Package ‘RDAVIDWebService’
November 24, 2018

Type Package

Title An R Package for retrieving data from DAVID into R objects using Web Services API.

Version 1.20.0

Date 2014-04-15

Author Cristobal Fresno and Elmer A. Fernandez

Maintainer Cristobal Fresno <cfresno@bdmg.com.ar>

Description Tools for retrieving data from the Database for Annotation, Visualization and Integrated Discovery (DAVID) using Web Services into R objects. This package offers the main functionalities of DAVID website including: i) user friendly connectivity to upload gene/background list/s, change gene/background position, select current specie/s, select annotations, etc. ii) Reports of the submitted Gene List, Annotation Category Summary, Gene/Term Clusters, Functional Annotation Chart, Functional Annotation Table

License GPL (>=2)


Imports Category, GO.db, RBGL, rJava

Depends R (>= 2.14.1), methods, graph, GOstats, ggplot2

Collate 'DAVIDdemo-ids.R' 'DAVIDdemo-geneList.R'
'DAVIDdemo-annotationSummary.R'
'DAVIDdemo-functionalAnnotationChart.R'
'DAVIDdemo-annotationTable.R' 'DAVIDdemo-clusterReport.R'
'DAVIDResult-class.R' 'DAVIDGenes-class.R'
'DAVIDFunctionalAnnotationChart-class.R' 'DAVIDCluster-class.R'
'DAVIDGeneCluster-class.R' 'DAVIDTermCluster-class.R'
'DAVIDFunctionalAnnotationTable-class.R' 'DAVIDGODag-class.R'
'DAVIDWebService-class.R' 'DAVIDClasses-show.R'
'DAVIDClasses-ids.R' 'DAVIDClasses-plot2D.R'
'DAVIDClasses-summary.R' 'DAVIDClasses-genes.R'
'DAVIDClasses-categories.R' 'DAVIDResult-getters.R'
'DAVIDGenes-methods.R' 'DAVIDCluster-methods.R'
'DAVIDFunctionalAnnotationTable-methods.R'
'DAVIDGODag-methods.R' 'DAVIDWebService-accessors.R'
'DAVIDWebService-methods.R' 'DAVIDWebService-reports.R'
'DAVIDClasses-constructor.R' 'RDAVIDWebService-package.R'


**Suggests**  Rgraphviz  
**Installable Everywhere**  yes  
**biocViews**  Visualization, DifferentialExpression, GraphAndNetwork  
**git_url**  https://git.bioconductor.org/packages/RDAVIDWebService  
**git_branch**  RELEASE_3_8  
**git_last_commit**  87bd43b  
**git_last_commit_date**  2018-10-30  
**Date/Publication**  2018-11-23  

---  

**R topics documented:**

- DAVIDWebService-package ........................................ 3  
- annotationSummary1 .............................................. 3  
- annotationTable1 ................................................. 4  
- categories ...................................................... 5  
- cluster .......................................................... 7  
- DAVIDCluster-class ............................................. 9  
- DAVIDFunctionalAnnotationChart-class ......................... 10  
- DAVIDFunctionalAnnotationTable-class ......................... 11  
- DAVIDGeneCluster-class ...................................... 14  
- DAVIDGenes ..................................................... 15  
- DAVIDGenes-class .............................................. 20  
- DAVIDGODag-class ........................................... 21  
- DAVIDResult-class ........................................... 23  
- DAVIDTermCluster-class .................................... 25  
- DAVIDWebService-class .................................. 26  
- demoList1 .................................................... 29  
- funChart1 ...................................................... 30  
- geneCluster1 ................................................. 31  
- geneList1 ..................................................... 32  
- genes .......................................................... 33  
- getGeneCategoriesReport ................................... 35  
- ids ............................................................. 39  
- is.connected ................................................... 42  
- plot2D .......................................................... 49  
- setEmail ...................................................... 52  
- show ........................................................... 54  
- species ....................................................... 55  
- subset .......................................................... 56  
- summary ...................................................... 58  
- terms .......................................................... 60  
- type ............................................................ 63  

---  

**Index** 64
**Description**

Tools for retrieving data from the Database for Annotation, Visualization and Integrated Discovery (DAVID) using Web Services into R objects. This package offers the main functionalities of DAVID website including:

**Connectivity:** upload gene/background list/s, change gene/background position, select current specie/s, select annotations, etc. from R.

**Exploration:** native R objects of submitted Gene List, Annotation Category Summary, Gene/Term Clusters, Functional Annotation Chart and Functional Annotation Tables. In addition it offers the usual many-genes-to-many-terms visualization and induced Gene Ontology direct acyclic graph GOstats-based conversion method, in order to visualize GO structure.

**Author(s)**

Cristobal Fresno <cristobalfresno@gmail.com> and Elmer A. Fernandez <elmerfer@gmail.com>

**References**

1. The Database for Annotation, Visualization and Integrated Discovery (david.abcc.ncifcrf.gov)

**annotationSummary1**

*DAVID’s website annotation summary example files*

**Description**

These datasets correspond to the unfolded main summary categories data obtained in the Annotation Summary Results page in the Database for Annotation, Visualization and Integrated Discovery (DAVID) website, using as input file, the ones provided for demo purposes (demoList1 or demoList2) with default options. No statistical analysis is performed on these results.
Usage

data(translationSummary1)

data(translationSummary2)

Format

translationSummary1/2 are data.frame for demoList1/2 input ids, respectively, with the following columns.

**Main.Category** factor with the main categories used in the present analysis.

**ID** integer to identify the annotation category.

**Name** character with the name of category (the ones available in getAllAnnotationCategoryNames function).

**X.** numeric with the percentage of the gene list ids present in the term.

**Count** integer with the number of ids of the gene list that belong to this term.

Author(s)

Cristobal Fresno and Elmer A Fernandez

References

1. The Database for Annotation, Visualization and Integrated Discovery (davidgeneList.abcc.ncifcrf.gov)


3. DAVID Help page http://david.abcc.ncifcrf.gov/helps/functional_annotation.html#summary

See Also

Other DataExamples: demoList1, demoList2, geneList1, geneList2

---

**annotationTable1**  
*DAVID’s website functional annotation table example files*

Description

These datasets correspond to the Functional Annotation Table report obtained in the Database for Annotation, Visualization and Integrated Discovery (DAVID) website, using as input file, the ones provided for demo purposes (demoList1 or demoList2) for GOTERM_BP_ALL, GOTERM_MF_ALL and GOTERM_CC_ALL categories. No statistical analysis is performed on these results.
**categories**

**Usage**

```
data((annotationTable1)

data(annotationTable2)
```

**Format**

`annotationTable1/2` are `data.frame` for `demoList1/2` input ids, respectively, with the following columns.

- **Gene**: Three columns with the same data included in Gene List Report (ID, Gene.Name and Species) but coding for DAVID ID, i.e., comma separated character with input ids if, two or more stands for the same gene.

- **Annotation**: As many columns as Annotation Categories were used. In each column, a comma separated style is used to delimitate the different terms where evidence reported for DAVID ID record.

**Author(s)**

Cristobal Fresno and Elmer A Fernandez

**References**

1. The Database for Annotation, Visualization and Integrated Discovery ([davidgeneList.abcc.ncifcrf.gov](http://davidgeneList.abcc.ncifcrf.gov))

---

**Description**

Obtain ids related information, according to the given function call (see Values).

**Usage**

```
categories(object)
```

...
6

categories

**Arguments**

object  DAVIDWebService class object. Possible values are: DAVIDFunctionalAnnotationChart or DAVIDFunctionalAnnotationTable.

**Value**

according to the call, one of the following objects can be returned:

DAVIDFunctionalAnnotationChart
factor vector of the “Category” column.

DAVIDFunctionalAnnotationTable
character vector with the name of available main categories in the dictionary/membership.

**Author(s)**

Cristobal Fresno and Elmer A Fernandez

**See Also**

Other DAVIDFunctionalAnnotationChart: DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart-class, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDGODag, DAVIDGODag, DAVIDGeneCluster, DAVIDGeneCluster, DAVIDGenes, DAVIDGenes, DAVIDGenes, DAVIDTermCluster, DAVIDTermCluster, as, as, as, ids, ids, ids, ids, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, plot2D, plot2D, plot2D, plot2D, plot2D

Other DAVIDFunctionalAnnotationTable: DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable-class, DAVIDGODag, DAVIDGODag, DAVIDGeneCluster, DAVIDGeneCluster, DAVIDGenes, DAVIDGenes, DAVIDGenes, DAVIDGenes, DAVIDTermCluster, DAVIDTermCluster, as, as, as, dictionary, dictionary, genes, genes, genes, genes, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, membership, membership, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D, subset, subset

**Examples**

```r
{
##DAVIDFunctionalAnnotationChart example
##Load the Functional Annotation Chart file report for the input demo
##file 2, using data function. Then, create a DAVIDFunctionalAnnotationChart
##object using the loaded data.frame funChart2.
data(funChart2)
davidFunChart2<-DAVIDFunctionalAnnotationChart(funChart2)

##In Addition to the usual data.frame accessors, the user can inspect the
##main categories used in the analysis.
categories(davidFunChart2)

##DAVIDFunctionalAnnotationTable example
##Load the Functional Annotation Table file report for the input demo
##file 1, using data function. Then, create a DAVIDFunctionalAnnotationTable
##object using the loaded data.frame annotationTable1.
data( annotationTable1)
```
cluster7

davidFunTable<-DAVIDFunctionalAnnotationTable(annotationTable1)

##Now, the user can inspect the main categories used in the analysis.
categories(davidFunTable1)

Methods for DAVIDCluster class object

Description
Obtain DAVIDCluster related information, according to the given function call (see Values).

Usage

cluster(object)

## S4 method for signature 'DAVIDCluster'
cluster(object)

enrichment(object)

## S4 method for signature 'DAVIDCluster'
enrichment(object)

members(object)

## S4 method for signature 'DAVIDCluster'
members(object)

Arguments

object DAVIDCluster class object.

Value

according to the call, one of the following objects can be returned:

cluster list with DAVIDCluster object slot.
enrichment numeric vector with DAVID cluster’s enrichment score.
members list with DAVID Cluster’s members.

Author(s)

Cristobal Fresno and Elmer A Fernandez
See Also

Other DAVIDCluster: DAVIDCluster-class, dictionary, dictionary, membership, membership, subset, subset, summary, summary, summary, summary
Other DAVIDCluster: DAVIDCluster-class, dictionary, dictionary, membership, membership, subset, subset, summary, summary, summary, summary
Other DAVIDCluster: DAVIDCluster-class, dictionary, dictionary, membership, membership, subset, subset, summary, summary, summary, summary

Examples

```r
# DAVIDGeneCluster example:
# Load the Gene Functional Classification Tool file report for the
# input demo list 1 file to create a DAVIDGeneCluster object.
setwd(tempdir())
fileName <- system.file("files/geneClusterReport1.tab.tar.gz",
package="RDAVIDWebService")
untar(fileName)
davidGeneCluster1 <- DAVIDGeneCluster(untar(fileName, list=TRUE))
davidGeneCluster1
## Now we can invoke DAVIDCluster ancestor functions to inspect the report
## data of each cluster. For example, we can call summary to get a general
## idea, and then inspect the cluster with the higher Enrichment Score, to see
## which members belong to it, etc. or simply, returning the whole cluster as
## a list with EnrichmentScore and Members.
summary(davidGeneCluster1)
higherEnrichment <- which.max(enrichment(davidGeneCluster1))
clusterGenes <- members(davidGeneCluster1)[[higherEnrichment]]
wholeCluster <- cluster(davidGeneCluster1)[[higherEnrichment]]

# DAVIDTermCluster example:
# Load the Gene Functional Classification Tool file report for the
# input demo file 2 to create a DAVIDGeneCluster object.
setwd(tempdir())
fileName <- system.file("files/termClusterReport2.tab.tar.gz",
package="RDAVIDWebService")
untar(fileName)
davidTermCluster2 <- DAVIDTermCluster(untar(fileName, list=TRUE))
davidTermCluster2
## Now we can invoke DAVIDCluster ancestor functions to inspect the report
## data of each cluster. For example, we can call summary to get a general
## idea, and then inspect the cluster with the higher Enrichment Score, to see
## which members belong to it, etc. Or simply returning the whole cluster as
## a list with EnrichmentScore and Members.
summary(davidTermCluster2)
higherEnrichment <- which.max(enrichment(davidTermCluster2))
clusterGenes <- members(davidTermCluster2)[[higherEnrichment]]
wholeCluster <- cluster(davidTermCluster2)[[higherEnrichment]]
```

Description
This virtual class represents the output of a DAVID "Cluster" report, with "DAVIDTermCluster" and "DAVIDGeneCluster" as possible heirs, according to the report used.

Type
This class is a "Virtual" one.

Extends
• DAVIDResult in the conceptual way.

Heirs
• DAVIDTermCluster: DAVID's Functional Annotation Clustering report.
• DAVIDGeneCluster: DAVID's Functional Classification Tool report.

Slots
cluster named list with the different clustered terms/genes: Members, represented as DAVID-Genes object; and EnrichmentScore, a numeric with DAVID cluster enrichment score.

Methods
show signature(object="DAVIDCluster"): basic console output.
summary signature(object="DAVIDCluster"): basic summary console output.
initialize signature(object="DAVIDCluster", fileName="character"): basic cluster report file parser.
cluster signature(object="DAVIDCluster"): getter for the corresponding slot.
enrichment signature(object="DAVIDCluster"): obtain the enrichment score of each cluster.
members signature(object="DAVIDCluster"): obtain the corresponding cluster members.

Author(s)
Cristobal Fresno and Elmer A Fernandez

References
1. The Database for Annotation, Visualization and Integrated Discovery (david.abcc.ncifcrf.gov)
DAVIDFunctionalAnnotationChart-class

class "DAVIDFunctionalAnnotationChart"

Description

This class represents the output of "Functional Annotation Chart" of DAVID. It is an heir of DAVIDResult in the conceptual way, and also a data.frame with additional features, such as identifying the unique and duplicate ids, searching for genes with a given id, etc.

Type

This class is a "Concrete" one.

Extends

- DAVIDResult in the conceptual way.
- data.frame in order to extend the basic features.

Slots

no additional to the ones inherited from DAVIDResult and data.frame classes.

Methods

show signature(object="DAVIDFunctionalAnnotationChart"): returns a basic console output.
valid signature(object="DAVIDFunctionalAnnotationChart"): logical which checks DAVID's file output name ("Category", "Term", "Count", etc.) presence.
DAVIDFunctionalAnnotationChart signature( object="character"): constructor with the name of the .tab file report to load.
DAVIDFunctionalAnnotationChart signature( object="data.frame"): data.frame already loaded to use when constructing the object.
as signature(object="DAVIDFunctionalAnnotationChart"): coerce a data.frame into a DAVIDFunctionalAnnotationChart object.
categories signature( object="DAVIDFunctionalAnnotationChart"): obtain the factor vector of the "Category" column.
ids signature(object="DAVIDFunctionalAnnotationChart"): obtain a list with character/integer vector with the ids of the corresponding term.

Author(s)

Cristobal Fresno and Elmer A Fernandez
References

1. The Database for Annotation, Visualization and Integrated Discovery (david.abcc.ncifcrf.gov)

See Also

Other DAVIDFunctionalAnnotationChart: DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable

Examples

```
{  
##Load the Functional Annotation Chart file report for the input demo  
##file 2, using data function. Then, create a DAVIDFunctionalAnnotationChart  
##object using the loaded data.frame funChart2. In addition, the user can  
##use the file name of the downloaded file report.  
data(funChart2)  
davidFunChart2<-DAVIDFunctionalAnnotationChart(funChart2)  

##In Addition to the usual data.frame accessors, the user can inspect the  
##main categories used in the analysis.  
categories(davidFunChart2)  

##Obtain the ids of the genes present in each Term, as a list of character  
##vector  
ids(davidFunChart2)  

##Or plot a 2D tile matrix with the reported evidence (green) or not (black).  
##Just to keep it simple, for the first five terms present in funChart2  
##object.  
plot2D(DAVIDFunctionalAnnotationChart(funChart2[1:5, ]),  
color=c("FALSE"="black", "TRUE"="green"))  
}
```
DAVIDFunctionalAnnotationTable-class

Type

This class is a "Concrete" one.

Extends

- DAVIDResult in the conceptual way, and to reuse some functionalities such as plot2D, type and so on.

Slots

- Genes a DAVIDGenes object with the submitted genes.
- Dictionary a look up list of data.frame of each main annotation category, where the specified IDs and Terms used can be found.
- Membership list with logical membership matrix, where gene ids are coded by rows and the respective annotation category ids as columns.

Methods

- initialize signature(.Object= "DAVIDFunctionalAnnotationTable", fileName= "character"): basic Functional Annotation Table report file parser.
- DAVIDFunctionalAnnotationTable signature(fileName= "character"): high level Functional Annotation Table report file parser.
- valid signature(object= "DAVIDFunctionalAnnotationTable"): logical which checks for Membership, Dictionary and Genes cohesion.
- show signature(object= "DAVIDFunctionalAnnotationTable"): returns a basic console output.
- genes signature(object= "DAVIDFunctionalAnnotationTable"): returns a DAVIDGenes object.
- subset signature(object= "DAVIDFunctionalAnnotationTable", selection=c("Membership","Dictionary"), category, drop=TRUE): returns a subset list using the selection slot, looking up the category parameter if provided. Otherwise, it returns all the available main categories. Drop parameter indicates whether to drop list structure or not, if a list of length==1 is to be returned.
- dictionary signature(object= "DAVIDFunctionalAnnotationTable", category, drop=TRUE): returns subset using selection="Dictionary" and category and drop parameters.
- membership signature(object= "DAVIDFunctionalAnnotationTable", category= "character", drop=TRUE): returns subset using selection="Membership" and category and drop parameters.
- genes signature(object= "DAVIDFunctionalAnnotationTable", ...): returns a DAVIDGenes object slot, according to additional ... parameters.
- categories signature(object= "DAVIDFunctionalAnnotationTable"): returns a character vector with the main annotation categories available.
- plot2D signature(object= "DAVIDFunctionalAnnotationTable", category, id, names.genes=FALSE, names.category=FALSE): ggplot2 tile plot of genes id vs functional annotation category membership. If missing, all available data is used. In addition, names.genes and names.category parameters indicate whether to use or not, genes and category names respectively. Default value is FALSE.

Author(s)

Cristobal Fresno and Elmer A Fernandez
References

1. The Database for Annotation, Visualization and Integrated Discovery (david.abcc.ncifcrf.gov)

See Also

Other DAVIDFunctionalAnnotationTable: DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDGODag, DAVIDGODag, DAVIDGeneCluster, DAVIDGeneCluster, DAVIDGenes, DAVIDGenes, DAVIDGenes, DAVIDTermCluster, DAVIDTermCluster, as, as, as, categories, categories, dictionary, dictionary, genes, genes, genes, genes, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, membership, membership, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D, subset, subset

Examples

```r
##Load the Functional Annotation Table file report for the input demo
##file 1, using data function. Then, create a DAVIDFunctionalAnnotationTable
##object using the loaded data.frame annotationTable1. In addition, the user
##can use the file name of the downloaded file report.

#data(AnnotationTable1)
davidFunTable1<-DAVIDFunctionalAnnotationTable(annotationTable1)

##Now we can obtain the genes for the given ids, or the complete list if the
##parameter is omitted.

genes(davidFunTable1, id=c("37166_at","41703_r_at"))

##Or the main categories used on the analysis, in order to get the
##dictionary for a specific category (ID and Term fields), for the head of
##the data.frame.

categories(davidFunTable1)

doI dictionary(davidFunTable1, categories(davidFunTable1)[1]))

##And what about the membership of the genes in these terms? Just for the
##first six ids we can use:

head(membership(davidFunTable1, categories(davidFunTable1)[1]))

##Or simply plot the membership of only for the first six terms in this
##category, with only the genes of the first six terms with at least one
##evidence code.

##Category filtering...

categorySelection<-list(head(dictionary(davidFunTable1, categories(davidFunTable1)[1]))$ID))

names(categorySelection)<-categories(davidFunTable1)[1]

#Gene filter...

id<-membership(davidFunTable1, categories(davidFunTable1)[1]),[1:6]
id<-ids(genes(davidFunTable1))[rowSums(id)>0]

##Finally the membership tile plot
```
```
DAVIDGeneCluster-class

class "DAVIDGeneCluster"

Description
This class represents the output of a DAVID Gene Functional Classification Tool report.

Type
This class is a "Concrete" one.

Extends
• DAVIDCluster and uses its constructor to parse the report.

Slots
the ones inherited from DAVIDCluster.

Methods
initialize signature(.Object="DAVIDGeneCluster", fileName="character"):: basic cluster report file parser.
DAVIDGeneCluster signature(fileName="character"):: high level gene cluster report file parser.
ids signature(object="DAVIDGeneCluster"):: list with the member ids within each cluster.
genesis signature(object="DAVIDGeneCluster"):: list with the DAVIDGenes members within each cluster.
plot2D signature(object="DAVIDGeneCluster", color=c("FALSE"="black","TRUE"="green"), names=FALSE)
ggplot2 tile plot with gene membership to each cluster.

Author(s)
Cristobal Fresno and Elmer A Fernandez

References
1. The Database for Annotation, Visualization and Integrated Discovery (david.abcc.ncifcrf.gov)
See Also

Other DAVIDGeneCluster: DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationTable, DAVIDGODag, DAVIDTermCluster, DAVIDGeneCluster, DAVIDGenes, DAVIDGenes, DAVIDGenes, DAVIDGenes, DAVIDTermCluster, DAVIDTermCluster, as, as, genes, genes, genes, genes, ids, ids, ids, ids, ids, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D

Examples

{
##Load the Gene Functional Classification Tool file report for the
##input demo list 1 file to create a DAVIDGeneCluster object.
setwd(tempdir())
fileName<-system.file("files/geneClusterReport1.tab.tar.gz",
  package="RDAVIDWebService")
untar(fileName)
davidGeneCluster1<-'DAVIDGeneCluster(untar(fileName, list=TRUE))
davidGeneCluster1

##Now we can invoke DAVIDCluster ancestor functions to inspect the report
##data, of each cluster. For example, we can call summary to get a general
##idea, and the inspect the cluster with higher Enrichment Score, to see
##which members belong to it, etc. Or simply returning the whole cluster as
##a list with EnrichmentScore and Members.
summary(davidGeneCluster1)
higherEnrichment<-which.max(enrichment(davidGeneCluster1))
clusterGenes<-'members(davidGeneCluster1)[[higherEnrichment]]
wholeCluster<-'cluster(davidGeneCluster1)[[higherEnrichment]]

##Then, we can obtain the ids of the members calling clusterGenes object
##which is a DAVIDGenes class or directly using ids on davidGeneCluster1.
ids(clusterGenes)
ids(davidGeneCluster1)[[higherEnrichment]]

##Obtain the genes of the first cluster using davidGeneCluster1 object.
##Or, using genes on DAVIDGenes class once we get the members of the cluster.
genes(davidGeneCluster1)[[1]]
genes(members(davidGeneCluster1)[[1]])

##Finally, we can inspect a 2D tile membership plot, to visually inspect for
##overlapping of genes across the clusters. Or use a scaled version of gene
##names to see the association of gene cluster, e.g., cluster 3 is related to
##ATP genes.
plot2D(davidGeneCluster1)
plot2D(davidGeneCluster1, names=TRUE)+
  theme(axis.text.y=element_text(size=rel(0.9)))
}

---

**DAVIDGenes**

*High level constructors for DAVIDWebService package’s classes.*

**Description**

Different ways to build the different DAVIDWebService’s object according to the signature in use.
Usage

DAVIDGenes(object)

## S4 method for signature 'character'
DAVIDGenes(object)

## S4 method for signature 'data.frame'
DAVIDGenes(object)

## S4 method for signature 'DAVIDGenes'
initialize(.Object, fileName)

as(object, Class, strict=TRUE,
   ext=possibleExtends(thisClass, Class))

DAVIDFunctionalAnnotationChart(object)

## S4 method for signature 'character'
DAVIDFunctionalAnnotationChart(object)

## S4 method for signature 'data.frame'
DAVIDFunctionalAnnotationChart(object)

## S4 method for signature 'DAVIDFunctionalAnnotationChart'
initialize(.Object, fileName)

as(object, Class, strict=TRUE,
   ext=possibleExtends(thisClass, Class))

## S4 method for signature 'DAVIDCluster'
initialize(.Object, fileName)

DAVIDGeneCluster(object)

## S4 method for signature 'character'
DAVIDGeneCluster(object)

## S4 method for signature 'DAVIDTermCluster'
initialize(.Object, fileName)

DAVIDTermCluster(object)

## S4 method for signature 'character'
DAVIDTermCluster(object)

## S4 method for signature 'DAVIDFunctionalAnnotationTable'
initialize(.Object,
    fileName)

    as(object, Class, strict=TRUE,
    ext=possibleExtends(thisClass, Class))

    DAVIDFunctionalAnnotationTable(object)

    ## S4 method for signature 'character'
    DAVIDFunctionalAnnotationTable(object)

    ## S4 method for signature 'data.frame'
    DAVIDFunctionalAnnotationTable(object)

    ## S4 method for signature 'DAVIDGODag'
    initialize(.Object,funChart,type=c("BP","MF","CC"),pvalueCutoff=0.1,removeUnattached=FALSE,...)

    DAVIDGODag(funChart, ...)

    ## S4 method for signature 'DAVIDFunctionalAnnotationChart'
    DAVIDGODag(funChart, ...)

Arguments

object
    could be a character with the file name of the .tab report or data.frame already loaded.

fileName
    character with the file name of the .tab report to load.

.Object
    character to use in new function call. Possible values are: "DAVIDGenes", "DAVIDFunctionalAnnotationChart" or "DAVIDCluster".

Class
    character to use in the as function call. Possible values are: "DAVIDGenes" and "DAVIDFunctionalAnnotationChart".

strict,ext
    see as function.

funChart
    DAVIDFunctionalAnnotationChart object.

type
    character to indicate Gene Ontology main category: "BP" , "MF" or "CC".

pvalueCutoff
    numeric >0 <=1 to indicate the p-value to use as the threshold for enrichment. Default value is 0.1

removeUnattached
    Should unattached nodes be removed from GO DAG? Default value is FALSE.

...
    Additional parameters for lower level constructors (initialize).

Value

    a DAVIDWebService object according to function call:

    DAVIDGenes    object with genes description related data.
    DAVIDFunctionalAnnotationChart
        object with the respective report.
    DAVIDFunctionalAnnotationTable
        object with the respective report.
DAVIDCluster Not possible to invoke as it is a Virtual class.
DAVIDGeneCluster object with the respective report.
DAVIDTermCluster object with the respective report.
DAVIDGODag derived GOstats GO Direct Acyclic Graph from DAVIDFunctionalAnnotationChart data.

Author(s)
Cristobal Fresno and Elmer A Fernandez

See Also
Other DAVIDFunctionalAnnotationChart: DAVIDFunctionalAnnotationChart-class, categories, categories, ids, ids, ids, ids, plot2D, plot2D, plot2D, plot2D, plot2D
Other DAVIDFunctionalAnnotationTable: DAVIDFunctionalAnnotationTable-class, categories, categories, dictionary, dictionary, genes, genes, genes, genes, membership, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D, subset, subset
Other DAVIDGODag: DAVIDGODag-class, benjaminis, benjaminis, bonferronis, bonferronis, counts, counts, fdrs, fdrs, foldEnrichments, foldEnrichments, listTotals, listTotals, percentages, percentages, popHits, popHits, popTotals, popTotals, summary, summary, summary, summary, terms, terms, universeCounts, universeMappedCount, upsideDown, upsideDown
Other DAVIDGeneCluster: DAVIDGeneCluster-class, genes, genes, genes, genes, ids, ids, ids, ids, plot2D, plot2D, plot2D, plot2D, plot2D
Other DAVIDGenes: DAVIDGenes-class, genes, genes, genes, genes, ids, ids, ids, ids, ids, ids
Other DAVIDTermCluster: DAVIDTermCluster-class, ids, ids, ids, ids, ids, ids, ids, ids, plot2D, plot2D, plot2D, plot2D

Examples
{
  ##DAVIDGenes example:
  ##Load Show Gene List file report for the input demo file 1, using data
  ##function. Then, create a DAVIDGenes object using the loaded data.frame
  ##geneList1.
  data(geneList1)
  davidGenes1<-DAVIDGenes(geneList1)

  ##In addition, the user can use the file name of the downloaded file report.
  ##Here, we need to first uncompressed the report included in the package, in
  ##order to load it.
  setwd(tempdir())
  fileName<-system.file("files/geneListReport1.tab.tar.gz",
                        package="RDAVIDWebService")
  untar(fileName)
  davidGenes1<-DAVIDGenes(untar(fileName,list=TRUE))

  ##DAVIDFunctionalAnnotationChart example
  ##Load the Functional Annotation Chart file report for the input demo
  ##file 2, using data function. Then, create a DAVIDFunctionalAnnotationChart
## object using the loaded data.frame funChart2.
```r
data(funChart2)
davidFunChart2 <- DAVIDFunctionalAnnotationChart(funChart2)
```

## In addition, the user can use the file name of the downloaded file report.
## Here, we need to first uncompress the report included in the package, in
## order to load it.
```r
setwd(tempdir())
fileName <- system.file("files/functionalAnnotationChartReport2.tab.tar.gz", 
  package="RDAVIDWebService")
untar(fileName)
davidFunChart2 <- DAVIDFunctionalAnnotationChart(untar(fileName, list=TRUE))
```

## DAVIDFunctionalAnnotationTable example
## Load the Functional Annotation Table file report for the input demo
## file 1, using data function. Then, create a DAVIDFunctionalAnnotationTable
## object using the loaded data.frame annotationTable1.
```r
data(annotationTable1)
davidFunTable1 <- DAVIDFunctionalAnnotationTable(annotationTable1)
```

## In addition, the user can use the file name of the downloaded file report.
## Here, we need to first uncompress the report included in the package, in
## order to load it.
```r
setwd(tempdir())
fileName <- system.file("files/annotationTableReport1.tab.tar.gz", 
  package="RDAVIDWebService")
untar(fileName)
davidFunTable1 <- DAVIDFunctionalAnnotationTable(untar(fileName, list=TRUE))
```

## Example DAVIDGODag
## Load the Functional Annotation Chart file report for the input demo
## file 2, using data function. Then, create a DAVIDGODag object using
## Molecular Function main category of DAVIDFunctionalAnnotationChart object,
## obtained from the loaded data.frame funChart2. In addition, we have
## selected a threshold pvalue of 0.001 and removed unattached nodes, in case
## DAVID/GO.db database are not using the same version.
```r
data(funChart2)
davidGODag <- DAVIDGODag(DAVIDFunctionalAnnotationChart(funChart2), type="MF", 
pvalueCutoff=0.001, removeUnattached=TRUE)
```

## DAVIDGeneCluster example:
## Load the Gene Functional Classification Tool file report for the
## input demo list 1 file to create a DAVIDGeneCluster object.
```r
setwd(tempdir())
fileName <- system.file("files/geneClusterReport1.tab.tar.gz", 
  package="RDAVIDWebService")
untar(fileName)
davidGeneCluster1 <- DAVIDGeneCluster(untar(fileName, list=TRUE))
```

## DAVIDTermCluster example:
## Load the Gene Functional Classification Tool file report for the
## input demo file 2 to create a DAVIDGeneCluster object.
```r
setwd(tempdir())
```
(fileName <- system.file("files/termClusterReport2.tab.tar.gz", package="RDAVIDWebService")
untar(fileName)
davidTermCluster2 <- DAVIDTermCluster(untar(fileName, list=TRUE))
)

DAVIDGenes-class class "DAVIDGenes"

Description

This class represents the output of "Show Genes Result" of DAVID. It is an heir of DAVIDResult in the conceptual way, and also a data.frame with additional features, such as identifying the unique and duplicate ids, searching for genes with a given id, etc.

Type

This class is a "Concrete" one.

Extends

• DAVIDResult in the conceptual way.
• data.frame in order to extend the basic features.

Slots

none additional to the ones inherited from DAVIDResult and data.frame classes.

Methods

valid signature(object="DAVIDGenes"): logical which checks for data.frame name (ID, Name) presence.
DAVIDGenes signature(object="character"): constructor with the name of the .tab file report to load.
DAVIDGenes signature(object="data.frame"): data.frame already loaded to use when constructing the object.
ids signature(object="DAVIDGenes"): character vector with gene submitted ids.

Author(s)

Cristobal Fresno and Elmer A Fernandez

References

1. The Database for Annotation, Visualization and Integrated Discovery (david.abcc.ncifcrf.gov)
See Also

Other DAVIDGenes: DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDGODag, DAVIDGODag, DAVIDGeneCluster, DAVIDGeneCluster, DAVIDGenes, DAVIDGenes, DAVIDGenes, DAVIDTermCluster, DAVIDTermCluster, as, as, as, genes, genes, genes, ids, ids, ids, ids, ids, initialize, initialize, initialize, initialize, initialize, initialize, initialize.

Examples

{
  ##Load Show Gene List file report for the input demo file 1, using data
  ##function. Then, create a DAVIDGenes object using the loaded data.frame
  ##geneList1. In addition, the user can use the file name of the downloaded
  ##file report.
  data(geneList1)
  davidGenes1<-DAVIDGenes(geneList1)

  ##Now we can inspect davidGenes1 as it was an common data.frame
  head(davidGenes1)

  ##Additional getters for this object are also available, to obtain the
  ##different columns: ids, genes and species.
  ids(davidGenes1)
  genes(davidGenes1)
  species(davidGenes1)

  ##Or even look up for a particular gene id, which will return only the
  ##matched ones.
  genes(davidGenes1, ids=c("38926_at", "35367_at", "no match"))

  ##Obtain the genes with duplicate manufacturer ids or just the genes that
  ##do not have duplicate ids (uniqueIds).
  duplicateIds(davidGenes1)
  uniqueIds(davidGenes1)
}

DAVIDGODag-class  
class "DAVIDGODag"

Description

This concrete class represents an induced GO DAG generated by the DAVID Functional Annotation Chart report a.k.a a DAVIDFunctionalAnnotationChart object.

Type

This class is a "Concrete" one.

Extends

- `GOHyperGResult` directly, in order to reuse GOstats functionalities.
DAVIDGODag-class

Slots

the ones inherited from GOHyperGResult

Methods

show signature(object="DAVIDGODag"): basic console output.
summary signature(object="DAVIDGODag", ...): basic summary console output.
initialize signature(object="DAVIDGODag", fileName="character"): basic cluster report file parser.
DAVIDGODag signature(object="DAVIDGODag", fileName="character"): high level constructor to parse the file report.
universeMappedCount, universeCounts, counts signature(object="DAVIDGODag"): modifications to the corresponding GOstats/Category library functions, to keep the same behavior, for DAVIDGODag object.
fdrs, benjaminis, bonferronis signature(object="DAVIDGODag"): Adjusted method specific p-values for the corresponding nodes/terms.
terms signature(object="DAVIDGODag"): character vector with GO node names.

See Also

Other DAVIDGODag: DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDGODag, DAVIDGODag, DAVIDGeneCluster, DAVIDGeneCluster, DAVIDGenes, DAVIDGenes, DAVIDGenes, DAVIDTermCluster, DAVIDTermCluster, as, as, as, benjaminis, benjaminis, benjaminis, bonferronis, bonferronis, counts, counts, fdrs, fdrs, foldEnrichments, foldEnrichments,
Examples

```r
##Load the Functional Annotation Chart file report for the input demo file 2, using data function. Then, create a DAVIDGODag object using Molecular Function main category of DAVIDFunctionalAnnotationChart object, obtained from the loaded data.frame funChart2. In addition, we have selected a threshold pvalue of 0.001 and removed unattached nodes, in case DAVID/GO.db database are not using the same version.
data(funChart2)
davidGODag <- DAVIDGODag(DAVIDFunctionalAnnotationChart(funChart2), type="MF", pvalueCutoff=0.001, removeUnattached=TRUE)

##Now, we can inspect the enrichment GO DAG using GOstats functionalities: counts, pvalues, sigCategories, universeCounts, geneMappedCount, etc. However, oddsRatios, expectedCounts and universeMappedCount are not available because these results are not available on DAVID's Functional Annotation Chart report. In addition geneIdUniverse are not the ones of the universe but the ids on the category (geneIdsByCategory).
davidGODag
counts(davidGODag)
pvalues(davidGODag)
sigCategories(davidGODag, p=0.0001)
universeCounts(davidGODag)
geneMappedCount(davidGODag)
geneIdsByCategory(davidGODag)
summary(davidGODag)

##In addition, the new nodeData attributes (term, listTotal, popHit, popTotal, foldEnrichment, bonferroni, benjamini, fdr) can be retrieved.
terms(davidGODag)
listTotals(davidGODag)
popHits(davidGODag)
popTotals(davidGODag)
foldEnrichments(davidGODag)
bonferronis(davidGODag)
benjaminis(davidGODag)
fdrs(davidGODag)

##The user can even plot the enrichment GO DAG if Rgraphviz package is available.
plotGOTermGraph(g=goDag(davidGODag), r=davidGODag, max.nchar=30, node.shape="ellipse")
```

**DAVIDResult-class**

`DAVIDResult-class`  
`initialize`, `initialize`, `initialize`, `initialize`, `initialize`, `initialize`, `initialize`,  
`initialize`, `initialize`, `initialize`, `initialize`, `initialize`,  
`listTotals`, `listTotals`, `percentages`, `percentages`, `popHits`, `popHits`, `popTotals`, `popTotals`,  
`summary`, `summary`, `summary`, `summary`, `summary`, `summary`, `terms`, `terms`, `universeCounts`, `universeMappedCount`,  
`upsideDown`, `upsideDown`
DAVIDResult-class

Description
This class represents the most generic result obtained in the Database for Annotation, Visualization and Integrated Discovery (DAVID) website (see References).

Type
This class is a "Virtual" one.

Heirs
- DAVIDGenes: basic gene information (ID, Name and Specie)
- DAVIDCluster: generic Cluster result (Term or Gene).
- DAVIDFunctionalAnnotationChart: EASE results on each Functional Category (see references).
- DAVIDFunctionalAnnotationTable: annotation for each gene, no statistical analysis.

Slots
- type Object of class "character". Contains the name of DAVID's result.

Methods
- show signature(object="DAVIDResult"): returns a basic console output.
- type signature(object="DAVIDResult"): getter for type slot.
- plot2D signature(object="DAVIDResult", dataFrame="data.frame"): internal ggplot tile plot for gene/term cluster and annotation heirs.

Author(s)
Cristobal Fresno and Elmer A Fernandez

References
1. The Database for Annotation, Visualization and Integrated Discovery (david.abcc.ncifcrf.gov)
3. Huang, D. W.; Sherman, B. T. & Lempicki, R. A. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. Nucleic Acids Res, Laboratory of Immunopathogenesis and Bioinformatics, Clinical Services Program, SAIC-Frederick, Inc., National Cancer Institute at Frederick, Frederick, MD 21702, USA., 2009, 37, 1-13

See Also
Other DAVIDResult: plot2D, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D, type, type
Description
This class represents the output of a DAVID Functional Annotation Clustering report.

Type
This class is a "Concrete" one.

Extends
• DAVIDCluster and uses its constructor to parse the report.

Slots
the ones inherited from DAVIDCluster.

Methods
initialize signature(.Object="DAVIDTermCluster", fileName="character"): basic cluster report file parser.
DAVIDTermCluster signature(fileName="character"): high level gene cluster report file parser.
ids signature(object="DAVIDTermCluster"): list with the member ids within each cluster.
plot2D signature(object="DAVIDTermCluster", number=1, color=c("FALSE"="black","TRUE"="green"): ggplot2 tile plot of genes vs functional annotation category membership of the given cluster number.

Author(s)
Cristobal Fresno and Elmer A Fernandez

References
1. The Database for Annotation, Visualization and Integrated Discovery (david.abcc.ncifcrf.gov)

See Also
Other DAVIDTermCluster: DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDGeneCluster, DAVIDGeneCluster, DAVIDGenes, DAVIDGenes, DAVIDGenes, DAVIDTermCluster, DAVIDTermCluster, as, as, as, ids, ids, ids, initialize, initialize, initialize, initialize, initialize, initialize, initialize, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D
DAVIDWebService-class

Main class to connect to DAVID Web Service

Description
A reference class to manage DAVID’s Web Service connectivity, to run Set Enrichment Analysis (SEA) or Modular Enrichment Analysis (MEA) on a candidate list of gene/protein(s) with respect to a background list (the genome of the organism by default).

Usage
DAVIDWebService(...)

Arguments

... additional parameters. See Methods section.

Details
DAVIDWebService class is implemented as a reference class, to manage a single instance connection to DAVIS’s server by means of web services using a registered e-mail. For user registration, go to...
http://david.abcc.ncifcrf.gov/webservice/register.html. The implementation uses Java Remote Method Implementation (RMI) to connect the client and server side of DAVID. The main functionalities include:

1. Connectivity: upload gene/background list/s, change gene/background position, select current specie/s, select annotations, etc. from R.

2. Reports: Submitted Gene List, Annotation Category Summary, Gene/Term Clusters, Functional Annotation Chart and Functional Annotation Table as native R objects.

**Fields**

- **stub**: Java jobjRef which corresponds to a sample/session/client/stub/DAVIDWebServiceStub object for the client side of DAVID.
- **email**: character.

**Methods**

- **show()**: prints DAVIDWebService object.
- **summary()**: return a data.frame with a summary of all available annotations in DAVID in terms of percentage of gene list ids present in the category and numbers of terms where they can be found (see getAnnotationSummary)
- **initialize(email="", ..., url)**: constructor for DAVIDWebService object, which includes: Java Virtual Machine initialization (... if required), and stub initialization with the provided email (if present) and using the url parameter for the API website.
- **setEmail(mail)**: Set the email field with the given registered user email parameter for authentication purposes.
- **getEmail()**: Returns the current authentication email in use.
- **getStub()**: Returns jobRef object with the current stub field in use.
- **is.connected()**: Check if connected to the DAVID server.
- **connect()**: Try to establish a connection with the DAVID server using the provided email.
- **getIdTypes()**: Returns all acceptable DAVID idTypes.
- **getAllAnnotationCategoryNames()**: Returns all available annotation category names.
- **getDefaultCategoryNames()**: Returns all default category names.
- **getGeneListNames()**: Returns submitted gene list names.
- **getBackgroundColorNames()**: Returns submitted background names.
- **getListName(listType=c("Gene", "Background"), position=1L)**: Get the name of the selected list type at a given position.
- **getSpecieNames()**: Return specie/s of the current gene list.
- **getCurrentGeneListPosition()**: Return the position of current gene list.
- **getCurrentBackgroundListPosition()**: Return the position of current background list.
- **getCurrentSpeciesPosition()**: Return current specie/s used positions for the uploaded gene list.
- **setCurrentGeneListPosition(position)**: Use the gene list of the given position.
- **setCurrentBackgroundPosition(position)**: Use the gene list of the given position.
- **setCurrentSpecies(species)**: Select the specie/s of the submitted gene list to use in the analysis.
setAnnotationCategories(categories): Select the specie/s of the submitted gene list to use in the analysis.

addList(inputIds, idType, listName, listType=c("Gene", "Background")): Add a gene or background to the current session.

geneCategoriesReport(): Get the gene report categories.

geneAnnotationSummary(): Generate the summary of all available annotation in DAVID in terms of percentage of gene list ids present in the category and numbers of terms where the can be found.


getFunctionalAnnotationChartFile(fileName, threshold=0.1, count=2L): Generate the Functional Annotation Chart Report for the selected functional categories, for the given EASE threshold and number of genes and save it to a file.

getFunctionalAnnotationChart(...): getFunctionalAnnotationChart but as an R object.


getFunctionalAnnotationTableFile(fileName): Generate Functional Annotation Table Report File, which is a gene-centric view of the genes and their associated annotation terms (selected only). There is no statistics applied in this report.

getFunctionalAnnotationTable(): getFunctionalAnnotationTable but as an R object.

Limitations

1. A job with more than 3000 genes to generate gene or term cluster report will not be handled by DAVID due to resource limit.
2. No more than 200 jobs in a day from one user or computer.
3. DAVID Team reserves right to suspend any improper uses of the web service without notice.

Author(s)

Cristobal Fresno <cristobalfresno@gmail.com> and Elmer A. Fernandez <elmerfer@gmail.com>

References

1. The Database for Annotation, Visualization and Integrated Discovery (david.abcc.ncifcrf.gov)
3. Huang, D. W.; Sherman, B. T. & Lempicki, R. A. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. Nucleic Acids Res, Laboratory of Immunopathogenesis and Bioinformatics, Clinical Services Program, SAIC-Frederick, Inc., National Cancer Institute at Frederick, Frederick, MD 21702, USA., 2009, 37, 1-13
### demoList1

**DAVID's website demoList1 example id files**

**Description**

This dataset contains the same example input id files present in the Database for Annotation, Visualization and Integrated Discovery.

**Usage**

```r
data(demoList1)

data(demoList2)
```

**Format**

character vector with `AFFYMETRIX_3PRIME_IVT_ID` manufacturer identification codes (ids)

- **demoList1**: 164 ids in total.
- **demoList2**: 403 ids in total.

**Author(s)**

Cristobal Fresno and Elmer A Fernandez
References

1. The Database for Annotation, Visualization and Integrated Discovery (david.abcc.ncifcrf.gov)


See Also

Other DataExamples: annotationSummary1, annotationSummary2, geneList1, geneList2

funChart1/2 are data.frame for demoList1/2 input ids, respectively, with the following columns.

- **Category** factor with the main categories under used in the present analysis.
- **Term** character with the name of the term in format id~name (if available).
- **Count** integer with the number of ids of the gene list that belong to this term.
- **X** after converting user input gene IDs to corresponding DAVID gene ID, it refers to the percentage of DAVID genes in the list associated with particular annotation term. Since DAVID gene ID is unique per gene, it is more accurate to use DAVID ID percentage to present the gene-annotation association by removing any redundancy in user gene list, i.e. two user IDs represent same gene.
- **PValue** numeric with the EASE Score of the term (see DAVID Help page).
- **Genes** character in comma separated style with the genes present in the term.
- **List.Total, Pop.Hits, Pop.Total** integers (in addition to Count) to build the 2x2 contingency table in order to compute the EASE Score (see DAVID Help page).
- **Fold.Enrichment** numeric with the ratio of the two proportions. For example, if 40/400 (i.e. 10%) of your input genes involved in "kinase activity" and the background information is 300/30000 genes (i.e. 1%) associating with "kinase activity", roughly 10%/1%=10 fold enrichment.
- **Bonferroni, Benjamini, FDR** numerics with p-value adjust different criterias (see p.adjust)
geneCluster1

Author(s)

Cristobal Fresno and Elmer A Fernandez

References

1. The Database for Annotation, Visualization and Integrated Discovery (david.abcc.ncifcrf.gov)


3. DAVID Help page http://david.abcc.ncifcrf.gov/helps/functional_annotation.html#

Description

These datasets correspond to the Functional Annotation Clustering or Gene Functional Classification report obtained in the Database for Annotation, Visualization and Integrated Discovery (DAVID) website, using as input file the ones provided for demo purposes (demoList1 or demoList2) with GOTERM_BP_ALL, GOTERM_MF_ALL and GOTERM_CC_ALL categories.

Format

geneCluster1/2 or termCluster1/2 are tab delimitate unstructured files with DAVID format where:

Cluster header  1. TypeGene Cluster or Annotation Cluster.
                2. Numberinteger to indicate the cluster label.
                3. Enrichment Score numeric with the geometric mean (in -log scale) of members p-values in a corresponding annotation cluster, is used to rank their biological significance. Thus, the top ranked annotation groups most likely have consistent lower p-values for their annotation members.

Members Header according to the type of cluster it can be:

1. Gene the character vector with "ID", "Gene" and "Name".
2. Annotation the same columns of a Functional Annotation Chart (see getFunctionalAnnotationChart).

Members Body member data per line according to the respective type of cluster.

Author(s)

Cristobal Fresno and Elmer A Fernandez
References

1. The Database for Annotation, Visualization and Integrated Discovery (davidgeneList.abcc.ncifcrf.gov)
3. DAVID Help page http://david.abcc.ncifcrf.gov/helps/functional_classification.html#textmode

---

geneList1

DAVID's website gene list example files

Description

These datasets correspond to the reports obtained using Show Gene List in the Database for Annotation, Visualization and Integrated Discovery (DAVID) website, using as input file the ones provided for demo purposes (demoList1 or demoList2) with default options.

Usage

data(geneList1)
data(geneList2)

Format

geneList1/2 are data.frame for demoList1/2 input ids, respectively, with the following columns.

ID character with the Gene List ID present in DAVID knowledge base, in the submitted type. If more than one ids map to the same DAVID ID, the record is a comma separated character.

Name character with the name of the gene as seen in DAVID knowledge base, in a comma separated fashion (if more than one ID maps to the same DAVID ID).

Species factor with the name of the Specie.

Author(s)

Cristobal Fresno and Elmer A Fernandez

References

1. The Database for Annotation, Visualization and Integrated Discovery (david.abcc.ncifcrf.gov)
3. DAVID Help page http://david.abcc.ncifcrf.gov/helps/functional_annotation.html#E3
genes

See Also

Other DataExamples: annotationSummary1, annotationSummary2, demoList1, demoList2

genes for the different DAVIDWebService package class objects.

Description

Obtain genes related information, according to the given function call (see Values).

Usage

genes(object, ...)

## S4 method for signature 'DAVIDGenes'
genes(object, ids)

## S4 method for signature 'DAVIDGeneCluster'
genes(object)

## S4 method for signature 'DAVIDFunctionalAnnotationTable'
genes(object, ...)

Arguments

object    DAVIDGenes or DAVIDGeneCluster class object.
ids       character vector with the ids to fetch.
...       Additional parameters for internal functions (if applicable).

Value

according to the call one of the following objects can be returned

DAVIDGenes  a DAVIDGenes object with the matched genes of ids parameter. If missing, returns all the genes.
DAVIDGeneCluster list with DAVIDGenes objects for each cluster.
DAVIDFunctionalAnnotationTable a DAVIDGenes objects, according to ... parameter used internally on genes(DAVIDGenes, ...).

Author(s)

Cristobal Fresno and Elmer A Fernandez
See Also

Other DAVIDFunctionalAnnotationTable: DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable-class, DAVIDGODag, DAVIDGeneCluster, DAVIDGeneCluster, DAVIDGenes, DAVIDGenes, DAVIDGenes, DAVIDTermCluster, DAVIDTermCluster, as, as, as, categories, categories, categories, categories, categories, categories, dictionary, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, membership, membership, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D, subset, subset

Other DAVIDGeneCluster: DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable-class, DAVIDGenes, DAVIDGenes, DAVIDGenes, DAVIDGeneCluster, DAVIDGeneCluster, as, as, ids, ids, ids, ids, ids, initialize, initialize, initialize, initialize, initialize, initialize, initialize, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D

Other DAVIDGenes: DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDGODag, DAVIDGODag, DAVIDGeneCluster, DAVIDGeneCluster, DAVIDGenes, DAVIDGenes, DAVIDGenes-class, DAVIDTermCluster, DAVIDTermCluster, as, as, ids, ids, ids, ids, ids, initialize, initialize, initialize, initialize, initialize, initialize, initialize

Examples

```r
## DAVIDGenes example:
## Load Show Gene List file for the input demo file 1, using data
data(geneList1)
davidGenes1 <- DAVIDGenes(geneList1)

## Now, get the genes using the ids look up parameter with the first six ids. If ids omitted, all the available are returned.
genesis(davidGenes1, ids = head(ids(davidGenes1)))

## DAVIDFunctionalAnnotationTable example:
## Load the Functional Annotation Table file report for the input demo
## file 1, using data function. Then, create a DAVIDFunctionalAnnotationTable
## object using the loaded data.frame annotationTable1.
data( annotationTable1 )
davidFunTable1 <- DAVIDFunctionalAnnotationTable( annotationTable1 )

## Now we can obtain the genes for the given ids, or the complete list if the parameter is omitted.
genesis(davidFunTable1, id = c("37166_at", "41703_r_at"))

## DAVIDGeneCluster example:
## Load the Gene Functional Classification Tool file report for the input demo list 1 file to create a DAVIDGeneCluster object.
## setwd(tempdir())
## fileName <- system.file("files/geneClusterReport1.tab.tar.gz", package = "RDAVIDWebService")
```
untar(fileName)
davidGeneCluster1<-DAVIDGeneCluster(untar(fileName, list=TRUE))

##Then, we can obtain the genes of the first cluster using davidGeneCluster1
##object. Or, using genes on DAVIDGenes class once we get the members of the
##cluster
genes(davidGeneCluster1)[[1]]
genes(members(davidGeneCluster1)[[1]])
}

---

**Description**

DAVIDWebService class methods to obtain DAVID website reports from R. This includes the different functionalities starting from the basic "Show Gene List" or "Annotation Summary", to Set Enrichment Analysis using "Functional Annotation Chart" or Modular Enrichment Analysis using "Functional Annotation Clustering" or "Gene Functional Classification Tool". Note that DAVIDWebService is a Reference class, hence invoke it using object_name$method_name(parameters). In addition, the user can use the S4 version style function call (see Details).

**Usage**

```r
getGeneCategoriesReport(object)

## S4 method for signature 'DAVIDWebService'
getGeneCategoriesReport(object)
```

```r
getAnnotationSummary(object)

## S4 method for signature 'DAVIDWebService'
getAnnotationSummary(object)
```

```r
getGeneListReportFile(object, fileName)

## S4 method for signature 'DAVIDWebService'
getGeneListReportFile(object, fileName)
```

```r
getGeneListReport(object)

## S4 method for signature 'DAVIDWebService'
getGeneListReport(object)
```

```r
getFunctionalAnnotationChartFile(object, fileName, threshold=0.1, count=2L)

## S4 method for signature 'DAVIDWebService'
getFunctionalAnnotationChartFile(object, fileName, threshold=0.1, count=2L)
```
getFunctionalAnnotationChart(object, ...) 

## S4 method for signature 'DAVIDWebService'
getFunctionalAnnotationChart(object, ...)

getClusterReportFile(object, fileName, type=c("Term", "Gene"), overlap=4L, initialSeed=4L, finalSeed=4L, linkage=0.5, kappa=35L)

## S4 method for signature 'DAVIDWebService'
getClusterReportFile(object, fileName, type=c("Term", "Gene"), overlap=4L, initialSeed=4L, finalSeed=4L, linkage=0.5, kappa=35L)

getClusterReport(object, type=c("Term", "Gene"), ...)

## S4 method for signature 'DAVIDWebService'
getClusterReport(object, type=c("Term", "Gene"), ...)

getFunctionalAnnotationTableFile(object, fileName)

## S4 method for signature 'DAVIDWebService'
getFunctionalAnnotationTableFile(object, fileName)

getFunctionalAnnotationTable(object)

## S4 method for signature 'DAVIDWebService'
getFunctionalAnnotationTable(object)

**Arguments**

- `object`: DAVIDWebService class object.
- `fileName`: character with the name of the file to store the Report.
- `threshold`: numeric with the EASE score (at most equal) that must be present in the category to be included in the report. Default value is 0.1.
- `count`: integer with the number of genes (greater equal) that must be present in the category to be included in the report. Default value is 2.
- `type`: character with the type of cluster to obtain Term/Genes. Default value "Term".
- `overlap`: integer with the minimum number of annotation terms overlapped between two genes in order to be qualified for kappa calculation. This parameter is to maintain necessary statistical power to make kappa value more meaningful. The higher value, the more meaningful the result is. Default value is 4L.
- `initialSeed`, `finalSeed`: integer with the number of genes in the initial (seeding) and final (filtering) cluster criteria. Default value is 4L for both.
- `linkage`: numeric with the percentage of genes that two clusters share in order to become one.
getGeneCategoriesReport

kappa integer (kappa * 100), with the minimum kappa value to be considered biological significant. The higher setting, the more genes will be put into unclustered group, which lead to higher quality of functional classification result with a fewer groups and a fewer gene members. Kappa value 0.3 starts giving meaningful biology based on our genome-wide distribution study. Anything below 0.3 have great chance to be noise.

... additional parameters for getXXFile functions.

Details

Available functions include:

**getGeneCategoriesReport**: Get the gene categories report.

**getAnnotationSummary**: Generate the summary of all available annotation in DAVID in terms of percentage of gene list ids present in the category and numbers of terms where the can be found.

**getGeneListReportFile**: Generate the Gene List Report a.k.a Show Gene List in DAVID website and save it into a file.

**getGeneListReport**: Generate Gene List Report a.k.a Show Gene List in DAVID website and import it as a DAVIDGenes object into R.

**getFunctionalAnnotationChartFile**: Generate the Functional Annotation Chart Report for the selected functional categories, for the given EASE threshold and number of genes and save it to a file.

**getFunctionalAnnotationChart**: Generate the Functional Annotation Chart Report for the selected functional categories, for the given EASE threshold and number of genes, and import it as a DAVIDFunctionalAnnotationChart object in R.

**getClusterReportFile**: Generate the Term/Gene Cluster Report for the given configuration.

**getClusterReport**: Generate the Term/Gene Cluster Report for the given configuration, and import it as a DAVIDGeneCluster or DAVIDTermCluster object, according to function call.

**getFunctionalAnnotationTableFile**: Generate Functional Annotation Table Report File, which is a gene-centric view of the genes and their associated annotation terms (selected only). There is no statistics applied in this report.

**getFunctionalAnnotationTable**: Generate Functional Annotation Table Report and import it as a DAVIDFunctionalAnnotationTable object in R.

Value

according to the call one of the following objects can be returned

**getGeneCategoriesReport**

integer vector with the IDs of the categories.

**getAnnotationSummary**

data.frame with the annotation summary report with the following columns:

1. **Main.Category**: factor with the main categories under used in the present analysis.
2. **ID**: integer to identify the annotation category.
3. **Name**: character with the name of category (the available ones in getAnnotationCategoryNames function).
4. **X.**: numeric with the percentage of the gene list ids present in the term.
5. **Count**: integer with the number of ids of the gene list that belong to this term.

`getGeneListReportFile` generates a data.frame with the Gene List Report with the following columns:

1. **ID**: character with the Gene List ID present in DAVID knowledge base, in the submitted type. If more than one ids map to the same DAVID ID, the record is a comma separated character.
2. **Name**: character with the name of the gene as seen in DAVID knowledge base, in a comma separated fashion (if more than one ID maps to the same DAVID ID).
3. **Species**: factor with the name of the Specie.

`getGeneListReport` generates Gene List Report a.k.a Show Gene List in DAVID website and import it as a DAVIDGenes object in R.

`getFunctionalAnnotationChartFile` generates a file with the following columns:

1. **Category**: factor with the main categories under used in the present analysis.
2. **Term**: character with the name of the term in format id~name (if available).
3. **Count**: integer with the number of ids of the gene list that belong to this term.
4. **X.**: after converting user input gene IDs to corresponding DAVID gene ID, it refers to the percentage of DAVID genes in the list associated with a particular annotation term. Since DAVID gene ID is unique per gene, it is more accurate to use DAVID ID percentage to present the gene-annotation association by removing any redundancy in user gene list, i.e. two user IDs represent same gene.
5. **PValue**: numeric with the EASE Score of the term (see DAVID Help page).
6. **Genes**: character in comma separated style with the genes present in the term.
7. **List.Total, Pop.Hits, Pop.Total**: integers (in addition to Count) to build the 2x2 contingency table in order to compute the EASE Score (see DAVID Help page).
8. **Fold.Enrichment**: numeric with the ratio of the two proportions. For example, if 40/400 (i.e. 10%) of your input genes involved in "kinase activity" and the background information is 300/30000 genes (i.e. 1%) associating with "kinase activity", roughly 10% / 1% = 10 fold enrichment.
9. **Bonferroni, Benjamini, FDR**: numerics with p-value adjust different criteria (see p.adjust).

`getFunctionalAnnotationChart` generates the Functional Annotation Chart Report for the selected functional categories, for the given EASE threshold and number of genes, and import it as a DAVIDFunctionalAnnotationChart object in R.

`getClusterReportFile` generates a file with the following columns:

1. **Annotation/Gene Cluster**: integer with the number of cluster.
2. **EnrichmentScore**: numeric with the geometric mean (in -log scale) of members p-values in a corresponding annotation cluster, is used to rank their biological significance. Thus, the top ranked annotation groups most likely have consistent lower p-values for their annotation members.
3. **Members**: according to the type of cluster, changes the associated data to include Gene List or Functional Chart Report (see `getGeneListReport` and `getFunctionalAnnotationChart`).

**getClusterReport**

Generate the Term/Gene Cluster Report for the given configuration, and import it as a DAVIDGeneCluster or DAVIDTermCluster according to function call.

**getFunctionalAnnotationTableFile**

file with the following columns:

1. **Gene**: Three Columns with the same data included in Gene List Report (ID, Gene.Name and Species) but coding for DAVID ID, i.e., comma separated character with input ids if two or more stands for the same gene.
2. **Annotation**: as many columns as Annotation Categories were in used. In each column, a comma separated style is use to delimitate the different terms where is reported evidence for DAVID ID record.

**getFunctionalAnnotationTable:**

Generate Functional Annotation Table Report, which is a gene-centric view of the genes and their associated annotation terms (selected only), and import it as a DAVIDFunctionalAnnotationTable object in R.

**References**


**See Also**

`p.adjust` and `fisher.test`

Other DAVIDWebService: DAVIDWebService-class, addList, addList.connect, connect, getAnnotationCategoryNames, getAnnotationCategories, getBackgroundListNames, getBackgroundListNames, getCurrentBackgroundPosition, getCurrentGeneListPosition, getCurrentGeneListPosition, getCurrentSpeciesPosition, getCurrentSpeciesPosition, getDefaultCategoryNames, getDefaultCategoryNames, getEMail, getEMail, getGeneListNames, getGeneListNames, getIdTypes, getIdTypes, getListName, getListName, getSpeciesNames, getSpeciesNames, getStub, getStub, is.connected, is.connected, setAnnotationCategories, setAnnotationCategories, setAnnotationCategories, setAnnotationCategories, setAnnotationCategories, setAnnotationCategories, setCurrentBackgroundPosition, setCurrentBackgroundPosition, setCurrentGeneListPosition, setCurrentGeneListPosition, setCurrentGeneListPosition, setCurrentGeneListPosition, setCurrentGeneListPosition, setCurrentGeneListPosition, setSpecies, setSpecies, setEmail, setEmail, setEmail, DAVIDWebService-method, summary, summary, summary, summary

| ids | ids for the different DAVIDWebService package class objects |

**Description**

Obtain ids related information, according to the given function call (see Values).
Usage

ids(object)

## S4 method for signature 'DAVIDGenes'
ids(object)

## S4 method for signature 'DAVIDFunctionalAnnotationChart'
ids(object)

## S4 method for signature 'DAVIDGeneCluster'
ids(object)

## S4 method for signature 'DAVIDTermCluster'
ids(object)

Arguments

object DAVIDWebService class object. Possible values are: DAVIDGenes, DAVID-
FunctionalAnnotationChart, DAVIDGeneCluster or DAVIDTermCluster.

Value

taking to the call one of the following objects can be returned

DAVIDGenes character vector with gene submitted ids.
DAVIDFunctionalAnnotationChart
list with character/integer vector of ids of the corresponding "Category".
DAVIDGeneCluster, DAVIDTermCluster
list with character/integer vector of ids of the members of each cluster.

Author(s)

Cristobal Fresno and Elmer A Fernandez

See Also

Other DAVIDFunctionalAnnotationChart: DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart,
DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart-class, DAVIDFunctionalAnnotationTable,
DAVIDFunctionalAnnotationTable, DAVIDGODag, DAVIDGODag,
DAVIDGeneCluster, DAVIDGeneCluster, DAVIDGenes, DAVIDGenes, DAVIDGenes, DAVIDTermCluster,
DAVIDTermCluster, as, as, as, categories, categories, categories, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, plot2D, plot2D, plot2D, plot2D, plot2D

Other DAVIDGeneCluster: DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart,
DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDGODag, DAVIDGODag, DAVIDGeneCluster, DAVIDGeneCluster-class, DAVIDGenes, DAVIDGenes, DAVIDGenes, DAVIDTermCluster, DAVIDTermCluster, as, as, as, genes, genes, genes, genes, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, plot2D, plot2D, plot2D, plot2D, plot2D

Other DAVIDGenes: DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDGODag, DAVIDGODag, DAVIDGeneCluster, DAVIDGeneCluster,
Examples

{
  ##DAVIDGenes example:
  ##Load Show Gene List file report for the input demo file 1, using data
  ##function. Then, create a DAVIDGenes object using the loaded data.frame
  ##geneList1. Once, the report is loaded, we can retrieve the ids.
  data(geneList1)
  davidGenes1<-DAVIDGenes(geneList1)
  ids(davidGenes1)

  ##DAVIDFunctionalAnnotationChart example:
  ##Load the Functional Annotation Chart file report for the input demo
  ##file 2, using data function. Then, create a DAVIDFunctionalAnnotationChart
  ##object using the loaded data.frame funChart2. Once the report is loaded,
  ##the user can obtain the ids of the genes present in each Term, as a list of
  ##character vector.
  data(funChart2)
  davidFunChart2<-DAVIDFunctionalAnnotationChart(funChart2)
  ids(davidFunChart2)

  ##DAVIDGeneCluster example:
  ##Load the Gene Functional Classification Tool file report for the
  ##input demo list 1 file to create a DAVIDGeneCluster object.
  setwd(tempdir())
  fileName<-system.file("files/geneClusterReport1.tab.tar.gz",
       package="RDAVIDWebService")
  untar(fileName)
  davidGeneCluster1<-DAVIDGeneCluster(untar(fileName, list=TRUE))
  davidGeneCluster1

  ##Now we can invoke DAVIDCluster ancestor functions to inspect the report
  ##data, of each cluster. For example, we can call summary to get a general
  ##idea, and the inspect the cluster with higher Enrichment Score, to see
  ##which members belong to it, etc. Or simply returning the whole cluster as
  ##a list with EnrichmentScore and Members.
  summary(davidGeneCluster1)
  higherEnrichment<-which.max(enrichment(davidGeneCluster1))
  clusterGenes<-members(davidGeneCluster1)[[higherEnrichment]]
  wholeCluster<-cluster(davidGeneCluster1)[[higherEnrichment]]

  ##Now, we can obtain the ids of the first cluster directly using
  ##davidGeneCluster1 or by using DAVIDGenes class on the same cluster.
  ids(davidGeneCluster1)[[1]]
  ids(members(davidGeneCluster1)[[1]])}
### DAVIDTermCluster example:

Load the Gene Functional Classification Tool file report for the input demo file 2 to create a DAVIDGeneCluster object.
```
setwd(tempdir())
fileName<-system.file("files/termClusterReport2.tab.tar.gz", package="RDAVIDWebService")
untar(fileName)
davidTermCluster2<-DAVIDTermCluster(untar(fileName, list=TRUE))
davidTermCluster2
```

Now we can invoke DAVIDCluster ancestor functions to inspect the report data, of each cluster. For example, we can call summary to get a general idea, and the inspect the cluster with higher Enrichment Score, to see which members belong to it, etc. Or simply returning the whole cluster as a list with EnrichmentScore and Members.
```
summary(davidTermCluster2)
higherEnrichment<-which.max(enrichment(davidTermCluster2))
clusterGenes<-members(davidTermCluster2)[[higherEnrichment]]
wholeCluster<-cluster(davidTermCluster2)[[higherEnrichment]]
```

Then, we can obtain the ids of the term members calling clusterGenes object which is a DAVIDFunctionalAnnotationChart class or directly using ids on davidTermCluster2 for the higherEnrichment cluster.
```
geneIds <- ids(clusterGenes)
geneIds(davidTermCluster2)[[higherEnrichment]]
```
## S4 method for signature 'DAVIDWebService'
getIdTypes(object)

addList(object, inputIds, idType, listName, listType=c("Gene", "Background"))

## S4 method for signature 'DAVIDWebService'
addList(object, inputIds, idType, listName, listType=c("Gene", "Background"))

getAllAnnotationCategoryNames(object)

## S4 method for signature 'DAVIDWebService'
getAllAnnotationCategoryNames(object)

getDefaultCategoryNames(object)

## S4 method for signature 'DAVIDWebService'
getDefaultCategoryNames(object)

## S4 method for signature 'DAVIDWebService'
getGeneListNames(object)

geneListNames(object)

## S4 method for signature 'DAVIDWebService'
geneListNames(object)

## S4 method for signature 'DAVIDWebService'
geneListNames(object)

## S4 method for signature 'DAVIDWebService'
getSpecieNames(object)

## S4 method for signature 'DAVIDWebService'
getSpecieNames(object)

## S4 method for signature 'DAVIDWebService'
getCurrentGeneListPosition(object)

## S4 method for signature 'DAVIDWebService'
getCurrentGeneListPosition(object)

## S4 method for signature 'DAVIDWebService'
getCurrentBackgroundListPosition(object)

## S4 method for signature 'DAVIDWebService'
getCurrentBackgroundListPosition(object)
getCurrentSpeciesPosition(object)

## S4 method for signature 'DAVIDWebService'
getCurrentSpeciesPosition(object)

getTimeOut(object)

## S4 method for signature 'DAVIDWebService'
getTimeOut(object)

getHttpProtocolVersion(object)

## S4 method for signature 'DAVIDWebService'
getHttpProtocolVersion(object)

setCurrentGeneListPosition(object, position)

## S4 method for signature 'DAVIDWebService'
setCurrentGeneListPosition(object, position)

setCurrentBackgroundPosition(object, position)

## S4 method for signature 'DAVIDWebService'
setCurrentBackgroundPosition(object, position)

setCurrentSpecies(object, species)

## S4 method for signature 'DAVIDWebService'
setCurrentSpecies(object, species)

setAnnotationCategories(object, categories)

## S4 method for signature 'DAVIDWebService'
setAnnotationCategories(object, categories)

setTimeout(object, milliSeconds)

## S4 method for signature 'DAVIDWebService'
setTimeout(object, milliSeconds)

setHttpProtocolVersion(object, version)

## S4 method for signature 'DAVIDWebService'
setHttpProtocolVersion(object, version)

Arguments

object DAVIDWebService class object.
is.connected

inputIds character vector with the associated ids.
idType character with the type of submitted ids.
listName character to identify the submitted list.
listType character with the type of list (Gene, Background). Default value is "Gene".
position integer with the position of the gene/background list to set.
species numeric vector with the specie/s to use.
categories character vector with the category name/s to use in the analysis.
milliSeconds integer with time defined in milli seconds.
version character with HTTP_PROTOCOL_VERSION to use. At present available strings are: "1.1", "1.0", "HTTP/1.1" and "HTTP/1.0"

Details

Available functions include:

connect: Try to establish a connection with DAVID server using the provided email.
is.connected: Check if connected to DAVID server.
getIdTypes: Returns all acceptable DAVID idTypes.
addList: Add a gene or background to the current session.
getAllAnnotationCategoryNames: Returns all available annotation category names.
getDefaultCategoryNames: Returns all default category names.
getGeneListNames: Returns all list names
getBackgroundListNames: Returns background names.
getListName: Get the name of the selected list type at a given position.
getSpecieNames: Return specie/s of the current gene list.
getCurrentGeneListPosition: Return the position of current gene list.
getCurrentBackgroundListPosition: Return the position of current background list.
getCurrentSpeciesPosition: Return current specie/s used positions for the uploaded gene list.
setCurrentGeneListPosition: Use the gene list of the given position.
setCurrentBackgroundPosition: Use the background list of the given position.
setCurrentSpecies: Select the specie/s of the submitted gene list to use in the analysis.
setAnnotationCategories: Let the user to select specific annotation categories.
getTimeOut: Get apache Axis time out in milliSeconds.
setTimeOut: Set apache Axis time out in milliSeconds.
getHttpProtocolVersion: Get apache Axis HTTP_PROTOCOL_VERSION.
setHttpProtocolVersion: Set apache Axis HTTP_PROTOCOL_VERSION. possible values are defined in org.apache.axis2.transport.http.HTTPConstants class with HEADER_PROTOCOL_XX property. At present available strings are: "1.1", "1.0", "HTTP/1.1" and "HTTP/1.0".
is.connected

Value

according to the call one of the following objects can be returned

is.connected  TRUE if user has registered email with DAVID knowledge base, FALSE otherwise.

getIdTypes  character vector with the available DAVID input ID type.

addList  list with two items: i)inDavid, a numeric with the percentage of the inputIds in DAVID knowledge database, ii) unmappedIds, a character vector with the unmapped ids if listType is "Gene", NA_character_ otherwise.

getAllAnnotationCategoryNames  character vector with the available DAVID annotation categories.

getDefaultCategoryNames  character vector with a subset of the available DAVID annotation categories, chosen by default.

geneListNames  return a character vector with the name of the submitted gene list/s.

getBackgroundListNames  character vector with the name of the available background gene list/s for the submitted gene list/s.

getListName  character with the name of the list.

getSpecieNames  character vector with the specie/s and in brackets the number of DAVID Ids of the current gene list, e.g. Homo sapiens(155).

getCurrentGeneListPosition  integer with the position of current gene list if available, NA_integer_ otherwise.

getCurrentBackgroundListPosition  integer with the position of current background list if available, NA_integer_ otherwise.

getCurrentSpeciesPosition  integer vector with the specie/s position under use for the gene list under use if available, NA_character_ otherwise.

See Also


plot2D

Visualization of biological relationships

Description

plot2D uses a 2D tile ggplot to explore biological relationships between two variables such as annotation category and genes, for Functional Annotation Chart/Table or Term cluster results. For Gene cluster, the cluster number vs genes membership is plotted.

Usage

plot2D(object,...)

## S4 method for signature 'DAVIDResult'
plot2D(object, dataFrame)

## S4 method for signature 'DAVIDFunctionalAnnotationChart'
plot2D(object,color=c("FALSE"="black", "TRUE"="green"))

## S4 method for signature 'DAVIDGeneCluster'
plot2D(object,color=c("FALSE"="black", "TRUE"="green"),names=FALSE)

## S4 method for signature 'DAVIDTermCluster'
plot2D(object,number=1,color=c("FALSE"="black", "TRUE"="green"),names=FALSE)

Examples

david <- DAVIDWebService$new()
david$is.connected()

## Or the equivalent S4 style function call
is.connected(david)
## S4 method for signature 'DAVIDFunctionalAnnotationTable'

```r
plot2D(object,
    category, id, names.genes=FALSE,
    names.category=FALSE, color=c("FALSE"="black","TRUE"="green"))
```

### Arguments

- **object**
  - DAVIDResult heirs (DAVIDFunctionalAnnotationChart/Table or DAVIDGeneCluster/TermCluster)

- **dataFrame**
  - data.frame with three columns (x, y and fill) to be used in ggplot. X(Y) is a character/factor with the X(Y)-axis labels and "fill" is a the color to be used for x-y labels.

- **color**
  - named character vector to indicate tile color. Default value is c("FALSE"="black","TRUE"="green").

- **names**
  - should gene names be plotted? Default value is FALSE, i.e, use ids.

- **number**
  - integer to indicate which cluster to plot. Default value is 1.

- **category**
  - character vector to select the main annotation categories. By default is missing in order to use all the available ones.

- **id**
  - character vector to indicate which gene ids to use. By default is missing in order to use all the available ones.

- **names.genes, names.category**
  - Should genes and/or category names used? Default value is FALSE, i.e., use both ids.

- **...**
  - Additional parameters for heirs functions.

### Value

- a ggplot object if the object is not empty.

### Author(s)

Cristobal Fresno and Elmer A Fernandez

### See Also

- Other DAVIDFunctionalAnnotationChart: DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart-class, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable-class, DAVIDGeneCluster, DAVIDGeneCluster, DAVIDGenes, DAVIDGenes, DAVIDGeneCluster, DAVIDTermCluster, DAVIDTermCluster, as, as, as, categories, categories, categories, categories, ids, ids, ids, ids, initialize, initialize, initialize, initialize, initialize, initialize
- Other DAVIDFunctionalAnnotationTable: DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable-class, DAVIDGeneCluster, DAVIDGeneCluster, DAVIDGenes, DAVIDGenes, DAVIDGeneCluster, DAVIDTermCluster, as, as, as, categories, categories, categories, categories, dictionary, dictionary, genes, genes, genes, genes, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, membership, membership, subset, subset
plot2D

Other DAVIDGeneCluster: DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationTable, DAVIDGODag, DAVIDGeneCluster, DAVIDTermCluster, DAVIDGeneCluster-class, DAVIDGenes, DAVIDGenes, DAVIDGeneCluster, DAVIDGeneCluster-class, DAVIDGenes, DAVIDGenes, DAVIDGenes, DAVIDGenes, DAVIDGenes, DAVIDTermCluster, DAVIDTermCluster, DAVIDTermCluster-class, as, as, as, genes, genes, genes, ids, ids, ids, ids, ids, ids, ids, ids, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize

Other DAVIDResult: DAVIDResult-class, type, type

Other DAVIDTermCluster: DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDGODag, DAVIDGODag, DAVIDGeneCluster, DAVIDGeneCluster, DAVIDGeneCluster, DAVIDGeneCluster-class, DAVIDGenes, DAVIDGenes, DAVIDGenes, DAVIDTermCluster, DAVIDTermCluster, DAVIDTermCluster-class, as, as, as, ids, ids, ids, ids, ids, ids, ids, ids, ids, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize

Examples

{
    ##DAVIDFunctionalAnnotationChart example:
    ##Load the Functional Annotation Chart file report for the input demo file 2, using data function. Just to keep it simple, for the first five terms present in funChart2 object, create a DAVIDFunctionalAnnotationChart object and plot a 2D tile matrix with the reported evidence (green) or not (black).
    data(funChart2)
    plot2D(DAVIDFunctionalAnnotationChart(funChart2[1:5,]), color=c("FALSE"="black", "TRUE"="green"))

    ##DAVIDFunctionalAnnotationTable example
    ##Load the Functional Annotation Table file report for the input demo file 1, using data function. Then, create a DAVIDFunctionalAnnotationTable object using the loaded data.frame annotationTable1. Then, plot the membership of only for the first six terms in this category, with only the genes of the first six terms with at least one evidence code.
    data(annotationTable1)
    davidFunTable1<-DAVIDFunctionalAnnotationTable(annotationTable1)
    categorySelection<-list(head(dictionary(davidFunTable1, categories(davidFunTable1)[1]))$ID))
    names(categorySelection)<-categories(davidFunTable1)[1]

    ##Gene filter...
    id<-membership(davidFunTable1, categories(davidFunTable1)[1])[,1:6]
    id<-ids(genes(davidFunTable1)))[rowSums(id)>0]

    ##Finally the membership tile plot
    plot2D(davidFunTable1, category=categorySelection, id=id, names.category=TRUE)

    ##DAVIDGeneCluster example:
    ##Load the Gene Functional Classification Tool file report for the input demo list 1 file to create a DAVIDGeneCluster object.
    setwd(tempdir())
    fileName<-system.file("files/geneClusterReport1.tab.tar.gz",
}
## Accessor methods for DAVIDWebService class

### Description

Setter/getters for DAVIDWebService class fields.

### Usage

```r
setEmail(object, mail)
```

## setEmail

### Description

Setter/getters for DAVIDWebService class fields.

### Usage

```r
setEmail(object, mail)
```
**setEmail**

**Arguments**

- **object**: DAVIDWebService class object.
- **mail**: character with a registered e-mail account at DAVID’s website.

**Details**

Note that DAVIDWebService is a Reference class, hence invoke it using object_name$setter/getter(parameters). In addition, the user can use the S4 version style function call.

**Value**

according to the call one of the following objects can be returned

- **setEmail**: character with the given e-mail to set.
- **getEmail**: character with the e-mail under use.
- **getstub**: jobjRef object with the stub java object to interface with DAVID API.

**References**

1. DAVID web [http://david.abcc.ncifcrf.gov](http://david.abcc.ncifcrf.gov)

**See Also**


**Examples**

```r
(  
  ##Create a DAVIDWebService object  
david<-DAVIDWebService$new()  
  
  ##Invoke Reference class style function setter/getters  
david$setEmail("valid_mail@david.org")  
david$getEmail()  
  stub<-david$getStub()  
  
  ##Or the equivalent S4 style function call setter/getters  
  setEmail(david, "valid_mail@david.org")  
  getEmail(david)  
```
show<-getStub(david)
)

show Basic console output

Description
The different implementations of show function for the DAVIDWebService package classes.

Usage
## S4 method for signature 'DAVIDResult'
show(object)

## S4 method for signature 'DAVIDGenes'
show(object)

## S4 method for signature 'DAVIDFunctionalAnnotationChart'
show(object)

## S4 method for signature 'DAVIDCluster'
show(object)

## S4 method for signature 'DAVIDFunctionalAnnotationTable'
show(object)

## S4 method for signature 'DAVIDWebService'
show(object)

Arguments
object DAVIDXX class members (where XX stands for Result, Genes, Term/GeneCluster, FunctionalAnnotationChart/Table or DAVIDWebService).

Value
Basic console output.

Author(s)
Cristobal Fresno and Elmer A Fernandez

Examples
{
##DAVIDGenes example:
##Load Show Gene List file report for the input demo file 1, using data
data function. Then, create a DAVIDGenes object using only the head of the
##loaded data frame geneList1 (just to keep it simple).
data(geneList1)
davidGenes1<-DAVIDGenes(head(geneList1))
davidGenes1
## DAVID Functional Annotation Chart example

Load the Functional Annotation Chart file report for the input demo file 2, using data function. Then, create a DAVIDFunctionalAnnotationChart object using the head of the loaded data.frame funChart2 (just to keep it simple).

```r
data(funChart2)
davidFunChart2 <- DAVIDFunctionalAnnotationChart(head(funChart2))
davidFunChart2
```

## DAVID Functional Annotation Table example:

Load the Functional Annotation Table file report for the input demo file 1, using data function. Then, create a DAVIDFunctionalAnnotationTable object using the loaded data.frame annotationTable1.

```r
data(AnnotationTable1)
davidFunTable1 <- DAVIDFunctionalAnnotationTable(AnnotationTable1)
davidFunTable1
```

---

### Methods for DAVIDGenes class object

#### Description

Obtain DAVIDGenes related information, according to the given function call (see Values).

#### Usage

```r
species(object)
```

#### Arguments

- `object` DAVIDGenes class object.
- `collapse` logical indicating if duplicate ids should be grouped as a comma separated id. Default value is FALSE.
- `...` Additional parameters for internal functions (if applicable).
Value

according to the call one of the following objects can be returned

- show: console output of the class and associated data.
- species: character vector with the levels of Species if available.
- uniqueIds: a DAVIDGenes object with only the gene names with a unique id.
- duplicateIds: a DAVIDGenes object with only the gene names with at least two ids. If collapse is TRUE, a data.frame in where all the ids that matched the same gene name, are coded in comma separated style.

Author(s)

Cristobal Fresno and Elmer A Fernandez

Examples

```r
##Load Show Gene List file report for the input demo file 1, using data function. Then, create a DAVIDGenes object using the loaded data.frame geneList1. In addition, the user can use the file name of the downloaded file report.
data(geneList1)
davidGenes1<-DAVIDGenes(geneList1)

##Now we can inspect davidGenes1 as it was an common data.frame head(davidGenes1)

##Additional getters for this object are also available, to obtain the different columns: ids, genes and species.
ids(davidGenes1)
genes(davidGenes1)
species(davidGenes1)

##Or even look up for a particular gene id, which will return only the matched ones.
genes(davidGenes1, ids=c("38926_at", "35367_at", "no match"))

##Obtain the genes with duplicate manufacturer ids or just the genes that do not have duplicate ids (uniqueIds).
duplicateIds(davidGenes1)
uniqueIds(davidGenes1)
```
**Usage**

```r
subset(x, ...)  
## S4 method for signature 'DAVIDFunctionalAnnotationTable'
subset(x, selection = c("Membership", "Dictionary"), category, drop = TRUE)

dictionary(object, ...)
## S4 method for signature 'DAVIDFunctionalAnnotationTable'
dictionary(object, ...)

membership(object, ...)
## S4 method for signature 'DAVIDFunctionalAnnotationTable'
membership(object, ...)
```

**Arguments**

- `object, x` DAVIDFunctionalAnnotationTable class object.
- `selection` which slot to use to obtain the subset. Possible values are "Membership" or "Dictionary".
- `category` named list with main annotation category, which contains a character vector with the ids to use. Default value is missing in order to use all available categories of the report.
- `drop` Should list structure be drop if length==1? Default value TRUE.
- `...` Additional parameters for subset function call.

**Value**

according to the call one of the following objects can be returned

- `subset` list with filtered categories/ids according to function call.
- `enrichment` numeric vector with DAVID cluster’s enrichment score.
- `members` list with DAVID Cluster’s members.

**Author(s)**

Cristobal Fresno and Elmer A Fernandez

**See Also**

Other DAVIDCluster: DAVIDCluster-class, cluster, cluster, cluster, enrichment, enrichment, members, members, summary, summary, summary, summary

Other DAVIDFunctionalAnnotationTable: DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable-class, DAVIDGODag, DAVIDGODag, DAVIDGeneCluster, DAVIDGeneCluster, DAVIDGenes, DAVIDGenes, DAVIDGenes, DAVIDTermCluster, DAVIDTermCluster, as, as, as, categories, categories, categories, genes,
genes, genes, genes, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D

Other DAVIDFunctionalAnnotationTable: DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable-class, DAVIDGODag, DAVIDGODag, DAVIDGeneCluster, DAVIDGeneCluster, DAVIDGenes, DAVIDGenes, DAVIDGenes, DAVIDTermCluster, DAVIDTermCluster, as, as, as, categories, categories, categories, genes, genes, genes, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D

Examples

{##Load the Functional Annotation Table file report for the input demo##file 1, using data function. Then, create a DAVIDFunctionalAnnotationTable##object using the loaded data.frame annotationTable1.##
data(annotationTable1)##
davidFunTable1<-DAVIDFunctionalAnnotationTable(annotationTable1)

##Obtain the head of the dictionary and the membership matrix for the first##annotated genes used in davidFunTable1 object.##
head(membership(davidFunTable1, categories(davidFunTable1)[1]))##
head(dictionary(davidFunTable1, categories(davidFunTable1)[1]))##
head(genes(davidFunTable1))
}

summary (Basic summary for DAVIDWebService package classes.)

Description

The different implementations of summary function for the DAVIDWebService package classes.

Usage

summary(object, ...)## S4 method for signature 'DAVIDCluster'## summary(object)

## S4 method for signature 'DAVIDGODag'## summary(object, ...)## S4 method for signature 'DAVIDWebService'## summary(object)

Arguments

object DAVIDXX class members (where XX stands for Term/GeneCluster, GODag or DAVIDWebService).

... Additional parameters.
summary

Value

data.frame with summary output.

Author(s)

Cristobal Fresno and Elmer A Fernandez

See Also

Other DAVIDCluster: DAVIDCluster-class, cluster.cluster, dictionary.dictionary, enrichment, membership.membership, subset.

Other DAVIDGODag: DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationTable, DAVIDGeneCluster, DAVIDGODag-class, DAVIDTermCluster, DAVIDGODag, DAVIDGODag-class, DAVIDGODag-class, DAVIDGODag-class.


Examples

{
  ##DAVIDGODag example:
  ##Load the Functional Annotation Chart file report for the input demo
  ##file 2, using data function. Then, create a DAVIDGODag object using
  ##Molecular Function main category of DAVIDFunctionalAnnotationChart object,
  ##obtained from the loaded data.frame funChart2. In addition, we have
  ##selected a threshold pvalue of 0.001 and removed unattached nodes, in case
  ##DAVID/GO.db database are not using the same version.
  data(funChart2)
  davidGODag<-DAVIDGODag(DAVIDFunctionalAnnotationChart(funChart2), type="MF",
                         pvalueCutoff=0.001, removeUnattached=TRUE)
  summary(davidGODag)

  ##DAVIDGeneCluster example:
## Load the Gene Functional Classification Tool file report for the input demo list 1 file to create a DAVIDGeneCluster object.
setwd(tempdir())
fileName<-system.file("files/geneClusterReport1.tab.tar.gz", package="RDAVIDWebService")
untar(fileName)
davidGeneCluster1<-DAVIDGeneCluster(untar(fileName, list=TRUE))
davidGeneCluster1

## Now we can invoke DAVIDCluster ancestor functions to inspect the report data, of each cluster. For example, we can call summary to get a general idea
summary(davidGeneCluster1)

## DAVIDTermCluster example:
## Load the Gene Functional Classification Tool file report for the input demo file 2 to create a DAVIDGeneCluster object.
setwd(tempdir())
fileName<-system.file("files/termClusterReport2.tab.tar.gz", package="RDAVIDWebService")
untar(fileName)
davidTermCluster2<-DAVIDTermCluster(untar(fileName, list=TRUE))
davidTermCluster2

## Now we can invoke DAVIDCluster ancestor functions to inspect the report data, of each cluster. For example, we can call summary to get a general idea
summary(davidTermCluster2)

---

### terms

*Methods for DAVIDGODag class object*

**Description**

Obtain DAVIDGODag related information, according to the given function call (see Values).

**Usage**

```r
terms(x, ...)

## S4 method for signature 'DAVIDGODag'
terms(x, ...)

percentages(object)

## S4 method for signature 'DAVIDGODag'
percentages(object)

listTotals(object)

## S4 method for signature 'DAVIDGODag'
```
listTotals(object)

popHits(object)

## S4 method for signature 'DAVIDGODag'
popHits(object)

popTotals(object)

## S4 method for signature 'DAVIDGODag'
popTotals(object)

foldEnrichments(object)

## S4 method for signature 'DAVIDGODag'
foldEnrichments(object)

bonferronis(object)

## S4 method for signature 'DAVIDGODag'
bonferronis(object)

benjaminis(object)

## S4 method for signature 'DAVIDGODag'
benjaminis(object)

fdrs(object)

## S4 method for signature 'DAVIDGODag'
fdrs(object)

counts(object, ...)

## S4 method for signature 'DAVIDGODag'
counts(object, ...)

upsideDown(graph)

## S4 method for signature 'graph'
upsideDown(graph)

universeCounts(r)

## S4 method for signature 'DAVIDGODag'
universeCounts(r)

## S4 method for signature 'DAVIDGODag'
universeMappedCount(r)

Arguments

object,x,r DAVIDGODag class object.

graph a graph object with the GO DAG structure.
Additional parameters (if required).

**Value**

according to the call one of the following objects can be returned

upsideDown  the same graph but the arcs with its directions in the other way around. Hence, plot layout would make upside down the graph.

universeMappedCount, universeCounts, counts  modifications to the corresponding GOstats/Category library functions, to keep the same behavior for DAVIDGODag objects.

fdr, benjamini, bonferroni  

adjusted method specific p-values for the corresponding nodes/terms.

terms  character vector with GO node names.

popTotals, popHits, listTotals  integer vector with the number of ids, to use in the EASE score calculations, when building the 2x2 contingency table.

percentages  numeric vector with the percentage of the gene list ids present in the term.

foldEnrichments  numeric vector with the ratio of the two proportions for each node/term. For example, if 40/400 (i.e. 10%) of your input genes involved in "kinase activity" and the background information is 300/30000 genes (i.e. 1%) associating with "kinase activity", roughly 10%/1%=10 fold enrichment.

**Author(s)**

Cristobal Fresno and Elmer A Fernandez

**See Also**

Other DAVIDGODag: DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationChart, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDFunctionalAnnotationTable, DAVIDGODag, DAVIDGODag, DAVIDGODag-class, DAVIDGeneCluster, DAVIDGeneCluster, DAVIDGenes, DAVIDGenes, DAVIDGenes, DAVIDTermCluster, DAVIDTermCluster, as, as, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize, initialize.

**Examples**

```r
##Load the Functional Annotation Chart file report for the input demo
##file 2, using data function. Then, create a DAVIDGODag object using
##Molecular Function main category of DAVIDFunctionalAnnotationChart object,
##obtained from the loaded data.frame funChart2. In addition, we have
##selected a threshold pvalue of 0.001 and removed unattached nodes, in case
##DAVID/GO.db database are not using the same version.
data(funChart2)
davidGODag<-DAVIDGODag(DAVIDFunctionalAnnotationChart(funChart2), type="MF", pvalueCutoff=0.001, removeUnattached=TRUE)

##Now, we can inspect the enrichment GO DAG using GOstats functionalities:
counts, pvalues, sigCategories, universeCounts, geneMappedCount, etc.
##However, oddsRatios, expectedCounts and universeMappedCount are not
##available because these results are not available on DAVID’s Functional
```
#Annotation Chart report. In addition geneIdUniverse are not the ones of the universe but the ids on the category (geneIdsByCategory).

davidGODag
counts(davidGODag)
pvalues(davidGODag)
sigCategories(davidGODag, p=0.0001)
universeCounts(davidGODag)
geneMappedCount(davidGODag)
geneIdsByCategory(davidGODag)
summary(davidGODag)

##In addition, the new nodeData attributes (term, listTotal, popHit, popTotal, foldEnrichment, bonferroni, benjamini, fdr) can be retrieved.

terms(davidGODag)
listTotals(davidGODag)
popHits(davidGODag)
popTotals(davidGODag)
foldEnrichments(davidGODag)
bonferronis(davidGODag)
benjaminis(davidGODag)
fdrs(davidGODag)
}

<table>
<thead>
<tr>
<th>type</th>
<th>Getters for DAVIDResult object</th>
</tr>
</thead>
</table>

**Description**

Obtain DAVIDResult slot information, according to the given function call (see values).

**Usage**

```
## S4 method for signature 'DAVIDResult'
type(object)
```

**Arguments**

- `object` DAVIDResult class object.

**Value**

according to the call one of the following objects can be returned

- `type` character with type slot datum.

**Author(s)**

Cristobal Fresno and Elmer A Fernandez

**See Also**

Other DAVIDResult: DAVIDResult-class, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D, plot2D
## Index

### Topic **DAVID**
- DAVIDGenes-class, 20
- DAVIDResult-class, 23
- DAVIDWebService-class, 26
- DAVIDWebService-package, 3

### Topic **MEA**
- DAVIDWebService-class, 26
- DAVIDWebService-package, 3

### Topic **SEA**
- DAVIDWebService-class, 26
- DAVIDWebService-package, 3

### Topic **classes**
- DAVIDCluster-class, 9
- DAVIDFunctionalAnnotationChart-class, 10
- DAVIDFunctionalAnnotationTable-class, 11
- DAVIDGeneCluster-class, 14
- DAVIDGenes-class, 20
- DAVIDGODag-class, 21
- DAVIDResult-class, 23
- DAVIDTermCluster-class, 25

### Topic **datasets**
- annotationSummary1, 3
- annotationTable1, 4
- demoList1, 29
- funChart1, 30
- geneCluster1, 31
- geneList1, 32

### Topic **genes**
- DAVIDGenes-class, 20

```r
addList, 29, 39, 53, 59
addList (is.connected), 42
addList, DAVIDWebService-method (is.connected), 42
annotationSummary1, 3, 30, 33
annotationSummary2, 30, 33
annotationSummary2 (annotationSummary1), 3
annotationTable1, 4
annotationTable2 (annotationTable1), 4
as, 6, 11, 13, 15, 17, 21, 22, 25, 34, 40, 41, 50, 51, 57–59, 62
as (DAVIDGenes), 15
benjaminis, 18, 22, 59
benjaminis (terms), 60
benjaminis, DAVIDGODag-method (terms), 60
bonferronis, 18, 22, 59
bonferronis (terms), 60
bonferronis, DAVIDGODag-method (terms), 60
categories, 5, 11, 13, 18, 34, 40, 50, 57, 58
categories, DAVIDFunctionalAnnotationChart-method (categories), 5
categories, DAVIDFunctionalAnnotationTable-method (categories), 5
category, 7, 10, 57, 59
category, DAVIDCluster-method (cluster), 7
category-methods (cluster), 7
connect, 29, 39, 53, 59
connect (is.connected), 42
connect, DAVIDWebService-method (is.connected), 42
counts, 18, 22, 59
counts (terms), 60
counts, DAVIDGODag-method (terms), 60
DAVIDCluster-class, 9
DAVIDFunctionalAnnotationChart, 6, 11, 13, 15, 21, 22, 25, 34, 40, 41, 50, 51, 57–59, 62
DAVIDFunctionalAnnotationChart (DAVIDGenes), 15
DAVIDFunctionalAnnotationChart, character-method (DAVIDGenes), 15
DAVIDFunctionalAnnotationChart, data.frame-method (DAVIDGenes), 15
DAVIDFunctionalAnnotationChart-class, 10
DAVIDFunctionalAnnotationChart-methods (DAVIDGenes), 15
DAVIDFunctionalAnnotationChart-tables, 6, 11, 13, 15, 21, 22, 25, 34, 40, 41, 50, 51, 57–59, 62
```
DAVIDFunctionalAnnotationTable
  (DAVIDGenes), 15
duplicateIds(species), 55
duplicateIds-methods(species), 55
duplicateIds-methods(species), 55
enrichment, 10, 57, 59
enrichment(cluster), 7
enrichment,DAVIDCluster-method (cluster), 7
dict.

dict.
dict., 4, 30, 32
dict.

geneCluster1, 31
geneCluster2(geneCluster1), 31
genelist1, 4, 30, 32
genelist2, 4, 30
genelist2(genelist1), 32
genes, 6, 13, 15, 18, 21, 33, 40, 41, 50, 51, 57, 59, 58
genes,DAVIDFunctionalAnnotationTable-method (genes), 33
genes,DAVIDGeneCluster-method (genes), 33
genes,DAVIDGenes-method (genes), 33
genes-methods(genes), 33
genes(allAnnotationCategoryNames, 29, 39, 53, 59
genes,allAnnotationCategoryNames(is.connected), 42
genes,allAnnotationCategoryNames,DAVIDWebService-method (is.connected), 42
genes,allAnnotationSummary,DAVIDWebService-method (is.connected), 42
genes,allAnnotationSummary
  (getGeneCategoriesReport), 35
genes,allAnnotationSummary,DAVIDWebService-method (getGeneCategoriesReport), 35
genes,backgroundListNames, 29, 39, 53, 59
genes,backgroundListNames(is.connected), 42
genes,backgroundListNames,DAVIDWebService-method (is.connected), 42

dict.
dict.
dict.
dict., 56
dict.

DAVIDGeneCluster
  6, 11, 13, 15, 21, 22, 25, 34, 40, 41, 50, 51, 57–59, 62
DAVIDGeneCluster(DAVIDGenes), 15
DAVIDGeneCluster,character-method
  (DAVIDGenes), 15
DAVIDGeneCluster-class, 14
DAVIDGeneCluster-methods(DAVIDGenes), 15
DAVIDGenes, 6, 11, 13, 15, 21, 22, 25, 34, 40, 41, 50, 51, 57–59, 62
DAVIDGenes,character-method
  (DAVIDGenes), 15
DAVIDGenes,data.frame-method
  (DAVIDGenes), 15
DAVIDGenes-class, 20
DAVIDGenes-methods(DAVIDGenes), 15
DAVIDGODag, 6, 11, 13, 15, 21, 22, 25, 34, 40, 41, 50, 51, 57–59, 62
DAVIDGODag(DAVIDGenes), 15
DAVIDGODag,DAVIDFunctionalAnnotationChart-method
  (DAVIDGenes), 15
DAVIDGODag-class, 21
DAVIDGODag-methods(DAVIDGenes), 15
DAVIDResult-class, 23
DAVIDTermCluster, 6, 11, 13, 15, 21, 22, 25, 34, 40, 41, 50, 51, 57–59, 62
DAVIDTermCluster(DAVIDGenes), 15
DAVIDTermCluster,character-method
  (DAVIDGenes), 15
DAVIDTermCluster-class, 25
DAVIDTermCluster-methods(DAVIDGenes), 15
DAVIDWebService
  (DAVIDWebService-class), 26
DAVIDWebService-class, 26
DAVIDWebService-package, 3
demoList1, 4, 29, 33
demoList2, 4, 33
demoList2(demoList1), 29
dict.

DAVIDFunctionalAnnotationTable-methods(subset), 56
dict.

dict.

dict.
getClusterReport
  (getGeneCategoriesReport), 35
getClusterReport, DAVIDWebService-method
  (getGeneCategoriesReport), 35
getClusterReportFile, 29, 46–49, 53, 59
getClusterReportFile
  (getGeneCategoriesReport), 35
getClusterReportFile, DAVIDWebService-method
  (getGeneCategoriesReport), 35
getCurrentBackgroundListPosition
  (is.connected), 42
getCurrentBackgroundListPosition, DAVIDWebService-method
  (is.connected), 42
getCurrentGeneListPosition
  (is.connected), 42
getCurrentGeneListPosition, DAVIDWebService-method
  (is.connected), 42
getCurrentSpeciesPosition
  (is.connected), 42
getCurrentSpeciesPosition, DAVIDWebService-method
  (is.connected), 42
genericCategoryNames, 29, 39, 53, 59
genericCategoryNames (is.connected), 42
genericCategoryNames, DAVIDWebService-method
  (is.connected), 42
genericCategoryNames, DAVIDWebService-method
  (is.connected), 42
genericListReportFile, 29, 46–49, 53, 59
genericListReportFile
  (getGeneCategoriesReport), 35
genericListReport, DAVIDWebService-method
  (getGeneCategoriesReport), 35
autocompleteSpecies, 29, 39, 53, 59
autocompleteSpecies, DAVIDWebService-method
  (getGeneCategoriesReport), 35
getEmail
  (setEmail), 52
getEmail, DAVIDWebService-method
  (setEmail), 52
getFunctionalAnnotationChart, 29, 46–49, 53, 59
getFunctionalAnnotationChart
  (getGeneCategoriesReport), 35
getFunctionalAnnotationChart, DAVIDWebService-method
  (getGeneCategoriesReport), 35
getFunctionalAnnotationChartFile, 29, 46–49, 53, 59
getFunctionalAnnotationChartFile
  (getGeneCategoriesReport), 35
getFunctionalAnnotationChartFile, DAVIDWebService-method
  (getGeneCategoriesReport), 35
getFunctionalAnnotationTable, 29, 46–49, 53, 59
getFunctionalAnnotationTable
  (getGeneCategoriesReport), 35
getFunctionalAnnotationTable, DAVIDWebService-method
  (getGeneCategoriesReport), 35
getGeneCategoriesReport, 29, 35, 46–49, 53, 59
getGeneCategoriesReport, DAVIDWebService-method
  (getGeneCategoriesReport), 35
geneListNames, 29, 39, 53, 59
geneListNames (is.connected), 42
geneListNames, DAVIDWebService-method
  (is.connected), 42
geneListReport, 29, 46–49, 53, 59
geneListReport
  (getGeneCategoriesReport), 35
geneListReport, DAVIDWebService-method
  (getGeneCategoriesReport), 35
getHttpProtocolVersion
  (is.connected), 42
getHttpProtocolVersion, DAVIDWebService-method
  (is.connected), 42
getIdTypes
  (is.connected), 42
getIdTypes, DAVIDWebService-method
  (is.connected), 42
getListName
  (is.connected), 42
getListName
  (is.connected), 42
getListName, DAVIDWebService-method
  (is.connected), 42
getSpecieNames
  (is.connected), 42
getSpecieNames
  (is.connected), 42
getSpecieNames, DAVIDWebService-method
  (is.connected), 42
getStub
  (setEmail), 52
getStub, DAVIDWebService-method
  (setEmail), 52
getTimeOut
  (is.connected), 42
timeOut, DAVIDWebService-method
  (is.connected), 42
ids, 6, 11, 15, 18, 21, 25, 34, 39, 50, 51
ids, DAVIDFunctionalAnnotationChart-method
  (ids), 39
  (ids), DAVIDGeneCluster-method (ids), 39
subset, 6, 8, 10, 13, 18, 34, 50, 56, 59
subset, DAVID Functional Annotation Table-method (subset), 56
summary, 8, 10, 18, 23, 29, 39, 46–49, 53, 57, 58, 62
summary, DAVID Cluster-method (summary), 58
summary, DAVID GODag-method (summary), 58
summary, DAVID Web Service-method (summary), 58

termCluster1 (geneCluster1), 31
termCluster2 (geneCluster1), 31
terms, 18, 23, 59, 60
terms, DAVID GODag-method (terms), 60
type, 24, 51, 63
type, DAVID Result-method (type), 63

uniqueIds (species), 55
uniqueIds, DAVID Genes-method (species), 55
uniqueIds-method (species), 55
universeCounts, 18, 23, 59
universeCounts (terms), 60
universeCounts, DAVID GODag-method (terms), 60
universeMappedCount, 18, 23, 59
universeMappedCount (terms), 60
universeMappedCount, DAVID GODag-method (terms), 60
upsideDown, 18, 23, 59
upsideDown (terms), 60
upsideDown, graph-method (terms), 60