Package: AWFisher (via r-universe)

October 29, 2026

Type Package

Title An R package for fast computing for adaptively weighted fisher's method

Version 1.27.0

Date 2020-02-08

Author Zhiguang Huo

Maintainer Zhiguang Huo <zhuo@ufl.edu>

biocViews StatisticalMethod, Software

VignetteBuilder knitr

Description Implementation of the adaptively weighted fisher's method, including fast p-value computing, variability index, and meta-pattern.

License GPL-3

Depends R (>= 3.6)

Imports edgeR, limma, stats

BugReports https://github.com/Caleb-Huo/AWFisher/issues

Suggests knitr, tightClust

RoxygenNote 6.1.1

NeedsCompilation no

Repository https://bioc.r-universe.dev

Date/Publication 2026-04-28 12:50:30 UTC

RemoteUrl https://github.com/bioc/AWFisher

RemoteRef HEAD

RemoteSha bc26c90de3eca276edac1d834b05c90ef650e53a

Contents

AWFisher_pvalue 2
biomarkerCategorization 3
data_mouseMetabolism 4
function_edgeR 5
function_limma 6
variabilityIndex 7

Index **9**

`AWFisher_pvalue` *AWFisher*

Description

R package for fast computing for adaptively weighted fisher's method

Usage

```
AWFisher_pvalue(p.values)
```

Arguments

`p.values`	Input G by K p-value matrix. Each row represent a gene and each column represent a study. Note that K has to be >=2 and <=100.

Details

fast computing for adaptively weighted fisher's method

Value

A list consisting of AWFisher pvalues and AWweight.

`pvalues`	AWFisher pvalues.
`weights`	G by K binary weight matrix W. $W_gk = 1$ represents for gene g, study k contributes to the meta-analysis result. $W_gk = 0$ otherwise.

Author(s)

Zhiguang Huo

Examples

```
K <- 40
G <- 10000
p.values = matrix(rbeta(K*G, 1,1), ncol=K)
res = AWFisher_pvalue(p.values)
hist(res$pvalues, breaks=40)
table(rowSums(res$weights))
pvalues=res$pvalues[order(res$pvalues)]
plot(-log10((1:NROW(pvalues))/(1+NROW(pvalues))),
 -log10(pvalues),xlab='theoretical quantile', ylab='observed quantile')
lines(c(0,100), c(0,100), col=2)
```

---

biomarkerCategorization

biomarker categrorization

Description

biomarker categrorization

Usage

```
biomarkerCategorization(studies, afunction, B = 10, DEindex = NULL,
  fdr = NULL, silence = FALSE)
```

Arguments

studies	a list of K studies. Each element (kth study) of the list is another list consisting gene expression matrix and and label information.
afunction	A function for DE analysis. Options can be function_limma or function_edgeR. Default option is function_limma. However, use could define their own function. The input of afunction should be list(data, label) which is consistent with one element of the studies list/argument. The return of afunction should be list(pvalue=apvalue, effectSize=aeffectsize)
B	number of permutation should be used. B=1000 is suggested.
DEindex	If NULL, BH method will be applied to p-values and FDR 0.05 will be used. User could specify a logical vector as DEindex.
fdr	Default is 0.05. The co-membership matrix calculation will base on genes with this specified fdr.
silence	If TRUE, will print out the bootstrapping procedure.

Details

biomarker categrorization via boostrap AW weight.

Value

A list consisting of biomarker categrorization result.

varibility	Varibility index for all genes
dissimilarity	Dissimilarity matrix of genes of DEindex==TRUE
DEindex	DEindex for Dissimilarity

Author(s)

Zhiguang Huo

Examples

```
N0 = 10
G <- 1000
GDEp <- 50
GDEn <- 50
K = 4

studies <- NULL
set.seed(15213)
for(k in seq_len(K)){
    astudy <- matrix(rnorm(N0*2*G),nrow=G,ncol=N0*2)
    ControlLabel <- seq_len(N0)
    caseLabel <- (N0 + 1):(2*N0)

    astudy[1:GDEp,caseLabel] <- astudy[1:GDEp,caseLabel] + 2
    astudy[1:GDEp + GDEn,caseLabel] <- astudy[1:GDEp + GDEn,caseLabel] - 2

    alabel = c(rep(0,length(ControlLabel)),rep(1,length(caseLabel)))

    studies[[k]] <- list(data=astudy, label=alabel)
}


result <- biomarkerCategorization(studies,function_limma,B=100,DEindex=NULL)
sum(result$DEindex)
head(result$varibility)
print(result$dissimilarity[1:4,1:4])
```

---

data_mouseMetabolism *Mouse metabolism microarray data*

Description

The purpose of the multi-tissue mouse metabolism transcriptomic data is to study how the gene expression changes with respect to the energy deficiency using mouse models. Very long-chain acyl-CoA dehydrogenase (VLCAD) deficiency was found to be associated with energy metabolism

disorder in children. Two genotypes of the mouse model - wild type (VLCAD +/+) and VLCAD-deficient (VLCAD -/-) - were studied for three types of tissues (brown fat, liver, heart) with 3 to 4 mice in each genotype group. The sample size information is available in the table below. A total of 6,883 genes are available in this example dataset.

Usage

```
data_mouseMetabolism
```

Format

A list of data.frame with 6,883 genes (rows) and 3 - 4 mouse samples in each genotype group (columns).

brown data for the brown fat tissue

heart data for the heart tissue

liver data for the liver tissue

Source

https://projecteuclid.org/download/pdfview_1/euclid.aoas/1310562214

Examples

```
data(data_mouseMetabolism)
```

`function_edgeR` *use edgeR function to get pvalue*

Description

use edgeR function to get pvalue

Usage

```
function_edgeR(astudy)
```

Arguments

`astudy`	A list contains a data matrix and a vector of group label

Details

use edgeR function to get pvalue

Value

A list of pvalue and effect size

Author(s)

Zhiguang Huo

Examples

```
N0 = 10
G <- 1000
GDEp <- 50
GDEn <- 50

set.seed(15213)

astudy <- matrix(rpois(N0*2*G,10),nrow=G,ncol=N0*2)
ControlLabel <- 1:N0
caseLabel <- (N0 + 1):(2*N0)

astudy[1:GDEp,caseLabel] <- astudy[1:GDEp,caseLabel] + 2
astudy[1:GDEp + GDEn,caseLabel] <- astudy[1:GDEp,caseLabel] - 2

alabel <- c(rep(0,length(ControlLabel)),rep(1,length(caseLabel)))
Study <- list(data=astudy, label=alabel)

result <- function_edgeR(Study)
fdr <- p.adjust(result$pvalue)
sum(fdr<=0.05)
```

---

function_limma *use limma function to get pvalue*

Description

use limma function to get pvalue

Usage

```
function_limma(astudy)
```

Arguments

astudy	A list contains a data matrix and a vector of group label

Details

use limma function to get pvalue

Value

A list of pvalue and effect size

Author(s)

Zhiguang Huo

Examples

```
N0 = 10
G <- 1000
GDEp <- 50
GDEn <- 50

set.seed(15213)

astudy <- matrix(rnorm(N0*2*G),nrow=G,ncol=N0*2)
ControlLabel <- 1:N0
caseLabel <- (N0 + 1):(2*N0)

astudy[1:GDEp,caseLabel] <- astudy[1:GDEp,caseLabel] + 2
astudy[1:GDEp + GDEn,caseLabel] <- astudy[1:GDEp,caseLabel] - 2

alabel <- c(rep(0,length(ControlLabel)),rep(1,length(caseLabel)))
Study <- list(data=astudy, label=alabel)

result <- function_limma(Study)
fdr <- p.adjust(result$pvalue)
sum(fdr<=0.05)
```

---

variabilityIndex *Variability Index*

Description

Variability Index

Usage

```
variabilityIndex(studies, afunction, B = 10, silence = FALSE)
```

Arguments

studies	a list of K studies. Each element (kth study) of the list is another list consisting gene expression matrix and and label information.
afunction	A function for DE analysis. Options can be function_limma or function_edgeR. Default option is function_limma. However, use could define their own function. The input of afunction should be list(data, label) which is consistent with one element of the studies list/argument. The return of afunction should be list(pvalue=apvalue, effectSize=aeffectsize)
B	number of permutation should be used. B=1000 is suggested.
silence	If TRUE, will print out the bootstrapping procedure.

Details

Variability Index via boostrap AW weight.

Value

A list consisting of biomarker categrorization result.

`varibility` Varibility index for all genes

Author(s)

Zhiguang Huo

Examples

```
N0 = 10
G <- 1000
GDEp <- 50
GDEn <- 50
K = 4

studies <- NULL
set.seed(15213)
for(k in 1:K){
    astudy <- matrix(rnorm(N0*2*G),nrow=G,ncol=N0*2)
    ControlLabel <- 1:N0
    caseLabel <- (N0 + 1):(2*N0)

    astudy[1:GDEp,caseLabel] <- astudy[1:GDEp,caseLabel] + 2
    astudy[1:GDEp + GDEn,caseLabel] <- astudy[1:GDEp + GDEn,caseLabel] - 2

    alabel = c(rep(0,length(ControlLabel)),rep(1,length(caseLabel)))

    studies[[k]] <- list(data=astudy, label=alabel)
}


result <- variabilityIndex(studies,function_limma,B=100)
head(result)
```

Index

∗ **datasets**
data_mouseMetabolism, 4

AWFisher_pvalue, 2

biomarkerCategorization, 3

data_mouseMetabolism, 4

function_edgeR, 5
function_limma, 6

variabilityIndex, 7