Package ‘sigora’

October 14, 2022

Type Package
Title Signature Overrepresentation Analysis
Version 3.1.1
Date 2021-11-29
Depends R (>= 3.0.0)
URL https://github.com/wolski/sigora
BugReports https://github.com/wolski/sigora/issues
Imports stats, slam
Suggests knitr
Description Pathway Analysis is statistically linking observations on the molecular level to biological processes or pathways on the systems (i.e., organism, organ, tissue, cell) level. Traditionally, pathway analysis methods regard pathways as collections of single genes and treat all genes in a pathway as equally informative. However, this can lead to identifying spurious pathways as statistically significant since components are often shared amongst pathways. SIGORA seeks to avoid this pitfall by focusing on genes or gene pairs that are (as a combination) specific to a single pathway. In relying on such pathway gene-pair signatures (Pathway-GPS), SIGORA inherently uses the status of other genes in the experimental context to identify the most relevant pathways. The current version allows for pathway analysis of human and mouse datasets. In addition, it contains pre-computed Pathway-GPS data for pathways in the KEGG and Reactome pathway repositories and mechanisms for extracting GPS for user-supplied repositories.

License GPL-3

NeedsCompilation no

biocViews GeneSetEnrichment, GO, Software, Pathways, KEGG

RoxygenNote 7.1.2
Function to randomly select genes associated with randomly pathways.

This function first randomly selects a number (np) of pathways, then randomly selects a number (ng) of genes that are associated with at least one of the selected pathways. The function can be used to compare Sigora’s performance to traditional overrepresentation tests.

```r
genesFromRandomPathways(GPSrepo, np, ng)
```
getGenes

Value

selectedPathways
A vector containing the "np" originally selected pathways.

genes
A vector containing the "ng" selected genes from selectedPathways.

References

Foroushani AB, Brinkman FS and Lynn DJ (2013). “Pathway-GPS and SIGORA: identifying relevant pathways based on the over-representation of their gene-pair signatures.” PeerJ, 1

See Also

sigora-package

Examples

data('kegH')
## select 50 genes from 3 human KEGG pathways
seed=1234
set.seed(seed)
a1 <- genesFromRandomPathways(kegH,3,50)
## originally selected pathways:
a1[['selectedPathways']]    
## what are the genes
a1[['genes']]            
## sigora's results
sigoraRes <- sigora(GPSrepo =kegH, queryList = a1[['genes']],
                    level = 4)
## compare to traditional methods results:
oraRes <- ora(a1[['genes']],kegH)
dim(oraRes)
oraRes

getGenes

List genes involved in present GPS for a specific pathway in the summary_results

Description

This function lists the genes involved in the present GPS for a pathway of interest, ordered by their contribution to the significance of the pathway.

Usage

genesFromRandomPathways(kegH,3,50)
Arguments

`yy` A sigora analysis result object (created by `sigora`).
`i` The rank position of the pathway of interest in `summary_results`
`idmap` A dataframe for converting between different gene-identifier types (e.g. ENSEMBL, ENTREZ and HGNC-Symbols of genes). Most users do not need to set this argument, as there is a built-in conversion table.

Value

A table (dataframe) with ids of the relevant genes, ranked by their contribution to the statistical significance of the pathway.

See Also

`sigora`

Examples

```r
# original selected pathways:
a1[["selectedPathways"]]
# what are the genes
a1[["genes"]]
# sigora's results with this input:
sigoraRes <- sigora(GPSrepo = kegH, queryList = a1[["genes"]], level = 2)
# Genes related to the second most significant result:
head(getGenes(sigoraRes, 2))
```

### getURL

`getURL` *Highlight the relevant genes for a specific pathway in its pathway diagram*

**Description**

This function highlights the genes involved in the present GPS for a pathway of interest in its diagram. Please note that this functionality is only implemented for results of Reactome or KEGG based analyses.

**Usage**

`getURL(yy, i)`
**Arguments**

- **yy**
  - A sigora analysis result object (created by `sigora`).
- **i**
  - The rank position of the pathway of interest in `summary_results`.

**Value**

The URL of the pathway diagram, where the relevant genes from your original query list are highlighted.

**See Also**

- `sigora`

**Examples**

```r
data('kegH')
set.seed(seed=12345)
a1<-genesFromRandomPathways(kegH,3,50)
## originally selected pathways:
    a1[["selectedPathways"]]
## what are the genes
    a1[["genes"]]
## sigora's results with this input:
    sigoraRes <- sigora(GPSrepo =kegH, queryList = a1[["genes"]],level = 2)
## Diagram for the most significant result, where the genes from our list are highlighted in red:
    getURL(sigoraRes,1)
```

---

**idmap**

*Identifier mappings for protein coding genes.*

**Description**

A mapping table for ENSEMBL, ENTREZ and gene names (HGNC/MGI symbols) of Human and mouse protein coding gene.

**Source**

- www.ensembl.org/biomart/martview

**Examples**

```r
data(idmap)
head(idmap)
```
**kegH**  
*Pathway GPS data, extracted from KEGG repository (Human).*

**Description**

KEGG human pathway GPS data, extracted by `makeGPS`, default settings. This data can be used by `sigora` to preform signature overrepresenation.

**Source**

<http://www.genome.jp/kegg/pathway.html>

**References**


**See Also**

`makeGPS`, `sigora`, `reaH`

**Examples**

```r
data(kegH)  
str(kegH)
```

**kegM**  
*Pathway GPS data, extracted from KEGG repository (Mouse).*

**Description**

KEGG mouse pathway GPS data, extracted by `makeGPS`, default settings. This data can be used by `sigora` to preform signature overrepresenation.

**Source**

<http://www.genome.jp/kegg/pathway.html>

**References**

### load_data

**Description**
load and return data when lazyLoad false insted of using data(datastr)

**Usage**
load_data(datastr, package = "sigora")

**Arguments**
- datastr: name of datasets
- package: default sigora

**Value**
returns the data

**Examples**

```r
idmap <- load_data("idmap")
```

---

**makeGPS**

Create your own Signature Object.

**Description**
Given a repository of gene-pathway associations either in a tab delimited file with three columns (pathwayID,pathway Description,Gene) or a corresponding dataframe, this function identifies all Gene Pair Signatures (pairs of genes that are as a combination unique to a single pathway) and Pathway Unique Genes (genes that are uniquely associated with a single pathway) and stores them in a format that is usable by sigora. Please also see the "details" and "note" sections below.
Usage

makeGPS(
    pathwayTable = NULL,
    fn = NULL,
    maxLevels = 5,
    saveFile = NULL,
    repoName = "userrepo",
    maxFunperGene = 100,
    maxGenesperPathway = 500,
    minGenesperPathway = 10
)

Arguments

pathwayTable  A data frame describing gene-pathway associations in the following format: pathwayID,pathwayName,Gene. Either pathwayTable or fn should be provided.
fn            Where to find the repository. Either pathwayTable or fn should be provided.
maxLevels     For hierarchical repositories, the number of levels to consider.
saveFile      Where to save the object as an rda file.
repoName      Repository name.
maxFunperGene A cutoff threshold, genes with more than this number of associated pathways are excluded to speed up the GPS identification process.
maxGenesperPathway A cutoff threshold, pathways with more than this number of associated genes are excluded to speed up the GPS identification process.
minGenesperPathway A cutoff threshold, pathways with less than this number of associated genes are excluded to speed up the GPS identification process.

Details

The primary purpose of makeGPS is to convert a user-supplied gene-pathway association table into a repository of weighted Gene Pair Signatures (GPS) that are unique features of pathways. Such GPS can then be used for signature (gene-pair) based analyses using sigora. Additionally, the resulting object also retains the original "single gene"-"pathway" associations for the purpose of followup analyses, such as comparison of sigora-results to traditional methods. ora is an implementation of the traditional (individual gene) Overrepresentation Analysis.

Value

A GPS repository, to be used by sigora and ora.

Note

This function relies on package slam, which should be installed from CRAN. It is fairly memory intensive, and it is recommended to be run on a machine with at least 6GB of RAM. Also, make sure to save and reuse the resulting GPS repository in future analyses!
References

Foroushani AB, Brinkman FS and Lynn DJ (2013). “Pathway-GPS and SIGORA: identifying relevant pathways based on the over-representation of their gene-pair signatures.” PeerJ, 1

See Also

sigora, sigora-package

Examples

data(nciTable); data(idmap)
# what the input looks like:
head(nciTable)
# create a SigObject. use the saveFile parameter for reuse.

nciH<-makeGPS(pathwayTable=load_data(’nciTable’))
ils<-grep("^IL", idmap[, "Symbol"], value=TRUE)
ilnci<-sigora(queryList=ils, GPSrepo=nciH, level=3)

nciTable

NCI human gene-pathway associations.

Description

PID-NCI human pathway repository, as a data frame with three columns corresponding to : pathwayId, pathwayName, gene. This is an example of the expected format for the input data to makeGPS.

Details

This dataset is provided to illustrate how to create your own GPS repositories. nciTable is a dataframe with three columns corresponding to pathwayId, pathwayName, and gene. Each row describes the association between an individual gene and a PID-NCI pathway. As you see in the examples section, one can convert this dataframe to a GPS repository using makeGPS, and save the results for future reuse. Using the thus created GPS repository one can preform Signature Overrepresentation Analysis on lists of genes of interest.

Source

<https://github.com/NCIP/pathway-interaction-database/tree/master/download>
Examples

data(n.ciTable)
nciH<-makeGPS(pathwayTable=load_data('nciTable'))
data(idmap)
ils<-grep("IL",idmap[,"Symbol"],value=TRUE)
ilnci<-sigora(queryList=ils,GPSrepo=nciH,level=3)

ora Traditional Overrepresentation Analysis.

Description

Traditional Overrepresentation Analysis by hypergeometric test: pathways are treated as collections
of individual genes and all genes are treated as equally informative. This function is provided for
comparison of the results of traditional methods to Sigora.

Usage

ora(genesList, GPSrepo, idmap = load_data("idmap"))

Arguments

genesList A vector containing the list of genes of interest (e.g. differentially expressed
genes). Following Identifier types are supported: Gene Symbols, ENTREZ-IDs
or ENSEMBL-IDs.

GPSrepo A GPS-repository (either one of the provided precomputed GPS repositories) or
one created by makeGPS.

idmap A dataframe for converting between different gene-identifier types (e.g. EN-
SEMBL, ENTREZ and HGNC-Symbols of genes). Most users do not need to
set this argument, as there is a built-in conversion table.

Details

The primary purpose of makeGPS is to create a GPS repository. It does, however, retain the original
"single gene"-"pathway" associations for the purpose of followup analyses, such as comparison of
sigora-results to traditional methods. ora is an implementation of the traditional (individual gene)
Overrepresentation Analysis.

Value

A dataframe with individual gene ORA results.

See Also

sigora-package
Examples

```r
data(kegM)
## select 50 genes from 3 mouse pathways
set.seed(seed=345)
a1<-genesFromRandomPathways(kegM,3,50)
## originally selected pathways:
a1[["selectedPathways"]]
## compare to traditional methods results:
oraRes <- ora(a1[["genes"]],kegM)
dim(oraRes)
oraRes
```

---

**reaH**  
*Pathway GPS data, extracted from the Reactome repository (Human).*

**Description**

Reactome human pathway GPS data, extracted by `makeGPS`, default settings. This data can be used by `sigora` to preform signature overrepresenation.

**Source**

<http://www.reactome.org/>

**References**


**Examples**

```r
data(reaH)
## maybe str(reaH) ; ...```
**Description**

Reactome mouse pathway GPS data, extracted by makeGPS, default settings. This data can be used by sigora to perform signature overrepresentation.

**Source**

<http://www.reactome.org/>

**References**


**See Also**

makeGPS, sigora, kegM

**Examples**

```r
data(reaM)
str(reaM)
```

**Description**

This function determines which Signatures (GPS) from a collection of GPS data (GPSrepo argument) for the specified pathway repository are present in the specified list of genes of interest (queryList argument). It then uses the distribution function of hypergeometric probabilities to identify the pathways whose GPS are over-represented among the present GPS and saves the results to the file specified in the saveFile argument.
Usage

```r
sigora(
  GPSrepo,
  level,
  markers = FALSE,
  queryList = NULL,
  saveFile = NULL,
  weighting.method = "invhm",
  idmap = load_data("idmap")
)
```

Arguments

- **GPSrepo**: An object created by `makeGPS` or one of the precompiled GPS data collections that are provided with this package (currently for KEGG and Reactome). e.g. `reaH` for human Reactome GPS, `kegH` for human KEGG GPS, and `reaM` and `kegM` for corresponding mouse GPS. See the examples section for creating and using your own GPS.

- **level**: In hierarchical repositories (e.g. Reactome) number of levels to consider. Recommended value for KEGG: 2, for Reactome: 4.

- **markers**: Whether to take single genes that are uniquely associated with only one pathway into account (i.e. should pathway unique genes/PUGs be considered GPS?). Recommended value: TRUE (1).

- **queryList**: A user specified list of genes of interest (`query list`), as a vector of ENSEMBL/ENTREZ IDs or gene symbols (HGNC/MGI).

- **saveFile**: If provided, the results are saved here as a tab delimited File (including , for each pathway, a list of genes ordered by their contribution to the statistical significance of the pathway).

- **weighting.method**: The weighting method or GPS. The default weighting scheme for the GPS is the reciproc of the harmonic mean of the degrees of the two component genes of a GPS. A wide range of alternative weighting schemes are pre-implemented (see below). Additional user defined weighting schemes are also supported. Currently, the following alternatives are pre-implemented:
  - `'noweights'`: assigns a constant of 1 to all GPS.
  - `'cosine'`: all GPS are weighted by the cosine of the degrees of their constituent genes.
  - `'topov'`: all GPS are weighted by topological overlap of their constituent genes.
  - `'reciprod'`: all GPS are weighted by reciproc of product of the number of pathway annotations of their constituent genes.
  - `'jac'`: all GPS are weighted by the jaccard similarity of the pathway annotations constituent genes.
  - `'justPUGs'`: Analysis is performed using PUGs only.
'invhm': all GPS are weighted by the reciprocal of the harmonic mean of the degrees of their constituent genes (default).

`idmap` A dataframe for converting between different gene-identifier types (e.g. ENSEMBL, ENTREZ and HGNC-Symbols of genes). Most users do not need to set this argument, as there is a built-in conversion table.

Value

`summary_results` A dataframe listing the analysis results.
`detailed_results` A dataframe describing the detailed evidence (present Gene-Pair Signatures) for each pathway.

References

Foroushani AB, Brinkman FS and Lynn DJ (2013). “Pathway-GPS and SIGORA: identifying relevant pathways based on the over-representation of their gene-pair signatures.” PeerJ, 1

See Also

`sigora-package`, `makeGPS`

Examples

```r
##query list
ils <- grep("^IL",load_data('idmap')[['Symbol']],value=TRUE)
## using precompiled GPS repositories:
sigRes.ilreact <- sigora(queryList=ils,GPSrepo=load_data('reaH'),level=4)
sigRes.ilkeg <- sigora(queryList=ils,GPSrepo=load_data('kegH'),level=2)
## user created GPS repository:
nciH<-makeGPS(pathwayTable=load_data('nciTable'))
sigRes.ilnci<-sigora(queryList=ils,GPSrepo=nciH,level=2)
## user defined weighting schemes :
myfunc<-function(a,b){1/log(a+b)}
sigora(queryList=ils,GPSrepo=nciH,level=2, weighting.method = myfunc)
```
Index

* datasets
  - idmap, 5
  - kegH, 6
  - kegM, 6
  - nciTable, 9
  - reaH, 11
  - reaM, 12

* functions
  - genesFromRandomPathways, 2
  - getGenes, 3
  - getURL, 4
  - makeGPS, 7
  - ora, 10
  - sigora, 12

genesFromRandomPathways, 2
getGenes, 3
getURL, 4

idmap, 5

kegH, 6
kegM, 6, 12

load_data, 7

makeGPS, 6, 7, 12, 14

nciTable, 9

ora, 10

reaH, 6, 11
reaM, 12

sigora, 4–6, 9, 12, 12