Package: GeneOverlap (via r-universe)

November 1, 2026

Type Package

Title Test and visualize gene overlaps

Version 1.48.0

Date 2013-12-13

Author Li Shen, Icahn School of Medicine at Mount Sinai

<shenli.sam@gmail.com>

Maintainer António Miguel de Jesus Domingues, Max-Planck Institute for Cell Biology and Genetics <amjdomingues@gmail.com>

Description Test two sets of gene lists and visualize the results.

License GPL-3

Suggests RUnit, BiocGenerics, BiocStyle

Imports stats, RColorBrewer, gplots, methods

URL http://shenlab-sinai.github.io/shenlab-sinai/

Collate AllClasses.R AllGenerics.R GeneOverlap-accessors.R GeneOverlap-methods.R GeneOverlapMatrix-accessors.R GeneOverlapMatrix-methods.R

biocViews MultipleComparison, Visualization

Repository Bioconductor 3.23

Date/Publication 2026-04-28 12:38:32 UTC

RemoteUrl https://github.com/bioc/GeneOverlap

RemoteRef RELEASE_3_23

RemoteSha 9f3c0b42af73dbfda06e0ab99019fcfe0fe95bbe

Contents

GeneOverlap-package 2
drawHeatmap 3
GeneOverlap 4
GeneOverlapMatrix 6

getGenomeSize 7
getList 8
getReadonly 9
getReadonlyMatrix 10
gs.RNASeq 11
hESC.ChIPSeq.list 12
hESC.RNASeq.list 12
newGeneOverlap 13
newGOM 14
testGeneOverlap 15

Index **16**

`GeneOverlap-package` *Test and visualize overlaps between gene lists*

Description

Given two sets of gene lists, this package calculates the overlaps between all pairs of lists from the two sets. Fisher's exact test is used to determine the p-value and odds ratio in comparison to a genomic background. Plotting functions are provided to visualize the results.

Details

Package:	GeneOverlap
Type:	Package
Version:	0.99.0
Date:	2013-11-01
License:	GPL-3

To use the package, construct one or two named lists each representing a gene set. Each list should contain one or more vectors of gene names. Then use GeneOverlapMatrix to perform pairwise comparisons. It will return an object that can be used for visualization. The GeneOverlapMatrix calls GeneOverlap internally to perform comparison between two gene lists.

Author(s)

Li Shen <<shenli.sam@gmail.com>>

Lab:http://shenlab-sinai.github.io/shenlab-sinai/

Personal:http://www.linkedin.com/in/lshen/

drawHeatmap *Visualize GeneOverlapMatrix objects as heatmaps*

Description

Visualization function for GeneOverlapMatrix objects. Use color gradients to represent the odds ratios or Jaccard indices and the superimposed texts on the grids to represent the p-values of overlaps.

Usage

```
## S4 method for signature 'GeneOverlapMatrix'
drawHeatmap(object,
    what=c("odds.ratio", "Jaccard"), log.scale=F, adj.p=F, cutoff=.05,
    ncolused=9, grid.col=c("Greens", "Blues", "Greys",
    "Oranges", "Purples", "Reds"), note.col="red")
```

Arguments

object	A GeneOverlapMatrix object.
what	What to plot? Odds ratio or Jaccard index.
log.scale	Whether log2 scale shall be used for odds ratios.
adj.p	Boolean label for whether p-values should be adjusted (using the Benjamin-Hochberg method) before showing.
cutoff	P-value cutoff to mask the insignificant comparisons.
ncolused	Number of colors used to represent the scale of odds ratios.
grid.col	Color for odds ratios.
note.col	Color for p-value texts.

Details

The grids that are below the p-value cutoff will be masked and shown as the lightest color.

Examples

```
data(GeneOverlap)
gom.obj <- newGOM(hESC.ChIPSeq.list, genome.size=gs.RNASeq)
drawHeatmap(gom.obj, adj.p=TRUE, cutoff=1,  # show all.
    ncolused=5, grid.col="Blues", note.col="black")
drawHeatmap(gom.obj, log.scale=TRUE, ncolused=5)
drawHeatmap(gom.obj, what="Jaccard", ncolused=5)
```

GeneOverlap *Test overlap between two gene lists using Fisher's exact test.*

Description

Given two gene lists, tests the significance of their overlap in comparison with a genomic background. The null hypothesis is that the odds ratio is no larger than 1. The alternative is that the odds ratio is larger than 1.0. It returns the p-value, estimated odds ratio and intersection.

Usage

```
## S4 method for signature 'GeneOverlap'
show(object)
## S4 method for signature 'GeneOverlap'
print(x, ...)
```

Arguments

object	A GeneOverlap object.
x	A GeneOverlap object.
...	They are not used.

Details

The problem of gene overlap testing can be described by a hypergeometric distribution where one gene list A defines the number of white balls in the urn and the other gene list B defines the number of white balls in the draw. Assume the total number of genes is n, the number of genes in A is a and the number of genes in B is b. If the intersection between A and B is t, the probability density of seeing t can be calculated as:

`dhyper(t, a, n - a, b)`

without loss of generality, we can assume $b <= a$. So the largest possible value for t is b. Therefore, the p-value of seeing intersection t is:

`sum(dhyper(t:b, a, n - a, b))`

The Fisher's exact test forms this problem slightly different but its calculation is also based on the hypergeometric distribution. It starts by constructing a contingency table:

`matrix(c(n - union(A,B), setdiff(A,B), setdiff(B,A), intersect(A,B)), nrow=2)`

It therefore tests the independence between A and B and is conceptually more straightforward. The GeneOverlap class is implemented using Fisher's exact test.

It is better to illustrate a concept using some example. Let's assume we have a genome of size 200 and two gene lists with 70 and 30 genes each. If the intersection between the two is 10, the hypergeometric way to calculate the p-value is:

sum(dhyper(10:30, 70, 130, 30))

which gives us p-value 0.6561562. If we use Fisher's exact test, we should do:

`fisher.test(matrix(c(110, 20, 60, 10), nrow=2), alternative="greater")`

which gives exactly the same p-value. In addition, the Fisher's test function also provides an estimated odds ratio, confidence interval, etc.

The Jaccard index is a measurement of similarity between two sets. It is defined as the number of intersections over the number of unions.

Note

Although Fisher's exact test is chosen for implementation, it should be noted that the R implementation of Fisher's exact test is slower than using `dhyper` directly. As an example, run:

`system.time(sum(dhyper(10e3:30e3, 70e3, 130e3, 30e3)))`

uses around 0.016s to finish. While run:

`system.time(fisher.test(matrix(c(110e3, 20e3, 60e3, 10e3), nrow=2), alternative="greater"))`

uses around 0.072s. In practice, this time difference can often be ignored.

Author(s)

Li Shen <<shenli.sam@gmail.com>>

Lab:http://shenlab-sinai.github.io/shenlab-sinai/

Personal:http://www.linkedin.com/in/lshen/

References

http://en.wikipedia.org/wiki/Fisher's_exact_test

http://en.wikipedia.org/wiki/Jaccard_index

See Also

GeneOverlapMatrix-class

Examples

```
data(GeneOverlap)
go.obj <- newGeneOverlap(hESC.ChIPSeq.list$H3K4me3,
                         hESC.ChIPSeq.list$H3K9me3,
                         gs.RNASeq)
go.obj <- testGeneOverlap(go.obj)
go.obj  # show.
print(go.obj)  # more details.
getContbl(go.obj)  # contingency table.
```

GeneOverlapMatrix *Matrix representation of the pairwise overlaps between two gene sets*

Description

Given one or two gene sets each contains one or more gene lists, create a matrix to represent all pairwise comparisons between the gene lists. This class also provides functions to visualize the matrix.

Usage

```
## S4 method for signature 'GeneOverlapMatrix'
show(object)
## S4 method for signature 'GeneOverlapMatrix'
print(x, ...)
```

Arguments

object	A GeneOverlapMatrix object.
x	A GeneOverlapMatrix object.
...	They are not used.

Details

The problem is stated as the representation of all pairwise comparisons between two gene sets each contains one or more gene lists. This is represented as a matrix where the rows correspond to one gene set and the columns correspond to the other. Each grid represents the overlap between two gene lists of the corresponding row and column. This class calls the GeneOverlap constructor to create objects that represent the overlapping information. When there is only one gene set, the matrix represents the self-comparison within the gene set and only the upper triangular matrix is used.

The significance of gene overlap is characterized by two pieces of information: odds ratio and p-value. This class provides functions to visualize these information as a heatmap. The color gradients of each grid represents the odds ratio while the texts superimposed on the grids state the p-values. It is also possible to let the color gradients represent Jaccard index - a measurement of similarity between two sets.

Author(s)

Li Shen <<shenli.sam@gmail.com>>

Lab:http://shenlab-sinai.github.io/shenlab-sinai/

Personal:http://www.linkedin.com/in/lshen/

See Also

GeneOverlap-class

Examples

```
data(GeneOverlap)
gom.obj <- newGOM(hESC.ChIPSeq.list, hESC.RNASeq.list, gs.RNASeq)
gom.obj
print(gom.obj)
drawHeatmap(gom.obj)
```

---

getGenomeSize *Accessors for the "genome.size" slot of the GeneOverlap class*

Description

The genome.size slot contains the number of genes in the genome as an integer.

Usage

```
## S4 method for signature 'GeneOverlap'
getGenomeSize(object)
## S4 replacement method for signature 'GeneOverlap'
setGenomeSize(object) <- value
```

Arguments

object	A GeneOverlap object.
value	An integer representing genomic background.

Details

After setGenomeSize function is called, the tested Boolean label will be reset to false.

Value

An integer representing the genome size.

See Also

GeneOverlap-class

Examples

```
data(GeneOverlap)
go.obj <- newGeneOverlap(hESC.ChIPSeq.list$H3K27me3,
                         hESC.RNASeq.list$"Exp Medium",
                         genome.size=gs.RNASeq)
getGenomeSize(go.obj)
v.gs <- c(12e3, 14e3, 16e3, 18e3, 20e3)
setNames(sapply(v.gs, function(g) {
    setGenomeSize(go.obj) <- g
```

```
    go.obj <- testGeneOverlap(go.obj)
    getPval(go.obj)
}), v.gs)
```

---

getList *Accessors for the "listA" and "listB" slots of GeneOverlap class*

Description

The listA and listB slots hold the gene lists A and B as character vectors.

Usage

```
## S4 method for signature 'GeneOverlap'
getListA(object)
## S4 replacement method for signature 'GeneOverlap'
setListA(object) <- value
## S4 method for signature 'GeneOverlap'
getListB(object)
## S4 replacement method for signature 'GeneOverlap'
setListB(object) <- value
```

Arguments

object	A GeneOverlap object.
value	A character vector representing gene names.

Details

After setListX function is called, the tested Boolean label will be reset to false.

Value

A character vector representing gene list A/B.

See Also

GeneOverlap-class, newGeneOverlap

Examples

```
data(GeneOverlap)
go.obj <- newGeneOverlap(hESC.ChIPSeq.list$H3K4me3,
                         hESC.ChIPSeq.list$H3K27me3,
                         genome.size=gs.RNASeq)
go.obj <- testGeneOverlap(go.obj)
head(getListB(go.obj))
getTested(go.obj)  # true.
setListB(go.obj) <- hESC.ChIPSeq.list$H3K9me3
getTested(go.obj) # false.
```

getReadonly *Read-only accessors for the "intersection", "union", "is.tested", "cont.tbl", "pval", "odds.ratio", "Jaccard" slots of the GeneOverlap class*

Description

The intersection and union slots contain the gene names as character vectors. The is.tested slot contains a Boolean label indicating whether the object has been tested or not. The cont.tbl slot contains the contingency table as a matrix. The pval and odds.ratio slots contain the p-value and odds ratio as numerics, respectively. The Jaccard slot contains the Jaccard index as a numeric.

Usage

```
## S4 method for signature 'GeneOverlap'
getIntersection(object)
## S4 method for signature 'GeneOverlap'
getUnion(object)
## S4 method for signature 'GeneOverlap'
getTested(object)
## S4 method for signature 'GeneOverlap'
getContbl(object)
## S4 method for signature 'GeneOverlap'
getPval(object)
## S4 method for signature 'GeneOverlap'
getOddsRatio(object)
## S4 method for signature 'GeneOverlap'
getJaccard(object)
```

Arguments

object	A GeneOverlap object.

Details

If the GeneOverlap object has not been tested yet, the returned Jaccard index, p-value and odds ratio will be NA, the contingency table will be an empty matrix.

Value

intersection	A character vector represents the overlapped genes.
union	A character vector represents the genes in the union of A and B.
is.tested	A Boolean represents whether the overlapping test has been performed or not.
cont.tbl	A matrix represents the contingency table.
pval	A numeric represents the significance of overlap.
odds.ratio	A numeric represents the odds ratio in comparison to the genomic background.
Jaccard	A numeric represents the Jaccard index between two sets.

Examples

```
data(GeneOverlap)
go.obj <- newGeneOverlap(hESC.ChIPSeq.list$H3K4me3,
                         hESC.ChIPSeq.list$H3K27me3,
                         genome.size=gs.RNASeq)
go.obj <- testGeneOverlap(go.obj)
head(getIntersection(go.obj))
head(getUnion(go.obj))
getTested(go.obj)
getContbl(go.obj)
getPval(go.obj)
getOddsRatio(go.obj)
getJaccard(go.obj)
```

---

getReadonlyMatrix *Read-only accessors for the various slots of the GeneOverlapMatrix class*

Description

The gsetA and gsetB slots contain the gene set A and B as named lists. The self.compare slot contains the Boolean label for whether self-comparison is performed. Use getMatrix to retrieve intersection, union, Jaccard index, p-value and odds ratio as a matrix. Use getNestedList to retrieve the intersection and union gene lists and contingency tables as a nested list (outer list represents columns and inner list represents rows). Use brackets [] to retrieve a particular GeneOverlap object within the GeneOverlapMatrix object.

Usage

```
## S4 method for signature 'GeneOverlapMatrix'
getGsetA(object)
## S4 method for signature 'GeneOverlapMatrix'
getGsetB(object)
## S4 method for signature 'GeneOverlapMatrix'
getSelfCompare(object)
## S4 method for signature 'GeneOverlapMatrix'
getMatrix(object, name=c("pval",
    "odds.ratio", "intersection", "union", "Jaccard"))
## S4 method for signature 'GeneOverlapMatrix'
getNestedList(object, name=c(
    "intersection", "union", "cont.tbl"))
## S4 method for signature 'GeneOverlapMatrix'
x[i, j]
```

Arguments

object	A GeneOverlapMatrix object.
x	A GeneOverlapMatrix object.

name	A string description of the information to retrive. Use pval and odds.ratio to retrieve p-values and odds ratios. Use Jaccard to retrieve Jaccard indices. In the context of matrix retrieval, intersection and union will return the numbers of genes. While in the case of nested list retrieval, intersection and union will return the actual gene lists. Use cont.tbl to retrieve the contingency tables.
i, j	Integer or character indices to retrieve GeneOverlap objects from the matrix.

Details

When character indices are used, they should match the names of gsetA or gsetB.

See Also

GeneOverlapMatrix-class

Examples

```
data(GeneOverlap)
gom.obj <- newGOM(hESC.ChIPSeq.list, hESC.RNASeq.list, gs.RNASeq)
getMatrix(gom.obj, "odds.ratio")
inter.nl <- getNestedList(gom.obj, "intersection")
str(inter.nl)
go.k4.high <- gom.obj[1, 1]
go.k4.high
```

---

gs.RNASeq *Genome size based on RNA-seq data*

Description

See vignette for data source and processing.

Usage

```
data(GeneOverlap)
```

Format

An integer representing the genomic background.

Examples

```
data(GeneOverlap)
gs.RNASeq
```

hESC.ChIPSeq.list *ChIP-seq gene lists*

Description

See vignette for data source and processing.

Usage

```
data(GeneOverlap)
```

Format

A named list of four character vectors.

Examples

```
data(GeneOverlap)
str(hESC.ChIPSeq.list)
```

hESC.RNASeq.list *RNA-seq gene lists*

Description

See vignette for data source and processing.

Usage

```
data(GeneOverlap)
```

Format

A named list of four character vectors.

Examples

```
data(GeneOverlap)
str(hESC.RNASeq.list)
```

newGeneOverlap *Constructor for the GeneOverlap class*

Description

Use this function to create objects of the GeneOverlap class.

Usage

```
newGeneOverlap(listA, listB, genome.size = NULL,
    spec = c("mm9.gene", "hg19.gene", "rn4.gene"))
```

Arguments

listA	Gene list A. This should be a character vector or a factor.
listB	Gene list B. This should be a character vector or a factor.
genome.size	An integer represents the number of genes on the genome. If not specified, it will use the preset number based on "spec".
spec	A character string of the genome name. Currently choose one of: mm9.gene, hg19.gene, rn4.gene. The gene numbers are based on protein coding genes.

Value

A GeneOverlap object.

Note

Both listA and listB will be converted to unique character vectors before testing, that means, the duplicated gene names are removed and therefore not counted.

Examples

```
data(GeneOverlap)
go.obj <- newGeneOverlap(hESC.ChIPSeq.list$H3K4me3,
                         hESC.ChIPSeq.list$H3K9me3,
                         gs.RNASeq)
print(go.obj)  # not tested yet.
go.obj <- testGeneOverlap(go.obj)
print(go.obj)
```

newGOM *Constructor for the GeneOverlapMatrix class*

Description

Use this function to create objects of the GeneOverlapMatrix class.

Usage

```
newGOM(gsetA, gsetB=list(), genome.size=NULL,
    spec=c('mm9.gene', 'hg19.gene', 'rn4.gene'))
```

Arguments

gsetA	Gene set A as a named list with each element being a vector/factor of gene names.
gsetB	Gene set B as a named list with each element being a vector/factor of gene names.
genome.size	The number of genes in the genome as an integer.
spec	A string description of the genome to use. There are a few presetted genome sizes to choose from for a user's convenience.

Details

This will create a matrix so that each grid represents the comparison between the two gene lists from the two gene sets. Given two gene sets A and B, the matrix will represent all comparisons bewteen gene lists in A vs. gene lists in B. The set A will be shown as rows and the set B will be shown as columns. If only gene set A is given, the matrix will represent all comparisons between gene lists within the gene set and only the upper triangular matrix will be used.

Value

A GeneOverlapMatrix object. Use accesssors to access to its internal structure and members. Use show or print to obtain summarized information. Use drawHeatmap to visualize it.

See Also

GeneOverlapMatrix-class, GeneOverlap-class

Examples

```
data(GeneOverlap)
gom.obj <- newGOM(hESC.ChIPSeq.list, hESC.RNASeq.list, gs.RNASeq)
gom.obj
drawHeatmap(gom.obj)
```

testGeneOverlap *Test function for the GeneOverlap class*

Description

Perform Fisher's exact test based on the information supplied in the GeneOverlap object, i.e. gene list A, B and genome size. This function also calculates the Jaccard index. Will set the tested Boolean label after done.

Usage

```
## S4 method for signature 'GeneOverlap'
testGeneOverlap(object)
```

Arguments

object	A GeneOverlap object.

Value

A GeneOverlap object with valid p-value, odds ratio, Jaccard index and contingency table. The tested Boolean label is set to true. Use show or print to display a summary of the object. Use accessors to get information of each slot.

See Also

GeneOverlap-class

Examples

```
data(GeneOverlap)
go.obj <- newGeneOverlap(hESC.ChIPSeq.list$H3K4me3,
                         hESC.ChIPSeq.list$H3K27me3,
                         genome.size=gs.RNASeq)
go.obj <- testGeneOverlap(go.obj)
getPval(go.obj)
getOddsRatio(go.obj)
getJaccard(go.obj)
getContbl(go.obj)
print(go.obj)
```

Index

∗ **datasets**
gs.RNASeq, 11
hESC.ChIPSeq.list, 12
hESC.RNASeq.list, 12
∗ **graphs**
GeneOverlap, 4
GeneOverlap-package, 2
GeneOverlapMatrix, 6
∗ **hplot**
GeneOverlap, 4
GeneOverlap-package, 2
GeneOverlapMatrix, 6
∗ **htest**
GeneOverlap, 4
GeneOverlap-package, 2
GeneOverlapMatrix, 6
[*(*getReadonlyMatrix*)*, 10
[,GeneOverlapMatrix-method *(*getReadonlyMatrix*)*, 10

drawHeatmap, 3
drawHeatmap,GeneOverlapMatrix-method *(*drawHeatmap*)*, 3

GeneOverlap, 4
GeneOverlap-class *(*GeneOverlap*)*, 4
GeneOverlap-package, 2
GeneOverlapMatrix, 6
GeneOverlapMatrix-class *(*GeneOverlapMatrix*)*, 6
getContbl *(*getReadonly*)*, 9
getContbl,GeneOverlap-method *(*getReadonly*)*, 9
getGenomeSize, 7
getGenomeSize,GeneOverlap-method *(*getGenomeSize*)*, 7
getGsetA *(*getReadonlyMatrix*)*, 10
getGsetA,GeneOverlapMatrix-method *(*getReadonlyMatrix*)*, 10
getGsetB *(*getReadonlyMatrix*)*, 10
getGsetB,GeneOverlapMatrix-method *(*getReadonlyMatrix*)*, 10
getIntersection *(*getReadonly*)*, 9
getIntersection,GeneOverlap-method *(*getReadonly*)*, 9
getJaccard *(*getReadonly*)*, 9
getJaccard,GeneOverlap-method *(*getReadonly*)*, 9
getList, 8
getListA *(*getList*)*, 8
getListA,GeneOverlap-method *(*getList*)*, 8
getListB *(*getList*)*, 8
getListB,GeneOverlap-method *(*getList*)*, 8
getMatrix *(*getReadonlyMatrix*)*, 10
getMatrix,GeneOverlapMatrix-method *(*getReadonlyMatrix*)*, 10
getNestedList *(*getReadonlyMatrix*)*, 10
getNestedList,GeneOverlapMatrix-method *(*getReadonlyMatrix*)*, 10
getOddsRatio *(*getReadonly*)*, 9
getOddsRatio,GeneOverlap-method *(*getReadonly*)*, 9
getPval *(*getReadonly*)*, 9
getPval,GeneOverlap-method *(*getReadonly*)*, 9
getReadonly, 9
getReadonlyMatrix, 10
getSelfCompare *(*getReadonlyMatrix*)*, 10
getSelfCompare,GeneOverlapMatrix-method *(*getReadonlyMatrix*)*, 10
getTested *(*getReadonly*)*, 9
getTested,GeneOverlap-method *(*getReadonly*)*, 9
getUnion *(*getReadonly*)*, 9
getUnion,GeneOverlap-method *(*getReadonly*)*, 9
gs.RNASeq, 11

hESC.ChIPSeq.list, 12
hESC.RNASeq.list, 12

newGeneOverlap, *8*, 13
newGOM, 14

print,GeneOverlap-method *(*GeneOverlap*)*, 4
print,GeneOverlapMatrix-method *(*GeneOverlapMatrix*)*, 6

setGenomeSize<- *(*getGenomeSize*)*, 7
setGenomeSize<-,GeneOverlap-method *(*getGenomeSize*)*, 7
setListA<- *(*getList*)*, 8
setListA<-,GeneOverlap-method *(*getList*)*, 8
setListB<- *(*getList*)*, 8
setListB<-,GeneOverlap-method *(*getList*)*, 8
show,GeneOverlap-method *(*GeneOverlap*)*, 4
show,GeneOverlapMatrix-method *(*GeneOverlapMatrix*)*, 6

testGeneOverlap, 15
testGeneOverlap,GeneOverlap-method *(*testGeneOverlap*)*, 15