Package: scPairs (via r-universe)

October 20, 2026

Version 0.1.9

Title Identifying Synergistic Gene Pairs in Single-Cell and Spatial Transcriptomics

Description Discovers synergistic gene pairs in single-cell RNA-seq and spatial transcriptomics data. Unlike conventional pairwise co-expression analyses that rely on a single correlation metric, scPairs integrates 14 complementary metrics across five orthogonal evidence layers to compute a composite synergy score with optional permutation-based significance testing. The five evidence layers span cell-level co-expression (Pearson, Spearman, biweight midcorrelation, mutual information, ratio consistency), neighbourhood-aware smoothing (KNN-smoothed correlation, neighbourhood co-expression, cluster pseudo-bulk, cross-cell-type, neighbourhood synergy), prior biological knowledge (GO/KEGG co-annotation Jaccard, pathway bridge score), trans-cellular interaction, and spatial co-variation (Lee's L, co-location quotient). This multi-scale design enables researchers to move beyond simple co-expression towards a comprehensive characterisation of cooperative gene regulation at transcriptomic and spatial resolution. For more information, see the package documentation at <https://github.com/zhaoqing-wang/scPairs>.

License MIT + file LICENSE

URL https://github.com/zhaoqing-wang/scPairs

BugReports https://github.com/zhaoqing-wang/scPairs/issues

Depends R (>= 4.1.0)

Imports data.table, ggplot2, ggraph, ggrepel, igraph, methods, Matrix, patchwork, Seurat (>= 4.0), SeuratObject, stats, tidygraph, tidyr

Suggests AnnotationDbi, org.Mm.eg.db, org.Hs.eg.db, crayon, ggExtra, RANN, testthat (>= 3.0.0)

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Date 2026-04-23

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

Repository https://zhaoqing-wang.r-universe.dev

Date/Publication 2026-04-23 12:29:49 UTC

RemoteUrl https://github.com/zhaoqing-wang/scpairs

RemoteRef HEAD

RemoteSha 34ad4a7b3fd3108a7375ac65a27bd335ae7a774f

Contents

AssessGenePair 2
FindAllPairs 5
FindGenePairs 7
PlotBridgeNetwork 9
PlotPairCrossType 11
PlotPairDimplot 13
PlotPairHeatmap 14
PlotPairNetwork 15
PlotPairScatter 16
PlotPairSmoothed 17
PlotPairSpatial 19
PlotPairSummary 20
PlotPairSynergy 21
PlotPairViolin 23
print.scPairs_gene_result 24
print.scPairs_pair_result 25
print.scPairs_result 25
scpairs_testdata 26

Index **28**

`AssessGenePair` *Assess the Synergy of a Specific Gene Pair*

Description

Given two genes, `AssessGenePair` performs an in-depth evaluation of their co-regulatory relationship. In addition to the standard multi-metric scoring, it computes:

- **Per-cluster co-expression** – correlation within each cell cluster.
- **Expression distribution overlap** – Jaccard index of expressing cells.
- **Permutation-based significance** – 999 permutations by default.

Usage

```
AssessGenePair(
  object,
  gene1,
  gene2,
  assay = NULL,
  slot = "data",
  cluster_col = NULL,
  mode = c("all", "expression", "prior_only"),
  use_prior = TRUE,
  organism = "mouse",
  custom_pairs = NULL,
  use_neighbourhood = TRUE,
  neighbourhood_k = 20,
  neighbourhood_reduction = "pca",
  smooth_alpha = 0.3,
  use_spatial = TRUE,
  spatial_k = 6,
  n_perm = 999,
  verbose = TRUE
)
```

Arguments

object	A Seurat object.
gene1	Character; first gene.
gene2	Character; second gene.
assay	Character; assay name.
slot	Character; data slot.
cluster_col	Character; cluster column.
mode	Character; "all", "expression", or "prior_only".
use_prior	Logical; prior knowledge scores.
organism	Character; "mouse" or "human".
custom_pairs	Optional data.frame of custom interactions.
use_neighbourhood	Logical; neighbourhood metrics.
neighbourhood_k	Integer; KNN k.
neighbourhood_reduction	Character; reduction for KNN.
smooth_alpha	Numeric; self-weight for smoothing.
use_spatial	Logical.
spatial_k	Integer; spatial KNN k.
n_perm	Integer; permutations (default 999).
verbose	Logical.

Details

The mode parameter controls which layers are scored:

- "all" (default) – full multi-evidence assessment.
- "expression" – expression and neighbourhood metrics only.
- "prior_only" – prior knowledge scores only.

Value

A list with class "scPairs_pair_result":

gene1, gene2 The query genes.

pairs Single-row data.table with all metric columns and synergy_score, rank, confidence (same format as FindAllPairs output for unified downstream processing).

metrics Named list of all computed metrics.

per_cluster data.frame of per-cluster correlations.

synergy_score Composite score.

p_value Permutation p-value.

confidence Categorical confidence label.

jaccard_index Expression overlap Jaccard index.

has_spatial Logical.

n_cells Integer.

mode Character.

See Also

FindAllPairs for genome-wide screening, FindGenePairs for query-centric partner search, PlotPairSynergy and PlotPairSummary for visualising pair-level evidence.

Other Section_1_Discovery: FindAllPairs(), FindGenePairs()

Examples

```
# Assess the injected co-expressed pair GENE3 & GENE4.
result <- AssessGenePair(scpairs_testdata,
                         gene1   = "GENE3",
                         gene2   = "GENE4",
                         mode    = "expression",
                         verbose = FALSE)
print(result)
```

FindAllPairs *Discover All Synergistic Gene Pairs*

Description

The primary discovery function of **scPairs**. Given a Seurat object, FindAllPairs identifies synergistic gene pairs by integrating multiple lines of evidence: co-expression, neighbourhood smoothing, prior biological knowledge, and spatial co-variation.

Usage

```
FindAllPairs(
  object,
  features = NULL,
  n_top_genes = 2000,
  assay = NULL,
  slot = "data",
  cluster_col = NULL,
  mode = c("all", "expression", "prior_only"),
  cor_method = c("pearson", "spearman", "biweight"),
  n_mi_bins = 5,
  min_cells_expressed = 10,
  use_prior = TRUE,
  organism = "mouse",
  custom_pairs = NULL,
  use_neighbourhood = TRUE,
  neighbourhood_k = 20,
  neighbourhood_reduction = "pca",
  smooth_alpha = 0.3,
  use_spatial = TRUE,
  spatial_k = 6,
  n_perm = 0,
  weights = NULL,
  top_n = NULL,
  verbose = TRUE
)
```

Arguments

object	A Seurat object (scRNA-seq or spatial).
features	Character vector of gene names to consider. NULL (default) uses Seurat VariableFeatures; if unavailable, selects the top n_top_genes by mean expression.
n_top_genes	Integer; maximum number of genes to analyse when features = NULL. Default 2000.
assay	Character; assay to use. Default: DefaultAssay(object).
slot	Character; data slot. Default "data" (log-normalised).

cluster_col	Character; column in meta.data with cluster IDs. NULL = use Idents(object).
mode	Character; "all", "expression", or "prior_only".
cor_method	Character vector; correlation methods to compute. Default c("pearson", "spearman", "biweight").
n_mi_bins	Integer; bins for mutual information. 0 = skip MI.
min_cells_expressed	Integer; minimum co-expressing cells to keep a pair. Default 10.
use_prior	Logical; integrate prior knowledge (GO/KEGG). Default TRUE.
organism	Character; "mouse" or "human".
custom_pairs	Optional data.frame with columns gene1, gene2.
use_neighbourhood	Logical; compute neighbourhood-aware metrics. Default TRUE.
neighbourhood_k	Integer; KNN k. Default 20.
neighbourhood_reduction	Character; reduction for KNN. Default "pca".
smooth_alpha	Numeric in [0,1]; self-weight for KNN smoothing.
use_spatial	Logical; compute spatial metrics when available.
spatial_k	Integer; spatial KNN k.
n_perm	Integer; permutations for p-values. 0 = skip.
weights	Named numeric; metric weights for score integration.
top_n	Integer or NULL; return only top *n* pairs.
verbose	Logical.

Details

Metrics are rank-normalised and combined via weighted summation. Optional permutation testing provides empirical p-values.

The mode parameter controls which metric layers are computed:

- "all" (default) – compute all available metrics.
- "expression" – expression and neighbourhood metrics only (no prior knowledge).
- "prior_only" – prior knowledge scores only (fast).

Value

A list with class "scPairs_result" containing:

pairs data.table of gene pairs with all metric columns, synergy_score, rank, p_value (if permutation), p_adj, confidence.

parameters List of analysis parameters.

n_genes Number of genes analysed.

n_cells Number of cells.

has_spatial Logical.

mode Character; the mode used.

See Also

FindGenePairs for query-centric partner search, AssessGenePair for in-depth single-pair assessment, PlotPairNetwork and PlotPairHeatmap for visualising discovery results.

Other Section_1_Discovery: AssessGenePair(), FindGenePairs()

Examples

```
# scpairs_testdata is a built-in Seurat object with 100 cells x 20 genes.
# GENE3 & GENE4 are injected as the top co-expressed pair.
result <- FindAllPairs(scpairs_testdata,
                       n_top_genes = 20,
                       top_n       = 10,
                       mode        = "expression",
                       verbose     = FALSE)
print(result)
```

FindGenePairs *Find Synergistic Partners for a Given Gene*

Description

Given a gene of interest, FindGenePairs identifies and ranks all genes that act synergistically with it. Uses the same multi-evidence framework as FindAllPairs() but focuses computation on pairs involving the query gene, making it much faster for targeted queries.

Usage

```
FindGenePairs(
  object,
  gene,
  candidates = NULL,
  n_top_genes = 2000,
  assay = NULL,
  slot = "data",
  cluster_col = NULL,
  mode = c("all", "expression", "prior_only"),
  cor_method = c("pearson", "spearman", "biweight"),
  n_mi_bins = 5,
  min_cells_expressed = 10,
  use_prior = TRUE,
  organism = "mouse",
  custom_pairs = NULL,
  use_neighbourhood = TRUE,
  neighbourhood_k = 20,
  neighbourhood_reduction = "pca",
  smooth_alpha = 0.3,
  use_spatial = TRUE,
```

```
  spatial_k = 6,
  n_perm = 0,
  weights = NULL,
  top_n = NULL,
  verbose = TRUE
)
```

Arguments

object	A Seurat object.
gene	Character; the query gene name.
candidates	Character vector of candidate partner genes. NULL = auto-select.
n_top_genes	Integer; max candidates when candidates = NULL.
assay	Character; assay name.
slot	Character; data slot.
cluster_col	Character; cluster column in meta.data.
mode	Character; "all", "expression", or "prior_only".
cor_method	Correlation methods.
n_mi_bins	Bins for mutual information.
min_cells_expressed	Minimum cells co-expressing both genes.
use_prior	Logical; compute prior knowledge metrics.
organism	Character; "mouse" or "human".
custom_pairs	Optional data.frame of custom interactions.
use_neighbourhood	Logical; compute neighbourhood metrics.
neighbourhood_k	Integer; KNN k.
neighbourhood_reduction	Character; reduction for KNN.
smooth_alpha	Numeric; self-weight for smoothing.
use_spatial	Logical; compute spatial metrics.
spatial_k	Integer; spatial neighbourhood k.
n_perm	Integer; permutations for p-values.
weights	Named numeric; metric weights.
top_n	Integer; return only top partners.
verbose	Logical.

Details

The mode parameter controls which metric layers are computed:

- "all" (default) – all available metrics.
- "expression" – expression and neighbourhood only.
- "prior_only" – prior knowledge scores only.

Value

A list with class "scPairs_gene_result":

query_gene The input gene.

pairs data.table of partners ranked by synergy score.

parameters Analysis parameters.

n_candidates Number of candidates tested.

n_cells Number of cells.

has_spatial Logical.

mode Character.

See Also

FindAllPairs for genome-wide screening, AssessGenePair for in-depth single-pair assessment, PlotPairNetwork for visualising partner networks.

Other Section_1_Discovery: AssessGenePair(), FindAllPairs()

Examples

```
# Find synergistic partners of GENE3.  GENE4 is expected to rank first.
result <- FindGenePairs(scpairs_testdata,
                        gene    = "GENE3",
                        top_n   = 10,
                        mode    = "expression",
                        verbose = FALSE)
print(result)
```

PlotBridgeNetwork *Plot Bridge Gene Network*

Description

Draws a publication-ready radial bridge gene network showing the prior-knowledge connections between a focal gene pair via shared GO/KEGG pathway intermediaries (bridge genes). Focal genes are placed at the centre; bridge genes are arranged on a ring whose radius is inversely proportional to their shared pathway count with the focal pair (more shared terms => closer to centre, reflecting stronger biological relevance). Solid edges connect focal genes to bridge genes: red edges originate from gene1, blue edges from gene2. Edge width encodes shared term count. Pairwise Jaccard similarity between bridge genes is overlaid as thin dotted lines, whose opacity reflects similarity strength, revealing functional clusters among the intermediaries.

Usage

```
PlotBridgeNetwork(
  object,
  gene1,
  gene2,
  organism = "mouse",
  prior_net = NULL,
  top_bridges = 15,
  layout = "auto",
  assay = NULL,
  slot = "data",
  label_size = 3,
  pt_size_range = c(3, 9),
  edge_width_range = c(0.4, 2),
  sim_threshold = 0.05,
  title = NULL
)
```

Arguments

object	A Seurat object.
gene1	Character; first focal gene.
gene2	Character; second focal gene.
organism	Character; "mouse" or "human". Used when prior_net is NULL.
prior_net	Optional prior network object from .build_prior_network(). If NULL, built automatically using organism.
top_bridges	Integer; maximum number of bridge genes to display. Default 15.
layout	Ignored; layout is always radial (kept for API compatibility).
assay	Character; assay name. NULL uses the default assay.
slot	Character; data layer/slot. Default "data".
label_size	Numeric; gene label font size. Default 3.
pt_size_range	Numeric vector of length 2; minimum and maximum node sizes for bridge genes. Default c(3, 9).
edge_width_range	Numeric vector of length 2; minimum and maximum edge widths for focal-to-bridge connections, scaled by shared term count. Default c(0.4, 2).
sim_threshold	Numeric (0 to 1); minimum Jaccard similarity between two bridge genes required to draw a dotted similarity edge. Default 0.05.
title	Character; plot title. NULL generates a default title.

Value

A ggplot object. Focal genes appear as large red nodes at the centre. Bridge genes are arranged radially, sized by node degree and coloured by mean expression. Solid coloured edges (red = gene1, blue = gene2) connect focal genes to bridge genes, with width proportional to shared term count. Thin dotted grey lines between bridge genes encode Jaccard pathway similarity.

See Also

PlotPairSynergy for the 4-panel synergy dashboard that embeds this network as one panel, AssessGenePair for extracting bridge genes programmatically.

Other Section_2_Visualization: PlotPairCrossType(), PlotPairDimplot(), PlotPairHeatmap(), PlotPairNetwork(), PlotPairScatter(), PlotPairSmoothed(), PlotPairSpatial(), PlotPairSummary(), PlotPairSynergy(), PlotPairViolin()

Examples

```
## Not run:
# Requires Bioconductor annotation packages (org.Hs.eg.db or org.Mm.eg.db)
PlotBridgeNetwork(seurat_obj, gene1 = "Adora2a", gene2 = "Ido1",
                  organism = "mouse")

## End(Not run)
```

PlotPairCrossType *Plot Cross-Cell-Type Interaction Heatmap*

Description

Visualises the cross-cell-type interaction structure for a gene pair as a heatmap. Each tile represents a directed cell-type pair (source type → neighbour type), coloured by the Pearson correlation between gene A expression in the source cells and gene B expression in the neighbouring cells.

This is the primary visualisation for the trans-cellular synergy metric introduced in scPairs 0.1.3. It reveals *which* cell-type interfaces carry the cross-type signal (e.g. Adora2a in T-cells correlated with Ido1 in dendritic cells).

Usage

```
PlotPairCrossType(
  object,
  gene1,
  gene2,
  result = NULL,
  assay = NULL,
  slot = "data",
  cluster_col = NULL,
  neighbourhood_k = 20,
  neighbourhood_reduction = "pca",
  min_cross_pairs = 30,
  min_pct_expressed = 0.01,
  show_n = TRUE,
  show_reverse = TRUE,
  diverging = TRUE,
  title = NULL
)
```

Arguments

object	A Seurat object.
gene1	Character; first gene.
gene2	Character; second gene.
result	Optional scPairs_pair_result from AssessGenePair. If NULL, the pair is assessed internally.
assay	Character; assay name. Default: DefaultAssay(object).
slot	Character; data slot. Default "data".
cluster_col	Character; meta.data column with cell-type labels. NULL = use Idents(object).
neighbourhood_k	Integer; k for KNN graph. Default 20.
neighbourhood_reduction	Character; reduction for KNN graph. Default "pca".
min_cross_pairs	Integer; minimum cross-type pairs per tile. Tiles with fewer pairs are greyed out. Default 30.
min_pct_expressed	Numeric; minimum percentage of cells (0-1) in a cell type that must express a gene. Default 0.01 (1%). Prevents spurious correlations with very sparse genes.
show_n	Logical; annotate each tile with the number of cross-type neighbour pairs. Default TRUE.
show_reverse	Logical; if TRUE (default), show a second panel for the reverse direction (gene2 in source → gene1 in neighbour).
diverging	Logical; use a diverging red–white–blue colour scale centred at 0. Default TRUE.
title	Character; overall title. NULL = auto-generated.

Value

A ggplot heatmap (or two-panel patchwork when show_reverse = TRUE) with cell-type pairs on axes and synergy enrichment encoded by colour.

See Also

AssessGenePair for cross-cell-type metrics, PlotPairDimplot for individual-cell UMAP display.

Other Section_2_Visualization: PlotBridgeNetwork(), PlotPairDimplot(), PlotPairHeatmap(), PlotPairNetwork(), PlotPairScatter(), PlotPairSmoothed(), PlotPairSpatial(), PlotPairSummary(), PlotPairSynergy(), PlotPairViolin()

Examples

```
# scpairs_testdata has clusters (seurat_clusters) and PCA already built in.
PlotPairCrossType(scpairs_testdata,
                  gene1 = "GENE3",
                  gene2 = "GENE4")
```

PlotPairDimplot *Plot Gene Pair Co-Expression on UMAP / Dimensionality Reduction*

Description

Displays the co-expression of two genes on the UMAP (or other reduction) embedding. Three panels show: gene 1 expression, gene 2 expression, and their element-wise product (co-expression intensity). This allows visual assessment of whether co-expressing cells cluster together.

Usage

```
PlotPairDimplot(
  object,
  gene1,
  gene2,
  reduction = "umap",
  assay = NULL,
  slot = "data",
  pt_size = 0.5,
  alpha = 0.8,
  title = NULL
)
```

Arguments

object	A Seurat object with a dimensionality reduction.
gene1	Character; first gene.
gene2	Character; second gene.
reduction	Character; reduction to use. Default "umap".
assay	Character; assay.
slot	Character; data slot.
pt_size	Numeric; point size.
alpha	Numeric; point alpha.
title	Character; overall title.

Value

A combined ggplot (patchwork) of three panels: individual gene expression and their co-expression product, all overlaid on the dimensionality-reduction embedding.

See Also

PlotPairSmoothed for KNN-smoothed panels, PlotPairScatter for gene-gene scatter, PlotPairSpatial for spatial data.

Other Section_2_Visualization: PlotBridgeNetwork(), PlotPairCrossType(), PlotPairHeatmap(), PlotPairNetwork(), PlotPairScatter(), PlotPairSmoothed(), PlotPairSpatial(), PlotPairSummary(), PlotPairSynergy(), PlotPairViolin()

Examples

```
# scpairs_testdata has a real UMAP embedding; GENE3 & GENE4 are co-expressed.
PlotPairDimplot(scpairs_testdata, gene1 = "GENE3", gene2 = "GENE4")
```

`PlotPairHeatmap` *Plot Synergy Score Heatmap*

Description

Displays a symmetric heatmap of synergy scores among a set of genes. Useful for visualising the overall co-expression landscape of top synergistic genes or genes of interest.

Usage

```
PlotPairHeatmap(
  result,
  top_n = 30,
  genes = NULL,
  cluster_genes = TRUE,
  low_color = "#F7FBFF",
  high_color = "#08306B",
  title = "Gene pair synergy heatmap"
)
```

Arguments

`result`	An `scPairs_result` or `scPairs_gene_result` object, or a `data.frame` with `gene1`, `gene2`, `synergy_score`.
`top_n`	Integer; include the top N genes by number of significant partnerships. Default 30.
`genes`	Character vector; specific genes to include. NULL = auto.
`cluster_genes`	Logical; cluster rows/columns by score similarity. Default TRUE.
`low_color`	Character; colour for low scores.
`high_color`	Character; colour for high scores.
`title`	Character; plot title.

Value

A `ggplot` object; rows and columns are genes, fill encodes synergy score.

See Also

`FindAllPairs`, `FindGenePairs`, `PlotPairNetwork` for an alternative network view.

Other Section_2_Visualization: `PlotBridgeNetwork()`, `PlotPairCrossType()`, `PlotPairDimplot()`, `PlotPairNetwork()`, `PlotPairScatter()`, `PlotPairSmoothed()`, `PlotPairSpatial()`, `PlotPairSummary()`, `PlotPairSynergy()`, `PlotPairViolin()`

Examples

```
result <- FindAllPairs(scpairs_testdata,
                       n_top_genes = 20,
                       top_n       = 15,
                       mode        = "expression",
                       verbose     = FALSE)
PlotPairHeatmap(result, top_n = 10)
```

`PlotPairNetwork` *Plot Gene Interaction Network*

Description

Draws a publication-ready gene interaction network from scPairs results. Nodes represent genes; edges represent synergistic relationships. Edge width encodes synergy score; edge colour encodes confidence. Node size optionally reflects the number of significant partners (degree centrality).

Usage

```
PlotPairNetwork(
  result,
  top_n = 50,
  min_score = 0,
  confidence = NULL,
  layout = "fr",
  node_color = "#2C3E50",
 edge_palette = c(High = "#E74C3C", Medium = "#F39C12", Low = "#95A5A6", NS = "#D5D8DC"),
  label_size = 3.5,
  title = NULL,
  show_legend = TRUE
)
```

Arguments

`result`	An object of class `"scPairs_result"`, `"scPairs_gene_result"`, or a `data.frame` / `data.table` with columns `gene1`, `gene2`, `synergy_score`.
`top_n`	Integer; show only the top N edges. Default 50.
`min_score`	Numeric; minimum synergy score to display an edge.
`confidence`	Character vector; filter to these confidence levels (e.g., `c("High", "Medium")`). NULL = no filter.
`layout`	Character; ggraph layout algorithm. Default `"fr"` (Fruchterman-Reingold).
`node_color`	Character; colour for nodes. Default `"#2C3E50"`.
`edge_palette`	Character vector of 3 colours for confidence (High, Medium, Low). Default blue-orange-grey scheme.

label_size	Numeric; node label font size.
title	Character; plot title.
show_legend	Logical.

Value

A ggplot object; nodes are genes, edges are gene pairs coloured and weighted by synergy score.

See Also

FindAllPairs, FindGenePairs, PlotPairHeatmap for an alternative matrix view.

Other Section_2_Visualization: PlotBridgeNetwork(), PlotPairCrossType(), PlotPairDimplot(), PlotPairHeatmap(), PlotPairScatter(), PlotPairSmoothed(), PlotPairSpatial(), PlotPairSummary(), PlotPairSynergy(), PlotPairViolin()

Examples

```
result <- FindAllPairs(scpairs_testdata,
                       n_top_genes = 20,
                       top_n       = 10,
                       mode        = "expression",
                       verbose     = FALSE)
PlotPairNetwork(result, top_n = 8)
```

PlotPairScatter *Scatter Plot of Two Genes (Cell-Level)*

Description

Plots cell-level expression of gene1 vs. gene2 as a scatter plot, coloured by cluster identity. Marginal density curves (optional) help reveal cluster-specific co-expression patterns.

Usage

```
PlotPairScatter(
  object,
  gene1,
  gene2,
  group_by = NULL,
  assay = NULL,
  slot = "data",
  pt_size = 0.5,
  alpha = 0.6,
  add_density = FALSE,
  title = NULL
)
```

Arguments

object	Seurat object.
gene1	Character; x-axis gene.
gene2	Character; y-axis gene.
group_by	Character; colour cells by this meta.data column.
assay	Character; assay.
slot	Character; data slot.
pt_size	Numeric.
alpha	Numeric.
add_density	Logical; add marginal density. Requires ggExtra package.
title	Character.

Value

A ggplot scatter plot; cells are coloured by cluster/group, with optional marginal density panels when **ggExtra** is installed.

See Also

PlotPairViolin, PlotPairDimplot.

Other Section_2_Visualization: PlotBridgeNetwork(), PlotPairCrossType(), PlotPairDimplot(), PlotPairHeatmap(), PlotPairNetwork(), PlotPairSmoothed(), PlotPairSpatial(), PlotPairSummary(), PlotPairSynergy(), PlotPairViolin()

Examples

```
PlotPairScatter(scpairs_testdata, "GENE3", "GENE4",
                group_by = "seurat_clusters")
```

PlotPairSmoothed *Enhanced Co-Expression Visualization with Neighbourhood Smoothing*

Description

A six-panel visualization that shows both raw and KNN-smoothed expression for a gene pair on the UMAP (or other reduction) embedding. The top row shows raw expression (gene1, gene2, product); the bottom row shows KNN-smoothed expression. This is particularly informative for gene pairs that are not co-expressed in the same cell but share neighbourhood-level co-expression patterns.

Usage

```
PlotPairSmoothed(
  object,
  gene1,
  gene2,
  reduction = "umap",
  smooth_reduction = "pca",
  k = 20,
  alpha = 0.3,
  assay = NULL,
  slot = "data",
  pt_size = 0.3,
  pt_alpha = 0.8,
  title = NULL
)
```

Arguments

object	A Seurat object with a dimensionality reduction.
gene1	Character; first gene.
gene2	Character; second gene.
reduction	Character; reduction for plotting. Default "umap".
smooth_reduction	Character; reduction for KNN graph. Default "pca".
k	Integer; neighbourhood size for smoothing. Default 20.
alpha	Numeric in [0,1]; self-weight for smoothing. Default 0.3.
assay	Character; assay.
slot	Character; data slot.
pt_size	Numeric; point size.
pt_alpha	Numeric; point alpha.
title	Character; overall title.

Value

A combined ggplot (patchwork) with 6 panels: three showing raw expression and three showing KNN-smoothed expression, for gene1, gene2, and their co-expression product.

See Also

PlotPairDimplot for the raw-only 3-panel variant, PlotPairSummary for a comprehensive multi-evidence summary.

Other Section_2_Visualization: PlotBridgeNetwork(), PlotPairCrossType(), PlotPairDimplot(), PlotPairHeatmap(), PlotPairNetwork(), PlotPairScatter(), PlotPairSpatial(), PlotPairSummary(), PlotPairSynergy(), PlotPairViolin()

Examples

```
# scpairs_testdata has PCA (smooth_reduction) and UMAP (reduction) ready.
PlotPairSmoothed(scpairs_testdata, gene1 = "GENE3", gene2 = "GENE4")
```

`PlotPairSpatial` *Plot Spatial Co-Expression Map*

Description

For spatial transcriptomics data, visualises the spatial distribution of two genes and their co-expression product on the tissue. Three panels are shown side by side:

1. Expression of gene 1.
2. Expression of gene 2.
3. Co-expression product (gene1 * gene2), highlighting spots where both genes are simultaneously active.

Usage

```
PlotPairSpatial(
  object,
  gene1,
  gene2,
  assay = NULL,
  slot = "data",
  pt_size = 1.2,
  alpha = 0.8,
  title = NULL
)
```

Arguments

`object`	A Seurat object with spatial coordinates.
`gene1`	Character; first gene.
`gene2`	Character; second gene.
`assay`	Character; assay.
`slot`	Character; data slot.
`pt_size`	Numeric; point size.
`alpha`	Numeric; point alpha.
`title`	Character; overall title.

Value

A combined `ggplot` (patchwork) with three panels: spatial expression of `gene1`, spatial expression of `gene2`, and their co-expression product, overlaid on physical tissue coordinates.

See Also

PlotPairDimplot for UMAP-based display, FindAllPairs which automatically computes spatial metrics when a spatial modality is detected.

Other Section_2_Visualization: PlotBridgeNetwork(), PlotPairCrossType(), PlotPairDimplot(), PlotPairHeatmap(), PlotPairNetwork(), PlotPairScatter(), PlotPairSmoothed(), PlotPairSummary(), PlotPairSynergy(), PlotPairViolin()

Examples

```
## Not run:
# Requires a Seurat object with spatial assay (e.g. Visium, MERFISH)
PlotPairSpatial(spatial_obj, gene1 = "CD8A", gene2 = "CD8B")

## End(Not run)
```

PlotPairSummary *Comprehensive Synergy Summary Plot*

Description

A multi-panel publication-ready figure combining:

1. Raw UMAP co-expression (3 panels)
2. KNN-smoothed UMAP (3 panels)
3. Per-cluster expression comparison
4. Metric radar/bar chart

Usage

```
PlotPairSummary(
  object,
  gene1,
  gene2,
  result = NULL,
  reduction = "umap",
  smooth_reduction = "pca",
  k = 20,
  alpha = 0.3,
  assay = NULL,
  slot = "data",
  pt_size = 0.3
)
```

Arguments

object	A Seurat object.
gene1	Character; first gene.
gene2	Character; second gene.
result	Optional scPairs_pair_result from AssessGenePair(). If NULL, assessment is run internally.
reduction	Character; reduction for plotting.
smooth_reduction	Character; reduction for KNN graph.
k	Integer; KNN k for smoothing.
alpha	Numeric; smoothing alpha.
assay	Character; assay.
slot	Character; data slot.
pt_size	Numeric; point size.

Value

A combined ggplot (patchwork) with up to 10 panels: raw UMAP co-expression (3 panels), KNN-smoothed UMAP (3 panels), per-cluster expression bar chart, and metric evidence bar chart.

See Also

PlotPairSmoothed, AssessGenePair.

Other Section_2_Visualization: PlotBridgeNetwork(), PlotPairCrossType(), PlotPairDimplot(), PlotPairHeatmap(), PlotPairNetwork(), PlotPairScatter(), PlotPairSmoothed(), PlotPairSpatial(), PlotPairSynergy(), PlotPairViolin()

Examples

```
PlotPairSummary(scpairs_testdata, gene1 = "GENE3", gene2 = "GENE4")
```

PlotPairSynergy *Visualize Synergistic Relationship Between Gene Pairs*

Description

Publication-ready multi-panel visualization that integrates prior knowledge, expression evidence, and neighbourhood context to show the synergistic relationship between two genes. This goes beyond co-expression to reveal *why* two genes may be functionally synergistic.

Usage

```
PlotPairSynergy(
  object,
  gene1,
  gene2,
  prior_net = NULL,
  organism = "mouse",
  reduction = "umap",
  smooth_reduction = "pca",
  k = 20,
  alpha = 0.3,
  cluster_col = NULL,
  assay = NULL,
  slot = "data",
  top_bridges = 10,
  pt_size = 0.3
)
```

Arguments

object	A Seurat object.
gene1	Character; first gene.
gene2	Character; second gene.
prior_net	Optional prior network from .build_prior_network(). If NULL, built automatically.
organism	Character; "mouse" or "human". Used if prior_net is NULL.
reduction	Character; reduction for UMAP plotting.
smooth_reduction	Character; reduction for KNN.
k	Integer; KNN k.
alpha	Numeric; smoothing alpha.
cluster_col	Character; cluster column in meta.data.
assay	Character; assay.
slot	Character; data slot.
top_bridges	Integer; maximum bridge genes to show.
pt_size	Numeric; point size.

Value

A combined ggplot (patchwork) with up to 4 panels:

1. UMAP coloured by per-cell neighbourhood synergy score.
2. Bridge gene network showing shared GO/KEGG pathway intermediaries.
3. Per-cluster expression bar chart for both genes.
4. Multi-evidence metric comparison bar chart (expression + prior).

Falls back gracefully when prior knowledge is unavailable (panels 2 and 4 are omitted).

See Also

PlotBridgeNetwork for a standalone bridge gene network, AssessGenePair for the underlying metrics.

Other Section_2_Visualization: PlotBridgeNetwork(), PlotPairCrossType(), PlotPairDimplot(), PlotPairHeatmap(), PlotPairNetwork(), PlotPairScatter(), PlotPairSmoothed(), PlotPairSpatial(), PlotPairSummary(), PlotPairViolin()

Examples

```
## Not run:
# Requires Bioconductor annotation packages: org.Hs.eg.db or org.Mm.eg.db
# and AnnotationDbi.
PlotPairSynergy(scpairs_testdata, gene1 = "GENE3", gene2 = "GENE4",
                organism = "human")

## End(Not run)
```

PlotPairViolin *Violin Plot of Pair Expression Across Clusters*

Description

Displays side-by-side violin plots of two genes across cell clusters or groups, enabling visual assessment of whether their expression patterns are coordinated across populations.

Usage

```
PlotPairViolin(
  object,
  gene1,
  gene2,
  group_by = NULL,
  assay = NULL,
  slot = "data",
  pt_size = 0,
  title = NULL
)
```

Arguments

object	Seurat object.
gene1	Character; first gene.
gene2	Character; second gene.
group_by	Character; column in meta.data for grouping. NULL = Idents.
assay	Character; assay.

slot	Character; data slot.
pt_size	Point size for jitter (0 = no points).
title	Character.

Value

A ggplot with violin (and optional jitter) panels for gene1, gene2, and their expression product, split by group.

See Also

PlotPairScatter, PlotPairDimplot.

Other Section_2_Visualization: PlotBridgeNetwork(), PlotPairCrossType(), PlotPairDimplot(), PlotPairHeatmap(), PlotPairNetwork(), PlotPairScatter(), PlotPairSmoothed(), PlotPairSpatial(), PlotPairSummary(), PlotPairSynergy()

Examples

```
PlotPairViolin(scpairs_testdata, "GENE3", "GENE4",
               group_by = "seurat_clusters")
```

---

print.scPairs_gene_result

Print method for scPairs_gene_result

Description

Print method for scPairs_gene_result

Usage

```
## S3 method for class 'scPairs_gene_result'
print(x, ...)
```

Arguments

x	An scPairs_gene_result object.
...	Ignored.

Value

The input object x, returned invisibly.

print.scPairs_pair_result *Print method for scPairs_pair_result*

Description

Print method for scPairs_pair_result

Usage

```
## S3 method for class 'scPairs_pair_result'
print(x, ...)
```

Arguments

x	An scPairs_pair_result object.
...	Ignored.

Value

The input object x, returned invisibly.

print.scPairs_result *Print method for scPairs_result*

Description

Print method for scPairs_result

Usage

```
## S3 method for class 'scPairs_result'
print(x, ...)
```

Arguments

x	An scPairs_result object.
...	Ignored.

Value

The input object x, returned invisibly.

`scpairs_testdata` *Synthetic Seurat Test Object for scPairs Examples and Tests*

Description

A minimal synthetic Seurat object with deliberately injected co-expression patterns, intended for use in package examples and unit tests. The object ships with normalised expression, variable-feature selection, scaled data, a 5-component PCA, and a 2-D UMAP embedding, so it can be passed directly to every scPairs discovery and visualisation function without any additional setup.

Usage

```
data(scpairs_testdata)
```

Format

A `Seurat` object with:

Assay RNA
- **counts**: raw integer count matrix (20 genes x 100 cells).
- **data**: log-normalised expression matrix.
- **scale.data**: z-score-scaled matrix for all variable features.

Reductions
- **pca**: 5-component principal-component embedding.
- **umap**: 2-D UMAP embedding derived from the top 5 PCs.

Metadata • **seurat_clusters**: factor with three balanced cluster labels (`"1"`, `"2"`, `"3"`).

Genes `GENE1`–`GENE20` (synthetic gene identifiers).

Cells `CELL001`–`CELL100`.

Details

Two co-expression patterns are injected at data-generation time:

- **GENE3 & GENE4** – strongly correlated across all 100 cells (Pearson r approximately 0.89 in the normalised data). These are the recommended genes for discovery and assessment examples.
- **GENE1 & GENE2** – moderately correlated within cluster 1 only (cluster-specific pattern).
- All remaining gene pairs are near-independent noise.

The data are generated with a fixed random seed (`set.seed(7391)`) so the object is fully reproducible. The generation script is provided in `data-raw/make_testdata.R`.

Source

Generated by `data-raw/make_testdata.R` with `set.seed(7391)`.

See Also

`FindAllPairs`, `FindGenePairs`, `AssessGenePair`

Examples

```
# Load and inspect the object
data(scpairs_testdata)
scpairs_testdata

# Verify the injected GENE3 / GENE4 co-expression
norm <- SeuratObject::LayerData(scpairs_testdata, layer = "data")
cor(as.numeric(norm["GENE3", ]), as.numeric(norm["GENE4", ]))
```

Index

∗ **Section_0_Data**
scpairs_testdata, 26
∗ **Section_1_Discovery**
AssessGenePair, 2
FindAllPairs, 5
FindGenePairs, 7
∗ **Section_2_Visualization**
PlotBridgeNetwork, 9
PlotPairCrossType, 11
PlotPairDimplot, 13
PlotPairHeatmap, 14
PlotPairNetwork, 15
PlotPairScatter, 16
PlotPairSmoothed, 17
PlotPairSpatial, 19
PlotPairSummary, 20
PlotPairSynergy, 21
PlotPairViolin, 23
∗ **datasets**
scpairs_testdata, 26

AssessGenePair, 2, *7*, *9*, *11, 12*, *21*, *23*, *26*

FindAllPairs, *4*, 5, *9*, *14*, *16*, *20*, *26*
FindAllPairs(), *7*
FindGenePairs, *4*, 7, *7*, *14*, *16*, *26*

PlotBridgeNetwork, 9, *12–14*, *16–18*, *20, 21*, *23, 24*
PlotPairCrossType, 11, *11*, *13, 14*, *16–18*, *20, 21*, *23, 24*
PlotPairDimplot, *11, 12*, 13, *14*, *16–18*, *20*, *21*, *23, 24*
PlotPairHeatmap, *7*, *11–13*, 14, *16–18*, *20*, *21*, *23, 24*
PlotPairNetwork, *7*, *9*, *11–14*, 15, *17, 18*, *20*, *21*, *23, 24*
PlotPairScatter, *11–14*, 16, *16*, *18*, *20, 21*, *23, 24*
PlotPairSmoothed, *11–14*, *16*, 17, *17*, *20, 21*, *23, 24*
PlotPairSpatial, *11–14*, *16–18*, 19, *21*, *23*, *24*
PlotPairSummary, *4*, *11–14*, *16–18*, 20, *20*, *23, 24*
PlotPairSynergy, *4*, *11–14*, *16–18*, *20*, 21, *21*, *24*
PlotPairViolin, *11–14*, *16–18*, *20, 21*, 23, *23*
print.scPairs_gene_result, 24
print.scPairs_pair_result, 25
print.scPairs_result, 25

scpairs_testdata, 26
Seurat, *26*