Package 'PADOG'

October 18, 2017

Version 1.18.0

Date 2015-03-20

Title Pathway Analysis with Down-weighting of Overlapping Genes (PADOG)

Author Adi Laurentiu Tarca <atarca@med.wayne.edu>; Zhonghui Xu <zhonghui.xu@gmail.com>

Depends R (>= 3.0.0), KEGGdzPathwaysGEO, methods,Biobase

Suggests doParallel, parallel

Maintainer Adi Laurentiu Tarca <atarca@med.wayne.edu>

Description This package implements a general purpose gene set analysis method called PADOG that downplays the importance of genes that apear often accross the sets of genes to be analyzed. The package provides also a benchmark for gene set analysis methods in terms of sensitivity and ranking using 24 public datasets from KEGGdzPathwaysGEO package.

License GPL (>= 2)

URL

Collate padog.R compPADOG.R filteranot.R

Imports limma, AnnotationDbi, GSA, foreach, doRNG, hgu133plus2.db, hgu133a.db, KEGG.db, nlme

LazyLoad yes

biocViews Microarray, OneChannel, TwoChannel

NeedsCompilation no

R topics documented:

compPADOG *Benchmark for gene set analysis methods using 24 datasets*

Description

This is a general purpose function to compare a given gene set analysis method in terms of sensitivity and ranking agains PADOG and GSA (if installed) using 24 public datasets.

Usage

```
compPADOG(datasets=NULL,existingMethods=c("GSA","PADOG"),mymethods=NULL,gs.names=NULL,gslist="KE
        ncr=NULL, pkgs=NULL, expVars=NULL, dseed=NULL, plots=FALSE,verbose=FALSE)
```

Arguments

datasets	A character vector with valid names of datasets to use from the PADOGsets package. If left NULL all datasets avalibale in PADOGsets will be used.
existingMethods	A character vector with one or more of the predefined methods c("GSA","PADOG"). The first is used as reference method.
mymethods	A list whose elements are valid functions implementing gene set analysis methods. See the example to see what arguments the functions have to take in and what kind of output they need to produce.
gslist	Either the value "KEGG.db" or a list with the gene sets. If set to "KEGG.db", then gene sets will be made of all KEGG pathways for human since all datasets available in PADOG are for human.
organism	A three letter string giving the name of the organism supported by the "KEGG.db" package.
gs.names	A character vector giving additional information about each gene set. For instance when gene seta are pathways, the full name of the pathway would be a meaningful gene set name.
Nmin	The minimum size of gene sets to be included in the analysis for all methods.
NI	Number of iterations to determine the gene set score significance p-values in PADOG and GSA methods.
parallel	Should paralell be used if multiple cores are available and the package parallel is available. If se to TRUE one dataset will be run on on multiple CPU at a time (Not available on Windows).
ncr	The number of CPU cores used when use.parallel set to TRUE. Default is to use all CPU cores detected.
pkgs	Character vector of packages that the existingMethods and mymethods depend on (NULL for "PADOG" and "GSA"). Consult the .packages argument in foreach function from foreach package.
expVars	Character vector of variables to export. Consult the .export argument in foreach function from foreach package.
dseed	Optional initial seed for random number generator (integer) used in padog.
plots	If set to TRUE will plot the ranks of the target genesets and the ranks differences between a methods and the reference method.
verbose	This argument will be passed to PADOG and AbsmT methods. If set to TRUE it will show the interations performed so far.

Details

See cited documents for more details.

Value

A data frame containing the : `Method` is the name of the geneset analysis method; `p geomean` geometric mean of nominal p-values for the target genesets (genesets expected to be relevant); `p med` median of nominal p-values for the target genesets; `% p<0.05` is the fraction of target genesets significant at 0.05 level (this is the sensitivity); `% q<0.05` is the fraction of target genesets significant at 0.05 level after FDR correction; `rank mean` mean rank of the target genesets; `rank med` median rank of the target genesets; `p Wilcox.` p value from a Wilcoxon test paired at dataset level comparing the rank of target genesets ; `p LME` p value from a linear mixed effects (LME) model which unlike the Wilcoxon test above accounts for the fact that ranks for the same pathway may be correlated; `coef LME` Coefficient from the LME model giving the difference in ranks of the target genesets between the current geneset analysis Method and the reference method chose to be the first method in the `existingMethods` argument;

Author(s)

Adi Laurentiu Tarca <atarca@med.wayne.edu>

References

Adi L. Tarca, Sorin Draghici, Gaurav Bhatti, Roberto Romero, Down-weighting overlapping genes improves gene set analysis, BMC Bioinformatics, 2012, submitted.

See Also

compPADOG

Examples

```
#compare a new geneset analysis method with PADOG and GSA

#define your new gene set analysis method that takes as input:
#set- the name of dataset file from the PADOGsetspackage
#mygslist - a list with the genesets
#minsize- minimum number of genes in a geneset to be considered for analysis

randomF=function(set,mygslist,minsize){
set.seed(1)
#this loads the dataset in an ExpressionSet object called x
data(list=set,package="KEGGdzPathwaysGEO")
x=get(set)

#Extract from the dataset the required info to be passed to padog
exp=experimentData(x);
dat.m=exprs(x)
ano=pData(x)
dataset= exp@name
design= notes(exp)$design
annotation= paste(x@annotation,".db",sep="")
targetGeneSets= notes(exp)$targetGeneSets
```

```
#get rid of duplicates probesets per ENTREZ ID by keeping the probeset
#with smallest p-value (computed using limma)
aT1=filteranot(esetm=dat.m,group=ano$Group,paired=(design=="Paired"),
 block=ano$Block,annotation=annotation)
#create an output dataframe for this toy method with random gene set p-values
mygslistSize=unlist(lapply(mygslist,function(x){length(intersect(aT1$ENTREZID,x))}))
res=data.frame(ID=names(mygslist),P=runif(length(mygslist)),
 Size=mygslistSize,stringsAsFactors=FALSE)
res$FDR=p.adjust(res$P,"fdr")
#drop genesets with less than minsize genes in the current dataset
res=res[res$Size>=minsize,]
#compute ranks
res$Rank=rank(res$P)/dim(res)[1]*100
#needed to compare ranks between methods; must be the same as given
#in mymethods argument "list(myRand="
res$Method="myRand";
#needed because comparisons of ranks between methods is paired at dataset level
res$Dataset<-dataset;
#output only result for the targetGeneSets
#which are gene sets expected to be relevant in this dataset
return(res[res$ID %in% targetGeneSets,])
}

#run the analysis on all 24 datasets and compare the new method "myRand" with
#PADOG and GSA (if installed) (chosen as reference since is listed first in the existingMethods)
#if the package parallel is installed datasets are analyzed in parallel.
#out=compPADOG(datasets=NULL,existingMethods=c("GSA","PADOG"),
 #mymethods=list(myRand=randomF),
 #gslist="KEGG.db",Nmin=3,NI=1000,plots=TRUE,verbose=FALSE)

#compare myRand against PADOG on 4 datasets only
#mysets=data(package="PADOGsets")$results[,"Item"]
mysets=c("GSE9348","GSE8671","GSE1297")
out=compPADOG(datasets=mysets,existingMethods=c("PADOG"),
 mymethods=list(myRand=randomF),
 gslist="KEGG.db",Nmin=3,NI=20,plots=FALSE,verbose=FALSE)
```

`filteranot` *Remove duplicate probesets/probes from an gene expression matrix based on p-values from a moderated t-test, in order to apply a gene set analysis.*

Description

This function helps to deal with multiple probesets/probes per gene prior to geneset analysis.

Usage

```
filteranot(esetm=NULL,group=NULL,paired=FALSE,block=NULL,annotation=NULL,include.details=FALSE)
```

Arguments

esetm	A matrix containing log transfomed and normalized gene expression data. Rows correspond to genes and columns to samples. Rownames of esetm need to be valid probeset or probe names.
group	A character vector with the class labels of the samples. It can only contain "c" for control samples or "d" for disease samples.
paired	A logical value to indicate if the samples in the two groups are paired.
block	A character vector indicating the block ids of the samples classified by the group variable, if paired=TRUE. The paired samples must have the same block value.
annotation	A valid chip annotation package name (e.g. "hgu133plus2.db")
include.details	If set to true, will include all columns from limma's topTable for this dataset.

Details

See cited documents for more details.

Value

A data frame containing the probeset IDs (and corresponding ENTREZ IDs) of the best probesets for each gene ;

Author(s)

Adi Laurentiu Tarca <atarca@med.wayne.edu>

References

Adi L. Tarca, Sorin Draghici, Gaurav Bhatti, Roberto Romero, Down-weighting overlapping genes improves gene set analysis, BMC Bioinformatics, 2012, submitted.

See Also

padog

Examples

```
#run padog on a colorectal cancer dataset of the 24 datasets benchmark GSE9348
set="GSE9348"
data(list=set,package="KEGGdzPathwaysGEO")
x=get(set)
#Extract from the dataset the required info
exp=experimentData(x);
dataset= exp@name
dat.m=exprs(x)
ano=pData(x)
design= notes(exp)$design
annotation= paste(x@annotation,".db",sep="")

dim(dat.m)
#get rid of duplicates in the same way as is done for PADOG and assign probesets to ENTREZ IDS
```

```
#get rid of duplicates by choosing the probe(set) with lowest p-value; get ENTREZIDs for probes
aT1=filteranot(esetm=dat.m,group=ano$Group,paired=(design=="Paired"),block=ano$Block,annotation)

#filtered expression matrix
filtexpr=dat.m[rownames(dat.m)%in%aT1$ID,]
dim(filtexpr)
```

---

padog *Pathway Analysis with Down-weighting of Overlapping Genes (PADOG)*

Description

This is a general purpose gene set analysis method that downplays the importance of genes that apear often accross the sets of genes analyzed. The package provides also a benchmark for gene set analysis in terms of sensitivity and ranking using 24 public datasets.

Usage

```
padog(esetm=NULL,group=NULL,paired=FALSE,block=NULL,gslist="KEGG.db",organism="hsa",
      annotation=NULL,gs.names=NULL,NI=1000,plots=FALSE,targetgs=NULL,Nmin=3,
      verbose=TRUE,parallel=FALSE,dseed=NULL,ncr=NULL)
```

Arguments

esetm	A matrix containing log transfomed and normalized gene expression data. Rows correspond to genes and columns to samples.
group	A character vector with the class labels of the samples. It can only contain "c" for control samples or "d" for disease samples.
paired	A logical value to indicate if the samples in the two groups are paired.
block	A character vector indicating the block ids of the samples classified by the group variable, if paired=TRUE. The paired samples must have the same block value.
gslist	Either the value "KEGG.db" or a list with the gene sets. If set to "KEGG.db", then gene sets will be made of all KEGG pathways for the organism specified. If a list is provided, instead, each element of the list should be a character vector with the identifiers for the genes. The identifiers can be probe(sets) ids if the annotation argument is set to a valid annotation package, otherwise the gene identifiers must be of the same kind as the rownames of the matrix esetm.
annotation	A valid chip annotation package if the rownames of esetm are probe(set) ids and gslist contains ENTREZ identifiers or gslist is set to "KEGG.db". If the rownames are other gene identifies, then annotation has tyo be set to NULL, and the row names of esetm needs to be unique and be found among elements of gslist
organism	A three letter string giving the name of the organism supported by the "KEGG.db" package.
gs.names	Character vector with the names of the gene sets. If specified, must have the same length as gslist.

NI	Number of iterations to determine the gene set score significance p-values.
plots	If set to TRUE then the distribution of the PADOG scores with and without weighting the genes in raw and standardized form are shown using boxplots. A pdf file will be created in the current directory having the name provided in the targetgs field. The scores for the targetgs gene set will be shown in red.
targetgs	The identifier of a traget gene set for which the scores will be highlighted in the plots produced if plots=TRUE
Nmin	The minimum size of gene sets to be included in the analysis.
verbose	If set to TRUE, displays the number of iterations elapsed is displayed.
parallel	If set to TRUE, the NI iterations will be executed in parallel if multiple CPU cores are available and foreach and doRNG packages are installed.
dseed	Optional initial seed for random number generator (integer).
ncr	The number of CPU cores used when parallel set to TRUE. Default is to use all CPU cores detected.

Details

See cited documents for more details.

Value

A data frame containing the ranked pathways and various statistics: Name is the name of the gene set; ID is the gene set identifier; Size is the number of genes in the geneset; meanAbsT0 is the mean of absolute t-scores; padog0 is the mean of weighted absolute t-scores; PmeanAbsT significance of the meanAbsT0; Ppadog is the significance of the padog0 score;

Author(s)

Adi Laurentiu Tarca <atarca@med.wayne.edu>

References

Adi L. Tarca, Sorin Draghici, Gaurav Bhatti, Roberto Romero, Down-weighting overlapping genes improves gene set analysis, BMC Bioinformatics, 2012, submitted.

See Also

padog

Examples

```
#run padog on a colorectal cancer dataset of the 24 datasets benchmark GSE9348
#use NI=1000 for accurate results.
set="GSE9348"
data(list=set,package="KEGGdzPathwaysGEO")
x=get(set)
#Extract from the dataset the required info
exp=experimentData(x);
dataset= exp@name
dat.m=exprs(x)
ano=pData(x)
```

```
design= notes(exp)$design
annotation= paste(x@annotation,".db",sep="")
targetGeneSets= notes(exp)$targetGeneSets


myr=padog(
esetm=dat.m,
group=ano$Group,
paired=design=="Paired",
block=ano$Block,
targetgs=targetGeneSets,
annotation=annotation,
gslist="KEGG.db",
organism="hsa",
verbose=TRUE,
Nmin=3,
NI=25,
plots=FALSE,
dseed=1)


myr2=padog(
esetm=dat.m,
group=ano$Group,
paired=design=="Paired",
block=ano$Block,
targetgs=targetGeneSets,
annotation=annotation,
gslist="KEGG.db",
organism="hsa",
verbose=TRUE,
Nmin=3,
NI=25,
plots=FALSE,
dseed=1,
paral=TRUE,
ncr=2)


myr[1:20,]

all.equal(myr, myr2)
```

Index