dcNaivePredict

April 23, 2015

`dcNaivePredict` *Function to perform naive prediction from input known annotations*

Description

`dcNaivePredict` is supposed to perform naive prediction from input known annotations. For each gene/protein, a term to be predicted are simply the frequency of that term appearing in the known annotations.

Usage

```
dcNaivePredict(data, GSP.file, output.file = NULL, ontology = c(NA,
"GOBP",
"GOMF", "GOCC", "DO", "HPPA", "HPMI", "HPON", "MP", "EC", "KW", "UP"),
max.num = 1000, verbose = T, RData.ontology.customised = NULL,
RData.location = "http://dcgor.r-forge.r-project.org/data")
```

Arguments

`data`	an input vector containing genes/proteins to be predicted
`GSP.file`	a Glod Standard Positive (GSP) file containing known annotations between proteins/genes and ontology terms. For example, a file containing annotations between human genes and HP terms can be found in http://dcgor.r-forge.r-project.org/data/Algo/HP_anno.txt. As seen in this example, the input file must contain the header (in the first row) and two columns: 1st column for 'SeqID' (actually these IDs can be anything), 2nd column for 'termID' (HP terms). Alternatively, the GSP.file can be a matrix or data frame, assuming that GSP file has been read. Note: the file should use the tab delimiter as the field separator between columns
`output.file`	an output file containing predicted results. If not NULL, a tab-delimited text file will be also written out; otherwise, there is no output file (by default)
`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. If NA, then the user has to input a customised RData-formatted file (see RData.ontology.customised below)

`max.num` an integer to specify how many terms will be predicted for each gene/protein

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

`RData.ontology.customised`
a file name for RData-formatted file containing an object of S4 class 'Onto' (i.e. ontology). By default, it is NULL. It is only needed when the user wants to perform customised analysis using their own ontology. See dcBuildOnto 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

a data frame containing three columns: 1st column the same as the input file (e.g. 'SeqID'), 2nd for 'Term' (predicted ontology terms), 3rd for 'Score' (along with predicted scores)

Note

When 'output.file' is specified, a tab-delimited text file is written out, with the column names: 1st column the same as the input file (e.g. 'SeqID'), 2nd for 'Term' (predicted ontology terms), 3rd for 'Score' (along with predicted scores).

See Also

dcRDataLoader, dcAlgoPropagate

Examples

```
# 1) prepare genes to be predicted
input.file <-
"http://dcgor.r-forge.r-project.org/data/Algo/HP_anno.txt"
#input.file <- "http://dcgor.r-forge.r-project.org/data/Algo/SCOP_architecture.txt"
input <- utils::read.delim(input.file, header=T, sep="\t",
colClasses="character")
data <- unique(input[,1])

# 2) Calculate Precision and Recall
GSP.file <- "http://dcgor.r-forge.r-project.org/data/Algo/HP_anno.txt"
res <- dcNaivePredict(data=data, GSP.file=GSP.file, ontology="HPPA")
res[1:10,]
```

```
# 3) calculate Precision and Recall
res_PR <- dcAlgoPredictPR(GSP.file=GSP.file, prediction.file=res,
ontology="HPPA")
res_PR

# 4) Plot PR-curve
plot(res_PR[,2], res_PR[,1], xlim=c(0,1), ylim=c(0,1), type="b",
xlab="Recall", ylab="Precision")
```