Package ‘XYomics’

November 13, 2025

Title Analysis of Sex Differences in Omics Data for Complex Diseases

Version 0.1.1

Maintainer Enrico Glaab <enrico.glaab@uni.lu>

Description Tools to analyze sex differences in omics data for complex diseases. It includes functions for differential expression analysis using the 'limma' method <doi:10.1093/nar/gkv007>, interaction testing between sex and disease, pathway enrichment with 'clusterProfiler' <doi:10.1089/omi.2011.0118>, and gene regulatory network (GRN) construction and analysis using 'igraph'. The package enables a reproducible workflow from raw data processing to biological interpretation.

Depends R (>= 3.6)

Imports limma, igraph, edgeR, Seurat, SeuratObject, clusterProfiler, org.Hs.eg.db, ReactomePA, data.table, ggplot2, tidyr, grid, ggraph, dplyr, ggrepel, scales, Rcpp, methods

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Suggests R.utils, DT, gridExtra, knitr, htmltools, kableExtra, rmarkdown, stringr

LinkingTo Rcpp

VignetteBuilder knitr, rmarkdown

NeedsCompilation yes

Author Enrico Glaab [aut, cre],
Sophie Le Bars [aut],
Mohamed Soudy [aut],
Murodzhon Akhmedov [cph]

Repository CRAN

Date/Publication 2025-11-13 21:20:08 UTC

Contents

`categorized_enrich_sc` *Perform Pathway Enrichment Analysis for Pre-Categorized Differentially Expressed Genes (DEGs)*

Description

This function performs pathway enrichment analysis for differentially expressed genes (DEGs), which are already categorized into different types (e.g., Dimorphic, Neutral, Sex-specific) via the ‘categorize_sex_sc‘ function. The function analyzes their enrichment in KEGG, GO, or Reactome pathways.

Usage

```
categorized_enrich_sc(
  DEGs_category,
  enrichment_db = "KEGG",
  organism = "hsa",
  org_db = org.Hs.eg.db,
  pvalueCutoff = 0.05,
  qvalueCutoff = 0.2
)
```

Arguments

`DEGs_category`	Data frame containing gene symbols and their corresponding DEG types. Must include columns ’DEG_Type’ (DEGs categories) and ’Gene_Symbols’.
`enrichment_db`	Character string specifying the enrichment database to use: "KEGG", "GO", or "REACTOME" (default: "KEGG").

`organism`	Character string representing the organism code. For KEGG enrichment, use "hsa" (default). For Reactome enrichment, use "human".
`org_db`	databse of the organism (e.g: Org.Hs.eg.db)
`pvalueCutoff`	Numeric value specifying the p-value cutoff for statistical significance (default: 0.05).
`qvalueCutoff`	Numeric value specifying the q-value cutoff for multiple testing correction (default: 0.2).

Details

- The input DEGs are already categorized by the ‘categorize_sex_sc‘ function. - For GO enrichment, an appropriate OrgDb object (e.g., org.Hs.eg.db for humans) must be available. - For KEGG and Reactome enrichment, gene symbols are first converted to ENTREZ IDs. - Requires the ’clusterProfiler’ package for enrichment analysis. - Ensures appropriate error handling for missing genes or database issues.

Value

A named list of enriched pathways for each DEG category, structured as a data frame.

categorize_sex_sc — *Compute sex-specific differentially expressed genes (DEGs) per category*

Description

Identifies male-specific, female-specific, sex-dimorphic, and sex-neutral DEGs from differential expression results.

Usage

```
categorize_sex_sc(
  male_degs,
  female_degs,
  target_fdr = 0.05,
  exclude_pval = 0.5,
  min_abs_logfc = 0.25
)
```

Arguments

`male_degs`	Data frame containing male differential expression results from one specific cell-type or bulk dataset.
`female_degs`	Data frame containing female differential expression results from one specific cell-type or bulk dataset.
`target_fdr`	Numeric. FDR threshold for significance.
`exclude_pval`	Numeric. P-value threshold for excluding genes in opposite sex.
`min_abs_logfc`	Numeric. Minimum absolute log2 fold change threshold.

Value

Data frame containing categorized DEGs with associated statistics.

construct_ppi_pcsf *Construct Protein-protein interaction Network using Prize-Collecting Steiner Forest*

Description

Constructs a condition-specific gene regulatory network based on differential expression results using the PCSF algorithm.

Usage

```
construct_ppi_pcsf(
  g,
  prizes,
  w = 2,
  b = 1,
  mu = 5e-04,
  seed = 1,
  min_nodes = 1
)
```

Arguments

`g`	An igraph object representing the base network.
`prizes`	A named numeric vector of gene scores (prizes). Names must match vertex names in g.
`w`	Numeric. Edge cost scaling weight. Default is 2.
`b`	Numeric. Balance between prizes and edge costs. Default is 1.
`mu`	Numeric. Trade-off parameter for sparsity. Default is 5e-04.
`seed`	Integer. Random seed. Default is 1.
`min_nodes`	Integer. Minimum number of nodes in subnetwork. Default is 1.

Value

An igraph object representing the extracted subnetwork. Returns NULL invisibly if no prize genes are present, the subnetwork is too small, or the PCSF algorithm fails.

convertdf2enr *Convert Data Frame to enrichResult*

Description

Converts a data frame containing enrichment results into a clusterProfiler enrichResult object. Assumes the data frame has columns: ID, geneID, pvalue, and optionally p.adjust.

Usage

```
convertdf2enr(df, pvalueCutoff = 0.1, pAdjustMethod = "BH")
```

Arguments

df	Data frame containing enrichment results.
pvalueCutoff	Numeric. P-value cutoff for the enrichment object (default: 0.1).
pAdjustMethod	Character string specifying the p-value adjustment method (default: "BH").

Value

An enrichResult object compatible with clusterProfiler plotting functions.

generate_boxplot *Generate Boxplots for Expression Data*

Description

Creates boxplots to visualize expression differences across conditions and genders.

Usage

```
generate_boxplot(
  x,
  index,
  phenotype,
  gender,
  title = "Expression Boxplot",
  xlab = "Conditions",
  ylab = "Expression Level"
)
```

Arguments

x	Expression data matrix.
index	Numeric vector indicating which features (rows) to plot.
phenotype	Vector of phenotype labels.
gender	Vector of gender labels.
title	Title for the plot.
xlab	Label for the x-axis.
ylab	Label for the y-axis.

Value

A boxplot is generated.

generate_cat_report *Generate a Comprehensive Analysis Report*

Description

This function creates an integrated report that combines key analysis outputs,

Usage

```
generate_cat_report(
  results_cat = results_cat,
  enrichment_cat = results_cat,
  grn_object = grn_object,
  output_file = "cat_analysis_report.html",
  output_dir = tempdir(),
  template_path = NULL,
  quiet = TRUE
)
```

Arguments

results_cat	A data frame or list containing differential expression results.
enrichment_cat	A list with enrichment objects (e.g., BP, MF, KEGG, and optionally GSEA results).
grn_object	An igraph object representing the gene regulatory network (e.g., from PCSF analysis).
output_file	Character. The desired name (and optionally path) for the rendered report (default: "analysis_report.html").
output_dir	Character. Output directory to save the report to.
template_path	Character. Path to the R Markdown template file. If `NULL`, the function uses the built-in template located in `inst/rmd/template_report.Rmd`.
quiet	Logical. If `TRUE` (default), rendering will be quiet.

Value

A character string with the path to the rendered report.

`generate_report` *Generate a Comprehensive Analysis Report*

Description

Creates an integrated HTML report combining differential expression results, enrichment analyses (GO, KEGG, GSEA), and gene regulatory network (GRN) data. Uses a parameterized R Markdown template for rendering.

Usage

```
generate_report(
  de_results,
  enrichment_results,
  grn_object,
  output_file = "analysis_report.html",
  template_path = NULL,
  params_list = list(),
  quiet = TRUE
)
```

Arguments

`de_results`	Data frame or list with differential expression results.
`enrichment_results`	List of enrichment results (e.g., BP, MF, KEGG, GSEA).
`grn_object`	An igraph object of the gene regulatory network.
`output_file`	Output report name (default: "analysis_report.html").
`template_path`	Path to the R Markdown template. If NULL, uses the built-in template.
`params_list`	Named list of extra parameters passed to the R Markdown report.
`quiet`	Logical; if TRUE (default), rendering is quiet.

Value

Character string with the path to the rendered report.

get_string_network *Download and Process STRING Protein-Protein Interaction Network*

Description

Downloads and processes the STRING protein-protein interaction network, converting it to a simplified igraph object. The function downloads the network from STRING database, filters interactions by confidence score, converts STRING IDs to ENTREZ IDs, and returns the largest connected component as an undirected graph.

Usage

```
get_string_network(
  organism = "9606",
  score_threshold = 700,
  use_default = TRUE
)
```

Arguments

`organism`	Character string specifying the NCBI taxonomy identifier. Default is "9606" (Homo sapiens).
`score_threshold`	Numeric value between 0 and 1000 specifying the minimum combined score threshold for including interactions. Default is 700.
`use_default`	it will return the default network (9606 and score of 700)

Details

The function performs the following steps:

1. Downloads protein interactions from STRING database
2. Filters interactions based on combined score
3. Downloads and processes STRING ID to ENTREZ ID mappings
4. Creates an igraph object with filtered interactions
5. Removes self-loops and multiple edges
6. Extracts the largest connected component

Value

An igraph object representing the largest connected component of the filtered STRING network, with the following properties:

- Undirected edges
- No self-loops

- No multiple edges
- Edge weights (1000 - combined_score)
- Vertex names as ENTREZ IDs

`identify_sex_specific_genes`

Identify sex-specific and sex-dimorphic genes

Description

This function identifies truly sex-specific and sex-dimorphic genes by analyzing differential expression results from both sexes.

Usage

```
identify_sex_specific_genes(
  male_results,
  female_results,
  target_fdr = 0.05,
  exclude_fdr = 0.5
)
```

Arguments

`male_results`	Data frame of differential expression results for males (from differential_expression).
`female_results`	Data frame of differential expression results for females (from differential_expression).
`target_fdr`	Numeric. FDR threshold for significant differential expression (default: 0.05).
`exclude_fdr`	Numeric. FDR threshold for excluding effects in the opposite sex (default: 0.5).

Details

This function implements a two-step approach to identify sex-specific effects: 1. Identifies genes significantly affected in one sex (target_fdr) 2. Confirms lack of effect in the other sex (exclude_fdr) Additionally identifies genes with opposite (dimorphic) or same (shared) effects in both sexes.

Value

A data frame with identified genes categorized as: - male-specific: significant in males, not significant in females - female-specific: significant in females, not significant in males - sex-dimorphic: significant in both sexes with opposite effects - sex-shared: significant in both sexes with same direction Including columns for gene IDs, logFC values, and FDR values for both sexes.

`improved_pathway_enrichment` *Improved Pathway Enrichment Analysis*

Description

Performs pathway enrichment analysis on a set of sex-biased genes using clusterProfiler.

Usage

```
improved_pathway_enrichment(
  gene_list,
  enrichment_db = "KEGG",
  organism = "hsa",
  org_db = org.Hs.eg.db,
  pvalueCutoff = 0.05,
  qvalueCutoff = 0.2
)
```

Arguments

`gene_list`	A character vector of gene identifiers.
`enrichment_db`	Character string specifying the database for enrichment. Options include "KEGG", "GO", and "Reactome". Default is "KEGG".
`organism`	Character string specifying the organism code (e.g., "hsa" for human).
`org_db`	database of the organism (e.g: "org.Hs.eg.db")
`pvalueCutoff`	Numeric. P-value cutoff for enrichment (default: 0.05).
`qvalueCutoff`	Numeric. Q-value cutoff for enrichment (default: 0.2).

Value

An enrichment result object.

`PCSF` *Prize-collecting Steiner Forest (PCSF)*

Description

`PCSF` returns a subnetwork obtained by solving the PCSF on the given interaction network.

Usage

```
PCSF(ppi, terminals, w = 2, b = 1, mu = 5e-04, dummies)
```

Arguments

`ppi` An interaction network, an **igraph** object.

`terminals` A list of terminal genes with prizes to be analyzed in the PCSF context. A named `numeric` vector, where terminal genes are named same as in the interaction network and numeric values correspond to the importance of the gene within the study.

`w` A `numeric` value for tuning the number of trees in the output. A default value is 2.

`b` A `numeric` value for tuning the node prizes. A default value is 1.

`mu` A `numeric` value for a hub penalization. A default value is 0.0005.

`dummies` A list of nodes that are to connected to the root of the tree. If missing the root will be connected to all terminals.

Details

The PCSF is a well-know problem in graph theory. Given an undirected graph *G = (V, E)*, where the vertices are labeled with prizes p_v and the edges are labeled with costs $c_e > 0$, the goal is to identify a subnetwork *G’ = (V’, E’)* with a forest structure. The target is to minimize the total edge costs in *E’*, the total node prizes left out of *V’*, and the number of trees in *G’*. This is equivalent to minimization of the following objective function:

$$F(G') = Minimize \sum_{e \in E'} c_e + \beta * \sum_{v \notin V'} p_v + \omega * k$$

where, *k* is the number of trees in the forest, and it is regulated by parameter ω. The parameter β is used to tune the prizes of nodes.

This optimization problem nicely maps onto the problem of finding differentially enriched subnetworks in the cell protein-protein interaction (PPI) network. The vertices of interaction network correspond to genes or proteins, and edges represent the interactions among them. We can assign prizes to vertices based on measurements of differential expression, copy number, or mutation, and costs to edges based on confidence scores for those intra-cellular interactions from experimental observation, yielding a proper input to the PCSF problem. Vertices that are assigned a prize are referred to *terminal* nodes, whereas the vertices which are not observed in patient data are not assigned a prize and are called *Steiner* nodes. After scoring the interactome, the PCSF is used to detect a relevant subnetwork (forest), which corresponds to a portion of the interactome, where many genes are highly correlated in terms of their functions and may regulate the differentially active biological process of interest. The PCSF aims to identify neighborhoods in interaction networks potentially belonging to the key dysregulated pathways of a disease. In order to avoid a bias towards the hub nodes of PPI networks to appear in solution of PCSF, we penalize the prizes of *Steiner* nodes according to their degree distribution in PPI, and it is regulated by parameter μ:

$$p'_v = p_v - \mu * degree(v)$$

The parameter μ also affects the total number of *Steiner* nodes in the solution. Higher the value of μ smaller the number of *Steiners* in the subnetwork, and vice-versa. Based on our previous analysis the recommended range of μ for biological networks is between 1e-4 and 5e-2, and users can choose the values resulting subnetworks with vertex sets that have desirable *Steiner/terminal* node ratio and average *Steiner/terminal* in-degree ratio in the template interaction network.

Value

The final subnetwork obtained by the PCSF. It return an **igraph** object with the node prize and edge cost attributes.

Author(s)

Murodzhon Akhmedov

References

Akhmedov M., LeNail A., Bertoni F., Kwee I., Fraenkel E., and Montemanni R. (2017) A Fast Prize-Collecting Steiner Forest Algorithm for Functional Analyses in Biological Networks. *Lecture Notes in Computer Science*, to appear.

`plot_network` *Plot a Condition-Specific protein-protein interaction network with DEG Annotations*

Description

Visualizes a gene regulatory or protein–protein interaction network for a given cell type and differential expression group. Nodes are sized and colored by degree, and key hub genes are optionally annotated with their barplots of log fold-changes across sexes.

Usage

```
plot_network(g, cell_type, DEG_type, result_categories)
```

Arguments

`g`	An ‘igraph‘ object representing the gene or protein interaction network.
`cell_type`	Character string. The cell type label used in the plot title.
`DEG_type`	Character string. The differential expression category to visualize (e.g., ‘"sex-dimorphic"‘).
`result_categories`	A ‘data.frame‘ or tibble containing at least the columns: ‘"DEG_Type"‘, ‘"Gene_Symbols"‘, ‘"Male_avg_logFC"‘, and ‘"Female_avg_logFC"‘.

Value

A ‘ggplot‘ object representing the visualized network.

sex_interaction_analysis_bulk

Perform Sex-Phenotype Interaction Analysis for Bulk Data (Interaction Term)

Description

This function performs a formal interaction analysis on bulk expression data to identify genes whose expression is significantly modulated by the interaction between sex and a given phenotype/condition. It uses a linear model with a multiplicative interaction term (‘phenotype * sex‘).

Usage

```
sex_interaction_analysis_bulk(
  x,
  phenotype,
  gender,
  phenotype_labels = c("WT", "TG"),
  sex_labels = c("F", "M")
)
```

Arguments

`x`	A numeric matrix of expression data (features x samples).
`phenotype`	A character or factor vector indicating the condition for each sample.
`gender`	A character or factor vector indicating the sex for each sample.
`phenotype_labels`	Character vector. Labels for phenotype groups (default: c("WT", "TG")).
`sex_labels`	Character vector. Labels for sexes (default: c("F", "M")).

Details

This function constructs a design matrix that includes a formal interaction term between the phenotype and sex (e.g., ‘~ phenotype * sex‘). It then uses ‘limma‘ to test for genes where the effect of the phenotype differs significantly between sexes. This is a statistically rigorous approach to identify sex-modulated genes.

Value

A data frame with differential expression statistics for the interaction term, including logFC, t-statistic, P-value, and adjusted P-value.

sex_interaction_analysis_sc

Perform Sex-Phenotype Interaction Analysis for Single-Cell Data

Description

Performs differential difference analysis for a given cell type to identify genes modulated by sex-phenotype interactions using limma.

Usage

```
sex_interaction_analysis_sc(
  seurat_obj,
  target_cell_type,
  sex_col = "sex",
  phenotype_col = "status",
  celltype_col = "cell_type",
  min_logfc = 0.25,
  fdr_threshold = 0.05,
  sex_labels = c("F", "M"),
  phenotype_labels = c("WT", "TG")
)
```

Arguments

seurat_obj	A Seurat object.
target_cell_type	Character. Cell type to analyze.
sex_col	Character. Column name for sex (default "sex").
phenotype_col	Character. Column name for phenotype (default "status").
celltype_col	Character. Column name for cell type (default "cell_type").
min_logfc	Numeric. Minimum absolute log fold change (default 0.25).
fdr_threshold	Numeric. FDR threshold for significance (default 0.05).
sex_labels	Character vector of sex labels (default c("F","M")).
phenotype_labels	Character vector of phenotype groups (default c("WT","TG")).

Value

A list with complete DE results, significant results, and summary statistics.

sex_stratified_analysis_bulk

Perform differential expression analysis within each sex

Description

This function identifies differentially expressed genes between conditions separately for each sex using a linear modeling approach.

Usage

```
sex_stratified_analysis_bulk(
  x,
  phenotype,
  gender,
  analysis_type = c("male", "female")
)
```

Arguments

`x`	A numeric matrix of expression data (features × samples).
`phenotype`	A vector indicating condition labels for each sample.
`gender`	A vector indicating gender for each sample. Labels must start with "f" (female) and "m" (male).
`analysis_type`	Character. Type of analysis to perform: "dimorphic" (difference in differences), "female" (female condition effect), or "male" (male condition effect). Default is "dimorphic".

Details

This function performs differential expression analysis within each sex separately. For male analysis, it compares conditions within males. For female analysis, it compares conditions within females. For dimorphic analysis, it tests for difference in condition effects between sexes. Note: To identify truly sex-specific genes, use the output of this function as input for identify_sex_specific_genes().

Value

A data frame with differential expression statistics including logFC, AveExpr, t-statistic, P-value, and adjusted P-value.

sex_stratified_analysis_sc

Compute sex-specific differentially expressed genes (DEGs)

Description

Identifies differentially expressed genes (DEGs) separately for male and female samples within different cell types using the Seurat package. Compares gene expression between control and perturbed groups in each sex.

Usage

```
sex_stratified_analysis_sc(
  seurat_obj,
  sex_column = "sex",
  phenotype_column = "status",
  celltype_column = "cell_type",
  sex_labels_vector = c("F", "M"),
  min_logfc = 0.25,
  phenotype_labels_vector = c("WT", "TG"),
  method = "wilcox"
)
```

Arguments

`seurat_obj` Seurat object containing the single-cell data.

`sex_column` Character. Column name in metadata for sex (default "sex").

`phenotype_column` Character. Column name in metadata for phenotype (default "status").

`celltype_column` Character. Column name in metadata for cell type (default "cell_type").

`sex_labels_vector` Character vector of sex labels (default c("F","M")).

`min_logfc` Numeric. Minimum absolute log fold change threshold (default 0.25).

`phenotype_labels_vector` Character vector of phenotype groups (default c("WT","TG")).

`method` Character. Statistical test to use for differential expression (default "wilcox").

Value

A list with male and female DEGs results.

`visualize_network` *Visualize Gene Regulatory Network with Pie Charts*

Description

Plots a network with nodes represented by pie charts that display male and female effects.

Usage

```
visualize_network(
  g,
  female_res,
  male_res,
  vertex.size = 5,
  vertex.label.cex = 0.8,
  ...
)
```

Arguments

`g`	An igraph network object.
`female_res`	Differential expression results for females.
`male_res`	Differential expression results for males.
`vertex.size`	Size of the network nodes.
`vertex.label.cex`	Text size for vertex labels.
`...`	Additional graphical parameters.

Value

The modified igraph object with visualization attributes.

Index