dcAlgoPredictPR

December 22, 2014

`dcAlgoPredictPR` *Function to assess the prediction performance via Precision-Recall (PR) analysis*

Description

`dcAlgoPredictPR` is supposed to assess the prediction performance via Precision-Recall (PR) analysis. It requires two input files: 1) a Glod Standard Positive (GSP) file containing known annotations between proteins/genes and ontology terms; 2) a prediction file containing predicted terms for proteins/genes. Note: the known annotations will be recursively propagated towards the root of the ontology.

Usage

```
dcAlgoPredictPR(GSP.file, prediction.file, ontology = c(NA, "GOBP",
"GOMF",
"GOCC", "DO", "HPPA", "HPMI", "HPON", "MP", "EC", "KW", "UP"),
num.threshold = 10, bin = c("uniform", "quantile"), verbose = T,
RData.ontology.customised = NULL,
RData.location = "http://dcgor.r-forge.r-project.org/data")
```

Arguments

`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

`prediction.file` a prediction file containing proteins/genes, their predicted terms along with predictive scores. As seen in an example below, this file is usually created via `dcAlgoPredictMain`, containing three columns: 1st column for 'SeqID' (actually these IDs can be anything), 2nd column for 'Term' (ontology terms), 3rd column for 'Score' (predictive score). Alternatively, the prediction.file can be a matrix or data frame, assuming that prediction file has been read. Note: the file should use the tab delimiter as the field separator between columns

`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)

`num.threshold` an integer to specify how many PR points (as a function of the score threshold) will be calculated

`bin` how to bin the scores. It can be "uniform" for binning scores with equal interval (ie with uniform distribution), and 'quantile' for binning scores with eual frequency (ie with equal number)

`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.g. 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 two columns: 1st column 'Precision' for precision, 2nd 'Recall' for recall. The row has the names corresponding to the score threshold.

Note

Prediction coverage: the ratio between predicted targets in number and GSP targets in number F-measure: the maximum of a harmonic mean between precision and recall along PR curve

See Also

`dcRDataLoader`, `dcConverter`, `dcDuplicated`, `dcAlgoPredictMain`

Examples

```
# 1) Generate prediction file with HPPA predicitions for human genes
architecture.file <-
"http://dcgor.r-forge.r-project.org/data/Algo/SCOP_architecture.txt"
prediction.file <- "SCOP_architecture.HPPA_predicted.txt"
res <- dcAlgoPredictMain(input.file=architecture.file,
output.file=prediction.file, RData.HIS="Feature2HPPA.sf",
parallel=FALSE)

# 2) Calculate Precision and Recall
GSP.file <- "http://dcgor.r-forge.r-project.org/data/Algo/HP_anno.txt"
res_PR <- dcAlgoPredictPR(GSP.file=GSP.file,
prediction.file=prediction.file, ontology="HPPA")
res_PR

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