dDAGancestor

July 17, 2015

dDAGancestor	*Function to find common ancestors of two terms/nodes from a direct acyclic graph (DAG)*

Description

dDAGancestor is supposed to find a list of common ancestors shared by two terms/nodes, given a direct acyclic graph (DAG; an ontology). If two terms are given as NULL, then a sparse matrix of children x ancestors is built for all terms. If one of them is null, then a sparse matrix of children x ancestors is built but only for non-null input terms.

Usage

```
dDAGancestor(g, term1 = NULL, term2 = NULL, verbose = T)
```

Arguments

g	an object of class "igraph" or "graphNEL"
term1	the first term/node as input
term2	the second term/node as input
verbose	logical to indicate whether the messages will be displayed in the screen. By default, it sets to true for display

Value

- When two terms are given: a list of terms/nodes that are common ancestors for two input terms/nodes

- When two terms are given as NULL: a sparse matrix of children x ancestors is built for all terms, with '1' for the reachable and otherwise '0'.

- When one of terms is given as NULL: a sparse matrix of children x ancestors is built but only for non-null input terms, with '1' for the reachable and otherwise '0'.

Note

See Also

[dDAGinduce](#)

Examples

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

# 2) randomly give two terms
term1 <- sample(V(g)$name,1)
term2 <- sample(V(g)$name,1)

# 3) find common ancestors
dDAGancestor(g, term1, term2)
```