dcEnrichment

December 23, 2014

dcEnrichment	*Function to conduct ontology enrichment analysis given a group of domains*

Description

dcEnrichment is supposed to conduct enrichment analysis for an input group of domains using a specified ontology. It returns an object of S4 class "Eoutput". Enrichment analysis is based on either Fisher's exact test or Hypergeometric test. The test can respect the hierarchy of the ontology. The user can customise the background domains; otherwise, the function will use all annotatable domains as the test background

Usage

```
dcEnrichment(data, background = NULL, domain = c(NA, "SCOP.sf",
"SCOP.fa",
"Pfam", "InterPro", "Rfam"), ontology = c(NA, "GOBP", "GOMF", "GOCC",
"DO",
"HPPA", "HPMI", "HPON", "MP", "EC", "KW", "UP"), sizeRange = c(10,
1000),
min.overlap = 3, which_distance = NULL, test = c("HypergeoTest",
"FisherTest", "BinomialTest"), p.adjust.method = c("BH", "BY",
"bonferroni",
"holm", "hochberg", "hommel"), ontology.algorithm = c("none", "pc",
"elim",
"lea"), elim.pvalue = 0.01, lea.depth = 2, verbose = T,
domain.RData = NULL, ontology.RData = NULL, annotations.RData = NULL,
RData.location = "http://dcgor.r-forge.r-project.org/data")
```

Arguments

data	an input vector. It contains id for a list of domains, for example, sunids for SCOP domains
background	a background vector. It contains id for a list of background domains, for example, sunids for SCOP domains. If NULL, by default all annotatable domains are used as background
domain	the domain identity. It can be one of 'SCOP.sf' for SCOP superfamilies, 'SCOP.fa' for SCOP families, 'Pfam' for Pfam domains, 'InterPro' for InterPro domains, 'Rfam' for Rfam RNA families

`ontology` the ontology identity. It can be "GOBP" for Gene Ontology Biological Process, "GOMF" for Gene Ontology Molecular Function, "GOCC" for Gene Ontology Cellular Component, "DO" for Disease Ontology, "HPPA" for Human Phenotype Phenotypic Abnormality, "HPMI" for Human Phenotype Mode of Inheritance, "HPON" for Human Phenotype ONset and clinical course, "MP" for Mammalian Phenotype, "EC" for Enzyme Commission, "KW" for UniProtKB KeyWords, "UP" for UniProtKB UniPathway. For details on the eligibility for pairs of input domain and ontology, please refer to the online Documentations at http://supfam.org/dcGOR/docs.html

`sizeRange` the minimum and maximum size of members of each term in consideration. By default, it sets to a minimum of 10 but no more than 1000

`min.overlap` the minimum number of overlaps. Only those terms that overlap with input data at least min.overlap (3 domains by default) will be processed

`which_distance` which distance of terms in the ontology is used to restrict terms in consideration. By default, it sets to 'NULL' to consider all distances

`test` the statistic test used. It can be "FisherTest" for using fisher's exact test, "HypergeoTest" for using hypergeometric test, or "BinomialTest" for using binomial test. Fisher's exact test is to test the independence between domain group (domains belonging to a group or not) and domain annotation (domains annotated by a term or not), and thus compare sampling to the left part of background (after sampling without replacement). Hypergeometric test is to sample at random (without replacement) from the background containing annotated and non-annotated domains, and thus compare sampling to background. Unlike hypergeometric test, binomial test is to sample at random (with replacement) from the background with the constant probability. In terms of the ease of finding the significance, they are in order: hypergeometric test > binomial test > fisher's exact test. In other words, in terms of the calculated p-value, hypergeometric test < binomial test < fisher's exact test

`p.adjust.method` the method used to adjust p-values. It can be one of "BH", "BY", "bonferroni", "holm", "hochberg" and "hommel". The first two methods "BH" (widely used) and "BY" control the false discovery rate (FDR: the expected proportion of false discoveries amongst the rejected hypotheses); the last four methods "bonferroni", "holm", "hochberg" and "hommel" are designed to give strong control of the family-wise error rate (FWER). Notes: FDR is a less stringent condition than FWER

`ontology.algorithm` the algorithm used to account for the hierarchy of the ontology. It can be one of "none", "pc", "elim" and "lea". For details, please see 'Note'

`elim.pvalue` the parameter only used when "ontology.algorithm" is "elim". It is used to control how to declare a signficantly enriched term (and subsequently all domains in this term are eliminated from all its ancestors)

`lea.depth` the parameter only used when "ontology.algorithm" is "lea". It is used to control how many maximum depth is uded to consider the children of a term (and subsequently all domains in these children term are eliminated from the use for the recalculation of the signifance at this term)

`verbose` logical to indicate whether the messages will be displayed in the screen. By default, it sets to TRUE for display

`domain.RData` a file name for RData-formatted file containing an object of S4 class 'InfoDataFrame' (i.g. domain). By default, it is NULL. It is only needed when

the user wants to customise enrichment analysis using their own data. See dcBuildInfoDataFrame for how to creat this object

ontology.RData a file name for RData-formatted file containing an object of S4 class 'Onto' (i.g. ontology). By default, it is NULL. It is only needed when the user wants to customise enrichment analysis using their own data. See dcBuildOnto for how to creat this object

annotations.RData
a file name for RData-formatted file containing an object of S4 class 'Anno' (i.g. annotations). By default, it is NULL. It is only needed when the user wants to customise enrichment analysis using their own data. See dcBuildAnno for how to creat this object

RData.location the characters to tell the location of built-in RData files. By default, it remotely locates at "http://supfam.org/dcGOR/data" or "http://dcgor.r-forge.r-project.org/data". For the user equipped with fast internet connection, this option can be just left as default. But it is always advisable to download these files locally. Especially when the user needs to run this function many times, there is no need to ask the function to remotely download every time (also it will unnecessarily increase the runtime). For examples, these files (as a whole or part of them) can be first downloaded into your current working directory, and then set this option as: $RData.location = "."$. If RData to load is already part of package itself, this parameter can be ignored (since this function will try to load it via function data first)

Value

an object of S4 class Eoutput, with following slots:

- domain: a character specifying the domain identity
- ontology: a character specifying the ontology used
- term_info: a matrix of nTerm X 5 containing term information, where nTerm is the number of terms in consideration, and the 5 columns are "term_id" (i.e. "Term ID"), "term_name" (i.e. "Term Name"), "namespace" (i.e. "Term Namespace"), "distance" (i.e. "Term Distance") and "IC" (i.e. "Information Content for the term based on annotation frequency by it")
- anno: a list of terms, each storing annotated domain members (also within the background domains). Always, terms are identified by "term_id" and domain members identified by their ids (e.g. sunids for SCOP domains)
- data: a vector containing input data in consideration. It is not always the same as the input data as only those mappable and annotatable are retained
- background: a vector containing background in consideration. It is not always the same as the input background as only those mappable/annotatable are retained
- overlap: a list of terms, each storing domains overlapped between domains annotated by a term and domains in the input data (i.e. the domains of interest). Always, terms are identified by "term_id" and domain members identified by their IDs (e.g. sunids for SCOP domains)
- zscore: a vector containing z-scores
- pvalue: a vector containing p-values
- adjp: a vector containing adjusted p-values. It is the p value but after being adjusted for multiple comparisons

Note

The interpretation of the algorithms used to account for the hierarchy of the ontology is:

- "none": does not consider the ontology hierarchy at all.
- "lea": computers the significance of a term in terms of the significance of its children at the maximum depth (e.g. 2). Precisely, once domains are already annotated to any children terms with a more signficance than itself, then all these domains are eliminated from the use for the recalculation of the signifance at that term. The final p-values takes the maximum of the original p-value and the recalculated p-value.
- "elim": computers the significance of a term in terms of the significance of its all children. Precisely, once domains are already annotated to a signficantly enriched term under the cutoff of e.g. pvalue<1e-2, all these domains are eliminated from the ancestors of that term).
- "pc": requires the significance of a term not only using the whole domains as background but also using domains annotated to all its direct parents/ancestors as background. The final p-value takes the maximum of both p-values in these two calculations.
- "Notes": the order of the number of significant terms is: "none" > "lea" > "elim" > "pc".

See Also

dcRDataLoader, dcDAGannotate, Eoutput-class, visEnrichment, dcConverter

Examples

```
# 1) Enrichment analysis for SCOP domain superfamilies (sf)
## 1a) load SCOP.sf (as InfoDataFrame object)
SCOP.sf <- dcRDataLoader(SCOP.sf)
### randomly select 50 domains as a list of domains of interest
data <- sample(rowNames(SCOP.sf), 50)
## 1b) perform enrichment analysis, producing an object of S4 class Eoutput
eoutput <- dcEnrichment(data, domain="SCOP.sf", ontology="GOMF")
eoutput
## 1c) view the top 10 significance terms
view(eoutput, top_num=10, sortBy="pvalue", details=TRUE)
## 1d) visualise the top 10 significant terms in the ontology hierarchy
### color-coded according to 10-based negative logarithm of adjusted p-values (adjp)
visEnrichment(eoutput)
## 1e) the same as above but using a customised background
### randomly select 500 domains as background
background <- sample(rowNames(SCOP.sf), 500)
### perform enrichment analysis, producing an object of S4 class Eoutput
eoutput <- dcEnrichment(data, background=background, domain="SCOP.sf",
ontology="GOMF")
eoutput
### view the top 10 significance terms
view(eoutput, top_num=10, sortBy="pvalue", details=TRUE)
### visualise the top 10 significant terms in the ontology hierarchy
### color-coded according to 10-based negative logarithm of adjusted p-values (adjp)
visEnrichment(eoutput)

############################################################
# 2) Enrichment analysis for Pfam domains (Pfam)
## 2a) load Pfam (as InfoDataFrame object)
Pfam <- dcRDataLoader(Pfam)
### randomly select 100 domains as a list of domains of interest
```

```
data <- sample(rowNames(Pfam), 100)
## 2b) perform enrichment analysis, producing an object of S4 class Eoutput
eoutput <- dcEnrichment(data, domain="Pfam", ontology="GOMF")
eoutput
## 2c) view the top 10 significance terms
view(eoutput, top_num=10, sortBy="pvalue", details=TRUE)
## 2d) visualise the top 10 significant terms in the ontology hierarchy
### color-coded according to 10-based negative logarithm of adjusted p-values (adjp)
visEnrichment(eoutput)
## 2e) the same as above but using a customised background
### randomly select 1000 domains as background
background <- sample(rowNames(Pfam), 1000)
### perform enrichment analysis, producing an object of S4 class Eoutput
eoutput <- dcEnrichment(data, background=background, domain="Pfam",
ontology="GOMF")
eoutput
### view the top 10 significance terms
view(eoutput, top_num=10, sortBy="pvalue", details=TRUE)
### visualise the top 10 significant terms in the ontology hierarchy
### color-coded according to 10-based negative logarithm of adjusted p-values (adjp)
visEnrichment(eoutput)

############################################################
# 3) Enrichment analysis for InterPro domains (InterPro)
## 3a) load InterPro (as InfoDataFrame object)
InterPro <- dcRDataLoader(InterPro)
### randomly select 100 domains as a list of domains of interest
data <- sample(rowNames(InterPro), 100)
## 3b) perform enrichment analysis, producing an object of S4 class Eoutput
eoutput <- dcEnrichment(data, domain="InterPro", ontology="GOMF")
eoutput
## 3c) view the top 10 significance terms
view(eoutput, top_num=10, sortBy="pvalue", details=TRUE)
## 3d) visualise the top 10 significant terms in the ontology hierarchy
### color-coded according to 10-based negative logarithm of adjusted p-values (adjp)
visEnrichment(eoutput)
## 3e) the same as above but using a customised background
### randomly select 1000 domains as background
background <- sample(rowNames(InterPro), 1000)
### perform enrichment analysis, producing an object of S4 class Eoutput
eoutput <- dcEnrichment(data, background=background, domain="InterPro",
ontology="GOMF")
eoutput
### view the top 10 significance terms
view(eoutput, top_num=10, sortBy="pvalue", details=TRUE)
### visualise the top 10 significant terms in the ontology hierarchy
### color-coded according to 10-based negative logarithm of adjusted p-values (adjp)
visEnrichment(eoutput)

############################################################
# 4) Enrichment analysis for Rfam RNA families (Rfam)
## 4a) load Rfam (as InfoDataFrame object)
Rfam <- dcRDataLoader(Rfam)
### randomly select 100 RNAs as a list of RNAs of interest
data <- sample(rowNames(Rfam), 100)
## 4b) perform enrichment analysis, producing an object of S4 class Eoutput
eoutput <- dcEnrichment(data, domain="Rfam", ontology="GOBP")
```

```
eoutput
## 4c) view the top 10 significance terms
view(eoutput, top_num=10, sortBy="pvalue", details=FALSE)
## 4d) visualise the top 10 significant terms in the ontology hierarchy
### color-coded according to 10-based negative logarithm of adjusted p-values (adjp)
visEnrichment(eoutput)
## 4e) the same as above but using a customised background
### randomly select 1000 RNAs as background
background <- sample(rowNames(Rfam), 1000)
### perform enrichment analysis, producing an object of S4 class Eoutput
eoutput <- dcEnrichment(data, background=background, domain="Rfam",
ontology="GOBP")
eoutput
### view the top 10 significance terms
view(eoutput, top_num=10, sortBy="pvalue", details=FALSE)
### visualise the top 10 significant terms in the ontology hierarchy
### color-coded according to 10-based negative logarithm of adjusted p-values (adjp)
visEnrichment(eoutput)

############################################################
# 5) Advanced usage: customised data for domain, ontology and annotations
# 5a) create domain, ontology and annotations
## for domain
domain <-
dcBuildInfoDataFrame(input.file="http://dcgor.r-forge.r-project.org/data/InterPro/InterPro.txt",
output.file="domain.RData")
## for ontology
dcBuildOnto(relations.file="http://dcgor.r-forge.r-project.org/data/onto/igraph_GOMF_edges.txt",
nodes.file="http://dcgor.r-forge.r-project.org/data/onto/igraph_GOMF_nodes.txt",
output.file="ontology.RData")
## for annotations
dcBuildAnno(domain_info.file="http://dcgor.r-forge.r-project.org/data/InterPro/InterPro.txt",
term_info.file="http://dcgor.r-forge.r-project.org/data/InterPro/GO.txt",
association.file="http://dcgor.r-forge.r-project.org/data/InterPro/Domain2GOMF.txt",
output.file="annotations.RData")
## 5b) prepare data and background
### randomly select 100 domains as a list of domains of interest
data <- sample(rowNames(domain), 100)
### randomly select 1000 domains as background
background <- sample(rowNames(domain), 1000)
## 5c) perform enrichment analysis, producing an object of S4 class Eoutput
eoutput <- dcEnrichment(data, background=background,
domain.RData=domain.RData, ontology.RData=ontology.RData,
annotations.RData=annotations.RData)
eoutput
## 5d) view the top 10 significance terms
view(eoutput, top_num=10, sortBy="pvalue", details=TRUE)
### visualise the top 10 significant terms in the ontology hierarchy
### color-coded according to 10-based negative logarithm of adjusted p-values (adjp)
visEnrichment(eoutput)
```