Package ‘goTools’

March 22, 2024

Version 1.76.0
Date 2009-10-23
Title Functions for Gene Ontology database
Depends GO.db
Imports AnnotationDbi, GO.db, graphics, grDevices
Suggests hgu133a.db
biocViews Microarray,GO,Visualization
Author Yee Hwa (Jean) Yang <jean@biostat.ucsf.edu>, Agnes Paquet
       <paquetagnes@yahoo.com>
Maintainer Agnes Paquet <paquetagnes@yahoo.com>
Description Wrapper functions for description/comparison of oligo ID
       list using Gene Ontology database
License GPL-2
LazyLoad yes
git_url https://git.bioconductor.org/packages/goTools
git_branch RELEASE_3_18
git_last_commit d860fb7
git_last_commit_date 2023-10-24
Repository Bioconductor 3.18
Date/Publication 2024-03-22

R topics documented:

EndNodeList ............................................................. 2
goTools ................................................................. 3
Internal functions ...................................................... 4
probeID ................................................................. 4

Index 5
Description

The function `EndNodeList` builds the default end node list used in `ontoCompare`. `CustomEndNodeList` helps you build a list of children nodes starting from a GO id of interest.

Usage

```
EndNodeList()
CustomEndNodeList(id, rank=1)
```

Arguments

- **id**: Valid GO id: "GO:XXXXXXX".
- **rank**: Number of levels of children of the GO DAG under id you want to add to your nodes list.

Value

`EndNodeList` returns a vector of all GO ids 1 level below MF("GO:0003674"), BP("GO:0008150") and CC("GO:0005575"). MF, BP and CC nodes are included. `CustomEndNodeList` returns a vector of all GO ids children of id, rank levels below it.

Author(s)

Yee Hwa (Jean) Yang, Agnes Paquet

See Also

`ontoCompare`

Examples

```
## Examples use the probeID dataset. For description type ? probeID.
## library(GO.db)
## EndNodeList()
## MFendnode <- CustomEndNodeList("GO:0003674", rank=2)

## Example (not run)
## data(probeID)
## ontoCompare(affylist, probeType="hgu133a", endnode=MFendnode, goType="MF")
```
goTools

Wrapper functions

Description

This function will allow you to describe and compare sets of oligo ids using Gene Ontology database.

Usage

ontoCompare(genelist, probeType=c("GO", "hgu133a"),
goType="All", endnode, method=c("TGenes", "TIDS", "none"), plot=FALSE, ...)

ontoPlot(objM, names.arg=NULL, beside=TRUE, las=2, legend.text=TRUE, ...)

Arguments

genelist: list of list of valid probe ids.
method: method used to evaluate the percentage of oligos for each end-node. 'TGenes' = for each end node, return the number of direct children found / total number of probe ids. (default). This includes oligos which do not have GO annotations. 'TIDS' = for each end node, return the number of direct children found / total number of GO ids describing the list. 'none' = for each end node, return the number of direct children found.
probeType: type of input given to the function. Valid probe types include GO ids and any probes ids for which a BioC annotation package providing a mapping to GO is available. ontoCompare is expecting valid probe ids.
goType: help sort the data by type. If 'All' (default), all oligos are taken into account. 'BP' restricts information to Biological Process, 'CC' to Cellular Component, and 'MF' to Molecular Function.
plot: logical: if 'TRUE', results are output as a graph.
endnode: list of GO ids corresponding to end-nodes of interest.
beside: Logical. If 'TRUE', the bars of the barplot are portrayed as juxtaposed bars. See ?barplot for more details.
las: numeric: if las=2, the axis labels are displayed perpendicular to the axis. See ?par for more details.
legend.text: vector of text used to construct a legend for the plot. See ?barplot for more details.
objM: results from ontoCompare.
names.arg: Labels to use in ontoPlot.
...: extra layout parameters to be passed to ontoPlot.
Value

Returns the percentage of probes children of nodes contained in endnode. If ‘plot’ = TRUE, results are plotted as a pie chart or a bargraph.

Author(s)

Yee Hwa (Jean) Yang, Agnes Paquet

Examples

# Examples use the probeID dataset. For description type ?probeID.
# Not run

#library(GO.db)
data(probeID)
#ontoCompare(affylist, probeType="hgu133a", plot=TRUE)
#res <- ontoCompare(operonlist["L1"], probeType="operon", method="TIDS")
#ontoPlot(res, cex=0.7)

---

**Description**

Internal goTools functions

**Details**

These are not to be called by the user.

---

**probeID**

*List of probe ids from Affymetrix hgu133a chip and Operon Version 2 Human oligos*

**Description**

The probeID dataset consists of two lists of randomly chosen probe Ids. affylist contains 3 sets of ids from Affymetrix hgu133a. operonlist contains 2 sets of ids from Operon Version 2.

**Usage**

data(probeID)
Index

* datasets
  probeID, 4
* file
  EndNodeList, 2
  goTools, 3
* hplot
  goTools, 3
* manip
  EndNodeList, 2
* methods
  Internal functions, 4
affylist (probeID), 4
CustomEndNodeList (EndNodeList), 2
EndNodeList, 2
getGOID (Internal functions), 4
getOntology (Internal functions), 4
goChildren (Internal functions), 4
goParents (Internal functions), 4
goTools, 3
Internal functions, 4
isEndNode (Internal functions), 4
ontoCompare, 2
ontoCompare (goTools), 3
ontoCompare.main (Internal functions), 4
ontoPlot (goTools), 3
operonlist (probeID), 4
parentsVectWrapper (Internal functions), 4
probeID, 4