Ejl())
      data <- matrix(as.numeric(data), ncol = 1)
      data })
```

The genes ranking generated by the SEA is presented at the end of the *Differential Expression Analysis with SEA* section, as depicted in Figure 10.

	Bj	rank
	All	All
ENSMUSG00000000001	0.0000767475217369971	8328
ENSMUSG00000000028	0.00012697851239282	13922
ENSMUSG00000000056	0.00012622539994289	13819
ENSMUSG00000000058	0.0000279690815920451	3407
ENSMUSG00000000078	0.0000147386904941405	1840
ENSMUSG00000000085	0.0000129937890229985	1603
ENSMUSG00000000088	0.0000461419591786146	5360
ENSMUSG00000000093	0.0000156152342218665	1965
ENSMUSG00000000120	0.000111227905305684	12009
ENSMUSG00000000126	0.000106196106529425	11473

Showing 1 to 10 of 14,896 entries Previous 1 2 3 4 5 … 1,490 Next

Figure 10: *Differential Expression Analysis with SEA*: genes ranking from the SEA.

In addition to the SEA, the tool provides a comparative table of the gene ranking between the DEG lists from the selected normalization methods and the SEA gene list, as shown in Figure 11. Furthermore, a table with a 'Gene score' is displayed, where the score represents the frequency with which a specific gene appears among the top n genes across multiple gene lists derived from the selected normalization methods (Figure 12). Finally, a Venn diagram is included at the end of the *Differential Expression Analysis with SEA* section, offering a visual comparison between the SEA results and the different DEG lists.

	Median	PoissonSeq	RLE	TMM	Upper Quartile	Shannon
	All	All	All	All	All	All
ENSMUSG00000000001	8380	8383	8284	8287	8294	8328
ENSMUSG00000000028	13935	13979	13829	13849	13897	13922
ENSMUSG00000000056	13754	13756	13869	13813	13909	13819
ENSMUSG00000000058	3479	3494	3380	3320	3376	3407
ENSMUSG00000000078	1881	1873	1811	1816	1815	1840
ENSMUSG00000000085	1583	1586	1619	1621	1627	1603
ENSMUSG00000000088	5354	5348	5361	5388	5384	5360
ENSMUSG00000000093	1723	1723	2088	2084	2084	1965
ENSMUSG00000000120	12588	12631	11587	11587	11660	12009
ENSMUSG00000000126	11270	11352	11573	11575	11653	11473

Showing 1 to 10 of 14,896 entries Previous 1 2 3 4 5 … 1,490 Next

Figure 11: *Differential Expression Analysis with SEA*: comparative table of the genes ranking.

	Gene	n	score
	All	All	All
1	ENSMUSG00000021268	5	1
2	ENSMUSG00000022425	5	1
3	ENSMUSG00000092274	5	1
4	ENSMUSG00000033826	5	1
5	ENSMUSG00000050520	5	1
6	ENSMUSG00000039116	5	1
7	ENSMUSG00000050069	5	1
8	ENSMUSG00000027347	5	1
9	ENSMUSG00000040046	5	1
10	ENSMUSG00000042371	5	1

Showing 1 to 10 of 20 entries Previous 1 2 Next

Figure 12: *Differential Expression Analysis with SEA*: 'Gene score'.

The source code for the SEA can be seen between the script lines 9545 and 10144.

The theory behind Shannon entropy can be further studied in [22–24].

3.3 Volcano Plots

Volcano plots represent a useful way to visualize the results of DGE analysis. It is a type of scatter plot that shows statistical significance (adjusted p-value) versus magnitude of change (fold change), where each dot represents a gene. Volcano plots display the statistical significance of the difference relative to the magnitude of difference for every single gene in the comparison, usually through the negative base-10 log adjusted p-value and base-2 log fold-change, respectively. Therefore, it is possible to quickly identify genes with large fold changes that are also statistically significant.

In this plot, the most statistically significant genes are towards the top, as they have the lowest adjusted p-values. If the expression of a certain gene is higher in group B compared to group A, the fold change will be positive. This means that the most up-regulated genes will be towards the right, and have a positive fold change. On the other hand, if the expression of a certain gene is higher in group A, when compared to group B, the fold change will be negative. The most down-regulated genes are towards the left, and have a negative fold change. This means they have a lower expression in group B when compared to group A. Furthermore, if the fold change is equal to zero, the expression does not change between one group and another. Lastly, a wider dispersion indicates two treatment groups that have a higher level of difference regarding the gene expression.

Fold change: the fold change of a gene is the ratio between the gene expression of the two groups. For a given comparison, a positive fold change value indicates an increase of expression, while a negative fold change indicates a decrease in expression. For example, a fold change of 1.5 for a specific gene in the ***Knockout*** vs. ***Control*** means that the expression of that gene is increased in ***Knockout*** relatively to ***Control*** by a multiplicative factor of $2^{1.5} \approx 2.82$.

P-value: this value indicates whether the gene analyzed is statistically likely to be differentially expressed in that comparison. This applies to each gene individually, assuming that the gene was tested on its own without consideration that all other genes were also tested.

Adjusted p-value: the p-value obtained for each gene above is re-calculated to correct for running many statistical tests (as many as the number of genes). In the result, we can say that all genes with adjusted p-values lower than a specific significance level (α) are significantly differentially expressed in the compared samples.

In the GeneSEA Explorer, after submitting the data, the first step to start the analysis is the *Differential Gene Expression Analysis* section, where, in a first moment, volcano plots provided by nine different normalization methods will be loaded.

Regarding the demo dataset example, the comparative analysis ***Knockout*** vs. ***Control***, it is possible to retrieve some of the following volcano plots, depicted in Figure 13.

The analysis from the demo dataset clearly demonstrates that the outcomes for the DEGs are dependent on the normalization method selected and, therefore, they highlight the importance of a thoughtful normalization method selection. Even thought the number of genes identified as statistically differentially expressed are similar across the normalization methods, when we proceed to identify the list, in the *Differential Expression Analysis: Results* subsection, we ascertain that the lists differ considerably, as shown in Figure 14. Therefore, these results corroborate the importance of the need to extract more information from each one of the differential expression results from each one of the normalization methods.

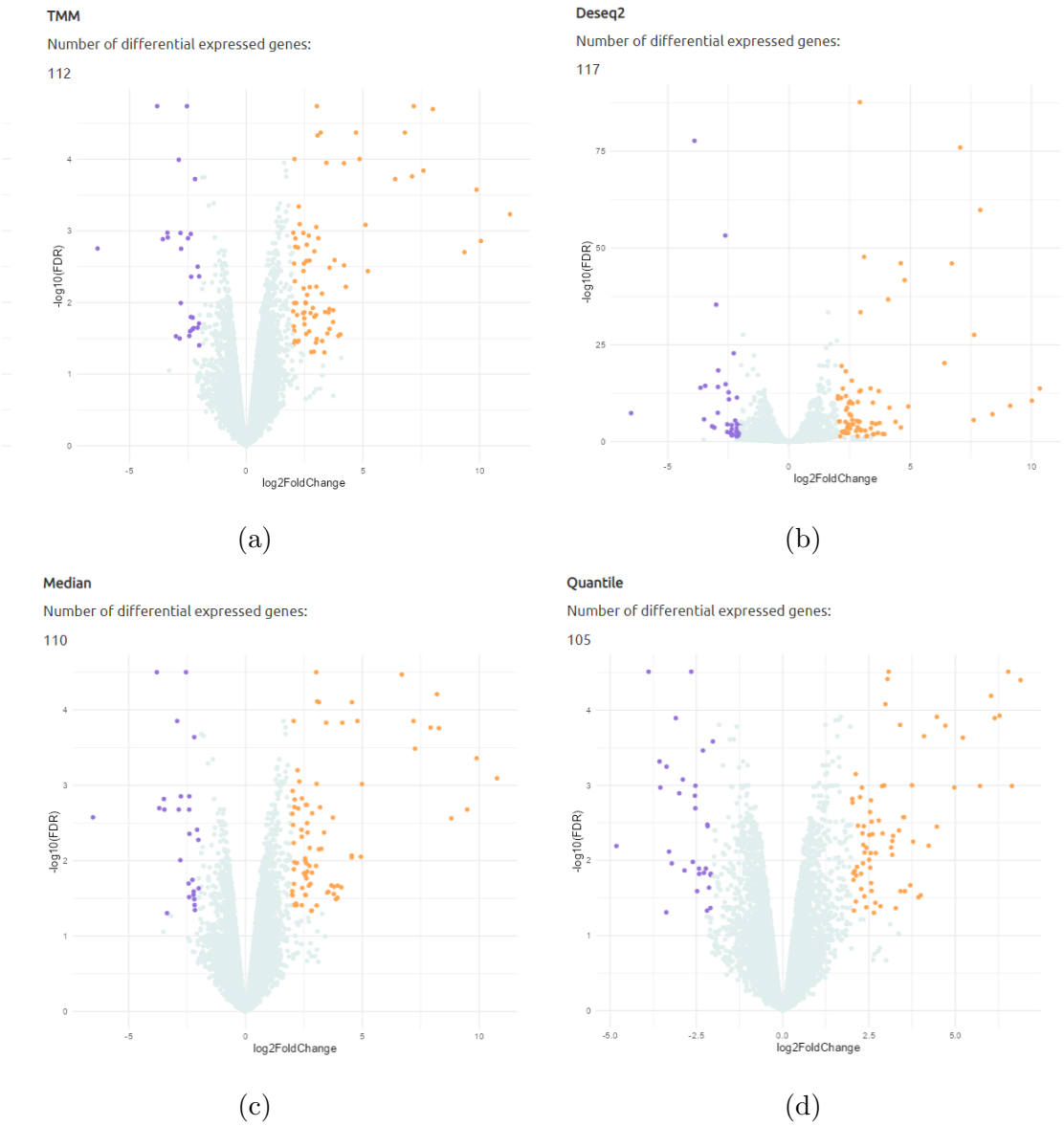


Figure 13: Volcano plots retrieved from the GeneSEA Explorer, *Differential Expression Analysis: Volcano plots*. Results from the demo dataset.

Current normalization method: TMMwsp

Show 10 entries Search:

	DESeq2	Median	No.Normalization	PoissonSeq	Quantile	RLE	TMM	Upper.Quartile
Up-regulated genes	79	84	86	83	74	84	85	82
Down-regulated genes	38	26	27	26	31	27	27	27
Equal DE genes	101	108	112	106	98	111	112	109
Different DE genes	11	4	0	6	14	1	0	3

Showing 1 to 4 of 4 entries Previous 1 Next

Figure 14: Illustration of the number of up- and down-regulated DEG for each normalization method, along with the number of genes from the DEG list that differ between the current normalization method (TMMwsp) and the others.

3.4 Multidimensional Scaling

The RNA-Seq data can be further explored by generating multidimensional scaling (MDS) plots. This plot allows the user to visualize the differences between the gene expression profiles of different sample groups and/or sample replicates in two dimensions, as shown in Figure 15.

Distances on an MDS plot of a DGEList object (edgeR object) correspond to leading log-fold-change between each pair of samples, i.e. the root-mean-square average of the largest log2-fold-changes between each pair of samples [8]. The same interpretation can be made with the MDS plot retrieved from the DESeq2 workflow. Each pair of samples extracted at each time tend to cluster together, suggesting a batch effect.

Replicate samples from the same group cluster together in the plot, while samples from different groups define separate clusters. This indicates that the differences between groups are larger than those within groups. This means that the differential expression is greater than the variance and can be detected.

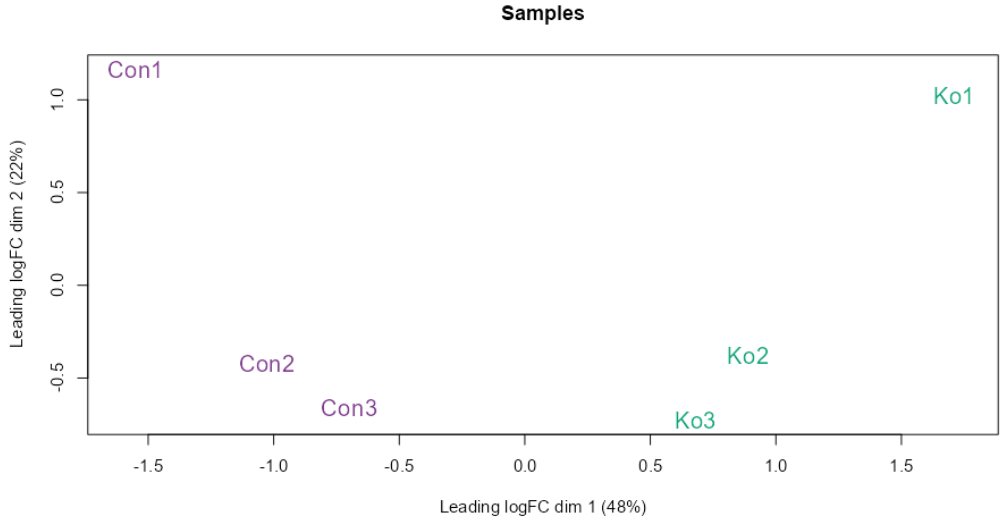


Figure 15: MDS plot retrieved from the demo dataset, normalized by TMMwsp. Shows the relative similarities of the six samples.

The MDS plot from the demo dataset (Figure 15) depicts the ***Control*** and ***Knockout*** samples are well separated in the first dimension, whereas the Con1 and Ko1 samples are separated in the second dimension relatively to the samples replicates 2 and 3. The distance between ***Control*** samples on the left and ***Knockout*** on the right is about 1 unit, corresponding to a leading fold change of about 2-fold ($2^1 = 2$) between ***Control*** and ***Knockout***. The expression differences within the ***Control*** and ***Knockout*** samples are similar.

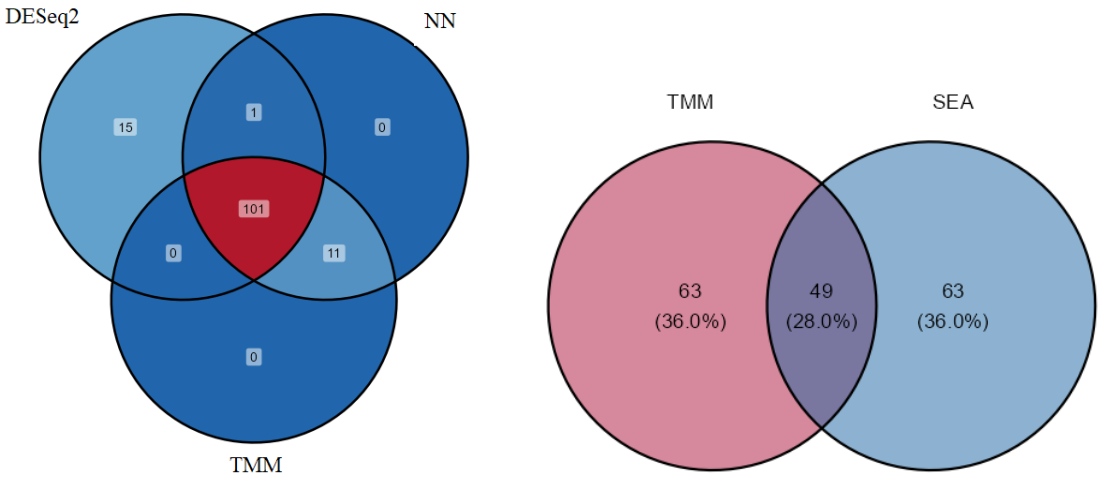
3.5 Venn Diagrams

The Venn diagram plot makes it possible to compare the overlap of differentially expressed features, namely the number of genes, in two or more statistical comparisons. The genes considered to be differentially expressed for each normalization method are shown as counts in the Venn diagrams, as shown in Figure 16.

The Venn diagram, available in the GeneSEA Explorer, in the *Differential Gene Expression Analysis: Results* section (depicted in Figure 16a), exhibits the results from the DEG

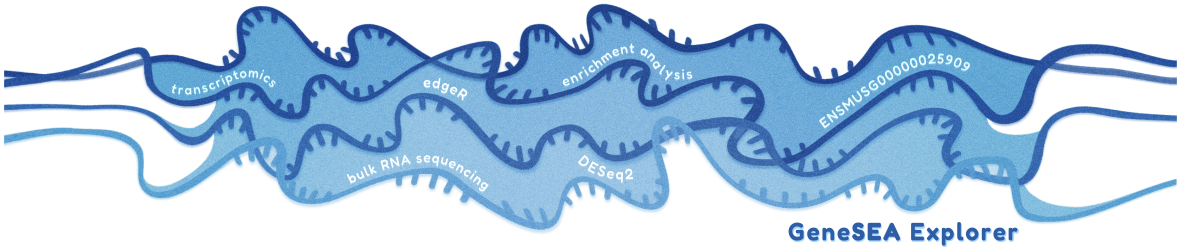
lists from the normalization methods selected by the user in the Shiny application interface. A maximum of seven DEG lists can be selected. Here, it is possible to observe how many genes from each DEG list differ between each other. This plot allows the users to quickly understand if the normalization methods provide or not DEG lists with discrepancies.

Regarding the Venn diagram shown in the *Differential Gene Expression Analysis with SEA* subsection, the differences between the DEG list acquired from a specific normalization method (selected by the user) and the DEG list obtained through SEA can be observed, as shown in Figure 16b. While the DEG list from the selected normalization method accounts for both the log2-fold-change and the adjusted p-value thresholds specified by the user (in the *Results*: *Differential Gene Expression Analysis* subsection), the SEA gene list only considers the complete list of adjusted p-value for each gene (as provided by the DEG results from the edgeR workflow) to acquire the genes ranking. Since there is no fixed threshold for the SEA, the Venn diagram compares the top genes from the SEA gene list, equivalent to the number of DEGs identified by the specific normalization method (i.e., the number of genes to be compared between the methods is the same).



(a) (b)

Figure 16: Venn diagram plots retrieved from the GeneSEA Explorer, *Differential Gene Expression Analysis:* (a) *Results* and (b) *Differential Gene Expression Analysis with SEA*. Results retrieved from the demo dataset.



Functional Enrichment Analysis

In omics analysis, researchers obtain data on various biological entities, such as genes, with the primary objective of investigating their functional significance and behavior in living organisms. In many cases, the gene lists are challenging to interpret due to their large size and lack of useful annotations.

In the DGE analysis, our intention is typically to use the list of the DEGs to gain novel insights. Hence, after acquiring the lists, we proceed to do a functional enrichment analysis. Here, we want to investigate if the deletion of genes that are "real" differentially expressed or the entry of genes that are "real" non differentially expressed, in the DEGs list, affect the discovery of gene terms that are more significant and, therefore, give "false" insights to the researcher. To aid this analysis, GO [3,4] and KEGG [5,6,25] provide a comprehensive set of tools and databases specifically designed to facilitate the understanding of systems biology.

Functional annotation is an analytical technique commonly applied to different types of big data (e.g., sets of genes, transcripts or proteins) to infer associated biological functions. This type of analysis, which is currently gaining notable interest and relevance, relies on the existence of manually curated libraries that annotate and classify the data (that is the sets of genes, transcripts or proteins) on the basis of their function [26]. Grouping significant DEGs based on functional similarity can systematically enhance biological interpretation of large lists of genes derived from the high throughput studies [27].

The annotation enrichment analysis is an automated and statistically rigorous technique to analyze and interpret large gene lists using a priori-knowledge. Enrichment analysis assesses the over- (or under-) representation of a known set of genes (e.g., a biological pathway) within the input gene list. If a statistically significant number of genes from the known set are present in the gene list, it may indicate that the biological pathway plays a role in the biological condition under study. This analysis is repeated for all available known gene-sets [26].

4.1 Gene Ontology

The GO tool [3,4] provides a framework and set of concepts for describing the functions of gene products from all organisms, specifically designed for supporting the computational representation of biological systems.

The GO enrichment analysis is a common downstream procedure to interpret the differential expression results in a biological context. Given a set of genes that are up- or down-regulated under a certain contrast of interest, a GO enrichment analysis will find which GO terms are over- or under-represented using annotations for the genes in that set [3,4]. Therefore, the GO enrichment analysis compares the distribution of GO terms in the sample set (list of genes of interest, the DEGs) versus that observed in the reference set (genome): if a certain GO term is more frequent in the sample set than in the reference set, it is said to be enriched, indicating functional specificity [26].

Nowadays, there is a variety of R packages that aid performing functional enrichment analyses (e.g., clusterProfiler, through the *enrichGO()* function). In the GeneSEA Explorer, the packages clustifyr, org.Hs.eg.db, org.Mm.eg.db, org.Rn.eg.db, clusterProfiler, AnnotationDbi, accessible in [28], are needed to perform the GO enrichment analysis. Bellow it is shown the code to perform the functional enrichment analysis.

```
Go_results <- enrichGO(gene = genes_to_test, OrgDb = "org.Mm.eg.db",
keyType = "ENSEMBL", ont = "BP")
```

For the GO enrichment analysis, the *enrichGO* function and a list of mouse genes from 'org.Mm.eg.db' (in the case of the demo dataset) are used to perform the hypergeometric test, which is corrected by the Benjamini-Hochberg method.

GO Terms

An ontology consists of a structured set of well-defined terms and relationships, representing the current understanding of biological knowledge. Data can be annotated at different levels depending on the amount and completeness of the available information, allowing flexibility to focus on broader or more specific aspects of the analysis [3]. The three GO terms, required for selection in the Shiny application platform, defined in the code as the following,

```
ont = c("BP","MF","CC"),
```

are defined as:

1. **biological process (BP)** which refers to a biological function to which the gene or gene product contributes. A process is accomplished via one or more ordered assemblies of molecular functions. Examples of broad biological process terms are "Deoxyribonucleic acid (DNA) repair" or "signal transduction" [3, 4];
2. **molecular function (MF)** which is defined as the biochemical activity, including specific binding to ligands or structures, of a gene product (i.e. a protein or RNA). Molecular functions generally correspond to activities that can be performed by individual gene products, but some activities are performed by molecular complexes composed of multiple gene products. Examples of broad functional terms are "catalytic activity" and "transporter activity". This term describes only what is done without specifying where or when the event actually occurs [3, 4];
3. **cellular component (CC)** which refers to the location, relative to cellular compartments and structures, where a gene product is active. These terms reflect our understanding of eukaryotic cell structure [3, 4]. There are two ways in which the gene ontology describes locations of gene products: (1) the cellular anatomical entities, in which a gene product carries out a molecular function, and (2) the stable macromolecular complexes of which they are parts [4].

   Not all terms are applicable to all organisms; the set of terms is meant to be inclusive.

GO Annotations

Most enrichment tools derive gene-sets from GO annotations, because they are readily accessible for many organisms and cover many genes, yet many other sources of gene-sets exist and are used by some tools in addition to GO.

In the *enrichGO()* function, the key *OrgDb* defines in which database the organism annotation is present in the dataset:

- org.Mm.eg.db: genome wide annotation for mouse.
- org.Rn.eg.db: genome wide annotation for rat.
- org.Hs.eg.db: genome wide annotation for human.

4.2 Kyoto Encyclopedia of Genes and Genomes

KEGG is a database resource for understanding high-level functions and utilities of the biological system, such as the cell, the organism and the ecosystem, from molecular-level information, especially large-scale molecular datasets generated by genome sequencing and other high-throughput experimental technologies [5, 6, 25].

In opposition to GO, the KEGG pathway database provides functional annotation as well as information about gene products that interact with each other in a given pathway, how they interact (e.g., activation or inhibition), and where they interact (for example, in the cytoplasm). Hence, KEGG has the potential to provide extra insight beyond annotation lists from GO terms.

Relatively to KEGG, there is also a variety of R packages that aid performing the KEGG enrichment analysis of a gene set. Given a vector of genes, this function will return the enrichment KEGG categories with FDR control. In the GeneSEA Explorer, the packages clusterProfiler (e.g., the *enrichKEGG()* function) and AnnotationDbi, accessible in [28], are needed to perform the KEGG enrichment analysis. Bellow it is shown the code to perform the functional enrichment analysis.

```
kk1 <- enrichKEGG(gene = names(kegg), organism=KEGGorg(),
          pvalueCutoff = 0.05, keyType = KEGGterms())
```

KEGG Annotation

In the *enrichKEGG()* function, the key *organism* defines which organism annotation is present in the dataset:

- mmu: reference genome for Mus musculus (house mouse).
- hsa: reference genome for Homo sapiens (human).

KEGG GENES Database

KEGG GENES is a collection of genes and proteins in complete genomes of cellular organisms and viruses generated from publicly available resources and annotated by KEGG in the form of KEGG Orthology (KO). Each GENES entry is identified by the combination of organism code and gene identifier in the form of org:gene such as, for example, hsa:250 for human alkaline phosphatase gene.

The public databases present in the GeneSEA Explorer, defined in the function *keyType = KEGGterms()*, are

- kegg: KO database – a database of molecular functions represented in terms of functional orthology;
- ncbi-geneid: gene database from National Center for Biotechnology Information [29];
- ncib-proteinid: protein database from National Center for Biotechnology Information [29];
- uniprot: UniProt databases [30].

4.3 Over-Representation Analysis

ORA uses hypergeometric test to calculate p-values of the observed number of genes in one gene set versus the expected number of genes in that set from the reference genome. ORA is used to determine which a priori defined gene sets are more present (over-represented) in a subset of "interesting" genes than what would be expected by chance. The FDR value shown in the analysis output is the p-value corrected for multiple testing with Benjamini-Hochberg method [31].

Due to its simplicity, well-established statistical framework, and easy implementation, ORA is widely available across numerous tools. However, ORA works under the assumptions that genes work independently and contribute equally to biological processes. While these assumptions simplify the modeling, they do not accurately reflect the biological reality, where genes, proteins, and other biomolecules often work together. Furthermore, ORA only considers the DEG list, typically derived from log2-fold-change and adjusted p-value thresholds, while disregarding quantitative information from other genes. This limitation overlooks the potential contributions of genes with expression changes just above (in terms of the adjusted p-value) or bellow (in terms of the log2-fold-change) the cutoff value, which may still influence pathway activity [31].

ORA with Shannon Aggregation

Given that the ORA method only uses the DEG list, new insights can be brought while using the SEA. Therefore, the proposed method is performed within the ORA.

Since no threshold is applied to the SEA gene list, the number of genes selected for the enrichment analysis matches the number of genes in the DEG list from the chosen normalization method, as specified in the *Functional Enrichment Analysis* section.

The enriched terms from the different analysis can be compared simultaneously, as both results are displayed concurrently in the interface.

4.4 Gene Set Enrichment Analysis

Although DGE analysis of RNA-Seq data has become a routine tool in biomedical research, extracting biological insight, that is, interpreting the results from the biological mechanisms remains a major challenge. The GSEA is an analytical tool for interpreting gene expression data, proposed in [32], to evaluate data at the level of gene sets and to overcome the interpretation challenges. A goal of GSEA is to provide a more robust way to compare independently derived gene expression data sets and obtain more consistent results [32].

GSEA is a rank-based approach that determines whether predefined groups of genes, proteins, or other biomolecules, are primarily up- or down-regulated in one condition relative

to another. Nowadays, GSEA is typically performed as a follow-up to DGE analysis, being preferred to ORA.

In the GSEA method, in a first moment the genes are ordered in a ranked list L, according to their differential expression results. The goal of GSEA is to determine whether members of a gene set S 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. Furthermore, rather than focus on high scoring genes (which can be poorly annotated and may not be reproducible), researchers can focus on gene sets, which tend to be more reproducible and more interpretative [32].

GSEA with Shannon Aggregation

Given that the GSEA method uses the complete list of genes (the complete list of genes after the RNA-Seq count matrix filtering), no new insights would be brought while using the SEA method. Therefore, the proposed method is not performed within the GSEA.

4.5 Functional Enrichment Analysis Outputs

clusterProfiler [7] supports enrichment analysis of GO and KEGG with either hypergeometric test (ORA) or GSEA [32]. clusterProfiler adjust the estimated significance level to account for multiple hypothesis testing and also q-values were calculated for FDR control. It supports several visualization methods, including barplot and cnetplot. The enrichplot package [33] is also implemented in the application for interpreting functional enrichment results obtained from ORA and GSEA analysis.

Visualization

The following examples were retrieved from the demo dataset. Here, the selected options for the GO where the following

- GO terms: BP;
- Normalization method: TMM;
- OrgDb terms: org.Mm.eg.db;
- keyType: ENSEMBL;
- Number of Categories to be shown: 10.

The following analysis provides insights into potential biological process affected in the ***Knockout*** samples. Enrichment significance was determined using a hypergeometric test with p-values adjusted by the Benjamini-Hochberg method (cutoff < 0.05). All DEGs were considered for this analysis. Furthermore, some insights of possible errors that may occur in the analysis are presented. In this manual, no examples where retrieved from the KEGG enrichment analysis.

Barplot

In Figure 17, both the results of the statistical test (adjusted p-value) and the number of genes (indicated by the "Count" variable on the x-axis, represented by the bar size) from the DEG list associated with each GO term (y-axis) are shown.

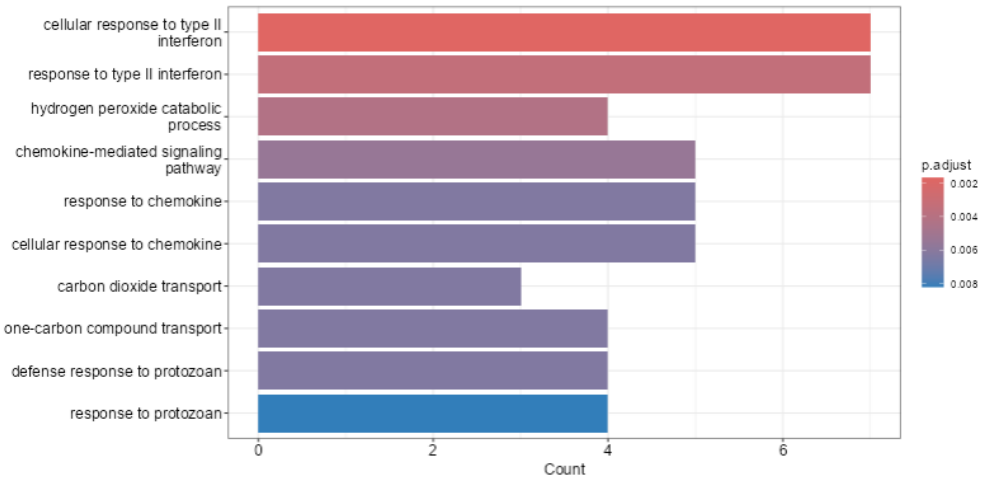


Figure 17: ORA output from the demo dataset: barplot.

Error: replacement has length zero

If the aforementioned warning shows up in the application, that means no enriched term was found in the dataset resulting in a null output.

Dotplot

In this example (Figure 18), the results of the statistical test are displayed, along with the number of genes (i.e., the "Count" variable, expressed as the point size) from the DEG list that are associated to each one of the GO terms (y-axis). Additionally, the GeneRatio (x-axis), representing the fraction of DEGs within each gene set, is also shown.

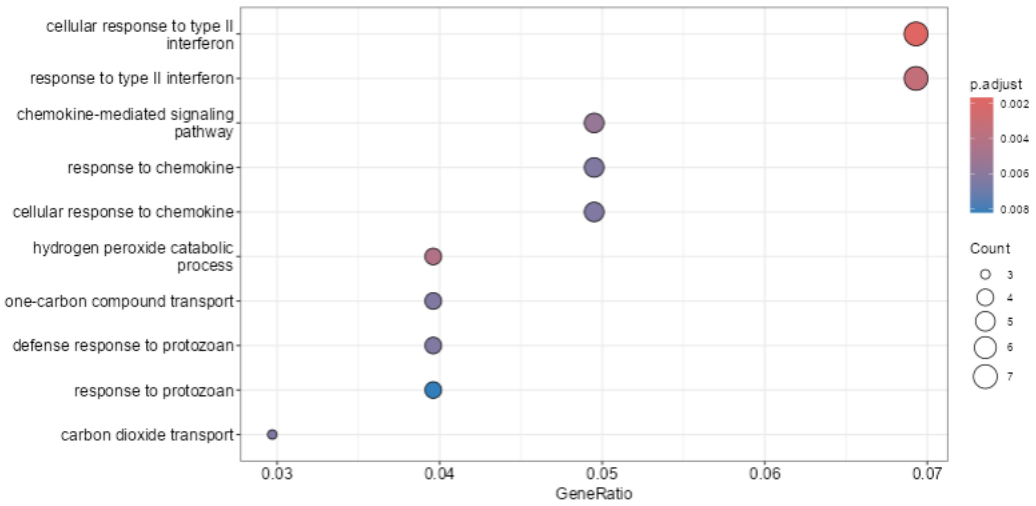


Figure 18: ORA output from the demo dataset: dotplot.

Error: replacement has length zero

If the aforementioned warning shows up in the application, that means no enriched term was found in the dataset resulting in a null output.

Heatmap

Both barplot and dotplot graphs display only the most significant or selected enriched terms. However, users may be interested in identifying the specific genes involved in these significant terms. The heatmap is one potential option for providing this information. In Figure 19, the results of the statistical test are shown, along with the genes from the DEG list that are associated with each GO terms.

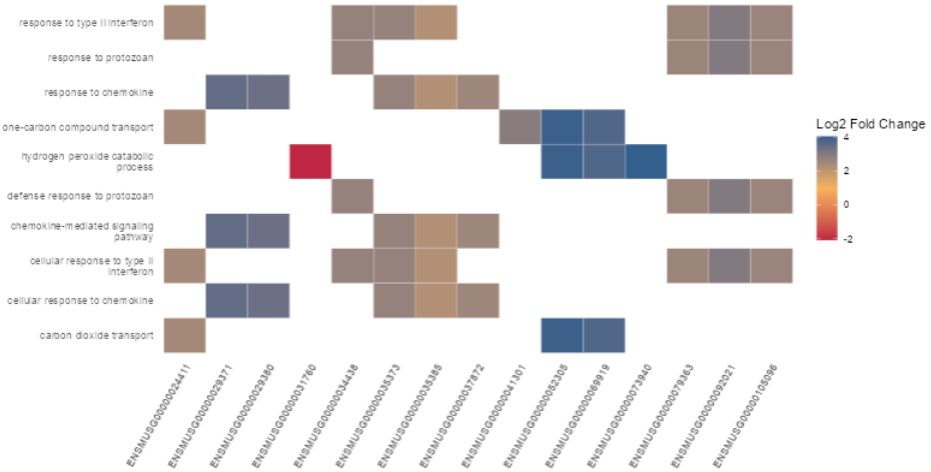


Figure 19: ORA output from the demo dataset: heatmap.

Error: undefined columns selected

If this warning shows up in the application, that means no enriched term was found in the dataset results and, therefore, no output can be displayed.

Emapplot

In the following figure, both the results of the statistical test and the number of genes (represented by the "number of genes" variable and indicated by point size) from the DEG list associated with each GO term are displayed, with the corresponding GO term shown adjacent to each point.

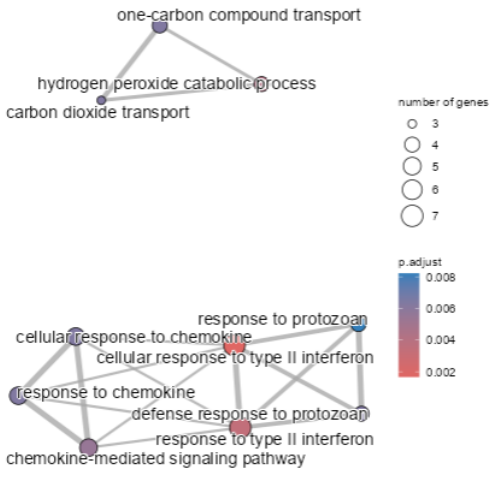


Figure 20: ORA output from the demo dataset: emapplot.

Error: error in evaluating the argument 'x' in selecting a method for function 'emapplot': no enriched term found...

If the current warning shows up in the application, that means no enriched term was found in the dataset results and, therefore, no plot can be produced given that the GSEA or ORA output is null.

Cnetplot

To account for the potentially biological complexities where a gene may belong to multiple annotation categories and to provide information on numeric changes when available [7, 34], a cnetplot can be performed.

The cnetplot allows us to visualize the relationships and overlap of genes associated with the top enriched terms in a network diagram. This approach effective highlights genes that are relevant for multiple enriched terms, helping to extract complex associations [7, 34].

Similarly to the emapplot, the cnetplot also allows the visualization of statistical test results, where the point size represents the number of genes from the DEG list associated with each GO term, with the corresponding term displayed next to the point.

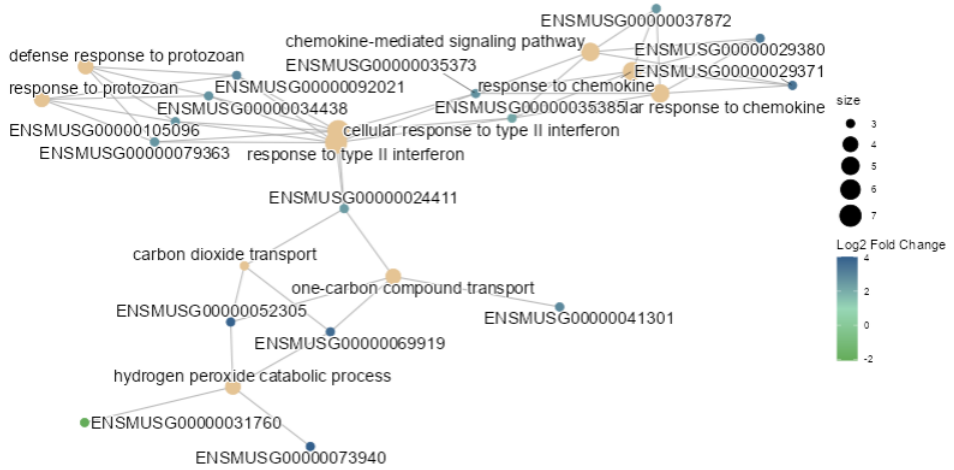


Figure 21: ORA output from the demo dataset: cnetplot.

Error: the data frame should contain at least two columns

If this warning shows up in the application, that means no enriched term was found in the dataset results and, therefore, no output can be displayed.

Upset plot

The upset plot serves as an alternative to the cnetplot for visualizing complex associations between genes and gene sets. It highlights the overlap of genes across different gene sets and provides the number of genes associated with each individual term or combinations of multiple terms.

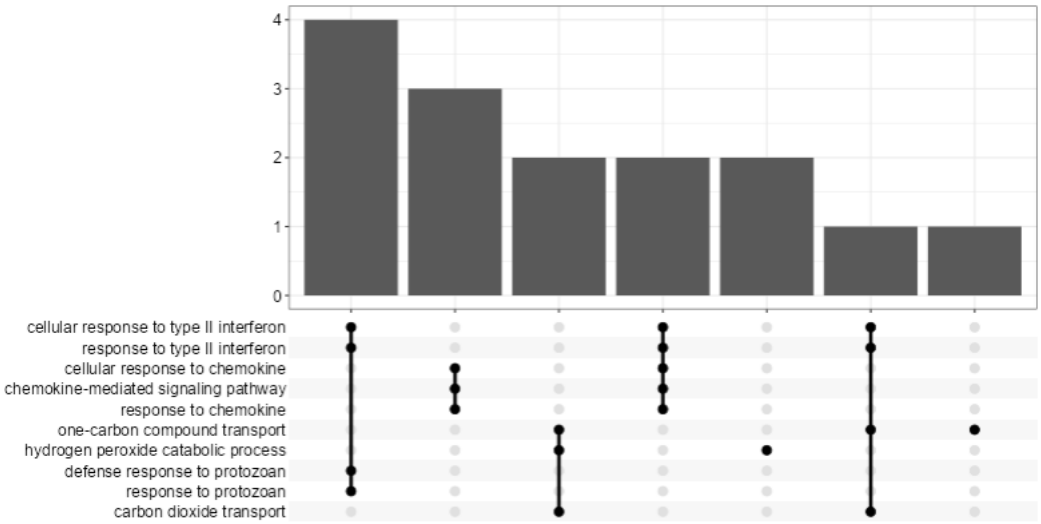


Figure 22: ORA output from the demo dataset: upset plot.

Error: undefined columns selected

If this warning shows up in the application, that means no enriched term was found in the dataset results and, therefore, no output can be displayed.

Treeplot

In the treeplot graph, gene sets are visualized as a hierarchical tree. Gene sets with high similarity are clustered together, making interpretation easier. The *treeplot()* function performs hierarchical clustering of enriched terms based on the pairwise similarities calculated by the *pairwise_termsim()* function.

Similar to the emapplot, the treeplot allows visualization of the statistical test results, where the point size indicates the number of genes from the DEG list associated with each GO term, with the corresponding GO term displayed next to each point.

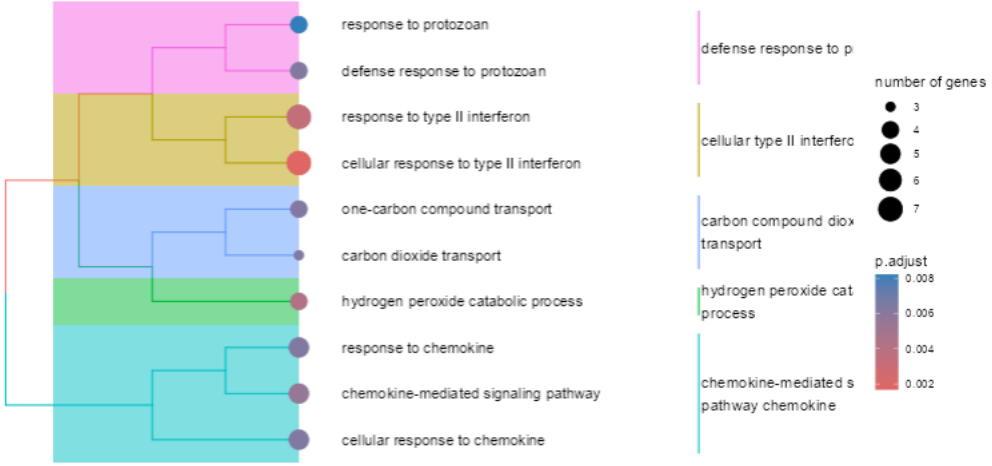
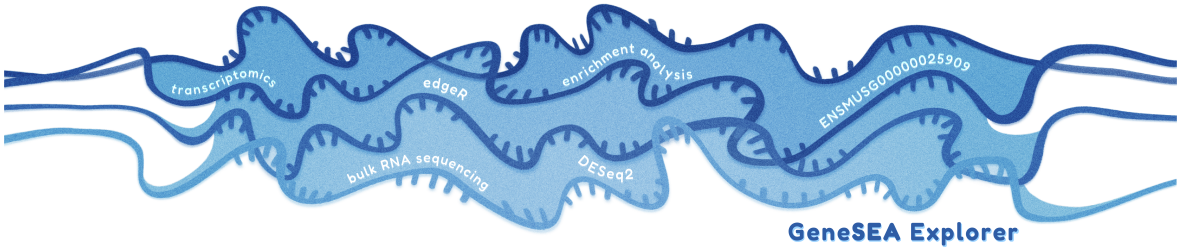


Figure 23: ORA output from the demo dataset: treeplot.

Error: elements of 'k' must be between 1 and 2

Error: error in evaluating the argument 'x' in selecting a method for function 'treeplot': no enriched term found...

If the warnings aforementioned show up in the application, that means no enriched term was found in the dataset results and, therefore, no output can be displayed.



GeneSEA Explorer Advantages

The GeneSEA Explorer bioinformatics tool offers several key advantages, making it a valuable resource for RNA-Seq data analysis, namely:

1. user-friendly interface: GeneSEA Explorer is an Shiny web application with an intuitive interface, enabling users to easily explore and interpret their RNA-Seq data;
2. easy accessibility: The tool is available in an open repository, providing free access to researchers. Both the source code and a detailed user manual are included, ensuring straightforward usage;
3. adaptable source code: Despite its complexity, with over 10 000 lines of code, the GeneSEA Explorer's source code can be adapted by experienced programmers to suit specific analysis requirements;
4. comprehensive analysis: GeneSEA Explorer supports both DGE and functional enrichment analyses, allowing a complete analysis and interpretation of the researchers RNA-Seq data;
5. addressing variability in DEG lists: The tool tackles the challenges of DEG list variability by employing SEA to aggregate results from different normalization methods. This allows users to delve deeper into their data and be more confident with the DEGs list;
6. future development potential: The proposed aggregation method is an initial step toward addressing the challenges in RNA-Seq analysis. Ongoing development, validation and optimization are essential to further refine the tool, making GeneSEA Explorer a beginning of future advancements.

GeneSEA Explorer Limitations

The GeneSEA Explorer pipeline is designed to integrate functionality, reproducibility, and require minimal programming skills to the users. Despite its designed efficiencies, the pipeline has certain limitations that potential users need to consider.

Firstly, the pipeline present customization limitations for advanced users. Those seeking to tailor de pipeline for specific needs or to incorporate particular modifications may find themselves needing to adapt the dataset. For example, if the users want to consider more than one condition in the study design there will be the need to create a column in the metadata which combines the variables of interest, such as the example shown in Table 2, Condition 3 column, or change the contrast vector directly in the code script.

Secondly, a significant limitation is the absence of batch correction. The pipeline assumes that all samples used are directly comparable, either originating from identical laboratory

conditions or produced in similar environments. A solution to contradict this assumption may be by adding a variable designed *Batch* in the contrast (as shown in Table 2, Condition 4 column).

Table 2: Examples of possible modifications in the metadata.

	Condition 1	Condition 2	Condition 3	Condition 4
	Group	Time Point	Combined Variable	Batch
Ko1	Knockout	Pre-clinical	Knockout_Pre-clinical	1
Con1	Control	At-diagnosis	Control_At-diagnosis	2
...	...	...	...	...
Con3	Control	Pre-clinical	Control_Pre-clinical	2

Additionally, the genes ranking acquired from the proposed aggregation method, SEA, are influenced by the chosen normalization methods. The default normalization methods specified in the *Differential Expression Analysis: Differential Expression Analysis with SEA* subsection – namely the Median, PoissonSeq, RLE, TMM and Upper Quartile – are selected to minimize potential bias during the analysis. For instance, selecting both TMM and TMMwsp, which employ similar normalization methodologies, in the same analysis could introduce bias into the SEA results.

Furthermore, the functional enrichment analysis pipeline is inherently tied to the size of the DEG list it generates. Smaller lists may fail to obtain any enrichment results and, therefore, message errors might show up.

Final Remarks

Author contributions

Ana M. Gonçalves was the main developer, with support from the remaining authors. All authors revised the user's guide.

Competing interests/Conflicts of interest

The authors declare no conflict of interests. No competing interests were disclosed.

Acknowledgments and Funding

This work was supported by National funds, through the Portuguese Foundation for Science and Technology (FCT) - project UIDB/50026/2020 (https://doi.org/10.54499/UIDB/50026/2020), UIDP/50026/2020 (https://doi.org/10.54499/UIDP/50026/2020), LA/P/0050/2020(https://doi.org/10.54499/LA/P/0050/2020) and 10.54499/CEECINST/00018/2021/CP2806/CT0011 (https://doi.org/10.54499/CEECINST/00018/2021/CP2806/CT0011). The authors acknowledge the support by CIDMA (Center for Research and Development in Mathematics and Applications) under the FCT (Portuguese Foundation for Science and Technology) Multi-Annual Financing Program for R&D Units (Reference UID/04106).

The authors would like to thank the researchers who provided the original demo data implemented in the application as open-source data, published in GEO.

A special thank you to Catarina Lourenço who designed our logo.
Go to Catarina Lourenço's Behance account for more information.

R session

R Version

GeneSEA Explorer currently supports version 4.3.3 and up to the current the R version 4.5.0.

Packages Used

The GeneSEA Explorer depends on various packages from version 3.19 of the Bioconductor project [28] and many others, running on R version 3.3.0 or higher. The complete list of the packages used in the GeneSEA Explorer workflow are shown below.

```
> sessionInfo()
R version 4.3.3 (2024-02-29 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19045)

Matrix products: default

locale:
[1] LC_COLLATE=Portuguese_Portugal.utf8
[2] LC_CTYPE=Portuguese_Portugal.utf8
[3] LC_MONETARY=Portuguese_Portugal.utf8
```

```
[4] LC_NUMERIC=C
[5] LC_TIME=Portuguese_Portugal.utf8

time zone: Europe/Lisbon
tzcode source: internal

attached base packages:
[1] grid      stats4    stats     graphics
[5] grDevices utils     datasets  methods
[9] base

other attached packages:

[1] extrafont_0.19              [26] org.Mm.eg.db_3.18.0       [51] forcats_1.0.0
[2] RColorBrewer_1.1-3          [27] org.Hs.eg.db_3.18.0       [52] stringr_1.5.1
[3] ggvenn_0.1.10               [28] clustifyr_1.14.0          [53] purrr_1.0.2
[4] ggVennDiagram_1.5.2         [29] gprofiler2_0.2.3          [54] tidyr_1.3.1
[5] genekitr_1.2.5              [30] AnnotationDbi_1.64.1      [55] tibble_3.2.1
[6] ggbump_0.1.0                [31] clusterProfiler_4.10.1    [56] tidyverse_2.0.0
[7] GGally_2.2.1                [32] BiocManager_1.30.22       [57] bslib_0.6.2
[8] shinythemes_1.2.0           [33] shinyBS_0.61.1            [58] readr_2.1.5
[9] shinyWidgets_0.8.3          [34] Glimma_2.12.0             [59] readxl_1.4.3
[10] R.utils_2.12.3             [35] paletteer_1.6.0           [60] DESeq2_1.42.1
[11] R.oo_1.26.0                [36] wesanderson_0.3.7         [61] SummarizedExperiment_1.32.0
[12] R.methodsS3_1.8.2          [37] densityClust_0.3.3        [62] Biobase_2.62.0
[13] vidger_1.22.0              [38] inops_0.0.1               [63] MatrixGenerics_1.14.0
[14] ggupset_0.3.0              [39] affydata_1.50.0           [64] matrixStats_1.2.0
[15] topGO_2.54.0               [40] affy_1.80.0               [65] GenomicRanges_1.54.1
[16] SparseM_1.82               [41] sva_3.50.0                [66] GenomeInfoDb_1.38.8
[17] GO.db_3.18.0               [42] BiocParallel_1.36.0       [67] IRanges_2.36.0
[18] graph_1.80.0               [43] genefilter_1.84.0         [68] S4Vectors_0.40.2
[19] pathview_1.42.0            [44] mgcv_1.9-1                [69] BiocGenerics_0.48.1
[20] GOSemSim_2.28.1            [45] nlme_3.1-164              [70] dplyr_1.1.4
[21] enrichplot_1.22.0          [46] edgeR_4.0.16              [71] plotly_4.10.4
[22] DEGreport_1.38.5           [47] limma_3.58.1              [72] DT_0.32
[23] preprocessCore_1.64.0      [48] DOSE_3.28.2               [73] ggplot2_3.5.0
[24] org.Mmu.eg.db_3.18.0       [49] waiter_0.2.5              [74] shiny_1.8.1
[25] org.Rn.eg.db_3.18.0        [50] lubridate_1.9.3

loaded via a namespace (and not attached):

[1] fs_1.6.3                    [20] scatterpie_0.2.2          [39] fansi_1.0.6
[2] bitops_1.0-7                [21] prismatic_1.1.2           [40] abind_1.4-5
[3] fontawesome_0.5.2           [22] labeling_0.4.3            [41] lifecycle_1.0.4
[4] HDO.db_0.99.1               [23] entropy_1.3.1             [42] yaml_2.3.8
[5] httr_1.4.7                  [24] sass_0.4.9                [43] qvalue_2.34.0
[6] doParallel_1.0.17           [25] KEGGgraph_1.62.0          [44] SparseArray_1.2.4
[7] Rgraphviz_2.46.0            [26] geneset_0.2.7             [45] Rtsne_0.17
[8] tools_4.3.3                 [27] systemfonts_1.0.6         [46] blob_1.2.4
[9] backports_1.4.1             [28] yulab.utils_0.1.4         [47] promises_1.2.1
[10] utf8_1.2.4                 [29] gson_0.1.0                [48] crayon_1.5.2
[11] R6_2.5.1                   [30] rstudioapi_0.16.0         [49] lattice_0.22-5
[12] lazyeval_0.2.2             [31] RSQLite_2.3.5             [50] cowplot_1.1.3
[13] GetoptLong_1.0.5           [32] FNN_1.1.4                 [51] PoissonSeq_1.1.2
[14] withr_3.0.0                [33] generics_0.1.3            [52] annotate_1.80.0
[15] prettyunits_1.2.0          [34] gridGraphics_0.5-1        [53] KEGGREST_1.42.0
[16] gridExtra_2.3              [35] shape_1.4.6.1             [54] pillar_1.9.0
[17] textshaping_0.3.7          [36] crosstalk_1.2.1           [55] knitr_1.47
[18] cli_3.6.2                  [37] zip_2.3.1                 [56] ComplexHeatmap_2.18.0
[19] logging_0.10-108           [38] Matrix_1.6-5              [57] fgsea_1.28.0
```

[58] rjson_0.2.21
[59] codetools_0.2-19
[60] fastmatch_1.1-4
[61] glue_1.7.0
[62] ggfun_0.1.5
[63] data.table_1.15.2
[64] urltools_1.7.3
[65] vctrs_0.6.5
[66] png_0.1-8
[67] treeio_1.26.0
[68] cellranger_1.1.0
[69] gtable_0.3.5
[70] rematch2_2.1.2
[71] cachem_1.0.8
[72] openxlsx_4.2.5.2
[73] xfun_0.44
[74] europepmc_0.4.3
[75] S4Arrays_1.2.1
[76] mime_0.12
[77] tidygraph_1.3.1
[78] ConsensusClusterPlus_1.66.0
[79] survival_3.5-8
[80] SingleCellExperiment_1.24.0
[81] iterators_1.0.14
[82] statmod_1.5.0
[83] ggtree_3.10.1
[84] bit64_4.0.5
[85] progress_1.2.3
[86] affyio_1.72.0
[87] colorspace_2.1-0
[88] DBI_1.2.3
[89] mnormt_2.1.1
[90] tidyselect_1.2.1
[91] extrafontdb_1.0
[92] bit_4.0.5
[93] compiler_4.3.3
[94] xml2_1.3.6
[95] ggdendro_0.2.0
[96] DelayedArray_0.28.0
[97] triebeard_0.4.1
[98] shadowtext_0.1.3
[99] scales_1.3.0
[100] psych_2.4.3
[101] digest_0.6.35
[102] rmarkdown_2.27
[103] XVector_0.42.0
[104] htmltools_0.5.8
[105] pkgconfig_2.0.3
[106] fastmap_1.1.1
[107] rlang_1.1.3
[108] GlobalOptions_0.1.2
[109] htmlwidgets_1.6.4
[110] farver_2.1.1
[111] jquerylib_0.1.4
[112] jsonlite_1.8.8
[113] RCurl_1.98-1.14
[114] magrittr_2.0.3
[115] GenomeInfoDbData_1.2.11
[116] ggplotify_0.1.2
[117] patchwork_1.2.0
[118] munsell_0.5.1
[119] Rcpp_1.0.12
[120] ggnewscale_0.4.10
[121] ape_5.7-1
[122] viridis_0.6.5
[123] stringi_1.8.3
[124] ggraph_2.2.1
[125] zlibbioc_1.48.2
[126] MASS_7.3-60.0.1
[127] plyr_1.8.9
[128] ggstats_0.6.0
[129] parallel_4.3.3
[130] ggrepel_0.9.5
[131] Biostrings_2.70.3
[132] graphlayouts_1.1.1
[133] splines_4.3.3
[134] hms_1.1.3
[135] circlize_0.4.16
[136] locfit_1.5-9.9
[137] igraph_2.0.3
[138] reshape2_1.4.4
[139] XML_3.99-0.16.1
[140] evaluate_0.23
[141] tzdb_0.4.0
[142] foreach_1.5.2
[143] tweenr_2.0.3
[144] httpuv_1.6.15
[145] Rttf2pt1_1.3.12
[146] polyclip_1.10-6
[147] reshape_0.8.9
[148] clue_0.3-65
[149] ggforce_0.4.2
[150] broom_1.0.6
[151] xtable_1.8-4
[152] tidytree_0.4.6
[153] later_1.3.2
[154] ragg_1.3.0
[155] viridisLite_0.4.2
[156] snow_0.4-4
[157] aplot_0.2.2
[158] memoise_2.0.1
[159] cluster_2.1.6
[160] timechange_0.3.0

References

[1] "National Center for Biotechnology Information, Gene Expression Omnibus (GEO), howpublished = https://www.ncbi.nlm.nih.gov/geo/, note = Accessed: 10/07/2024,."

[2] S. Su, C. W. Law *et al.*, "Glimma: interactive graphics for gene expression analysis," *Bioinformatics*, vol. 33, no. 13, pp. 2050–2052, 02 2017. [Online]. Available: https://doi.org/10.1093/bioinformatics/btx094

[3] M. Ashburner, C. Ball *et al.*, "Gene ontology: tool for the unification of biology. the gene ontology consortium," *Nature genetics*, vol. 25, pp. 25–9, 06 2000.

[4] "Gene Ontology, Unifying Biology : The Gene Ontology Resource," https://geneontology.org/, accessed: 16/07/2024.

[5] "KEGG: Kyoto Encyclopedia of Genes and Genomes," https://www.genome.jp/kegg/, accessed: 10/07/2024.

[6] M. Kanehisa, Y. Sato *et al.*, "KEGG as a reference resource for gene and protein annotation," *Nucleic Acids Research*, vol. 44, no. D1, pp. D457–D462, 10 2015. [Online]. Available: https://doi.org/10.1093/nar/gkv1070

[7] T. Wu, E. Hu *et al.*, "clusterprofiler 4.0: A universal enrichment tool for interpreting omics data," *The Innovation*, vol. 2, no. 3, p. 100141, 2021. [Online]. Available: https://doi.org/10.1016/j.xinn.2021.100141

[8] "edgeR User's Guide, [edgeR package version 3.42.4]," https://www.bioconductor.org/packages/devel/bioc/vignettes/edgeR/inst/doc/edgeRUsersGuide.pdf, accessed: 29/07/2024.

[9] A. Chawade, E. Alexandersson, and F. Levander, "Normalyzer: A tool for rapid evaluation of normalization methods for omics data sets," *Journal of Proteome Research*, vol. 13, pp. 3114–3120, 6 2014. [Online]. Available: https://doi.org/10.1021/pr401264n

[10] M.-A. Dillies, A. Rau *et al.*, "A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis," *Briefings in Bioinformatics*, vol. 14, no. 6, pp. 671–683, 09 2012. [Online]. Available: https://doi.org/10.1093/bib/bbs046

[11] T. Välikangas, T. Suomi, and L. L. Elo, "A systematic evaluation of normalization methods in quantitative label-free proteomics," *Briefings in Bioinformatics*, vol. 19, pp. 1–11, 1 2018. [Online]. Available: https://doi.org/10.1093/bib/bbw095

[12] P. R. Bushel, S. S. Ferguson *et al.*, "Comparison of normalization methods for analysis of tempo-seq targeted rna sequencing data," *Frontiers in Genetics*, vol. 11, 6 2020. [Online]. Available: https://doi.org/10.3389/fgene.2020.00594

[13] J. Li, D. M. Witten *et al.*, "Normalization, testing, and false discovery rate estimation for rna-sequencing data," *Biostatistics*, vol. 13, pp. 523–538, 7 2012. [Online]. Available: https://doi.org/10.1093/biostatistics/kxr031

[14] M. E. Ritchie, B. Phipson *et al.*, "limma powers differential expression analyses for RNA-sequencing and microarray studies," *Nucleic Acids Research*, vol. 43, no. 7, pp. e47–e47, 01 2015. [Online]. Available: https://doi.org/10.1093/nar/gkv007

[15] "Bolstad B (2024). preprocessCore: A collection of pre-processing functions. R package version 1.66.0," https://github.com/bmbolstad/preprocessCore, accessed: 17/07/2024.

[16] M. D. Robinson and A. Oshlack, "A scaling normalization method for differential expression analysis of rna-seq data." 2010. [Online]. Available: https://doi.org/10.1186/GB-2010-11-3-R25

[17] E. Saccenti, "Correlation patterns in experimental data are affected by normalization procedures: Consequences for data analysis and network inference," *Journal of proteome research*, vol. 16, no. 2, p. 619—634, February 2017. [Online]. Available: https://doi.org/10.1021/acs.jproteome.6b00704

[18] Y. Chen, L. Chen *et al.*, "edger 4.0: powerful differential analysis of sequencing data with expanded functionality and improved support for small counts and larger datasets." [Online]. Available: https://doi.org/10.1101/2024.01.21.576131

[19] "Analyzing RNA-seq data with DESeq2, [DESeq2 package version: 1.41.12]," https://bioconductor.org/packages/devel/bioc/vignettes/DESeq2/inst/doc/DESeq2.html, accessed: 12/10/2023.

[20] F. Abbas-Aghababazadeh, Q. Li, and B. L. Fridley, "Comparison of normalization approaches for gene expression studies completed with high-throughput sequencing," *PLOS ONE*, vol. 13, no. 10, pp. 1–21, 10 2018. [Online]. Available: https://doi.org/10.1371/journal.pone.0206312

[21] M. I. Love, W. Huber, and S. Anders, "Moderated estimation of fold change and dispersion for rna-seq data with deseq2," *Genome Biology*, vol. 15, 12 2014. [Online]. Available: https://doi.org/10.1186/s13059-014-0550-8

[22] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948. [Online]. Available: https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

[23] M. Soleimani-damaneh and M. Zarepisheh, "Shannon's entropy for combining the efficiency results of different dea models: Method and application," *Expert Systems with Applications*, vol. 36, no. 3, Part 1, pp. 5146–5150, 2009. [Online]. Available: https://doi.org/10.1016/j.eswa.2008.06.031

[24] J. Wu, J. Sun *et al.*, "Determination of weights for ultimate cross efficiency using shannon entropy," *Expert Systems with Applications*, vol. 38, pp. 5162–5165, 5 2011. [Online]. Available: https://doi.org/10.1016/j.eswa.2010.10.046

[25] H. Ogata, S. Goto *et al.*, "KEGG: Kyoto Encyclopedia of Genes and Genomes," *Nucleic Acids Research*, vol. 27, no. 1, pp. 29–34, 01 1999. [Online]. Available: https://doi.org/10.1093/nar/27.1.29

[26] C. Manzoni, D. A. Kia *et al.*, "Genome, transcriptome and proteome: The rise of omics data and their integration in biomedical sciences," *Briefings in Bioinformatics*, vol. 19, pp. 286–302, 3 2018. [Online]. Available: https://doi.org/10.1093/BIB/BBW114

[27] "David: Gene Functional Classification Tool, National Institutes of Health - David Bioinformatics, howpublished = https://david.ncifcrf.gov/gene2gene.jsp, note = Accessed: 16/07/2024,."

[28] "Bioconducter: Open source software for Bioinformatics," https://www.bioconductor.org/, accessed: 10/07/2024.

[29] "National Center for Biotechnology Information, howpublished = https://www.ncbi.nlm.nih.gov/, note = Accessed: 11/09/2024,."

[30] A. Bateman, M. J. Martin *et al.*, "Uniprot: the universal protein knowledgebase in 2023," *Nucleic Acids Research*, vol. 51, pp. D523–D531, 1 2023. [Online]. Available: https://doi.org/10.1093/nar/gkac1052

[31] F. Maleki, K. Ovens *et al.*, "Gene set analysis: Challenges, opportunities, and future research," 6 2020. [Online]. Available: https://doi.org/10.3389/fgene.2020.00654

[32] A. Subramanian, P. Tamayo *et al.*, "Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles," *Proceedings of the National Academy of Sciences*, vol. 102, no. 43, pp. 15 545–15 550, 2005. [Online]. Available: https://doi.org/10.1073/pnas.0506580102

[33] "Yu G (2024). enrichplot: Visualization of Functional Enrichment Result. R package version 1.24.2," https://yulab-smu.top/biomedical-knowledge-mining-book/, accessed: 08/08/2024.

[34] "Biomedical Knowledge Mining using GOSemSim and clusterProfiler," https://yulab-smu.top/biomedical-knowledge-mining-book/index.html, accessed: 05/09/2024.

List of Acronyms

CPM counts per million. 12, 13

DEG Differentially expressed gene. 4, 6, 7, 14, 17, 19, 21–25, 28–33, 35, 36

DGE Differential gene expression. 4, 13, 17, 20, 25, 28, 29, 35

DNA Deoxyribonucleic acid. 26

FDR false discovery rate. 4, 27–29

GEO Gene Expression Omnibus. 5

GO Gene Ontology. 9, 25–27, 29–33

GSEA Gene Set Enrichment Analysis. 2, 5, 9, 28, 29, 32

KEGG Kyoto Encyclopedia of Genes and Genomes. 9, 25, 27, 29

KO KEGG Orthology. 27, 28

MDS Multidimensional scaling. 5, 7, 23

NIH National Institutes of Health. 5, 6

ORA Over-Representation Analysis. 2, 5, 9, 28–33

RNA-Seq RNA sequencing. 4–7, 10, 12, 14, 16, 23, 28, 29, 35

SEA Shannon Entropy Aggregation. 4, 5, 16–20, 24, 28, 29, 35, 36