Package: rSEA (via r-universe)

August 21, 2024

Type Package

Title Simultaneous Enrichment Analysis

Version 2.1.2

Author Mitra Ebrahimpoor

Maintainer Mitra Ebrahimpoor<m.ebrahimpoor@lumc.nl>

Description SEA performs simultaneous feature-set testing for (gen)omics data. It tests the unified null hypothesis and controls the family-wise error rate for all possible pathways. The unified null hypothesis is defined as: ``The proportion of true features in the set is less than or equal to a threshold." Family-wise error rate control is provided through use of closed testing with Simes test. There are some practical functions to play around with the pathways of interest.

Depends R (>= 2.10), hommel (>= 1.4), ggplot2

Suggests knitr, rmarkdown

License GPL (>= 2)

Date 2019-10-02

Encoding UTF-8

VignetteBuilder knitr

RoxygenNote 7.3.1

Repository https://mitra-ep.r-universe.dev

RemoteUrl https://github.com/mitra-ep/rsea

RemoteRef HEAD

RemoteSha e1da2d83d96adcd3917098afdb0239962c759ddb

Contents

`rSEA-package` *Simultaneous Enrichment Analysis (SEA) of all possible feature-sets using the unified null hypothesis*

Description

This package uses raw p-values of genomic features as input and evaluates any given list of feature-sets or pathways. For each set the adjusted p-value and TDP lower-bound are calculated. The type of test can be defined by arguments and can be refined as necessary. The p-values are corrected for every possible set of features, making the method flexible in choice of pathway list and test type. For more details see: Ebrahimpoor, M (2019) <doi:10.1093/bib/bbz074>

Details

The unified null hypothesis is tested using closed testing procedure and all-resolutions inference. It combines the self-contained and ompetitive approaches in one framework. In short, using p-values of the individual features as input, the package can provide an FWER-adjusted p-value along with a lower bound and a point estimate for the proportion of true discoveries per feature-set. The flexibility in revising the choice of feature-sets without inflating type-I error is the most important property of SEA.

Author(s)

Mitra Ebrahimpoor.

Maintainer: Mitra Ebrahimpoor<m.ebrahimpoor@lumc.nl>

References

Mitra Ebrahimpoor, Pietro Spitali, Kristina Hettne, Roula Tsonaka, Jelle Goeman, Simultaneous Enrichment Analysis of all Possible Gene-sets: Unifying Self-Contained and Competitive Methods, Briefings in Bioinformatics,bbz074 https://doi.org/10.1093/bib/bbz074

plotSEA *topSEA*

Description

returns a plotof SEA-chart which illustrates proportion of discoveries per pathway.

Usage

```
plotSEA(object, by = "TDP.estimate", threshold = 0.005, n = 20)
```

Arguments

object	A SEA-chart object which is the output of SEA function
by	the Variable which will we mapped. It should be either the TDP estimate or TDP bound.The default is TDP bound.
threshold	A real number between 0 and 1. Which will be used as a visual aid to distinguish significant pathways
n	Integer. Number of rows from SEA-chart object to be plotted.

Value

Returns a plot of SEA_chart according to the selected arguments

Author(s)

Mitra Ebrahimpoor

<m.ebrahimpoor@lumc.nl>

References

Mitra Ebrahimpoor, Pietro Spitali, Kristina Hettne, Roula Tsonaka, Jelle Goeman, Simultaneous Enrichment Analysis of all Possible Gene-sets: Unifying Self-Contained and Competitive Methods, Briefings in Bioinformatics,bbz074

See Also

SEA

Examples

```
#See the examples for \code{\link{SEA}}
```

SEA *SEA*

Description

returns SEA chart (a data.frame) including the test results and estimates for the specified feature-sets from `pathlist`.

Usage

```
SEA(
  pvalue,
  featureIDs,
  data,
  pathlist,
  select,
  tdphat = TRUE,
  selfcontained = TRUE,
  competitive = TRUE,
  thresh = NULL,
  alpha = 0.05
)
```

Arguments

`pvalue`	Vector of p-values. It can be the name of the covariate representing the Vector of all raw p-values in the `data` or a single vector but in the latter case it should match the `featureIDs` vector
`featureIDs`	Vector of feature IDs. It can be the name of the covariate representing the IDs in the `data` or a single vector but in the latter case it should match the `pvalue` vector
`data`	Optional data frame or matrix containing the variables in `pvalue` and `featureIDs`
`pathlist`	A list containing pathways defined by `featureIDs`. Checkout the vignette for more details and available codes to create your own pathway
`select`	A vector. Number or names of pathways of interest from the `pathlist` of choice. If missing, all pathways of the database will be included
`tdphat`	Logical. If `TRUE` the point estimate of the True Discoveries Proportion within each pathway will be calculated
`selfcontained`	Logical. If `TRUE` the self-contained null hypothesis will be tested for each pathway and the corresponding adj. p-value is returned
`competitive`	Logical. If `TRUE` the default competitive null hypothesis will be tested for each pathway and the corresponding adj. p-value is returned, you can define a threshold with `thresh` argument

thresh	A real number between 0 and 1. If specified, the competitive null hypothesis will be tested against this threshold for each pathway and the corresponding adj. p-value is returned
alpha	The type I error allowed for TDP bound. The default is 0.05.

Value

A data.frame is returned including a list of pathways with corresponding TDP bound estimate, and if specified, TDP point estimate and adjusted p-values

Author(s)

Mitra Ebrahimpoor

<m.ebrahimpoor@lumc.nl>

References

Mitra Ebrahimpoor, Pietro Spitali, Kristina Hettne, Roula Tsonaka, Jelle Goeman, Simultaneous Enrichment Analysis of all Possible Gene-sets: Unifying Self-Contained and Competitive Methods, Briefings in Bioinformatics, , bbz074, https://doi.org/10.1093/bib/bbz074

See Also

setTest, topSEA,

Examples

```
## Not run:
##Generate a vector of pvalues for a toy example
set.seed(159)

m<- 100
pvalues <- runif(m,0,1)^5
featureIDs <- as.character(1:m)

# perform a self-contained test for all features
setTest(pvalues, featureIDs, testype = "selfcontained")

# create 3 random pathway of size 60, 20 and 45
randpathlist=list(A=as.character(c(sample(1:m, 60))),
             B=as.character(c(sample(1:m, 20))),
             C=as.character(c(sample(1:m, 45))))


# get the seachart for the whole pathlist
S1<-SEA(pvalues, featureIDs, pathlist=randpathlist)
S1

# get the seachart for only first two pathways of the randpathlist
S2<-SEA(pvalues, featureIDs, pathlist=randpathlist, select=1:2)
S2
```

```
#sort the list by competitve p-value and select top 2
topSEA(S2, by=Comp.adjP, descending = FALSE, n=2)

#make an enrichment plot based on TDP.estimated of pathways
plotSEA(S1,n=3)

## End(Not run)
```

---

setTDP *setTDP*

Description

Estimates the TDP of the specified set of features.

Usage

```
setTDP(pvalue, featureIDs, data, set, alpha = 0.05)
```

Arguments

pvalue	The vector of p-values. It can be the name of the covariate representing the Vector of raw p-values in the data or a single vector but in the latter case it should match the featureIDs vector
featureIDs	The vector of feature IDs. It can be the name of the covariate representing the IDs in the data or a single vector but in the latter case it should match the pvalue vector
data	Optional data frame or matrix containing the variables in pvalue and featureIDs
set	The selection of features defining the feature-set based on the the featureIDs. If missing, the set of all features is evaluated
alpha	The type I error allowed. The default is 0.05. NOTE: this shouls be consistent across the study

Value

A named vector including the lower bound and point estimate for the true discovery proportion (TDP) of the specified test for the feature-set is returned.

Author(s)

Mitra Ebrahimpoor

<m.ebrahimpoor@lumc.nl>

References

Mitra Ebrahimpoor, Pietro Spitali, Kristina Hettne, Roula Tsonaka, Jelle Goeman, Simultaneous Enrichment Analysis of all Possible Gene-sets: Unifying Self-Contained and Competitive Methods, Briefings in Bioinformatics, , bbz074, https://doi.org/10.1093/bib/bbz074

See Also

setTest, SEA

Examples

```
## Not run:
set.seed(159)
#generate random p-values with pseudo IDs
m<- 100
pvalues <- runif(m,0,1)^5
featureIDs <- as.character(1:m)

# perform a self-contained test for all features
settest(pvalues, featureIDs, testype = "selfcontained")

# estimate the proportion of true discoveries among all m features
settdp(pvalues, featureIDs)

# create a random pathway of size 60
randset=as.character(c(sample(1:m, 60)))


# estimate the proportion of true discoveries in a random set of size 50
settdp(pvalues, featureIDs, set=randset)


## End(Not run)
```

---

setTest *setTest*

Description

calculates the adjusted p-value for the local hypothesis as defined by `testtype` and `testvalue`.

Usage

```
setTest(pvalue, featureIDs, data, set, testype, testvalue)
```

Arguments

pvalue	The vector of p-values. It can be the name of the covariate representing the Vector of raw p-values in the data or a single vector but in the latter case it should match the featureIDs vector
featureIDs	The vector of feature IDs. It can be the name of the covariate representing the IDs in the data or a single vector but in the latter case it should match the pvalue vector
data	Optional data frame or matrix containing the variables in pvalue and featureIDs
set	The selection of features defining the feature-set based on the the featureIDs. If missing, the set of all features is selected
testype	Character, type of the test: "selfcontained" or "competitive". Choosing the self-contained option will automatically set the threshold to zero and the testvalue is ignored. Choosing the competitive option without a testvalue will set the threshold to the overall estimated proportion of true hypotheses
testvalue	Optional value to test against. Setting this value to c along with testype=="competitive" will lead to testing the null hypothesis against a threshold c. Note: this value needs to be a proportion

Value

The adjusted p-value of the specified test for the feature-set is returned.

Author(s)

Mitra Ebrahimpoor

<m.ebrahimpoor@lumc.nl>

References

Mitra Ebrahimpoor, Pietro Spitali, Kristina Hettne, Roula Tsonaka, Jelle Goeman, Simultaneous Enrichment Analysis of all Possible Gene-sets: Unifying Self-Contained and Competitive Methods, Briefings in Bioinformatics, , bbz074, https://doi.org/10.1093/bib/bbz074

See Also

setTDP SEA

Examples

```
## Not run:
#Generate a vector of pvalues
set.seed(159)

m<- 100
pvalues <- runif(m,0,1)^5
featureIDs <- as.character(1:m)

# perform a self-contained test for all features
```

```
settest(pvalues, featureIDs, testype = "selfcontained")

# create a random pathway of size 60
randset=as.character(c(sample(1:m, 60)))

# perform a competitive test for the random pathway
settest(pvalues, featureIDs, set=randset, testype = "competitive")

# perform a unified null hypothesis test against 0.2 for a set of size 50
settest(pvalues, featureIDs, set=randset, testype = "competitive", testvalue = 0.2 )


## End(Not run)
```

---

topSEA *topSEA*

Description

returns a permutation of SEA-chart which rearranges the feature-sets according to the selected argument into ascending or descending order.

Usage

```
topSEA(object, by, thresh = NULL, descending = TRUE, n = 20, cover)
```

Arguments

object	A SEA-chart object which is the output of SEA function
by	Variable name by which the ordering should happen. It should be a column of SEA-chart. The default is TDP_bound.
thresh	A real number between 0 and 1. If specified the values of the variable defined in by will be threshold accordingly.
descending	Logical. If TRUE The output chart is organized in a descending order
n	Integer. Number of raws of the output chart
cover	An optional threshold for coverage, which must be a real number between 0 and 1. If specified, feature-sets with a coverage lower than or equal to this value are removed.

Value

Returns a subset of SEA_chart sorted according to the arguments

Author(s)

Mitra Ebrahimpoor

<m.ebrahimpoor@lumc.nl>

References

Mitra Ebrahimpoor, Pietro Spitali, Kristina Hettne, Roula Tsonaka, Jelle Goeman, Simultaneous Enrichment Analysis of all Possible Gene-sets: Unifying Self-Contained and Competitive Methods, Briefings in Bioinformatics,bbz074

See Also

SEA

Examples

```
#See the examples for \code{\link{SEA}}
```

Index