Package 'GOdescrCalculus'

January 27, 2016

Type Package

Title Creates GO descriptors

Version 1.0

Date 2015-04-16

Author Georgia Tsiliki

Maintainer Georgia Tsiliki <gtsiliki@central.ntua.gr>

Depends RCurl, pmml, org.Hs.eg.db, GSEABase, GOstats, Rgraphviz, blockcluster, Matrix, vegan

Description
This package produces GO descriptors based on GO ontology and proteomics data supplied.

License GPL-2

R topics documented:

`GOdescrCalculus-package`

Produce GO descriptors given an omics data

Description

This package expects an omics data (e.g. proteomics/genomics), and produces a set of GO descriptors given the data and the parameters supplied based on the data and the algorirthm selected; bi-clustering and hierarchical clustering algorithms are currently implemented. The clustering algorithm parameters can be stored as an R raw model and used for 'prediction', i.e. to construct GO decsriptors for the same data.

Details

Package:	GOdescrCalculus
Type:	Package
Version:	1.0
Date:	2015-04-16
License:	GPL-2

Generate GO descriptors given an omics data. Important functions are generate.descr.biclust and generate.descr.hierar.

Author(s)

Georgia Tsiliki

Maintainer: Georgia Tsiliki <gtsiliki@central.ntua.gr>

References

Package source is in the help file for package.skeleton

Examples

```
data("dat1p")
data("dat1m1")

adInfo<- list()

res1<- read.in.json.for.pred(dat1p, dat1m1, adInfo)
```

---

dat1 *A sample data object*

Description

The dataset for this test is a data frame

Usage

```
data("dat1")
```

Format

A list of two objects

datasetURI a character vector- ambit data set uri

dataEntry a data frame containing two columns: compound and values. Compound is a character vector with all compound anbit uris, and values is a data frame with all numberic values of the proteomics data set (compounds by features). Only dependent features are included.

Details

There are no more details

Source

The source of this function is in the

References

There are no references

Examples

```
data(dat1)
## maybe str(dat1) ; plot(dat1) ...
```

---

dat1m1 *Serialized list of parameters for biclustering algorithm*

Description

A character string for a serialized parameters list, i.e. a set of values needed from the biclusteruing algorithm in order to produce the GO descriptors. The list includes 'key' (e.g. 'UNIPROT'), 'onto' (e.g. c('GO','MF')), 'pvalCutoff' for hypergeometric test (e.g. 0.05), 'nclust' for the number of clusters in the x and y axis respectively (e.g c(4,2)), and 'FUN' to specify the function used to summarize the omics data (e.g. 'mean').

Usage

```
data("dat1m1")
```

Format

A character string

Details

Example set of parameters needed for generate.descr.biclust function

Source

The source of this function is in the

References

There are no references

Examples

```
data(dat1m1)
## maybe str(dat1m1) ; plot(dat1m1) ...
```

---

dat1m2 *Serialized list of parameters for hierarchical clustering algorithm*

Description

A character string for a serialized parameters list, i.e. a set of values needed from the biclusteruing algorithm in order to produce the GO descriptors. The list includes 'key' (e.g. 'UNIPROT'), 'onto' (e.g. c('GO','MF')), 'pvalCutoff' for hypergeometric test (e.g. 0.05), 'distMethod' (e.g. 'euclidean' or other alternatives from the vegan package), 'hclustMethod' (e.g. 'ward.D2' or other alternatives from the vegan package), 'nORh' th enumber of clusters in the data (e.g. 10), and 'FUN' to specify the function used to summarize the omics data (e.g. 'mean').

Usage

```
data("dat1m2")
```

Format

A character string

Details

Example set of parameters needed for generate.descr.hierar function

Source

The source of this function is in the

References

There are no references

Examples

```
data(dat1m2)
## maybe str(dat1m2) ; plot(dat1m2) ...
```

dat1p *A sample data object*

Description

The dataset for this test is a data frame

Usage

```
data("dat1p")
```

Format

A list of two objects

datasetURI a character vector- ambit data set uri

dataEntry a data frame containing two columns: compound and values. Compound is a character vector with all compound anbit uris, and values is a proteomics data frame with all numberic values of the data set (compounds by features)

Details

Data set for prediction with dat1m

Source

The source of this function is in the

References

There are no references

Examples

```
data(dat1p)
## maybe str(dat1p) ; plot(dat1p) ...
```

`generate.descr.biclust` *Produce GO descriptors using bi-clustering algorithm*

Description

This function is used to estimate GO descriptors for omics data, based on GO ontology. Data are clustered based on bi-clustering algorithm from the blockcluster R package using default values. The user needs to specify the number of clusters for each axes.

Usage

```
generate.descr.biclust(dataset, rawModel, additionalInfo)
```

Arguments

`dataset`	Data for prediction. A list of two objects: datasetURI (a character string), dataEntry (a data frame).
`rawModel`	A serialized list of parameters for biclustering, as produced by generate.param.model
`additionalInfo`	Any additional information needed for rawModel. Here an empty list.

Details

No details required

Value

A data frame giving the new GO descriptors given the function's parameters. Number of columns are the number of new GO descriptors, number of rows is the number of features used.

Note

No notes for this function

Author(s)

Georgia Tsiliki

References

The help file of skeleton

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

data("dat1p")
data("dat1m1")

adInfo<- list()

pred.res<- generate.descr.biclust(dat1p, dat1m1, adInfo)
```

`generate.descr.hierar` *Produce GO descriptors using hierarchical clustering algorithm*

Description

This function is used to estimate GO descriptors for omics data, based on GO ontology. Data are clustered based on hierarchical clustering algorithm from vegan R package using default values. The user needs to specify the number of clusters, the distance matrix method, and the hierarchical clust method.

Usage

```
generate.descr.hierar(dataset, rawModel, additionalInfo)
```

Arguments

`dataset`	Data for prediction. A list of two objects: datasetURI (a character string), dataEntry (a data frame).
`rawModel`	A serialized list of parameters for hierarchical clustering, as produced by generate.param.model
`additionalInfo`	Any additional information needed for rawModel. Here an empty list.

Details

No details required

Value

A data frame giving the new GO descriptors given the function's parameters. Number of columns are the number of new GO descriptors, number of rows is the number of features used.

Note

No notes for this function

Author(s)

Georgia Tsiliki

References

The help file of skeleton

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

data("dat1p")
data("dat1m2")

adInfo<- list()

pred.res<- generate.descr.hierar(dat1p, dat1m2, adInfo)
```

`generate.param.model` *A 'training' function to produce a serialized list for the parameters needed to produce GO descriptors*

Description

This function is used to pass on all the neccessary information to generate.descr.biclust and generate.descr.hierar functions.

Usage

```
generate.param.model(dataset, predictionFeature, parameters)
```

Arguments

`dataset`	list of 2 objects, datasetURI:= character sring, code name of dataset, dataEntry:= data frame with 2 columns
`predictionFeature`	character string specifying which is the prediction feature in dataEntry, here empty
`parameters`	list with parameter values for ontology and biclustering. 5 or 7 objects should be included depending on whether we then intend to use generate.descr.biclust or generate.descr.hierar function, respectively. In the first case: 'key'a character sring for gene/protein/etc names id (for dat1 'UNIPROT' is the right key), 'onto' a character vector showing the ontology and sub.ontologies used (c('GO','MF')), 'pvalCutoff' a numeric value for hypergeometric p-values cutoff (e.g. 0.05), 'nclust' a numeric vector indicating the number of clusters for GOs(x axis)

and genes/proteins (y axis) (e.g. c(5,4)), 'FUN' a string, R function to summarize vector's groups (e.g. mean). In the second case: 'key'a character sring for gene/protein/etc names id (for dat1 'UNIPROT' is the right key), 'onto' a character vector showing the ontology and sub.ontologies used (c('GO','MF')), 'pvalCutoff' a numeric value for hypergeometric p-values cutoff (e.g. 0.05), 'distMethod' distance method (could be one of those provided via vegan R package), 'hclustMethod' (could be one of those provided via vegan R package) , 'nORh' either a numeric value or character giving number of clusters or a function to define height respectively, 'FUN' a string, R function to summarize vector's groups (e.g. mean).

Details

No details required

Value

`rawModel` A serialized object of the parameters list supplied.

`pmmlModel` A pmml GO descriptors object - now empty

`independentFeatures`
A list with Ambit names for all genes/ proteins features included in the model

#itempredictedFeaturesA character vector with dummy names for the GO descriptors that will be produced by functions generate.descr.biclust or generate.descr.hierar

`additionalInfo` A data frame with all independent features included in the model and their dummy name in the model - here empty

Note

No notes for this function

Author(s)

Georgia Tsiliki

References

The help file of skeleton

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

data("dat1")

predF<- list()

required.param<- list(key='UNIPROT',onto=c('GO','MF'),pvalCutoff=0.05,nclust=c(3,2),FUN='mean')
```

```
params1<- generate.param.model(dat1,predF,required.param)
```

`read.in.json.for.pred` *Read in function for json files for prediction*

Description

This function reads in a json data file and produces a list with independent features, parameters list for GO clustering saved as raw model

Usage

```
read.in.json.for.pred(dataset, rawModel, additionalInfo)
```

Arguments

`dataset`	Data for prediction. A list of two objects: datasetURI (a character string), dataEntry (a data frame) which should include an omics data set (e.g. proteomics/genomics/etc).
`rawModel`	Raw model for prediction. Here a seerilized list of parameters to be used by genrate.descr.biclust or generate.descr.hierar functions.
`additionalInfo`	Any additional information needed for rawModel. Here an empty list.

Details

No further details required

Value

A List including:

`x.mat`	data frame with independent variables values (proteins/genes/ etc)
`model`	R model (a list of parameters to be used by genrate.descr.biclust or generate.descr.hierar functions)
`additionalInfo`	Any additional information needed for rawModel. Here a data frame giving the Ambit names of the independent features included in x.mat.

Note

No notes for this function

Author(s)

Georgia Tsiliki

References

The help file of skeleton

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

data("dat1p")
data("dat1m1")

adInfo<- list()

res1<- read.in.json.for.pred(dat1p, dat1m1, adInfo)
```

Index