Package ‘BioCor’

April 25, 2017

Title   Functional similarities
Version 1.0.0
Author  Lluís Revilla Sancho <lluis.revilla@gmail.com>
Maintainer Lluís Revilla Sancho <lluis.revilla@gmail.com>
Description Calculates functional similarities based on the pathways described on KEGG and REACTOME or in gene sets. These similarities can be calculated for pathways or gene sets, genes, or clusters and combined with other similarities. They can be used to improve networks, gene selection, testing relationships...
Depends R (>= 3.4.0)
License GPL-3 | file LICENSE
Encoding UTF-8
LazyData true
biocViews Software, StatisticalMethod, Clustering, GeneExpression, Reactome, Network, KEGG, Pathways
Imports org.Hs.eg.db, AnnotationDbi, reactome.db, graph, methods, utils
Suggests WGCNA, testthat, knitr, rmarkdown, BiocStyle, GOSemSim, GSEABase
BugReports https://github.com/llrs/BioCor/issues
URL https://github.com/llrs/BioCor/
VignetteBuilder knitr
RoxygenNote 6.0.1
NeedsCompilation no

R topics documented:

BioCor-package .................................................. 2
addSimilarities .............................................. 2
AintoB ......................................................... 3
clusterGeneSim ............................................. 4
clusterSim ................................................... 5
combinadic .................................................. 6
BioCor-package

BioCor: A package to calculate functional similarities

Description

Calculates a functional similarity measure between gene identifiers based on the pathways described on KEGG and REACTOME.

Important functions

- **pathSim**: Calculates the similarity between two pathways
- **geneSim**: Calculates the similarity (based on pathSim) between two genes
- **clusterSim**: Calculates the similarity between two clusters of genes by joining pathways of each gene.
- **clusterGeneSim**: Calculates the similarity between two clusters of genes by comparing the similarity between the genes of a cluster
- **similarities**: Allows to combine the value of matrices of similarities
- **conversions**: Two functions to convert similarity measures
- **weighted**: Functions provided to combine similarities

addSimilarities

Additive integration of similarities

Description

Function that use the previously calculated similarities into a single similarity matrix.

Usage

addSimilarities(x, bio_mat, weights = c(0.5, 0.18, 0.1, 0.22))

Arguments

- **x**: A matrix with the similarity of expression
- **bio_mat**: A list of matrices of the same dimension as x.
- **weights**: A numeric vector of weight to multiply each similarity
AintoB

Details
The total weight can’t be higher than 1 to prevent values above 1 but can be below 1. It uses weighted.sum with abs = TRUE internally.

Value
A square matrix of the same dimensions as the input matrices.

Author(s)
Lluís Revilla

See Also
similarities, weighted.

Examples
set.seed(100)
a <- seq2mat(LETTERS[1:5], rnorm(10))
b <- seq2mat(LETTERS[1:5], seq(from = 0.1, to = 1, by = 0.1))
sim <- list(b)
addSimilarities(a, sim, c(0.5, 0.5))

AintoB

Description
Insert values from a matrix into another matrix bade on the rownames and colnames replacing the values.

Usage
AintoB(A, B)

Arguments
A
A matrix to be inserted.

B
A matrix to insert in.

Details
If all the genes with pathway information are already calculated but you would like to use more genes when performing analysis. insert the once you have calculated on the matrix of genes.

Value
A matrix with the values of A in the matrix B

Author(s)
Lluís Revilla
Examples

```r
B <- matrix(ncol = 10, nrow = 10,
           dimnames = list(letters[1:10], letters[1:10]))
A <- matrix(c(1:15), byrow = TRUE, nrow = 5,
           dimnames = list(letters[1:5], letters[1:3]))
AintoB(A, B)

# Mixed orders
colnames(A) <- c("c", "h", "e")
rownames(A) <- c("b", "a", "f", "c", "j")
AintoB(A, B)

# Missing columns or rows
colnames(A) <- c("d", "f", "k")
AintoB(A, B)
```

---

**clusterGeneSim**

*Similarity score between clusters of genes based on genes similarity*

**Description**

Looks for the similarity between genes of a group and then between each group.

**Usage**

```r
clusterGeneSim(cluster1, cluster2, info, method = c("max", "rcmax.avg"), ...)
mclusterGeneSim(clusters, info, method = c("max", "rcmax.avg"), ...)
```

**Arguments**

- **cluster1**: A vector with genes.
- **cluster2**: A vector with genes.
- **info**: A list of genes and the pathways they are involved.
- **method**: A vector with two or one argument to be passed to combineScores the first one is used to summarize the similarities of genes, the second one for clusters.
- **...**: Other arguments passed to `combineScores`
- **clusters**: A list of clusters of genes to be found in `id`.

**Details**

Differs with `clusterGeneSim` that first each combination between genes is calculated, and with this values then the comparison between the two clusters is done. Thus applying `combineScores` twice, one at gene level and another one at cluster level.

**Value**

- `clusterGeneSim` returns a similarity score of the two clusters or the similarity between the genes of the two clusters.
- `mclusterGeneSim` returns a matrix with the similarity scores for each cluster comparison.
clusterSim

Similarity score between clusters of genes based on pathways similarity

Description

Looks for the similarity between genes in groups

Usage

clusterSim(cluster1, cluster2, info, method = "max", ...)

mclusterSim(clusters, info, method = "max", ...)

Arguments

cluster1, cluster2
A vector with genes.

info
A list of genes and the pathways they are involved.

method
To combine the scores of each pathway, one of c("avg", "max", "rcmax", "rcmax.avg", "BMA"), if NULL returns the matrix of similarities.

...
Other arguments passed to combineScores

clusters
A list of clusters of genes to be found in id.
Details
Once the pathways for each cluster are found they are combined using combineScores.

Value
clusterSim returns a similarity score of the two clusters
mclusterSim returns a matrix with the similarity scores for each cluster comparison.

Author(s)
Lluís Revilla

See Also
For a different approach see clusterGeneSim, combineScores and conversions

Examples

```r
library("org.Hs.eg.db")
# Extract the paths of all genes of org.Hs.eg.db from KEGG (last update in # data of June 31st 2011)
genes.kegg <- as.list(org.Hs.egPATH)
clusterSim(c("9", "15", "10"), c("33", "19", "20"), genes.kegg)
clusterSim(c("9", "15", "10"), c("33", "19", "20"), genes.kegg, NULL)
clusterSim(c("9", "15", "10"), c("33", "19", "20"), genes.kegg, "avg")

clusters <- list(cluster1 = c("18", "01", "10"),
                 cluster2 = c("100", "10", "1"),
                 cluster3 = c("18", "10", "83"))
mclusterSim(clusters, genes.kegg)
mclusterSim(clusters, genes.kegg, "avg")
```

combinadic

\textit{i-th combination of} \textit{n elements taken from} \textit{r}

Description
Function similar to combn but for larger vectors. To avoid allocating a big vector with all the combinations each one can be computed with this function.

Usage
combinadic(n, r, i)

Arguments

\begin{itemize}
\item \texttt{n} \hspace{1em} Elements to extract the combination from
\item \texttt{r} \hspace{1em} Number of elements per combination
\item \texttt{i} \hspace{1em} ith combination
\end{itemize}
**Value**

The combination $i$th of the elements

**Author(s)**

Joshua Ulrich

**References**

StackOverflow answer 4494469/2886003

**See Also**

`combn`

**Examples**

```r
# Output of all combinations
combn(LETTERS[1:5], 2)

# Output of the second combination
combinadic(LETTERS[1:5], 2, 2)
```

---

**Description**

Combine several values into one by several methods.

**Usage**

```r
combineScores(scores, method, round = FALSE)
```

**Arguments**

- `scores`: Matrix of scores to be combined
- `method`: one of c("avg", "max", "rcmax", "rcmax.avg", "BMA") see details
- `round`: Should the resulting value be rounded to the third digit?

**Details**

The methods return:

- `avg`: The average or mean value
- `max`: The max value
- `rcmax`: The max of the column means or row means
- `rcmax.avg`: The sum of the max values by rows and columns divided by the number of columns and rows
- `BMA`: The same as `rcmax.avg`
Value

A numeric value as described in details.

Note

This is a version of combineScores from `combineScores` with optional rounding and some internal differences.

Author(s)

Lluís Revilla based on Guangchuang Yu

Examples

d <- structure(c(0.4, 0.6, 0.222222222222222, 0.4, 0.4, 0, 0.25, 0.5, 0.285714285714286), .Dim = c(3L, 3L), .Dimnames = list(c("a", "b", "c"), c("d", "e", "f")))
dsapply(c("avg", "max", "rcmax", "rcmax.avg", "BMA"), combineScores, scores = d)
d[1,2] <- NA
sapply(c("avg", "max", "rcmax", "rcmax.avg", "BMA"), combineScores, scores = d)

conversions

Convert the similarities formats

Description

Functions to convert the similarity coefficients between Jaccard and Dice. D2J is the opposite of J2D.

Usage

D2J(D)
J2D(J)

Arguments

D Dice coefficient, as returned by `diceSim`, `geneSim`, `clusterSim` and `clusterGeneSim`
J Jaccard coefficient

Value

A numeric value.

Author(s)

Lluís Revilla
**diceSim**

**Examples**

D2J(0.5)  
J2D(0.5)  
D2J(J2D(0.5))

---

**diceSim**  
*Compare pathways*

**Description**

Function to estimate how much two graphs or list of genes overlap by looking how much of the nodes are shared.

**Usage**

```r
diceSim(g1, g2)
```

**Arguments**

- `g1`, `g2`  
  Graph in GraphNEL format, or a character list with the names of the proteins in each pathway.

**Value**

A score between 0 and 1 calculated as the double of the proteins shared by `g1` and `g2` divided by the number of genes in both groups.

**Author(s)**

Lluís Revilla

**See Also**

Used for `geneSim`, see `conversions` help page to transform Dice score to Jaccard score.

**Examples**

```r
genes.id2 <- c("52", "11342", "80895", "57654", "548953", "11586", "45985")
genes.id1 <- c("52", "11342", "80895", "57654", "58493", "1164", "1163",  
"4150", "2130", "159")
diceSim(genes.id1, genes.id2)  
diceSim(genes.id2, genes.id2)
```
duplicateIndices

Finds the indices of the duplicated events of a vector

Description
Finds the indices of duplicated elements in the vector given.

Usage
duplicateIndices(vec)

Arguments
vec Vector of identifiers presumably duplicated

Details
For each duplication it can return a list or if all the duplication events are of the same length it returns a matrix, where each column is duplicated.

Value
The format is determined by the simplify2array

Author(s)
Lluís Revilla

See Also
removeDup

Examples
duplicateIndices(c("52", "52", "53", "55")) # One repeated element
duplicateIndices(c("52", "52", "53", "55", "55")) # Repeated elements
duplicateIndices(c("52", "55", "53", "55", "52")) # Mixed repeated elements

geneSim

Similarity score genes based on pathways similarity

Description
Given two genes, calculates the Dice similarity between each pathway which is combined to obtain a similarity between the genes.

Usage
geneSim(gene1, gene2, info, method = "max", ...)
mgeneSim(genes, info, method = "max", ...)

**geneSim**

**Arguments**

- `gene1, gene2`: Ids of the genes to calculate the similarity, to be found in genes.
- `info`: A list of genes and the pathways they are involved.
- `method`: To combine the scores of each pathway, one of c("avg", "max", "rcmax", "rcmax.avg", "BMA"), if NULL returns the matrix of similarities.
- `...`: Other arguments passed to `combineScores`.
- `genes`: A vector of genes.

**Details**

Given the information about the genes and their pathways, uses the ids of the genes to find the Dice similarity score for each pathway comparison between the genes. Later this similarities are combined using `combineScores`.

**Value**

The highest Dice score of all the combinations of pathways between the two ids compared if a method to combine scores is provided or NA if there isn’t information for one gene. If an NA is returned this means that there isn’t information available for any pathways for one of the genes. Otherwise a number between 0 and 1 (both included) is returned. Note that there isn’t a negative value of similarity.

`mgeneSim` returns the matrix of similarities between the genes in the vector.

**Author(s)**

Lluis Revilla

**See Also**

- `conversions` help page to transform Dice score to Jaccard score. For the method to combine the scores see `combineScores`.

**Examples**

```r
library("org.Hs.eg.db")
library("reactome.db")
# Extract the paths of all genes of org.Hs.eg.db from KEGG (last update in # data of June 31st 2011)
genes.kegg <- as.list(org.Hs.egPATH)
# Extracts the paths of all genes of org.Hs.eg.db from reactome
genes.react <- as.list(reactomeEXTID2PATHID)
geneSim("81", "18", genes.react)
geneSim("81", "18", genes.kegg)
geneSim("81", "18", genes.react, NULL)
geneSim("81", "18", genes.kegg, NULL)

mgeneSim(c("81", "18", "10"), genes.react)
mgeneSim(c("81", "18", "10"), genes.react, "avg")
```
pathSim  Calculates the Dice similarity between pathways

Description
Calculates the similarity between pathways using dice similarity score.

Usage

pathSim(pathway1, pathway2, info)

mpathSim(pathways, info, method = "max", ...)

Arguments

pathway1, pathway2
A single pathway to calculate the similarity

info
A list of genes and the pathways they are involved.

pathways
Pathways to calculate the similarity for

method
To combine the scores of each pathway, one of c("avg", "max", "rcmax", "rcmax.avg", "BMA"),
if NULL returns the matrix of similarities.

...
Other arguments passed to combineScores

Details
diceSim is used to calculate similarities between the two pathways.
mpathSim compares the similarity between several pathways and can use combineScores to extract
the similarity between those pathways. If one needs the matrix of similarities between pathways set
the argument methods to NULL.

Value
The similarity between those pathways or all the similarities between each comparison.

Author(s)
Lluís Revilla

See Also
diceSim and combineScores and conversions help page to transform Dice score to Jaccard score.

Examples

library("reactome.db")
# Extracts the paths of all genes of org.Hs.eg.db from reactome
genes.react <- as.list(reactomeEXTID2PATHID)
pathways <- c("112315", "112310", "112316", "373753", "916853", "109582",
"114608", "1500931")
pathSim("112310", "112316", genes.react)
mpathSim(pathways, genes.react, NULL)
### removeDup

**Remove duplicated rows and columns**

**Description**

Given the indices of the duplicated entries remove the columns and rows until just one is left, it keeps the duplicated with the highest absolute mean value.

**Usage**

```r
removeDup(cor_mat, dupli)
```

**Arguments**

- `cor_mat`: List of matrices
- `dupli`: List of indices with duplicated entries

**Value**

A matrix with only one of the columns and rows duplicated

**Author(s)**

Lluis Revilla

**See Also**

- `duplicateIndices` to obtain the list of indices with duplicated entries.

**Examples**

```r
a <- seq2mat(c("52", "52", "53", "55"), runif(choose(4, 2)))
b <- seq2mat(c("52", "52", "53", "55"), runif(choose(4, 2)))
mat <- list("kegg" = a, "react" = b)
mat
dupli <- duplicateIndices(rownames(a))
remat <- removeDup(mat, dupli)
remat
```

### seq2mat

**Transforms a vector to a symmetric matrix**

**Description**

Fills a matrix of ncol = length(x) and nrow = length(x) with the values in dat and setting the diagonal to 1.

**Usage**

```r
seq2mat(x, dat)
```
similarities

Arguments

x     names of columns and rows, used to define the size of the matrix
dat   Data to fill with the matrix with except the diagonal.

Details

dat should be at least \( \text{choose}(\text{length}(x), 2) \) of length. It assumes that the data provided comes from using the row and column id to obtain it.

Value

A square matrix with the diagonal set to 1 and dat on the upper and lower triangle with the columns ids and row ids from x.

Author(s)

Llúis Revilla

See Also

upper.tri and lower.tri

Examples

```
seq2mat(LETTERS[1:5], 1:10)
seq2mat(LETTERS[1:5], seq(from = 0.1, to = 1, by = 0.1))
```

description

Function to join list of similarities by a function provided by the user.

Usage

```
similarities(sim, func, ...)
```

Arguments

sim  list of similarities to be joined. All similarities must have the same dimensions. The genes are assumed to be in the same order for all the matrices.
func function to perform on those similarities: prod, sum... It should accept as many arguments as similarities matrices are provided, and should use numbers.
...  Other arguments passed to the function func. Usually na.rm or similar.

Value

A matrix of the size of the similarities
Weighted operations

Description

Calculates the weighted sum or product of \( x \). Each values should have its weight, otherwise it will throw an error.

Usage

\[
\text{weighted.sum}(x, w, \text{abs} = \text{TRUE})
\]

\[
\text{weighted.prod}(x, w)
\]

Arguments

- \( x \) an object containing the values whose weighted operations is to be computed
- \( w \) a numerical vector of weights the same length as \( x \) giving the weights to use for elements of \( x \).
- \( \text{abs} \) If any \( x \) is negative you want the result negative too?

Details

This functions are thought to be used with \textit{similarities}. As some similarities might be positive and others negative the argument \( \text{abs} \) is provided for \textit{weighted.sum}, assuming that only one similarity will be negative (usually the one coming from expression correlation).

Value

\textit{weighted.sum} returns the sum of the product of \( x \)*\( w \)s removing all NA values. See parameter \( \text{abs} \) if there are any negative values.

\textit{weighted.prod} returns the product of product of \( x \)*\( w \)s removing all NA values.
Author(s)

Lluís Revilla

See Also

similarities and addSimilarities

Examples

```r
expr <- c(-0.2, 0.3, 0.5, 0.8, 0.1)
weighted.sum(expr, c(0.5, 0.2, 0.1, 0.1, 0.1))
weighted.sum(expr, c(0.5, 0.2, 0.1, 0.2, 0.1), FALSE)
weighted.sum(expr, c(0.4, 0.2, 0.1, 0.2, 0.1))
weighted.sum(expr, c(0.4, 0.2, 0.1, 0.2, 0.1), FALSE)
weighted.sum(expr, c(0.4, 0.2, 0, 0.2, 0.1))
weighted.sum(expr, c(0.5, 0.2, 0, 0.2, 0.1))
# Compared to weighted.prod:
weighted.prod(expr, c(0.5, 0.2, 0.1, 0.1, 0.1))
weighted.prod(expr, c(0.4, 0.2, 0.1, 0.2, 0.1))
weighted.prod(expr, c(0.4, 0.2, 0, 0.2, 0.1))
weighted.prod(expr, c(0.5, 0.2, 0, 0.2, 0.1))
```
Index

addSimilarities, 2, 15, 16
AintoB, 3

BioCor (BioCor-package), 2
BioCor-package, 2

clusterGeneSim, 2, 4, 5, 6, 8
clusterSim, 2, 5, 8
combinadic, 6
combineScores, 4–6, 7, 8, 11, 12
combn, 7
conversions, 2, 5, 6, 8, 9, 11, 12

D2J (conversions), 8
diceSim, 8, 9, 12
duplicateIndices, 10, 13
geneSim, 2, 8, 9, 10
J2D (conversions), 8

lower.tri, 14

mclusterGeneSim (clusterGeneSim), 4
mclusterSim (clusterSim), 5
mgeneSim (geneSim), 10
mpathSim (pathSim), 12

pathSim, 2, 12

removeDup, 10, 13

seq2mat, 13
similarities, 2, 3, 14, 16

upper.tri, 14

weighted, 2, 3, 15, 15