

# Package ‘hicVennDiagram’

May 6, 2024

**Title** Venn Diagram for genomic interaction data

**Version** 1.3.0

**Description** A package to generate high-resolution Venn and Upset plots for genomic interaction data from HiC, ChIA-PET, HiChIP, PLAC-Seq, Hi-TrAC, HiCAR and etc. The package generates plots specifically crafted to eliminate the deceptive visual representation caused by the counts method.

**License** GPL-3

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.1

**VignetteBuilder** knitr

**biocViews** DNA3DStructure, HiC, Visualization

**Depends** R (>= 4.3.0)

**Imports** GenomeInfoDb, GenomicRanges, IRanges, InteractionSet, rtracklayer, ggplot2, ComplexUpset, reshape2, eulerr, S4Vectors, methods, utils, htmlwidgets, svglite

**Suggests** BiocStyle, knitr, rmarkdown, testthat, ChIPpeakAnno, grid, TxDb.Hsapiens.UCSC.hg38.knownGene

**URL** <https://github.com/jianhong/hicVennDiagram>

**BugReports** <https://github.com/jianhong/hicVennDiagram/issues>

**git\_url** <https://git.bioconductor.org/packages/hicVennDiagram>

**git\_branch** devel

**git\_last\_commit** e922b4d

**git\_last\_commit\_date** 2024-04-30

**Repository** Bioconductor 3.20

**Date/Publication** 2024-05-06

**Author** Jianhong Ou [aut, cre] (<<https://orcid.org/0000-0002-8652-2488>>)

**Maintainer** Jianhong Ou <jianhong.ou@duke.edu>

Contents

browseVenn . . . . .	2
browseVenn-shiny . . . . .	3
createGIbackground . . . . .	3
gleamTest . . . . .	4
upsetPlot . . . . .	4
vennCount . . . . .	6
vennPlot . . . . .	6
vennTable-class . . . . .	7
<b>Index</b>	<b>9</b>

---

browseVenn	<i>Browse the venn plot</i>
------------	-----------------------------

---

Description

Brow the venn plot in a web browser to adjust the plot and export the result.

Usage

```
browseVenn(plot, width = NULL, height = NULL)
```

Arguments

plot	plots of <a href="#">vennPlot</a> or <a href="#">upsetPlot</a>
width	width of the figure
height	height of the figure

Value

An object of class `htmlwidget` that will intelligently print itself into HTML in a variety of contexts including the R console, within R Markdown documents, and within Shiny output bindings.

Examples

```
pd <- system.file("extdata", package = "hicVennDiagram", mustWork = TRUE)
fs <- dir(pd, pattern = ".bedpe", full.names = TRUE)
vc <- vennCount(fs)
p <- vennPlot(vc)
browseVenn(p)
```

---

browseVenn-shiny	<i>Shiny bindings for browseVenn</i>
------------------	--------------------------------------

---

**Description**

Output and render functions for using browseVenn within Shiny applications and interactive Rmd documents.

**Usage**

```
browseVennOutput(outputId, width = "100%", height = "400px")
```

```
renderbrowseVenn(expr, env = parent.frame(), quoted = FALSE)
```

**Arguments**

outputId	output variable to read from
width, height	Must be a valid CSS unit (like '100%', '400px', 'auto') or a number, which will be coerced to a string and have 'px' appended.
expr	An expression that generates a browseVenn
env	The environment in which to evaluate expr.
quoted	Is expr a quoted expression (with quote())? This is useful if you want to save an expression in a variable.

---

createGIbackground	<i>Create background by input GInteractions</i>
--------------------	---

---

**Description**

Create background based on the distance distribution of input GInteractions.

**Usage**

```
createGIbackground(gi, size = 2 * lengths(gi))
```

**Arguments**

gi	A vector of bedpe files or a list of genomic interaction data ( <a href="#">Pairs</a> or <a href="#">GInteractions</a> ).
size	The maximal size of the background

**Examples**

```
pd <- system.file("extdata", package = "hicVennDiagram", mustWork = TRUE)
fs <- dir(pd, pattern = ".bedpe", full.names = TRUE)[1]
set.seed(123)
# createGIbackground(fs)
```

---

gleamTest	<i>Perform GLEAM test</i>
-----------	---------------------------

---

### Description

Run Genomic Loops Enrichment Analysis Method test

### Usage

```
gleamTest(query, subject, background, method = c("binom", "hyper"), ...)
```

### Arguments

query, subject	A vector of bedpe files or a list of genomic interaction data ( <a href="#">Pairs</a> or <a href="#">GInteractions</a> ) or a list of GRanges object. 'subject' is optional if length of query > 1.
background	The test will restricted within the region. The background is the background of subject if subject is available. Otherwise, the background is the the background of second element of comparison group.
method	Distribution type for p-value.
...	parameters used by <a href="#">findOverlaps</a> .

### Examples

```
# example code
pd <- system.file("extdata", package = "hicVennDiagram", mustWork = TRUE)
fs <- dir(pd, pattern = ".bedpe", full.names = TRUE)
library(TxDb.Hsapiens.UCSC.hg38.knownGene)
## set.seed(123)
## background <- createGIbackground(fs)
## gleamTest(fs, background = background)
## gleamTest(fs, background = background, method = 'hyper')
grl <- GRangesList(exons=reduce(exons(TxDb.Hsapiens.UCSC.hg38.knownGene)),
  genes=reduce(genes(TxDb.Hsapiens.UCSC.hg38.knownGene)))
gleamTest(fs[seq_along(grl)], grl, background = grl[['exons']])
gleamTest(grl[c(2, 1)], grl, background = grl[['exons']])
gleamTest(grl, background = grl[['genes']])
```

---

upsetPlot	<i>UpSet plot for the Venn count table</i>
-----------	--

---

### Description

Plot the overlaps counts by ComplexUpset.

**Usage**

```
upsetPlot(
  vennTable,
  label_all = list(na.rm = TRUE, color = "gray30", alpha = 0.7, label.padding = unit(0.1,
    "lines")),
  coln_prefix = NULL,
  ...
)
```

**Arguments**

vennTable	An vennTable object, the first element in the output of <a href="#">vennCount</a> .
label_all	A list of parameters used by <a href="#">geom_label</a> for text labels of counts for each group. If it set to FALSE or length of the list is zero, the labels will be ignored.
coln_prefix	The prefix to be removed for colnumn names of vennTable.
...	Parameters could be passed to <a href="#">upset</a> except data and intersect.

**Value**

A ggplot object.

**Examples**

```
pd <- system.file("extdata", package = "hicVennDiagram", mustWork = TRUE)
fs <- dir(pd, pattern = ".bedpe", full.names = TRUE)
vc <- vennCount(fs)
upset_themes_fix <- lapply(ComplexUpset::upset_themes, function(.ele){
  .ele[names(.ele) %in% names(formals(ggplot2::theme))]
})
upsetPlot(vc, theme = upset_themes_fix)
## change the font size of lables and numbers
themes <- ComplexUpset::upset_modify_themes(
  ## get help by vignette('Examples_R', package = 'ComplexUpset')
  list('intersections_matrix'=
    ggplot2::theme(axis.text.y=ggplot2::element_text(size=24)))
)
themes <- lapply(themes, function(.ele){
  .ele[names(.ele) %in% names(formals(ggplot2::theme))]
})
upsetPlot(vc, label_all=list(
  na.rm = TRUE,
  color = 'gray30',
  alpha = .7,
  label.padding = grid::unit(0.1, "lines"),
  size = 5
), themes = themes)
```

---

vennCount	<i>Construct intersections of sets</i>
-----------	--

---

### Description

Given a collection of bedpe files or a list of genomic interaction data, `vennCount` will compute all possible combinations of interactions and return an object of class `vennTable`, storing the combinations as well as the number of elements in each intersection.

### Usage

```
vennCount(gi, FUN = min, ...)
```

### Arguments

<code>gi</code>	A vector of bedpe files or a list of genomic interaction data ( <a href="#">Pairs</a> or <a href="#">GInteractions</a> )
<code>FUN</code>	Function to summarize the overlapping number.
<code>...</code>	parameters used by <a href="#">findOverlaps</a>

### Value

An object of `vennTable`

### Examples

```
pd <- system.file("extdata", package = "hicVennDiagram", mustWork = TRUE)
fs <- dir(pd, pattern = ".bedpe", full.names = TRUE)
vc <- vennCount(fs)
```

---

vennPlot	<i>Venn diagram for the Venn count table</i>
----------	--

---

### Description

Plot the overlaps counts by euler.

### Usage

```
vennPlot(vennTable, shape = "circle", ...)
```

**Arguments**

vennTable	An vennTable object, the first element in the output of <a href="#">vennCount</a> .
shape	Geometric shape used in the diagram used by <a href="#">euler</a> .
...	parameters to update fills and edges with and thereby a shortcut to set these parameters <a href="#">plot.euler</a> .

**Value**

A grid object.

**Examples**

```
pd <- system.file("extdata", package = "hicVennDiagram", mustWork = TRUE)
fs <- dir(pd, pattern = ".bedpe", full.names = TRUE)
vc <- vennCount(fs)
vennPlot(vc)
## change the font size of venn plot lables and numbers,
## both cex or fontsize should work
vennPlot(vc, quantities=list(fontsize=24), labels=list(cex=1.5))
```

---

vennTable-class	<i>Class "vennTable"</i>
-----------------	--------------------------

---

**Description**

An object of class "vennTable" represents Venn counts.

**Usage**

```
vennTable(...)

## S4 method for signature 'vennTable'
x$name

## S4 replacement method for signature 'vennTable'
x$name <- value

## S4 method for signature 'vennTable,ANY,ANY'
x[[i]]

## S4 replacement method for signature 'vennTable,ANY,ANY'
x[[i]] <- value

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

**Arguments**

<code>...</code>	Each argument in <code>...</code> becomes an slot in the new <code>vennTable</code> .
<code>x</code>	an object of <code>vennTable</code>
<code>name</code>	slot name of <code>vennTable</code>
<code>value</code>	values to assign
<code>i</code>	slot name of <code>vennTable</code>
<code>object</code>	an object of <code>vennTable</code> .

**Value**

An object of `vennTable`.

**Slots**

`combinations` A logical "matrix", specify the combinations.  
`counts` A "numeric" vector, the overall counts number for each combination.  
`vennCounts` A "matrix" object, specify the counts number for each sample in the combination.  
`overlapList` "list", overlapping list of the genomic interactions.

**Examples**

```
vt <- vennTable()
```



# Index

`[[`, `vennTable`, `ANY`, `ANY-method`  
    (`vennTable-class`), [7](#)  
`[[<-`, `vennTable`, `ANY`, `ANY-method`  
    (`vennTable-class`), [7](#)  
`$`, `vennTable-method` (`vennTable-class`), [7](#)  
`$<-`, `vennTable-method` (`vennTable-class`),  
    [7](#)  
`'[[<-'`, `vennTable-method`  
    (`vennTable-class`), [7](#)  
`'[['`, `vennTable-method`  
    (`vennTable-class`), [7](#)  
  
`browseVenn`, [2](#)  
`browseVenn-shiny`, [3](#)  
`browseVennOutput` (`browseVenn-shiny`), [3](#)  
  
`createGIbackground`, [3](#)  
  
`euler`, [7](#)  
  
`findOverlaps`, [4](#), [6](#)  
  
`geom_label`, [5](#)  
`GInteractions`, [3](#), [4](#), [6](#)  
`gleamTest`, [4](#)  
  
`Pairs`, [3](#), [4](#), [6](#)  
`plot.euler`, [7](#)  
  
`renderbrowseVenn` (`browseVenn-shiny`), [3](#)  
  
`show`, `vennTable-method`  
    (`vennTable-class`), [7](#)  
  
`upset`, [5](#)  
`upsetPlot`, [2](#), [4](#)  
  
`vennCount`, [5](#), [6](#), [7](#)  
`vennPlot`, [2](#), [6](#)  
`vennTable`, [6](#)  
`vennTable` (`vennTable-class`), [7](#)  
`vennTable-class`, [7](#)