Package 'RVenn'

July 21, 2025

Type Package

Title Set Operations for Many Sets

Version 1.1.0

Maintainer Turgut Yigit Akyol <tyakyol@gmail.com>

Description Set operations for many sets. The base functions for set operations
in R can be used for only two sets. This package uses 'purr' to find the
union, intersection and difference of three or more sets. This package also
provides functions for pairwise set operations among several sets. Further,
based on 'ggplot2' and 'ggforce', a Venn diagram can be drawn for two or
three sets. For bigger data sets, a clustered heatmap showing presence/absence
of the elements of the sets can be drawn based on the 'pheatmap' package.
Finally, enrichment test can be applied to two sets whether an overlap is
statistically significant or not.

License GPL-3

Encoding UTF-8

LazyData true

Imports ggforce (>= 0.2.1), ggplot2 (>= 3.0.0), magrittr (>= 1.5),
purrr (>= 0.2.5), methods (>= 3.5.1), rlang (>= 0.2.2), vegan
(>= 2.5.2), pheatmap (>= 1.0.10)

RoxygenNote 6.1.1

Collate 'Construct.R' 'Plot.R' 'SetOperations.R' 'RVenn.R'
'Enrichment.R' 'Pairs.R' 'Heatmap.R'

Suggests testthat, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Turgut Yigit Akyol [aut, cre]

Repository CRAN

Date/Publication 2019-07-18 21:40:02 UTC

Contents

---

discern,Venn-method   *Set difference.*

Description

discern returns the difference between two group of sets selected from a Venn object. If multiple sets are chosen for the slices, union of those sets will be used.

Usage

```
## S4 method for signature 'Venn'
discern(venn, slice1, slice2 = "all")
```

Arguments

venn	(Required) A Venn object.
slice1	(Required) The name or the index of the set of interest. Multiple sets can be selected.
slice2	(Optional) The name or the index of the set of interest. Multiple sets can be selected. Default is all the sets except the sets of slice1.

Value

A vector showing the difference between slice1 and slice2.

Examples

```
venn = Venn(list(letters[1:10], letters[3:12], letters[6:15]))
discern(venn, slice1 = 1)
discern(venn, slice1 = c(1, 2), slice2 = 3)
```

---

discern_pairs   *Pairwise difference of many sets.*

Description

discern_pairs returns the pairwise differences of the sets in a Venn object.

Usage

```
## S4 method for signature 'Venn'
discern_pairs(venn, slice = "all")
```

Arguments

venn   (Required) A Venn object.

slice   (Optional) The names or the indices of sets of interest. Default is "all", meaning the pairwise differences will be calculated for all the sets.

Value

A list showing the pairwise differences of the sets.

Examples

```
venn = Venn(list(letters[1:10], letters[3:12],
                 letters[6:15], letters[9:18]))
discern_pairs(venn)
discern_pairs(venn, slice = 1:3)
```

---

enrichment_test   *Perform an enrichment test.*

Description

Calculate the p-value of occurrence of an overlap between two sets by chance.

Usage

```
## S4 method for signature 'Venn'
enrichment_test(venn, set1, set2, univ = "all",
  n = 10000, seed = 42)
```

Arguments

venn	(Required) A Venn object.
set1	(Required) The name or the index of the set of interest.
set2	(Required) The name or the index of the set to be checked whether enriched in set1.
univ	(Optional) Population size. Default is "all", implying the union of all the sets in the Venn object will be used. Another set as the whole population can be assigned as well.
n	(Optional) Number of randomly generated sets. Default is 10,000 and minimum is 1,000.
seed	(Optional) An integer passed to set.seed function. It is used to fix a seed for reproducibly random number generation. Default is 42.

Details

This type of analysis can also be performed by hypergeometric test or Fisher's exact test. Here, the approach is similar to that described in (Austin et al., 2016). Briefly, the test is based on randomly generation of sets with equal size to set1 from the background (universal) set. After creating n (default is 10,000) random sets, the overlap between these and set2 is calculated to make a null distribution. When this distribution is true, the probability of seeing an overlap at least as extreme as what was observed (overlap between set1 and set2) will be returned as the p-value.

Value

Returns a list containing the probability (Significance) of occurrence of an overlap between two sets by chance and the number of occurrences (Overlap_Counts) in randomly generated sets.

Examples

```
set1 = c(1:20, letters[1:10])
set2 = letters[-26]
univ = unique(c(set1, set2, 21:200))
venn = Venn(list(set1, set2, univ))
e = enrichment_test(venn, 1, 2)
e$Significance
```

---

ggvenn,Venn-method   *Draw the Venn diagram.*

Description

Draw the Venn diagram for 2 or 3 sets.

Usage

```
## S4 method for signature 'Venn'
ggvenn(venn = "Venn", slice = "all",
  fill = c("gold", "dodgerblue3", "deeppink"), alpha = 0.5,
  thickness = 1.5)
```

Arguments

venn	(Required) A Venn object.
slice	(Optional) The names or the indices of the sets of interest. Default is "all", which is for the cases the Venn object only contains 2 or 3 sets. If you have 4 or more sets, this argument is required.
fill	(Optional) Fill color of the sets.
alpha	(Optional) Opacity of the fill colors. Default is 0.5 in the range of (0, 0.5).
thickness	(Optional) Stroke size of the sets.

Details

This function is based on the packages 'ggplot2' and 'ggforce.' It has been designed for 2 or 3 sets because Venn diagrams are terrible for showing the interactions of 4 or more sets. If you need to visualize such interactions, consider using setmap.

Value

The function returns the plot in ggplot2 style.

Examples

```
venn = Venn(list(letters[1:10], letters[3:12], letters[6:15]))
ggvenn(venn)
ggvenn(venn, slice = c(1, 2), thickness = 0, alpha = 0.3)
```

---

overlap,Venn-method   *Intersection of many sets.*

Description

overlap returns the same elements of the sets in a Venn object.

Usage

```
## S4 method for signature 'Venn'
overlap(venn, slice = "all")
```

Arguments

venn	(Required) A Venn object.
slice	(Optional) The names or the indices of sets of interest. Default is "all", meaning the intersection will be calculated for all the sets.

Value

A vector showing the intersection of the sets.

Examples

```
venn = Venn(list(letters[1:10], letters[3:12], letters[6:15]))
overlap(venn)
overlap(venn, slice = c(1, 2))
```

---

overlap_pairs	*Pairwise intersections of many sets.*

Description

overlap_pairs returns the pairwise intersections of the sets in a Venn object.

Usage

```
## S4 method for signature 'Venn'
overlap_pairs(venn, slice = "all")
```

Arguments

venn	(Required) A Venn object.
slice	(Optional) The names or the indices of sets of interest. Default is "all", meaning the pairwise intersections will be calculated for all the sets.

Value

A list showing the pairwise intersections of the sets.

Examples

```
venn = Venn(list(letters[1:10], letters[3:12],
                 letters[6:15], letters[9:18]))
overlap_pairs(venn)
overlap_pairs(venn, slice = 1:3)
```

RVenn   *RVenn: A package for set operations for many sets.*

Description

Set operations for many sets. The base functions for set operations in R can be used for only two
sets. This package uses 'purr' to find the union, intersection and difference of three or more sets.
This package also provides functions for pairwise set operations among several sets. Further, based
on 'ggplot2' and 'ggforce', a Venn diagram can be drawn for two or three sets. For bigger data sets,
a clustered heatmap showing presence/absence of the elements of the sets can be drawn based on
the 'pheatmap' package. Finally, enrichment test can be applied to two sets whether an overlap is
statistically significant or not.

---

setmap   *Draw a clustered heatmap showing presence/absence of the elements.*

Description

This function is based on the package 'pheatmap'. [ggvenn](#) function is useful for 2-3 sets, if you
need to show interactions between many sets, you can show the presence/absence of the elements
among all the sets and cluster both the sets and the elements based on Jaccard distances.

Usage

```
## S4 method for signature 'Venn'
setmap(venn, slice = "all", element_clustering = TRUE,
  set_clustering = TRUE, method = "average", legend = TRUE,
  title = NA, element_fontsize = 10, set_fontsize = 10)
```

Arguments

venn	(Required) A Venn object.
slice	(Optional) The names or the indices of sets of interest. Default is "all", meaning the union will be calculated for all the sets.
element_clustering	
	(Optional) Boolean values determining if elements should be clustered.
set_clustering	(Optional) Boolean values determining if sets should be clustered.
method	(Optional) Clustering method used. Accepts the same values as [hclust](#).
legend	(Optional) Boolean values determining if the legend should be drawn.
title	(Optional) Title of the heatmap.
element_fontsize	
	(Optional) Font size of the elements.
set_fontsize	(Optional) Font size of the sets.

Value

Presence/absence heatmap of the sets.

Examples

```
venn = Venn(list(letters[1:10], letters[3:12], letters[6:15], letters[9:16],
letters[15:25], letters[12:20]))
setmap(venn)
setmap(venn, slice = 1:4, element_clustering = FALSE, set_clustering = FALSE)
```

---

unite,Venn-method   *Union of many sets.*

Description

unite returns the union of the sets in a Venn object.

Usage

```
## S4 method for signature 'Venn'
unite(venn, slice = "all")
```

Arguments

venn   (Required) A Venn object.

slice   (Optional) The names or the indices of sets of interest. Default is "all", meaning
                the union will be calculated for all the sets.

Value

A vector showing the union of the sets.

Examples

```
venn = Venn(list(letters[1:10], letters[3:12], letters[6:15]))
unite(venn)
unite(venn, slice = c(1, 2))
```

---

unite_pairs   *Pairwise unions of many sets.*

Description

unite_pairs returns the pairwise unions of the sets in a Venn object.

Usage

```
## S4 method for signature 'Venn'
unite_pairs(venn, slice = "all")
```

Arguments

venn   (Required) A Venn object.

slice   (Optional) The names or the indices of sets of interest. Default is "all", meaning
                the pairwise intersections will be calculated for all the sets.

Value

A list showing the pairwise unions of the sets.

Examples

```
venn = Venn(list(letters[1:10], letters[3:12],
                 letters[6:15], letters[9:18]))
unite_pairs(venn)
unite_pairs(venn, slice = 1:3)
```

Venn,ANY-method   *Build a* Venn *object.*

Description

Venn builds a Venn object from a list.

Usage

```
## S4 method for signature 'ANY'
Venn(sets)
```

Arguments

sets   (Required) A list containing vectors in the same class. If a vector contains du-
                plicates they will be discarded. If the list doesn't have names the sets will be
                named as "Set_1", "Set_2", "Set_3" and so on.

Value

A Venn object.

Examples

```
venn = Venn(list(letters[1:10], letters[3:12], letters[6:15]))
print(venn)
```

Venn-class   *An S4 class to represent multiple sets.*

Description

An S4 class to represent multiple sets.

Slots

sets A `list` object containing vectors in the same type.

names The names of the `sets` if it has names. If the `list` doesn't have names, the sets will be named as "Set_1", "Set_2", "Set_3" and so on.

Index