Package: iTALK (via r-universe)

May 23, 2026

Type Package

Title Characterize and Illustrate Intercellular Communication

Version 0.1.1

Date 2026-01-23

Author Zaoqu Liu [aut, cre], Yuanxin Wang [aut]

Maintainer Zaoqu Liu <`liuzaoqu@163.com`>

URL `https://github.com/Zaoqu-Liu/iTALK`

BugReports `https://github.com/Zaoqu-Liu/iTALK/issues`

Description iTALK, a computational approach to characterize, compare, and illustrate intercellular communication signals in the multicellular ecosystem using either bulk RNA sequencing data or single cell RNAseq data. iTALK can in principle be used to dissect the complexity, diversity, and dynamics of cell-cell communication from a wide range of cellular processes.

License file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0)

Imports methods, progress, pbapply, dplyr, tidyr, tibble, graphics, randomcoloR, circlize, ggplot2, igraph, biomaRt, httr, R.cache, digest

Suggests BiocParallel, Biobase, DESeq2, edgeR, monocle, scde, DEsingle, MAST, knitr, rmarkdown, testthat (>= 3.0.0), RColorBrewer, viridis

VignetteBuilder knitr

biocViews RNASequencing, SingleCell, GeneExpression, CellBiology

RoxygenNote 7.3.3

Config/testthat/edition 3

Config/pak/sysreqs libglpk-dev libicu-dev libpng-dev libxml2-dev libssl-dev libnode-dev zlib1g-dev

Repository https://zaoqu-liu.r-universe.dev

Date/Publication 2026-01-23 13:52:12 UTC

RemoteUrl https://github.com/Zaoqu-Liu/iTALK

RemoteRef main

RemoteSha b561893fcf305e109ea5d424f86a0609dba5cb9d

Contents

`convert_expression_matrix` *Convert Expression Matrix Between Species*

Description

Applies species conversion mapping to an expression matrix, handling one-to-many gene mappings with specified aggregation method.

Usage

```
convert_expression_matrix(
  expr_matrix,
  gene_mapping,
  handle_duplicates = c("mean", "sum", "max")
)
```

Arguments

`expr_matrix`	Matrix or data.frame. Expression data (genes x cells/samples)
`gene_mapping`	Data.frame. Output from convert_species_biomart()$mapping
`handle_duplicates`	Character. Method for aggregating expression when multiple source genes map to one target gene: • "mean": Average expression (default, conservative) • "sum": Sum expression (appropriate for count data) • "max": Maximum expression

Value

List with elements:

- expr_matrix: Converted expression matrix (target genes x cells)
- conversion_info: data.frame with mapping details
- stats: list with conversion statistics

Examples

```
## Not run:
# After obtaining mapping
mapping_result <- convert_species_biomart(
  genes = rownames(mouse_expr),
  from_species = "Mus_musculus"
)

# Convert expression matrix
converted <- convert_expression_matrix(
  expr_matrix = mouse_expr,
  gene_mapping = mapping_result$mapping,
  handle_duplicates = "mean"
)

# Use converted matrix
human_expr <- converted$expr_matrix

## End(Not run)
```

`convert_species_biomart` *Convert Genes Between Species Using BioMart*

Description

Converts gene symbols between species using Ensembl BioMart ortholog mappings. Provides accurate, biologically-validated homolog mappings rather than simple name transformation.

Usage

```
convert_species_biomart(
  genes,
  from_species,
  to_species = "Homo_sapiens",
  ensembl_version = 103,
  mirror = NULL,
  cache = TRUE,
  max_tries = 5
)
```

Arguments

`genes`	Character vector. Gene symbols to convert
`from_species`	Character. Source species: • "Homo_sapiens" (human) • "Mus_musculus" (mouse)
`to_species`	Character. Target species (default: "Homo_sapiens")
`ensembl_version`	Character or numeric. Ensembl version (default: 103). Using a fixed version ensures reproducibility. Use "current_release" for latest version.
`mirror`	Character or NULL. Ensembl mirror for faster access: • "www": Main server (Europe) • "uswest": US West Coast • "useast": US East Coast • "asia": Asia
`cache`	Logical. Cache BioMart results for faster repeated queries (default: TRUE)
`max_tries`	Integer. Maximum retry attempts for network operations (default: 5)

Details

Ortholog Mapping: Uses Ensembl's "associated_gene_name" attribute which provides the primary ortholog symbol. For mouse→human conversion, this maps:

- Tgfb1 → TGFB1
- Vegfa → VEGFA
- Ctnnb1 → CTNNB1

One-to-Many Mappings: Some genes have multiple orthologs (e.g., Tgfb1 might map to TGFB1, TGFB2, TGFB3). By default, all mappings are returned. Downstream functions handle aggregation.

Caching: When cache=TRUE, results are stored using R.cache with key based on:

- Gene set (hashed)
- Source and target species

- Ensembl version

Cache dramatically speeds up repeated analyses.

Network Requirements: Requires internet connection to query Ensembl BioMart (first time). Queries typically take 10-30 seconds depending on gene count and network speed.

Value

List with elements:

- mapping: data.frame with columns from_gene, to_gene
- unmapped: character vector of genes without orthologs
- stats: list with mapping statistics (n_input, n_mapped, mapping_rate, etc.)
- cache_key: cache identifier (if cache=TRUE)

Examples

```
## Not run:
# Convert mouse genes to human
result <- convert_species_biomart(
  genes = c("Tgfb1", "Vegfa", "Ctnnb1"),
  from_species = "Mus_musculus",
  to_species = "Homo_sapiens"
)

# Check mapping
result$mapping
#   from_gene to_gene
#   Tgfb1     TGFB1
#   Vegfa     VEGFA
#   Ctnnb1    CTNNB1

# Check statistics
result$stats$mapping_rate  # Proportion successfully mapped

# Unmapped genes
result$unmapped

## End(Not run)
```

`database` *Ligand-Receptor Interaction Database*

Description

A data frame containing ligand-receptor pairs for cell-cell communication analysis.

Usage

```
database
```

Format

A data frame with columns:

Pair.Name Name of the ligand-receptor pair

Ligand.ApprovedSymbol Official gene symbol of the ligand

Ligand.Name Full name of the ligand

Receptor.ApprovedSymbol Official gene symbol of the receptor

Receptor.Name Full name of the receptor

Classification Type of interaction (e.g., Cytokine, Growth factor)

Source

Curated from multiple public databases including CellPhoneDB, CellChatDB, and literature.

Examples

```
data(database)
head(database)
```

`DEG` *Call DEGenes*

Description

This function loads the data as a dataframe, and method as a string. It assumes that each line contains gene expression profile of one single cell, and each column contains the one single gene expression profile in different cells. The dataframe should also contain the cell type information with column name 'cell_type', as well as group information as 'compare_group' Batch information as 'batch' is optional. If included, users may want to use the raw count data for later analysis. Differential expressed genes will be called within each cell type by the method users select. For bulk RNAseq, we provide edgeR, DESeq2. And for scRNA-seq, popular methods in packages scde, monocle, DEsingle and MAST are available.

Usage

```
DEG(
  data,
  method,
  min_gene_expressed = 0,
  min_valid_cells = 0,
  contrast = NULL,
  q_cut = 0.05,
```

```
  add = TRUE,
  top = 50,
  stats = "mean",
  ...
)
```

Arguments

`data`	Input raw or normalized count data with column 'cell_type' and 'compare_group'
`method`	Method used to call DEGenes. Available options are: • Wilcox: Wilcoxon rank sum test • DESeq2: Negative binomial model based differential analysis (Love et al, Genome Biology, 2014) • SCDE: Bayesian approach to single-cell differential expression analysis (Kharchenko et al, Nature Method, 2014) • monocle: Census based differential analysis (Qiu et al, Nature Methods, 2017) • edgeR: Negative binomial distributions, including empirical Bayes estimation, exact tests, generalized linear models and quasi-likelihood tests based differential analysis (McCarthy et al, Nucleic Acids Research, 2012) • DESingle: Zero-Inflated Negative Binomial model to estimate the proportion of real and dropout zeros and to define and detect the 3 types of DE genes (Miao et al, Bioinformatics, 2018) • MAST: GLM-framework that treates cellular detection rate as a covariate (Finak et al, Genome Biology, 2015)
`min_gene_expressed`	Genes expressed in minimum number of cells
`min_valid_cells`	Minimum number of genes detected in the cell
`contrast`	String vector specifying the contrast to be tested against the log2-fold-change threshold
`q_cut`	Cut-off for q value
`add`	Whether add genes that are not differentially expressed but highly expressed for finding the significant pairs later
`top`	Same as in function rawParse
`stats`	Same as in function rawParse
`...`	Additional arguments passed to the specific differential expression test function

Value

A matrix of the differential expressed genes

`DESeq2Test` *Differential expression using DESeq2*

Description

Identifies differentially expressed genes between two groups of cells using DESeq2

Usage

```
DESeq2Test(
  sub_data,
  min_gene_expressed,
  min_valid_cells,
  contrast = unique(sub_data$compare_group),
  test = "Wald",
  fitType = "parametric",
  sfType = "ratio",
  betaPrior = FALSE,
  quiet = FALSE,
  modelMatrixType = "standard",
  minReplicatesForReplace = 7,
  useT = FALSE,
  minmu = 0.5,
  parallel = FALSE,
  BPPARAM = NULL
)
```

Arguments

`sub_data`	Count data removed cell_type and selected certain two compare_group
`min_gene_expressed`	Genes expressed in minimum number of cells
`min_valid_cells`	Minimum number of genes detected in the cell
`contrast`	String vector specifying the contrast to be tested against the log2-fold-change threshold
`test`	either "Wald" or "LRT", which will then use either Wald significance tests (defined by `nbinomWaldTest`), or the likelihood ratio test on the difference in deviance between a full and reduced model formula (defined by `nbinomLRT`)
`fitType`	either "parametric", "local", or "mean" for the type of fitting of dispersions to the mean intensity. See `estimateDispersions` in DESeq2 for description.
`sfType`	either "ratio", "poscounts", or "iterate" for the type of size factor estimation. See `estimateSizeFactors` in DESeq2 for description.

`betaPrior`	whether or not to put a zero-mean normal prior on the non-intercept coefficients. See `nbinomWaldTest` for description of the calculation of the beta prior. In versions `>=1.16`, the default is set to `FALSE`, and shrunken LFCs are obtained afterwards using `lfcShrink`.
`quiet`	whether to print messages at each step
`modelMatrixType`	either "standard" or "expanded", which describe how the model matrix, X of the GLM formula is formed. "standard" is as created by `model.matrix` using the design formula. "expanded" includes an indicator variable for each level of factors in addition to an intercept. For more information see the Description of `nbinomWaldTest`. betaPrior must be set to TRUE in order for expanded model matrices to be fit.
`minReplicatesForReplace`	the minimum number of replicates required in order to use `replaceOutliers` on a sample. If there are samples with so many replicates, the model will be refit after these replacing outliers, flagged by Cook's distance. Set to `Inf` in order to never replace outliers.
`useT`	logical, passed to `nbinomWaldTest`, default is FALSE, where Wald statistics are assumed to follow a standard Normal
`minmu`	lower bound on the estimated count for fitting gene-wise dispersion and for use with `nbinomWaldTest` and `nbinomLRT`
`parallel`	if FALSE, no parallelization. if TRUE, parallel execution using BiocParallel, see next argument `BPPARAM`. A note on running in parallel using BiocParallel: it may be advantageous to remove large, unneeded objects from your current R environment before calling `DESeq`, as it is possible that R's internal garbage collection will copy these files while running on worker nodes.
`BPPARAM`	an optional parameter object passed internally to `bplapply` when `parallel=TRUE`. If not specified, the parameters last registered with `register` will be used.

Details

This test does not support pre-processed genes. To use this method, please install DESeq2, using the instructions at https://bioconductor.org/packages/release/bioc/html/DESeq2.html

Value

A matrix of differentially expressed genes and related statistics.

References

Love MI, Huber W and Anders S (2014). "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2." Genome Biology. https://bioconductor.org/packages/release/bioc/html/DESeq2.html

---

`DESingleTest` *Differential expression using DEsingle*

Description

Identifies differentially expressed genes between two groups of cells using DEsingle

Usage

```
DESingleTest(
  sub_data,
  min_gene_expressed,
  min_valid_cells,
  contrast = unique(sub_data$compare_group),
  parallel = FALSE,
  BPPARAM = NULL
)
```

Arguments

`sub_data`	Count data removed cell_type and selected certain two compare_group
`min_gene_expressed`	Genes expressed in minimum number of cells
`min_valid_cells`	Minimum number of genes detected in the cell
`contrast`	String vector specifying the contrast to be tested against the log2-fold-change threshold
`parallel`	If FALSE (default), no parallel computation is used; if TRUE, parallel computation using BiocParallel, with argument `BPPARAM`.
`BPPARAM`	An optional parameter object passed internally to `bplapply` when `parallel=TRUE`. If not specified, `bpparam()` (default) will be used.

Details

This test does not support pre-processed genes. To use this method, please install DEsingle, using the instructions at https://github.com/miaozhun/DEsingle

Value

A matrix of differentially expressed genes and related statistics.

References

Zhun Miao, Ke Deng, Xiaowo Wang, Xuegong Zhang (2018). DEsingle for detecting three types of differential expression in single-cell RNA-seq data. Bioinformatics, bty332. 10.1093/bioinformatics/bty332.

`detect_species` *Detect Species from Gene Names*

Description

Automatically detects species based on gene naming patterns with high confidence. Uses statistical analysis of naming conventions to distinguish human vs mouse genes.

Usage

```
detect_species(genes, confidence_threshold = 0.7)
```

Arguments

`genes`	Character vector. Gene symbols to analyze
`confidence_threshold`	Numeric. Minimum confidence score (0-1) to return a species determination (default: 0.7)

Details

Detection Logic:

- Human genes: ALL UPPERCASE (TGFB1, VEGFA, CD8A)
- Mouse genes: First letter uppercase, rest lowercase (Tgfb1, Vegfa, Cd8a)

Analyzes up to 100 genes and calculates proportion matching each pattern. Species is determined if confidence exceeds threshold (default 70

Marker Gene Validation (future enhancement): Could be enhanced to check for species-specific marker genes like:

- Human-specific: HBA1, HBB (hemoglobin)
- Mouse-specific: Gm genes (predicted genes)

Value

List with elements:

- species: "Homo_sapiens", "Mus_musculus", or "unknown"
- confidence: Confidence score (0-1)
- method: Detection method used
- patterns: List of pattern statistics

Examples

```
## Not run:
# Detect human genes
detect_species(c("TGFB1", "VEGFA", "CTNNB1"))
# Returns: list(species = "Homo_sapiens", confidence = 1.0)

# Detect mouse genes
detect_species(c("Tgfb1", "Vegfa", "Ctnnb1"))
# Returns: list(species = "Mus_musculus", confidence = 1.0)

# Mixed or ambiguous
detect_species(c("TGFB1", "Vegfa", "CD8A", "Ctnnb1"))
# Returns: list(species = "unknown", confidence = 0.5)

## End(Not run)
```

`edgeRTest` *Differential expression using edgeR*

Description

Identifies differentially expressed genes between two groups of cells using edgeR

Usage

```
edgeRTest(
  sub_data,
  min_gene_expressed,
  min_valid_cells,
  contrast = unique(sub_data$compare_group),
  calcNormMethod = "TMM",
  trend.method = "locfit",
  tagwise = TRUE,
  robust = FALSE
)
```

Arguments

`sub_data`	Count data removed cell_type and selected certain two compare_group
`min_gene_expressed`	Genes expressed in minimum number of cells
`min_valid_cells`	Minimum number of genes detected in the cell
`contrast`	String vector specifying the contrast to be tested against the log2-fold-change threshold
`calcNormMethod`	normalization method to be used

`trend.method`	method for estimating dispersion trend. Possible values are "none", "movingave", "loess" and "locfit" (default).
`tagwise`	logical, should the tagwise dispersions be estimated
`robust`	logical, should the estimation of prior.df be robustified against outliers

Details

This test does not support pre-processed genes. To use this method, please install edgeR, using the instructions at http://bioconductor.org/packages/release/bioc/html/edgeR.html

Value

A matrix of differentially expressed genes and related statistics.

References

McCarthy, J. D, Chen, Yunshun, Smyth, K. G (2012). “Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation.” Nucleic Acids Research, 40(10), 4288-4297.

Robinson MD, McCarthy DJ, Smyth GK (2010). “edgeR: a Bioconductor package for differential expression analysis of digital gene expression data.” Bioinformatics, 26(1), 139-140. https://github.com/cole-trapnell-lab/monocle-release

---

`FindLR` *Finding ligand-receptor pairs*

Description

This function loads the highly expressed genes or differentail expressed genes as a dataframe. Significant interactions are found through mapping these genes to our ligand-receptor database.

Usage

```
FindLR(
  data_1,
  data_2 = NULL,
  datatype,
  comm_type,
  database = NULL,
  convert_species = TRUE,
  ensembl_version = 103,
  mirror = NULL,
  cache = TRUE
)
```

Arguments

`data_1`	Data used to find the ligand-receptor pairs
`data_2`	Second dataset used to find ligand-receptor pairs. If set NULL, paris will be found within data_1. Otherwise, pairs will be found between data_1 and data_2. Default is NULL.
`datatype`	Type of data used as input. Options are "mean count" and "DEG"
`comm_type`	Communication type. Available options are "cytokine", "checkpoint", "growth factor", "other"
`database`	Database used to find ligand-receptor pairs. If set NULL, the build-in database will be used.
`convert_species`	Logical. Enable automatic species conversion (default: TRUE). When TRUE, automatically detects mouse genes and converts to human orthologs.
`ensembl_version`	Ensembl version for gene conversion (default: 103)
`mirror`	Ensembl mirror for faster access (default: NULL)
`cache`	Cache conversion results (default: TRUE)

Value

A dataframe of the significant interactions

References

Cytokines, Inflammation and Pain. Zhang et al,2007.

Cytokines, Chemokines and Their Receptors. Cameron et al, 2000-2013

Robust prediction of response to immune checkpoint blockade therapy in metastatic melanoma. Auslander et al, 2018.

A draft network of ligand-receptor-mediated multicellular signalling in human, Jordan A. Ramilowski, Nature Communications, 2015

`LRPlot` *Plotting ligand-receptor pairs*

Description

This function loads the significant interactions as a dataframe. A circle plot will be generated using package circlize. The width of the arrow represents the expression level/log fold change of the ligand; while the width of arrow head represents the expression level/log fold change of the receptor. Different color and the type of the arrow stands for whether the ligand and/or receptor are upregulated or downregulated. Users can select the colors represent the cell type by their own or chosen randomly by default.

Usage

```
LRPlot(
  data,
  datatype,
  gene_col = NULL,
  transparency = 0.5,
  link.arr.lwd = 1,
  link.arr.lty = NULL,
  link.arr.col = NULL,
  link.arr.width = NULL,
  link.arr.type = NULL,
  facing = "clockwise",
  cell_col = NULL,
  print.cell = TRUE,
  track.height_1 = uh(2, "mm"),
  track.height_2 = uh(12, "mm"),
  annotation.height_1 = 0.01,
  annotation.height_2 = 0.01,
  text.vjust = "0.4cm",
  ...
)
```

Arguments

`data`	A dataframe contains significant ligand-receptor pairs and related information such as expression level/log fold change and cell type
`datatype`	Type of data. Options are "mean count" and "DEG"
`gene_col`	Colors used to represent different categories of genes.
`transparency`	Transparency of link colors, 0 means no transparency and 1 means full transparency. If transparency is already set in col or row.col or column.col, this argument will be ignored. NAalso ignores this argument.
`link.arr.lwd`	line width of the single line link which is put in the center of the belt.
`link.arr.lty`	line type of the single line link which is put in the center of the belt.
`link.arr.col`	color or the single line link which is put in the center of the belt.
`link.arr.width`	size of the single arrow head link which is put in the center of the belt.
`link.arr.type`	Type of the arrows, pass to Arrowhead. Default value is triangle. There is an additional option big.arrow
`facing`	Facing of text.
`cell_col`	Colors used to represent types of cells. If set NULL, it will be generated randomly
`print.cell`	Whether or not print the type of cells on the outer layer of the graph.
`track.height_1`	height of the cell notation track
`track.height_2`	height of the gene notation track

`annotation.height_1`	Track height corresponding to values in annotationTrack.
`annotation.height_2`	Track height corresponding to values in annotationTrack.
`text.vjust`	adjustment on 'vertical' (radical) direction. Besides to set it as numeric values, the value can also be a string contain absoute unit, e.g. "2.1mm", "-1 inche", but only "mm", "cm", "inches"/"inche" are allowed.
`...`	Additional arguments passed to circlize plotting functions

Value

A figure of the significant interactions

References

Gu, Z. (2014) circlize implements and enhances circular visualization in R. Bioinformatics.

`MASTTest` *Differential expression using MAST*

Description

Identifies differentially expressed genes between two groups of cells using MAST

Usage

```
MASTTest(
  sub_data,
  min_gene_expressed,
  min_valid_cells,
  contrast = unique(sub_data$compare_group),
  method = "glm",
  silent = FALSE,
  check_logged = TRUE
)
```

Arguments

`sub_data`	Count data removed cell_type and selected certain two compare_group
`min_gene_expressed`	Genes expressed in minimum number of cells
`min_valid_cells`	Minimum number of genes detected in the cell
`contrast`	String vector specifying the contrast to be tested against the log2-fold-change threshold
`method`	Character vector, either 'glm', 'glmer' or 'bayesglm'

`silent`	Common problems with fitting some genes
`check_logged`	Set FALSE to override sanity checks that try to ensure that the default assay is log-transformed and has at least one exact zero

Details

To use this method, please install MAST, using the instructions at https://github.com/RGLab/MAST

Value

A matrix of differentially expressed genes and related statistics.

References

MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data G Finak, A McDavid, M Yajima, J Deng, V Gersuk, AK Shalek, CK Slichter et al Genome biology 16 (1), 278

`MonocleTest` *Differential expression using monocle*

Description

Identifies differentially expressed genes between two groups of cells using monocle

Usage

```
MonocleTest(
  sub_data,
  min_gene_expressed,
  min_valid_cells,
  contrast = unique(sub_data$compare_group),
  batch = NULL,
  cores = 4
)
```

Arguments

`sub_data`	Count data removed cell_type and selected certain two compare_group
`min_gene_expressed`	Genes expressed in minimum number of cells
`min_valid_cells`	Minimum number of genes detected in the cell
`contrast`	String vector specifying the contrast to be tested against the log2-fold-change threshold
`batch`	Different batch identifier
`cores`	The number of cores to be used while testing each gene for differential expression.

Details

This test does not support pre-processed genes. To use this method, please install monocle, using the instructions at https://bioconductor.org/packages/release/bioc/html/monocle.html

Value

A matrix of differentially expressed genes and related statistics.

References

Qiu X, Hill A, Packer J, Lin D, Ma Y, Trapnell C (2017). “Single-cell mRNA quantification and differential analysis with Census.” Nature Methods. https://github.com/cole-trapnell-lab/monocle-release

---

`NetView` *Network Viewing of cell-cell communication*

Description

This function loads the significant interactions as a dataframe, and colors represent different types of cells as a structure. The width of edges represent the strength of the communication. Labels on the edges show exactly how many interactions exist between two types of cells.

Usage

```
NetView(
  data,
  col,
  label = TRUE,
  edge.curved = 0.5,
  shape = "circle",
  layout = igraph::nicely(),
  vertex.size = 20,
  margin = 0.2,
  vertex.label.cex = 1.5,
  vertex.label.color = "black",
  arrow.width = 1.5,
  edge.label.color = "black",
  edge.label.cex = 1,
  edge.max.width = 10
)
```

Arguments

`data`	A dataframe containing ligand-receptor pairs and corresponding cell typesused to do the plotting
`col`	Colors used to represent different cell types

`label`	Whether or not shows the label of edges (number of connections between different cell types)
`edge.curved`	Specifies whether to draw curved edges, or not. This can be a logical or a numeric vector or scalar. First the vector is replicated to have the same length as the number of edges in the graph. Then it is interpreted for each edge separately. A numeric value specifies the curvature of the edge; zero curvature means straight edges, negative values means the edge bends clockwise, positive values the opposite. TRUE means curvature 0.5, FALSE means curvature zero
`shape`	The shape of the vertex, currently “circle”, “square”, “csquare”, “rectangle”, “crectangle”, “vrectangle”, “pie” (see vertex.shape.pie), ‘sphere’, and “none” are supported, and only by the plot.igraph command. “none” does not draw the vertices at all, although vertex label are plotted (if given). See shapes for details about vertex shapes and vertex.shape.pie for using pie charts as vertices.
`layout`	The layout specification. It must be a call to a layout specification function.
`vertex.size`	The size of vertex
`margin`	The amount of empty space below, over, at the left and right of the plot, it is a numeric vector of length four. Usually values between 0 and 0.5 are meaningful, but negative values are also possible, that will make the plot zoom in to a part of the graph. If it is shorter than four then it is recycled.
`vertex.label.cex`	The label size of vertex
`vertex.label.color`	The color of label for vertex
`arrow.width`	The width of arrows
`edge.label.color`	The color for single arrow
`edge.label.cex`	The size of label for arrows
`edge.max.width`	The maximum arrow size

Value

A network graph of the significant interactions

References

Csardi G, Nepusz T: The igraph software package for complex network research, InterJournal, Complex Systems 1695. 2006. http://igraph.org

---

`rawParse` *Parsing the data to get top expressed genes*

Description

This function loads the count data as a dataframe. It assumes that each line contains gene expression profile of one single cell, and each column contains the one single gene expression profile in different cells. The dataframe should also contain the cell type information with column name 'cell_type'. Group information should also be included as 'compare_group' if users want to call differntial expressed ligand-receptor pairs. Batch information as 'batch' is optional. If included, users may want to use the raw count data for later analysis.

Usage

```
rawParse(data, top_genes = 50, stats = "mean")
```

Arguments

`data`	Input data, raw or normalized count with 'cell_type' column
`top_genes`	(scale 1 to 100) Top percent highly expressed genes used to find ligand-receptor pairs, default is 50
`stats`	Whether calculates the mean or the median of the data. Available options are 'mean' and 'median'.

Value

A dataframe of the data

`SCDETest` *Differential expression using scde*

Description

Identifies differentially expressed genes between two groups of cells using scde

Usage

```
SCDETest(
  sub_data,
  min_gene_expressed,
  min_valid_cells,
  contrast = unique(sub_data$compare_group),
  batch = NULL,
  n.randomizations = 150,
  n.cores = 10,
```

```
  batch.models = NULL,
  return.posteriors = FALSE,
  verbose = 1
)
```

Arguments

`sub_data`	Count data removed cell_type and selected certain two compare_group
`min_gene_expressed`	Genes expressed in minimum number of cells
`min_valid_cells`	Minimum number of genes detected in the cell
`contrast`	String vector specifying the contrast to be tested against the log2-fold-change threshold
`batch`	Different batch identifier
`n.randomizations`	number of bootstrap randomizations to be performed
`n.cores`	number of cores to utilize
`batch.models`	(optional) separate models for the batch data (if generated using batch-specific group argument). Normally the same models are used.
`return.posteriors`	whether joint posterior matrices should be returned
`verbose`	integer verbose level (1 for verbose)

Details

This test does not support pre-processed genes. To use this method, please install scde, using the instructions at http://hms-dbmi.github.io/scde/tutorials.html

Value

A matrix of differentially expressed genes and related statistics.

References

"Bayesian approach to single-cell differential expression analysis" (Kharchenko PV, Silberstein L, Scadden DT, Nature Methods, doi:10.1038/nmeth.2967) https://github.com/hms-dbmi/scde

---

`species_conversion` *Species Conversion System for iTALK*

Description

Complete species conversion framework enabling iTALK to work seamlessly with mouse, human, and other species data by mapping genes to orthologs using Ensembl BioMart.

Details

iTALK's ligand-receptor database uses human gene symbols (e.g., TGFB1, VEGFA). This module automatically detects input species and converts gene names to human orthologs for database matching, then optionally converts results back.

`TimePlot` *Plotting ligand-receptor pairs*

Description

This function loads count data as dataframe, ligand, receptor and two interactive cells' names as strings. The plot shows the expression level of ligand and receptor at different time, thus illustrates a dynamic change of a ligand-receptor pairs.

Usage

```
TimePlot(data, ligand, receptor, cell_from, cell_to, Time = NULL)
```

Arguments

`data`	A dataframe contains significant ligand-receptor pairs and related information such as expression level/log fold change and cell type
`ligand`	String as selected ligand
`receptor`	String as selected receptor
`cell_from`	The cell type ligand gene belongs to
`cell_to`	The cell type receptor gene belongs to
`Time`	Different time points showing on the plot

Value

A figure of the paired interactions

`WilcoxTest` *Differential expression using wilcox*

Description

Identifies differentially expressed genes between two groups of cells using a Wilcoxon Rank Sum test

Usage

```
WilcoxTest(
  sub_data,
  min_gene_expressed,
  min_valid_cells,
  contrast = unique(sub_data$compare_group),
  datatype = "raw count",
  verbose = FALSE
)
```

Arguments

`sub_data`	Count data removed cell_type and selected certain two compare_group
`min_gene_expressed`	Genes expressed in minimum number of cells
`min_valid_cells`	Minimum number of genes detected in the cell
`contrast`	String vector specifying the contrast to be tested against the log2-fold-change threshold
`datatype`	Type of data. Available options are: • 'raw data': Raw count data without any pre-processing • 'log count': Normalized and log-transformed data
`verbose`	Whether show the progress of computing

Value

A matrix of differentially expressed genes and related statistics.

Index