Package: CACIMAR (via r-universe)

September 2, 2024

Title cross-species analysis of cell identities, markers and regulations

Version 1.0.0

Description A toolkit to perform cross-species analysis based on scRNA-seq data. CACIMAR contains 5 main features. (1) identify Markers in each cluster. (2) Cell type annotaion (3) identify conserved markers. (4) identify conserved cell types. (5) identify conserved modules of regulatory networks.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Imports pheatmap, pbapply, psych, ROCR, gridExtra, reshape2, dplyr, grDevices, stats, methods, viridisLite, biomaRt, ape, ggplot2, igraph, purrr, stringr

Depends R (>= 4.0), Seurat, tidyverse, ggtree

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Repository https://jiang-junyao.r-universe.dev

RemoteUrl https://github.com/jiang-junyao/cacimar

RemoteRef HEAD

RemoteSha f6cb72d9554855d42d46c15c5c61f4d5496cb42a

Contents

add_cellchat_prob *filter cellchat CCC networks*

Description

Use conserved CCC to filter cellchat CCC networks

Usage

```
add_cellchat_prob(conserved_ccc_df, cellchat_df)
```

Arguments

```
cellchat_df
```

buildHomDatabase *Build homologous gene database*

Description

Build homologous gene database according to Vertebrate Homology data in MGI database. This function currently supports ten species: cattle, chicken, chimpanzee, dog, frog, human, macaque, mouse, rat, and zebrafish. After building the database, this function also integrates biomaRt to add ENSEMBEL ID for each gene in the database.

Usage

```
buildHomDatabase(MGI, Species_name1, Species_name2)
```

Arguments

MGI	MGI database, download from http://www.informatics.jax.org/
Species_name1	The name of the first species. input 'mm' for mouse, 'hs' for human, 'zf' for zebrafish, 'ch' for chicken, 'cf' for dog, 'pt' for chimpanzee, 'xt' for frog, 'rn' for rat, 'bt' for cattle, and 'rh' for macaque.
Species_name2	The name of the second species. input 'mm' for mouse, 'hs' for human, 'zf' for zebrafish, 'ch' for chicken, 'cf' for dog, 'pt' for chimpanzee, 'xt' for frog, 'rn' for rat, 'bt' for cattle, and 'rh' for macaque.

Value

homologous gene database of two species

CACIMAR_cols *CACIMAR colors palette*

Description

CACIMAR colors palette

Usage

```
CACIMAR_cols(color_number)
```

Arguments

color_number	numeric, indicating used colors number

Value

vector of colors

Examples

```
CACIMAR_cols(10)
CACIMAR_cols(20)
```

Caculate_cell_pair_cci_score *Caculate conserved score of cell-cell interaction*

Description

caculate the conserved score of cell-cell interaction with summed weights of ligand-receptor interactions for two species

Usage

```
Caculate_cell_pair_cci_score(
  conserved_result_df,
  species1_cci,
  species2_cci,
  conserved_cell_types_df,
  species_name1,
  species_name2
)
```

Arguments

conserved_result_df	dataframe, result from function Identify_Conserved_CCI1
species1_cci	dataframe, result from SingleCellSignalR of species1
species2_cci	dataframe, result from SingleCellSignalR of species2
conserved_cell_types_df	dataframe, contain the conserved cell type for each species, like conserved_cell_types_mm_zf <- data.frame("mm" = c("mmRods", "mmRod BC", "mmCones", "mmPericytes", "mmV/E cells", "mmRGC", "mmGABAergic AC", "mmRPE", "mmResting MG", "mmActivated MG", "mmMicroglia"), "zf" = c("zfRods", "zfCone BC", "zfCones", "zfPericytes", "zfV/E cells", "zfRGC", "zfGABAergic AC", "zfRPE", "zfResting MG", "zfActivated MG", "zfMicroglia"))
species_name1	two character to represent species1, like "mm". You should set this value from dataframe species_names_ref.rda
species_name2	two character to represent species2, like "zf". You should set this value from dataframe species_names_ref.rda

Value

dataframe, conserved Weights table

Examples

```
conserved_cell_types_mm_zf <- data.frame("mm" = c("mmRods", "mmRod BC", "mmCones", "mmPericytes", "mmV/E cells", "
cci_conserved_Weights_table_mm_zf <- Caculate_cell_pair_cci_score(conserved_result_df=conserved_result_mm_zf,
species1_cci = SingleCellSignalR_mouse_result,
species2_cci = SingleCellSignalR_zebrafish_result,
conserved_cell_types_df = conserved_cell_types_mm_zf,
species_name1 = "mm",
species_name2 = "zf")
```

Caculate_cell_pair_cci_score2

Caculate conserved score of cell-cell interaction for three species

---

Description

caculate the conserved score of cell-cell interaction with summed weights of ligand-receptor interactions for three species

Usage

```
Caculate_cell_pair_cci_score2(
  conserved_result_df,
  species1_cci,
  species2_cci,
```

```
  species3_cci,
  conserved_cell_types_df,
  species_name1,
  species_name2,
  species_name3
)
```

Arguments

conserved_result_df	dataframe, result from function Identify_Conserved_CCI2
species1_cci	dataframe, result from SingleCellSignalR of species1
species2_cci	dataframe, result from SingleCellSignalR of species2
species3_cci	dataframe, result from SingleCellSignalR of species3
conserved_cell_types_df	dataframe, contain the conserved cell type for each species, like conserved_cell_types_mm_zf_ch <- data.frame('mm' = c("mmResting MG","mmGABAergic AC", "mmRGC", "mmCones"), "zf" = c("zfResting MG", "zfGABAergic AC", "zfRGC", "zf-Cones"), 'ch' = c("chResting MG","chGABAergic AC", "chRGC", "chCones"))
species_name1	two character to represent species1, like "mm". You should set this value from dataframe species_names_ref.rda
species_name2	two character to represent species2, like "zf". You should set this value from dataframe species_names_ref.rda
species_name3	two character to represent species3, like "ch". You should set this value from dataframe species_names_ref.rda

Value

dataframe, conserved Weights table

---

calculate_Weights *Sum weight of cell-cell interactions*

Description

Here calculate overall weight of cell-cell interactions within each pair of cell types

Usage

```
calculate_Weights(
  species1_cci,
  species2_cci,
  specie_name1 = "Mm",
  specie_name2 = "Zf"
)
```

Arguments

species1_cci	data frame, cell-cell interactions result with perform_CCI_analysis of species 1
species2_cci	data frame, cell-cell interactions result with perform_CCI_analysis of species 2
specie_name1	character, name for species 1, like "Mm"
specie_name2	character, name for species 2, like "Zf"

Value

data frame of weight result

Examples

```
all_weight_df_long <- calculate_Weights(species1_cci = SingleCellSignalR_mouse_result,
species2_cci = SingleCellSignalR_zebrafish_result)

head(all_weight_df_long)
          source      target    weight species scale_weight           Source          Source2
1    Activated MG Activated MG 152.07967     Mm  0.008256281    MmActivated MG    MmActivated MG
2     Astrocytes Activated MG 230.97254     Mm  0.012539310     MmAstrocytes     MmAstrocytes
3         Cones Activated MG  51.89929     Mm  0.002817570         MmCones          MmCones
4   GABAergic AC Activated MG  99.92955     Mm  0.005425093   MmGABAergic AC    MmGABAergic AC
5  Glycinergic AC Activated MG  78.95619     Mm  0.004286467  MmGlycinergic AC  MmGlycinergic AC
6 Horizontal cells Activated MG 106.23687     Mm  0.005767513 MmHorizontal cells MmHorizontal cells
```

ChordDiagram *Make a ChordDiagram*

Description

Construct a ChordDiagram to show the conservation score for intercellular interactions

Usage

```
ChordDiagram(
  net,
  Score_factor_size = 20,
  brewer_pal_used = "Set3",
  grid.col = NULL,
  grid.border = "#F8F8F8",
  is_link_colors_threshold = TRUE,
  link_colors_threshold = 0.5,
  link_colors_down_threshold_col = "grey",
  filename = "chordDiagram.pdf",
  order_grid = NULL,
  directional = 1,
  direction.type = c("arrows"),
  diffHeight = -0.03,
```

```
  annotationTrack = "grid",
  reduce = -1,
  link.arr.type = "big.arrow",
  link.border = "white",
  link.lwd = 1,
  link.sort = TRUE,
  link.decreasing = TRUE,
  link.largest.ontop = TRUE,
  link.overlap = F,
  transparency = 0,
  cex = 1.5,
  ylim_edit = 0,
  facing = "clockwise",
  niceFacing = TRUE,
  col = "black",
  adj = c(0, 0.5),
  start.degree = -5,
  magnify_set = TRUE,
  picture_width = 14,
  picture_height = 10
)
```

Arguments

net	data frame, a data frame contain the source, target, and weight
Score_factor_size	numeric, scale the weight score for showing
brewer_pal_used	character, the brewer_pal in RColorBrewer, like "Set1", "Set3", "Paired"
grid.col	vector, color vectors with names. By default, it is NULL and it will asign colors automatically.
grid.border	character, corlors for borders of grids. If it is NULL, the border color is same as grid color
is_link_colors_threshold	logical, whether change the colors of links which below a threshold to a specific color
link_colors_threshold	numeric, the threshold used to change colors for links
link_colors_down_threshold_col	character, specific corlor for links which below a threshold
filename	character, save the plot with this filename
order_grid	vector, a ordered vector for grid. By default, it is NULL
directional	Directions for the links. 1 means the direction is from the first column in df to the second column, -1 is the reverse, 0 is no direction, and 2 for two directional. The value can be a vector which has same length as number of rows in df
direction.type	type for representing directions. Can be one or two values in "diffHeight" and "arrows"

diffHeight	numeric, The difference of height between two 'roots' if directional is set to TRUE
annotationTrack	annotation for the track. It can set to c("name","grid", "axis"), or one of them.
reduce	numeric, if the ratio of the width of certain grid compared to the whole circle is less than this value, the grid is removed on the plot. Set it to value less than zero if you want to keep all tiny grid.
link.arr.type	type for the arrow, "big.arrow" is set by default
link.border	character, corlors for links
link.lwd	numeric, width for link borders
link.sort	logical, whether sort links on every sector based on the width of the links on it
link.decreasing	logical, for link.sort
link.largest.ontop	logical, controls the order of adding links
link.overlap	logical, whether the links that come or end in a same sector overlap
transparency	numeric, transparency for link colors
cex	numeric, font size for the text in annotationTrack
ylim_edit	numeric, control the text height from the grid
facing	control the direction of the text in annotationTrack, it can be one of c("inside", "outside", "reverse.clockwise", "clockwise", "downward", "bending", "bending.inside", "bending.outside")
niceFacing	logical, adjusted the text to fit human eyes
col	character, colors for the text in annotationTrack
adj	offset for text, like c(0, 0.5). By default the text position adjustment is either horizontal or vertical in the canvas coordinate system.
start.degree	numeric, rotation the ChordDiagram with this angle
picture_width	numeric, width of the ChordDiagram for saving
picture_height	numeric, height of the ChordDiagram for saving

Value

saving ChordDiagram in current directory

Examples

```
load(system.file("extdata", "CCC_conserved_sumary.rda", package = "CACIMAR"))
cci_data = CCC_conserved_sumary[, c("Source", "Target", "score_weight")]
ChordDiagram(net = cci_data, filename = "chordDiagram.pdf", link_colors_threshold = 0.75)
```

`conserved_interaction_score` *Caculate conserved interaction score and Make data for ChordDiagram*

Description

Caculate conserved interaction score and Make data for ChordDiagram

Usage

```
conserved_interaction_score(
  conserved_result_species,
  SingleCellSignalR_sp1_result,
  SingleCellSignalR_sp2_result,
  SingleCellSignalR_sp3_result,
  conserved_cell_types_df,
  species_name1 = "mm",
  species_name2 = "zf",
  species_name3 = "ch"
)
```

Arguments

`conserved_result_species`	dataframe, result from function Identify_Conserved_CCI2
`SingleCellSignalR_sp1_result`	dataframe, result from SingleCellSignalR of species1
`SingleCellSignalR_sp2_result`	dataframe, result from SingleCellSignalR of species2
`SingleCellSignalR_sp3_result`	dataframe, result from SingleCellSignalR of species3
`conserved_cell_types_df`	dataframe, contain the conserved cell type for each species, like conserved_cell_types_mm_zf_ch <- data.frame('mm' = c("mmResting MG","mmGABAergic AC", "mmRGC", "mmCones"), "zf" = c("zfResting MG", "zfGABAergic AC", "zfRGC", "zfCones"), 'ch' = c("chResting MG","chGABAergic AC", "chRGC", "chCones"))
`species_name1`	two character to represent species1, like "mm". You should set this value from dataframe species_names_ref.rda
`species_name2`	two character to represent species2, like "zf". You should set this value from dataframe species_names_ref.rda
`species_name3`	two character to represent species3, like "ch". You should set this value from dataframe species_names_ref.rda

Value

list of conserved Weights table and data for chordDiagram

Examples

```
#not run
#conserved_result_mm_zf_ch is the result from function Identify_Conserved_CCI2
conserved_cci_result <- conserved_interaction_score(conserved_result_species = conserved_result_mm_zf_ch,
SingleCellSignalR_sp1_result = SingleCellSignalR_mouse_result,
SingleCellSignalR_sp2_result = SingleCellSignalR_zebrafish_result,
SingleCellSignalR_sp3_result = SingleCellSignalR_chick_result,
conserved_cell_types_df = conserved_cell_types_mm_zf_ch,
species_name1 = "mm",
species_name2 = "zf",
species_name3 = "ch")
```

---

create_sankey *Create sankey plot*

Description

Build a sankey plot to show the cell-cell interaction profile of two species

Usage

```
create_sankey(
  links,
  specie_name1 = "Mm",
  specie_name2 = "Zf",
  output_file = "sankey.html",
  colors_file = NULL,
  brewer.pal_set = "Set3",
  node_width = 18,
  node_padding = 15,
  node_stroke_width = 0,
  node_corner_radius = 0,
  drag_x = TRUE,
  drag_y = TRUE,
  units = "TWh",
  node_pos_x = "group",
  align = "none",
  scale_node_breadths_by_string = F,
  show_node_values = FALSE,
  height = 1400,
  width = 1800,
  linkColor = "#A0A0A0",
  link_type = "bezier",
```

```
  curvature = 0.5,
  link_opacity = 5,
  link_gradient = FALSE,
  node_shadow = FALSE,
  node_label_margin = 5,
  zoom = TRUE,
  font_family = NULL,
  iterations = 0,
  x_scaling_factor = 1.4,
  font_size = 16,
  ...
)
```

Arguments

links	dataframe, a links file containing source, target, and weight
specie_name1	character, name for specie 1
specie_name2	character, name for specie 2
output_file	character, save sankey plot with this file name
colors_file	character, file path for the color file, it is NULL by default and it will assign colors automatically
brewer.pal_set	if colors_file is null, the brewer.pal can be choose, like "Set1", "Set3", "Paired", ...
node_width	numeric, width of the node
node_padding	numeric, distance between the node
node_stroke_width	numeric, width of the stroke around nodes
node_corner_radius	numeric, the radius of the node
drag_x	logical, TRUE indicates that the plot can be horizontally dragged
drag_y	logical, TRUE indicates that the plot can be vertically dragged
units	character, name for units
node_pos_x	character, variable used for grouping nodes on the x-axis
align	character, alignment of the nodes. One of 'right', 'left', 'justify', 'center', 'none'. If 'none', then the labels of the nodes are always to the right of the node
scale_node_breadths_by_string	logical, Put nodes at positions relatively to string lengths - only work well currently with align='none'
show_node_values	logical, Show values above nodes. Might require and increased node margin.
height	numeric, height of the sankey plot for saving
width	numeric, width of the sankey plot for saving
linkColor	numeric, color of links

link_type	character, one of 'bezier', 'l-bezier', 'trapezoid', 'path1' and 'path2'
curvature	numeric, curvature parameter for bezier links - between 0 and 1
link_opacity	numeric, opacity of links
link_gradient	logical, add a gradient to the links
node_shadow	logical, add a shadow to the nodes
node_label_margin	numeric, distance between the node and font
zoom	logical, value to enable (TRUE) or disable (FALSE) zooming
iterations	numeric, number of iterations in the diagramm layout for computation of the depth (y-position) of each node. Note: this runs in the browser on the client so don't push it too high
x_scaling_factor	numeric, scale the computed x position of the nodes by this value
font_size	numeric, size of the font of the plot
...	param of sankeyD3::sankeyNetwork()

Value

a sankey plot

Examples

```
create_sankey(links = all_weight_df_long[, c("Source2", "target", "scale_weight")],
output_file = "sankey_scale_weight.html",
specie_name1 = "Mm",
specie_name2 = "Zf",
colors_file = NULL)
```

FormatConservedMarkers

Format Conserved markers to plot heatmap

Description

Format Conserved markers to plot heatmap

Usage

```
FormatConservedMarkers(ConservedMarker)
```

Arguments

ConservedMarker	Result from 'Identify_ConservedMarkers'

Value

list contains two data.frame

Examples

```
load(system.file("extdata", "zf_mm_markers.rda", package = "CACIMAR"))
ConservedMarker <- Identify_ConservedMarkers(OrthG_Mm_Zf,Mm_marker,Zf_marker,
Species_name1 = 'mm',Species_name2 = 'zf')
MarkersPlot <- FormatConservedMarkers(ConservedMarker)
```

Format_Markers_Frac *Format marker genes for plotting*

Description

Order the gene expression in each cluster to make the heatmap look better

Usage

```
Format_Markers_Frac(Marker_genes)
```

Arguments

Marker_genes	data.frame, generated by Identify_Markers

Examples

```
data("pbmc_small")
all.markers <- Identify_Markers(pbmc_small)
all.markers2 <- Format_Markers_Frac(all.markers)
```

---

get_average_expression *Geometric mean of ligand and receptor*

Description

This function caculates the geometric mean of ligand and receptor. Average expression of ligand comes from source celltype, and verage expression of receptor comes from target celltype

Usage

```
get_average_expression(
  specie_conserved_CCC,
  seurat_object,
  avg_group,
  species_set,
  assay_set
)
```

Arguments

specie_conserved_CCC	dataframe, also result of the ligand receptor analysis(for example, Conserved-CCI$Conserved_ligand_receptor$sp1_orthg_ccc_df). It must contain 4 column for ligand, receptor, source celltype, target celltype
seurat_object	a seurat object contains the expression data for celltypes of source and target
avg_group	a character, which variable used to average the expression, usally "celltype"
species_set	species_set should only contain two characters, like "Mm" to label the species
assay_set	Assay to get the expression, like "RNA", or "SCT"

Examples

```
LRpair_show <- get_average_expression(CCC_conserved_sumary = ConservedCCI[[2]], specie_conserved_CCC = ConservedC
```

Heatmap_Cor *plot the heatmap of marker genes across different species*

Description

plot the heatmap of marker genes across different species

Usage

```
Heatmap_Cor(
  RNA1,
  RowType1 = "",
  ColType1 = "",
  cluster_cols = T,
  cluster_rows = F,
  Color1 = NULL,
  ...
)
```

Arguments

RNA1	correlation of expression in each cell type
RowType1	character, indicating the cell types that you want to show on the row in heatmap. RowType1=" means show all cell types
ColType1	character, indicating the cell types that you want to show on the column in heatmap. RowType1=" means show all cell types
cluster_cols	boolean values determining if columns should be clustered or hclust object
cluster_rows	boolean values determining if rows should be clustered or hclust object
Color1	vector of colors used in heatmap
...	parameter in pheatmap

Value

pheatmap object

Examples

```
load(system.file("extdata", "network_example.rda", package = "CACIMAR"))
n1 <- Identify_ConservedNetworks(OrthG_Mm_Zf,mmNetwork,zfNetwork,'mm','zf')
Heatmap_Cor(n1[[2]],cluster_cols=TRUE, cluster_rows=FALSE)
```

Identify_CellType *Identify cell type of each cluster*

Description

This function has three steps to identify cell type of each cluster. (1) Calculate the power of each known marker based on AUC (area under the receiver operating characteristic curve of gene expression) which indicates the capability of marker i from cell type m to distinguish cluster j and the other clusters. (2) Calculate the united power (UP) for cell type m across each cluster j. (3) For each cluster j we determine the cell type according to UP. Generally, the cluster beongs to the cell type which have the highest united power or higher than the threshold of the united power (for example > 0.9 power).

Usage

```
Identify_CellType(seurat_object, Marker_gene_table)
```

Arguments

seurat_object	seurat object
Marker_gene_table	data.frame, indicating marker gene and its corresponding cell type. Marker_gene_table should contain two columns: 'CellType' represent corresponding cell types of each marker and 'Marker' represent Markers

Value

Cell type with the highest power in each cluster

Examples

```
KnownMarker=data.frame(c('AIF1','BID','CCL5','CD79A','CD79B','MS4A6A'),c('a','a','a','b','b','b'))
data("pbmc_small")
colnames(KnownMarker)=c('Marker','CellType')
CT <- Identify_CellType(pbmc_small,KnownMarker)
```

`Identify_ConservedCellTypes` *Identify conserved cell types based on power of genes and orthologs database*

Description

Identify conserved cell types based on power of genes and orthologs database

Usage

```
Identify_ConservedCellTypes(
  OrthG,
  Species1_Marker_table,
  Species2_Marker_table,
  Species_name1,
  Species_name2
)
```

Arguments

`OrthG`	ortholog genes database
`Species1_Marker_table`	data.frame of species 1, should contain three column: 'gene', 'cluster' and 'power'
`Species2_Marker_table`	data.frame of species 2, should contain three column: 'gene', 'cluster' and 'power'
`Species_name1`	character, indicating the species names of Species1_Marker_table
`Species_name2`	character, indicating the species names of Species2_Marker_table

Value

list contains two elements: first one is details of conserved cell types, second one is matrix of cell types conserved score

Author(s)

Jie Wang

Examples

```
load(system.file("extdata", "CellTypeAllMarkers.rda", package = "CACIMAR"))
expression <- Identify_ConservedCellTypes(OrthG_Mm_Zf,mm_Marker[1:30,],zf_Marker[1:30,],'mm','zf')
```

`Identify_ConservedMarkers` *Identify orthologs marker genes for two species*

Description

Identify orthologs marker genes for two species based on orthologs database

Usage

```
Identify_ConservedMarkers(
  OrthG,
  Species1_Marker_table,
  Species2_Marker_table,
  Species_name1,
  Species_name2,
  match_cell_name = NULL,
  filter_marker = TRUE
)
```

Arguments

`OrthG`	ortholog genes database
`Species1_Marker_table`	data.frame of species 1, should contain 'gene' and 'Allcluster' columns.
`Species2_Marker_table`	data.frame of species 2, should contain 'gene' and 'Allcluster' columns.
`Species_name1`	character, indicating the species names of Species1_Marker_table.
`Species_name2`	character, indicating the species names of Species2_Marker_table
`match_cell_name`	characters contained in both cell names to match similar cell types
`filter_marker`	logical, indicating whether filter markers

Value

Data frame of conserved markers

Examples

```
load(system.file("extdata", "zf_mm_markers.rda", package = "CACIMAR"))
ConservedMarker <- Identify_ConservedMarkers(OrthG_Mm_Zf,Mm_marker,Zf_marker,
Species_name1 = 'mm',Species_name2 = 'zf')
```

`Identify_ConservedNetworks` *Identify conserved regulatory networks*

Description

Use Score of Conserved network to identify conserved regulatory network modules based on homologous genes databased and topology of networks

Usage

```
Identify_ConservedNetworks(
  OrthG,
  Species1_GRN,
  Species2_GRN,
  Species_name1,
  Species_name2
)
```

Arguments

`OrthG`	ortholog genes database
`Species1_GRN`	gene regulatory network of species 1
`Species2_GRN`	gene regulatory network of species 2
`Species_name1`	character, indicating the species names of Species1_GRN
`Species_name2`	character, indicating the species names of Species2_GRN

Value

list contains two df. First df contains details of conserved regulatory network, second df contains NCS between module pairs

Examples

```
load(system.file("extdata", "gene_network.rda", package = "CACIMAR"))
n1 <- Identify_ConservedNetworks(OrthG_Mm_Zf,mm_gene_network,zf_gene_network,'mm','zf')
```

Identify_Conserved_CCI

Title

Description

Title

Usage

```
Identify_Conserved_CCI(OrthG, sp1_ccc, sp2_ccc, Species_name1, Species_name2)
```

Arguments

OrthG	ortholog genes database
sp1_ccc	cell-cell interactions in species1. First column should be ligand, second column should be target, third column should be corresponding cell type of ligand, fourth column should be corresponding cel type of target, fifth column should be weight of ligand-receptor interaction
sp2_ccc	cell-cell interactions in species2. First column should be ligand, second column should be target, third column should be corresponding cell type of ligand, fourth column should be corresponding cel type of target, fifth column should be weight of ligand-receptor interaction
Species_name1	character, indicating the species names of Species1_GRN
Species_name2	character, indicating the species names of Species2_GRN

Identify_Conserved_CCI1

Identify conserved cell-cell interaction from two species

Description

get conserved cell-cell interaction between two species

Usage

```
Identify_Conserved_CCI1(
  species1_cci,
  species2_cci,
  species_names_ref = NULL,
  conserved_cell_types_df,
  species_name1 = "mm",
  species_name2 = "zf",
  HOM_matrix = NULL
)
```

Arguments

species1_cci	result of SingCellSignalR of species1
species2_cci	result of SingCellSignalR of species2
species_names_ref	dataframe, if you construct new homolog database and use new species homolog information in this analysis, you should provide the names corresponding relationship. it should be a dataframe contain at lease two column full_name and sy_name.
conserved_cell_types_df	dataframe, like conserved_cell_types_mm_zf <- data.frame("mm" = c("mmRods", "mmRod BC", "mmCones", "mmPericytes", "mmV/E cells", "mmRGC", "mmGABAergic AC", "mmRPE", "mmResting MG", "mmActivated MG", "mmMicroglia"), "zf" = c("zfRods", "zfCone BC", "zfCones", "zfPericytes", "zfV/E cells", "zfRGC", "zfGABAergic AC", "zfRPE", "zfResting MG", "zfActivated MG", "zfMicroglia"))
species_name1	two character to represent species1, like "mm". You should set this value from dataframe species_names_ref.rda
species_name2	two character to represent species2, like "zf". You should set this value from dataframe species_names_ref.rda
HOM_matrix	dataframe, default is NULL, if you used new created homolog database, you should put it here

Value

list for conserved result for species1 and species2

Examples

```
conserved_cell_types_mm_zf <- data.frame("mm" = c("mmRods", "mmRod BC", "mmCones", "mmPericytes", "mmV/E cells", "
conserved_result_mm_zf <- Identify_Conserved_CCI1(species1_cci=SingleCellSignalR_mouse_result,
                                        species2_cci=SingleCellSignalR_zebrafish_result,
                                    conserved_cell_types_df=conserved_cell_types_mm_zf,
                                                species_name1 = "mm",
                                                species_name2 = "zf")
```

Identify_Conserved_CCI2

Identify conserved cell-cell interaction from three species

Description

find the conserved cell-cell interaction from three species and caculate the conserved score of cell-cell interaction

Usage

```
Identify_Conserved_CCI2(
  species1_cci,
  species2_cci,
  species3_cci,
  species_names_ref = NULL,
  conserved_cell_types_df,
  species_name1 = "mm",
  species_name2 = "zf",
  species_name3 = "ch",
  HOM_matrix = NULL
)
```

Arguments

species1_cci	result of SingCellSignalR of species1
species2_cci	result of SingCellSignalR of species2
species3_cci	result of SingCellSignalR of species3
species_names_ref	dataframe, if you construct new homolog database and use new species homolog information in this analysis, you should provide the names corresponding relationship. it should be a dataframe contain at lease two column full_name and sy_name.
conserved_cell_types_df	dataframe, like conserved_cell_types_mm_zf_ch <- data.frame('mm' = c("mmResting MG","mmGABAergic AC", "mmRGC", "mmCones"), "zf" = c("zfResting MG", "zfGABAergic AC", "zfRGC", "zfCones"), 'ch' = c("chResting MG","chGABAergic AC", "chRGC", "chCones"))
species_name1	two character to represent species1, like "mm". You should set this value from dataframe species_names_ref.rda
species_name2	two character to represent species2, like "zf". You should set this value from dataframe species_names_ref.rda
species_name3	two character to represent species3, like "ch". You should set this value from dataframe species_names_ref.rda
HOM_matrix	dataframe, default is NULL, if you used new created homolog database, you should put it here

Value

list for conserved result for each species

identify_conserved_gene *Identify conserved gene*

Description

Identify conserved gene

Usage

```
identify_conserved_gene(
  OrthG,
  spc1_marker,
  spc2_marker,
  Species_name1,
  Species_name2
)
```

Arguments

OrthG	ortholog genes database
spc1_marker	vector, indicating the gene of species 1
spc2_marker	vector, indicating the gene of species 2
Species_name1	character, indicating the species names of Species1_Marker_table.
Species_name2	character, indicating the species names of Species2_Marker_table

Identify_Conserved_LR *Identify conserved ligand receptor interaction*

Description

Identify conserved cell-cell interactions in conserved cell types

Usage

```
Identify_Conserved_LR(
  OrthG,
  Species1_CCI,
  Species2_CCI,
  ConservedCellType = NULL,
  Species_name1,
  Species_name2
)
```

Arguments

OrthG	ortholog genes database
Species1_CCI	cell-cell interactions in species1. First column should be ligand, second column should be target, third column should be corresponding cell type of ligand, fourth column should be corresponding cel type of target
Species2_CCI	cell-cell interactions in species2. First column should be ligand, second column should be target, third column should be corresponding cell type of ligand, fourth column should be corresponding cel type of target
Species_name1	character, indicating the species names of Species1_GRN
Species_name2	character, indicating the species names of Species2_GRN

Examples

```
load(system.file("extdata", "cci_test.rda", package = "CACIMAR"))
Identify_ConservedCCI(OrthG_Hs_Mm,hs_cci_test,mm_cci_test,celltype,'hs','mm')
```

---

identify_conserved_marker

Identify conserved markers in conserved celltype

Description

Identify orthologs marker genes for two species based on orthologs database

Usage

```
identify_conserved_marker(
  OrthG,
  Species1_Marker_table,
  Species2_Marker_table,
  Species_name1,
  Species_name2,
  conserved_celltype_pair
)
```

Arguments

OrthG	ortholog genes database
Species1_Marker_table	data.frame of species 1, should contain 'gene' and 'Allcluster' columns.
Species2_Marker_table	data.frame of species 2, should contain 'gene' and 'Allcluster' columns.
Species_name1	character, indicating the species names of Species1_Marker_table.
Species_name2	character, indicating the species names of Species2_Marker_table
conserved_celltype_pair	character, indicating the conserved celltypes

Details

Identify orthologs marker genes for two species

`identify_conserved_pair` *Identify conserved pair*

Description

Identify conserved pair

Usage

```
identify_conserved_pair(conserved_score, quantile_threshold = 0.75)
```

Arguments

`conserved_score` conserved celltype/GRN score

`quantile_threshold` numeric, indicating

`identify_ct_ConservedNetworks` *Identify conserved cell type specific regulatory networks*

Description

Identify conserved cell type specific regulatory networks

Usage

```
identify_ct_ConservedNetworks(
  OrthG,
  Species1_GRN,
  Species2_GRN,
  Species_name1,
  Species_name2,
  network_regulation_num = NULL
)
```

Arguments

OrthG	ortholog genes database
Species1_GRN	gene regulatory network of species 1
Species2_GRN	gene regulatory network of species 2
Species_name1	character, indicating the species names of Species1_GRN
Species_name2	character, indicating the species names of Species2_GRN

Identify_Markers *Identify markers of each cluster*

Description

This function first identify marker genes in each cluster with Roc threshold > RocThr. Then, based on marker genes identified above, this function calculates the difference and power of marker genes in each cluster, and marker genes with Difference threshold > DiffThr will be retained. Next, gene with the largest power in which cluster will be the marker gene in this cluster. Eventually, make fisher test for power of each cluster, cluster with p.value < 0.05 will be retained as the final cluster for marker gene

Usage

```
Identify_Markers(
  Seurat_object,
  PowerCutoff = 0.4,
  DifferenceCutoff = 0,
  PvalueCutoff = 0.05
)
```

Arguments

Seurat_object	Seurat object, should contain cluster information
PowerCutoff	numeric, indicating the cutoff of gene power to refine marker genes
DifferenceCutoff	numeric, indicating the cutoff of difference in marker genes between clusters to refine marker genes
PvalueCutoff	numeric, indicating the p.value cutoff of chi-square test to refine marker genes

Value

Data frame of conserved markers

Examples

```
data("pbmc_small")
all.markers <- Identify_Markers(pbmc_small)
```

`identify_network_relationships` *Identify conserved node and edge in the networks Based on the Conservation analysis result from Identify_ConservedNetworks, this function further identify the conserved node and edge in the network(graph)*

Description

Identify conserved node and edge in the networks Based on the Conservation analysis result from Identify_ConservedNetworks, this function further identify the conserved node and edge in the network(graph)

Usage

```
identify_network_relationships(
  Species1_GRN,
  Species2_GRN,
  ConservedNetworkTable,
  Species1_group = "",
  Species2_group = ""
)
```

Arguments

`Species1_GRN`	gene regulatory network of species 1
`Species2_GRN`	gene regulatory network of species 2
`ConservedNetworkTable`	result from Identify_ConservedNetworks
`Species1_group`	character,indicating interested GRN group in species 1
`Species2_group`	character,indicating interested GRN group in species 2

`Make_ChordDiagram_data` *Make data fit for ChordDiagram*

Description

Make data fit for ChordDiagram

Usage

```
Make_ChordDiagram_data(
  cci_conserved_Weights_table,
  species_name1,
  species_name2
)
```

Arguments

cci_conserved_Weights_table	dataframe, result from function Caculate_cell_pair_cci_score
species_name1	two character to represent species1, like "mm"
species_name2	two character to represent species1, like "zf"

Value

dataframe, data for ChordDiagram

Examples

```
cci_data_mm_zf <- Make_ChordDiagram_data(cci_conserved_Weights_table = cci_conserved_Weights_table_mm_zf, specie
```

Make_ChordDiagram_data2

Make data fit for ChordDiagram for conserved cell types from three species

---

Description

Make data fit for ChordDiagram for conserved cell types from three species

Usage

```
Make_ChordDiagram_data2(
  cci_conserved_Weights_table,
  species_name1,
  species_name2,
  species_name3
)
```

Arguments

cci_conserved_Weights_table	dataframe, result from function Caculate_cell_pair_cci_score2
species_name1	two character to represent species1, like "mm"
species_name2	two character to represent species2, like "zf"
species_name3	two character to represent species3, like "ch"

Value

dataframe, data for ChordDiagram

Examples

```
# cci_conserved_Weights_table_sp1_sp2_sp3 is the result of function Caculate_cell_pair_cci_score2
 cci_data_sp1_sp2_sp3 <- Make_ChordDiagram_data2(cci_conserved_Weights_table = cci_conserved_Weights_table_sp1_s
 species_name1 = "mm",
 species_name2 = "zf",
 species_name3 = "ch")
```

---

make_pheatmap_LR_data *Make ligand-receptor data for pheatmap*

Description

prepared average expression data for pheatmap

Usage

```
make_pheatmap_LR_data(
  cci_conserved_results,
  seurat_object_sp1,
  seurat_object_sp2,
  seurat_object_sp3,
  species_name1 = "mm",
  species_name2 = "zf",
  species_name3 = "ch",
  avg_group = "cellname",
  cci_conserved_Weights,
  subset_quantile = 0.75,
  assay_set = "RNA"
)
```

Arguments

cci_conserved_results	dataframe, the result from function Identify_Conserved_CCI2
seurat_object_sp1	seurat object of species1
seurat_object_sp2	seurat object of species2
seurat_object_sp3	seurat object of species3
species_name1	character, two character the representation the species1, like "mm"
species_name2	character, two character the representation the species2, like "zf"

species_name3	character, two character the representation the species2, like "ch"
avg_group	character, which group used to calculate the average expression, like "celltype" in metadata
cci_conserved_Weights	dataframe, the result from function Caculate_cell_pair_cci_score
subset_quantile	numeric, the value to filter the conserved score of the cell-cell interaction, default is 0.75 of the consered score of cell-cell interaction
assay_set	which assay used for average expression, like "RNA", or "SCT"

Value

list, contains the dataframe for heatmap, and the average expression for each species

Examples

```
merge_avg_width_df <- make_pheatmap_LR_data(cci_conserved_results = conserved_result_mm_zf_ch,
seurat_object_sp1 = Mm_seurat,
seurat_object_sp2 = Zf_seurat,
seurat_object_sp3 = ch_seurat,
avg_group = "cellname",
species_name1 = "mm",
species_name2 = "zf",
species_name3 = "ch",
cci_conserved_Weights = conserved_cci_result$cci_conserved_Weights)
```

---

make_pheatmap_LR_data1

Make ligand-receptor data for pheatmap

Description

prepared average expression data for pheatmap

Usage

```
make_pheatmap_LR_data1(
  cci_conserved_results,
  seurat_object_sp1,
  seurat_object_sp2,
  species_name1 = "mm",
  species_name2 = "zf",
  species_name3 = "ch",
  avg_group = "cellname",
  cci_conserved_Weights,
  subset_quantile = 0.75,
  assay_set = "RNA"
)
```

Arguments

cci_conserved_results	dataframe, the result from function Identify_Conserved_CCI2
seurat_object_sp1	seurat object of species1
seurat_object_sp2	seurat object of species2
species_name1	character, two character the representation the species1, like "mm"
species_name2	character, two character the representation the species2, like "zf"
avg_group	character, which group used to calculate the average expression, like "celltype" in metadata
cci_conserved_Weights	dataframe, the result from function Caculate_cell_pair_cci_score
subset_quantile	numeric, the value to filter the conserved score of the cell-cell interaction, default is 0.75 of the consered score of cell-cell interaction
assay_set	which assay used for average expression, like "RNA", or "SCT"

Value

list, contains the dataframe for heatmap, and the average expression for each species

Examples

```
merge_avg_width_df <- make_pheatmap_LR_data1(cci_conserved_results = conserved_result_mm_zf,
seurat_object_sp1 = Mm_seurat,
seurat_object_sp2 = Zf_seurat,
avg_group = "cellname",
species_name1 = "mm",
species_name2 = "zf",
cci_conserved_Weights = conserved_cci_result$cci_conserved_Weights)
```

OrthG_Hs_Ch *Orthologs genes database for homo sapiens and zebrafish*

Description

Orthologs genes database for homo sapiens and zebrafish

Usage

```
OrthG_Hs_Ch
```

Format

An object of class `data.frame` with 16754 rows and 5 columns.

OrthG_Hs_Mm *Orthologs genes database for homo sapiens and mus musculus*

Description

Orthologs genes database for homo sapiens and mus musculus

Usage

```
OrthG_Hs_Mm
```

Format

An object of class `data.frame` with 16754 rows and 5 columns.

OrthG_Hs_Zf *Orthologs genes database for homo sapiens and zebrafish*

Description

Orthologs genes database for homo sapiens and zebrafish

Usage

```
OrthG_Hs_Zf
```

Format

An object of class `data.frame` with 12017 rows and 5 columns.

OrthG_Mm_Ch *Orthologs genes database for mus musculus and chicken*

Description

Orthologs genes database for mus musculus and chicken

Usage

```
OrthG_Mm_Ch
```

Format

An object of class `data.frame` with 62661 rows and 5 columns.

OrthG_Mm_Zf *Orthologs genes database for mus musculus and zebrafish*

Description

Orthologs genes database for mus musculus and zebrafish

Usage

```
OrthG_Mm_Zf
```

Format

An object of class `data.frame` with 65631 rows and 5 columns.

OrthG_Zf_Ch *Orthologs genes database for mus zebrafish and chicken*

Description

Orthologs genes database for mus zebrafish and chicken

Usage

```
OrthG_Zf_Ch
```

Format

An object of class `data.frame` with 38394 rows and 5 columns.

perform_CCI_analysis *Cell-cell interaction analysis with SingleCellSignalR algorithm*

Description

Perform cell-cell interaction with SingleCellSignalR algorithm contained in liana package.

Usage

```
perform_CCI_analysis(
  seurat_obj,
  expre_cell = 10,
  select_DB = "Consensus",
  target_organism,
  method = "sca",
  LRscore_threshold = 0.5,
  scale_score = TRUE
)
```

Arguments

seurat_obj	seurat object, a seurat object with cell types in active idents
expre_cell	numeric, filter genes expressed in more than expre_cell cells
select_DB	character, databased used for analysis, more option can run liana::show_resources()
target_organism	ncbi_taxid' or 'name' of the target organism. See 'show_homologene' for available organisms via OmnipathR's 'HomoloGene'
method	character, method use for cell-cell interaction analysis. By default, we set it to SingleCellSignalR
LRscore_threshold	numeric, the threshold the filter the cell-cell interactions, the larger the lesser interactions left
scale_score	logical, whether scale the score of the interaction,default is TRUE

Value

data frame of cell cell interaction analysis result

Examples

```
# Mouse analysis
SingleCellSignalR_mouse_result <- perform_CCI_analysis(seurat_obj=Mm_seurat, target_organism=10090)

SingleCellSignalR_zebrafish_result <- perform_CCI_analysis(seurat_obj=Zf_seurat, target_organism=7955)
```

---

pheatmap_LR1 *Create a pheatmap for average expression of ligand-receptor pairs*

Description

make a pheatmap for average expression of ligand-receptor pairs using geometric mean of corresponding cell type

Usage

```
pheatmap_LR1(
  width_df,
  filename = "pheatmap_LR.pdf",
  color_pheatmap_set = NULL,
  species_colors = NULL,
  species_name1 = "mm",
  species_name2 = "zf",
  annotation_names_col = TRUE,
  border_color = "white",
  scale = "none",
  cluster_rows = FALSE,
  cluster_cols = FALSE,
  legend = TRUE,
  legend_breaks = NA,
  legend_labels = NA,
  show_rownames = T,
  show_colnames = T,
  fontsize = 10,
  cellwidth = 16,
  cellheight = 12,
  width = 14,
  height = 14,
  ...
)
```

Arguments

width_df	width dataframe, result of function make_pheatmap_LR_data1
filename	name to save the pheatmap, default is "pheatmap_LR.pdf"
color_pheatmap_set	colors for the pheatmap, default is viridis::viridis(8)
species_colors	vector, colors vectors for the species
species_name1	two character, like "mm", for species1
species_name2	two character, like "zf", for species2
annotation_names_col	logical, whether show the column title name
border_color	character, for the color of the border, like "white"
scale	how to scale the data, one of "none", "row", "column"
cluster_rows	logical, whether cluster the row
cluster_cols	logical, whether cluster the column
legend	logical, whether show the legend
legend_breaks	break legend with this vector value
legend_labels	labels for the break legend

show_rownames	logical, whether show the rownames
show_colnames	logical, whether show the colnames
fontsize	numeric, size of the font
cellwidth	numeric, width of the cell
cellheight	numeric, heigth of the cell
width	numeric, width of the heatmap
height	numeric, height of the heatmap
...	other parameters in pheatmap

Value

pheatmap

Examples

```
merge_avg_width_df <- make_pheatmap_LR_data1(cci_conserved_results = conserved_result_mm_zf,
seurat_object_sp1 = Mm_seurat,
seurat_object_sp2 = Zf_seurat,
avg_group = "cellname",
species_name1 = "mm",
species_name2 = "zf",
cci_conserved_Weights = conserved_cci_result$cci_conserved_Weights)
pheatmap_LR_multi(width_df = merge_avg_width_df$merge_width_df, filename = "lr_avg_pheatmap.pdf")
```

---

pheatmap_LR_multi *Create a pheatmap for average expression of ligand-receptor pairs*

Description

make a pheatmap for average expression of ligand-receptor pairs using geometric mean of corresponding cell type

Usage

```
pheatmap_LR_multi(
  width_df,
  filename = "pheatmap_LR.pdf",
  color_pheatmap_set = NULL,
  species_colors = NULL,
  species_name1 = "mm",
  species_name2 = "zf",
  species_name3 = "ch",
  annotation_names_col = TRUE,
  border_color = "white",
  scale = "none",
  cluster_rows = FALSE,
```

```
  cluster_cols = FALSE,
  legend = TRUE,
  legend_breaks = NA,
  legend_labels = NA,
  show_rownames = T,
  show_colnames = T,
  fontsize = 10,
  cellwidth = 16,
  cellheight = 12,
  width = 14,
  height = 14,
  ...
)
```

Arguments

width_df	width dataframe, result of function make_pheatmap_LR_data
filename	name to save the pheatmap, default is "pheatmap_LR.pdf"
color_pheatmap_set	colors for the pheatmap, default is viridis::viridis(8)
species_colors	vector, colors vectors for the species
species_name1	two character, like "mm", for species1
species_name2	two character, like "zf", for species2
species_name3	two character, like "ch", for species3
annotation_names_col	logical, whether show the column title name
border_color	character, for the color of the border, like "white"
scale	how to scale the data, one of "none", "row", "column"
cluster_rows	logical, whether cluster the row
cluster_cols	logical, whether cluster the column
legend	logical, whether show the legend
legend_breaks	break legend with this vector value
legend_labels	labels for the break legend
show_rownames	logical, whether show the rownames
show_colnames	logical, whether show the colnames
fontsize	numeric, size of the font
cellwidth	numeric, width of the cell
cellheight	numeric, heigth of the cell
width	numeric, width of the heatmap
height	numeric, height of the heatmap
...	other parameters in pheatmap

Value

pheatmap

Examples

```
merge_avg_width_df <- make_pheatmap_LR_data(cci_conserved_results = conserved_result_mm_zf_ch,
seurat_object_sp1 = Mm_seurat,
seurat_object_sp2 = Zf_seurat,
seurat_object_sp3 = ch_seurat,
avg_group = "cellname",
species_name1 = "mm",
species_name2 = "zf",
species_name3 = "ch",
cci_conserved_Weights = conserved_cci_result$cci_conserved_Weights)
pheatmap_LR_multi(width_df = merge_avg_width_df$merge_width_df, filename = "lr_avg_pheatmap.pdf")
```

Plot_Celltype.Communication

Plot communication between cell types

Description

This function takes a dataframe containing information about communication between cell types, and plots a graph to visualize the communication pattern. The graph represents sender and receiver cell types as nodes, and the strength of communication as the width of edges connecting them.

Usage

```
Plot_Celltype.Communication(
  graph_df,
  sendercolumn,
  receivercolumn,
  widthcolumn,
  colors = NULL,
  useLabels = TRUE,
  nodeSize = 5,
  vertex.label.color,
  edge.label.color,
  vertex.label.cex,
  ...
)
```

Arguments

graph_df	The input dataframe containing communication data, see the example in tutorial.
sendercolumn	single column name, The name of the column containing sender cell types.
receivercolumn	single column name, The name of the column containing receiver cell types.

widthcolumn	single column name, The name of the column containing the width of edges (communication strength).
colors	Optional vector of colors for nodes and edges.
useLabels	Logical value indicating whether to display labels for nodes.
nodeSize	Size of the nodes.
vertex.label.color	Color of node labels.
vertex.label.cex	Size of node labels.
...	

Value

The plotted graph as an invisible object

Examples

```
plot_graph_network(long_df, sendercolumn = "celltype1", receivercolumn = "celltype2", widthcolumn = "score", useLa
```

---

plot_interaction_heatmap

Heapmap of cell-cell interaction

Description

Heapmap of cell-cell interaction

Usage

```
plot_interaction_heatmap(
  SingleCellSignalR_sp1_result,
  SingleCellSignalR_sp2_result,
  SingleCellSignalR_sp3_result,
  value_name = "scale_weight",
  used_col = c("source", "target", "LRscore_scale"),
  species_name1 = "mm",
  species_name2 = "zf",
  species_name3 = "ch",
  sp1_conserved_celltype,
  sp2_conserved_celltype,
  sp3_conserved_celltype,
  celltype_colors = NULL,
  brewer.pal_set = "Set3",
  species_colors = NULL,
  ph_colors = viridis::viridis(8),
```

```
  border_color = "white",
  scale = "none",
  cluster_rows = FALSE,
  cluster_cols = FALSE,
  legend = TRUE,
  show_rownames = F,
  show_colnames = F,
  fontsize = 8,
  cellwidth = 20,
  cellheight = 24,
  angle_col = "45",
  annotation_names_col = TRUE,
  annotation_names_row = TRUE,
  filename = "CCI_pheatmap.pdf",
  width = 18,
  height = 10,
  ...
)
```

Arguments

SingleCellSignalR_sp1_result	dataframe, result of the SingCellSignalR, must contain "source", "target", "LRscore_scale"
SingleCellSignalR_sp2_result	dataframe, result of the SingCellSignalR, must contain "source", "target", "LRscore_scale"
SingleCellSignalR_sp3_result	dataframe, result of the SingCellSignalR, must contain "source", "target", "LRscore_scale"
value_name	character, names of the sum weight, default is "scale_weight"
used_col	default is c("source", "target", "LRscore_scale"), they should be in the column of result of SingCellSignalR
species_name1	character, species_name must be only two characters, like "mm"
species_name2	character, species_name must be only two characters, like "zf"
species_name3	character, species_name must be only two characters, like "ch"
sp1_conserved_celltype	vector, a vector contained conserved cell type names in species1
sp2_conserved_celltype	vector, a vector contained conserved cell type names in species2
sp3_conserved_celltype	vector, a vector contained conserved cell type names in species3
celltype_colors	vector, colors vector with names. names of celltype_colors must be paste0(species_name, celltype_name), like "mmCones"
brewer.pal_set	character, pal in RColorBrewer
species_colors	vector, colors vector without names. colors for species
ph_colors	vector, colors for pheatmap

border_color	character, colors for border
scale	how to scale the data, one of "none", "row", "column"
cluster_rows	logical, whether cluster the row
cluster_cols	logical, whether cluster the column
legend	logical, whether show the legend
show_rownames	logical, whether show the rownames
show_colnames	logical, whether show the colnames
fontsize	numeric, size of the font
cellwidth	numeric, width of the cell
cellheight	numeric, heigth of the cell
angle_col	numeric, angle of the colnames
annotation_names_col	logical, whether show the name of column
annotation_names_row	logical, whether show the name of row
filename	the names used to save the heatmap
width	numeric, width of the heatmap
height	numeric, height of the heatmap
...	other parameters in pheatmap

Value

a list contain the data and the pheatmap object

Functions

- `plot_interaction_heatmap()`: heapmap to show interaction between conserved cell types

Examples

```
mm_conserved_celltype <- c("Activated MG", "Cones", "GABAergic AC","Microglia", "Pericytes", "Resting MG", "RGC",
zf_conserved_celltype <- c("Activated MG",  "Cones", "GABAergic AC", "Microglia", "Pericytes", "Resting MG", "RGC"
ch_conserved_celltype <- c("Activated MG", "Cones", "GABAergic AC", "Resting MG", "RGC", "Cone BC", "Rods")
plot_interaction_heatmap(SingleCellSignalR_sp1_result = SingleCellSignalR_mouse_result,
                      SingleCellSignalR_sp2_result = SingleCellSignalR_zebrafish_result,
                         SingleCellSignalR_sp3_result = SingleCellSignalR_chick_result,
                         species_name1 = "mm",
                         species_name2 = "zf",
                         species_name3 = "ch",
                         sp1_conserved_celltype = mm_conserved_celltype,
                         sp2_conserved_celltype = zf_conserved_celltype,
                         sp3_conserved_celltype = ch_conserved_celltype)
```

`Plot_MarkersHeatmap` *Plot Markers in each cell type*

Description

This function integrate R package pheatmap to plot markers in each cell type

Usage

```
Plot_MarkersHeatmap(
  ConservedMarker,
  start_col = 2,
  module_colors = NA,
  heatmap_colors = NA,
  cluster_rows = F,
  cluster_cols = F,
  show_rownames = F,
  show_colnames = F,
  cellwidth = 6,
  cellheight = 2,
  legend = F,
  annotation_legend = F,
  annotation_names_row = F,
  ...
)
```

Arguments

`ConservedMarker`	Markers table
`start_col`	numeric, indicating the start column of marker power in each cell type
`module_colors`	vector, indicating colors of modules (annotation_colors)
`heatmap_colors`	vector, indicating colors used in heatmap
`cluster_rows`	boolean values determining if rows should be clustered or hclust object
`cluster_cols`	boolean values determining if columns should be clustered or hclust object
`show_rownames`	boolean specifying if column names are be shown
`show_colnames`	boolean specifying if column names are be shown
`cellwidth`	individual cell width in points. If left as NA, then the values depend on the size of plotting window
`cellheight`	individual cell height in points. If left as NA, then the values depend on the size of plotting window
`legend`	logicalal to determine if legend should be drawn or not
`annotation_legend`	boolean value showing if the legend for annotation tracks should be drawn

annotation_names_row
: boolean value showing if the names for row annotation tracks should be drawn

... : parameter in pheatmap

Value

pheatmap object

Examples

```
data("pbmc_small")
all.markers <- Identify_Markers(pbmc_small)
all.markers <- Format_Markers_Frac(all.markers)
Plot_MarkersHeatmap(all.markers[,c(2,6,7,8)])
```

Plot_phylogenetic_tree

Plot phylogenetic tree

Description

Build a phylogenetic tree to identify the conservation of the cell types across species

Usage

```
Plot_phylogenetic_tree(
  SCT_matrix,
  tree_method = "hierarchical clustering",
  hcluster.method = "average",
  species.vector,
  layout.tree = "rectangular",
  tree_size = 1.4,
  xlim_tree = 0.65,
  fontface_for_tiplab = TRUE,
  set_fontface_for_tiplab = c("bold.italic", "italic", "plain"),
  conserved_hm_celltype = NULL,
  bar_width = 0.025,
  Show_Pairwise_distances = FALSE,
  boot_times = 100,
  offset = 0.04,
  offset2 = 0.022,
  offset_tiplab = 0.07,
  fontface_tiplab_with_colors = "bold",
  tiplab.size = 7,
  tiplab_cols = NULL,
  tippoint.shape = 21,
  tippoint.shape.size = 0,
```

```
  geom_nodepoint = 5,
  show_branch.length = FALSE,
  round_x = 2,
  fill_brandlength = "lightgreen",
  size_brandlength = 6,
  annotation_colors_df = NULL,
  brewer_pal_used = "Set3",
  col.value = c(rgb(102/255, 46/255, 115/255), rgb(31/255, 153/255, 139/255)),
  colors_labels = NULL,
  show_colnames = FALSE,
  colnames_angle = 0,
  colnames_position = "top",
  font.size_colnames = 5,
  colnames_offset_y = 0.8,
  custom_column_labels = c("Species", "Celltypes"),
  bootstrap_value_size = 0,
  bootstrap_value_col = "black",
  legend.text_size = 18,
  width = 13,
  height = 22,
  plot.margin = margin(10, 10, 10, -10),
  legend.position = "top"
)
```

Arguments

SCT_matrix	matrix, contain the conservation scores of cell types across species
tree_method	character, method used for constructing phylogenetic tree, it can be "hierarchical clustering", or "classic Neighbor-Joining"
hcluster.method	character, method used for clustering phylogenetic tree, it can be one of "average", "single", "complete", "mcquitty", "median", "centroid"
species.vector	vector, length is equal to nrow(SCT_matrix), used for annotation bar
layout.tree	character, method used for the layout of the tree. it can be one of 'rectangular', 'dendrogram', 'fan', 'circular'
tree_size	numeric, size of the tree's branch
xlim_tree	numeric, x limit for the picture, used for the width of the picture
fontface_for_tiplab	logical, whether used fontface for tip label. If it is TRUE, then the tiplab_cols for tip label won't work
set_fontface_for_tiplab	if fontface_for_tiplab is TRUE, the parameter will work. By default, it is set with c("bold.italic", "italic", "plain")
conserved_hm_celltype	vector, contain the conserved cell types in heatmap, or celltypes defined by yourself

`bar_width` integer, width of annotation bar

`Show_Pairwise_distances`
logical, whether save the plot of pairwise distances between input data and constructed tree or not

`boot_times` integer, the times for bootstrap test, usually 100, or you can set larger 1000

`offset` numeric, the distance of celltype annotation bar from tip

`offset2` numeric, the distance of species annotation bar from tip

`offset_tiplab` numeric, the distance of labels from tip

`fontface_tiplab_with_colors`
charcter, colors for tip label. if fontface_for_tiplab is FALSE, then can set one fontface for all tip labels, like "bold", "italic", "plain"

`tiplab.size` numeric, the size of tip labels

`tiplab_cols` vector, colors for tip group

`tippoint.shape` numeric, the shape used for tip point, such as 21, 17, 22, ...

`tippoint.shape.size`
numeric, the size of the tip point shape

`geom_nodepoint` integer, the size of the node point

`show_branch.length`
logical, whether show the branch length, default is TRUE

`round_x` numeric, the decimal places contained for round() function for branch length number

`fill_brandlength`
character, colors for branch length

`size_brandlength`
numeric, the size for the branch length label

`annotation_colors_df`
dataframe, must contain two column, named "colors" and "celltype", default is null

`brewer_pal_used`
character, the brewer_pal in RColorBrewer, like "Set1", "Set3", "Paired". It set colors for tip label. It works when fontface_for_tiplab is FALSE

`col.value` chacterize vector, colors used for species bar annotation, must be the same length of the number of species

`colors_labels` chacterize vector, labels for the legend, the length of colors_breaks

`show_colnames` logical, whether show the colnames of annotation bar

`colnames_angle` integer, the angle of the colnames of annotation bar

`colnames_position`
character, the position of the colnames of annotation bar, can be top or bottom

`font.size_colnames`
integer, the size of the colnames of annotation bar

`colnames_offset_y`
numeric, the distance of the colnames from the annotation bar

custom_column_labels
: character, edit the colnames of annotation bar by yourself

bootstrap_value_size
: numeric, the size of the bootstrap values showed in the tree. If you don't want to showed the bootstrap values in the tree, you can set the parameter to 0

bootstrap_value_col
: character, the color of the bootstrap values showed in the tree

legend.text_size
: integer, the size of the legend

width
: numeric, the width of the plot

height
: numeric, the height of the plot

plot.margin
: the plot margin, it should be something like margin(10, 10, 10, -10), the four number represent top, right, bottom, left, respectively

legend.position
: character, one of "right", "left", "top", "bottom", the position of the legend

fontface_tiplab
: charcter, the fontface used for tip label

Value

save plot in present file directory

Examples

```
library(CACIMAR, lib.loc = "/usr/local/lib/R/site-library")
load(system.file("extdata", "zf_mm_markers.rda", package = "CACIMAR"))
expression <- CACIMAR::Identify_ConservedCellTypes(OrthG_Mm_Zf,Zf_marker,Mm_marker,'zf','mm')
SCT_matrix <- expression[[2]]
# get the species names
species.vector <- substr(rownames(SCT_matrix), 1, 2)
# get the conserved celltypes in heatmap
SNT_h <- SCT_matrix[grep('mm',rownames(SCT_matrix)),as.numeric(grep('zf',colnames(SCT_matrix)))]
conserved_hm_celltypes <- get_conserved_hm_celltypes(SNT_h)
Plot_phylogenetic_tree(SCT_matrix = SCT_matrix,
                       species.vector = species.vector,
                       conserved_hm_celltype = conserved_hm_celltypes)
```

Plot_Species_CellType_Tree

Plot Phylogenetic Tree of Different Species Cell Types

Description

Plot a tree of different species cell types using the conserved score of cell types

Usage

```
Plot_Species_CellType_Tree(
  dist_matrix,
  hcluster.method = c("average", "single", "complete"),
  species.vector,
  layout.tree = c("rectangular", "dendrogram", "fan", "circular"),
  offset = 0.01,
  tiplab.size = 5,
  tippoint.shape = c(22, 21),
  tippoint.shape.size = 4,
  geom_nodepoint = 0,
  col.value = c(rgb(102/255, 46/255, 115/255), rgb(31/255, 153/255, 139/255))
)
```

Arguments

dist_matrix	Distance matrix
hcluster.method	chariter, Hierarchical clustering method, possible values are "average" (default), "single", "complete"
species.vector	vector, Species vector
layout.tree	Tree layout ("rectangular", "dendrogram", "fan", "circular"), default is "rectangular"
offset	Tiplab offset, default is 0.01
tiplab.size	Tiplab size, default is 5
tippoint.shape	Tippoint shape, default is 22
tippoint.shape.size	Tippoint size, default is 4
geom_nodepoint	Nodepoint size, default is 0
col.value	Color vector for species, default is c(rgb(102/255,46/255,115/255), rgb(31/255,153/255,139/255))

Value

a list, which contain the Tree data and the Tree plot

Examples

```
load(system.file("extdata", "zf_mm_markers.rda", package = "CACIMAR"))
expression <- Identify_ConservedCellTypes(OrthG_Mm_Zf,Zf_marker,Mm_marker,'zf','mm')
dist_matrix <- expression[[2]]
species.vector <- substr(rownames(dist_matrix), 1, 2)
Species_CellType_Tree <- Plot_Species_CellType_Tree(dist_matrix = dist_matrix, species.vector = species.vector, h
Species_CellType_Tree$Plot
```

Index