Package 'VennDetail'

July 17, 2025

Type Package

Title Comprehensive Visualization and Analysis of Multi-Set Intersections

Version 1.25.2

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, methods, ggplot2, grDevices, magrittr, patchwork, plotly, purrr, rlang, shiny, stats, tibble, tidyr, htmlwidgets, utils

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

VignetteBuilder knitr

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

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

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

RoxygenNote 7.3.2

biocViews DataRepresentation, GraphAndNetwork, Visualization, Software

NeedsCompilation no

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

git_branch devel

git_last_commit 5af2273

git_last_commit_date 2025-07-14

Repository Bioconductor 3.22

Date/Publication 2025-07-16

Author Kai Guo [aut, cre],
Brett McGregor [aut]

Maintainer Kai Guo <guokai8@gmail.com>

Contents

`VennDetail-package` *VennDetail: Comprehensive Visualization and Analysis of Multi-Set Intersections*

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 ^ 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

add_prefix *Add prefix to column names*

Description

Adds a prefix to column names except for "RowNxyz"

Usage

```
add_prefix(x, prefix, sep = "_")
```

Arguments

x	A data.frame
prefix	Prefix to add
sep	Separator between prefix and original column name

Value

A data.frame with prefixed column names

Author(s)

Kai Guo

add_rownames_column *Add row names as a column*

Description

Adds row names as a column named "RowNxyz"

Usage

```
add_rownames_column(x)
```

Arguments

x	A data.frame

Value

A data.frame with row names as a column

Author(s)

Kai Guo

as.data.frame.Venn *Convert a Venn object to a data frame*

Description

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 summarizes 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 <- venndetail(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 <- venndetail(list(A = A, B = B, C = C))
detail(res)
```

---

detail,Venn-method *Get subset details from a Venn object*

Description

Returns a named numeric vector with counts for each subset

Usage

```
## 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 <- venndetail(list(A = A, B = B, C = C))
detail(res)
```

dim.Venn *Get the dimensions of a Venn object*

Description

Returns the dimensions of the result data frame

Usage

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

Arguments

x	A Venn object

Value

A numeric vector with the number of rows and columns

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 dimensions
dim(res)
```

dplot *Create a bar plot of subset counts*

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 <- venndetail(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 <- venndetail(list(A = A, B = B, C = C))
dplot(res, order = TRUE, textsize = 3)
```

---

dplot,Venn-method *Create a bar plot of subset counts*

Description

Creates a bar plot showing counts for each subset

Usage

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

Arguments

object	A Venn object
order	Logical: should bars be ordered by count?
textsize	Numeric: size of text labels above bars
color	Optional vector of colors for the bars
theme	The ggplot2 theme to use
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 <- venndetail(list(A = A, B = B, C = C))
# Simple bar plot
dplot(res)
# Ordered bars with custom title
dplot(res, order = TRUE, textsize = 3, title = "Set Intersections")
```

---

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 venndetail 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 userowname=FALSE)
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))
```

`getFeature,Venn-method`

Extract feature data for specific subsets

Description

Combines subset information with user-supplied data frames

Usage

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

Arguments

`object`	A Venn object
`subset`	Character vector of subset names to extract
`rlist`	List of user-supplied data frames
`userowname`	Logical: use row names for joining?
`gind`	Column names or indices to use for joining if userowname=FALSE
`sep`	Character used to separate group names in column names
`wide`	Logical: return results in wide format?

Value

A data.frame combining subset information with user data

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))
```

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")
```

getSet,Venn-method *Extract specific subsets from a Venn object*

Description

Extracts elements from specified subsets

Usage

```
## S4 method for signature 'Venn'
getSet(object, subset = NULL, min = 0, wide = FALSE)
```

Arguments

object	A Venn object
subset	Character vector of subset names to extract
min	Minimum number of sets an element must be in
wide	Logical: return results in wide format?

Value

A data.frame with elements from the specified subsets

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")
```

head.Venn *Extract the first few rows of a Venn object*

Description

Extract the first n rows of the result data frame

Usage

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

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

Arguments

x	A Venn object
n	Number of rows to extract
...	Additional arguments passed to head

Value

A data frame with the first n rows

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 the first few rows
head(res)
```

---

loadVenn *Load a Venn object from a file*

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` *Get subset from list of input groups*

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)
```

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` *Utility functions for VennDetail package*

Description

Internal utility functions for the VennDetail package

Combines multiple Venn objects into a single Venn object

Usage

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

Arguments

`x`	First Venn object
`y`	Second Venn object
`ignore.case`	Logical: ignore case in group names? Default: FALSE
`useupper`	Logical: convert all group names to upper case? Default: TRUE
`plot`	Logical: plot the combined result? Default: FALSE
`...`	Additional arguments passed to venndetail

Value

A new Venn object combining the input objects

Author(s)

Kai Guo

Examples

```
# Create two Venn objects
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))

# Merge the two objects
res <- merge(res1, res2)

# Plot the merged result
plot(res)
```

merge.Venn *Merge two or more venndetail obejcts*

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 *Extract subset names from a Venn object*

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)
...	Additional arguments passed to the specific plotting function

Value

A ggplot2 or plotly object

Parameters for Traditional Venn Diagram (type = "venn")

fill Colors for filling the circles

alpha Transparency level for the circles (0-1), default: 0.5

labels Logical: show set labels? Default: TRUE

counts Logical: show counts? Default: TRUE

showNumbers Logical: whether to show counts and percentages in each region, default: TRUE

numberSize Size of the count labels, default: 4

numberColor Color of the count labels, default: "black"

labelSize Size of the set labels, default: 4

labelColor Color of the set labels, default: "black"

borderCol Logical: whether to color the borders of circles, default: FALSE

fillCol Logical: whether to fill circles with colors, default: TRUE

fixedCoords Logical: whether to use fixed coordinates, default: TRUE

xlim Vector with 2 numbers, x axis limits for the venn diagram, default: c(0, 1)

ylim Vector with 2 numbers, y axis limits for the venn diagram, default: c(0, 1)

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

Parameters for Venn-Pie Plot (type = "vennpie")

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, default: "_"

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

Parameters for UpSet Plot (type = "upset")

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

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", or "custom"

sort_sets_by How to sort sets: "size" (default), "name", or "custom"

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

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, default: -0.01

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="black", off_bar="black")

theme_params List of theme parameters for customization

return_data Whether to return the data along with the plot, default: FALSE

Parameters for Bar Plot (type = "bar")

order Logical: should bars be ordered by count? Default: FALSE

textsize Numeric: size of text labels above bars, default: 5

color Optional vector of colors for the bars

theme The ggplot2 theme to use, default: theme_light()

xlabel Optional x-axis label

ylabel Optional y-axis label

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 <- venndetail(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
plot(res, type = "venn", filename = "my_venn.png", width = 10, height = 8)

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

`result` *Extract results from a Venn object*

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)
```

---

result,Venn-method *Extract results from a Venn object*

Description

Retrieves results from a Venn object in long or wide format

Usage

```
## 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)
```

rowjoin *Join data.frames by row names or specified columns*

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)
```

rowjoin,data.frame,data.frame-method *Join data.frames by row names or specified columns*

Description

Joins two data.frames using various join methods

Usage

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

Arguments

x	First data.frame
y	Second data.frame
fun	Join function to use: "full_join", "left_join", "right_join", etc.
by	Optional vector of column names to join by

Value

A joined data.frame

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")
```

saveVenn *Save a Venn object to a file*

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 *return colors with given a vector*

Description

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

Returns a color palette for Venn diagrams

Returns a color palette for Venn diagrams

Usage

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

setcolor(n, palette = "default")

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

Arguments

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

Value

color vector

A vector of colors

A vector of colors

Author(s)

Kai Guo

Examples

```
mycol <- setcolor(10)
mycol
# Get 5 colors from the default palette
cols <- setcolor(5)

# Get 10 colors from the categorical palette
cols <- setcolor(10, palette = "categorical")
# Get 5 colors from the default palette
cols <- setcolor(5)

# Get 10 colors from the categorical palette
cols <- setcolor(10, palette = "categorical")
```

---

set_rownames *Set row names from a column*

Description

Sets row names from a specified column

Usage

```
set_rownames(x, col)
```

Arguments

x	A data.frame
col	Column index or name to use for row names

Value

A data.frame with row names from the specified column

Author(s)

Kai Guo

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 <- venndetail(list(A = A, B = B, C = C))
show(res)
```

show,Venn-method — *Show method for Venn objects*

Description

Display method for Venn objects

Display method for Venn objects

Usage

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

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

Arguments

object	A Venn object

Value

Invisibly returns the Venn object

Invisibly returns the Venn object

Author(s)

Kai Guo

---

summary.Venn *Return summary information about a Venn object*

Description

Summary method for Venn objects

Summary method for Venn objects

Usage

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

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

Arguments

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

Value

Invisibly returns the Venn object

Invisibly returns 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 <- venndetail(list(A = A, B = B, C = C))
summary(res)
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))
summary(res)
```

---

summary.Venn *Give summary information of Venn object*

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 <- venndetail(list(A = A, B = B, C = C))
summary(res)
```

T2DM *T2DM Dataset*

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
```

---

tail.Venn *Extract the last few rows of a Venn object*

Description

Extract the last n rows of the result data frame

Usage

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

Arguments

x	A Venn object
n	Number of rows to extract
...	Additional arguments passed to tail

Value

A data frame with the last n rows

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 the last few rows
tail(res)
```

---

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

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

highlight_color
: 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

data_list	A named list of vectors, each containing elements in a set
sets	Optional vector of set names to include (default: all sets in data_list)
min_intersection_size	Minimum intersection size to include (default: 1)
max_sets_display	Maximum number of sets to display (default: all)
sort_sets_by	How to sort the sets: "size", "name", or "custom" (default: "size")
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_by	How to sort intersections: "freq", "degree", or "custom" (default: "freq")
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")
main_bar_color	Color for the intersection size bars (default: "steelblue")
sets_bar_colors	Named vector of colors for each set (default: auto-generated)
highlight_intersections	Vector of intersection IDs to highlight (default: NULL)
highlight_color	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, bar_offset = -0.02)

# 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 *Class 'Venn' This class includes all information from venndetail*

Description

S4 class to store and manage set intersection data and visualizations

Slots

`input` orginal input datasets

`raw` summary of the input datasets

`sep` separation character

`GroupNames` input group names

`result` shared or unique sets

`detail` shared of unique number belongs to each sets

`wide` result in wide format

`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 <- venndetail(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 <- venndetail(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

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

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

Vennpie uses the venn object and to creates a figure in the form of a venn pie diagram rather than a traditional venn diagram. Users can highlight a specific sections of the venn pie.

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,
  ...
)

## 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
)
```

Arguments

object	Venn object
subset	Character vector giving the subset users want to highlight.
top	number of subsets with largest to display (default: 31)
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.
color	Character vector giving the colors of the subsets.
revcolor	Character giving the color for the non-selected subsets.
any	Number to indicate selected subsets, such as 1 means any unique subsets, 2 means any subsets shared by two groups.
show.number	Boolean indicating whether to display the element numbers of the subsets or not (default: TRUE).
show.x	Boolean indicating whether to show subset labels outside the circle (default: TRUE).
sep	Character string used to separate the terms when concatenating group names into new column names (colnames).
log	Boolean indicating whether to transform the data in log scale .
base	Base value for log transformation.
percentage	Boolean indicating whether to display subset percentages (default: FALSE).
title	Optional plot title
interactive	Logical: create interactive plot?
...	Additional arguments

Value

A ggplot2 or plotly object

vennpie figure

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))
vennpie(res)
```

`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
)
```

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` *Perform statistical tests on set intersections*

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 rows and/or columns from the result data frame

Usage

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

Arguments

`x`	A Venn object
`i`	Row indices
`j`	Column indices or names

Value

A subset of the result data frame

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))

# Extract the first 5 rows
res[1:5, ]

# Extract a specific column
res[, "Subset"]
```

`$.Venn` *Extract a column from a Venn object*

Description

Extract a column from the result data frame

Usage

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

Arguments

`x`	A Venn object
`name`	Column name

Value

A vector with the column values

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))

# Extract the Subset column
res$Subset
```

Index