Package 'SuperExactTest'

June 21, 2019

Type Package

Title Exact Test and Visualization of Multi-Set Intersections

Version 1.0.7

Date 2019-06-14

Author Minghui Wang, Yongzhong Zhao and Bin Zhang

Maintainer Minghui Wang <minghui.wang@mssm.edu>

Contact Minghui Wang <minghui.wang@mssm.edu>, Bin Zhang <bin.zhang@mssm.edu>

Description
Identification of sets of objects with shared features is a common operation in all disciplines. Analysis of intersections among multiple sets is fundamental for in-depth understanding of their complex relationships. This package implements a theoretical framework for efficient computation of statistical distributions of multi-set intersections based upon combinatorial theory, and provides multiple scalable techniques for visualizing the intersection statistics. The statistical algorithm behind this package was published in Wang et al. (2015) <doi:10.1038/srep16923>.

Depends grid (>= 3.1.0), methods, R (>= 3.1.0)

Suggests knitr, rmarkdown

VignetteBuilder knitr

URL https://github.com/mw201608/SuperExactTest/

BugReports https://github.com/mw201608/SuperExactTest/issues

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-06-21 19:10:03 UTC

R topics documented:

`Cancer` *Cancer Census Dataset*

Description

This example dataset contains a list of seven cancer predisposition gene sets.

Usage

```
Cancer
```

Details

The seven cancer predisposition gene sets are:

- NRG (Rahman, N. Realizing the promise of cancer predisposition genes. Nature 2014, 505:302-308);
- NBG (Tamborero, D. et al. Comprehensive identification of mutational cancer driver genes across 12 tumor types. Scientific reports 2013, 3:2650);
- LDG (Kandoth, C. et al. Mutational landscape and significance across 12 major cancer types. Nature 2013, 502:333-339);
- GGG (Lawrence, M. S. et al. Discovery and saturation analysis of cancer genes across 21 tumour types. Nature 2014, 505:495-501);
- ELG (Garraway, L. A. & Lander, E. S. Lessons from the cancer genome. Cell 2013, 153:17-37);
- CCG (Futreal, P. A. et al. A census of human cancer genes. Nature reviews. Cancer 2004, 4:177-183);
- BVG (Vogelstein, B. et al. Cancer genome landscapes. Science 2013, 339:1546-1558).

References

Minghui Wang, Yongzhong Zhao, and Bin Zhang (2015). Efficient Test and Visualization of Multi-Set Intersections. *Scientific Reports* 5: 16923.

See Also

supertest

`cis.eqtls` *cis-eQTLs*

Description

This example dataset contains a list of cis-eQTL genes.

Details

A list is included in this dataset: `cis.eqtls`, which contains four sets of cis-eQTL genes published by Gibbs et al (PLOS Genetics 2010, 6:e1000952) as deposited in the eQTL Browser (http://www.ncbi.nlm.nih.gov/projects/g The four sets of cis-eQTL genes were detected in four different brain regions from Gibbs: brain cerebellum (CB), brain frontal cortex region (FC), brain temporal cortex region (TC), and brain pons region (PONS) respectively.

See Also

supertest

`cpsets` *Multi-Set Intersection Probability*

Description

Density and distribution function of multi-set intersection test.

Usage

```
dpsets(x,L,n,log.p =FALSE)
cpsets(x,L,n,lower.tail=TRUE,log.p=FALSE,
       simulation.p.value=FALSE,number.simulations=1000000)
```

Arguments

`x`	integer, number of elements overlap among all sets.
`L`	vector, set sizes.
`n`	integer, background population size.
`lower.tail`	logical; if TRUE, probability is `P[overlap <= x]`, otherwise, `P[overlap > x]`.
`log.p`	logical; if TRUE, probability p is given as `log(p)`.
`simulation.p.value`	logical; if TRUE, probability p is computed from simulation.
`number.simulations`	integer; number of simulations.

Value

dpsets gives the density and cpsets gives the distribution function.

Author(s)

Minghui Wang <minghui.wang@mssm.edu>

References

Minghui Wang, Yongzhong Zhao, and Bin Zhang (2015). Efficient Test and Visualization of Multi-Set Intersections. *Scientific Reports* 5: 16923.

See Also

supertest, MSET

Examples

```
## Not run:
#set up fake data
n=500; A=260; B=320; C=430; D=300; x=170
(d=dpsets(x,c(A,B,C,D),n))
(p=cpsets(x,c(A,B,C,D),n,lower.tail=FALSE))

## End(Not run)
```

---

deBarcode *Decrypt Barcode*

Description

Decrypt barcode information.

Usage

```
deBarcode(barcode, setnames, collapse=' & ')
```

Arguments

barcode	a vector of character strings, encoding the intersection combination.
setnames	set names.
collapse	an optional character string to separate the results. See paste.

Details

barcode are character strings of '0' and '1', indicating absence or presence of each set in a intersection combination.

Value

A vector.

Author(s)

Minghui Wang <minghui.wang@mssm.edu>

Examples

```
deBarcode(c('01011','10100'), c('S1','S2','S3','S4','S5'))
```

`GWAS` *GAWS Catalog Dataset*

Description

This example dataset contains a list of gene sets associated with six types of clinical traits curated in the GWAS Catalog.

Usage

```
GWAS
```

Details

The six clinical traits are:

- NEU (Bipolar disorder and schizophrenia, Schizophrenia, Major depressive disorder, Alzheimer's disease, Parkinson's disease, Cognitive performance, Bipolar disorder);
- INF (Crohn's disease, Ulcerative colitis, Inflammatory bowel disease, Rheumatoid arthritis, Multiple sclerosis, Systemic lupus erythematosus);
- CVD (Type 2 diabetes, Coronary heart disease, Blood pressure, total Cholesterol, HDL cholesterol, Triglycerides);
- HT (height);
- IgG (IgG glycosylation);
- OB (obesity, obesity related traits).

References

Minghui Wang, Yongzhong Zhao, and Bin Zhang (2015). Efficient Test and Visualization of Multi-Set Intersections. *Scientific Reports* 5: 16923.

See Also

`supertest`

`intersect` *Set Operations*

Description

Performs set union and intersection on multiple input vectors.

Usage

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

Arguments

`x, y, ...`	vectors (of the same mode) containing a sequence of items (conceptually) with no duplicated values.

Details

These functions extend the the same functions in the `base` package to handle more than two input vectors.

Value

A vector of the same mode as x or y for intersect, and of a common mode for union.

Author(s)

Minghui Wang <minghui.wang@mssm.edu>, Bin Zhang <bin.zhang@mssm.edu>

References

Minghui Wang, Yongzhong Zhao, and Bin Zhang (2015). Efficient Test and Visualization of Multi-Set Intersections. *Scientific Reports* 5: 16923.

Examples

```
##not run##
```

intersectElements — *Find Intersection Membership*

Description

Find intersections and assign element to intersection combinations.

Usage

```
intersectElements(x, mutual.exclusive=TRUE)
```

Arguments

`x`	list; a collection of sets.
`mutual.exclusive`	logical; see `Details`.

Details

See example below for the use of `mutual.exclusive`.

Value

A data.frame with two columns:

`Entry`	set elements.
`barcode`	intersection combination that each entry belongs to.

Author(s)

Minghui Wang <minghui.wang@mssm.edu>

Examples

```
set.seed(123)
sets=list(S1=sample(letters,10), S2=sample(letters,5), S3=sample(letters,7))
intersectElements(sets,mutual.exclusive=TRUE)
intersectElements(sets,mutual.exclusive=FALSE)
```

jaccard — *Calculate Jaccard Index*

Description

This function calculates Jaccard indices between pairs of sets.

Usage

```
jaccard(x)
```

Arguments

x	list, a collect of sets.

Value

A matrix of pairwise Jaccard indices.

Author(s)

Minghui Wang <minghui.wang@mssm.edu>

Examples

```
## Not run:
#set up fake data
x=list(S1=letters[1:20], S2=letters[10:26], S3=sample(letters,10), S4=sample(letters,10))
jaccard(x)

## End(Not run)
```

MSET — *Exact Test of Multi-Set Intersection*

Description

Calculate FE and significance of intersection among multiple sets.

Usage

```
MSET(x,n,lower.tail=TRUE,log.p=FALSE)
```

Arguments

`x`	list; a collection of sets.
`n`	integer; background population size.
`lower.tail`	logical; if TRUE, probability is `P[overlap < m]`, otherwise, `P[overlap >= m]`, where `m` is the number of elements overlap between all sets.
`log.p`	logical; if TRUE, probability p is given as log(p).

Details

This function implements an efficient statistical test for multi-set intersections. The algorithm behind this function was described in Wang et al 2015.

Value

A list with the following elements:

`intersects`	a vector of intersect items.
`FE`	fold enrichment of the intersection.
`p.value`	one-tail probability of observing equal to or larger than the number of intersect items.

Author(s)

Minghui Wang <minghui.wang@mssm.edu>, Bin Zhang <bin.zhang@mssm.edu>

References

Minghui Wang, Yongzhong Zhao, and Bin Zhang (2015). Efficient Test and Visualization of Multi-Set Intersections. *Scientific Reports* 5: 16923.

See Also

`supertest`, `cpsets`, `dpsets`

Examples

```
## Not run:
#set up fake data
x=list(S1=letters[1:20], S2=letters[10:26], S3=sample(letters,10), S4=sample(letters,10))
MSET(x, 26, FALSE)

## End(Not run)
```

msets *Class to Contain Multi-Set Intersections*

Description

This object contains data regarding the intersections between multiple sets. This object is usually created by the `supertest` function.

Details

Intersection combination is denoted by a `barcode` string of '0' and '1', where a value of '1' in the `i`th position of the string indicates that the intersection is involved with the `i`th set, 0 otherwise. E.g., string '000101' indicates that the intersection is an overlap between the 4th and 6th sets. Function `deBarcode` can be used to decrypt the barcode. Generic `summary` and `plot` functions can be applied to extract and visualize the results.

Value

`x`	a list of sets from input.
`set.names`	names of the sets. If the input sets do not have names, they will be automatically named as SetX where X is an integer from 1 to the total number of sets.
`set.sizes`	a vector of set sizes.
`n`	background population size.
`overlap.sizes`	a named vector of intersection sizes. Each intersection component is named by a barcoded character string of '0' and '1'. See `Details` for barcode.
`overlap.expected`	a named vector of expected intersection sizes when item `n` is available.
`P.value`	a vector of p values for the intersections when item `n` is available.

Author(s)

Minghui Wang <minghui.wang@mssm.edu>, Bin Zhang <bin.zhang@mssm.edu>

References

Minghui Wang, Yongzhong Zhao, and Bin Zhang (2015). Efficient Test and Visualization of Multi-Set Intersections. *Scientific Reports* 5: 16923.

See Also

`supertest`, `summary.msets`, `plot.msets`, `deBarcode` plot.msets *Draw Multi-Set Intersections*

Description

This function draws intersections among multiple sets.

Usage

```
## S3 method for class 'msets'
plot(x, Layout=c('circular','landscape'), degree=NULL,
keep.empty.intersections=TRUE,
sort.by=c('set','size','degree','p-value'),
min.intersection.size=0, max.intersection.size=Inf,
ylim=NULL, log.scale=FALSE, yfrac=0.8, margin=NULL,
color.scale.pos=c(0.85, 0.9), legend.pos=c(0.85,0.25),
legend.col=2, legend.text.cex=1, color.scale.cex=1,
color.scale.title=expression(paste(-Log[10],'(',italic(P),')')),
color.on='#2EFE64', color.off='#EEEEEE',
show.overlap.size=TRUE, show.set.size=TRUE,
overlap.size.cex=0.9, track.area.range=0.3, bar.area.range=0.2,
new.gridPage=TRUE, minMinusLog10PValue=0,
maxMinusLog10PValue=NULL, show.elements=FALSE, ...)
```

Arguments

`x`	a `msets` object.
`Layout`	layout for plotting.
`degree`	a vector of intersection degrees for plotting. E.g., when `degree=c(2:3)`, only those intersections involving two or three sets will be plotted. By default, `degree=NULL`, all possible intersections are plotted.
`keep.empty.intersections`	logical; if `FALSE`, empty intersection(s) will be discarded to save plotting space.
`min.intersection.size`	Minimum size of an intersection to be plotted.
`max.intersection.size`	Maximum size of an intersection to be plotted.
`sort.by`	how to sort intersections. It can be either one of the key words "`set`", "`size`", "`degree`", and "`p-value`", or a vector of custom ordered set combination strings.
`ylim`	the limits c(y1, y2) of plotting overlap size.
`log.scale`	logical; whether to plot with log transformed intersection sizes.
`yfrac`	numeric; the fraction (0 to 1) of canvas used for plotting bars. Only used for `landscape` Layout.

`margin`	numeric; a vector of 4 numeric values specifying the margins (bottom, left, top, & right) in unit of "lines". Default c(1,1,1,1)+0.1 for `circular` Layout and c(0.5,5,1.5,2)+0.1 for `landscape` Layout. Increase margin if the plot area is cropped.
`color.scale.pos`	numeric; x and y coordinates (0 to 1) for packing the color scale guide. It could be a keyword `"topright"` or `"topleft"` in the `landscape` layout, and one of `"topright"`, `"topleft"`, `"bottomright"` and `"bottomleft"` in the `circular` layout.
`legend.pos`	numeric; x and y coordinates (0 to 1) for packing the legend in the `circular` layout. It could be one of the keywords "bottomright", "bottomleft", "topleft" and "topright".
`legend.col`	integer; number of columns of the legend in the `circular` layout.
`legend.text.cex`	numeric; specifying the amount by which legend text should be magnified relative to the default.
`color.scale.cex`	numeric; specifying the amount by which color scale text should be magnified relative to the default.
`color.scale.title`	character or expression; a title for the color scale guide.
`color.on`	color code; specifying the color for set(s) which are `"present"` for an intersection.
`color.off`	color code; specifying the color for set(s) which are "absent" for an intersection.
`show.overlap.size`	logical; whether to show overlap size.
`show.set.size`	color code; whether to show set size in the `landscape` layout.
`overlap.size.cex`	numeric; specifying the amount by which overlap size text should be magnified relative to the default.
`track.area.range`	the magnitude of track area from origin in the `circular` layout.
`bar.area.range`	the magnitude of bar area from edge of the track area in the `circular` layout. The sum of `track.area.range` and `track.area.range` should not be larger than 0.5.
`new.gridPage`	logic; whether to start a new grid page. Set `FALSE` to allow for customized arrangement of the grid layout.
`minMinusLog10PValue`	numeric; minimum minus log10 P value for capping the scale of color map. Default 0.
`maxMinusLog10PValue`	numeric; maximum minus log10 P value for capping the scale of color map. Default maximum from the data.
`show.elements`	logical; whether to show the intersection elements on top of the bars with the `landscape` layout. See `Details` for more control options `elements.*`.
`...`	additional arguments for the plot function. See `Details`.

Details

The plot canvas has coordinates 0~1 for both x and y axes. Additional optional plot parameters include:

- `ylab`, a chracter string of y axis label.
- `heatmapColor`, a vector of customized heat colors.
- `show.expected.overlap`, whether to show expcted overlap in `landscape` Layout. Default 'FALSE'.
- `expected.overlap.style`, one of c("hatchedBox","horizBar","box"). Default 'hatchedBox'.
- `expected.overlap.lwd`, line width for expected.overlap "horizBar" and "box". Default 2.
- `color.expected.overlap`, color for showing expcted overlap in hatched lines. Default 'grey'.
- `alpha.expected.overlap`, alpha channel for transparency for showing expcted overlap hatched lines. Default 1 (normalized to the range 0 to 1).
- `cex`, scale of text font size.
- `cex.lab`, scale of axis label text font size.
- `show.track.id`, logic, whether to show the track id in the `circular` layout. Default `TRUE`.
- `phantom.tracks`, number of phantom tracks in the middle in the `circular` layout. Default 2.
- `gap.within.track`, ratio of gap width over block width on the same track. Default 0.1.
- `gap.between.track`, ratio of gap width over track width. Default 0.1.
- `bar.split`, a vector of two values specifying a continuous range that will be cropped in the y axis with the `landscape` layout.
- `elements.list`, a data.frame or matrix such as the one generated by the `summary` function from a `msets` object, with row names matching the barcodes of intersection combinations and at least one column named "Elements" listing the elements to be displayed (the elements should be concatenated by separator ", ").
- `elements.cex`, numeric; specifying the amount by which intersection element text should be magnified. Default 0.9.
- `elements.rot`, numeric; the angle to rotate the text of intersection elements. Default 45.
- `elements.col`, colour for intersection element text. Default black.
- `elements.maximum`, maximum number of elements to show.
- `intersection.size.rotate`, logic, whether to rotate the text of intersection size.
- `flip.vertical`, logic, whether to flip the bars to downwards in `landscape` Layout. Default 'FALSE'.

Value

No return.

Author(s)

Minghui Wang <minghui.wang@mssm.edu>, Bin Zhang <bin.zhang@mssm.edu>

References

Minghui Wang, Yongzhong Zhao, and Bin Zhang (2015). Efficient Test and Visualization of Multi-Set Intersections. *Scientific Reports* 5: 16923.

See Also

msets

Examples

```
## Not run:
#set up fake data
x=list(S1=letters[1:20], S2=letters[10:26], S3=sample(letters,10), S4=sample(letters,10))
obj=supertest(x,n=26)
plot(obj)

## End(Not run)
```

---

summary.msets   *Summarize an msets Object*

Description

This function outputs summary statistics of a msets object.

Usage

```
## S3 method for class 'msets'
summary(object, degree=NULL, ...)
```

Arguments

`object`	a `msets` object.
`degree`	a vector of intersection degrees to pull out.
`...`	additional arguments (not implemented).

Value

A list:

`Barcode`	a vector of 0/1 character strings, representing the set composition of each intersection.
`otab`	a vector of observed intersection size between any combination of sets.
`etab`	a vector of expected intersection size between any combination of sets if background population size is specified.
`set.names`	set names.

set.sizes	set sizes.
n	background population size.
P.value	upper tail p value for each intersection if background population size n is specified.
Table	a data.frame containing degree, otab, etab, fold change, p value and the overlap elements.

Author(s)

Minghui Wang <minghui.wang@mssm.edu>, Bin Zhang <bin.zhang@mssm.edu>

References

Minghui Wang, Yongzhong Zhao, and Bin Zhang (2015). Efficient Test and Visualization of Multi-Set Intersections. *Scientific Reports* 5: 16923.

See Also

msets

Examples

```
## Not run:
#set up fake data
x=list(S1=letters[1:20], S2=letters[10:26], S3=sample(letters,10), S4=sample(letters,10))
obj=supertest(x,n=26)
summary(obj)

## End(Not run)
```

SuperExactTest *SuperExactTest Package*

Description

Efficient Test and Visualization of Multi-set Intersections

Details

The main functions that most users may need from this package are supertest and MSET. For a brief introduction of using this package, please see vignette("set_html").

Author(s)

Minghui Wang <minghui.wang@mssm.edu>, Bin Zhang <bin.zhang@mssm.edu>

References

Minghui Wang, Yongzhong Zhao, and Bin Zhang (2015). Efficient Test and Visualization of Multi-Set Intersections. *Scientific Reports* 5: 16923.

See Also

supertest, MSET

Examples

```
## Not run:
#See a brieft instroduction of using this package
vignette("set_html")

## End(Not run)
```

---

supertest *Calculate Intersections Among Multiple Sets and Perform Statistical Tests*

Description

This function calculates intersection sizes among multiple sets and performs statistical tests of the intersections.

Usage

```
supertest(x, n=NULL, degree=NULL, ...)
```

Arguments

`x`	list; a collection of sets.
`n`	integer, background population size. Required for computing the statistical significance of intersections.
`degree`	a vector of intersection degrees for overlap analysis. E.g., when `degree=c(2:3)`, only those intersections involving two or three sets will be computed. By default, `degree=NULL`, all possible intersections are computed.
`...`	additional arguments (not implemented).

Details

This function calculates intersection sizes between multiple sets and, if background population size `n` is specified, performs statistical tests of the intersections. For a brief introduction of using this package, please see `vignette("set_html")`.

Value

An object of class `msets`.

Author(s)

Minghui Wang <minghui.wang@mssm.edu>, Bin Zhang <bin.zhang@mssm.edu>

References

Minghui Wang, Yongzhong Zhao, and Bin Zhang (2015). Efficient Test and Visualization of Multi-Set Intersections. *Scientific Reports* 5: 16923.

See Also

msets, MSET, Cancer, cpsets, dpsets

Examples

```
## Not run:
#Analyze the cancer gene sets
data(Cancer)
Result=supertest(Cancer, n=20687)
summary(Result)
plot(Result,degree=2:7,sort.by='size')

## End(Not run)
```

Index