Package 'HPOSim'

July 2, 2014

Type Package

Title Analysis semantic similarity between HPO terms and HPO-based phenotypic similarity between genes and between diseases.

Version 1.2

Date 2014-02-13

Author Yue Deng, Gang Wang, Xiaocheng Huang, Qinhao Wu , Yang Wang and Tingting Ma

Maintainer Yue Deng <anfdeng@163.com>

Description

This package implements multiple similarity measures for HPO terms, genes and diseases.
It is aiming at phenotype analysis for gene sets and disease sets.
Functions for HPO enrichment analysis is also provided.
The disease-term and gene-term associations contained in this version
are based on the official HPO data released in January 2014.

Depends R (>= 2.15.1)

Imports AnnotationDbi, HPO.db, igraph

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2014-02-21 11:15:24

R topics documented:

`HPOSim-package` *HPOSim package*

Description

This package implements multiple similarity measures and enrichment analysis functions for HPO terms ,gene products and diseases.

Details

Package:	HPOSim
Type:	Package
Version:	1.1
Date:	2013-05-25
License:	GPL-2

Author(s)

Yue Deng, Gang Wang, Xiaocheng Huang, Qinhao Wu and Tingting Ma Maintainer: Yue Deng <anfdeng@163.com>

`calcTermSim` *Semantic Similarity of Two HPO Terms*

Description

Given two HPO terms, return semantic similarity of the terms based on the map of HPO terms and genes.

Usage

```
calcTermSim(term1, term2, method = "Resnik", IC, verbose = FALSE)
```

Arguments

term1	one HPO term
term2	another HPO term
method	one of "Resnik", "JiangConrath", "Lin", "simIC", "relevance" , "GIC" and "Wang"
IC	the IC used in calculating similarity
verbose	print some information

Value

Semantic similarity of the two terms.

Author(s)

Yue Deng <anfdeng@163.com>

See Also

getGeneSim

Examples

```
.initialize()
IC<-get("termIC", envir=HPOSimEnv)
calcTermSim("HP:0000028","HP:0000033", method = "Wang", IC, verbose = FALSE)
IC<-get("DiseasetermIC", envir=HPOSimEnv)
calcTermSim("HP:0000028","HP:0000033", method = "Resnik", IC, verbose = FALSE)
```

---

getDiseaseListSim *Pairwise Similarity for a List of Diseases*

Description

Given a list of diseases, the function calculates the pairwise similarities for any two diseases in the list using different strategies.

Usage

```
getDiseaseListSim(diseaselist,combinemethod="funSimMax",
                  method="Resnik",ontology="PA",normalization=FALSE,
                  normalizationmethod="Lin",verbose=FALSE)
```

Arguments

diseaselist	character vector of disease OMIM IDs
combinemethod	method to calculate the similarity between diseases based on , one of "max", "mean", "funSimMax", "funSimAvg" and ""BMA
method	method to compute the similarity of HPO terms, one of "Resnik", "JiangConrath", "Lin", "simIC", "relevance" and "Wang"
ontology	the ontology used for similarity calculation
normalization	normalize similarities yes/no
normalizationmethod	one of "sqrt", "Lin" and "Tanimoto"
verbose	print out some information

Details

The combine method to calculate the pairwise disease similarity between disease can either be:

"max" the maximum similarity between any two HPO terms

"mean" the average similarity between any two HPO terms1[1]

funSimMax the average of best matching HPO term similarities. Take the maximum of the scores achieved by assignments of HPO terms from disease 1 to disease 2 and vice versa. [2]

funSimAvg the average of best matching HPO term similarities. Take the average of the scores achieved by assignments of HPO terms from disease 1 to disease 2 and vice versa. [2]

"BMA" best match average approach [3]

Value

n*n similarity matrix (n = number of diseases)

Author(s)

Yue Deng<anfdeng@163.com>

See Also

getDiseaseSim

Examples

```
list<-c("OMIM:101900","OMIM:102000","OMIM:143470")
getDiseaseListSim(list,combinemethod="funSimAvg")
```

getDiseaseSim *Phenotype-based similarity for two diseases base on HPO*

Description

Given two diseases, the function calculates similarity of the diseases based on HPO.

Usage

```
getDiseaseSim(disease1,disease2,combinemethod="funSimMax",
              method="Resnik",ontology="PA",normalization=FALSE,
              normalizationmethod="Lin", verbose=FALSE)
```

Arguments

disease1	one disease OMIM ID
disease2	another disease OMIM ID
combinemethod	method to calculate the similarity between diseases based on , one of "max", "mean", "funSimMax", "funSimAvg" and ""BMA
method	method to compute the similarity of HPO terms, one of "Resnik", "JiangConrath", "Lin", "simIC", "relevance", "GIC" and "Wang"
ontology	the ontology used to calculate the similarity, one of "PA"(Phenotypic Abnormality), "MI"(Mode of Inheritance) and "OC"(Onset and Clinical Course)
normalization	whether do the normalization
normalizationmethod	one of "sqrt", "Lin" and "Tanimoto"
verbose	print out some information

Details

The method to combine the similarities between HPO terms can either be:

"max" the maximum similarity between any two HPO terms

"mean" the average similarity between any two HPO terms1

"funSimMax" the average of best matching HPO term similarities. Take the maximum of the scores achieved by assignments of HPO terms from disease 1 to disease 2 and vice versa.

"funSimAvg" the average of best matching HPO term similarities. Take the average of the scores achieved by assignments of HPO terms from disease 1 to disease 2 and vice versa.

"BMA" best match average approach

Value

the value of the similarity of two diseases

Author(s)

Yue Deng<anfdeng@163.com>

References

[1] P. W. Lord, et al., "Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation," Bioinformatics, vol. 19, pp. 1275-83, Jul 1 2003.

[2] A. Schlicker, F. Domingues, J. Rahnenfuehrer, T. Lengauer, A new measure for functional similarity of gene products based on Gene Ontology, BMC Bioinformatics, 7, 302, 2006.

[3] James Z.Wang,Zhidian Du, et al. A new method to measure the semantic similarity of GO terms.Bioinformatics 2007,Vol 23,1274-1281.

See Also

getDiseaseListSim

Examples

```
getDiseaseSim("OMIM:101900","OMIM:102000")
```

---

getGeneListSim *Pairwise Similarity for a List of Genes*

Description

Given a list of genes, the function calculates the pairwise similarities for any two genes in the list using different strategies.

Usage

```
getGeneListSim(genelist,combinemethod="funSimMax",method="Resnik",
               ontology="PA",normalization=FALSE,normalizationmethod="Lin",
               verbose=FALSE)
```

Arguments

genelist	character vector of Entrez gene IDs
combinemethod	method to calculate the similarity between genes based on , one of "max", "mean", "funSimMax", "funSimAvg" and ""BMA
method	method to compute the similarity of HPO terms, one of "Resnik", "JiangConrath", "Lin", "simIC", "relevance" and "Wang"
ontology	the ontology used for similarity calculation
normalization	normalize similarities yes/no
normalizationmethod	one of "sqrt", "Lin" and "Tanimoto"
verbose	print out some information

Details

The method to calculate the pairwise disease similarity between gene products can either be:

"max" the maximum similarity between any two HPO terms

"mean" the average similarity between any two HPO terms1[1]

funSimMax the average of best matching HPO term similarities. Take the maximum of the scores achieved by assignments of HPO terms from gene 1 to gene 2 and vice versa. [2]

funSimAvg the average of best matching HPO term similarities. Take the average of the scores achieved by assignments of HPO terms from gene 1 to gene 2 and vice versa. [2]

"BMA" best match average approach [3]

Value

n*n similarity matrix (n = number of genes)

Author(s)

Yue Deng<anfdeng@163.com>

Examples

```
getGeneListSim(c("410","650","367"))
```

---

getGeneSim *Phenotype-based similarity for two genes base on HPO*

Description

Given two genes, the function calculates similarity of the genes based on HPO.

Usage

```
getGeneSim(gene1,gene2,combinemethod="funSimMax",method="Resnik",
           ontology="PA",normalization=FALSE,normalizationmethod="Lin",
           verbose=FALSE)
```

Arguments

gene1	one Entrez gene ID
gene2	another Entrez gene ID
combinemethod	method to calculate the similarity between genes based on , one of "max", "mean", "funSimMax", "funSimAvg" and "BMA"
method	method to compute the similarity of HPO terms, one of "Resnik", "JiangConrath", "Lin", "simIC", "relevance" and "Wang"

ontology	the ontology used to calculate the similarity, one of "PA"(Phenotypic Abnormality), "MI"(Mode of Inheritance) and "OC"(Onset and Clinical Course)
normalization	whether do the normalization
normalizationmethod	one of "sqrt", "Lin" and "Tanimoto"
verbose	print out some information

Details

The method to combine the similarities between HPO terms can either be:

"max" the maximum similarity between any two HPO terms

"mean" the average similarity between any two HPO terms1

"funSimMax" the average of best matching HPO term similarities. Take the maximum of the scores achieved by assignments of HPO terms from gene 1 to gene 2 and vice versa.

"funSimAvg" the average of best matching HPO term similarities. Take the average of the scores achieved by assignments of HPO terms from gene 1 to gene 2 and vice versa.

"BMA" best match average approach

Value

the value of the similarity of two genes

Author(s)

Yue Deng<anfdeng@163.com>

References

[1] P. W. Lord, et al., "Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation," Bioinformatics, vol. 19, pp. 1275-83, Jul 1 2003.

[2] A. Schlicker, F. Domingues, J. Rahnenfuehrer, T. Lengauer, A new measure for functional similarity of gene products based on Gene Ontology, BMC Bioinformatics, 7, 302, 2006.

[3] James Z.Wang,Zhidian Du, et al. A new method to measure the semantic similarity of GO terms.Bioinformatics 2007,Vol 23,1274-1281.

Examples

```
getGeneSim("6708","6710","funSimMax","Lin","PA",FALSE)
```

getSimWang *Semantic Similarity Between Two HPO Terms by Wang's Method*

Description

Given two HPO terms, this function will calculate the Wang's Semantic Similarity between them

Usage

```
getSimWang(term1, term2)
```

Arguments

term1	one HPO term
term2	another HPO term

Value

Semantic similarity.

Author(s)

Yue Deng <anfdeng@163.com>

References

[1] J. Z. Wang, Z. Du, R. Payattakool, P. S. Yu, and C.-F. Chen, "A new method to measure the semantic similarity of GO terms", Bioinformatics, vol. 23, no. 10, pp. 1274-1281, May. 2007.

Examples

```
getSimWang("HP:0000028","HP:0000033")
```

getTerm *Get information of HPO terms*

Description

Given a list of IDs of HPO term's, the function returns their names and synonyms.

Usage

```
getTerm(hpolist)
```

Arguments

hpolist	character vector of HPO IDs

Value

List of names and synonyms for each HPO ID.

Author(s)

Yue Deng<anfdeng@163.com>

See Also

calcTermSim

Examples

```
terms<-c("HP:0000002","HP:0000012")
result<-getTerm(terms)
print(result)
```

---

getTermAncestors *Get Ancestors of HPO Terms*

Description

Given a list of HPO terms, the function returns the ancestors of each term.

Usage

```
getTermAncestors(hpolist, verbose = FALSE)
```

Arguments

hpolist	character vector of HPO IDs
verbose	print out some information

Value

List with entry names for each HPO ID. Each entry contains a charcter vector with the ancestor HPO IDs

Author(s)

Yue Deng<anfdeng@163.com>

See Also

getTermOffsprings getTermChildren getTermParents

Examples

```
terms<-c("HP:0000002","HP:0002719")
result<-getTermAncestors(terms)
print(result)
```

---

getTermChildren *Get Direct Children of HPO terms*

Description

Given a list of HPO terms, the function returns the direct children of each term.

Usage

```
getTermChildren(hpolist, verbose = FALSE)
```

Arguments

hpolist	character vector of HPO IDs
verbose	print out some information

Value

List with entry names for each HPO ID. Each entry contains a character vector with the direct children of HPO IDs.

Author(s)

Yue Deng<anfdeng@163.com>

See Also

getTermOffsprings getTermParents getTermAncestors

Examples

```
terms<-c("HP:0000002","HP:0002719")
result<-getTermChildren(terms)
print(result)
```

`getTermListSim` *Similarity matrix for two sets of HPO terms*

Description

Given two sets of HPO terms, the function calculates similarity between two sets.

Usage

```
getTermListSim(anno1, anno2, combinemethod="funSimMax",
                  method="Resnik", IC, verbose=FALSE)
```

Arguments

`anno1`	one set of HPO terms
`anno2`	another set of HPO terms
`combinemethod`	method to calculate the similarity between diseases based on , one of "max", "mean", "funSimMax", "funSimAvg" and ""BMA
`method`	method to compute the similarity of HPO terms, one of "Resnik", "JiangConrath", "Lin", "simIC", "relevance", "GIC" and "Wang"
`IC`	IC used to calculate similarity between terms
`verbose`	print out some information

Details

The method to combine the similarities between HPO terms can either be:

"max" the maximum similarity between any two HPO terms

"mean" the average similarity between any two HPO terms1

"funSimMax" the average of best matching HPO term similarities. Take the maximum of the scores achieved by assignments of HPO terms from disease 1 to disease 2 and vice versa.

"funSimAvg" the average of best matching HPO term similarities. Take the average of the scores achieved by assignments of HPO terms from disease 1 to disease 2 and vice versa.

"BMA" best match average approach

Value

the value of the similarity of two sets of HPO terms

Author(s)

Yue Deng<anfdeng@163.com>

References

[1] P. W. Lord, et al., "Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation," Bioinformatics, vol. 19, pp. 1275-83, Jul 1 2003.

[2] A. Schlicker, F. Domingues, J. Rahnenfuehrer, T. Lengauer, A new measure for functional similarity of gene products based on Gene Ontology, BMC Bioinformatics, 7, 302, 2006.

[3] James Z.Wang,Zhidian Du, et al. A new method to measure the semantic similarity of GO terms.Bioinformatics 2007,Vol 23,1274-1281.

See Also

calcTermSim

Examples

```
.initialize()
IC<-get("termIC",envir=HPOSimEnv)
getTermListSim(c("HP:0000118", "HP:0000152", "HP:0000234", "HP:0000271"),
               c("HP:0000284", "HP:0000478", "HP:0000479", "HP:0000488"),
               combinemethod="funSimMax", method="Resnik", IC=IC, verbose=FALSE)
```

---

getTermOffsprings *Get Offspring of HPO terms*

Description

Given a list of HPO terms, the function returns the offsprings of each term.

Usage

```
getTermOffsprings(hpolist, verbose = FALSE)
```

Arguments

hpolist	character vector of HPO IDs
verbose	print out some information

Value

List with entry names for each HPO ID. Each entry contains a character vector with the offspring HPO IDs.

Author(s)

Yue Deng<anfdeng@163.com>

See Also

getTermChildren getTermParents getTermAncestors

Examples

```
terms<-c("HP:0000002","HP:0002719")
result<-getTermOffsprings(terms)
print(result)
```

`getTermParents` *Get Direct Parents of HPO Terms*

Description

Given a list of HPO terms, the function returns the direct parents of each term.

Usage

```
getTermParents(hpolist, verbose = FALSE)
```

Arguments

`hpolist`	character vector of HPO IDs
`verbose`	print out some information

Value

List with entry names for each HPO ID. Each entry contains a character vector with the direct parent of HPO IDs.

Author(s)

Yue Deng<anfdeng@163.com>

See Also

`getTermOffsprings` `getTermChildren` `getTermAncestors`

Examples

```
terms<-c("HP:0007408","HP:0006538")
result<-getTermParents(terms)
print(result)
```

HPODiseaseEnrichment *HPO enrichment analysis of Disease List*

Description

Given a list of disease, the function returns the enriched HPO terms.

Usage

```
HPODiseaseEnrichment(diseaselist,filter=5,cutoff=0.05,
                             background=getDiseaseDefaultBackground())
```

Arguments

diseaselist	character vector of disease OMIM IDs
filter	indicates that HPO terms must have at least 'filter' diseases annotated
cutoff	significant cutoff for HPO enrichment analysis
background	A character vector of diseases used as backgroud. Default is to use all the diseases annotated in HPO

Value

Return a data.frame object with 8 columns.Details are below:

"HPOID"	enriched HPO IDs
"annDiseaseNumber"	Disease number annotated to this HPO term in the inputed disease list
"annBgNumber"	Disease number in the inputed disease list
"diseaseNumber"	Disease number annotated to this HPO term in the backgroud list
"bgNumber"	Disease number in the backgroud list
"odds"	Calculated by $\frac{annDiseaseNumber/annBgNumber}{diseaseNumber/bgNumber}$
"pvalue"	siginicance of the hypergeometric test for this HPO term
"qvale"	multiple test correction value for pvalue using FDR

Author(s)

Yue Deng<anfdeng@163.com>

See Also

HPOGeneEnrichment

Examples

```
diseaselist<-c("OMIM:605685","OMIM:167100","OMIM:304200","OMIM:219300",cutoff=0.1)
res<-HPODiseaseEnrichment(diseaselist)
```

HPODiseaseNOASubNetEnrichment
HPO enrichment analysis of disease network using NOA

Description

Given a disease network, the function returns the enriched HPO terms using sub-net method of NOA.

Usage

```
HPODiseaseNOASubNetEnrichment(testfile,backgroundfile,filter=5,cutoff=0.05)
```

Arguments

testfile	CSV format disease network, each line of which is OMIM IDs of two diseases
backgroundfile	CSV format disease network used as the reference set, each line of which is OMIM IDs of two diseases
filter	indicates that HPO terms must have at least 'filter' edges annotated
cutoff	significant cutoff for HPO enrichment analysis

Value

Return a data.frame object with 8 columns.Details are below:

"HPOID"	enriched HPO IDs
"annEdgeNumber"	Edge number annotated to this HPO term in the inputed disease network
"annBgNumber"	Edge number in the inputed disease network
"edgeNumber"	Edge number annotated to this HPO term in the backgroud network which is chosen as all possible links in the test network
"bgNumber"	Edge number in the backgroud network
"odds"	Calculated by $\frac{annEdgeNumber/annBgNumber}{edgeNumber/bgNumber}$
"pvalue"	siginicance of the hypergeometric test for this HPO term
"qvale"	multiple test correction value for pvalue using FDR

Author(s)

Yue Deng<anfdeng@163.com>

See Also

HPODiseaseEnrichment HPODiseaseNOAWholeNetEnrichment

Examples

```
edges <- data.frame(node1=c("OMIM:275350","OMIM:212750","OMIM:212750",
                            "OMIM:275350","OMIM:212750","OMIM:260400"),
                    node2=c("OMIM:100050","OMIM:105600","OMIM:249270",
                            "OMIM:300751","OMIM:250250","OMIM:105600"))
graph<-igraph::graph.data.frame(edges,directed=FALSE)
write.csv(edges,file="HPODiseaseNOASubNetEnrichment-testnetwork.csv",row.names=FALSE)

edges <- data.frame(node1=c("OMIM:275350","OMIM:212750","OMIM:212750","OMIM:275350",
                            "OMIM:212750","OMIM:300835","OMIM:260400","OMIM:275350",
                            "OMIM:260400","OMIM:300835","OMIM:106230","OMIM:106240",
                            "OMIM:106250","OMIM:106260"),
                    node2=c("OMIM:100050","OMIM:105600","OMIM:249270","OMIM:300751",
                            "OMIM:250250","OMIM:300751","OMIM:250250","OMIM:250250",
                            "OMIM:105600","OMIM:105600","OMIM:106260","OMIM:106230",
                            "OMIM:106240","OMIM:106250"))
graph<-igraph::graph.data.frame(edges,directed=FALSE)
write.csv(edges,file="HPODiseaseNOASubNetEnrichment-backgroundnetwork.csv",row.names=FALSE)

HPODiseaseNOASubNetEnrichment("HPODiseaseNOASubNetEnrichment-testnetwork.csv",
                        "HPODiseaseNOASubNetEnrichment-backgroundnetwork.csv",cutoff=0.1)
```

`HPODiseaseNOAWholeNetEnrichment`
HPO enrichment analysis of disease network using NOA

Description

Given a disease network, the function returns the enriched HPO terms using whole-net method of NOA.

Usage

```
HPODiseaseNOAWholeNetEnrichment(file,filter=5,cutoff=0.05)
```

Arguments

`file`	CSV format disease network, each line of which is OMIM IDs of two diseases
`filter`	indicates that HPO terms must have at least 'filter' edges annotated
`cutoff`	significant cutoff for HPO enrichment analysis

Value

Return a data.frame object with 8 columns.Details are below:

`"HPOID"`	enriched HPO IDs
`"annEdgeNumber"`	Edge number annotated to this HPO term in the inputed disease network

"annBgNumber"	Edge number in the inputed disease network
"edgeNumber"	Edge number annotated to this HPO term in the backgroud network which is chosen as all possible links in the test network
"bgNumber"	Edge number in the backgroud network
"odds"	Calculated by $\frac{annEdgeNumber/annBgNumber}{edgeNumber/bgNumber}$
"pvalue"	siginicance of the hypergeometric test for this HPO term
"qvale"	multiple test correction value for pvalue using FDR

Author(s)

Yue Deng<anfdeng@163.com>

See Also

HPODiseaseEnrichment

Examples

```
edges <- data.frame(node1=c("OMIM:275350","OMIM:212750","OMIM:212750",
                            "OMIM:275350","OMIM:212750","OMIM:300835",
                            "OMIM:260400","OMIM:275350"),
                    node2=c("OMIM:100050","OMIM:105600","OMIM:249270",
                            "OMIM:300751","OMIM:250250","OMIM:300751",
                            "OMIM:250250","OMIM:250250"))
graph<-igraph::graph.data.frame(edges,directed=FALSE)
write.csv(edges,file="HPODiseaseNOAWholeNetEnrichment-testwork.csv",row.names=FALSE)
HPODiseaseNOAWholeNetEnrichment("HPODiseaseNOAWholeNetEnrichment-testwork.csv",cutoff=0.8)
```

HPOGeneEnrichment *HPO enrichment analysis of Gene List*

Description

Given a list of genes, the function returns the enriched HPO terms.

Usage

```
HPOGeneEnrichment(genelist,filter=5,cutoff=0.05,
                          background=getGeneDefaultBackground())
```

Arguments

genelist	character vector of gene names
filter	indicates that HPO terms must have at least 'filter' genes annotated
cutoff	significant cutoff for HPO enrichment analysis
background	A character vector of genes used as backgroud. Default is to use all the genes annotated in HPO

Value

Return a data.frame object with 8 columns.Details are below:

"HPOID"	enriched HPO IDs
"annGeneNumber"	Gene number annotated to this HPO term in the inputed gene list
"annBgNumber"	Gene number in the inputed gene list
"geneNumber"	Gene number annotated to this HPO term in the backgroud list
"bgNumber"	Gene number in the backgroud list
"odds"	Calculated by $\frac{annGeneNumber/annBgNumber}{geneNumber/bgNumber}$
"pvalue"	siginicance of the hypergeometric test for this HPO term
"qvale"	multiple test correction value for pvalue using FDR

Author(s)

Yue Deng<anfdeng@163.com>

See Also

HPODiseaseEnrichment

Examples

```
genelist<-c("6299","9241","1717","54880","85358")
res<-HPOGeneEnrichment(genelist)
print(res)
```

HPOGeneNOASubNetEnrichment

HPO enrichment analysis of gene network using NOA

Description

Given a gene network, the function returns the enriched HPO terms using sub-net method of NOA.

Usage

```
HPOGeneNOASubNetEnrichment(testfile,backgroundfile,filter=5,cutoff=0.05)
```

Arguments

testfile	CSV format gene network, each line of which is Entrez IDs of two genes
backgroundfile	CSV format gene network used as the reference set, each line of which is Entrez IDs of two genes
filter	indicates that HPO terms must have at least 'filter' edges annotated
cutoff	significant cutoff for HPO enrichment analysis

Value

Return a data.frame object with 8 columns.Details are below:

"HPOID"	enriched HPO IDs
"annEdgeNumber"	Edge number annotated to this HPO term in the inputed gene network
"annBgNumber"	Edge number in the inputed gene network
"edgeNumber"	Edge number annotated to this HPO term in the backgroud network which is chosen as all possible links in the test network
"bgNumber"	Edge number in the backgroud network
"odds"	Calculated by $\frac{annEdgeNumber/annBgNumber}{edgeNumber/bgNumber}$
"pvalue"	siginicance of the hypergeometric test for this HPO term
"qvale"	multiple test correction value for pvalue using FDR

Author(s)

Yue Deng<anfdeng@163.com>

See Also

HPOGeneEnrichment HPOGeneNOAWholeNetEnrichment

Examples

```
edges <- data.frame(node1=c("65018","65019","65055","65055","65125","65266"),
                    node2=c("65055","65055","65125","65266","65266","79001"))
graph<-igraph::graph.data.frame(edges,directed=FALSE)
write.csv(edges,file="HPOGeneNOASubNetEnrichment-testnetwork.csv",row.names=FALSE)

edges <- data.frame(node1=c("65018","65018","65018","65018","65018","65055","65055",
                      "65055","65055","65109","65109","65109","65125","65125","65266"),
                    node2=c("65055","65109","65125","65266","79001","65019","65125",
                      "65266","79001","65125","65266","79001","65266","79001","79001"))
graph<-igraph::graph.data.frame(edges,directed=FALSE)
write.csv(edges,file="HPOGeneNOASubNetEnrichment-backgroundnetwork.csv",row.names=FALSE)

HPOGeneNOASubNetEnrichment("HPOGeneNOASubNetEnrichment-testnetwork.csv",
                       "HPOGeneNOASubNetEnrichment-backgroundnetwork.csv",cutoff=0.4)
```

HPOGeneNOAWholeNetEnrichment
HPO enrichment analysis of gene network using NOA

Description

Given a gene interaction network, the function returns the enriched HPO terms using whole-net method of NOA.

Usage

```
HPOGeneNOAWholeNetEnrichment(file,filter=5,cutoff=0.05)
```

Arguments

file	CSV format gene interaction network, each line of which is Entrez IDs of two interacting genes/proteins
filter	indicates that HPO terms must have at least 'filter' edges annotated
cutoff	significant cutoff for HPO enrichment analysis

Value

Return a data.frame object with 8 columns.Details are below:

"HPOID"	enriched HPO IDs
"annEdgeNumber"	Edge number annotated to this HPO term in the inputed gene network
"annBgNumber"	Edge number in the inputed gene network
"edgeNumber"	Edge number annotated to this HPO term in the backgroud network which is chosen as all possible links in the test network
"bgNumber"	Edge number in the backgroud network
"odds"	Calculated by $\frac{annEdgeNumber/annBgNumber}{edgeNumber/bgNumber}$
"pvalue"	siginicance of the hypergeometric test for this HPO term
"qvale"	multiple test correction value for pvalue using FDR

Author(s)

Yue Deng<anfdeng@163.com>

See Also

HPOGeneEnrichment

Examples

```
edges <- data.frame(node1=c("581", "581", "581", "1499", "1499", "1630",
                            "60", "257", "999", "1282", "1723", "1760"),
                    node2=c("1499", "1630", "2033", "1630", "2033",
                            "2033", "581", "1499", "2033", "1630", "1760", "581"))
graph<-igraph::graph.data.frame(edges,directed=FALSE)
write.csv(edges,file="HPOGeneNOAWholeNetEnrichment-testwork.csv",row.names=FALSE)
HPOGeneNOAWholeNetEnrichment("HPOGeneNOAWholeNetEnrichment-testwork.csv",cutoff=0.1)
```

Index