Package: scMetaLink (via r-universe)

May 28, 2026

Type Package

Title Single-Cell Metabolite-Mediated Cell Communication Analysis

Version 0.99.1

Date 2026-01-26

Author Zaoqu Liu [aut, cre] (<https://orcid.org/0000-0002-0452-742X>)

Maintainer Zaoqu Liu <liuzaoqu@163.com>

Description A comprehensive framework for inferring metabolite-mediated cell-cell communication from single-cell transcriptomic data. scMetaLink integrates metabolite production potential via enzyme expression, metabolite sensing capability via receptor and transporter expression, and secretion potential to construct intercellular metabolic communication networks. The package leverages the MetalinksDB database containing 41894 metabolite-protein interactions covering 1128 metabolites and 4374 proteins. Key features include probabilistic inference of metabolite production, receptor-mediated metabolite sensing quantification, permutation-based statistical testing with multiple hypothesis correction, pathway-level aggregation analysis, and publication-ready visualization.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

LazyDataCompression xz

Depends R (>= 4.0.0)

Imports methods, stats, utils, grDevices, graphics, Matrix, ggplot2, ggraph, igraph, scales, circlize, ComplexHeatmap, grid, viridis, RColorBrewer, future, future.apply

Suggests Seurat (>= 4.0.0), SingleCellExperiment, SummarizedExperiment, testthat (>= 3.0.0), knitr, rmarkdown, BiocStyle, reshape2, pkgdown

VignetteBuilder knitr

biocViews Software, SingleCell, Transcriptomics, CellBiology, Metabolomics, Network, Visualization, GeneExpression, Spatial

RoxygenNote 7.3.3

Config/testthat/edition 3

URL https://github.com/Zaoqu-Liu/scMetaLink, https://Zaoqu-Liu.github.io/scMetaLink/

BugReports https://github.com/Zaoqu-Liu/scMetaLink/issues

Config/pak/sysreqs libfontconfig1-dev libfreetype6-dev libglpk-dev libicu-dev libpng-dev libxml2-dev perl

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

Date/Publication 2026-01-26 15:42:34 UTC

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

RemoteRef HEAD

RemoteSha d5e03bd61e72741e7fce77da7980a68261c230e3

Contents

scMetaLink-package . . . 4
accessors . . . 5
aggregateByPathway . . . 6
checkLactateGenes . . . 7
citationScMetaLink . . . 7
compareCommunication . . . 8
computeCommunication . . . 9
computeSpatialCommunication . . . 10
crc_expr . . . 13
crc_meta . . . 14
createScMetaLink . . . 14
createScMetaLinkFromSCE . . . 15
createScMetaLinkFromSeurat . . . 16
createScMetaLinkFromSeuratSpatial . . . 17
createScMetaLinkFromSpatial . . . 17
enrichPathways . . . 19
exportResults . . . 20
filterSignificantInteractions . . . 20
getCommunicationMatrix . . . 21
getDatabaseInfo . . . 22
getLactateGenes . . . 22
getLactateSignalingSummary . . . 23
getMetaboliteReceptors . . . 24
getPathwayCommunicationMatrix . . . 24
getPathwayMetaboliteNetwork . . . 25
getPathwayMetabolites . . . 25
getSpatialDistanceStats . . . 26

getSpatialLactateHotspots . . . 26
getSummaryStats . . . 27
getTopLactateProducers . . . 27
getTopLactateSensors . . . 28
getTopProducers . . . 28
getTopSensors . . . 29
identifyCellTypeSpecificMetabolites . . . 29
identifyCommunicationHotspots . . . 30
inferLactateSignaling . . . 31
inferProduction . . . 33
inferSensing . . . 34
inferSpatialLactateSignaling . . . 35
listGenes . . . 37
listMetabolites . . . 38
listTopPathways . . . 38
loadScMetaLink . . . 39
metalinksdb . . . 39
plotCommunicationCircle . . . 41
plotCommunicationHeatmap . . . 41
plotCommunicationNetwork . . . 42
plotDifferentialCommunication . . . 43
plotEnrichedPathways . . . 44
plotLactatePathwayComparison . . . 44
plotLactateSignaling . . . 45
plotMetaboliteProfile . . . 46
plotPathwayCommunication . . . 46
plotSpatialCellTypes . . . 47
plotSpatialCommunicationNetwork . . . 47
plotSpatialComparison . . . 48
plotSpatialDistanceDistribution . . . 49
plotSpatialFeature . . . 50
plotSpatialHotspots . . . 51
plotSpatialLactate . . . 52
plotTopInteractions . . . 52
runScMetaLink . . . 53
runScMetaLinkSeurat . . . 54
saveScMetaLink . . . 55
scMetaLink-class . . . 55
searchGene . . . 56
searchMetabolite . . . 57
st_expr . . . 57
st_meta . . . 58
st_scalefactors . . . 59
summarizeCommunicationPairs . . . 60
summarizePathwayActivity . . . 61

Index **62**

scMetaLink-package *scMetaLink: Single-Cell Metabolite-Mediated Cell Communication Analysis*

Description

A comprehensive framework for inferring metabolite-mediated cell-cell communication from single-cell transcriptomic data. scMetaLink integrates metabolite production potential via enzyme expression, metabolite sensing capability via receptor and transporter expression, and secretion potential to construct intercellular metabolic communication networks.

Details

The package provides the following main functions:

- `createScMetaLink`: Create analysis object from expression data
- `inferProduction`: Infer metabolite production potential
- `inferSensing`: Infer metabolite sensing capability
- `computeCommunication`: Compute cell-cell communication scores
- `filterSignificantInteractions`: Filter significant interactions
- `aggregateByPathway`: Aggregate by metabolic pathways
- `runScMetaLink`: Run complete analysis pipeline

For spatial transcriptomics data:

- `createScMetaLinkFromSpatial`: Create object from spatial data
- `computeSpatialCommunication`: Compute spatially-weighted communication

Database

scMetaLink utilizes MetalinksDB, containing:

- 41,894 metabolite-protein interactions
- 1,128 metabolites
- 4,374 proteins/genes
- 157,741 pathway associations

Author(s)

Zaoqu Liu <liuzaoqu@163.com>

See Also

Useful links:

- https://github.com/Zaoqu-Liu/scMetaLink
- https://Zaoqu-Liu.github.io/scMetaLink/
- Report bugs at https://github.com/Zaoqu-Liu/scMetaLink/issues

accessors *Accessor Functions for scMetaLink Objects*

Description

Functions to extract data from scMetaLink objects.

Usage

```
getProductionScores(object)

## S4 method for signature 'scMetaLink'
getProductionScores(object)

getSensingScores(object)

## S4 method for signature 'scMetaLink'
getSensingScores(object)

getCommunicationScores(object)

## S4 method for signature 'scMetaLink'
getCommunicationScores(object)

getSignificantInteractions(object)

## S4 method for signature 'scMetaLink'
getSignificantInteractions(object)

getPathwayAggregated(object)

## S4 method for signature 'scMetaLink'
getPathwayAggregated(object)

getParameters(object)

## S4 method for signature 'scMetaLink'
getParameters(object)
```

Arguments

object	A scMetaLink object

Value

The requested data from the scMetaLink object:

- getProductionScores: Matrix of metabolite production scores

- `getSensingScores`: Matrix of metabolite sensing scores
- `getCommunicationScores`: 3D array of communication scores
- `getSignificantInteractions`: data.frame of significant interactions
- `getPathwayAggregated`: data.frame of pathway-aggregated results
- `getParameters`: list of analysis parameters

Examples

```
data(crc_example)
obj <- createScMetaLink(crc_expr, crc_meta, "cell_type")
obj <- inferProduction(obj)
prod_scores <- getProductionScores(obj)
```

`aggregateByPathway` *Aggregate Communication by Pathway*

Description

Aggregate metabolite-mediated communication at the pathway level. Due to the large number of pathway associations, this function uses a simplified approach focusing on top pathways.

Usage

```
aggregateByPathway(object, top_pathways = 50, min_metabolites = 3)
```

Arguments

`object`	A scMetaLink object with significant interactions
`top_pathways`	Integer. Number of top pathways to analyze (default: 50)
`min_metabolites`	Integer. Minimum metabolites per pathway (default: 3)

Value

Updated scMetaLink object with pathway_aggregated slot filled

checkLactateGenes *Check Lactate Gene Availability*

Description

Checks which lactate signaling genes are available in the expression data. Useful for quality control before running analysis.

Usage

```
checkLactateGenes(object)
```

Arguments

object	scMetaLink object

Value

data.frame showing gene availability by category

Examples

```
data(crc_example)
obj <- createScMetaLink(crc_expr, crc_meta, "cell_type")
checkLactateGenes(obj)
```

citationScMetaLink *Show Package Citation*

Description

Display the recommended citation for scMetaLink

Usage

```
citationScMetaLink()
```

Value

Invisibly returns NULL, prints citation information

Examples

```
citationScMetaLink()
```

`compareCommunication` *Compare Two Conditions*

Description

Compare metabolite-mediated communication between two conditions

Usage

```
compareCommunication(
  object1,
  object2,
  condition_names = c("Condition1", "Condition2"),
  method = "log2fc"
)
```

Arguments

`object1`	scMetaLink object for condition 1
`object2`	scMetaLink object for condition 2
`condition_names`	Character vector of length 2 for condition names
`method`	Character. Comparison method: "difference", "ratio", or "log2fc"

Value

data.frame with differential communication results

Examples

```
## Not run:
# Compare tumor vs normal (requires two scMetaLink objects)
diff <- compareCommunication(tumor_obj, normal_obj,
  condition_names = c("Tumor", "Normal")
)

## End(Not run)
```

`computeCommunication` *Compute Metabolite-Mediated Cell Communication*

Description

Calculate communication scores between all cell type pairs mediated by metabolites. Communication strength represents the potential for signal transmission from sender to receiver cells via metabolites.

Usage

```
computeCommunication(
  object,
  method = "geometric",
  min_production = 0.1,
  min_sensing = 0.1,
  population.size = FALSE,
  n_permutations = 100,
  n_cores = 1,
  seed = 42,
  verbose = TRUE
)
```

Arguments

`object`	A scMetaLink object with production and sensing scores
`method`	Character. Communication score method: "geometric" (default), "product", "harmonic"
`min_production`	Numeric. Minimum production score threshold (0-1)
`min_sensing`	Numeric. Minimum sensing score threshold (0-1)
`population.size`	Logical. Whether to weight communication by cell type population sizes. When TRUE, communication strength is scaled by the relative abundance of sender and receiver cell types, reflecting the biological reality that larger populations contribute more to overall tissue signaling.
`n_permutations`	Integer. Number of permutations for significance testing (0 to skip)
`n_cores`	Integer. Number of cores for parallel computing
`seed`	Integer. Random seed for reproducibility
`verbose`	Logical. Print progress messages

Details

The communication score combines production and sensing capabilities:

- geometric: sqrt(production * sensing) - balanced measure

- product: production * sensing - emphasizes strong bilateral signals
- harmonic: 2*production*sensing/(production+sensing) - penalizes imbalance

When `population.size = TRUE`, scores are multiplied by: sqrt(n_sender/n_total * n_receiver/n_total), which accounts for the relative contribution of each cell type to tissue-level communication.

Value

Updated scMetaLink object with communication_scores and p-values

---

`computeSpatialCommunication` *Compute Spatial Communication*

Description

Calculate spatially-weighted metabolite-mediated communication. This function incorporates spatial distance between spots to weight communication strength.

Usage

```
computeSpatialCommunication(
  object,
  method = "knn",
  k_neighbors = 6,
  symmetric = TRUE,
  distance_threshold = 150,
  sigma = 50,
  lambda = 50,
  comm_method = "geometric",
  min_production = 0.1,
  min_sensing = 0.1,
  analysis_level = "region",
  n_permutations = 1000,
  n_cores = 1,
  seed = 42,
  verbose = TRUE
)
```

Arguments

`object`	A scMetaLink object with spatial information and production/sensing scores
`method`	Character. Spatial weighting method: • "knn": K-nearest neighbors only (recommended for Visium, most honest given the resolution limitations) • "gaussian": Gaussian decay exp(-d^2/2*sigma^2)

- "exponential": Exponential decay exp(-d/lambda)
- "linear": Linear decay max(0, 1 - d/d_max)
- "threshold": Binary cutoff (1 if d <= threshold, 0 otherwise)

`k_neighbors` Integer. Number of nearest neighbors for knn method. Default: 6. For Visium hexagonal grid, 6 corresponds to immediate neighbors. **Note**: The weight matrix is symmetrized, so actual neighbor count per spot may exceed k (typically k to 2k).

`symmetric` Logical. Whether to symmetrize the KNN weight matrix (default: TRUE). If TRUE, when spot A is a neighbor of B, B is also considered a neighbor of A. This ensures bidirectional communication potential.

`distance_threshold`
Numeric. Maximum distance for communication in micrometers. Spot pairs beyond this distance are considered non-interacting. Default: 150 um. Note: Most metabolites have effective diffusion ranges of 10-200 um in tissue. For Visium data (100 um spot spacing), values >200 um are rarely meaningful.

`sigma` Numeric. Sigma parameter for Gaussian decay (in um). Default: 50 um. Represents the characteristic decay distance. Literature suggests:

- Fast-turnover metabolites (adenosine, ATP): 20-30 um
- Medium-range metabolites (lactate): 50-80 um
- Stable metabolites (amino acids): 80-120 um

`lambda` Numeric. Lambda parameter for exponential decay (in um). Default: 50 um.

`comm_method` Character. Communication score method: "geometric", "product", "harmonic"

`min_production` Numeric. Minimum production score threshold (0-1). Cell types with production score below this are considered non-producers.

`min_sensing` Numeric. Minimum sensing score threshold (0-1). Cell types with sensing score below this are considered non-sensors.

`analysis_level` Character. Level of analysis:

- "region": Aggregate by cell type (default, recommended). Uses the full inferProduction/inferSensing logic.
- "spot": Compute spot-to-spot communication. **WARNING**: This is a simplified implementation for exploratory analysis. It does NOT include: degradation adjustment, secretion weighting, Hill transformation, or full normalization. Use for hotspot identification only.

`n_permutations` Integer. Number of permutations for significance testing. Default: 1000. For publication, recommend >= 1000 permutations. The permutation test shuffles cell type labels while preserving spatial structure, then **recalculates production and sensing scores** from scratch (consistent with non-spatial version), which is the scientifically correct null model for testing whether communication patterns are associated with cell type identity.

`n_cores` Integer. Number of cores for parallel computing

`seed` Integer. Random seed for reproducibility

`verbose` Logical. Print progress messages

Details

Important Notes on Spatial Resolution:

For Visium data (55 um spots, ~100 um spacing), each spot contains 1-10 cells. This means:

- Cell type annotations should ideally come from deconvolution methods (e.g., RCTD, cell2location, SPOTlight)
- The "knn" method (k=6) is recommended as it only considers immediate neighbors, which is most honest given the resolution limitations
- Distance-weighted methods (gaussian, exponential) may provide "false precision" when spot spacing is similar to metabolite diffusion distance

Permutation Test Design:

The permutation test shuffles cell type labels (not spatial positions) and **recalculates production/sensing scores from expression data**. This is consistent with the non-spatial version and tests whether the observed communication pattern is associated with cell type identity beyond what would be expected by chance.

Spatial communication is modeled as:

$$C_{spatial}(i->j,m) = MPP(m,i) * MSC(m,j) * w(d_{ij})$$

where w(d) is the spatial weight function:

- Gaussian: w(d) = exp(-d^2 / 2*sigma^2)
- Exponential: w(d) = exp(-d / lambda)
- Linear: w(d) = max(0, 1 - d/d_max)
- Threshold: w(d) = I(d <= d_threshold)

Value

Updated scMetaLink object with spatial_communication slot

Examples

```
data(st_expr)
data(st_meta)
data(st_scalefactors)

# Create object and run analysis
obj <- createScMetaLinkFromSpatial(st_expr, st_meta[,c("x","y")],
                                   st_meta, "cell_type", st_scalefactors)
obj <- inferProduction(obj)
obj <- inferSensing(obj)

# Compute spatial communication (knn method recommended for Visium)
obj <- computeSpatialCommunication(obj, method = "knn", k_neighbors = 6)

# Alternative: Gaussian decay with conservative parameters
obj <- computeSpatialCommunication(obj, method = "gaussian",
```

```
                                    sigma = 50,  # 50 um decay
                                    distance_threshold = 150)  # max 150 um
```

`crc_expr` *CRC Example Expression Data*

Description

Example single-cell RNA-seq expression matrix from colorectal cancer

Usage

```
crc_expr
```

Format

A sparse dgCMatrix with 4,210 genes (rows) x 2,850 cells (columns). Only genes present in MetalinksDB are included to reduce file size.

Details

This is a subset of colorectal cancer single-cell data containing:

- Tumor cells: Tumor Epithelial (600 cells)
- Immune cells: T (500), Plasma (250), B (150), TAM (150), Monocyte (120), Normal Macrophage (150), Mast (30)
- Stromal cells: CAF (200), Normal Fibroblast (200), Endothelial (100), Pericyte (50), SMC (30)
- Epithelial: Normal Epithelial (300)
- Other: Gliacyte (20)

Source

CellScope package example data

Examples

```
data(crc_expr)
data(crc_meta)

# Run scMetaLink analysis
obj <- createScMetaLink(crc_expr, crc_meta, "cell_type")
```

crc_meta *CRC Example Cell Metadata*

Description

Cell metadata for the CRC example dataset

Usage

```
crc_meta
```

Format

A data.frame with 2,850 rows (cells) and 3 columns:

cell_type Cell type annotation (15 types)

tumor_normal Tumor or Normal tissue origin

tissue_region Tissue region

Examples

```
data(crc_meta)
table(crc_meta$cell_type)
```

createScMetaLink *Create scMetaLink Object*

Description

Initialize a scMetaLink object from expression data and cell metadata

Usage

```
createScMetaLink(
  expression_data,
  cell_meta,
  cell_type_column = "cell_type",
  min_cells = 10,
  min_genes = 200
)
```

Arguments

expression_data	A matrix or dgCMatrix of normalized expression values (genes x cells)
cell_meta	A data.frame containing cell metadata
cell_type_column	Character. Column name in cell_meta containing cell type annotations
min_cells	Integer. Minimum number of cells per cell type (default: 10)
min_genes	Integer. Minimum number of genes detected per cell (default: 200)

Value

A scMetaLink object

Examples

```
# Create from expression matrix
expr_mat <- matrix(rpois(1000, 5), nrow = 100, ncol = 10)
rownames(expr_mat) <- paste0("Gene", 1:100)
colnames(expr_mat) <- paste0("Cell", 1:10)
meta <- data.frame(cell_type = rep(c("TypeA", "TypeB"), each = 5))
rownames(meta) <- colnames(expr_mat)
obj <- createScMetaLink(expr_mat, meta, "cell_type")
```

---

createScMetaLinkFromSCE

Create scMetaLink from SingleCellExperiment

Description

Initialize a scMetaLink object from a SingleCellExperiment object

Usage

```
createScMetaLinkFromSCE(
  sce,
  cell_type_column = "cell_type",
  assay_name = "logcounts",
  min_cells = 10
)
```

Arguments

sce	A SingleCellExperiment object
cell_type_column	Character. Column name in colData for cell type annotations
assay_name	Character. Name of assay to use (default: "logcounts")
min_cells	Integer. Minimum cells per cell type

Value

A scMetaLink object

Examples

```
if (requireNamespace("SingleCellExperiment", quietly = TRUE)) {
  library(SingleCellExperiment)
  # Create example SCE
  counts <- matrix(rpois(1000, 5), nrow = 100, ncol = 10)
  rownames(counts) <- paste0("Gene", 1:100)
  colnames(counts) <- paste0("Cell", 1:10)
  sce <- SingleCellExperiment(assays = list(counts = counts, logcounts = log1p(counts)))
  colData(sce)$cell_type <- rep(c("TypeA", "TypeB"), each = 5)
  # obj <- createScMetaLinkFromSCE(sce, "cell_type")
}
```

---

createScMetaLinkFromSeurat

Create scMetaLink from Seurat Object

Description

Initialize a scMetaLink object from a Seurat object

Usage

```
createScMetaLinkFromSeurat(
  seurat_obj,
  cell_type_column = "cell_type",
  assay = "RNA",
  slot = "data",
  min_cells = 10
)
```

Arguments

seurat_obj	A Seurat object
cell_type_column	Character. Column name in meta.data for cell type
assay	Character. Assay to use (default: "RNA")
slot	Character. Slot to use (default: "data" for normalized data)
min_cells	Integer. Minimum cells per cell type

Value

A scMetaLink object

createScMetaLinkFromSeuratSpatial *Create scMetaLink from Seurat Spatial Object*

Description

Initialize a scMetaLink object from a Seurat object with spatial data

Usage

```
createScMetaLinkFromSeuratSpatial(
  seurat_obj,
  cell_type_column = "cell_type",
  assay = "Spatial",
  slot = "data",
  image = NULL,
  min_cells = 5
)
```

Arguments

seurat_obj	A Seurat object with spatial assay and coordinates
cell_type_column	Character. Column name in meta.data for cell type
assay	Character. Assay to use (default: "Spatial")
slot	Character. Slot to use (default: "data" for normalized data)
image	Character. Name of the spatial image to use (default: first available)
min_cells	Integer. Minimum spots per cell type

Value

A scMetaLink object with spatial information

createScMetaLinkFromSpatial *Create scMetaLink Object from Spatial Data*

Description

Initialize a scMetaLink object from spatial transcriptomics data with spatial coordinate information.

Usage

```
createScMetaLinkFromSpatial(
  expression_data,
  spatial_coords,
  cell_meta,
  cell_type_column = "cell_type",
  scale_factors = NULL,
  min_cells = 5
)
```

Arguments

expression_data	A matrix or dgCMatrix of normalized expression values (genes x spots)
spatial_coords	A data.frame or matrix with spatial coordinates (spots x 2). Must have row names matching column names of expression_data.
cell_meta	A data.frame containing spot metadata (e.g., cell type from deconvolution)
cell_type_column	Character. Column name in cell_meta containing cell type annotations
scale_factors	List. Optional scale factors for coordinate conversion. Should contain 'pixels_per_um' for distance calculations. **IMPORTANT**: For Visium data, coordinates are in pixels, not micrometers. Without correct scale_factors, distance-based parameters will be wrong.
min_cells	Integer. Minimum number of spots per cell type (default: 5)

Details

This function creates a scMetaLink object with spatial information stored in additional slots. The spatial coordinates are used for distance-weighted communication analysis.

Important: Cell Type Annotation for Visium Data

For 10x Visium data, each spot (55 micrometer diameter) typically contains 1-10 cells of potentially different types. Therefore, cell type annotations should ideally come from deconvolution methods such as:

- RCTD (spacexr package)
- cell2location
- SPOTlight
- CARD

If using dominant cell type assignment per spot, be aware that this is a simplification and may miss important cell type heterogeneity within spots.

Important: Coordinate Units

For 10x Visium data, coordinates are typically in pixels. You MUST provide scale_factors with 'pixels_per_um' to convert to micrometers for biologically meaningful distance calculations. Without this, sigma=50 will be interpreted as 50 pixels instead of 50 micrometers.

Value

A scMetaLink object with spatial information

Examples

```
data(st_expr)
data(st_meta)
data(st_scalefactors)

obj <- createScMetaLinkFromSpatial(
  expression_data = st_expr,
  spatial_coords = st_meta[, c("x", "y")],
  cell_meta = st_meta,
  cell_type_column = "cell_type",
  scale_factors = st_scalefactors
)
```

---

enrichPathways *Pathway Enrichment Analysis*

Description

Perform hypergeometric test to identify enriched pathways among significant metabolite-mediated interactions

Usage

```
enrichPathways(
  object,
  pvalue_threshold = 0.05,
  min_overlap = 2,
  adjust_method = "BH"
)
```

Arguments

object	scMetaLink object with significant interactions
pvalue_threshold	Numeric. P-value cutoff for enrichment (default: 0.05)
min_overlap	Integer. Minimum overlap between significant metabolites and pathway (default: 2)
adjust_method	Character. P-value adjustment method (default: "BH")

Value

data.frame with enriched pathways and statistics

Examples

```
# Perform pathway enrichment
enriched <- enrichPathways(result)
head(enriched)
```

`exportResults` *Export Results to CSV*

Description

Export Results to CSV

Usage

```
exportResults(object, output_dir = ".", prefix = "scMetaLink")
```

Arguments

`object`	scMetaLink object
`output_dir`	Character. Output directory
`prefix`	Character. File prefix

Value

Invisibly returns file paths

`filterSignificantInteractions` *Filter Significant Interactions*

Description

Filter Significant Interactions

Usage

```
filterSignificantInteractions(
  object,
  pvalue_threshold = 0.05,
  adjust_method = "metabolite_stratified",
  min_score = 0
)
```

Arguments

object	scMetaLink object
pvalue_threshold	Numeric. P-value threshold (default: 0.05)
adjust_method	Character. Multiple testing correction method: "BH", "bonferroni", "holm", "none", or "metabolite_stratified" (recommended, performs BH within each metabolite)
min_score	Numeric. Minimum communication score (default: 0)

Details

The adjust_method parameter controls how p-values are corrected for multiple testing:

- "metabolite_stratified" (recommended): Performs BH correction within each metabolite, then combines results. This is less conservative than global correction and more biologically meaningful since metabolites are independent biological signals.
- "BH", "bonferroni", "holm": Standard global correction methods. Can be very conservative when testing many interactions.
- "none": No correction, uses raw p-values. Use with caution.

Value

Updated scMetaLink object

---

getCommunicationMatrix

Get Communication Summary Matrix

Description

Get Communication Summary Matrix

Usage

```
getCommunicationMatrix(object, aggregate_method = "sum")
```

Arguments

object	scMetaLink object
aggregate_method	Character. How to aggregate metabolites

Value

Matrix of aggregated communication scores

getDatabaseInfo *Get Database Information*

Description

Get Database Information

Usage

```
getDatabaseInfo()
```

Value

data.frame with database statistics

getLactateGenes *Get Lactate Signaling Gene Sets*

Description

Returns curated, literature-validated gene sets involved in lactate signaling pathways. Useful for pathway analysis, visualization, and validation.

Gene selection criteria:

- Synthesis: LDHA family (excludes LDHB which prefers reverse reaction)
- Export: MCT4 (SLC16A3) as primary exporter, plus other MCTs and BSG chaperone
- Direct sensing: HCAR1 only (the sole confirmed lactate GPCR)
- Indirect sensing: Classic proton-sensing GPCRs (GPR4, GPR65, GPR68, GPR132)
- Uptake: MCT1 (SLC16A1) as primary importer, plus BSG chaperone

Usage

```
getLactateGenes(category = "all")
```

Arguments

`category` Character. Gene category to return:

- "all" (default): All gene sets
- "production": Synthesis and export genes
- "degradation": Lactate consumption enzymes
- "direct_sensing": HCAR1 receptor
- "indirect_sensing": Proton-sensing GPCRs
- "uptake": Import transporters

Value

A named list of gene vectors. If category is "all", returns the complete nested list structure. Otherwise, returns the specific category.

Examples

```
# Get all gene sets
genes <- getLactateGenes()
names(genes)

# Get synthesis enzymes
genes$production$synthesis
# [1] "LDHA" "LDHC" "LDHAL6A" "LDHAL6B"

# Get proton-sensing receptors
genes$indirect_sensing$proton_receptors
# [1] "GPR4" "GPR65" "GPR68" "GPR132"

# Get only production genes
prod_genes <- getLactateGenes("production")

# Get direct sensing receptor
getLactateGenes("direct_sensing")$receptor
# [1] "HCAR1"
```

---

getLactateSignalingSummary

Get Lactate Signaling Summary

Description

Returns a summary of lactate-mediated communication between cell types.

Usage

```
getLactateSignalingSummary(object, pathway = "combined", top_n = 10)
```

Arguments

object	scMetaLink object with lactate signaling results
pathway	Character. "direct", "indirect", or "combined". Default "combined".
top_n	Integer. Number of top interactions to return. Default 10.

Value

data.frame summarizing top cell-cell communications

Examples

```
data(crc_example)
obj <- createScMetaLink(crc_expr, crc_meta, "cell_type")
obj <- inferLactateSignaling(obj)
getLactateSignalingSummary(obj)
```

`getMetaboliteReceptors` *Get Receptors for a Metabolite*

Description

Get Receptors for a Metabolite

Usage

```
getMetaboliteReceptors(metabolite, include_transporters = TRUE)
```

Arguments

`metabolite`	Character. Metabolite HMDB ID or name
`include_transporters`	Logical. Include transporters

Value

data.frame with receptor information

`getPathwayCommunicationMatrix` *Get Pathway Communication Matrix*

Description

Get Pathway Communication Matrix

Usage

```
getPathwayCommunicationMatrix(object, pathway)
```

Arguments

`object`	scMetaLink object
`pathway`	Character. Pathway name

Value

Matrix of pathway-specific communication

`getPathwayMetaboliteNetwork` *Get Pathway-Metabolite Network*

Description

Extract pathway-metabolite relationships for network visualization

Usage

```
getPathwayMetaboliteNetwork(object, pathways = NULL, top_n = 10)
```

Arguments

`object`	scMetaLink object
`pathways`	Character vector. Specific pathways to include (NULL for top pathways)
`top_n`	Integer. Number of top pathways if pathways is NULL

Value

data.frame with pathway-metabolite edges

`getPathwayMetabolites` *Get Metabolites in Pathway*

Description

Get Metabolites in Pathway

Usage

```
getPathwayMetabolites(pathway, only_signaling = FALSE)
```

Arguments

`pathway`	Character. Pathway name (can be partial match)
`only_signaling`	Logical. Only return metabolites with receptors

Value

data.frame with metabolites in the pathway

getSpatialDistanceStats

Get Spatial Distance Statistics

Description

Calculate statistics about spatial distances in the dataset

Usage

```
getSpatialDistanceStats(object)
```

Arguments

object	A scMetaLink object with spatial information

Value

A list with distance statistics

getSpatialLactateHotspots

Get Spatial Lactate Hotspots

Description

Identifies spatial hotspots of lactate production and sensing.

Usage

```
getSpatialLactateHotspots(object, type = "all", top_n = 20)
```

Arguments

object	scMetaLink object with spatial lactate signaling results
type	Character. Type of hotspot: "production", "direct_sensing", "indirect_sensing", or "all". Default "all".
top_n	Integer. Number of top spots to return. Default 20.

Value

data.frame with spot IDs, coordinates, scores, and cell types

getSummaryStats *Get Summary Statistics*

Description

Get summary statistics of scMetaLink analysis

Usage

```
getSummaryStats(object)
```

Arguments

object	scMetaLink object

Value

list with summary statistics

getTopLactateProducers *Get Top Lactate Producers*

Description

Returns cell types ranked by lactate production potential.

Usage

```
getTopLactateProducers(object, top_n = 5)
```

Arguments

object	scMetaLink object with lactate signaling results
top_n	Integer. Number of top cell types to return. Default 5.

Value

data.frame with cell types, production scores, and ranks

Examples

```
data(crc_example)
obj <- createScMetaLink(crc_expr, crc_meta, "cell_type")
obj <- inferLactateSignaling(obj)
getTopLactateProducers(obj)
```

getTopLactateSensors *Get Top Lactate Sensors*

Description

Returns cell types ranked by lactate sensing capability.

Usage

```
getTopLactateSensors(object, pathway = "both", top_n = 5)
```

Arguments

object	scMetaLink object with lactate signaling results
pathway	Character. "direct" (HCAR1), "indirect" (proton GPCRs), or "both". Default "both".
top_n	Integer. Number of top cell types to return. Default 5.

Value

data.frame with cell types, sensing scores, and ranks

Examples

```
data(crc_example)
obj <- createScMetaLink(crc_expr, crc_meta, "cell_type")
obj <- inferLactateSignaling(obj)

# Get top sensors for indirect pathway
getTopLactateSensors(obj, pathway = "indirect")
```

getTopProducers *Get Top Producing Cell Types for a Metabolite*

Description

Get Top Producing Cell Types for a Metabolite

Usage

```
getTopProducers(object, metabolite, top_n = 5)
```

Arguments

object	scMetaLink object
metabolite	Character. Metabolite HMDB ID or name
top_n	Integer. Number of top cell types to return

Value

data.frame with top producing cell types

---

getTopSensors *Get Top Sensing Cell Types for a Metabolite*

Description

Get Top Sensing Cell Types for a Metabolite

Usage

```
getTopSensors(object, metabolite, top_n = 5)
```

Arguments

object	scMetaLink object
metabolite	Character. Metabolite HMDB ID or name
top_n	Integer. Number of top cell types to return

Value

data.frame with top sensing cell types

---

identifyCellTypeSpecificMetabolites *Identify Cell Type Specific Metabolites*

Description

Find metabolites specifically produced or sensed by cell types

Usage

```
identifyCellTypeSpecificMetabolites(
  object,
  type = "production",
  specificity_threshold = 1.5
)
```

Arguments

object	scMetaLink object
type	Character. "production" or "sensing"
specificity_threshold	Numeric. Z-score threshold for specificity

Value

data.frame with cell type specific metabolites

---

identifyCommunicationHotspots *Identify Communication Hotspots*

Description

Find spatial regions with high metabolite communication activity. **Note**: This function requires running computeSpatialCommunication() with analysis_level='spot' first.

Usage

```
identifyCommunicationHotspots(
  object,
  metabolite = NULL,
  type = "sender",
  n_hotspots = 5,
  method = "density"
)
```

Arguments

object	A scMetaLink object with spatial communication results
metabolite	Character. Metabolite ID or name to analyze (NULL for aggregate)
type	Character. "sender" or "receiver" to identify production or sensing hotspots
n_hotspots	Integer. Number of hotspot regions to identify
method	Character. Method for hotspot detection: "density" or "clustering"

Value

A data.frame with hotspot information

`inferLactateSignaling` *Infer Lactate-Mediated Cell Communication*

Description

Infers both direct and indirect lactate signaling in single-cell data.

Direct signaling: Lactate binds to HCAR1 (GPR81), the only confirmed lactate GPCR (validated by cryo-EM structure, PLOS Biology 2024).

Indirect signaling: Lactate dissociation (pKa=3.86) produces H+ ions that activate proton-sensing GPCRs (GPR4, GPR65, GPR68, GPR132).

Usage

```
inferLactateSignaling(
  object,
  include_direct = TRUE,
  include_indirect = TRUE,
  method = "combined",
  comm_method = "geometric",
  min_pct = 0.1,
  min_production = 0,
  min_sensing = 0,
  consider_uptake = TRUE,
  consider_degradation = TRUE,
  normalize = TRUE,
  n_permutations = 100,
  seed = 42,
  verbose = TRUE
)
```

Arguments

`object`	A scMetaLink object with expression data
`include_direct`	Logical. Include direct lactate-HCAR1 signaling. Default TRUE.
`include_indirect`	Logical. Include indirect lactate-H+-GPCR signaling. Default TRUE.
`method`	Character. Scoring method: "combined" (recommended), "mean", or "proportion". Default "combined".
`comm_method`	Character. Communication score method: "geometric" (default), "product", or "harmonic".
`min_pct`	Numeric. Minimum percentage of expressing cells (0-1). Default 0.1.
`min_production`	Numeric. Minimum production score threshold (0-1). Default 0.
`min_sensing`	Numeric. Minimum sensing score threshold (0-1). Default 0.

`consider_uptake`
: Logical. Include MCT uptake transporters in direct sensing. Default TRUE.

`consider_degradation`
: Logical. Subtract degradation enzyme expression from production scores. Default TRUE.

`normalize`
: Logical. Normalize scores across cell types. Default TRUE.

`n_permutations`
: Integer. Number of permutations for significance testing. Set to 0 to skip. Default 100.

`seed`
: Integer. Random seed for reproducibility. Default 42.

`verbose`
: Logical. Print progress messages. Default TRUE.

Value

Updated scMetaLink object with lactate_signaling results stored in the parameters slot, containing:

`production`
: Lactate production scores per cell type

`direct_sensing`
: Direct sensing scores (HCAR1)

`indirect_sensing`
: Indirect sensing scores (proton GPCRs)

`direct_communication`
: Communication matrix via direct pathway

`indirect_communication`
: Communication matrix via indirect pathway

`combined_communication`
: Sum of direct and indirect communication

`pvalues`
: Permutation-based p-values (if n_permutations > 0)

`gene_contributions`
: Expression contribution of each gene

`parameters`
: Analysis parameters used

References

- HCAR1 structure: PLOS Biology (2024)
- GPR81 function: Nature Metabolism (2024)
- Lactate signaling: Signal Transduction and Targeted Therapy (2024)
- Proton-sensing GPCRs: Reactome R-HSA-444731

Examples

```
# Load example data
data(crc_example)

# Create scMetaLink object
obj <- createScMetaLink(crc_expr, crc_meta, "cell_type")

# Run lactate signaling analysis
```

```
obj <- inferLactateSignaling(obj)

# Access results
lactate_results <- obj@parameters$lactate_signaling
head(lactate_results$direct_communication)

# Analyze only indirect (proton) signaling
obj <- inferLactateSignaling(obj, include_direct = FALSE)
```

`inferProduction` *Infer Metabolite Production Potential*

Description

Infer metabolite production potential for each cell type based on enzyme expression patterns. Production potential reflects a cell type's capacity to synthesize and secrete metabolites for intercellular communication.

Usage

```
inferProduction(
  object,
  method = "combined",
  mean_method = "arithmetic",
  min_expression = 0,
  min_pct = 0.1,
  consider_degradation = TRUE,
  consider_secretion = TRUE,
  normalize = TRUE,
  verbose = TRUE
)
```

Arguments

`object`	A scMetaLink object
`method`	Character. Scoring method: "mean", "proportion", or "combined"
`mean_method`	Character. Method for calculating mean expression: "arithmetic" (standard mean) or "trimean" (more robust to outliers and dropout). Trimean is recommended for single-cell data with high dropout rates.
`min_expression`	Numeric. Minimum expression threshold
`min_pct`	Numeric. Minimum percentage of expressing cells (0-1)
`consider_degradation`	Logical. Subtract degradation enzyme expression
`consider_secretion`	Logical. Weight by secretion potential
`normalize`	Logical. Normalize scores across cell types
`verbose`	Logical. Print progress messages

Value

Updated scMetaLink object with production_scores slot filled

`inferSensing` *Infer Metabolite Sensing Capability*

Description

Infer metabolite sensing capability for each cell type based on receptor and transporter expression. Sensing capability reflects a cell type's capacity to detect and respond to extracellular metabolites.

Usage

```
inferSensing(
  object,
  method = "combined",
  mean_method = "arithmetic",
  min_expression = 0,
  min_pct = 0.1,
  weight_by_affinity = TRUE,
  include_transporters = TRUE,
  use_hill = FALSE,
  hill_n = 1,
  hill_Kh = 0.5,
  normalize = TRUE,
  verbose = TRUE
)
```

Arguments

`object`	A scMetaLink object
`method`	Character. Scoring method: "mean", "proportion", or "combined"
`mean_method`	Character. Method for calculating mean expression: "arithmetic" (standard mean) or "trimean" (more robust to outliers and dropout).
`min_expression`	Numeric. Minimum expression threshold
`min_pct`	Numeric. Minimum percentage of expressing cells (0-1)
`weight_by_affinity`	Logical. Weight by receptor-metabolite affinity score
`include_transporters`	Logical. Include uptake transporters in sensing
`use_hill`	Logical. Apply Hill function transformation to model receptor binding saturation kinetics. When TRUE, high expression levels show diminishing returns, reflecting biological receptor saturation.
`hill_n`	Numeric. Hill coefficient (cooperativity). Default 1 (no cooperativity). Values > 1 indicate positive cooperativity.

hill_Kh	Numeric. Half-maximal response threshold (0-1 scale after normalization). Default 0.5. Lower values mean saturation occurs at lower expression levels.
normalize	Logical. Normalize scores across cell types
verbose	Logical. Print progress messages

Details

The Hill function transformation models receptor-ligand binding dynamics:

$$P = \frac{E^n}{K_h^n + E^n}$$

where E is expression, n is the Hill coefficient, and Kh is the half-maximal threshold. This reflects the biological reality that receptor signaling saturates at high ligand/receptor concentrations.

Value

Updated scMetaLink object with sensing_scores slot filled

---

inferSpatialLactateSignaling

Infer Spatial Lactate Signaling

Description

Infers lactate signaling in spatial transcriptomics data with distance-weighted communication scores. Supports both direct (HCAR1) and indirect (proton-sensing GPCRs) pathways with spatial context.

Usage

```
inferSpatialLactateSignaling(
  object,
  max_distance = 200,
  distance_decay = "gaussian",
  sigma = 50,
  include_direct = TRUE,
  include_indirect = TRUE,
  method = "combined",
  comm_method = "geometric",
  aggregate_by = "cell_type",
  min_production = 0,
  min_sensing = 0,
  normalize = TRUE,
  verbose = TRUE
)
```

Arguments

object	A spatial scMetaLink object (created with createScMetaLinkFromSpatial)
max_distance	Numeric. Maximum communication distance in micrometers. Spot pairs beyond this distance are considered non-interacting. Default 200 um.
distance_decay	Character. Distance decay function: • "gaussian": Gaussian decay exp(-d^2/(2*sigma^2)) (default) • "exponential": Exponential decay exp(-d/sigma) • "linear": Linear decay max(0, 1 - d/max_distance) • "none": No distance weighting, use cell type level aggregation
sigma	Numeric. Decay parameter for gaussian/exponential (in um). Default 50 um. Literature suggests lactate has medium-range diffusion (~50-80 um).
include_direct	Logical. Include direct lactate-HCAR1 signaling. Default TRUE.
include_indirect	Logical. Include indirect lactate-H+-GPCR signaling. Default TRUE.
method	Character. Scoring method: "combined", "mean", or "proportion". Default "combined".
comm_method	Character. Communication score method: "geometric", "product", "harmonic". Default "geometric".
aggregate_by	Character. Aggregation level: • "cell_type": Aggregate by cell type (default) • "spot": Return spot-level scores (memory intensive) • "both": Return both levels
min_production	Numeric. Minimum production score threshold. Default 0.
min_sensing	Numeric. Minimum sensing score threshold. Default 0.
normalize	Logical. Normalize scores. Default TRUE.
verbose	Logical. Print progress messages. Default TRUE.

Value

Updated scMetaLink object with spatial_lactate_signaling results stored in the parameters slot, containing:

spot_production	Spot-level production scores
spot_direct_sensing	Spot-level direct sensing scores
spot_indirect_sensing	Spot-level indirect sensing scores
celltype_communication	Cell type level communication (if aggregate_by includes "cell_type")
spot_communication	Spot level communication (if aggregate_by includes "spot")
parameters	Analysis parameters

Examples

```
data(st_expr)
data(st_meta)
data(st_scalefactors)

# Create spatial object
obj <- createScMetaLinkFromSpatial(
  expression_data = st_expr,
  spatial_coords = st_meta[, c("x", "y")],
  cell_meta = st_meta,
  cell_type_column = "cell_type",
  scale_factors = st_scalefactors
)

# Run spatial lactate analysis
obj <- inferSpatialLactateSignaling(obj)

# Access results
spatial_results <- obj@parameters$spatial_lactate_signaling
```

---

listGenes *List Available Genes*

Description

Get a list of all genes in the database with their roles

Usage

```
listGenes(role = "all")
```

Arguments

role	Character. Filter by role: "all", "enzyme", "receptor", "transporter"

Value

data.frame with gene information

Examples

```
# Get all genes
genes <- listGenes()

# Get only receptor genes
receptors <- listGenes(role = "receptor")
```

listMetabolites *List Available Metabolites*

Description

Get a list of all metabolites in the database with their properties

Usage

```
listMetabolites(type = "all")
```

Arguments

type	Character. Filter by interaction type: "all", "signaling" (lr only), or "metabolic" (pd only)

Value

data.frame with metabolite information

Examples

```
# Get all metabolites
mets <- listMetabolites()
head(mets)

# Get only signaling metabolites (those with receptors)
signaling_mets <- listMetabolites(type = "signaling")
```

listTopPathways *List Top Pathways*

Description

List Top Pathways

Usage

```
listTopPathways(object, n = 20)
```

Arguments

object	scMetaLink object
n	Integer. Number of top pathways

Value

data.frame with top pathways

loadScMetaLink *Load scMetaLink Object*

Description

Load scMetaLink Object

Usage

```
loadScMetaLink(file)
```

Arguments

file	Character. File path

Value

scMetaLink object

metalinksdb *MetalinksDB Database*

Description

Pre-compiled metabolite-protein interaction database from MetalinksDB

Usage

```
metalinksdb
```

Format

A list containing the following components:

edges data.frame with 41,894 metabolite-protein interactions containing:

- hmdb: HMDB metabolite identifier
- uniprot: UniProt protein identifier
- source: Data source
- combined_score: Interaction confidence score (0-1000)
- mor: Mode of regulation (1=producing, -1=degrading, 0=binding)
- type: Interaction type ("lr"=ligand-receptor, "pd"=produce-degrade)
- transport_direction: For transporters ("in" or "out")

metabolites data.frame with 1,128 metabolites containing:

- hmdb: HMDB identifier

- metabolite: Metabolite name
- pubchem: PubChem identifier
- metabolite_subclass: Chemical classification

proteins data.frame with 4,374 proteins containing:

- uniprot: UniProt identifier
- gene_symbol: Gene symbol
- protein_type: Protein classification (enzyme, gpcr, transporter, etc.)

pathway data.frame with 157,741 metabolite-pathway associations

cell_location data.frame with 2,816 subcellular location annotations

tissue_location data.frame with 2,410 tissue location annotations

disease data.frame with 3,216 disease associations

Details

The database enables two types of metabolite-mediated communication inference:

Ligand-Receptor (lr) type: Direct metabolite-receptor binding interactions, primarily involving GPCRs, nuclear hormone receptors, and ion channels.

Produce-Degrade (pd) type: Enzyme-mediated metabolite production and consumption, enabling inference of metabolite availability from enzyme expression.

Source

MetalinksDB (https://metalinks.org/)

References

Schafer, S., et al. (2023). MetalinksDB: a knowledgebase of metabolite-centric signaling. Nature Communications.

Examples

```
# Access the database
db <- scMetaLink:::.load_metalinksdb()

# View available metabolites
head(db$metabolites)

# Check interaction types
table(db$edges$type)
```

`plotCommunicationCircle` *Plot Communication Circle*

Description

Create a chord diagram of cell-cell communication

Usage

```
plotCommunicationCircle(
  object,
  top_n = 50,
  metabolite = NULL,
  colors = NULL,
  transparency = 0.5,
  title = NULL
)
```

Arguments

`object`	scMetaLink object
`top_n`	Integer. Number of top interactions to show
`metabolite`	Character. Specific metabolite (NULL for aggregated)
`colors`	Named vector. Colors for cell types
`transparency`	Numeric. Link transparency (0-1)
`title`	Character. Plot title

Value

Invisibly returns NULL, plots chord diagram

`plotCommunicationHeatmap` *Plot Communication Heatmap*

Description

Create a heatmap of cell-cell communication

Usage

```
plotCommunicationHeatmap(
  object,
  metabolite = NULL,
  cluster_rows = TRUE,
  cluster_cols = TRUE,
  show_values = FALSE,
  colors = NULL,
  title = NULL
)
```

Arguments

object	scMetaLink object
metabolite	Character. Specific metabolite to show (NULL for aggregated)
cluster_rows	Logical. Cluster rows
cluster_cols	Logical. Cluster columns
show_values	Logical. Show values in cells
colors	Character vector. Color palette
title	Character. Plot title

Value

A ComplexHeatmap object

Examples

```
# Plot aggregated communication heatmap
plotCommunicationHeatmap(obj)

# Plot specific metabolite
plotCommunicationHeatmap(obj, metabolite = "HMDB0000148")
```

---

plotCommunicationNetwork

Plot Communication Network

Description

Create a network visualization of cell-cell communication

Usage

```
plotCommunicationNetwork(
  object,
  metabolite = NULL,
  min_score = 0,
  layout = "fr",
  node_size_by = "degree",
  edge_width_scale = 2,
  colors = NULL
)
```

Arguments

object	scMetaLink object
metabolite	Character. Specific metabolite (NULL for aggregated)
min_score	Numeric. Minimum score threshold
layout	Character. Network layout algorithm
node_size_by	Character. Size nodes by "degree" or "centrality"
edge_width_scale	Numeric. Scale factor for edge widths
colors	Named vector. Colors for cell types

Value

A ggplot object

---

plotDifferentialCommunication

Plot Differential Communication

Description

Visualize differential communication between conditions

Usage

```
plotDifferentialCommunication(diff_results, top_n = 30, type = "bar")
```

Arguments

diff_results	data.frame from compareCommunication()
top_n	Integer. Number of top changes to show
type	Character. "bar" or "volcano"

Value

A ggplot object

`plotEnrichedPathways` *Plot Enriched Pathways*

Description

Visualize enriched pathways from enrichPathways()

Usage

```
plotEnrichedPathways(enrichment_results, top_n = 20, show_overlap = TRUE)
```

Arguments

`enrichment_results`	data.frame from enrichPathways()
`top_n`	Integer. Number of top pathways to show
`show_overlap`	Logical. Show overlap count

Value

A ggplot object

`plotLactatePathwayComparison` *Plot Lactate Pathway Comparison*

Description

Creates a side-by-side comparison of direct and indirect lactate signaling pathways.

Usage

```
plotLactatePathwayComparison(object, show_production = TRUE)
```

Arguments

`object`	scMetaLink object with lactate signaling results
`show_production`	Logical. Also show production scores. Default TRUE.

Value

A ggplot2 object or base R plot

plotLactateSignaling *Plot Lactate Signaling Heatmap*

Description

Creates a heatmap visualization of lactate-mediated cell communication. Shows sender cell types on rows and receiver cell types on columns.

Usage

```
plotLactateSignaling(
  object,
  pathway = "combined",
  cluster_rows = TRUE,
  cluster_cols = TRUE,
  show_values = FALSE,
  colors = NULL,
  title = NULL
)
```

Arguments

object	scMetaLink object with lactate signaling results
pathway	Character. Which pathway to plot: "direct", "indirect", or "combined". Default "combined".
cluster_rows	Logical. Cluster rows. Default TRUE.
cluster_cols	Logical. Cluster columns. Default TRUE.
show_values	Logical. Show score values in cells. Default FALSE.
colors	Character vector. Color palette. Default viridis.
title	Character. Plot title. Default auto-generated.

Value

A ggplot2 object or ComplexHeatmap object (if available)

Examples

```
data(crc_example)
obj <- createScMetaLink(crc_expr, crc_meta, "cell_type")
obj <- inferLactateSignaling(obj)
plotLactateSignaling(obj, pathway = "indirect")
```

plotMetaboliteProfile *Plot Metabolite Profile*

Description

Visualize production and sensing profiles for a metabolite

Usage

```
plotMetaboliteProfile(object, metabolite, show_genes = FALSE)
```

Arguments

object	scMetaLink object
metabolite	Character. Metabolite HMDB ID or name
show_genes	Logical. Show contributing genes

Value

A ggplot object

plotPathwayCommunication *Plot Pathway Communication*

Description

Visualize pathway-level communication

Usage

```
plotPathwayCommunication(object, top_pathways = 20, type = "bar")
```

Arguments

object	scMetaLink object
top_pathways	Integer. Number of top pathways to show
type	Character. "heatmap" or "bar"

Value

A ggplot or ComplexHeatmap object

plotSpatialCellTypes *Plot Spatial Cell Types*

Description

Visualize cell type distribution on spatial coordinates

Usage

```
plotSpatialCellTypes(
  object,
  point_size = 1.5,
  alpha = 0.8,
  colors = NULL,
  show_legend = TRUE
)
```

Arguments

object	A scMetaLink object with spatial information
point_size	Numeric. Size of spot points
alpha	Numeric. Transparency (0-1)
colors	Character vector. Colors for cell types (NULL for default)
show_legend	Logical. Show legend

Value

A ggplot2 object

plotSpatialCommunicationNetwork *Plot Spatial Communication Network*

Description

Visualize cell-cell communication with spatial context

Usage

```
plotSpatialCommunicationNetwork(
  object,
  metabolite = NULL,
  sender_type = NULL,
  receiver_type = NULL,
  top_n = 20,
  arrow_scale = 1,
  point_size = 5,
  show_labels = TRUE
)
```

Arguments

object	A scMetaLink object with spatial communication results
metabolite	Character. Metabolite to visualize (NULL for aggregate)
sender_type	Character. Sender cell type (NULL for all)
receiver_type	Character. Receiver cell type (NULL for all)
top_n	Integer. Number of top interactions to show arrows for
arrow_scale	Numeric. Scale factor for arrow thickness
point_size	Numeric. Size of cell type center points
show_labels	Logical. Show cell type labels

Value

A ggplot2 object

---

plotSpatialComparison *Plot Spatial Communication Comparison*

Description

Compare communication patterns for multiple metabolites

Usage

```
plotSpatialComparison(
  object,
  metabolites,
  type = "production",
  ncol = 2,
  point_size = 1
)
```

Arguments

object	A scMetaLink object with spatial communication results
metabolites	Character vector. Metabolites to compare
type	Character. "production" or "sensing"
ncol	Integer. Number of columns in facet grid
point_size	Numeric. Size of spot points

Value

A ggplot2 object

---

plotSpatialDistanceDistribution
Plot Spatial Distance Distribution

Description

Visualize the distribution of spatial distances between spots

Usage

```
plotSpatialDistanceDistribution(
  object,
  by_celltype = FALSE,
  max_distance = NULL,
  bins = 50
)
```

Arguments

object	A scMetaLink object with spatial information
by_celltype	Logical. Show distances stratified by cell type pairs
max_distance	Numeric. Maximum distance to show (NULL for all)
bins	Integer. Number of histogram bins

Value

A ggplot2 object

`plotSpatialFeature` *Plot Spatial Communication Heatmap*

Description

Visualize metabolite communication on spatial coordinates

Usage

```
plotSpatialFeature(
  object,
  metabolite,
  type = "production",
  cell_type = NULL,
  point_size = 1.5,
  alpha = 0.8,
  low_color = "#FFFFCC",
  high_color = "#E31A1C",
  title = NULL
)
```

Arguments

`object`	A scMetaLink object with spatial information
`metabolite`	Character. Metabolite ID or name to visualize
`type`	Character. What to plot: "production", "sensing", or "communication"
`cell_type`	Character. For communication, specify sender or receiver cell type
`point_size`	Numeric. Size of spot points
`alpha`	Numeric. Transparency (0-1)
`low_color`	Character. Color for low values
`high_color`	Character. Color for high values
`title`	Character. Plot title (NULL for auto)

Value

A ggplot2 object

Examples

```
data(st_expr)
data(st_meta)
data(st_scalefactors)
obj <- createScMetaLinkFromSpatial(st_expr, st_meta[,c("x","y")],
                                   st_meta, "cell_type", st_scalefactors)
obj <- inferProduction(obj)
```

```
# Plot lactate production
plotSpatialFeature(obj, metabolite = "HMDB0000190", type = "production")
```

`plotSpatialHotspots` *Plot Spatial Hotspots*

Description

Visualize communication hotspots on spatial coordinates

Usage

```
plotSpatialHotspots(
  object,
  metabolite = NULL,
  type = "sender",
  point_size = 1,
  hotspot_size = 5,
  n_hotspots = 5
)
```

Arguments

`object`	A scMetaLink object with spatial communication results
`metabolite`	Character. Metabolite to analyze (NULL for aggregate)
`type`	Character. "sender" or "receiver"
`point_size`	Numeric. Size of spot points
`hotspot_size`	Numeric. Size of hotspot markers
`n_hotspots`	Integer. Number of hotspots to highlight

Value

A ggplot2 object

plotSpatialLactate *Plot Spatial Lactate Signaling*

Description

Visualizes spatial distribution of lactate production and sensing scores.

Usage

```
plotSpatialLactate(object, type = "production", point_size = 2, title = NULL)
```

Arguments

object	scMetaLink object with spatial lactate signaling results
type	Character. What to plot: "production", "direct_sensing", "indirect_sensing", or "all". Default "production".
point_size	Numeric. Size of spots. Default 2.
title	Character. Plot title. Default auto-generated.

Value

A ggplot2 object or base R plot

plotTopInteractions *Plot Top Interactions*

Description

Dot plot of top metabolite-mediated interactions

Usage

```
plotTopInteractions(object, top_n = 30, group_by = "metabolite")
```

Arguments

object	scMetaLink object
top_n	Integer. Number of top interactions
group_by	Character. Group by "sender", "receiver", or "metabolite"

Value

A ggplot object

`runScMetaLink` *Run Complete scMetaLink Analysis*

Description

One-step function to run the complete metabolite-mediated cell communication analysis pipeline

Usage

```
runScMetaLink(
  expression_data,
  cell_meta,
  cell_type_column = "cell_type",
  method = "combined",
  min_cells = 10,
  min_pct = 0.1,
  n_permutations = 1000,
  pvalue_threshold = 0.05,
  n_cores = 1,
  verbose = TRUE
)
```

Arguments

`expression_data`	A matrix or sparse matrix of normalized expression
`cell_meta`	A data.frame with cell metadata
`cell_type_column`	Character. Column name for cell type annotations
`method`	Character. Expression scoring method: "mean", "proportion", "combined"
`min_cells`	Integer. Minimum cells per cell type
`min_pct`	Numeric. Minimum percentage of expressing cells (0-1)
`n_permutations`	Integer. Number of permutations for significance testing
`pvalue_threshold`	Numeric. P-value threshold for significance
`n_cores`	Integer. Number of cores for parallel computing
`verbose`	Logical. Print progress messages

Details

This function runs the complete scMetaLink pipeline:

1. Create scMetaLink object from expression data
2. Infer metabolite production potential (MPP)

3. Infer metabolite sensing capability (MSC)
4. Compute cell-cell communication scores
5. Perform permutation-based significance testing
6. Filter significant interactions
7. Aggregate by pathway

Value

A scMetaLink object with all analysis completed

Examples

```
# Run complete analysis
result <- runScMetaLink(
  expression_data = expr_matrix,
  cell_meta = cell_metadata,
  cell_type_column = "cell_type",
  n_permutations = 1000
)

# View significant interactions
sig <- getSignificantInteractions(result)
head(sig)
```

---

runScMetaLinkSeurat *Run scMetaLink from Seurat Object*

Description

Convenience function to run analysis from Seurat object

Usage

```
runScMetaLinkSeurat(
  seurat_obj,
  cell_type_column = "cell_type",
  assay = "RNA",
  slot = "data",
  ...
)
```

Arguments

seurat_obj	A Seurat object
cell_type_column	Character. Column in meta.data for cell type
assay	Character. Assay to use
slot	Character. Slot to use ("data" or "counts")
...	Additional arguments passed to runScMetaLink

Value

A scMetaLink object

---

saveScMetaLink *Save scMetaLink Object*

Description

Save scMetaLink Object

Usage

```
saveScMetaLink(object, file)
```

Arguments

object	scMetaLink object
file	Character. File path (RDS format)

Value

Invisibly returns file path

---

scMetaLink-class *scMetaLink Class Definition*

Description

S4 class for storing metabolite-mediated cell communication analysis results

Usage

```
## S4 method for signature 'scMetaLink'
show(object)
```

Arguments

object	scMetaLink object

Slots

- expression_data Matrix. Normalized gene expression matrix (genes x cells)
- cell_meta data.frame. Cell metadata with cell type annotations
- cell_type_column Character. Column name for cell type in cell_meta
- production_scores Matrix. Metabolite production potential scores (metabolites x cell_types)
- sensing_scores Matrix. Metabolite sensing capability scores (metabolites x cell_types)
- communication_scores Array. Communication scores (sender x receiver x metabolite)
- communication_pvalues Array. P-values from permutation test
- significant_interactions data.frame. Filtered significant interactions
- pathway_aggregated data.frame. Pathway-level aggregated results
- parameters list. Analysis parameters
- database list. MetalinksDB reference data

---

searchGene *Search Gene in Database*

Description

Search Gene in Database

Usage

```
searchGene(query)
```

Arguments

query	Character. Gene symbol

Value

data.frame with gene information and associated metabolites

searchMetabolite *Search Metabolite in Database*

Description

Search Metabolite in Database

Usage

```
searchMetabolite(query, exact = FALSE)
```

Arguments

query	Character. Search query (name or HMDB ID)
exact	Logical. Exact match only

Value

data.frame with matching metabolites

st_expr *Colon Spatial Transcriptomics Expression Data*

Description

Example 10x Visium spatial transcriptomics expression matrix from colon tissue

Usage

```
st_expr
```

Format

A sparse dgCMatrix with 4,284 genes (rows) x 1,000 spots (columns). Only genes present in MetalinksDB are included to reduce file size.

Details

This is a subset of colon Visium data containing 1,000 randomly sampled spots. The data is suitable for testing spatial metabolite communication analysis.

Technical specifications:

- Platform: 10x Genomics Visium
- Spot diameter: 55 um
- Resolution: ~2.37 pixels/um
- Genes: 4,284 (filtered to MetalinksDB)
- Spots: 1,000 (subsampled from 3,313)

Source

Colon spatial transcriptomics dataset

See Also

st_meta, st_scalefactors

Examples

```
data(st_expr)
data(st_meta)
data(st_scalefactors)

# Check dimensions
dim(st_expr)

# View spatial coordinates
head(st_meta)

# Run spatial scMetaLink analysis
obj <- createScMetaLinkFromSpatial(
  expression_data = st_expr,
  spatial_coords = st_meta[, c("x", "y")],
  cell_meta = st_meta,
  cell_type_column = "cell_type"
)
```

---

st_meta *Colon Spatial Transcriptomics Metadata*

Description

Spot metadata for the colon Visium example dataset

Usage

```
st_meta
```

Format

A data.frame with 1,000 rows (spots) and 5 columns:

x X coordinate in image pixels (imagecol)

y Y coordinate in image pixels (imagerow)

array_row Row position in the Visium array

array_col Column position in the Visium array

cell_type Cell type annotation (mock clusters for testing)

Details

The cell_type column contains mock cell type annotations generated by spatial k-means clustering. In real analysis, users should provide cell type annotations from deconvolution methods (e.g., RCTD, cell2location, SPOTlight) or manual annotation.

Cell type distribution in this example:

- Epithelial: ~189 spots
- Fibroblast: ~232 spots
- Endothelial: ~174 spots
- Immune: ~145 spots
- Stromal: ~133 spots
- Tumor: ~127 spots

See Also

st_expr, st_scalefactors

Examples

```
data(st_meta)

# View spot distribution
table(st_meta$cell_type)

# Plot spatial coordinates
plot(st_meta$x, st_meta$y, col = as.factor(st_meta$cell_type), pch = 16)
```

---

st_scalefactors *Colon Spatial Transcriptomics Scale Factors*

Description

Scale factors for the colon Visium example dataset

Usage

```
st_scalefactors
```

Format

A list containing:

spot_diameter_fullres Spot diameter in full resolution pixels (~130)

tissue_hires_scalef Scale factor for high-res image (~0.12)

tissue_lowres_scalef Scale factor for low-res image (~0.037)

spot_diameter_um Spot diameter in micrometers (55 um)

pixels_per_um Pixels per micrometer (~2.37)

Details

These scale factors are essential for converting between pixel coordinates and physical distances (micrometers). The standard Visium spot diameter is 55 um, which is used to calculate the pixels_per_um conversion factor.

For spatial communication analysis, distances should be converted to micrometers to enable biologically meaningful distance thresholds.

See Also

st_expr, st_meta

Examples

```
data(st_scalefactors)

# Convert pixel distance to micrometers
pixel_distance <- 500
um_distance <- pixel_distance / st_scalefactors$pixels_per_um
cat("Distance:", um_distance, "um\n")
```

---

summarizeCommunicationPairs

Summarize Communication by Cell Type Pairs

Description

Summarize Communication by Cell Type Pairs

Usage

```
summarizeCommunicationPairs(object, aggregate_method = "sum")
```

Arguments

object	scMetaLink object
aggregate_method	Character. "sum", "mean", or "count"

Value

data.frame with summarized communication

summarizePathwayActivity
Summarize Pathway Activity

Description

Summarize Pathway Activity

Usage

```
summarizePathwayActivity(object)
```

Arguments

object	scMetaLink object

Value

data.frame with pathway activity summary

Index

∗ **datasets**
  crc_expr, 13
  crc_meta, 14
  metalinksdb, 39
  st_expr, 57
  st_meta, 58
  st_scalefactors, 59

accessors, 5
aggregateByPathway, *4*, 6

checkLactateGenes, 7
citationScMetaLink, 7
compareCommunication, 8
computeCommunication, *4*, 9
computeSpatialCommunication, *4*, 10
crc_expr, 13
crc_meta, 14
createScMetaLink, *4*, 14
createScMetaLinkFromSCE, 15
createScMetaLinkFromSeurat, 16
createScMetaLinkFromSeuratSpatial, 17
createScMetaLinkFromSpatial, *4*, 17

enrichPathways, 19
exportResults, 20

filterSignificantInteractions, *4*, 20

getCommunicationMatrix, 21
getCommunicationScores *(*accessors*)*, 5
getCommunicationScores,scMetaLink-method *(*accessors*)*, 5
getDatabaseInfo, 22
getLactateGenes, 22
getLactateSignalingSummary, 23
getMetaboliteReceptors, 24
getParameters *(*accessors*)*, 5
getParameters,scMetaLink-method *(*accessors*)*, 5
getPathwayAggregated *(*accessors*)*, 5
getPathwayAggregated,scMetaLink-method *(*accessors*)*, 5
getPathwayCommunicationMatrix, 24
getPathwayMetaboliteNetwork, 25
getPathwayMetabolites, 25
getProductionScores *(*accessors*)*, 5
getProductionScores,scMetaLink-method *(*accessors*)*, 5
getSensingScores *(*accessors*)*, 5
getSensingScores,scMetaLink-method *(*accessors*)*, 5
getSignificantInteractions *(*accessors*)*, 5
getSignificantInteractions,scMetaLink-method *(*accessors*)*, 5
getSpatialDistanceStats, 26
getSpatialLactateHotspots, 26
getSummaryStats, 27
getTopLactateProducers, 27
getTopLactateSensors, 28
getTopProducers, 28
getTopSensors, 29

identifyCellTypeSpecificMetabolites, 29
identifyCommunicationHotspots, 30
inferLactateSignaling, 31
inferProduction, *4*, 33
inferSensing, *4*, 34
inferSpatialLactateSignaling, 35

listGenes, 37
listMetabolites, 38
listTopPathways, 38
loadScMetaLink, 39

metalinksdb, 39

plotCommunicationCircle, 41
plotCommunicationHeatmap, 41

plotCommunicationNetwork, 42
plotDifferentialCommunication, 43
plotEnrichedPathways, 44
plotLactatePathwayComparison, 44
plotLactateSignaling, 45
plotMetaboliteProfile, 46
plotPathwayCommunication, 46
plotSpatialCellTypes, 47
plotSpatialCommunicationNetwork, 47
plotSpatialComparison, 48
plotSpatialDistanceDistribution, 49
plotSpatialFeature, 50
plotSpatialHotspots, 51
plotSpatialLactate, 52
plotTopInteractions, 52

runScMetaLink, *4*, 53
runScMetaLinkSeurat, 54

saveScMetaLink, 55
scMetaLink *(*scMetaLink-package*)*, 4
scMetaLink-class, 55
scMetaLink-package, 4
searchGene, 56
searchMetabolite, 57
show,scMetaLink-method
        *(*scMetaLink-class*)*, 55
st_expr, 57, *59, 60*
st_meta, 58, *58*, *60*
st_scalefactors, *58*, 59, *59*
summarizeCommunicationPairs, 60
summarizePathwayActivity, 61