

Package ‘VennDetail’

May 9, 2026

Type Package

Title Comprehensive Visualization and Analysis of Multi-Set Intersections

Version 1.29.0

Date 2025-07-01

Description A comprehensive package for visualizing multi-set intersections and extracting detailed subset information. VennDetail generates high-resolution visualizations including traditional Venn diagrams, Venn-pie plots, and UpSet-style plots. It provides functions to extract and combine subset details with user datasets in various formats. The package is particularly useful for bioinformatics applications but can be used for any multi-set analysis.

License GPL-2

Encoding UTF-8

LazyData true

Depends R (>= 4.0.0)

Imports dplyr, DT, ggplot2, grid, gridExtra, magrittr, methods, patchwork, plotly, purrr, rlang, shiny, stats, tibble, tidyr, htmlwidgets, utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0), markdown, RColorBrewer, rstudioapi

VignetteBuilder knitr

URL <https://github.com/guokai8/VennDetail>

BugReports <https://github.com/guokai8/VennDetail/issues>

Roxygen list(roclets = c("`namespace", "`rd"))

RoxygenNote 7.3.3

biocViews DataRepresentation, GraphAndNetwork, Visualization, Software

git_url <https://git.bioconductor.org/packages/VennDetail>

git_branch devel

git_last_commit 352913d

git_last_commit_date 2026-04-28

Repository Bioconductor 3.24

Date/Publication 2026-05-08

Author Kai Guo [aut, cre],
 Brett McGregor [aut],
 James Porter [aut],
 Junguk Hur [aut]

Maintainer Kai Guo <guokai8@gmail.com>

Contents

| | |
|--------------------------------------|----|
| VennDetail-package | 3 |
| .add_colnames | 4 |
| .findOverlap | 5 |
| .make.table | 5 |
| .vennDiagram | 6 |
| as.data.frame.Venn | 7 |
| compareVenn | 7 |
| create_interactive_vennpie | 8 |
| detail | 9 |
| dim.Venn | 10 |
| dplot | 10 |
| getFeature | 11 |
| getSet | 13 |
| head.Venn | 14 |
| loadVenn | 15 |
| make.subset | 15 |
| makeContent.upset_grob | 16 |
| make_subset | 16 |
| make_truth_table | 17 |
| merge.Venn | 18 |
| names.Venn | 18 |
| newVenn | 19 |
| plot.Venn | 20 |
| print.upset_grob | 25 |
| result | 26 |
| rowjoin | 27 |
| saveVenn | 28 |
| setcolor | 28 |
| show Venn | 29 |
| summary.Venn | 30 |
| T2DM | 30 |
| upsetPlot | 31 |
| upsetPlot,Venn-method | 34 |
| upset_plot | 37 |
| Venn-class | 40 |
| vennApp | 41 |
| venndetail | 42 |
| vennDiagram | 43 |
| vennDiagram,Venn-method | 44 |
| vennEnrichment | 46 |
| vennpie | 47 |
| vennpie,Venn-method | 48 |
| vennStats | 49 |

| | |
|---------------------------------|----|
| vennStats,Venn-method | 50 |
| [.Venn | 51 |
| \$.Venn | 52 |

| | |
|--------------|-----------|
| Index | 53 |
|--------------|-----------|

| | |
|--------------------|--|
| VennDetail-package | <i>VennDetail: Comprehensive Visualization and Analysis of Multi-Set Intersections</i> |
|--------------------|--|

Description

A comprehensive package for visualizing multi-set intersections and extracting detailed subset information. VennDetail generates high-resolution visualizations including traditional Venn diagrams, Venn-pie plots, and UpSet-style plots. It provides functions to extract and combine subset details with user datasets in various formats.

Details

The VennDetail package offers several powerful visualization and analysis tools:

Visualization methods:

- Traditional Venn diagram (for 2-5 sets)
- VennPie visualization (useful for more than 5 sets)
- UpSet plot (matrix-based visualization)
- Bar plot (simple visualization of subset sizes)

Key features:

- Extraction of elements in any subset combination
- Combining subset information with user-supplied data frames
- Statistical analysis of set intersections
- Enrichment analysis for set members
- Interactive visualizations
- High-resolution figure export
- Shiny app for interactive exploration

Getting Started

To create a Venn object for analysis:

```
““ # Create sample datasets A <- sample(1:100, 40, replace = FALSE) B <- sample(1:100, 60,
replace = FALSE) C <- sample(1:100, 40, replace = FALSE)
# Create a Venn object res <- venndetail(list(A = A, B = B, C = C)) ““
```

Visualization

```
““ # Traditional Venn diagram vennDiagram(res)
# VennPie visualization vennpie(res)
# UpSet plot upsetPlot(res)
# Bar plot dplot(res, order = TRUE)
# Generic plot function with type selection plot(res, type = "venn") ““
```

Data Extraction

```
““ # Extract elements shared by all sets shared <- getSet(res, "Shared")
# Extract elements unique to set A unique_to_A <- getSet(res, "A") ““
```

Statistical Analysis

```
““ # Test for significance of overlaps stats <- vennStats(res) ““
```

Author(s)

Kai Guo, Brett McGregor

See Also

Useful links:

- <https://github.com/guokai8/VennDetail>
- Report bugs at <https://github.com/guokai8/VennDetail/issues>

.add_colnames *Give first colname as RowNxyz*

Description

Give first colname as RowNxyz

Usage

```
.add_colnames(x)
```

Arguments

x data frame

Value

return data frame with the first colnames change to "RowNxyz"

.findOverlap *Find overlap between sets in a list*

Description

Find overlap between sets in a list

Usage

```
.findOverlap(setlist, xlim = c(0, 1), ylim = c(0, 1))
```

Arguments

setlist list of character vectors.
xlim vector with 2 numbers, x axis limits for the venn diagram.
ylim vector with 2 numbers, y axis limits for the venn diagram.

Value

list with 2 items: a data.frame with information about the groups (sizes, coordinates, etc.), and a data.frame containing the x and y coordinates for the venn diagram

.make.table *make table for venndetail modified from make.truth.table (VennDiagram)*

Description

make table for venndetail modified from make.truth.table (VennDiagram)

Usage

```
.make.table(x)
```

Arguments

x A list with input groups

Value

A data frame with logical vector columns and $2^{\text{length}(x)}-1$ rows.

Author(s)

Kai Guo

 .vennDiagram

Create a standalone Venn diagram

Description

This function creates a standalone Venn diagram for 2-5 sets

Usage

```
.vennDiagram(
  setlist,
  title = NULL,
  colors = NULL,
  alpha = 0.4,
  showNumbers = TRUE,
  numberSize = 4,
  numberColor = "black",
  labelSize = 4,
  labelColor = "black",
  borderCol = FALSE,
  fillCol = TRUE,
  fixedCoords = TRUE,
  xlim = c(0, 1),
  ylim = c(0, 1)
)
```

Arguments

| | |
|-------------|---|
| setlist | A named list of character vectors, each representing a set |
| title | Optional title for the plot |
| colors | Vector of colors for the sets. If NULL, default colors will be used |
| alpha | Transparency level for the circles (0-1) |
| showNumbers | Logical, whether to show counts and percentages in each region |
| numberSize | Size of the count labels |
| numberColor | Color of the count labels |
| labelSize | Size of the set labels |
| labelColor | Color of the set labels |
| borderCol | Logical, whether to color the borders of circles |
| fillCol | Logical, whether to fill circles with colors |
| fixedCoords | Logical, whether to use fixed coordinates |
| xlim | Vector with 2 numbers, x axis limits for the venn diagram |
| ylim | Vector with 2 numbers, y axis limits for the venn diagram |

Value

A ggplot object

Author(s)

Kai Guo

as.data.frame.Venn *Utility functions for VennDetail package*

Description

Internal utility functions for the VennDetail package
Converts a Venn object to a data frame for easier manipulation

Usage

```
## S3 method for class 'Venn'  
as.data.frame(x, ...)
```

Arguments

x A Venn object
... Additional arguments (not used)

Value

A data frame with subset information

Author(s)

Kai Guo

Examples

```
# Create a Venn object  
A <- sample(1:100, 40, replace = FALSE)  
B <- sample(1:100, 60, replace = FALSE)  
res <- vennDetail(list(A = A, B = B))  
  
# Convert to data frame  
df <- as.data.frame(res)  
head(df)
```

compareVenn *Compare two Venn objects*

Description

Compares two Venn objects and returns a list of differences

Usage

```
compareVenn(x, y, what = c("groups", "subsets", "all"))
```

Arguments

| | |
|------|--|
| x | First Venn object |
| y | Second Venn object |
| what | What to compare: "groups" (default), "subsets", or "all" |

Value

A list with differences between the objects

Author(s)

Kai Guo

Examples

```
# Create two Venn objects
A1 <- sample(1:100, 40, replace = FALSE)
B1 <- sample(1:100, 60, replace = FALSE)
res1 <- venndetail(list(A = A1, B = B1))

A2 <- sample(1:100, 45, replace = FALSE)
B2 <- sample(1:100, 55, replace = FALSE)
res2 <- venndetail(list(A = A2, B = B2))

# Compare the objects
compareVenn(res1, res2)
```

create_interactive_vennpie

Create an interactive vennpie chart with plotly

Description

Creates a simple pie chart visualization for interactive exploration of set intersections

Usage

```
create_interactive_vennpie(  
  object,  
  subset = NULL,  
  any = NULL,  
  color = NULL,  
  revcolor = "lightgrey",  
  title = NULL  
)
```

Arguments

| | |
|----------|--|
| object | A Venn object |
| subset | Character vector of subset names to highlight |
| any | Highlight subsets shared by exactly this many sets |
| color | Optional vector of colors for the subsets |
| revcolor | Color for non-highlighted subsets |
| title | Optional plot title |

Value

A plotly object

Author(s)

Kai Guo

detail

Get subset details from a Venn object

Description

Returns a named numeric vector with counts for each subset

The objective of this function is to summarize the overlaps across groups identified by `venndetail` without creating diagram.

Usage

```
detail(object)
```

```
## S4 method for signature 'Venn'  
detail(object)
```

Arguments

| | |
|--------|---------------|
| object | A Venn object |
|--------|---------------|

Value

A named numeric vector with counts for each subset

Author(s)

Kai Guo

Examples

```

A <- sample(1:100, 40, replace = FALSE)
B <- sample(1:100, 60, replace = FALSE)
C <- sample(1:100, 40, replace = FALSE)
res <- vennndetail(list(A = A, B = B, C = C))
detail(res)
A <- sample(1:100, 40, replace = FALSE)
B <- sample(1:100, 60, replace = FALSE)
C <- sample(1:100, 40, replace = FALSE)
res <- vennndetail(list(A = A, B = B, C = C))
detail(res)

```

| | |
|----------|--|
| dim.Venn | <i>Get dimensions of a Venn object</i> |
|----------|--|

Description

Returns the dimensions of the result slot in a Venn object.

Usage

```

## S3 method for class 'Venn'
dim(x)

```

Arguments

x Venn object

Value

Integer vector of length 2 (rows, columns)

| | |
|-------|---|
| dplot | <i>Create a bar plot of subset counts</i> |
|-------|---|

Description

Creates a bar plot showing counts for each subset

Usage

```

dplot(
  object,
  order = FALSE,
  textsize = 5,
  color = NULL,
  theme = ggplot2::theme_light(),
  title = NULL,
  xlabel = NULL,
  ylabel = NULL
)

```

```
)

## S4 method for signature 'Venn'
dplot(object, order = FALSE, textsize = 5)
```

Arguments

| | |
|----------|--|
| object | Venn object |
| order | Boolean indicating whether to sort the bar (default: FALSE). |
| textsize | Numeric vector giving the text size above the bar. |
| color | Optional vector of colors for the bars |
| theme | The ggplot2 theme to use. Default: theme_light |
| title | Optional plot title |
| xlabel | Optional x-axis label |
| ylabel | Optional y-axis label |

Value

A ggplot2 object

Author(s)

Kai Guo

Examples

```
A <- sample(1:100, 40, replace = FALSE)
B <- sample(1:100, 60, replace = FALSE)
C <- sample(1:100, 40, replace = FALSE)
res <- vennetail(list(A = A, B = B, C = C))
dplot(res, order = TRUE, textsize = 3)
A <- sample(1:100, 40, replace = FALSE)
B <- sample(1:100, 60, replace = FALSE)
C <- sample(1:100, 40, replace = FALSE)
res <- vennetail(list(A = A, B = B, C = C))
dplot(res, order = TRUE, textsize = 3)
```

getFeature

Extract feature data for specific subsets

Description

Combines subset information with user-supplied data frames

GetFeature allows users to extract subsets from venn object into a table format along with accompanying information from the data frames provided in the rlist argument

Usage

```

getFeature(
  object,
  subset,
  rlist,
  userowname = TRUE,
  gind = NULL,
  sep = "_",
  wide = FALSE
)

## S4 method for signature 'Venn'
getFeature(
  object,
  subset,
  rlist,
  userowname = TRUE,
  gind = NULL,
  sep = "_",
  wide = FALSE
)

```

Arguments

| | |
|------------|---|
| object | Venn object |
| subset | Character vector giving the names of the user-defined subset to extract |
| rlist | List of user-supplied data frames to combine with <code>venndetail</code> result |
| userowname | Boolean indicating whether to use row names to join data frames or not (default: TRUE) |
| gind | Column name or index of each user-supplied data.frame to use to join data frames(valid only when <code>userowname=FALSE</code>) |
| sep | Character string used to separate the terms when concatenating group names into new separation character for new column names in the resulting data frame |
| wide | Boolean indicating whether to use wide format(default:FALSE) |

Value

A data.frame combining subset information with user data
 data.frame with subsets information and details from the user supplied data frame

Author(s)

Kai Guo

Examples

```

A <- sample(1:100, 40, replace = FALSE)
B <- sample(1:100, 60, replace = FALSE)
C <- sample(1:100, 40, replace = FALSE)
dA <- data.frame(A = A, "FC" = rnorm(40))
dB <- data.frame(B = B, "FC" = rnorm(60))

```

```

dC <- data.frame(C = C, "FC" = rnorm(40))
res <- vennDetail(list(A = A, B = B, C = C))
features <- getFeature(res, subset = "Shared",
                      rlist = list(dA, dB, dC),
                      userowname = FALSE,
                      gind = rep(1, 3))
A <- sample(1:100, 40, replace = FALSE)
B <- sample(1:100, 60, replace = FALSE)
C <- sample(1:100, 40, replace = FALSE)
dA <- data.frame(A = A, "FC" = rnorm(40))
dB <- data.frame(B = B, "FC" = rnorm(60))
dC <- data.frame(C = C, "FC" = rnorm(40))
res <- vennDetail(list(A = A, B = B, C = C))
rhs <- getFeature(res, subset = "Shared", rlist = list(dA, dB, dC),
                 userowname= FALSE, gind = rep(1, 3))

```

getSet

Extract specific subsets from a Venn object

Description

Extracts elements from specified subsets

getSet function provides a way to extract subsets from vennDetail object

Usage

```
getSet(object, subset = NULL, min = 0, wide = FALSE)
```

```
## S4 method for signature 'Venn'
```

```
getSet(object, subset = NULL, min = 0, wide = FALSE)
```

Arguments

| | |
|--------|---|
| object | Venn object |
| subset | Character vector giving the subset names |
| min | The minimum number of input groups that a subset must belong to e.g. min = 2 will only report those subsets with elements shared by 2 or more input groups. |
| wide | Boolean indicating return wide format (default: FALSE). |

Value

A data.frame with elements from the specified subsets

Specific subset information

Author(s)

Kai Guo

Examples

```

A <- sample(1:100, 40, replace = FALSE)
B <- sample(1:100, 60, replace = FALSE)
C <- sample(1:100, 40, replace = FALSE)
res <- vennDetail(list(A = A, B = B, C = C))
# Get elements unique to set A
unique_to_A <- getSet(res, "A")
# Get elements shared by all sets
shared <- getSet(res, "Shared")
A <- sample(1:100, 40, replace = FALSE)
B <- sample(1:100, 60, replace = FALSE)
C <- sample(1:100, 40, replace = FALSE)
res <- vennDetail(list(A = A, B = B, C = C))
getSet(res, "A")

```

head.Venn

Return the first or last parts of a Venn object

Description

Methods to extract the first or last parts of a Venn object's result slot.

Usage

```
## S3 method for class 'Venn'
head(x, n = 6L, ...)
```

```
## S3 method for class 'Venn'
tail(x, n = 6L, ...)
```

Arguments

| | |
|-----|--|
| x | Venn object |
| n | number of rows to display |
| ... | other arguments ignored (for compatibility with generic) |

Value

A data.frame with the first n rows

A data.frame with the last n rows

| | |
|----------|---------------------------------------|
| loadVenn | <i>Load a Venn object from a file</i> |
|----------|---------------------------------------|

Description

Loads a Venn object from an RDS file

Usage

```
loadVenn(file)
```

Arguments

| | |
|------|------------------------|
| file | File name to load from |
|------|------------------------|

Value

A Venn object

Author(s)

Kai Guo

Examples

```
## Not run:  
# Load a saved Venn object  
res <- loadVenn("my_venn.rds")  
  
# Plot the loaded object  
plot(res)  
  
## End(Not run)
```

| | |
|-------------|---|
| make.subset | <i>Get subset from list of input groups</i> |
|-------------|---|

Description

Get subset from list of input groups

Usage

```
make.subset(x, sep = "_")
```

Arguments

| | |
|-----|--|
| x | A list with input groups |
| sep | symbol character used when concatenating group names into subset names |

Value

A list of subsets. The names on the list are the subset names and the list elements are the subset details.

Author(s)

Kai Guo

Examples

```
A <- sample(1:100, 40, replace = FALSE)
B <- sample(1:100, 60, replace = FALSE)
C <- sample(1:100, 40, replace = FALSE)
x <- list(A = A, B = B, C = C)
out <- make_subset(x)
```

makeContent.upset_grob

Draw method for upset_grob objects

Description

Internal method to draw upset_grob using grid viewports

Usage

```
## S3 method for class 'upset_grob'
makeContent(x)
```

Arguments

x An upset_grob object

Value

Draws the upset plot

make_subset

Create subsets from a list of sets

Description

Identifies all possible intersections between sets and returns a list of subsets

Usage

```
make_subset(x, sep = "_")
```

Arguments

x A list of vectors
sep Character used to separate set names in subset labels

Value

A named list where each element contains the unique items in that subset

Author(s)

Kai Guo

Examples

```
A <- sample(1:100, 40, replace = FALSE)
B <- sample(1:100, 60, replace = FALSE)
C <- sample(1:100, 40, replace = FALSE)
x <- list(A = A, B = B, C = C)
subsets <- make_subset(x)
lengths(subsets) # Number of elements in each subset
```

make_truth_table *Create a truth table for set combinations*

Description

Creates a logical matrix representing all possible combinations of sets

Usage

```
make_truth_table(x)
```

Arguments

x A list of vectors

Value

A data frame with logical columns for each set and rows for each combination

Author(s)

Kai Guo

| | |
|------------|---|
| merge.Venn | <i>Merge two or more venndetail obejcts</i> |
|------------|---|

Description

Merge will combine multiple venn diagrams to allow comparison between multiple groups

Usage

```
## S3 method for class 'Venn'
merge(x, y, ignore.case = FALSE, useupper = TRUE, plot = FALSE, ...)
```

Arguments

| | |
|-------------|--|
| x | Venn object |
| y | Venn object |
| ignore.case | Boolean indicating whether to ignore case of group names (default: FALSE) |
| useupper | Boolean indicating whether to use uppercases for group names (default: TRUE) |
| plot | Boolean indicating whether to plot figure or not (default: FALSE) |
| ... | arguments for venndetail |

Value

venn object

Examples

```
A <- sample(1:100, 40, replace = FALSE)
B <- sample(1:100, 60, replace = FALSE)
C <- sample(1:100, 40, replace = FALSE)
res1 <- venndetail(list(A = A, B = B))
res2 <- venndetail(list(A = A, C = C))
res <- merge(res1, res2)
```

| | |
|------------|--|
| names.Venn | <i>Extract subset names from a Venn object</i> |
|------------|--|

Description

Returns the names of all subsets in a Venn object

Usage

```
## S3 method for class 'Venn'
names(x)
```

Arguments

| | |
|---|---------------|
| x | A Venn object |
|---|---------------|

Value

A character vector of subset names

Author(s)

Kai Guo

Examples

```
# Create a Venn object
A <- sample(1:100, 40, replace = FALSE)
B <- sample(1:100, 60, replace = FALSE)
res <- venndetail(list(A = A, B = B))

# Get subset names
names(res)
```

newVenn

Create a new Venn object

Description

Constructor function for creating Venn objects with validation

Usage

```
newVenn(
  input,
  raw,
  sep = "_",
  GroupNames,
  result,
  detail,
  wide,
  metadata = list()
)
```

Arguments

| | |
|------------|--------------------------------|
| input | A list of input sets |
| raw | A named vector with counts |
| sep | The separator character |
| GroupNames | Names of the input groups |
| result | The result data.frame |
| detail | The detail vector |
| wide | The wide-format data.frame |
| metadata | Additional metadata (optional) |

Value

A new Venn object

Author(s)

Kai Guo

Examples

```
## Not run:
A <- sample(1:100, 40, replace = FALSE)
B <- sample(1:100, 60, replace = FALSE)
raw <- c(A = 40, B = 60)
groups <- c("A", "B")
# Create a new Venn object manually (normally done by venndetail function)
venn_obj <- newVenn(input = list(A = A, B = B), raw = raw,
                   GroupNames = groups, ...)

## End(Not run)
```

plot.Venn

Plot a Venn object

Description

Unified plotting function for Venn objects that supports multiple visualization types

Usage

```
## S3 method for class 'Venn'
plot(
  x,
  type = "venn",
  title = NULL,
  interactive = FALSE,
  filename = NULL,
  width = 8,
  height = 6,
  dpi = 300,
  fill = NULL,
  alpha = 0.5,
  labels = TRUE,
  counts = TRUE,
  showNumbers = TRUE,
  numberSize = 4,
  numberColor = "black",
  labelSize = 4,
  labelColor = "black",
  borderCol = FALSE,
  fillCol = TRUE,
  fixedCoords = TRUE,
  xlim = c(0, 1),
```

```
ylim = c(0, 1),
show_percentages = TRUE,
show_unique_only = FALSE,
scaled = FALSE,
subset = NULL,
top = 31,
min = 0,
color = NULL,
revcolor = "lightgrey",
any = NULL,
show.number = TRUE,
show.x = TRUE,
sep = "_",
log = FALSE,
base = NULL,
percentage = FALSE,
nintersects = 40,
min_size = 1,
sets_bar_color = NULL,
main_bar_color = "steelblue",
point_size = 3,
line_size = 1,
show_numbers = TRUE,
sort_intersections_by = "freq",
sort_sets_by = "size",
sort_sets_decreasing = TRUE,
custom_sets_order = NULL,
sort_intersections_decreasing = TRUE,
custom_intersections_order = NULL,
intersection_color = "black",
highlight_intersections = NULL,
highlight_color = "darkorange",
empty_point_size = 1.5,
bar_width = 0.7,
text_angle = 0,
text_size = 10,
set_label_size = 3,
intersection_label_size = 3,
point_outline_color = "black",
point_stroke = 0.3,
set_size_show_values = TRUE,
intersection_size_show_values = TRUE,
show_empty_intersections = FALSE,
show_set_labels = TRUE,
plot_margin = 0.5,
height_ratio = 0.7,
width_ratio = 0.3,
bar_offset = -0.01,
set_text_size = 10,
intersection_title = "Intersection Size",
set_size_title = "Set Size",
matrix_point_shape = 21,
```

```

number_color_threshold = 0.75,
number_colors = c(on_bar = "black", off_bar = "black"),
theme_params = list(background_color = "white", grid_color = "grey92", axis_text_color
  = "black", use_grid = TRUE, border_color = NA),
return_data = FALSE,
order = FALSE,
textsize = 5,
theme = ggplot2::theme_light(),
xlabel = NULL,
ylabel = NULL,
...
)

```

Arguments

| | |
|------------------|---|
| x | A Venn object |
| type | Type of plot: "venn" (traditional Venn diagram), "vennpie" (pie-chart style), "upset" (UpSet plot), or "bar" (bar plot of subset sizes) |
| title | Optional plot title |
| interactive | Logical: create an interactive plot? Default: FALSE |
| filename | Optional file name to save the plot |
| width | Width of the saved plot (default: 8) |
| height | Height of the saved plot (default: 6) |
| dpi | Resolution in dots per inch (default: 300) |
| fill | Colors for filling the circles (venn) |
| alpha | Transparency level (0-1), default: 0.5 (venn) |
| labels | Logical: show set labels? Default: TRUE (venn) |
| counts | Logical: show counts? Default: TRUE (venn) |
| showNumbers | Logical: show counts and percentages? Default: TRUE (venn) |
| numberSize | Size of count labels, default: 4 (venn) |
| numberColor | Color of count labels, default: "black" (venn) |
| labelSize | Size of set labels, default: 4 (venn) |
| labelColor | Color of set labels, default: "black" (venn) |
| borderCol | Logical: color borders? Default: FALSE (venn) |
| fillCol | Logical: fill circles? Default: TRUE (venn) |
| fixedCoords | Logical: fixed coordinates? Default: TRUE (venn) |
| xlim | X axis limits, default: c(0, 1) (venn) |
| ylim | Y axis limits, default: c(0, 1) (venn) |
| show_percentages | Logical: show percentages? Default: TRUE (venn) |
| show_unique_only | Logical: show unique only? Default: FALSE (venn) |
| scaled | Logical: scale circles? Default: FALSE (venn) |
| subset | Character vector of subsets to highlight (vennpie) |
| top | Maximum subsets to display, default: 31 (vennpie) |

| | |
|-------------------------------|---|
| min | Minimum set membership, default: 0 (vennpie/getSet) |
| color | Colors for subsets (vennpie/bar) |
| revcolor | Color for non-highlighted, default: "lightgrey" (vennpie) |
| any | Highlight subsets shared by this many sets (vennpie) |
| show.number | Logical: show counts? Default: TRUE (vennpie) |
| show.x | Logical: show labels? Default: TRUE (vennpie) |
| sep | Character separator, default: "_" (vennpie) |
| log | Logical: use log scale? Default: FALSE (vennpie) |
| base | Base for log transformation (vennpie) |
| percentage | Logical: show percentages? Default: FALSE (vennpie) |
| nintersects | Maximum intersections, default: 40 (upset) |
| min_size | Minimum intersection size, default: 1 (upset) |
| sets_bar_color | Colors for set size bars (upset) |
| main_bar_color | Color for intersection bars, default: "steelblue" (upset) |
| point_size | Size of matrix points, default: 3 (upset) |
| line_size | Width of matrix lines, default: 1 (upset) |
| show_numbers | Logical: show bar counts? Default: TRUE (upset) |
| sort_intersections_by | Sort method: "freq", "degree", or "custom" (upset) |
| sort_sets_by | Sort method: "size", "name", or "custom" (upset) |
| sort_sets_decreasing | Logical: decreasing order? Default: TRUE (upset) |
| custom_sets_order | Custom set order (upset) |
| sort_intersections_decreasing | Logical: decreasing? Default: TRUE (upset) |
| custom_intersections_order | Custom intersection order (upset) |
| intersection_color | Color for dots/lines, default: "black" (upset) |
| highlight_intersections | IDs to highlight (upset) |
| highlight_color | Highlight color, default: "darkorange" (upset) |
| empty_point_size | Empty point size, default: 1.5 (upset) |
| bar_width | Bar width (0-1), default: 0.7 (upset) |
| text_angle | Text angle, default: 0 (upset) |
| text_size | Text size, default: 10 (upset) |
| set_label_size | Set label size, default: 3 (upset) |
| intersection_label_size | Intersection label size, default: 3 (upset) |
| point_outline_color | Point outline color, default: "black" (upset) |

```

point_stroke    Point outline width, default: 0.3 (upset)
set_size_show_values
                Show set values? Default: TRUE (upset)
intersection_size_show_values
                Show intersection values? Default: TRUE (upset)
show_empty_intersections
                Show empty? Default: FALSE (upset)
show_set_labels
                Show set labels? Default: TRUE (upset)
plot_margin    Margin in cm, default: 0.5 (upset)
height_ratio   Matrix height ratio, default: 0.7 (upset)
width_ratio    Set size width ratio, default: 0.3 (upset)
bar_offset     Bar offset, default: -0.01 (upset)
set_text_size  Set label size, default: 10 (upset)
intersection_title
                Intersection title (upset)
set_size_title Set size title (upset)
matrix_point_shape
                Point shape, default: 21 (upset)
number_color_threshold
                Color threshold, default: 0.75 (upset)
number_colors  Label colors vector (upset)
theme_params   Theme parameters list (upset)
return_data    Return data? Default: FALSE (upset)
order          Logical: order bars? Default: FALSE (bar)
textsize      Text size, default: 5 (bar)
theme          ggplot2 theme (bar)
xlabel         X-axis label (bar)
ylabel         Y-axis label (bar)
...           Additional arguments passed to the specific plotting function

```

Value

A ggplot2 or plotly object

Author(s)

Kai Guo

Examples

```

# Create a Venn object
A <- sample(1:100, 40, replace = FALSE)
B <- sample(1:100, 60, replace = FALSE)
C <- sample(1:100, 40, replace = FALSE)
res <- vennndetail(list(A = A, B = B, C = C))

# Traditional Venn diagram

```

```
plot(res, type = "venn")

# Venn diagram with custom colors and transparency
plot(res, type = "venn", fill = c("red", "blue", "green"), alpha = 0.3)

# Venn-pie visualization
plot(res, type = "vennpie")

# Venn-pie with highlighted subsets
plot(res, type = "vennpie", any = 2, log = TRUE)

# UpSet plot
plot(res, type = "upset")

# UpSet plot with custom sorting and highlighting
plot(res, type = "upset",
      sort_sets_by = "size",
      highlight_intersections = c(1, 2),
      highlight_color = "red")

# Bar plot of subset sizes
plot(res, type = "bar")

# Ordered bar plot with larger text
plot(res, type = "bar", order = TRUE, textsize = 8)

# Save plot to file (not run during check)
## Not run:
plot(res, type = "venn", filename = "my_venn.png", width = 10, height = 8)

## End(Not run)

# Create interactive plot
if(interactive()) {
  plot(res, type = "venn", interactive = TRUE)
}
```

print.upset_grob *Print method for upset_grob objects*

Description

Print method for upset_grob objects

Usage

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

Arguments

| | |
|-----|--------------------------------|
| x | An upset_grob object |
| ... | Additional arguments (ignored) |

Value

Invisibly returns the input object

| | |
|--------|---|
| result | <i>Extract results from a Venn object</i> |
|--------|---|

Description

Retrieves results from a Venn object in long or wide format

Result will return output in a table format including the contents of the subsets included in the `venndetail` object

Usage

```
result(object, wide = FALSE)

## S4 method for signature 'Venn'
result(object, wide = FALSE)
```

Arguments

| | |
|--------|--|
| object | A Venn object |
| wide | Logical: should results be returned in wide format? Default: FALSE |

Value

A `data.frame` containing subset information

Author(s)

Kai Guo

Examples

```
A <- sample(1:100, 40, replace = FALSE)
B <- sample(1:100, 60, replace = FALSE)
C <- sample(1:100, 40, replace = FALSE)
res <- venndetail(list(A = A, B = B, C = C))
# Get results in long format
result_long <- result(res)
# Get results in wide format
result_wide <- result(res, wide = TRUE)
A <- sample(1:100, 40, replace = FALSE)
B <- sample(1:100, 60, replace = FALSE)
C <- sample(1:100, 40, replace = FALSE)
res <- venndetail(list(A = A, B = B, C = C))
result <- result(res)
```

| | |
|---------|---|
| rowjoin | <i>Join data.frames by row names or specified columns</i> |
|---------|---|

Description

Joins two data.frames using various join methods
join two dataframes by rownames

Usage

```
rowjoin(x, y, fun = "full_join", by = NULL)

## S4 method for signature 'data.frame,data.frame'
rowjoin(x, y, fun = "full_join")
```

Arguments

| | |
|-----|---|
| x | data.frame x |
| y | data.frame y |
| fun | Different join format: left_join, full_join, right_join (default:full_join) |
| by | Optional vector of column names to join by |

Value

A joined data.frame
dataframe with join results

Author(s)

Kai Guo

Examples

```
library(dplyr)
A <- sample(1:100, 40, replace = FALSE)
B <- sample(1:100, 60, replace = FALSE)
dA <- data.frame(A = A, "FC" = rnorm(40))
dB <- data.frame(B = B, "FC" = rnorm(60))
rownames(dA) <- A
rownames(dB) <- B
# Full join by row names
result <- rowjoin(dA, dB)
# Left join by row names
result <- rowjoin(dA, dB, fun = "left_join")
library(dplyr)
A <- sample(1:100, 40, replace = FALSE)
B <- sample(1:100, 60, replace = FALSE)
dA <- data.frame(A = A, "FC" = rnorm(40))
dB <- data.frame(B = B, "FC" = rnorm(60))
rownames(dA) <- A
rownames(dB) <- B
rowjoin(dA, dB)
```

| | |
|----------|-------------------------------------|
| saveVenn | <i>Save a Venn object to a file</i> |
|----------|-------------------------------------|

Description

Saves a Venn object to an RDS file for later use

Usage

```
saveVenn(object, file)
```

Arguments

| | |
|--------|----------------------|
| object | A Venn object |
| file | File name to save to |

Value

The file name (invisibly)

Author(s)

Kai Guo

Examples

```
## Not run:  
# Create a Venn object  
A <- sample(1:100, 40, replace = FALSE)  
B <- sample(1:100, 60, replace = FALSE)  
res <- vennDetail(list(A = A, B = B))  
  
# Save to a file  
saveVenn(res, "my_venn.rds")  
  
## End(Not run)
```

| | |
|----------|--|
| setcolor | <i>return colors with given a vector</i> |
|----------|--|

Description

Setcolor will provide a list of color vectors based on the number used as an input.

Usage

```
setcolor(n, palette = "default")
```

Arguments

n Number of colors needed
palette Type of palette: "default", "categorical", "sequential", or "diverging"

Value

color vector

Author(s)

Kai Guo

Examples

```
mycol <- setcolor(10)
mycol
```

show Venn

Show the summary of venn object

Description

This function provides a summary of the venn object, including a full results and subsets as well as an summary information.

Usage

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

Arguments

object venn object

Value

summary information for the venn object

Author(s)

Kai Guo

Examples

```
A <- sample(1:100, 40, replace = FALSE)
B <- sample(1:100, 60, replace = FALSE)
C <- sample(1:100, 40, replace = FALSE)
res <- vennetail(list(A = A, B = B, C = C))
show(res)
```

| | |
|--------------|--|
| summary.Venn | <i>Give summary information of Venn object</i> |
|--------------|--|

Description

print the summary information of Venn object

Usage

```
## S3 method for class 'Venn'  
summary(object, ...)
```

Arguments

| | |
|--------|--|
| object | Venn object |
| ... | other arguments ignored (for compatibility with generic) |

Value

summary information

Examples

```
A <- sample(1:100, 40, replace = FALSE)  
B <- sample(1:100, 60, replace = FALSE)  
C <- sample(1:100, 40, replace = FALSE)  
res <- vennndetail(list(A = A, B = B, C = C))  
summary(res)
```

| | |
|------|---------------------|
| T2DM | <i>T2DM Dataset</i> |
|------|---------------------|

Description

T2DM data are differential expression genes (DEGs) with annotation from the publication by Hinder et al.

T2DM data are differential expression genes (DEGs) with annotation from the publication by Hinder et al. The data contains three DEG sets from three different tissues (Cortex,SCN,Glom). DEGs were determined by using Cuffdiff with a false discovery rate (FDR) < 0.05 between groups with or without pioglitazone treatment.

Usage

```
T2DM
```

```
T2DM
```

Format

A list of data frames with five columns each

A list of data frame with five columns individually:

Entrez Entrez gene IDs

Symbol HGNC symbols

Annotation Gene function

log2FC log2 Fold Change

FDR False Discovery Rate

Examples

T2DM

upsetPlot

Create an UpSet plot

Description

Creates an UpSet plot to visualize set intersections

Usage

```
upsetPlot(  
  object,  
  nintersects = 40,  
  min_size = 1,  
  sets_bar_color = NULL,  
  main_bar_color = "steelblue",  
  point_size = 3,  
  line_size = 1,  
  show_numbers = TRUE,  
  sort_intersections_by = "freq",  
  sort_sets_by = "size",  
  sort_sets_decreasing = TRUE,  
  custom_sets_order = NULL,  
  sort_intersections_decreasing = TRUE,  
  custom_intersections_order = NULL,  
  intersection_color = "black",  
  highlight_intersections = NULL,  
  highlight_color = "darkorange",  
  empty_point_size = 1.5,  
  bar_width = 0.7,  
  text_angle = 0,  
  text_size = 10,  
  set_label_size = 3,  
  intersection_label_size = 3,  
  point_outline_color = "black",  
  point_stroke = 0.3,  
)
```

```

set_size_show_values = TRUE,
intersection_size_show_values = TRUE,
show_empty_intersections = FALSE,
show_set_labels = TRUE,
plot_margin = 0.5,
height_ratio = 0.7,
width_ratio = 0.3,
bar_offset = -0.01,
set_text_size = 10,
intersection_title = "Intersection Size",
set_size_title = "Set Size",
matrix_point_shape = 21,
number_color_threshold = 0.75,
number_colors = c(on_bar = "black", off_bar = "black"),
theme_params = list(background_color = "white", grid_color = "grey92", axis_text_color
  = "black", use_grid = TRUE, border_color = NA),
title = NULL,
interactive = FALSE,
return_data = FALSE,
...
)

```

Arguments

| | |
|--|---|
| <code>object</code> | A Venn object |
| <code>nintersects</code> | Maximum number of intersections to show |
| <code>min_size</code> | Minimum intersection size to include (default: 1) |
| <code>sets_bar_color</code> | Colors for the set size bars |
| <code>main_bar_color</code> | Color for the intersection size bars |
| <code>point_size</code> | Size of points in the matrix |
| <code>line_size</code> | Width of lines in the matrix |
| <code>show_numbers</code> | Logical: show counts on bars? |
| <code>sort_intersections_by</code> | How to sort intersections |
| <code>sort_sets_by</code> | How to sort sets |
| <code>sort_sets_decreasing</code> | Whether to sort sets in decreasing order |
| <code>custom_sets_order</code> | Custom order for sets if <code>sort_sets_by="custom"</code> |
| <code>sort_intersections_decreasing</code> | Whether to sort intersections in decreasing order |
| <code>custom_intersections_order</code> | Custom order for intersections |
| <code>intersection_color</code> | Color for intersection dots and lines |
| <code>highlight_intersections</code> | Vector of intersection IDs to highlight |
| <code>highlight_color</code> | Color for highlighted intersections |

| | |
|-------------------------------|---|
| empty_point_size | Size of empty points in the matrix |
| bar_width | Width of bars |
| text_angle | Angle for text labels |
| text_size | Size of text in the plot |
| set_label_size | Size of set size labels |
| intersection_label_size | Size of intersection size labels |
| point_outline_color | Color for the outline of points |
| point_stroke | Width of point outline |
| set_size_show_values | Whether to show set size values |
| intersection_size_show_values | Whether to show intersection size values |
| show_empty_intersections | Whether to show empty intersections |
| show_set_labels | Whether to show set labels |
| plot_margin | Margin around the plots in cm |
| height_ratio | Ratio of matrix to total height |
| width_ratio | Ratio of set size to total width |
| bar_offset | Horizontal offset for top bars |
| set_text_size | Size of set labels |
| intersection_title | Title for the intersection size plot |
| set_size_title | Title for the set size plot |
| matrix_point_shape | Shape of the dots in the matrix |
| number_color_threshold | Fraction of max value for label color switch |
| number_colors | Colors for labels on/off bars |
| theme_params | Theme parameters for customization |
| title | Optional plot title |
| interactive | Create interactive plot? |
| return_data | Whether to return the data along with the plot |
| ... | Additional arguments passed to internal functions |

Value

A ggplot object or a combined grid layout

Author(s)

Kai Guo

Examples

```
# Basic example
sets <- list(
  "Set A" = c(1:100),
  "Set B" = c(30:120),
  "Set C" = c(20:50, 90:110),
  "Set D" = c(10:40, 80:120)
)
ven <- venndetail(sets)
upsetPlot(ven, bar_offset = -0.02)

# With highlighting
upsetPlot(ven,
  highlight_intersections = c(1, 2),
  highlight_color = "darkorange",
  bar_offset = -0.02)
```

upsetPlot, Venn-method *Create an UpSet plot for a Venn object*

Description

Creates an UpSet plot to visualize set intersections

Usage

```
## S4 method for signature 'Venn'
upsetPlot(
  object,
  nintersects = 40,
  min_size = 1,
  sets_bar_color = NULL,
  main_bar_color = "steelblue",
  point_size = 3,
  line_size = 1,
  show_numbers = TRUE,
  sort_intersections_by = "freq",
  sort_sets_by = "size",
  sort_sets_decreasing = TRUE,
  custom_sets_order = NULL,
  sort_intersections_decreasing = TRUE,
  custom_intersections_order = NULL,
  intersection_color = "black",
  highlight_intersections = NULL,
  highlight_color = "darkorange",
  empty_point_size = 1.5,
  bar_width = 0.7,
  text_angle = 0,
  text_size = 10,
  set_label_size = 3,
  intersection_label_size = 3,
```

```

point_outline_color = "black",
point_stroke = 0.3,
set_size_show_values = TRUE,
intersection_size_show_values = TRUE,
show_empty_intersections = FALSE,
show_set_labels = TRUE,
plot_margin = 0.5,
height_ratio = 0.7,
width_ratio = 0.3,
bar_offset = -0.01,
set_text_size = 10,
intersection_title = "Intersection Size",
set_size_title = "Set Size",
matrix_point_shape = 21,
number_color_threshold = 0.75,
number_colors = c(on_bar = "black", off_bar = "black"),
theme_params = list(background_color = "white", grid_color = "grey92", axis_text_color
  = "black", use_grid = TRUE, border_color = NA),
title = NULL,
interactive = FALSE,
return_data = FALSE
)

```

Arguments

| | |
|-------------------------------|---|
| object | A Venn object |
| nintersects | Maximum number of intersections to show (default: 40) |
| min_size | Minimum intersection size to include (default: 1) |
| sets_bar_color | Colors for the set size bars (default: NULL for auto-generate) |
| main_bar_color | Color for the intersection size bars (default: "steelblue") |
| point_size | Size of points in the matrix (default: 3) |
| line_size | Width of lines in the matrix (default: 1) |
| show_numbers | Logical: show counts on bars? (default: TRUE) |
| sort_intersections_by | How to sort intersections: "freq" (default), "degree" |
| sort_sets_by | How to sort sets: "size" (default), "name" |
| sort_sets_decreasing | Whether to sort sets in decreasing order (default: TRUE) |
| custom_sets_order | Custom order for sets if sort_sets_by="custom" |
| sort_intersections_decreasing | Whether to sort intersections in decreasing order (default: TRUE) |
| custom_intersections_order | Custom order for intersections if sort_intersections_by="custom" |
| intersection_color | Color for intersection dots and lines (default: "black") |
| highlight_intersections | Vector of intersection IDs to highlight (default: NULL) |

| | |
|-------------------------------|---|
| highlight_color | Color for highlighted intersections (default: "darkorange") |
| empty_point_size | Size of empty points in the matrix (default: 1.5) |
| bar_width | Width of bars (0-1 scale) (default: 0.7) |
| text_angle | Angle for text labels (default: 0) |
| text_size | Size of text in the plot (default: 10) |
| set_label_size | Size of set size labels (default: 3) |
| intersection_label_size | Size of intersection size labels (default: 3) |
| point_outline_color | Color for the outline of points (default: "black") |
| point_stroke | Width of point outline (default: 0.3) |
| set_size_show_values | Whether to show set size values (default: TRUE) |
| intersection_size_show_values | Whether to show intersection size values (default: TRUE) |
| show_empty_intersections | Whether to show empty intersections (default: FALSE) |
| show_set_labels | Whether to show set labels (default: TRUE) |
| plot_margin | Margin around the plots in cm (default: 0.5) |
| height_ratio | Ratio of matrix to total height (default: 0.7) |
| width_ratio | Ratio of set size to total width (default: 0.3) |
| bar_offset | Horizontal offset for top bars to improve alignment (default: 0) |
| set_text_size | Size of set labels (default: 10) |
| intersection_title | Title for the intersection size plot (default: "Intersection Size") |
| set_size_title | Title for the set size plot (default: "Set Size") |
| matrix_point_shape | Shape of the dots in the matrix (21=filled circle) (default: 21) |
| number_color_threshold | Fraction of max value where number color switches (default: 0.75) |
| number_colors | Named vector with colors for labels on/off bars (default: c(on_bar="white", off_bar="black")) |
| theme_params | List of theme parameters for customization (default: list of defaults) |
| title | Optional plot title |
| interactive | Logical: create interactive plot? (default: FALSE) |
| return_data | Whether to return the data along with the plot (default: FALSE) |

Value

A ggplot object or a combined grid layout

Author(s)

Kai Guo

Examples

```
# Basic example
sets <- list(
  "Set A" = c(1:100),
  "Set B" = c(30:120),
  "Set C" = c(20:50, 90:110),
  "Set D" = c(10:40, 80:120)
)
ven <- venndetail(sets)
upsetPlot(ven, bar_offset = -0.02)

# With highlighting
upsetPlot(ven,
  highlight_intersections = c(1, 2),
  highlight_color = "darkorange",
  bar_offset = -0.02)
```

upset_plot

Create an UpSet plot for set intersection visualization

Description

Creates a custom UpSet plot showing the intersections between sets. It displays the size of each intersection and the sets involved in each intersection.

Usage

```
upset_plot(
  data_list,
  sets = NULL,
  min_intersection_size = 1,
  max_sets_display = NULL,
  sort_sets_by = "size",
  sort_sets_decreasing = TRUE,
  custom_sets_order = NULL,
  sort_intersections_by = "freq",
  sort_intersections_decreasing = TRUE,
  custom_intersections_order = NULL,
  intersection_color = "black",
  main_bar_color = "steelblue",
  sets_bar_colors = NULL,
  highlight_intersections = NULL,
  highlight_color = "darkorange",
  point_outline_color = "black",
  filled_point_size = 2,
  empty_point_size = 1.5,
  line_size = 0.5,
  bar_width = 0.6,
  point_stroke = 0.3,
  text_angle = 0,
  text_size = 10,
```

```

set_text_size = 10,
set_label_size = 3,
intersection_label_size = 3,
intersection_title = "Intersection Size",
set_size_title = "Set Size",
matrix_point_shape = 21,
set_size_show_values = TRUE,
intersection_size_show_values = TRUE,
show_empty_intersections = FALSE,
show_set_labels = TRUE,
show_numbers_on_bars = TRUE,
number_color_threshold = 0.75,
number_colors = c(on_bar = "black", off_bar = "black"),
plot_margin = 0.5,
height_ratio = 0.5,
width_ratio = 0.3,
bar_offset = -0.01,
theme_params = list(background_color = "white", grid_color = "grey92", axis_text_color
  = "black", use_grid = TRUE, border_color = NA),
return_data = FALSE
)

```

Arguments

| | |
|--|--|
| <code>data_list</code> | A named list of vectors, each containing elements in a set |
| <code>sets</code> | Optional vector of set names to include (default: all sets in <code>data_list</code>) |
| <code>min_intersection_size</code> | Minimum intersection size to include (default: 1) |
| <code>max_sets_display</code> | Maximum number of sets to display (default: all) |
| <code>sort_sets_by</code> | How to sort the sets: "size", "name", or "custom" (default: "size") |
| <code>sort_sets_decreasing</code> | Whether to sort sets in decreasing order (default: TRUE) |
| <code>custom_sets_order</code> | Custom order for sets if <code>sort_sets_by="custom"</code> |
| <code>sort_intersections_by</code> | How to sort intersections: "freq", "degree", or "custom" (default: "freq") |
| <code>sort_intersections_decreasing</code> | Whether to sort intersections in decreasing order (default: TRUE) |
| <code>custom_intersections_order</code> | Custom order for intersections if <code>sort_intersections_by="custom"</code> |
| <code>intersection_color</code> | Color for intersection dots and lines (default: "black") |
| <code>main_bar_color</code> | Color for the intersection size bars (default: "steelblue") |
| <code>sets_bar_colors</code> | Named vector of colors for each set (default: auto-generated) |
| <code>highlight_intersections</code> | Vector of intersection IDs to highlight (default: NULL) |
| <code>highlight_color</code> | Color for highlighted intersections (default: "#FF5500") |

| | |
|-------------------------------|---|
| point_outline_color | Color for the outline of points (default: "black") |
| filled_point_size | Size of filled points in the matrix (default: 2) |
| empty_point_size | Size of empty points in the matrix (default: 1.5) |
| line_size | Width of connecting lines (default: 0.5) |
| bar_width | Width of bars (0-1 scale) (default: 0.7) |
| point_stroke | Width of point outline (default: 0.3) |
| text_angle | Angle for text labels (default: 0) |
| text_size | Size of text in the plot (default: 10) |
| set_text_size | Size of set labels (default: 10) |
| set_label_size | Size of set size labels (default: 3) |
| intersection_label_size | Size of intersection size labels (default: 3) |
| intersection_title | Title for the intersection size plot (default: "Intersection Size") |
| set_size_title | Title for the set size plot (default: "Set Size") |
| matrix_point_shape | Shape of the dots in the matrix (21=filled circle) (default: 21) |
| set_size_show_values | Whether to show set size values (default: TRUE) |
| intersection_size_show_values | Whether to show intersection size values (default: TRUE) |
| show_empty_intersections | Whether to show empty intersections (default: FALSE) |
| show_set_labels | Whether to show set labels (default: TRUE) |
| show_numbers_on_bars | Logical, whether to display counts on bars (default: TRUE) |
| number_color_threshold | Fraction of max value where number color switches (default: 0.75) |
| number_colors | Named vector with colors for labels on/off bars (default: c(on_bar="white", off_bar="black")) |
| plot_margin | Margin around the plots in cm (default: 0.5) |
| height_ratio | Ratio of matrix to total height (default: 0.7) |
| width_ratio | Ratio of set size to total width (default: 0.3) |
| bar_offset | Horizontal offset for top bars to improve alignment (default: 0) |
| theme_params | List of theme parameters for customization (default: list of defaults) |
| return_data | Whether to return the data along with the plot (default: FALSE) |

Value

If `return_data=FALSE`, returns the patchwork plot object. If `return_data=TRUE`, returns a list containing the plot and component data.

Author(s)

Kai Guo

Examples

```

# Basic example
sets <- list(
  "Set A" = c(1:100),
  "Set B" = c(30:120),
  "Set C" = c(20:50, 90:110),
  "Set D" = c(10:40, 80:120)
)
upset_plot(sets)

# With highlighting
upset_plot(sets,
  highlight_intersections = c(1, 2),
  highlight_color = "darkorange",
  bar_offset = -0.02)

# Custom colors
set_colors <- c("Set A" = "blue", "Set B" = "green",
  "Set C" = "orange", "Set D" = "purple")
upset_plot(sets, sets_bar_colors = set_colors,
  main_bar_color = "darkblue")

```

Venn-class

Venn Class

Description

S4 class to store and manage set intersection data and visualizations

Slots

input A list containing the original input datasets

raw A named vector with counts of elements in each input set

sep The character used to separate set names in subset labels

GroupNames A character vector of input group names

result A data.frame containing subset information (subset name and elements)

detail A named vector with counts of elements in each subset

wide A data.frame with subset information in wide format for easier analysis

metadata A list to store additional metadata about the analysis

Author(s)

Kai Guo

Examples

```
## Not run:
A <- sample(1:100, 40, replace = FALSE)
B <- sample(1:100, 60, replace = FALSE)
C <- sample(1:100, 40, replace = FALSE)
venn_obj <- vennndetail(list(A = A, B = B, C = C))
# Access the detail slot
venn_obj@detail

## End(Not run)
```

vennApp

Create an interactive Venn diagram app

Description

Creates a Shiny app for interactive exploration of Venn diagrams

Usage

```
vennApp(object, launch = TRUE, ...)
```

Arguments

| | |
|--------|--|
| object | A Venn object |
| launch | Launch the app immediately? Default: TRUE |
| ... | Additional arguments passed to shiny::runApp |

Value

A Shiny app object (invisibly)

Author(s)

Kai Guo

Examples

```
## Not run:
# Create a Venn object
A <- sample(1:100, 40, replace = FALSE)
B <- sample(1:100, 60, replace = FALSE)
C <- sample(1:100, 40, replace = FALSE)
res <- vennndetail(list(A = A, B = B, C = C))

# Launch interactive app
vennApp(res)

## End(Not run)
```

`venndetail`*Create a Venn object for set analysis*

Description

Extracts shared and unique elements from multiple sets and creates a Venn object for analysis and visualization

Usage

```
venndetail(  
  x,  
  sep = "_",  
  abbr = FALSE,  
  minlength = 3,  
  abbr.method = "both.sides",  
  verbose = FALSE  
)
```

Arguments

| | |
|--------------------------|--|
| <code>x</code> | A list of vectors with group names |
| <code>sep</code> | Symbol character used when concatenating group names into subset names (default: `_'`) |
| <code>abbr</code> | Logical: abbreviate subset names? Default: FALSE |
| <code>minlength</code> | Minimal length for abbreviated subset names. Default: 3 |
| <code>abbr.method</code> | Method for abbreviation: "both.sides", "left.sides", or "right.sides" |
| <code>verbose</code> | Logical: show progress messages? Default: FALSE |

Value

A Venn object

Author(s)

Kai Guo

Examples

```
# Create a Venn object with three sets  
A <- sample(1:100, 40, replace = FALSE)  
B <- sample(1:100, 60, replace = FALSE)  
C <- sample(1:100, 40, replace = FALSE)  
res <- venndetail(list(A = A, B = B, C = C))  
  
# Examine the results  
summary(res)  
  
# Plot the results  
plot(res, type = "venn")
```

```
# With abbreviated set names
sets <- list(
  LongNameGroup1 = sample(1:100, 40),
  LongNameGroup2 = sample(1:100, 50),
  LongNameGroup3 = sample(1:100, 45)
)
res <- vennDetail(sets, abbr = TRUE, minlength = 4)
```

vennDiagram*Create a Venn diagram*

Description

Creates a traditional Venn diagram for 2-5 sets

Usage

```
vennDiagram(
  object,
  fill = NULL,
  alpha = 0.5,
  labels = TRUE,
  counts = TRUE,
  showNumbers = TRUE,
  numberSize = 4,
  numberColor = "black",
  labelSize = 4,
  labelColor = "black",
  borderCol = FALSE,
  fillCol = TRUE,
  fixedCoords = TRUE,
  xlim = c(0, 1),
  ylim = c(0, 1),
  show_percentages = TRUE,
  show_unique_only = FALSE,
  scaled = FALSE,
  title = NULL,
  interactive = FALSE,
  ...
)
```

Arguments

| | |
|-------------|--|
| object | A Venn object |
| fill | Colors for filling the circles |
| alpha | Transparency level for the circles (0-1) |
| labels | Logical: show set labels? (default: TRUE) |
| counts | Logical: show counts? (default: TRUE) |
| showNumbers | Logical: whether to show counts and percentages in each region |
| numberSize | Size of the count labels |

| | |
|------------------|---|
| numberColor | Color of the count labels |
| labelSize | Size of the set labels |
| labelColor | Color of the set labels |
| borderCol | Logical: whether to color the borders of circles |
| fillCol | Logical: whether to fill circles with colors |
| fixedCoords | Logical: whether to use fixed coordinates |
| xlim | Vector with 2 numbers, x axis limits for the venn diagram |
| ylim | Vector with 2 numbers, y axis limits for the venn diagram |
| show_percentages | Logical: show percentages alongside counts? (default: TRUE) |
| show_unique_only | Logical: show counts only for unique elements? (default: FALSE) |
| scaled | Logical: scale circles by set size? (default: FALSE) |
| title | Optional plot title |
| interactive | Logical: create interactive plot? (default: FALSE) |
| ... | Additional arguments passed to ggplot2 functions |

Value

A ggplot2 or plotly object

Author(s)

Kai Guo

vennDiagram, Venn-method

Create a traditional Venn diagram

Description

Creates a traditional Venn diagram for 2-5 sets

Usage

```
## S4 method for signature 'Venn'
vennDiagram(
  object,
  fill = NULL,
  alpha = 0.5,
  labels = TRUE,
  counts = TRUE,
  showNumbers = TRUE,
  numberSize = 4,
  numberColor = "black",
  labelSize = 4,
  labelColor = "black",
```

```

borderCol = FALSE,
fillCol = TRUE,
fixedCoords = TRUE,
xlim = c(0, 1),
ylim = c(0, 1),
show_percentages = TRUE,
show_unique_only = FALSE,
scaled = FALSE,
title = NULL,
interactive = FALSE,
...
)

```

Arguments

| | |
|------------------|---|
| object | A Venn object |
| fill | Colors for filling the circles |
| alpha | Transparency level for the circles (0-1) |
| labels | Logical: show set labels? (default: TRUE) |
| counts | Logical: show counts? (default: TRUE) |
| showNumbers | Logical: whether to show counts and percentages in each region |
| numberSize | Size of the count labels |
| numberColor | Color of the count labels |
| labelSize | Size of the set labels |
| labelColor | Color of the set labels |
| borderCol | Logical: whether to color the borders of circles |
| fillCol | Logical: whether to fill circles with colors |
| fixedCoords | Logical: whether to use fixed coordinates |
| xlim | Vector with 2 numbers, x axis limits for the venn diagram |
| ylim | Vector with 2 numbers, y axis limits for the venn diagram |
| show_percentages | Logical: show percentages alongside counts? (default: TRUE) |
| show_unique_only | Logical: show counts only for unique elements? (default: FALSE) |
| scaled | Logical: scale circles by set size? (default: FALSE) |
| title | Optional plot title |
| interactive | Logical: create interactive plot? (default: FALSE) |
| ... | Additional arguments passed to ggplot2 functions |

Value

A ggplot2 or plotly object

Author(s)

Kai Guo

`vennEnrichment`*Perform enrichment analysis on set intersections*

Description

Performs enrichment analysis to identify overrepresented categories in set intersections

Usage

```
vennEnrichment(  
  object,  
  annotation,  
  id_col,  
  category_col,  
  subsets = NULL,  
  min_overlap = 3,  
  adjust.method = "BH",  
  sig_threshold = 0.05  
)
```

Arguments

| | |
|----------------------------|---|
| <code>object</code> | A Venn object |
| <code>annotation</code> | A data frame with annotation data (e.g., Gene Ontology terms) |
| <code>id_col</code> | Column in the annotation data frame containing identifiers matching elements in the Venn object |
| <code>category_col</code> | Column in the annotation data frame containing category information |
| <code>subsets</code> | Character vector of subset names to analyze (default: NULL, all subsets) |
| <code>min_overlap</code> | Minimum number of elements a category must share with a subset (default: 3) |
| <code>adjust.method</code> | Method for multiple testing correction (default: "BH") |
| <code>sig_threshold</code> | Significance threshold for p-values (default: 0.05) |

Value

A data.frame with enrichment analysis results

Author(s)

Kai Guo

Examples

```
# Create a Venn object with gene sets  
A <- sample(1:1000, 100, replace = FALSE)  
B <- sample(1:1000, 150, replace = FALSE)  
C <- sample(1:1000, 120, replace = FALSE)  
res <- vennDetail(list(A = A, B = B, C = C))  
  
# Create simulated annotation data  
gene_ids <- 1:1000
```

```
categories <- sample(c("Category1", "Category2", "Category3", "Category4"),
                    1000, replace = TRUE)
anno <- data.frame(GeneID = gene_ids, Category = categories)

# Perform enrichment analysis
enrichment <- vennEnrichment(res, anno, "GeneID", "Category")
```

vennpie

Create a Venn-pie visualization

Description

Creates a pie-chart-like visualization of set intersections

Usage

```
vennpie(
  object,
  subset = NULL,
  top = 31,
  min = 0,
  color = NULL,
  revcolor = "lightgrey",
  any = NULL,
  show.number = TRUE,
  show.x = TRUE,
  sep = "_",
  log = FALSE,
  base = NULL,
  percentage = FALSE,
  title = NULL,
  interactive = FALSE,
  ...
)
```

Arguments

| | |
|-------------|--|
| object | A Venn object |
| subset | Character vector of subset names to highlight |
| top | Maximum number of subsets to display |
| min | Minimum number of sets an element must be in |
| color | Optional vector of colors for the subsets |
| revcolor | Color for non-highlighted subsets |
| any | Highlight subsets shared by exactly this many sets |
| show.number | Logical: show counts in labels? |
| show.x | Logical: show subset labels? |
| sep | Character separator for subset names |
| log | Logical: use log scale for counts? |

| | |
|-------------|--|
| base | Base for log transformation if log=TRUE |
| percentage | Logical: show percentages instead of counts? |
| title | Optional plot title |
| interactive | Logical: create interactive plot? |
| ... | Additional arguments |

Value

A ggplot2 or plotly object

Author(s)

Kai Guo

vennpie, Venn-method *Create a Venn-pie visualization*

Description

Creates a pie-chart-like visualization of set intersections, which is particularly useful for visualizing more than 5 sets

Usage

```
## S4 method for signature 'Venn'
vennpie(
  object,
  subset = NULL,
  top = 31,
  min = 0,
  color = NULL,
  revcolor = "lightgrey",
  any = NULL,
  show.number = TRUE,
  show.x = TRUE,
  sep = "_",
  log = FALSE,
  base = NULL,
  percentage = FALSE,
  title = NULL,
  interactive = FALSE,
  ...
)
```

Arguments

| | |
|--------|---|
| object | A Venn object |
| subset | Character vector of subset names to highlight |
| top | Maximum number of subsets to display. Default: 31 |

| | |
|-------------|---|
| min | Minimum number of sets an element must be in. Default: 0 |
| color | Optional vector of colors for the subsets |
| revcolor | Color for non-highlighted subsets. Default: "lightgrey" |
| any | Highlight subsets shared by exactly this many sets |
| show.number | Logical: show counts in labels? Default: TRUE |
| show.x | Logical: show subset labels? Default: TRUE |
| sep | Character separator for subset names |
| log | Logical: use log scale for counts? Default: FALSE |
| base | Base for log transformation if log=TRUE |
| percentage | Logical: show percentages instead of counts? Default: FALSE |
| title | Optional plot title |
| interactive | Logical: create interactive plot? Default: FALSE |
| ... | Additional arguments |

Value

A ggplot2 or plotly object

Author(s)

Kai Guo

| | |
|-----------|---|
| vennStats | <i>Perform statistical tests on set intersections</i> |
|-----------|---|

Description

Performs statistical tests to evaluate the significance of set intersections

Usage

```
vennStats(
  object,
  universe = NULL,
  method = c("hypergeometric", "permutation"),
  nperm = 1000,
  adjust.method = "BH",
  include_singles = FALSE
)
```

Arguments

| | |
|----------|--|
| object | A Venn object |
| universe | Size of the universe for hypergeometric test. Default: NULL (will use the union of all sets) |
| method | Statistical method to use: "hypergeometric" or "permutation". Default: "hypergeometric" |

nperm Number of permutations if method="permutation". Default: 1000
 adjust.method Method for multiple testing correction. Default: "BH"
 include_singles Logical: include tests for single sets? Default: FALSE

Value

A data.frame with statistical test results

Author(s)

Kai Guo

Examples

```
A <- sample(1:1000, 100, replace = FALSE)
B <- sample(1:1000, 150, replace = FALSE)
C <- sample(1:1000, 120, replace = FALSE)
res <- vennDetail(list(A = A, B = B, C = C))
stats <- vennStats(res)
```

vennStats, Venn-method *Perform statistical tests on set intersections*

Description

Performs statistical tests to evaluate the significance of set intersections

Usage

```
## S4 method for signature 'Venn'
vennStats(
  object,
  universe = NULL,
  method = c("hypergeometric", "permutation"),
  nperm = 1000,
  adjust.method = "BH",
  include_singles = FALSE
)
```

Arguments

object A Venn object
 universe Size of the universe for hypergeometric test (default: NULL, will use the union of all sets)
 method Statistical method to use: "hypergeometric" or "permutation" (default: "hypergeometric")
 nperm Number of permutations if method="permutation" (default: 1000)
 adjust.method Method for multiple testing correction (default: "BH")
 include_singles Logical: include tests for single sets? (default: FALSE)

Value

A data.frame with statistical test results

Author(s)

Kai Guo

Examples

```
# Create a Venn object
A <- sample(1:1000, 100, replace = FALSE)
B <- sample(1:1000, 150, replace = FALSE)
C <- sample(1:1000, 120, replace = FALSE)
res <- venndetail(list(A = A, B = B, C = C))

# Perform statistical tests
stats <- vennStats(res)

# With custom universe size
stats <- vennStats(res, universe = 2000)

# Using permutation test
stats <- vennStats(res, method = "permutation", nperm = 500)
```

[.Venn

Subset a Venn object

Description

Extract elements from the result slot of a Venn object using bracket notation.

Usage

```
## S3 method for class 'Venn'
x[i, j]
```

Arguments

| | |
|---|----------------|
| x | Venn object |
| i | Row indices |
| j | Column indices |

Value

Subset of the result data.frame

| | |
|---------|--|
| \$.Venn | <i>Extract column from a Venn object</i> |
|---------|--|

Description

Extract a column from the result slot of a Venn object using \$ notation.

Usage

```
## S3 method for class 'Venn'  
x$name
```

Arguments

| | |
|------|------------------------|
| x | Venn object |
| name | Column name to extract |

Value

Vector containing the specified column

Index

- * **datasets**
 - T2DM, [30](#)
- * **internal**
 - .findOverlap, [5](#)
 - .vennDiagram, [6](#)
 - make_truth_table, [17](#)
 - makeContent.upset_grob, [16](#)
- * **package**
 - VennDetail-package, [3](#)
 - .add_colnames, [4](#)
 - .findOverlap, [5](#)
 - .make.table, [5](#)
 - .vennDiagram, [6](#)
 - [.Venn, [51](#)
 - \$.Venn, [52](#)
- as.data.frame.Venn, [7](#)
- compareVenn, [7](#)
- create_interactive_vennpie, [8](#)
- detail, [9](#)
- detail, Venn-method (detail), [9](#)
- dim.Venn, [10](#)
- dplot, [10](#)
- dplot, Venn-method (dplot), [10](#)
- getFeature, [11](#)
- getFeature, Venn-method (getFeature), [11](#)
- getSet, [13](#)
- getSet, Venn-method (getSet), [13](#)
- head.Venn, [14](#)
- loadVenn, [15](#)
- make_subset, [15](#)
- make_subset, [16](#)
- make_truth_table, [17](#)
- makeContent.upset_grob, [16](#)
- merge.Venn, [18](#)
- names.Venn, [18](#)
- newVenn, [19](#)
- plot.Venn, [20](#)
- print.upset_grob, [25](#)
- result, [26](#)
- result, Venn-method (result), [26](#)
- rowjoin, [27](#)
- rowjoin, data.frame, data.frame-method (rowjoin), [27](#)
- saveVenn, [28](#)
- setcolor, [28](#)
- show Venn, [29](#)
- show, Venn-method (show Venn), [29](#)
- summary.Venn, [30](#)
- T2DM, [30](#)
- tail.Venn (head.Venn), [14](#)
- upset_plot, [37](#)
- upsetPlot, [31](#)
- upsetPlot, Venn-method, [34](#)
- Venn-class, [40](#)
- vennApp, [41](#)
- VennDetail (VennDetail-package), [3](#)
- venndetail, [42](#)
- VennDetail-package, [3](#)
- vennDiagram, [43](#)
- vennDiagram, Venn-method, [44](#)
- vennEnrichment, [46](#)
- vennpie, [47](#)
- vennpie, Venn-method, [48](#)
- vennStats, [49](#)
- vennStats, Venn-method, [50](#)