Package 'GOCompare'

November 8, 2022

Title Comprehensive GO Terms Comparison Between Species

Version 1.0.2

Description Supports the assessment of functional enrichment analyses obtained for several lists of genes and provides a workflow to analyze them between two species via weighted graphs. Methods are described in Sosa et al. (2022) (Submitted to Genomics).

URL https://github.com/ccsosa/GOCompare

BugReports https://github.com/ccsosa/GOCompare/issues

Depends R (>= 4.0.0)

Imports base (>= 3.5),
utils (>= 3.5),
methods (>= 3.5),
stats,
grDevices,
ape,
vegan,
ggplot2,
ggrepel,
igraph,
parallel,
stringr,
mathjaxr,

RdMacros mathjaxr

License GPL (>= 3)

LazyData true

Encoding UTF-8

RoxygenNote 7.2.1

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

R topics documented:

`GOCompare-package` *GOCompare: An R package to compare GO terms of gene lists (categories) and their orthologs*

Description

GOCompare is a an R package used to compare a GO terms list between two species

Details

Package:	GOCompare
Type:	Package
Version:	1.0.2
Date:	2022-11-07
License:	GPL-3

`A_thaliana` *A thaliana functional enrichment analysis of 2224 ortholog genes related to cancer-hallmarks*

Description

This dataset is the original dataset obtained for Clavijo-Buriticá (In preparation)

Usage

```
A_thaliana
```

Format

A data frame with 4063 rows and 6 variables:

Enrichment_FDR Numeric: False discovery rate values for the GO term

Genes_in_list numeric: Number of genes in the list of genes for a given GO term

Total_genes numeric: Number of genes in the genome of a species for a given GO term

Functional_Category character: GO term name or GO term id

Genes character: Genes found fot a given GO term

feature character: A column representing the belonging of a group of comparison

Source

https://data.mendeley.com/datasets/myyy2wxd59/1

References

Clavijo-Buriticá, Sosa, C.C., Mosquera, A.J. Álvarez, A., Medina, J. Quimbaya, M.A. A systematic comparison of the molecular machinery associated with Cancer-Hallmarks between plants and humans reveals Arabidopsis thaliana as a useful model to understand specific carcinogenic events (to be submitted, Target journal: Plos Biology)

A_thaliana_compress *A thaliana functional enrichment analysis results for "AID","DCE","RCD","SPS" cancer-hallmarks*

Description

This dataset is a subset of the original dataset obtained for Clavijo-Buriticá (In preparation)

Usage

```
A_thaliana_compress
```

Format

A data frame with 120 rows and 6 variables (30 GO terms per cancer hallmark):

Enrichment_FDR Numeric: False discovery rate values for the GO term

Genes_in_list numeric: Number of genes in the list of genes for a given GO term

Total_genes numeric: Number of genes in the genome of a species for a given GO term

Functional_Category character: GO term name or GO term id

Genes character: Genes found fot a given GO term

feature character: A column representing the belonging of a group of comparison

Source

https://data.mendeley.com/datasets/myyy2wxd59/1

References

Clavijo-Buriticá, Sosa, C.C., Mosquera, A.J. Álvarez, A., Medina, J. Quimbaya, M.A. A systematic comparison of the molecular machinery associated with Cancer-Hallmarks between plants and humans reveals Arabidopsis thaliana as a useful model to understand specific carcinogenic events (to be submitted, Target journal: Plos Biology)

`compareGOspecies` *Visual representation for the results of functional enrichment analysis to compare two species and a series of categories*

Description

compareGOspecies function provides a simple workflow to compare results of functional enrichment analysis for two species.

To use this function you will need two matrices with a column which, represents the features to be compared (e.g.feature). This function will extract the unique GO terms for two matrices and it will generate a presence-absence matrix where rows will represent a combination of categories and species (e.g H.sapiens AID) and columns will represent the GO terms analyzed. Further, this function will calculate Jaccard distances and it will provide as outputs a list with four slots: 1.) A principal coordinates analysis (PCoA) 2.) The Jaccard distance matrix 3.) A list of shared GO terms between species 4.) Finally, a list of the unique GO terms and the belonging to the respective species.

Usage

```
compareGOspecies(
  df1,
  df2,
  GOterm_field,
  species1,
  species2,
  paired_lists = TRUE
)
```

Arguments

`df1`	A data frame with the results of a functional enrichment analysis for the species 1 with an extra column "feature" with the features to be compared
`df2`	A data frame with the results of a functional enrichment analysis for the species 2 with an extra column "feature" with the features to be compared
`GOterm_field`	This is a string with the column name of the GO terms (e.g; "Functional_Category")
`species1`	This is a string with the species name for species 1 (e.g; "H. sapiens")
`species2`	This is a string with the species name for species 2 (e.g; "A. thaliana")
`paired_lists`	This is a boolean to indicate if both species have same comparable categories (gene lists). If the paired_lists is FALSE the counts will be done only for species and categories will be kept in the outcomes. Please use carefully when paired_lists = FALSE.

Value

This function will return a list with four slots: graphics, distance shared_GO_list, and unique_GO_list

Examples

```
#Loading example datasets
data(H_sapiens_compress)
data(A_thaliana_compress)
#Defining the column with the GO terms to be compared
GOterm_field <- "Functional_Category"
#Defining the species names
species1 <- "H. sapiens"
species2 <- "A. thaliana"

#Running function
x <- compareGOspecies(df1=H_sapiens_compress,
                      df2=A_thaliana_compress,
                      GOterm_field=GOterm_field,
                      species1=species1,
                      species2=species2,
                      paired_lists=TRUE)

## Not run:
#Displaying PCoA results
 x$graphics
# Checking shared GO terms between species
 print(tapply(x$shared_GO_list$feature,x$shared_GO_list$feature,length))

## End(Not run)
```

---

comparison_ex_compress

Functional enrichment analysis comparison between H. sapiens and A. thaliana for "AID","DCE","RCD","SPS" cancer-hallmarks

---

Description

This dataset is the results of running the compareGOspecies species and it is composed of four slots:

graphics PCoA graphics

distance numeric: Jaccard distance matrix

shared_GO_list data.frame with shared GO terms between species

unique_GO_list data.frame with unique GO terms and their belonging two each species

Usage

```
comparison_ex_compress
```

Format

An object of class `list` of length 4.

Source

https://data.mendeley.com/datasets/myyy2wxd59/1

References

Clavijo-Buriticá, Sosa, C.C., Mosquera, A.J. Álvarez, A., Medina, J. Quimbaya, M.A. A systematic comparison of the molecular machinery associated with Cancer-Hallmarks between plants and humans reveals Arabidopsis thaliana as a useful model to understand specific carcinogenic events (to be submitted, Target journal: Plos Biology)

`comparison_ex_compress_CH`

Functional enrichment analysis comparison between H. sapiens and A. thaliana for "DCE", and "RCD" cancer-hallmarks. This dataset contains 10 GO terms per category to allow a fast run of the function graph_two_GOspecies.

Description

This dataset is the results of running the compareGOspecies species and it is composed of three slots:

distance numeric: Jaccard distance matrix

shared_GO_list data.frame with shared GO terms between species

unique_GO_list data.frame with unique GO terms and their belonging two each species

Usage

```
comparison_ex_compress_CH
```

Format

An object of class `list` of length 3.

Source

https://data.mendeley.com/datasets/myyy2wxd59/1

References

Clavijo-Buriticá, Sosa, C.C., Mosquera, A.J. Álvarez, A., Medina, J. Quimbaya, M.A. A systematic comparison of the molecular machinery associated with Cancer-Hallmarks between plants and humans reveals Arabidopsis thaliana as a useful model to understand specific carcinogenic events (to be submitted, Target journal: Plos Biology)

`evaluateCAT_species` *Comprehensive comparison between species using categories and Pearson's Chi-squared Tests*

Description

evaluateGO_species provides a simple function to compare results of functional enrichment analysis for two species through the use of proportion tests or Pearson's Chi-squared Tests and a False discovery rate correction

Usage

```
evaluateCAT_species(df1, df2, species1, species2, GOterm_field, test = "prop")
```

Arguments

`df1`	A data frame with the results of a functional enrichment analysis for the species 1 with an extra column "feature" with the features to be compared
`df2`	A data frame with the results of a functional enrichment analysis for the species 2 with an extra column "feature" with the features to be compared
`species1`	This is a string with the species name for the species 1 (e.g; "H. sapiens")
`species2`	This is a string with the species name for the species 2 (e.g; "A. thaliana")
`GOterm_field`	This is a string with the column name of the GO terms (e.g; "Functional_Category")
`test`	This is a string with the hypothesis test to be performed. Two options are provided, "prop" and "chi-squared" (default value="prop")

Value

This function will return a data.frame with the following fields:

CAT	Category
pvalue	p-value obtained through the use of Pearson's Chi-squared Test
FDR	Multiple comparison correction for the p-value column

Examples

```
#Loading example datasets
data(H_sapiens)
data(A_thaliana)
#Defining the column with the GO terms to be compared
GOterm_field <- "Functional_Category"
#Defining the species names
species1 <- "H. sapiens"
species2 <- "A. thaliana"
#Running function
x <- evaluateCAT_species(df1= H_sapiens,
                      df2=A_thaliana,
                      species1=species1,
                      species2=species2,
                      GOterm_field=GOterm_field,
                        test="prop")
print(x)
```

`evaluateGO_species` *Comprehensive comparison between species using GO terms and Pearson's Chi-squared Tests*

Description

evaluateGO_species provides a simple function to compare results of functional enrichment analysis for two species through the use of proportion tests or Pearson's Chi-squared Tests and a False discovery rate correction

Usage

```
evaluateGO_species(df1, df2, species1, species2, GOterm_field, test = "prop")
```

Arguments

`df1`	A data frame with the results of a functional enrichment analysis for the species 1 with an extra column "feature" with the features to be compared
`df2`	A data frame with the results of a functional enrichment analysis for the species 2 with an extra column "feature" with the features to be compared
`species1`	This is a string with the species name for the species 1 (e.g; "H. sapiens")
`species2`	This is a string with the species name for the species 2 (e.g; "A. thaliana")
`GOterm_field`	This is a string with the column name of the GO terms (e.g; "Functional_Category")
`test`	This is a string with the hypothesis test to be performed. Two options are provided, "prop" and "chi-squared" (default value="prop")

Value

This function will return a data.frame with the following fields:

GO	GO term analyzed
pvalue	p-value obtained through the use of Pearson's Chi-squared Test
FDR	Multiple comparison correction for the p-value column

Examples

```
#Loading example datasets
data(H_sapiens)
data(A_thaliana)
#Defining the column with the GO terms to be compared
GOterm_field <- "Functional_Category"
#Defining the species names
species1 <- "H. sapiens"
species2 <- "A. thaliana"
#Running function
x <- evaluateGO_species(df1= H_sapiens,
                        df2=A_thaliana,
                        species1=species1,
                        species2=species2,
                        GOterm_field=GOterm_field,
                        test="prop")
print(x)
```

graphGOspecies *Undirected network representation for the results of functional enrichment analysis for one species*

Description

graphGOspecies is a function to create undirected graphs using two options:

Categories option:

The nodes (V) represent groups of gene lists (categories), and the edges (E) represent GO terms co-occurring between pairs of categories. More specifically, Two categories: $u, v \epsilon V$ are connected by an edge $e = (u, v)$.the edge weights $w(e)$ are defined as the ratio of the number of GO terms co-occurring between two categories. Edge weights w(e) are defined as the ratio of the number of GO terms (e.g. biological processes) co-occurring between two categories BP_u nBP_v compared to the total number of GO terms available. A node weight $K_w(u)$ is defined as the sum of the edge weights where the node u is a participant. Thus, the node weight represents how frequently GO terms are reported and expressed in a biological phenomenon.

$$w(e) = \frac{\mid BP_u n BP_v \mid}{\mid BP \mid}$$

(1)

$$K_w = \sum_{v \epsilon V} w(u, v)$$

(2)

GO option:

The nodes V represent GO terms and the edges E' represent categories where a pair of GO terms co-occur. More specifically, two GO terms are connected by an edge $e' = (u, v')$. the edge weight $w'(e')$ corresponds to the number of categories co-occurring the GO terms u and v',compared with the total number of GO terms (Equation 3). A node weight $K'_w(u')$ is defined,in this case the weight represents the importance of a GO term (more frequent co-occurring).(Please be patient, it requires a long time to finish).

$$w'(e') = \frac{\mid Cu' \cap Cv' \mid}{\mid BP \mid}$$

(3)

$$K'_w(u') = \sum_{v' \epsilon V'} w'(u', v')$$

(4)

Usage

```
graphGOspecies(
  df,
  GOterm_field,
  option = "Categories",
```

```
  numCores = 2,
  saveGraph = FALSE,
  outdir = NULL,
  filename = NULL
)
```

Arguments

df	A data frame with the results of a functional enrichment analysis for a species with an extra column "feature" with the features to be compared
GOterm_field	This is a string with the column name of the GO terms (e.g: "Functional.Category")
option	(values: "GO" or "Categories"). This option allows create either a graph where nodes are GO terms and edges are features or alternatively a graph where nodes are features and edges are GO terms (default value="Categories")
numCores	numeric, Number of cores to use for the process (default value numCores=2). For the example below, only one core will be used
saveGraph	logical, if TRUE the function will allow save the graph in graphml format
outdir	This parameter will allow save the graph file in a folder described here (e.g: "D:").This parameter only works when saveGraph=TRUE
filename	The name of the graph filename to be saved in the outdir detailed by the user.This parameter only works when saveGraph=TRUE

Value

This function will return a list with two slots: edges and nodes.

(Categories): Edges list columns:

Column	Description
SOURCE and TARGET	The source and target categories (Nodes in the edge)
FEATURES_N	The number of GO terms between the categories
WEIGHT	Edge weight
FEATURES	GO terms available for both nodes

Node list columns:

Column	Description
feature	Category name
GO_count	GO terms counts for the node
WEIGHT	Node weight

(GO):

Edges list columns:

Column	Description
SOURCE and TARGET	The source and target GO terms (Nodes in the edge)
FEATURE	The number of Categories where both GO Terms were found
WEIGHT	Edge weight

Node list columns:

Column	Description
GO	GO term node name
GO_WEIGHT	Node weight

Examples

```
#Loading example datasets
data(H_sapiens_compress)

GOterm_field <- "Functional_Category"

#Running function
x <- graphGOspecies(df=H_sapiens_compress,
                    GOterm_field=GOterm_field,
                    option = "Categories",
                    numCores=1,
                    saveGraph=FALSE,
                    outdir = NULL,
                    filename=NULL)
```

`graph_two_GOspecies` *Undirected network representation for the results of functional enrichment analysis to compare two species and a series of categories*

Description

graph_two_GOspecies is a function to create undirected graphs

The graph_two_GOspecies is an analog of the graphGOspecies function, and it has the same options (" Categories " and " GO "). Nevertheless, the way in which the edge and node weights are calculated is slightly different. Since two species are compared, three possible graphs are available G_1, G_2, and G_3. G_1, and G_2 represent each of the species analyzed and G_3 is a subgraph of G_1, G_2, which contains the GO terms or Categories co-ocurring between both species.

Categories option: (Weight): The nodes (V) represent groups of gene lists (categories), and the edges (E) represent GO terms co-occurring between pairs of categories and the weight of the nodes provides a measure of how a GO term is conserved between two species and a series of categories but it is biased to categories.

$$\widehat{K}_w(u) = \sum_{v \epsilon V_1} w(u,v) + \sum_{v \epsilon V_2} w(u,v)$$

(5)

(shared weight): The nodes (V) represent groups of gene lists (categories), and the edges (E) represent GO terms co-occurring between pairs of categories that are only shared between species. This node weight K_s is computed from a shared weight of edges s, where $N1$ and $N2$ are the set of GO terms associated with the edge $e = (u, v)$ for species 1 and 2, respectively. Therefore the node shared weight $K_s(u)$ is the sum of s.

$$s(e) = \frac{\mid N1\ n\ N2 \mid}{\mid N1 \bigcup N2 \mid}$$

(6)

$$K_s(u) = \sum_{v \epsilon (V_1 \bigcup V_2)} s(u, v)$$

(7)

(combined weight): This node weight $K_c(u)$ is a combination of the weight and the shared weight. The idea of this combined weight is to find categories with more frequent GO terms co-ocurring in order to observe functional similarities between two species with a balance of GO terms co-occurring among gene lists (categories) and the two species. This node weight varies from -1 (categories with GO terms found only in one species and few categories) to 1 (categories with GO terms shared widely between species and among other categories). the combined node weight K_c is defined as the sum of the min-max normalized weights $\widehat{K}_w$ and K_s minus 1.

$$minmax(y) = \frac{y - min(y)}{max(y) - min(y)}$$

(8)

$$K_c(u) = minmax(\widehat{K}_w(u)) + minmax(K_s(u)) - 1$$

(9)

GO option: Given there are three possible graphs are available G_1, G_2, and G_3. G_1, and G_2 represent each of the species analyzed and G_3 is a subgraph of G_1, G_2, which contains the GO terms or Categories co-ocurring between both species. For this case, Nodes are GO terms and edges are categories where a GO terms is co-ocurring. This weight is similar to the GO weight calculated for graphGOspecies function. it is calculated as the equation 5.

$$\widehat{K}_w(u) = \sum_{v \epsilon V_1} w(u, v) + \sum_{v \epsilon V_2} w(u, v)$$

(5)

Usage

```
graph_two_GOspecies(
  x,
  species1,
  species2,
  GOterm_field,
  saveGraph = FALSE,
  option = "Categories",
  numCores = 2,
  outdir = NULL,
  filename = NULL
)
```

Arguments

x	is a list obtained as output of the comparegOspecies function
species1	This is a string with the species name for species 1 (e.g; "H. sapiens")
species2	This is a string with the species name for species 2 (e.g; "A. thaliana")
GOterm_field	This is a string with the column name of the GO terms (e.g; "Functional_Category")
saveGraph	logical, if TRUE the function will allow save the graph in graphml format
option	(values: "Categories or "GO"). This option allows create either a graph where nodes are GO terms and edges are features and GO as well as species belonging are edges attributes or a graph where nodes are GO terms and edges are species belonging (default value="Categories")
numCores	numeric, Number of cores to use for the process (default value numCores=2). For the example below, only one core will be used
outdir	This parameter will allow save the graph file in a folder described here (e.g: "D:").This parameter only works when saveGraph=TRUE
filename	The name of the graph filename to be saved in the outdir detailed by the user.This parameter only works when saveGraph=TRUE

Value

This function will return a list with two slots: edges and nodes. (Categories): Edges list columns:

Column	Description
SOURCE and TARGET	The source and target categories (Nodes in the edge)
GO_N	The number of GO terms between the categories
WEIGHT	Edge weight
GO	GO terms available for both nodes
SP1	Number of GO terms for the species 1
SP2	Number of GO terms for the species 2
SHARED	Number of GO terms shared or co-ocurring between the categories
SHARED_WEIGHT	Shared weight for the edge

Node list columns:

Column	Description
CAT	Category name
CAT_WEIGHT	Node weight
SHARED_WEIGHT	Shared weight for the node
COMBINED_WEIGHT	Combined weight for the node

(GO):

Edges list columns:

Column	Description
SOURCE and TARGET	The source and target GO terms (Nodes in the edge)
FEATURE	The number of Categories where both GO Terms were found
SP	Species where the GO terms was found (Species 1, Species 2 or Shared)

WEIGHT	Edge weight

Node list columns:

Column	Description
GO	GO term node name
GO_WEIGHT	Node weight

Examples

```
GOterm_field <- "Functional_Category"
data(comparison_ex_compress_CH)
#Defining the species names
species1 <- "H. sapiens"
species2 <- "A. thaliana"
x_graph <- graph_two_GOspecies(x=comparison_ex_compress_CH,
          species1=species1,
          species2=species2,
          GOterm_field=GOterm_field,
          numCores=1,
          saveGraph = FALSE,
          option= "Categories",
          outdir = NULL,
          filename= NULL)
```

`H_sapiens` *H. sapiens functional enrichment analysis of 5494 genes related to cancer-hallmarks*

Description

This dataset is a subset of the original dataset obtained for Clavijo-Buriticá (In preparation)

Usage

```
H_sapiens
```

Format

A data frame with 5000 rows and 6 variables:

Enrichment_FDR Numeric: False discovery rate values for the GO term

Genes_in_list numeric: Number of genes in the list of genes for a given GO term

Total_genes numeric: Number of genes in the genome of a species for a given GO term

Functional_Category character: GO term name or GO term id

Genes character: Genes found fot a given GO term

feature character: A column representing the belonging of a group of comparison

Source

https://data.mendeley.com/datasets/myyy2wxd59/1

References

Clavijo-Buriticá, Sosa, C.C., Mosquera, A.J. Álvarez, A., Medina, J. Quimbaya, M.A. A systematic comparison of the molecular machinery associated with Cancer-Hallmarks between plants and humans reveals Arabidopsis thaliana as a useful model to understand specific carcinogenic events (to be submitted, Target journal: Plos Biology)

`H_sapiens_compress` *H. sapiens functional enrichment analysis results for "AID","DCE","RCD","SPS" cancer-hallmarks*

Description

This dataset is a subset of the original dataset obtained for Clavijo-Buriticá (In preparation)

Usage

```
H_sapiens_compress
```

Format

A data frame with 120 rows and 6 variables (30 GO terms per cancer hallmark):

Enrichment_FDR Numeric: False discovery rate values for the GO term

Genes_in_list numeric: Number of genes in the list of genes for a given GO term

Total_genes numeric: Number of genes in the genome of a species for a given GO term

Functional_Category character: GO term name or GO term id

Genes character: Genes found fot a given GO term

feature character: A column representing the belonging of a group of comparison

Source

https://data.mendeley.com/datasets/myyy2wxd59/1

References

Clavijo-Buriticá, Sosa, C.C., Mosquera, A.J. Álvarez, A., Medina, J. Quimbaya, M.A. A systematic comparison of the molecular machinery associated with Cancer-Hallmarks between plants and humans reveals Arabidopsis thaliana as a useful model to understand specific carcinogenic events (to be submitted, Target journal: Plos Biology)

mostFrequentGOs *Most frequent GO terms among groups for a data.frame*

Description

Provides an easy way to get the frequency of GO terms such as biological processes for a data frame and a series of features

Usage

```
mostFrequentGOs(df, GOterm_field)
```

Arguments

df	A data frame with the results of a functional enrichment analysis for a species with an extra column "feature" with the features to be compared
GOterm_field	This is a string with the column name of the GO terms (e.g; "Functional.Category")

Value

This function will return a table with the frequency of GO terms per feature

Examples

```
#Loading example datasets
data(H_sapiens)
#Defining the column with the GO terms to be compared
GOterm_field <- "Functional_Category"
#Running function
x <- mostFrequentGOs(df=H_sapiens, GOterm_field=GOterm_field)
#Displaying results
head(x)
```

Index