# dDAGtermSim

March 27, 2017

| dDAGtermSim | *Function to calculate pair-wise semantic similarity between input terms based on a direct acyclic graph (DAG) with annotated data* |
|---|---|

## Description

dDAGtermSim is supposed to calculate pair-wise semantic similarity between input terms based on a direct acyclic graph (DAG) with annotated data. Parallel computing is also supported for Linux or Mac operating systems.

## Usage

```
dDAGtermSim(g, terms = NULL, method = c("Resnik", "Lin", "Schlicker",
"Jiang", "Pesquita"), fast = T, parallel = TRUE, multicores = NULL,
verbose = T)
```

## Arguments

| | |
|---|---|
| g | an object of class "igraph" or "graphNEL". It must contain a vertex attribute called 'annotations' for storing annotation data (see example for howto) |
| terms | the terms/nodes between which pair-wise semantic similarity is calculated. If NULL, all terms in the input DAG will be used for calcluation, which is very prohibitively expensive! |
| method | the method used to measure semantic similarity between input terms. It can be "Resnik" for information content (IC) of most informative common ancestor (MICA) (see http://arxiv.org/pdf/cmp-lg/9511007.pdf), "Lin" for 2*IC at MICA divided by the sum of IC at pairs of terms (see https://www.cse.iitb.ac.in/~cs626-449/Papers/WordSimilarity/3.pdf), "Schlicker" for weighted version of 'Lin' by the 1-prob(MICA) (see http://www.ncbi.nlm.nih.gov/pubmed/16776819), "Jiang" for 1 - difference between the sum of IC at pairs of terms and 2*IC at MICA (see http://arxiv.org/pdf/cmp-lg/9709008.pdf), "Pesquita" for graph information content similarity related to Tanimoto-Jacard index (ie. summed information content of common ancestors divided by summed information content of all ancestors of term1 and term2 (see http://www.ncbi.nlm.nih.gov/pubmed/18460186)). By default, it uses "Schlicker" method |

| | |
|---|---|
| fast | logical to indicate whether a vectorised fast computation is used. By default, it sets to true. It is always advisable to use this vectorised fast computation; since the conventional computation is just used for understanding scripts |
| parallel | logical to indicate whether parallel computation with multicores is used. By default, it sets to true, but not necessarily does so. It will depend on whether these two packages "foreach" and "doParallel" have been installed. It can be installed via: source("http://bioconductor.org/biocLite.R"); biocLite(c("foreach","doParalle If not yet installed, this option will be disabled |
| multicores | an integer to specify how many cores will be registered as the multicore parallel backend to the 'foreach' package. If NULL, it will use a half of cores available in a user's computer. This option only works when parallel computation is enabled |
| verbose | logical to indicate whether the messages will be displayed in the screen. By default, it sets to true for display |

## Value

It returns a sparse matrix containing pair-wise semantic similarity between input terms. This sparse matrix can be converted to the full matrix via the function as.matrix

## Note

none

## See Also

dDAGinduce, dDAGancestor, dDAGgeneSim, dCheckParallel

## Examples

```
# 1) load HPPA as igraph object
ig.HPPA <-dRDataLoader(RData='ig.HPPA')
g <- ig.HPPA

# 2) load human genes annotated by HPPA
org.Hs.egHPPA <- dRDataLoader(RData='org.Hs.egHPPA')

# 3) prepare for ontology and its annotation information
dag <- dDAGannotate(g, annotations=org.Hs.egHPPA,
path.mode="all_paths", verbose=TRUE)

# 4) calculate pair-wise semantic similarity between 5 randomly chosen terms
terms <- sample(V(dag)$name, 5)
sim <- dDAGtermSim(g=dag, terms=terms, method="Schlicker",
parallel=FALSE)
sim
```