Package 'venny'

May 27, 2026

Title Venn Diagram

Version 0.0.3

Description Generate Venn plots, summary tables, and ellipse paths for polygon clipping.
Provides direct access to subsets of interest and offers flexible customization of Venn diagrams.
Summary tables are also available when Venn diagram visualization is not suitable.

License MIT + file LICENSE

URL https://github.com/P10911004-NPUST/venny

BugReports https://github.com/P10911004-NPUST/venny/issues

Encoding UTF-8

RoxygenNote 7.3.3

Imports ggplot2, polyclip

Depends R (>= 4.1)

LazyData true

Suggests tidyr, dplyr, ggtext, BiocManager, org.At.tair.db, DESeq2,
clusterProfiler, testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Joon-Keat Lai [aut, cre, cph] (ORCID:
<https://orcid.org/0000-0002-9840-5836>)

Maintainer Joon-Keat Lai <p10911004@gmail.com>

Repository CRAN

Date/Publication 2026-05-27 07:00:02 UTC

Contents

`bits_encoding` *Create a bits matrix*

Description

Produce a bits matrix for all possible combinations of the input sets.

Usage

```
bits_encoding(x, rownames = TRUE, sep = "")
```

Arguments

`x`	A character vector.
`rownames`	Logical (default: TRUE). Whether to show rownames.
`sep`	A character used to separate the group names (default is `""`). Only working when `rownames = TRUE`.

Value

A numeric matrix with values of 0 or 1.

Examples

```
bits_encoding(c("A", "B", "C", "D"))
```

`ellipse_line` *Control the ellipse textures*

Description

This is used to generate the ellipse line color and transparency parameters, for passing into the `venny()`s `ellipse.line` argument.

Usage

```
ellipse_line(
  linetype = "blank",
  linewidth = 0.5,
  color = c("#CC79A7", "#009E73", "#E69F00", "#56B4E9"),
  alpha = 0.5
)

ellipse_fill(
  color = c("#CC79A7", "#009E73", "#E69F00", "#56B4E9"),
  alpha = 0.2
)
```

Arguments

`linetype`	A character. The options are: blank, solid, dashed, dotted, dotdash, longdash, twodash.
`linewidth`	A number (default: 0.5).
`color`	A character vector (default: `c("#009E73", "#E69F00", "#CC79A7", "#56B4E9")`).
`alpha`	A number range from 0 to 1 (default: 0.5). Set the line transparency.

Value

A list contains "color" and "alpha" values.

See Also

`ellipse_fill()`

Examples

```
ellipse_line()
```

ellipse_position *Ellipse positions*

Description

This is an internal function used to generate the ellipse parameters that was used by the `generate_ellipse_path()` function. It returns a list encompassing 3 lists. Each list consists of the x0, y0, long-arm, short-arm, and angle default values for generating 2, 3, or 4 ellipses.

Usage

```
ellipse_position()
```

Value

A two-layer nested list.

Examples

```
ellipse_position()
```

fixed_length *Fixed Length Vector*

Description

Create a desired length of vector by trimming/extending the `x` vector.

Usage

```
fixed_length(x, len, fill_with = NULL)
```

Arguments

`x`	A vector.
`len`	Desired length.
`fill_with`	Element used to extend the vector length. If set to `NULL`, extend by itself (default: NULL).

Value

A vector

Examples

```
# Extend `x` to fulfill the `len` requirement
fixed_length(1:5, 7)
# Trim `x` to fulfill the `len` requirement
fixed_length(1:5, 3)
```

`generate_ellipse_path` *Generate Ellipse Path*

Description

Generate Ellipse Path

Usage

```
generate_ellipse_path(x0 = 0, y0 = 0, a = 2, b = 1, angle = 0, density = 200)
```

Arguments

`x0`	The coordinate x of the polygon center point (default: 0).
`y0`	The coordinate y of the polygon center point (default: 0).
`a`	Long arm (default: 2).
`b`	Short arm (default: 1).
`angle`	Rotation angle in degree (default: 0).
`density`	Greater amount of points yield smoother ellipse (default: 200).

Value

A data.frame with the point coordinates (x, y) to construct the polygon

Examples

```
library(ggplot2)
# Draw a circle
circle <- generate_ellipse_path(a = 1, b = 1)
ggplot(circle, aes(x, y)) +
    geom_polygon() +
    coord_fixed()
# Draw an ellipse
ellipse <- generate_ellipse_path(a = 2, b = 1, angle = 45)
ggplot(ellipse, aes(x, y)) +
    geom_polygon() +
    coord_fixed()
```

highlight *Draw polygons to highlight subsets*

Description

A handy wrapper for the `ggplot2::geom_polygon()` used to represent the result of set operations.

Usage

```
highlight(
  venn,
  setops,
  color = "black",
  linewidth = 1.2,
  linetype = "solid",
  fill = "black",
  alpha = 0.2,
  ...
)
```

Arguments

`venn`	Venn diagram produced from `venny::venny()`.
`setops`	The result of set operations.
`color`	Character (default: "transparent"). The polygon line color.
`linewidth`	Numeric (default: 1.5). The polygon linewidth.
`linetype`	Character (default: "solid"). The polygon linetype.
`fill`	Character (default: "black"). The polygon color.
`alpha`	Numeric (default: 0.4). The polygon transparency (0-1).
`...`	The other arguments passed into `ggplot2::geom_polygon`.

Value

A ggplot object.

Examples

```
data <- list(
    Set_A = c(10:100, 500:600),
    Set_B = c(5:150, 550:650),
    Set_C = c(80:180, 580:680),
    Set_D = c(120:220, 520:620)
)
out <- venny(data, detail = TRUE)
p0 <- out$venn
ep <- out$ellipse_path
```

```
res <- intersect(ep$Set_A, ep$Set_B, ep$Set_D)
highlight(p0, res)
```

`how_many_subsets` *How many subsets*

Description

Calculate the number of combinations that can be derived from the given character vector.

Usage

```
how_many_subsets(x, detail = FALSE)
```

Arguments

`x`	A character vector.
`detail`	Logical (default: FALSE). Whether to output the combinations.

Value

An integer value or a list.

Examples

```
how_many_subsets(c("qqa", "bnk", "sdf", "123"))
```

`LGL23` *RNA-seq data*

Description

A list of essential information for RNA-seq analysis and an example for demonstration.

Usage

```
LGL23
```

Format

An object of class `list` of length 4.

Details

- sample_info: the information for the sample IDs noted in the count matrix.
- GFD: A data.frame of gene functional description.
- count_matrix: A matrix of the gene expression count reads for each sample. The row names are gene IDs; the column names are sample IDs.
- DEGs: An example demonstrating differentially expressed genes across four conditions.

`setops` *Set operations*

Description

Perform intersection (`intersect()`), union (`union()`), and difference (`setdiff()`) for multiple `venny_ep` objects representing sets. They are handy wrappers of the `polyclip::polyclip()` function.

Usage

```
intersect(x, y, ...)

union(x, y, ...)

setdiff(x, y, ...)

## S3 method for class 'venny_setops'
intersect(x, y, ...)

## S3 method for class 'venny_setops'
union(x, y, ...)

## S3 method for class 'venny_setops'
setdiff(x, y, ...)
```

Arguments

`x`	A two-column matrix consists of the point coordinates (x, y) used to construct the ellipse. This is the reference-ellipse used to be clipped by the others.
`y`	A two-column matrix consists of the point coordinates (x, y) used to construct the ellipse. This is the object-ellipse used to clip the reference ellipse, i.e. `x`.
`...`	The other object-ellipses.

Details

These functions override the base versions to make them generic so that `venny` can provide methods to proceed the `venny_ep` class object. The default methods call the base versions.

Value

A list contains one or multiple two-column dataframe. Each dataframe is the point coordinates (x, y) used to construct a polygon.

Examples

```
data <- list(
    Set_A = c(10:100, 500:600),
    Set_B = c(5:150, 550:650),
    Set_C = c(80:180, 580:680),
    Set_D = c(120:220, 520:620)
)
out <- venny(data, detail = TRUE)
p0 <- out$venn
ep <- out$ellipse_path

#------------- Intersection -------------#
res <- intersect(ep$Set_A, ep$Set_B, ep$Set_D)
highlight(p0, res)

#----------------- Union ----------------#
res <- union(ep$Set_A, ep$Set_C, ep$Set_D)
highlight(p0, res)

#--------------- Difference -------------#
res <- setdiff(ep$Set_B, ep$Set_D, ep$Set_A, ep$Set_C)
highlight(p0, res)

#---------- Multiple operations ---------#
res <- union(ep$Set_A, ep$Set_B, ep$Set_C) |>
    intersect(ep$Set_D) |>
    setdiff(ep$Set_B)
highlight(p0, res)
```

---

set_label_default *Set labels*

Description

This is an internal function for `venny()` to automatically generate set labels, when the input list is unnamed.

Usage

```
set_label_default(n_sets)

subset_label_default(n_sets)
```

Arguments

n_sets	An integer.

Value

A list.

Examples

```
set_label_default(3)
```

---

set_label_font *Set label font*

Description

Generate a list of the available set label font parameters.

Usage

```
set_label_font(
  family = "sans",
  face = "bold",
  size = 5,
  color = c("#CC79A7", "#009E73", "#E69F00", "#56B4E9"),
  angle = NULL
)

subset_label_font(
  family = "sans",
  face = "bold.italic",
  size = 4,
  color = "black",
  angle = 0
)

subset_count_font(
  family = "sans",
  face = "plain",
  size = 4,
  color = "grey20",
  angle = 0
)

subset_percentage_font(
  family = "sans",
  face = "plain",
```

```
  size = 4,
  color = "grey20",
  angle = 0
)
```

Arguments

family	Character (default: "sans").
face	Character (default: "bold").
size	Numeric (default: 5).
color	Character (default: c("#009E73", "#E69F00", "#CC79A7", "#56B4E9")).
angle	Numeric | NULL (default: NULL).

Value

A list.

Examples

```
set_label_font()
```

---

set_label_position *Set label position*

Description

Generate the coordinates for the set labels. This is passed into the venny()'s set.label.position arguments.

Usage

```
set_label_position(hjust = 0, vjust = 0, show = NULL, hide = NULL)

subset_label_position(hjust = 0, vjust = 0, show = NULL, hide = NULL)

subset_count_position(hjust = 0, vjust = 0, show = NULL, hide = NULL)

subset_percentage_position(hjust = 0, vjust = 0, show = NULL, hide = NULL)
```

Arguments

hjust	A numeric vector (default: 0). Horizontal adjustment, adjust the x-axis coordinates of the set labels.
vjust	A numeric vector (default: 0). Vertical adjustment, adjust the y-axis coordinates of the set labels.

show	A character vector (default: NULL). Show only the specified set labels. By default, show all labels.
hide	A character vector (default: NULL). Do not show the specified set labels. By default, show all labels.

Value

A list contains 3 named list. Each named list contains the x, y coordinates of the set labels for different venn diagram (2, 3, or 4 ellipses), respectively.

Examples

```
set_label_position()
```

---

venny *Venn diagram*

Description

Venn diagram

Usage

```
venny(
  data,
  detail = FALSE,
  ellipse.line = ellipse_line(),
  ellipse.fill = ellipse_fill(),
  ellipse.density = 200L,
  set.label = TRUE,
  set.label.position = set_label_position(),
  set.label.font = set_label_font(),
  subset.label = TRUE,
  subset.label.position = subset_label_position(),
  subset.label.font = subset_label_font(),
  subset.count = TRUE,
  subset.count.position = subset_count_position(),
  subset.count.font = subset_count_font(),
  subset.percentage = TRUE,
  subset.percentage.position = subset_count_position(vjust = -0.3),
  subset.percentage.font = subset_percentage_font(),
  subset.percentage.rounding = 1L
)
```

Arguments

data	A list with 2 to 4 vectors
detail	Logical (default: TRUE). If TRUE, output a list contains the venn diagram and summary table. Otherwise, output only venn diagram.
ellipse.line	A list. See ellipse_line().
ellipse.fill	A list. See ellipse_fill().
ellipse.density	An integer (default: 200L). Higher value yield smoother ellipse.
set.label	Logical (default: TRUE). If TRUE, show the set labels.
set.label.position	A list. See set_label_position().
set.label.font	A list. See set_label_font().
subset.label	Logical (default: TRUE). If TRUE, show the subset labels. If a named list is provided, then the selected subset name will be renamed. For example, list(AB = "new_AB") will change the original subset AB name to "new_AB".
subset.label.position	A list. See subset_label_position().
subset.label.font	A list. See subset_label_font().
subset.count	Logical (default: TRUE). If TRUE, show the element counts for each subset.
subset.count.position	A list. See subset_count_position().
subset.count.font	A list. See subset_count_font().
subset.percentage	Logical (default: TRUE). If TRUE, show the percentages of the counts.
subset.percentage.position	A list. See subset_percentage_position().
subset.percentage.font	A list. See subset_percentage_font().
subset.percentage.rounding	An integer (default: 1L). How many decimal points.

Value

A venn diagram which is a ggplot object.

Examples

```
lst <- LGL23$DEGs
setLabelPosition <- set_label_position(hjust = c(0.2, -0.5, 0.5, -0.2))
venny(lst, set.label.position = setLabelPosition)
```

venn_summary *Create a* venny *summary data.frame*

Description

A data.frame consists of each combination of the input sets.

Usage

```
venn_summary(data = list(), show_elements = TRUE)
```

Arguments

data	A named list contains 2 ~ 26 sets of elements. For example, `list(set1 = 1:5, set2 = 3:7)`.
show_elements	Logical (default: FALSE). List out the elements of each zone in the dataframe.

Value

A data frame contains necessary information for subsequent inferences and plotting.

Examples

```
venn_summary(LGL23$DEGs)
```

Index