Package: XYomics (via r-universe)

May 11, 2026

Title Analysis of Sex Differences in Omics Data for Complex Diseases

Version 0.1.4

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, SeuratObject, Seurat, data.table, ggplot2, tidyr, grid, ggraph, dplyr, AnnotationDbi, ggrepel, Rcpp, cowplot, patchwork, methods, DESeq2, S4Vectors, clusterProfiler

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Suggests R.utils, DT, gridExtra, knitr, htmltools, kableExtra, rmarkdown, stringr, ReactomePA, org.Hs.eg.db

LinkingTo Rcpp

VignetteBuilder knitr, rmarkdown

NeedsCompilation yes

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

Config/pak/sysreqs libcairo2-dev cmake libfontconfig1-dev libfreetype6-dev libglpk-dev make libicu-dev libpng-dev libuv1-dev libxml2-dev libssl-dev python3 zlib1g-dev

Repository https://eglaab.r-universe.dev

Date/Publication 2026-05-11 18:11:53 UTC

RemoteUrl https://github.com/cran/XYomics

RemoteRef HEAD

RemoteSha e776493453521403af812ea8f194d56205b34f7f

Contents

`categorize_sex` *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. "sex-modulated" requires the interaction mode and is not identified by categorize_sex.

Usage

```
categorize_sex(
  male_degs,
  female_degs,
  alpha = 0.05,
  beta = 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.
alpha	Numeric. FDR threshold for significance.
beta	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.

---

categorized_enrich *Perform Pathway Enrichment Analysis for Pre-Categorized DEGs*

Description

Performs pathway enrichment analysis for categorized DEGs using GO, KEGG, Reactome, or a custom TERM2GENE database.

Usage

```
categorized_enrich(
  DEGs_category,
  enrichment_db = "KEGG",
  organism = "hsa",
  gene_type = "SYMBOL",
  org_db = NULL,
  custom_db = NULL,
  pvalueCutoff = 0.05,
  qvalueCutoff = 0.2,
  return_df = FALSE
)
```

Arguments

DEGs_category	Data frame with columns 'DEG_Type' and 'Gene_Symbols'.
enrichment_db	"KEGG", "GO", "REACTOME", or "CUSTOM" (default: "KEGG").
organism	Organism code for KEGG (default: "hsa").
gene_type	Gene identifier type supported by OrgDb (e.g., "SYMBOL", "ENTREZID", "ENSEMBL", "UNIPROT").
org_db	OrgDb object for gene annotation (default: org.Hs.eg.db).
custom_db	TERM2GENE data frame, required if enrichment_db = "CUSTOM".
pvalueCutoff	Numeric p-value cutoff (default: 0.05).
qvalueCutoff	Numeric q-value cutoff (default: 0.2).
return_df	Logical, if TRUE returns data.frames, else enrichResult objects (default: FALSE).

Details

Gene IDs converted only when needed; KEGG/Reactome require ENTREZID; GO/CUSTOM use provided gene_type.

Value

Named list of enrichment results per DEG_Type.

---

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.

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

---

convert_gene_ids *Convert gene identifiers in a DEG table*

Description

Convert gene identifiers in a DEG table

Usage

```
convert_gene_ids(
  df,
  fromType,
  toType = "SYMBOL",
  org_db = NULL,
  gene_col = "Gene_Symbols",
  remove_na = TRUE,
  remove_duplicates = TRUE
)
```

Arguments

df	Data frame containing a gene column
fromType	Original ID type (e.g. "UNIPROT", "ENSEMBL")
toType	Target ID type (e.g. "SYMBOL", "ENTREZID")
org_db	OrgDb object (default: org.Hs.eg.db)
gene_col	Column name containing gene IDs (default: "Gene_Symbols")
remove_na	Remove unmapped genes (default: TRUE)
remove_duplicates	Remove duplicated mapped IDs (default: TRUE)

Value

Updated data frame with converted gene IDs

`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,
  enrichment_cat,
  grn_cat,
  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_cat`	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_dotplot_sc` *Generate dotplot of selected genes across groups*

Description

Visualizes gene expression across Sex / Phenotype / Cell type combinations.

Usage

```
generate_dotplot_sc(
  seurat_obj,
  genes,
  sex = "sex",
  phenotype = "status",
  celltype_col = "cell_type",
  target_cell_type_flag = "all"
)
```

Arguments

`seurat_obj`	A Seurat object
`genes`	Character vector of genes to plot
`sex`	Metadata column for sex
`phenotype`	Metadata column for phenotype/status
`celltype_col`	Metadata column for cell types
`target_cell_type_flag`	Either "all" or a specific cell type

Value

A ggplot object (DotPlot)

`generate_violinplot_bulk` *Generate Violin Plots for Bulk Expression Data*

Description

Creates violin plots displaying the distribution of gene expression values across Sex and Phenotype groups for a set of selected genes. Multiple genes (up to 10) are visualized using faceting.

Usage

```
generate_violinplot_bulk(
  expression_data,
  sex,
  phenotype,
  genes,
  sex_labels_vector = c("F", "M"),
  phenotype_labels_vector = c("WT", "TG")
)
```

Arguments

expression_data	A matrix or data.frame of bulk expression values with genes in rows and samples in columns.
sex	A vector indicating the sex of each sample.
phenotype	A vector indicating the phenotype or condition of each sample.
genes	Character vector of genes to visualize (maximum 10 recommended).
sex_labels_vector	Character vector specifying sex labels.
phenotype_labels_vector	Character vector specifying phenotype labels.

Value

A 'ggplot' object containing the violin plots.

---

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

STRING combined scores range from 0 (low confidence) to 1000 (high confidence). Because PCSF treats edge weights as traversal costs to be minimized, scores are inverted via 1 - score/1000 so that higher-confidence interactions correspond to lower costs. 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 (1 - combined_score/1000)
- Vertex names as ENTREZ IDs

---

get_top_hubs *Get Top Hub Genes by Betweenness Centrality*

Description

Computes normalized betweenness centrality for an igraph network and returns the top N hub nodes ranked by centrality.

Usage

```
get_top_hubs(g, top_n = 10, directed = FALSE)
```

Arguments

g	An 'igraph' object representing the network.
top_n	Integer. Number of top hub nodes to return. Default is 10.
directed	Logical. Whether the network is directed. Default is FALSE.

Value

A named numeric vector of betweenness scores for the top hub nodes.

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_enrichment_dotplots`

Create enrichment dotplots from enrichResult objects

Description

Generates dotplots from enrichment results. Input must contain enrichResult objects (clusterProfiler / ReactomePA). Invalid entries are skipped with messages. Works for single enrichResult, flat list (DEG types), or nested list (cell type → DEG type).

Usage

```
plot_enrichment_dotplots(enrichment_results, showCategory = 10, ncol = 2)
```

Arguments

enrichment_results	A list or nested list of enrichResult objects.
showCategory	Number of categories to display (default = 10).
ncol	Number of columns for arranged plots (default = 2).

Value

Named list of combined ggplots. For nested input, names correspond to cell types; for flat input, a single ‘"all"‘ entry is returned.

---

plot_network *Plot a condition-specific protein–protein interaction network with DEG annotations*

Description

Visualizes a gene or protein interaction network for a selected differential expression category. Nodes are sized by degree centrality, and hub genes can be optionally annotated with mini barplots showing sex-specific log fold changes.

Usage

```
plot_network(
  g,
  DEG_type,
  result_categories,
  cell_type = "",
  top_hubs = 20,
  show_barplot = FALSE
)
```

Arguments

g	An igraph object representing the interaction network.
DEG_type	Character string specifying the DEG category to visualize.
result_categories	A data.frame containing DEG results with at least the columns DEG_Type, Gene_Symbols, Male_avg_logFC, and Female_avg_logFC.
cell_type	Optional character string used in the plot title.
top_hubs	Integer specifying the number of hub genes to highlight.
show_barplot	Logical indicating whether hub barplots are displayed.

Details

Nodes represent genes or proteins and edges represent interactions. Node size reflects degree centrality (number of connections). The top top_hubs nodes by degree are labeled. When show_barplot = TRUE, mini barplots are overlaid on hub nodes to display sex-specific log fold changes. The visualization is restricted to the largest connected component of the network.

Value

A ggplot object representing the network visualization.

Examples

```
# Minimal reproducible example (CRAN-safe)
library(igraph)

# Create a small toy network
g <- make_ring(5)
V(g)$name <- paste0("Gene", 1:5)

# Create minimal DEG table
result_categories <- data.frame(
  DEG_Type = rep("example", 5),
  Gene_Symbols = paste0("Gene", 1:5),
  Male_avg_logFC = runif(5, -1, 1),
  Female_avg_logFC = runif(5, -1, 1)
)

# Run function
plot_network(g, DEG_type = "example", result_categories = result_categories)
```

plot_network_pipeline *Generate protein-protein interaction network plots for single-cell DEGs*

Description

Accepts a nested list of igraph networks (cell_type × DEG_type) object and produces annotated network visualizations.

Usage

```
plot_network_pipeline(
  network_list,
  result_categories_list,
  top_hubs = 20,
  ncol = 2,
  show_barplot = FALSE
)
```

Arguments

network_list	A single igraph object or a nested list: cell_type → DEG_type → igraph
result_categories_list	A data.frame or named list matching the networks, containing DEG results.
top_hubs	Number of hub genes to annotate (default = 20)
ncol	Number of columns when combining DEG type plots (default = 2)
show_barplot	Logical indicating whether hub barplots are displayed.

Value

Named list of network plots per cell type (or ‘"all"‘ for single network)

plot_volcano_deg *Generate volcano plots for categorized DEGs*

Description

Creates volcano plots for all DEG categories (male-specific, female-specific, sex-dimorphic, sex-neutral). Accepts either a single data.frame (bulk) or a list of data.frames per cell type.

Usage

```
plot_volcano_deg(de_results, top_n = 5, logfc_thresh = 1.5, ncol = 2)
```

Arguments

de_results	A data.frame or named list of data.frames containing categorized DEGs. Required columns: DEG_Type, Gene_Symbols, Male_avg_logFC, Male_FDR, Female_avg_logFC, Female_FDR
top_n	Number of top genes to highlight per sex (default = 5).
logfc_thresh	Minimum absolute logFC to consider a gene for highlighting (default = 1.5).
ncol	Number of columns for arranging plots (default = 2).

Value

Named list of combined volcano plots per cell type or "all" for bulk.

---

ppi_pipeline *Run PPI PCSf pipeline for single cell DEG categories*

Description

Builds protein protein interaction subnetworks for each cell type based on DEG categories and FDR derived node prizes.

Usage

```
ppi_pipeline(result_categories, g, target_cell_type_flag = "all")
```

Arguments

result_categories	A named list of data.frames containing DEG results per cell type. Each data.frame must include DEG_Type, Gene_Symbols, Male_FDR, and Female_FDR.
g	An igraph object representing the global protein interaction network.
target_cell_type_flag	Character. Specific cell type to analyze or "all".

Value

A named list containing PCSf networks per cell type and DEG category.

`sex_interaction_analysis_bulk` *Perform Sex-Phenotype Interaction Analysis for Bulk Data*

Description

Identifies genes whose expression is modulated by the interaction between sex and phenotype using a differential difference contrast

Usage

```
sex_interaction_analysis_bulk(
  expression_data,
  phenotype,
  sex,
  phenotype_labels_vector = c("WT", "TG"),
  sex_labels_vector = c("F", "M"),
  min_logfc = 0.25,
  alpha = 0.05,
  min_samples = 20
)
```

Arguments

`expression_data`	Numeric matrix of expression data (features x samples).
`phenotype`	Character or factor vector of sample phenotypes.
`sex`	Character or factor vector of sample sexes.
`phenotype_labels_vector`	Character vector of phenotype levels (default c("WT","TG")).
`sex_labels_vector`	Character vector of sex levels (default c("F","M")).
`min_logfc`	Numeric. Minimum absolute log fold change for reporting significant genes (default 0.25).
`alpha`	Numeric. FDR threshold for significance (default 0.05).
`min_samples`	Integer. Minimum number of samples per group (default 20).

Value

A list with all DE results, filtered significant DEGs, and summary statistics.

`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 = "sex",
  phenotype = "status",
  celltype_col = "cell_type",
  min_logfc = 0.25,
  alpha = 0.05,
  sex_labels_vector = c("F", "M"),
  phenotype_labels_vector = c("WT", "TG"),
  min_samples = 20
)
```

Arguments

`seurat_obj`	A Seurat object.
`target_cell_type`	Character. Cell type to analyze.
`sex`	Character. Column name for sex (default "sex").
`phenotype`	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 for reporting significant genes (default 0.25).
`alpha`	Numeric. FDR threshold for significance (default 0.05).
`sex_labels_vector`	Character vector of sex labels (default c("F","M")).
`phenotype_labels_vector`	Character vector of phenotype groups (default c("WT","TG")).
`min_samples`	Integer. Minimum number of cells per group (default 20).

Value

A list with all DE results, filtered significant DEGs, and summary statistics.

`sex_interaction_pipeline_sc` *Run sex phenotype interaction analysis pipeline for single cell data*

Description

Performs sex phenotype interaction analysis per cell type using a limma based difference in differences model.

Usage

```
sex_interaction_pipeline_sc(
  seurat_obj,
  target_cell_type_flag = "all",
  sex = "sex",
  phenotype = "status",
  celltype_col = "cell_type",
  min_logfc = 0.25,
  alpha = 0.05,
  sex_labels_vector = c("F", "M"),
  phenotype_labels_vector = c("WT", "TG"),
  min_samples = 20
)
```

Arguments

`seurat_obj`	Seurat object containing single cell data.
`target_cell_type_flag`	Character. Specific cell type to analyze or "all".
`sex`	Character. Metadata column for sex.
`phenotype`	Character. Metadata column for phenotype.
`celltype_col`	Character. Metadata column for cell type.
`min_logfc`	Numeric. Minimum absolute log fold change threshold.
`alpha`	Numeric. FDR threshold for significance.
`sex_labels_vector`	Character vector specifying sex levels.
`phenotype_labels_vector`	Character vector specifying phenotype levels.
`min_samples`	Integer. Minimum number of cells per group (default 20).

Value

A named list containing interaction analysis results per cell type.

`sex_stratified_analysis_bulk` *Perform differential expression analysis within each sex*

Description

This function identifies differentially expressed genes between phenotype groups separately for each sex using limma (default), DESeq2, or edgeR.

Usage

```
sex_stratified_analysis_bulk(
  expression_data,
  sex,
  phenotype,
  sex_labels_vector = c("F", "M"),
  phenotype_labels_vector = c("WT", "TG"),
  min_samples = 3,
  method = c("limma", "deseq2", "edger")
)
```

Arguments

`expression_data`	A numeric matrix of expression data with features in rows and samples in columns. For `method = "limma"`, this can be a normalized log-expression matrix or a raw count matrix. For `method = "deseq2"` or `method = "edger"`, this must be a raw (non-negative integer) count matrix.
`sex`	A vector indicating sex for each sample.
`phenotype`	A vector indicating phenotype labels for each sample.
`sex_labels_vector`	A character vector of length two defining the labels for sex (e.g., c("F", "M")). The first element must correspond to "female" and the second to "male".
`phenotype_labels_vector`	Character vector specifying phenotype levels. The first element is the reference, the second the comparison; contrast = 2 - 1 (e.g. c("WT","TG")).
`min_samples`	Integer. Minimum number of samples per group (default 3).
`method`	Character. Differential expression method to use. One of `"limma"` (default), `"deseq2"`, or `"edger"`.

Details

For each sex, the function subsets the data and fits the appropriate model:

- **limma**: Fits a linear model via `lmFit + eBayes` on the input matrix directly. Suitable for microarray, proteomics, or pre-normalized RNA-seq data. If raw RNA-seq counts are provided, the function automatically detects integer matrices and applies `limma-voom` internally with a warning. For full control, consider using `"edger"` or `"deseq2"` instead, or apply `voom` externally and pass the transformed data with `method = "limma"`.
- **DESeq2**: Fits a negative binomial GLM via `DESeq2::DESeq()`. Requires raw integer counts. Performs internal normalization (median-of-ratios) and dispersion estimation.
- **edgeR**: Fits a negative binomial GLM via `edgeR::glmQLFit() + edgeR::glmQLFTest()`. Requires raw integer counts. Performs TMM normalization internally.

Positive logFC values indicate higher expression in the test phenotype (second element of `phenotype_labels_vector`) compared to the reference (first element).

Value

A list with three elements:

male_DEGs A data.frame of DE results for males.

female_DEGs A data.frame of DE results for females.

validation Output from `validate_input_bulk()`.

Regardless of the method, each DEG data.frame contains the standardized columns `avg_log2FC`, `p_val`, `p_val_adj`, and `gene`, with gene names as rownames. Additional method-specific columns (e.g. `baseMean` for DESeq2, `logCPM` for edgeR, `AveExpr` for limma) are retained for reference.

---

sex_stratified_analysis_sc

Perform sex-stratified differential expression analysis for single-cell Seurat object

---

Description

Perform sex-stratified differential expression analysis for single-cell Seurat object

Usage

```
sex_stratified_analysis_sc(
  seurat_obj,
  sex = "sex",
  phenotype = "status",
  celltype_col = "cell_type",
  sex_labels_vector = c("F", "M"),
  phenotype_labels_vector = c("WT", "TG"),
  test.use = "wilcox",
  min_samples = 3
)
```

Arguments

seurat_obj	A Seurat object containing single-cell RNA-seq data.
sex	Character. Name of the column in metadata indicating sex.
phenotype	Character. Name of the column indicating phenotype/condition.
celltype_col	Character. Name of the column indicating cell type.
sex_labels_vector	A character vector of length two defining the labels for sex (e.g., c("F", "M")). The first element must correspond to "female" and the second to "male".
phenotype_labels_vector	Character vector of length 2 specifying the reference and test phenotypes (e.g., c("WT", "TG")).
test.use	Statistical test to use in FindMarkers.
min_samples	Integer. Minimum number of cells per group (default 3).

Value

A list with two elements: ‘male_DEGs‘ and ‘female_DEGs‘, each containing a list of DEGs per cell type.

---

sex_stratified_pipeline_sc

Run sex stratified differential expression pipeline for single cell data

---

Description

Performs sex specific differential expression analysis per cell type and categorizes genes into male specific, female specific, sex dimorphic, and sex neutral groups.

Usage

```
sex_stratified_pipeline_sc(
  seurat_obj,
  target_cell_type_flag = "all",
  sex = "sex",
  phenotype = "status",
  celltype_col = "cell_type",
  sex_labels_vector = c("F", "M"),
  phenotype_labels_vector = c("WT", "TG"),
  test.use = "wilcox",
  alpha = 0.05,
  beta = 0.5,
  min_abs_logfc = 0.25,
  min_samples = 3
)
```

Arguments

seurat_obj	Seurat object containing single cell data.
target_cell_type_flag	Character. Cell type to analyze.
sex	Character. Metadata column for sex.
phenotype	Character. Metadata column for phenotype.
celltype_col	Character. Metadata column for cell type.
sex_labels_vector	Character vector specifying sex levels.
phenotype_labels_vector	Character vector specifying phenotype levels.
test.use	Character. Statistical test used in FindMarkers.
alpha	Numeric. FDR threshold for significance.
beta	Numeric. P value threshold for exclusion in opposite sex.
min_abs_logfc	Numeric. Minimum absolute log2 fold change for categorization.
min_samples	Integer. Minimum number of cells per group (default 3).

Value

A named list of categorized differential expression results per cell type.

---

validate_input_bulk *Validate bulk input data for sex-stratified analysis*

Description

Validate bulk input data for sex-stratified analysis

Usage

```
validate_input_bulk(
  expression_data,
  sex,
  phenotype,
  sex_labels_vector = c("F", "M"),
  phenotype_labels_vector = c("WT", "TG"),
  min_samples = 3
)
```

Arguments

expression_data	Numeric matrix (features x samples)
sex	Vector of sex labels
phenotype	Vector of phenotype labels
sex_labels_vector	Expected sex labels (length 2)
phenotype_labels_vector	Expected phenotype labels (length 2)
min_samples	Minimum samples per group

Value

A list with validation status, ratios, imbalance flag, and group counts

---

validate_input_sc *Validate single-cell input data for sex-stratified analysis*

Description

Validate single-cell input data for sex-stratified analysis

Usage

```
validate_input_sc(
  seurat_obj,
  sex = "sex",
  phenotype = "status",
  celltype_col = "cell_type",
  sex_labels_vector = c("F", "M"),
  phenotype_labels_vector = c("WT", "TG"),
  min_cells = 3
)
```

Arguments

seurat_obj	Seurat object
sex	Column name for sex
phenotype	Column name for phenotype
celltype_col	Column name for cell type
sex_labels_vector	Expected sex labels (length 2)
phenotype_labels_vector	Expected phenotype labels (length 2)
min_cells	Minimum number of cells per group

Value

A list with validation status, ratios, imbalance flag, and group counts

Index