Package 'MetaDE'

May 5, 2017

Type Package

Title MetaDE: Transcriptome meta-analysis for differentially expressed gene detection

Version 2.2.3

Date 2017-05-05

Author Tianzhou Ma, Xingbin Wang, Jia Li and George C Tseng

Maintainer Tianzhou Ma <tianzhou.ma0105@gmail.com>

Depends survival, limma, samr, edgeR, DESeq2, impute, Biobase, combinat

Description MetaDE package implements 12 major meta-analysis methods for differential expression analysis.

License GPL-2

RoxygenNote 6.0.1

R topics documented:

`heatmap.sig.genes` *A function to plot the heatmap of DE genes detected at a given FDR threshold from the Meta-analysis.*

Description

Heatmap of selected DE genes The `heatmap.sig.genes` is a function to draw the Heatmap of DE genes given a FDR cut point obtained from the Meta-analysis.

Usage

```
heatmap.sig.genes(result, meta.method, fdr.cut, color = "GR")
```

Arguments

`result`	is the output from MetaDE.
`meta.method`	is the meta-analysis method used in MetaDE.
`fdr.cut`	is the FDR cutoff used to select the DE genes.
`color`	is the color of the heatmap.

Value

a figure shows the standardized expression levels for the DE genes detected by meta analysis across studies/datasets.

Examples

```
## Not run:
meta.method <- 'AW'
fdr.cut <- 1e-7
pdf('heatmap.test.pdf')
heatmap.sig.genes(meta.res, meta.method=meta.method,
                    fdr.cut=fdr.cut,color="GR")
dev.off()

## End(Not run)
```

`Indi.DE.Analysis` *Main Function for Individual Study DE: microarray & RNAseq.*

Description

Main Function for Individual Study DE: microarray & RNAseq The `Indi.DE.Analysis` is a function to perform individual association analysis between gene expression and the response/phenoype of interest (can be either group, continuous or survival).

Usage

```
Indi.DE.Analysis(data, clin.data, data.type, resp.type, response,
  covariate = NULL, ind.method, select.group = NULL, ref.level = NULL,
  paired = NULL, asymptotic = NULL, nperm = NULL, tail = "abs",
  seed = 12345, ...)
```

Arguments

`data`	is a list of K elements, where K is the number of studies, each element is a microarray or RNAseq expression matrix with G rows and N columns, where G is number of matched genes and N is the sample size.
`clin.data`	is a list of K elements, each element includes is a clinical data frame with N rows and p columns, where N is the sample size and p is the number of clinical variables (main response included).
`data.type`	is a character indicating the data type of the elements in `data`, must be "continuous" or "discrete".
`resp.type`	is a character indicating the response type of the `response` variable selected, must be one of "twoclass", "multiclass", "continuous" and "survival".
`response`	is one column name of `clin.data`, indicating the phenotype of interest. For survival, two column names have to be specified, the first is the survival time and the second is the censoring status.
`covariate`	are the clinical covariates you wish to adjust for in the DE analysis, can be a vector of column names or NULL.
`ind.method`	is a character vector to specify the method used to test if there is association between the gene expression and outcome variable. must be one of "limma", "sam" for "continuous" data type and "edgeR", "DESeq2" or "limmaVoom" for "discrete" data type.
`select.group:`	for two-class comparison only, specify the two groups for comparison when the group factor has more than two levels.
`ref.level:`	for two-class/multi-class comparison only, specify the reference level of the group factor.
`paired:`	logical value indicating whether paired design;
`asymptotic:`	a logical value indicating whether asymptotic distribution should be used. If FALSE, permutation will be performed.
`nperm:`	the number of permutations. Applicable when `asymptotic` is FALSE.
`tail:`	a character string specifying the alternative hypothesis, must be one of "abs" (default), "low" or "high". For resp.type = "continuous", "survival" only.
`seed:`	Optional initial seed for random number generator.

Value

a list with components:

- p: For all types of response, the p-value of the association test for each gene
- stat: For "continuous" and "survival" only, the value of test statistic for each ##' gene
- bp: For "continuous" and "survival" only, the p-value from `nperm:` permutations for each gene. It will be used for the meta analysis by default. It can be NULL if you chose asymptotic results.

- log2FC: For "twoclass" only, the log2 fold change for each gene
- lfcSE: For "twoclass" only, the standard error of log2 fold change for each gene

Examples

```
data('Leukemia')
data('LeukemiaLabel')
data <- Leukemia
K <- length(data)
clin.data <- lapply(label, function(x) {data.frame(x)} )
for (k in 1:length(clin.data)){
 colnames(clin.data[[k]]) <- "label"
}
select.group <- c('inv(16)','t(15;17)')
ref.level <- "inv(16)"
data.type <- "continuous"
ind.method <- c('limma','limma','limma')
resp.type <- "twoclass"
paired <- rep(FALSE,length(data))
ind.res <- Indi.DE.Analysis(data=data,clin.data= clin.data,
                        data.type=data.type,resp.type = resp.type,
                        response='label',
                        ind.method=ind.method,select.group = select.group,
                        ref.level=ref.level,paired=paired)
N <- sapply(data, FUN=function(x) ncol(x))
survival.time <- lapply(N,FUN = function(x) round(runif(x,10,2000)))
censor.status <- lapply(N,FUN = function(x) sample(c(0,1),x,replace=TRUE) )
for (k in 1:length(clin.data)){
  clin.data[[k]] <- cbind(clin.data[[k]],survival.time[[k]],censor.status[[k]])
  colnames(clin.data[[k]])[2:3] <- c("survival","censor")
}
ind.method <- c('logrank','logrank','logrank')
resp.type <- "survival"
ind.res <- Indi.DE.Analysis(data=data,clin.data= clin.data,
                         data.type=data.type,resp.type = resp.type,
                         response=c("survival","censor"),
                         ind.method=ind.method,asymptotic=TRUE)
```

Leukemia *Leukemia data*

Description

Leukemia expression data of 3 studies, each study has 3 groups

Usage

```
data("Leukemia")
```

Examples

```
data(Leukemia)
```

`LeukemiaLabel` *Leukemia data group labels*

Description

Leukemia data group labels of 3 studies

Usage

```
data("LeukemiaLabel")
```

Examples

```
data(LeukemiaLabel)
```

`MetaDE` *Main Function for Meta DE analysis: microarray & RNAseq.*

Description

Main Function for Meta analysis: microarray & RNAseq The `MetaDE` is a function to identify genes associated with the response/phenoype of interest (can be either group, continuous or survival) by integrating multiple studies(datasets). The main input consists of raw expression data.

Usage

```
MetaDE(data, clin.data, data.type, resp.type, response, covariate = NULL,
  ind.method, meta.method, select.group = NULL, ref.level = NULL,
  paired = NULL, rth = NULL, REM.type = NULL, asymptotic = NULL,
  tail = "abs", parametric = TRUE, nperm = NULL, seed = 12345, ...)
```

Arguments

`data`	is a list of K elements, where K is the number of studies, each element is a microarray or RNAseq expression matrix with G rows and N columns, where G is number of matched genes and N is the sample size.
`clin.data`	is a list of K elements, each element includes is a clinical data frame with N rows and p columns, where N is the sample size and p is the number of clinical variables (main response included).
`data.type`	is a character indicating the data type of the elements in `data`, must be "continuous" or "discrete".
`resp.type`	is a character indicating the response type of the `response` variable selected, must be one of "twoclass", "multiclass", "continuous" and "survival".
`response`	is one column name of `clin.data`, indicating the phenotype of interest. For survival, two column names have to be specified, the first is the survival time and the second is the censoring status.
`covariate`	are the clinical covariates you wish to adjust for in the DE analysis, can be a vector of column names or NULL.

`ind.method` is a character vector to specify the method used to test if there is association between the gene expression and outcome variable. must be one of "limma", "sam" for "continuous" data type and "edgeR", "DESeq2" or "limmaVoom" for "discrete" data type.

`meta.method` is a character to specify the Meta-analysis method used to combine the p-values, effect sizes or ranks.

`paired` is a logical vecter of size K to indicate whether the study is paired design?

`rth` is the option for roP and roP.OC method. rth means the rth smallest p-value.

`REM.type` is the option for "REM" method only, choose from "HS","HO", "DL", "SJ", "EB" or "RML".

`asymptotic` is a logical value indicating whether asymptotic distribution should be used. If FALSE, permutation will be performed.

`tail` is a character string specifying the alternative hypothesis, must be one of "abs" (default), "low" or "high".

`parametric` is a logical values indicating whether the parametric methods is chosen to calculate the p-values in meta-analysis.

`nperm` is the number of permutations. Applicable when `parametric` is FALSE.

`select.group:` for two-class comparison only, specify the two groups for comparison when the group factor has more than two levels.

`ref.level:` for two-class/multi-class comparison only, specify the reference level of the group factor.

`seed:` Optional initial seed for random number generator.

Value

a list with components:

- stat: a matrix with rows representing genes. It is the statistic for the selected meta analysis method of combining p-values.
- pval: the p-value from meta analysis for each gene for the above stat.
- FDR: the FDR of the p-value for each gene for the above stat.
- AW.weight: The optimal weight assigned to each dataset/study for each gene if the 'AW' method was chosen.

Examples

```
data('Leukemia')
data('LeukemiaLabel')
data <- Leukemia
K <- length(data)
clin.data <- lapply(label, function(x) {data.frame(x)} )
for (k in 1:length(clin.data)){
 colnames(clin.data[[k]]) <- "label"
}
select.group <- c('inv(16)','t(15;17)')
ref.level <- "inv(16)"
data.type <- "continuous"
ind.method <- c('limma','limma','sam')
resp.type <- "twoclass"
```

```
paired <- rep(FALSE,length(data))
meta.method <- "Fisher"
meta.res <- MetaDE(data=data,clin.data = clin.data,
                    data.type=data.type,resp.type = resp.type,
                    response='label',
                    ind.method=ind.method, meta.method=meta.method,
                    select.group = select.group, ref.level=ref.level,
                    paired=paired,tail='abs',parametric=TRUE)
meta.method <- "Fisher.OC"
meta.res <- MetaDE(data=data,clin.data = clin.data,
                    data.type=data.type,resp.type = resp.type,
                    response='label',
                    ind.method=ind.method, meta.method=meta.method,
                    select.group = select.group, ref.level=ref.level,
                    paired=paired,tail='high',parametric=FALSE,nperm=100)
meta.method <- "FEM"
meta.res <- MetaDE(data=data,clin.data = clin.data,
                    data.type=data.type,resp.type = resp.type,
                    response='label',
                    ind.method=ind.method, meta.method=meta.method,
                    select.group = select.group, ref.level=ref.level,
                    paired=paired, tail='abs')
meta.method <- "REM"
REM.type <- "HO"
meta.res <- MetaDE(data=data,clin.data = clin.data,
                    data.type=data.type,resp.type = resp.type,
                    response='label',
                    ind.method=ind.method, meta.method=meta.method,
                    select.group = select.group, ref.level=ref.level,
                    paired=paired,
                    REM.type=REM.type,tail='abs')
meta.method <- "SR"
meta.res <- MetaDE(data=data,clin.data = clin.data,
                    data.type=data.type,resp.type = resp.type,
                    response='label',
                    ind.method=ind.method, meta.method=meta.method,
                    select.group = select.group, ref.level=ref.level,
                    paired=paired,tail='abs',parametric=FALSE,nperm=100)
meta.method <- 'minMCC'
meta.res <- MetaDE(data=data,clin.data = clin.data,
                    data.type=data.type,resp.type = resp.type,
                    response='label',
                    ind.method=ind.method, meta.method=meta.method,
                    select.group = select.group, ref.level=ref.level,
                    paired=paired,tail='abs',parametric=FALSE,nperm=100)
meta.method <- "AW"
meta.res <- MetaDE(data=data,clin.data = clin.data,
                    data.type=data.type,resp.type = resp.type,
                    response='label',covariate = NULL,
                    ind.method=ind.method, meta.method=meta.method,
                    select.group = select.group, ref.level=ref.level,
                    paired=paired, rth=NULL,
                    REM.type=NULL,tail='abs',parametric=TRUE)
```

MetaDE.pvalue *Meta analysis by combining p-value*

Description

Meta analysis by combining p-value The MetaDE is a function to identify genes associated with the response/phenoype of interest (can be either group, continuous or survival) by combining p-values from multiple studies(datasets). The main input consists of p-values from your own method/calculations.

Usage

```
MetaDE.pvalue(x, meta.method, rth = NULL, parametric = TRUE)
```

Arguments

x is a list with components:

- p: a list of p values for each dataset.
- bp: a list of p values calculated from permutation for each dataset. This part can be NULL if you just have the p-values from your own method.

meta.method is a character to specify the Meta-analysis method used to combine the p-values.

rth is the option for roP and roP.OC method. rth means the rth smallest p-value.

parametric is a logical values indicating whether the parametric methods is chosen to calculate the p-values in meta-analysis.

x is a list with components:

Value

a list with components:

- stat: a matrix with rows representing genes. It is the statistic for the selected meta analysis method of combining p-values.
- pval: the p-value from meta analysis for each gene for the above stat.
- FDR: the FDR of the p-value for each gene for the above stat.
- AW.weight: The optimal weight assigned to each dataset/study for each gene if the 'AW' method was chosen.

Examples

```
data('Leukemia')
data('LeukemiaLabel')
data <- Leukemia
K <- length(data)
clin.data <- lapply(label, function(x) {data.frame(x)} )
for (k in 1:length(clin.data)){
 colnames(clin.data[[k]]) <- "label"
}
select.group <- c('inv(16)','t(15;17)')
ref.level <- "inv(16)"
data.type <- "continuous"
ind.method <- c('limma','limma','limma')
resp.type <- "twoclass"
paired <- rep(FALSE,length(data))
ind.res <- Indi.DE.Analysis(data=data,clin.data= clin.data,
                        data.type=data.type,resp.type = resp.type,
                        response='label',
                        ind.method=ind.method,select.group = select.group,
                        ref.level=ref.level,paired=paired)
meta.method <- "AW"
meta.res <- MetaDE.pvalue(ind.res,meta.method,rth=NULL,parametric=TRUE)
summary <- data.frame(ind.p = meta.res$ind.p,
                      stat = meta.res$meta.analysis$stat,
                      pval = meta.res$meta.analysis$pval,
                      FDR = meta.res$meta.analysis$FDR,
                      weight = meta.res$meta.analysis$AW.weight)
```

PathAnalysis *Main Function for pathway analysis.*

Description

Main Function for pathway analysis The PathAnalysis is a function to perform pathway analysis (a.k.a. gene set enrichment test) for functional annotation of a candidate gene list or an ordered gene result from Meta DE analysis output.

Usage

```
PathAnalysis(meta.p = NULL, pathway = c(Biocarta.genesets, GOBP.genesets,
  GOCC.genesets, GOMF.genesets, KEGG.genesets, Reactome.genesets),
  enrichment = c("KS", "Fisher's exact"), p.cut = NULL,
  DEgene.number = 200, size.min = 15, size.max = 500)
```

Arguments

meta.p	is a vector of meta-analysis p-value.
pathway	is a vector of pathway databases used for functional analysis, see data(pathways) for more details.
enrichment	is the method used for pathway analysis, must be one of "KS" and "Fisher's exact".
p.cut	is the p-value cutoff to select the DE genes, option for Fisher's exact method only.
DEgene.number	is the top number of DE genes, option for Fisher's exact method only.
size.min	is the minimum pathway size to be included in the functional analysis.
size.max	is the maximum pathway size to be included in the functional analysis.

Value

a data frame with columns:

- pvalue the p-value from pathway analysis for each pathway
- qvalue the q-value from pathway analysis for each pathway
- OddsRatio optional, the odds ratio from Fisher's exact test method
- logOR optional, the log odds ratio from Fisher's exact test method
- DEgenes optional, the set of DE genes in each pathway

Examples

```
## Not run:
meta.p  <- meta.res$meta.analysis$pval
ks.result <- PathAnalysis(meta.p = meta.p, enrichment = "KS")
fisher.result <- PathAnalysis(meta.p = meta.p, enrichment = "Fisher's exact")

## End(Not run)
```

pathways *Pathway Database*

Description

A total of 25 Pathway Databases

Usage

```
data("pathways")
```

Examples

```
data(pathways)
```

posthoc.aw *Post-hoc analysis on AW results.*

Description

Post-hoc analysis on AW results The `posthoc.aw` is a function to perform post-hoc analysis on AW results to determine the overall effect size directionality.

Usage

```
posthoc.aw(result)
```

Arguments

`result`	is the output from MetaDE AW method

Value

a new AW result with additional result (the overall effect size directionality) from the post-hoc analysis.

Examples

```
## Not run:
posthoc.result <- posthoc.aw(result=meta.res)

## End(Not run)
```

summary.meta *Function to summarize the meta-analysis results.*

Description

Function to summarize the results in tabular form The `summary.meta` is a function to summarize the meta-analysis results from the MetaDE output.

Usage

```
summary.meta(result, meta.method, resp.type)
```

Arguments

`result`	is the output from MetaDE
`meta.method`	is the meta-analysis method used in MetaDE
`resp.type`	is a character indicating the response type, must be one of "twoclass", "multiclass", "continuous" and "survival".

Value

a summary table including individual study statistics and pvalue, meta-analysis test statistics, pvalue, FDR, AW weights etc.

Examples

```
## Not run:
meta.method <- 'AW'
resp.type <- "twoclass"
summary.result <- summary.meta(result=meta.res,
                              meta.method = meta.method,
                              resp.type = resp.type)
posthoc.result <- posthoc.aw(result=meta.res)
summary.posthoc.result <- summary.meta(result=posthoc.result,
                              meta.method = meta.method,
                              resp.type = resp.type)

## End(Not run)
```

Index