Package 'DOSim'

August 7, 2010

Type Package

Title Disease Analysis toolkit for gene set; Computation of similarities between DO terms and disease similarity between gene products; Module detection among the gene set and multilayer annotation for each module.

Version 2.0

Date 2010-8-5

Author Jiang Li <riverlee2008@gmail.com>

Maintainer Jiang Li <riverlee2008@gmail.com>

Depends GOSim, SubpathwayMiner, dynamicTreeCut, moduleColor, RBGL, graph

Description This package implements multiple similarity measures for DO terms and gene products. It is aiming at disease analyis for gene sets. Modules of a gene set could be detected and further multilayer annoated on DO, GO and KEGG. We also provides users to conduct DO enrichment analysis and basic information fetching for DO.

License GPL (>= 2)

URL http://bioinfo.hrbmu.edu.cn/dosim/

LazyLoad yes

Repository CRAN

R topics documented:

DOSim-package 2
annoModule 3
detectModule 4
DOEnrichment 7
DOSimEnv 8
filterDO 8
getAncestors 9
getChildren 10

getDefaultBackground 11
getDisjCommAnc 11
getDoAnno 12
getDOGraph 13
getDoTerm 14
getGeneSim 15
getMinimumSubsumer 16
getOffsprings 17
getParents 18
getShortestPath 19
getTermSim 19
internal 21
saveAnnoModule 22
saveModule 23
viewModule 24

Index **25**

`DOSim-package` *DOSim package*

Description

This package implements multiple similarity measures for DO terms and gene products. It is aiming at disease analyis for gene sets. Modules of a gene set could be detected and further multilayer annoated on DO, GO and KEGG. We also provides users to conduct DO enrichment analysis and basic information fetching for DO.

Details

Package:	DOSim
Type:	Package
Version:	2.0
Date:	2010-8-5
License:	GPL (>= 2)
LazyLoad:	yes

Author(s)

Jiang Li

Maintainer: Jiang Li <riverlee2008@gmail.com>

Examples

```
################################
#example
terms<-c("DOID:1579","DOID:945")
tsim<-getTermSim(terms)

print(tsim)
```

`annoModule` *Multilayer annotation for each module*

Description

Multilayer annotation for each module from DO, GO, and KEGG, in order to explore the implied the functional meaning.

Usage

```
annoModule(object,dofilter=5,dolayer=5,docutoff=0.01,
      gocutoff=0.01,keggcutoff=0.01,goontologoy="BP",
      calculateMeanSim=TRUE)
```

Arguments

`object`	A list object which is the output of `detectModule`.
`dofilter`	indicate that there must be at least "dofilter" genes annotated on this do term, then it will be considered for analysis
`dolayer`	indicate that do terms beneath depth "dolayer" in the DAG of DO will be considered for analysis
`docutoff`	significant filter for do terms
`gocutoff`	significant filter for go terms
`keggcutoff`	significant filter for kegg pathways
`goontologoy`	which category of GO is used when conducting analysis for the module result, value is one of "BP","MF",and "CC"
`calculateMeanSim`	whether to calculate the mean similariy for each module

Details

The methodology for multilaper annotation is the enrichment analysis on DO, GO and KEGG.

Value

return a list object with size equal to the module number. Each item in the list contains all the information for a certain module, and it is also a list object contains 6 or 8 slots(is parameter 'calculateMeanSim' is set TRUE). They are: 1) DO, a data.frame object with four columns: ID, Name, pvalue and qvalue; 2) GO, a data.frame object with four columns: ID, Name, pvalue and qvalue; 3) KEGG, a data.frame object with four columns: ID, Name, pvalue and qvalue; 4) Genes, a character vector of genes covered by the module; 5) Size, the size of the module, which is equal to the number of genes covered by the module; 6) ModuleColor, color stands for the module; 7) MeanSimilarity, the average gene similarity for the module (if parameter calculateMeanSim is set TRUE); 8) pvalue, t-test significant for comparison of gene similarity in the module with gene similarity in the gene list (if parameter calculateMeanSim is set TRUE).

Author(s)

Jiang Li<<riverlee2008@gmail.com>>

See Also

detectModule, DOEnrichment.

Examples

```
##############################
#Examples

  #
  #data(obesity)
  #modules<-detectModule(obesity,method="tree",minClusterSize=10)
  #annoModule(modules,dolayer=NULL)
```

---

detectModule *Detect modules from gene set*

Description

Detect modules from the gene set, input is a gene similarity matrix or distance matirx which can be obtained by getGeneSim function.

Usage

```
detectModule(dat, isSimilarity=1, hierarchicalMethod="average", cutHeight = NULL, m
                        method = "hybrid", deepSplit = (ifelse(method=="hybrid", 1,
                        maxCoreScatter = NULL, minGap = NULL,
                        maxAbsCoreScatter = NULL, minAbsGap = NULL,
                        pamStage = TRUE, pamRespectsDendro = TRUE,
                        useMedoids = FALSE, maxDistToLabel = cutHeight,
                        respectSmallClusters = TRUE,
```

```
                    verbose = 2, indent = 0)
```

Arguments

dat	A symmetrical matrix object.Values can be either a similarity matrix or distance matrix, which can be obtained by getGeneSim function.
isSimilarity	Is the input data a similarity matrix or distance matrix
hierarchicalMethod	method can be "average","complete",etc., when using hclust to calculate the dendrogram
cutHeight	Maximum joining heights that will be considered. For `method=="tree"` it defaults to 0.99. For `method=="hybrid"` it defaults to 99 percentile and the maximum of the joining heights on the dendrogram.
minClusterSize	Minimum cluster size.
method	Chooses the method to use. Recognized values are "constant", "hybrid" and "tree".
deepSplit	For method "hybrid", can be either logical or integer in the range 0 to 4. For method "tree", must be logical. In both cases, provides a rough control over sensitivity to cluster splitting. The higher the value (or if `TRUE`), the more and smaller clusters will be produced. For the "hybrid" method, a finer control can be achieved via `maxCoreScatter` and `minGap` below.
maxCoreScatter	Only used for method "hybrid". Maximum scatter of the core for a branch to be a cluster, given as the fraction of `cutHeight` relative to the 5th percentile of joining heights. See Details.
minGap	Only used for method "hybrid". Minimum cluster gap given as the fraction of the difference between `cutHeight` and the 5th percentile of joining heights.
maxAbsCoreScatter	Only used for method "hybrid". Maximum scatter of the core for a branch to be a cluster given as absolute heights. If given, overrides `maxCoreScatter`.
minAbsGap	Only used for method "hybrid". Minimum cluster gap given as absolute height difference. If given, overrides `minGap`.
pamStage	Only used for method "hybrid". If TRUE, the second (PAM-like) stage will be performed.
pamRespectsDendro	Logical, only used for method "hybrid". If `TRUE`, the PAM stage will respect the dendrogram in the sense that objects and small clusters will only be assigned to clusters that belong to the same branch that the objects or small clusters being assigned belong to.
useMedoids	Only used for method "hybrid" and only if `labelUnlabeled==TRUE`. If TRUE, the second stage will be use object to medoid distance; if FALSE, it will use average object to cluster distance. The default (FALSE) is recommended.

`maxDistToLabel`
: Only used for method "hybrid" and only if `labelUnlabeled==TRUE`. Maximum object distance to closest cluster that will result in the object assigned to that cluster.

`respectSmallClusters`
: Only used for method "hybrid" and only if `labelUnlabeled==TRUE`. If TRUE, branches that failed to be clusters in stage 1 only because of insufficient size will be assigned together in stage 2. If FALSE, all objects will be assigned individually.

`verbose`
: Controls the verbosity of the output. 0 will make the function completely quiet, values up to 4 gradually increase verbosity.

`indent`
: Controls indentation of printed messages (see `verbose` above). Each unit adds two spaces before printed messages; useful when several functions' output is to be nested.

Details

This function use the output of `getGeneSim`, and further apply hierarchical clustering method on it, use three branch cutting methods to detect modules. Details for the branch cutting methods please see cutreeDynamic[1].

The input data will be scaled to 0-1 first using Min-Max method, which can be formulated as (x-min(x)/(max(x)-min(x))

The output of this function can be further used as the input of saveModule and viewModule

Value

A vector of numerical labels giving assignment of objects to modules. Unassigned objects are labeled 0, the largest module has label 1, next largest 2 etc. a list object contains three slots. They are: 1) "dendrogram", the dendrogram calculated by `hclust` on dat

2) "module",a vector of numerical labels giving assignment of objects to modules. Unassigned objects are labeled 0, the largest module has label 1, next largest 2 etc.

3) "simmatrix", the similarity matrix of input data after Min-Max processing. If the input dat is a distance matrix, it convert to similarity matrix by 1-Min-Max(dat)

Author(s)

Jiang Li<<riverlee2008@gmail.com>>

References

Langfelder P, Zhang B, Horvath S, 2007. http://www.genetics.ucla.edu/labs/horvath/CoexpressionNetwork/BranchCutting

See Also

`hclust`, `cutreeDynamic`,`getGeneSim`.

Examples

```
##############################
#Examples

  data(obesity)
  modules<-detectModule(obesity,method="tree",minClusterSize=10)
```

`DOEnrichment` *DO enrichment analysis*

Description

This function performs DO enrichment analysis using Hypergeometric Test.

Usage

```
DOEnrichment(genelist,filter=5,cutoff=0.05,layer=5,backgroud=getDefaultBackground()
```

Arguments

`genelist`	character vector of Entrez Gene IDs
`filter`	indicates that DO terms must have at least 'filter' genes annotated
`cutoff`	significant cutoff for DO enrichment analysis
`layer`	Control for DO terms, this means only those beneath the certain depth defined by users are considered for the analysis.
`backgroud`	A character vector of Entrez Gene IDs used as backgroud. Default is to use all the genes annotated in DO

Details

The methodology is Hypergeometric Test.

Value

Return a data.frame object with 9 columns.Details are below:

`"DOID"`	enriched DO ID name
`"Term"`	enriched DO Term name
`"annGeneNumber"`	Gene number annotated to this DO term in the inputed gene list
`"annBgNumber"`	Gene number in the inputed gene list
`"geneNumber"`	Gene number annotated to this DO 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 DO term
"qvale"	multiple test correction value for pvalue using FDR

Author(s)

Jiang Li<<riverlee2008@gmail.com>>

Examples

```
##############################
#Examples

  genelist=as.character(1:100)
  res<-DOEnrichment(genelist,filter=50)
  print(res)
```

DOSimEnv *Disease Ontology enviroment object*

Description

Disease Ontology enviroment object used for other functions

Usage

```
data(DOSimEnv)
```

Source

The original data is came from John D.Osborne's work.URL:http://projects.bioinformatics.northwestern.edu/do_rif/

Examples

```
data(DOSimEnv)

ls(DOSimEnv)
```

filterDO *Filter DO*

Description

Filter out genes from a list not mapping to the disease ontology.

Usage

```
filterDO(genelist)
```

Arguments

genelist	character vector of Entrez gene IDs

Details

Filter out genes from a list not mapping to the disease ontology,and return a list which the genes have DO term annotations in the disease ontology.

Value

List with items

"genename"	gene ID
"annotation"	character vector of DO IDs mapping to the gene

Author(s)

Jiang Li <riverlee2008@gmail.com>

Examples

```
##################################
#Example
genelist<-1:10
res<-filterDO(genelist)
print(res)
```

getAncestors *Get a list of all ancestors associated to each DO term*

Description

Returns the list of all ancestors associated to each DO term.

Usage

```
getAncestors(dolist, verbose = TRUE)
```

Arguments

dolist	character vector of DO IDs
verbose	print out some information

Value

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

Author(s)

Jiang Li<<riverlee2008@gmail.com>>

See Also

getOffsprings, getChildren, getParents

Examples

```
################################
#Example

terms<-c("DOID:934","DOID:1579")
res<-getAncestors(terms)
print(res)
```

getChildren *Get a list of all direct children of each DO term*

Description

Returns the list of all direct children associated to each DO term.

Usage

```
getChildren(dolist, verbose = TRUE)
```

Arguments

dolist	character vector of DO IDs
verbose	print out some information

Value

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

Author(s)

Jiang Li<<riverlee2008@gmail.com>>

See Also

getOffsprings, getParents, getAncestors

Examples

```
################################
#Example

terms<-c("DOID:934","DOID:1579")
res<-getChildren(terms)
print(res)
```

`getDefaultBackground` *Get a vector of genes annotated by current DO*

Description

Get a vector of genes annotated by current DO .

Usage

```
getDefaultBackground()
```

Value

return a character vector of genes which are the genes annotated by current version

Author(s)

Jiang Li <<riverlee2008@gmail.com>>

`getDisjCommAnc` *Get disjoint common ancestors.*

Description

Returns the DO terms representing the disjoint common ancestors of two DO terms.

Usage

```
getDisjCommAnc(term1, term2)
```

Arguments

`term1`	DO term 1
`term2`	DO term 2

Value

Character vector of DO terms

Author(s)

Jiang Li<<riverlee2008@gmail.com>>

References

Couto, F.; Silva, M. & Coutinho, P., Semantic Similarity over the Gene Ontology: Family Correlation and Selecting Disjunctive Ancestors, Conference in Information and Knowledge Management, 2005

Examples

```
getDisjCommAnc("DOID:934","DOID:95")
```

`getDoAnno` *Get gene list associated to each DO term*

Description

Get gene list associated to each DO term

Usage

```
getDoAnno(dolist)
```

Arguments

`dolist`	character vector of DO IDs

Value

List with entry names for each DO ID. Each entry contains a character vector with associated Entrez Gene IDs.

Author(s)

Jiang Li<<riverlee2008@gmail.com>>

See Also

`getDoTerm`

Examples

```
################################
#Example

terms<-c("DOID:934","DOID:1579")
res<-getDoAnno(terms)
print(res)
```

getDOGraph *Get DO graph with specified DO terms at its leave.*

Description

The function getDOGraph returns a graphNEL object representing the DO graph with leaves specified in the argument.

Usage

```
getDOGraph(term, prune = Inf)
```

Arguments

term	character vector of DO term IDs
prune	do not show the complete graph, but prune it after the specified number of ancestors

Value

graphNEL object(s)

Author(s)

Jiang Li<<riverlee2008@gmail.com>>

Examples

```
if(require(graph)){
g<-getDOGraph(c("DOID:95","DOID:8"))
if(require(Rgraphviz)){
plot(g)
}
}
```

getDoTerm *Get DO term's name*

Description

Returns the list of DO term's name associated to each DO ID.

Usage

```
getDoTerm(dolist)
```

Arguments

dolist	character vector of DO IDs

Value

List with entry names for each DO ID. Each entry contains a character represents DOID's term name.

Author(s)

Jiang Li<<riverlee2008@gmail.com>>

See Also

getDoAnno

Examples

```
################################
#Example

terms<-c("DOID:934","DOID:1579")
res<-getDoTerm(terms)
print(res)
```

---

getGeneSim *Compute disease similarity for genes base on DO*

Description

Calculate the pairwise disease similarities for a list of genes using different strategies

Usage

```
getGeneSim(genelist, similarity="funSimMax", similarityTerm="relevance", normalizat
```

Arguments

genelist	character vector of Entrez gene IDs
similarity	method to calculate the disease similarity between gene products
similarityTerm	method to compute the similarity of DO terms
normalization	normalize similarities yes/no
method	"sqrt": normalize sim(x,y) <- sim(x,y)/sqrt(sim(x,x)*sim(y,y)); "Lin": normalize sim(x,y) <- 2*sim(x,y)/(sim(x,x) + sim(y,y)); "Tanimoto": normalize sim(x,y) <- sim(x,y)/(sim(x,x) + sim(y,y) - sim(x,y)).
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 DO terms

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

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

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

"BMA" best match average approach [3]

Value

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

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

getTermSim

Examples

```
####################################
#Example
genelist=1:10
gsim<-getGeneSim(genelist)
print(gsim)
```

---

getMinimumSubsumer *Compute minimum subsumer (the most information common ancestor:MICA) of two DO terms*

Description

Returns the minimum subsumer(the common ancestor having the maximal information content) of two DO terms

Usage

```
getMinimumSubsumer(term1, term2)
```

Arguments

term1	DO term 1
term2	DO term 2

Details

The result is computed base on current disease ontology

Value

DO term representing the minmum subsumber. If there is no minumum subsumer,the result is the string "NA".

Author(s)

Jiang Li<<riverlee2008@gmail.com>>

References

P. Resnik, Using Information Content to evaluate semantic similarity in a taxonomy, Proc. 14th Int. Conf. Artificial Intel., 1995

Examples

```
############################
#Example
term1="DOID:8"
term2="DOID:95"
getMinimumSubsumer(term1,term2)
```

---

getOffsprings *Get all offspring associated with each DO term*

Description

Returns the list of all offspring associated to each DO term.

Usage

```
getOffsprings(dolist, verbose = TRUE)
```

Arguments

dolist	character vector of DO IDs
verbose	print out some information

Value

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

Author(s)

Jiang Li<<riverlee2008@gmail.com>>

See Also

getChildren, getParents, getAncestors

Examples

```
################################
#Example

terms<-c("DOID:934","DOID:1579")
res<-getOffsprings(terms)
print(res)
```

---

getParents *Get direct parents for each DO term*

Description

Returns a list of all direct parents associated to each DO term.

Usage

```
getParents(dolist, verbose = TRUE)
```

Arguments

dolist	character vector of DO IDs
verbose	print out some information

Value

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

Author(s)

Jiang Li<<riverlee2008@gmail.com>>

See Also

getOffsprings, getChildren, getAncestors

Examples

```
################################
#Example

terms<-c("DOID:934","DOID:1579")
res<-getParents(terms)
print(res)
```

---

getShortestPath *Get the shortest path between two terms*

Description

Get the shortest path between two terms.

Usage

```
getShortestPath(term1, term2)
```

Arguments

term1	DO term 1
term2	DO term 2

Value

return the shortest path between two terms, if two term are not connect, the return value is Inf

Author(s)

Jiang Li<<riverlee2008@gmail.com>>

Examples

```
#####################################
#example
term1="DOID:8"
term2="DOID:5"
getShortestPath(term1,term2)
```

`getTermSim` *Get pairwise DO term similarities.*

Description

Returns the pairwise similarities between DO terms based on different methods.

Usage

```
getTermSim(termlist, method = "relevance", verbose = TRUE)
```

Arguments

`termlist`	character vector of DO terms
`method`	one ot the supported methods for DO term similairty(see below)
`verbose`	print out various information or not

Details

Currently the following methods for computing DO term similarities are implemented:

"Resnik" information content of minimum subsumer (ICms) [1]

"JiangConrath" $1 - \min(1, IC(term1) - 2ICms + IC(term2))$[2]

"Lin" $\frac{2ICms}{(IC(term1)+IC(term2))}$ [3]

"CoutoResnik" average information content of common disjunctive ancestors of `term1` and `term2` (ICshare) [4]

"CoutoJiangConrath" $1 - \min(1, IC(term1) - 2ICshare + IC(term2))$ [4]

"CoutoLin" $\frac{2ICshare}{(IC(term1)+IC(term2))}$ [4]

"relevance" sim_Lin * (1 - exp(-ICms)) [5]

"GIC" summed information content of common ancestors divided by summed information content of all ancestors of term1 and term2 [7]

"simIC" sim_Lin * (1 - 1/(1 + ICms)) [7]

"Wang" $Sim(term1, term2) = \frac{\sum_{t \in T_{term1} \cap T_{term2}} (S_{term1}(t) + S_{term2}(t))}{SV(term1)+SV(term2)}$ [8]

Value

n x n matrix (n = number of DO terms) with similarities between DO terms.

Note

All calculations use normalized information contents for each DO term. Normalization is achieved by dividing each information content by the maximum information content.

Author(s)

Jiang Li<<riverlee2008@gmail.com>>

References

[1] P. Resnik, Using Information Content to evaluate semantic similarity in a taxonomy, Proc. 14th Int. Conf. Artificial Intel., 1995

[2] J. Jiang, D. Conrath, Semantic Similarity based on Corpus Statistics and Lexical Taxonomy, Proc. Int. Conf. Research in Comp. Ling., 1998

[3] D. Lin, An Information-Theoretic Definition of Similarity, Proc. 15th Int. Conf. Machine Learning, 1998

[4] Couto, F.; Silva, M. & Coutinho, P., Semantic Similarity over the Gene Ontology: Family Correlation and Selecting Disjunctive Ancestors, Conference in Information and Knowledge Management, 2005

[5] 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.

[6] C. Pesquita, D. Faria, H. Bastos, A. Falcao, F. Couto, Evaluating GO-based Semantic Similarity Measures, In: Proc. 10th Annual Bio-Ontologies Meeting 2007, 37 - 40, 2007

[7] B. Li, J. Wang, A. Feltus, J. Zhou, F. Luo, Effectively Integrating Information Content and Structural Relationship to Improve the GO-based Similarity Measure Between Proteins, BMC Bioinformatics, 2009.

[8] 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

getMinimumSubsumer, getDisjCommAnc

Examples

```
################################
#Example
getTermSim(c("DOID:8","DO:1117"),method="Lin")
```

---

internal *internal functions*

Description

internal functions: do not call these functions directly.

Usage

```
various
```

Arguments

various

Value

various

Author(s)

Jiang Li <<riverlee2008@gmail.com>>

`saveAnnoModule` *Detect modules from gene set*

Description

Detect modules from the gene set, input is a gene similarity matrix or distance matirx which can be obtained by getGeneSim function.

Usage

```
saveAnnoModule(annotatedModule,filename=NULL,sep=",")
```

Arguments

`annotatedModule`	A list object which is the output of `annoModule`
`filename`	Output filename, if filename is not set, it will be named as "module-annotation-year-month-day.txt" format
`sep`	separate character used in the output file

Author(s)

Jiang Li<<riverlee2008@gmail.com>>

See Also

`detectModule`, `annoModule`.

Examples

```
##############################
#Examples

  #data(obesity)
  #modules<-detectModule(obesity,method="tree",minClusterSize=10)
  #anno<-annoModule(modules)
  #saveAnnoModule(anno)
```

saveModule *Save the module result*

Description

Save the module result to files in a user friendly format.

Usage

```
saveModule(object,filename=NULL,sep="\t",iswriteout=TRUE)
```

Arguments

object	A list object which is the output of detectModule.
filename	Output filename, if filename is not set, it will be named as "module-year-month-day.txt" format
sep	separate character used in the output file
iswriteout	indicate whether write the result into file

Details

This function is to organize the module result in a user friendly format, more readable for users.

Value

Return a data.frame with three columns named as 1) module; 2) color; 3) genes. If parameter iswriteout is set TRUE, it will write the result into a file.

Author(s)

Jiang Li<<riverlee2008@gmail.com>>

See Also

detectModule, viewModule.

Examples

```
##############################
#Examples
 #
 # obesitysim<-data(obesity)
 # modules<-detectModule(obesitysim,method="tree",minClusterSize=10)
 # saveModule(modules)
```

viewModule *Display modules*

Description

Display modules from the output of `detectModule`

Usage

```
viewModule(object,main="Hierarchical dendrogram and module colors",...)
```

Arguments

`object`	A list object which is the output of `detectModule`.
`main`	The main title for the graph
`...`	other parameters set to display the graph

Details

visualize the modules detect by `detectModule` in a graph format.

Author(s)

Jiang Li<<riverlee2008@gmail.com>>

See Also

`detectModule`.

Examples

```
##############################
#Examples

  data(obesity)
  modules<-detectModule(obesity,method="tree",minClusterSize=10)
  viewModule(modules)
```

Index

∗Topic **\textasciitildekwd1**
annoModule, 2
detectModule, 4
DOEnrichment, 6
filterDO, 8
getAncestors, 9
getChildren, 10
getDisjCommAnc, 11
getDoAnno, 12
getDOGraph, 13
getDoTerm, 13
getGeneSim, 14
getMinimumSubsumer, 15
getOffsprings, 16
getParents, 17
getShortestPath, 18
getTermSim, 19
saveAnnoModule, 21
saveModule, 22
viewModule, 23
∗Topic **\textasciitildekwd2**
annoModule, 2
detectModule, 4
DOEnrichment, 6
filterDO, 8
getAncestors, 9
getChildren, 10
getDisjCommAnc, 11
getDoAnno, 12
getDOGraph, 13
getDoTerm, 13
getGeneSim, 14
getMinimumSubsumer, 15
getOffsprings, 16
getParents, 17
getShortestPath, 18
getTermSim, 19
saveAnnoModule, 21
saveModule, 22
viewModule, 23
∗Topic **datasets**
DOSimEnv, 8
∗Topic **file**
getDefaultBackground, 11
internal, 20
∗Topic **package**
DOSim-package, 2

annoModule, 2, *21*

calcTermSim *(internal)*, 20
cutreeDynamic, *5, 6*

detectModule, *3*, 4, *21–23*
DOEnrichment, *3*, 6
DOGenes *(getDefaultBackground)*, 11
DOGraph *(internal)*, 20
DOSim *(DOSim-package)*, 2
DOSim-package, 2
DOSimEnv, 8

filterDO, 8

getAncestors, 9, *10*, *17, 18*
getChildren, *9*, 10, *17, 18*
getDefaultBackground, 11
getDisjAnc *(internal)*, 20
getDisjCommAnc, 11, *20*
getDisjCommAncSim *(internal)*, 20
getDoAnno, 12, *14*
getDOGraph, 13
getDoTerm, *12*, 13
getGeneSim, *4*, *6*, 14, *21*
getGIC *(internal)*, 20
getGSim *(internal)*, 20
getMinimumSubsumer, 15, *20*
getOffsprings, *9, 10*, 16, *18*
getParents, *9, 10*, 17, *17*
getShortestPath, 18
getSimLch *(internal)*, 20

getSimPath *(internal)*, 20
getTermSim, *15*, 19

hclust, *4*, *6*

initialize *(internal)*, 20
internal, 20

normalize.kernel *(internal)*, 20

precomputeTermSims *(internal)*, 20

saveAnnoModule, 21
saveModule, *6*, 22

viewModule, *6*, *22*, 23