Package 'PROMISE'

January 21, 2025

Type Package

Title PRojection Onto the Most Interesting Statistical Evidence

Description A general tool to identify genomic features with a specific biologically interesting pattern of associations with multiple endpoint variables as described in Pounds et. al. (2009) Bioinformatics 25: 2013-2019

Version 1.58.0

Date 2014-6-24

Author Stan Pounds <stanley.pounds@stjude.org>, Xueyuan Cao <xueyuan.cao@stjude.org>

Maintainer Stan Pounds <stanley.pounds@stjude.org>, Xueyuan Cao <xueyuan.cao@stjude.org>

Depends R (>= 3.1.0), Biobase, GSEABase

Imports Biobase, GSEABase, stats

License GPL (>= 2)

LazyLoad yes

biocViews Microarray, OneChannel, MultipleComparison, GeneExpression

git_url https://git.bioconductor.org/packages/PROMISE

git_branch RELEASE_3_20

git_last_commit 9fb1418

git_last_commit_date 2024-10-29

Repository Bioconductor 3.20

Date/Publication 2025-01-20

Contents

PROMISE-package *PRojection Onto the Most Interesting Statistical Evidence*

Description

a tool to identify genomic geatures with a specific biologically interesting pattern of associations with multiple endpoint variables

Details

Package:	PROMISE
Type:	Package
Version:	1.17.0
Date:	2014-6-24
License:	GPL (>=2)
LazyLoad:	yes

The PROMISE (PRojection Onto the Most Interesting Statistical Evidence) is performed by calling function PROMISE. The array data and endpoint data are passed through an ExpressionSet; the gene set definition is passed through a GeneSetCollection, and PROMISE definition is passed through a data frame. *promise.genestat* and *avg.abs.genestat* are called internally by PROMISE. Two R routines for calculating association statistics with individual endpoint variable(*jung.rstat* and *spearman.rstat*) are provided in this version. Users could provide their own R routines written in a similar fashion.

Author(s)

Stan Pounds <stanley.pounds@stjude.org>; Xueyuan Cao <xueyuan.cao@stjude.org>

Maintainer: Stan Pound <stanley.pounds@stjude.org>; Xueyuan Cao <xueyuan.cao@stjude.org>

References

Jung, S-H, Owzar K, and Goerge SL (2005) A multiple testing procedure to associate gene expression levels with survival. Biostatistics 24: 3077-3088.

Goeman JJ and Buhlmann P (2007) Analyzing gene expression data in terms of gene sets: methodological issues. Bioinformatics 23: 980-987.

Pounds S, Cheng C, Cao X, Crews KR, Plunkett W, Gandhi V, Rubnitz J, Ribeiro RC, Downing JR, and Lamba J (2009) PROMISE: a tool to identify genomic features with a specific biologically interesting pattern of associations with multiple endpoint variables. Bioinformatics 25: 2013-2019

Examples

```
## load sampExprSet, sampGeneSet, phPatt.
data(sampExprSet)
data(sampGeneSet)
data(phPatt)

## Perform PROMISE procedure without GSEA
test1 <- PROMISE(exprSet=sampExprSet,
                 geneSet=NULL,
                 promise.pattern=phPatt,
                 strat.var=NULL,
                 proj0=FALSE,
                 nbperm=FALSE,
                 max.ntail=10,
                 seed=13,
                 nperms=100)

## Perform PROMISE procedure with GSEA and using fast permuation
test2 <- PROMISE(exprSet=sampExprSet,
                 geneSet=sampGeneSet,
                 promise.pattern=phPatt,
                 strat.var=NULL,
                 proj0=TRUE,
                 nbperm=TRUE,
                 max.ntail=10,
                 seed=13,
                 nperms=100)
```

---

avg.abs.genestat *Function to Compute Gene Set Statistics*

Description

A function to calculate the mean of absolute values of statistics based on a gene set definition

Usage

```
avg.abs.genestat(gene.res, probes, GS.data)
```

Arguments

gene.res	a data frame. Each row gives test statistics for a genomic variable. Each column corresponds to an endpoint variable.
probes	a vector that links the *gene.res* to *GS.data*.
GS.data	a data frame with first column for probe set identifier and second column for gene set identifier. Each row assigns a probe set to a gene set. Each probe set may be assigned to multiple gene sets or no gene set at all.

Value

Return a matrix of statistics. Each row gives the mean absolute value of test statistics of genes belonging to a gene set. The columns are same as in *gene.res*.

Note

A function internally called by *PROMISE*.

Author(s)

Stan Pounds <stanley.pounds@stjude.org>; Xueyuan Cao <xueyuan.cao@stjude.org>

References

Goeman JJ and Buhlmann P (2007) Analyzing gene expression data in terms of gene sets: methodological issues. Bioinformatics 23: 980-987.

See Also

PROMISE

Examples

```
## load sampExprSet sampGeneSet.
data(sampExprSet)
data(sampGeneSet)

## extract expression matrix from sampExprSet
Y <- exprs(sampExprSet)
probes <- rownames(Y)

## convert sampGeneSet to a data frame
GS.data <- NULL
for (i in 1:length(sampGeneSet)){
    tt <- sampGeneSet[i][[1]]
    this.name <- unlist(geneIds(tt))
    this.set <- setName(tt)
    GS.data <- rbind.data.frame(GS.data,
                cbind.data.frame(featureID=as.character(this.name),
                                 setID=rep(as.character(this.set),
                                 length(this.name))))
}

## Calculate the mean of absolute values of statistics
## This is only a demo, probe expression values are used
##in stead of statistics
test <- avg.abs.genestat(Y, probes, GS.data)
```

---

jung.rstat *Function to Compute Jung's Statistics*

Description

Compute statistic that measures the correlation of many continuous variables with a censored time-to-event variable

Usage

```
jung.rstat(x, time.cens, strat = NULL)
```

Arguments

x	a data frame with row corresponding to probe set and column corresponding to subjects, the order of columns (subjects) should match the order of rows in *time.cens*.
time.cens	a data frame with number of row equal to number of column in *x*. It contains two columns with first for time and second for censor (1 = event, 0 = censored).
strat	a vector of stratum to calculate stratified r-type association statistics, default = NULL.

Value

Returns a vector of Jun's r-type association statistics.

Note

The order of subjects in *x* (column), *time.cens*, and *strat* should all match. The original statistic proposed by Jung, Owzar, and George can be written as a dot-product. The statistic returned by this routine is expressed in the form of a correlation statistic by dividing the dot product by the square root of the lengths of the two vectors in the numerator.

Author(s)

Stan Pounds <stanley.pounds@stjude.org>; Xueyuan Cao <xueyuan.cao@stjude.org>

References

Jung SH, Owzar K, and George SL (2005) A multiple testing procedure to associate gene expression levels with survival. Stat Med 24:3077-88

See Also

PROMISE

Examples

```
## load sampExprSet.
data(sampExprSet)

## extract expression matrix from sampExprSet
Y <- exprs(sampExprSet)

## extract end point data from sampExprSet
time.cens <-pData(phenoData(sampExprSet))[, 3:4]
strat <- pData(phenoData(sampExprSet))$strat

## compute Jung's r-type association statistics
jungstat <- jung.rstat(Y, time.cens, strat = strat)
```

phPatt *Phenotype Pattern Definition Set*

Description

This hypothetical phenotype pattern definition set *phPatt* has three columns: stat.coef, stat.func, and endpt.vars. It defines an associatin pattern for three phenotypes.

Usage

```
data(phPatt)
```

PROMISE *PRojection onto the Most Interesting Statistical Evidence*

Description

Perform permutation-based test to identify genes with expression levels having a specific biologically interesting pattern of associations with multiple endpoint variables

Usage

```
PROMISE(exprSet, geneSet=NULL, promise.pattern, strat.var=NULL, proj0=FALSE, seed=13, nbperm=FALSE
        nperms=10000)
```

Arguments

exprSet	an ExpressionSet class contains minimum of *exprs* (*expression matrix*) and *phenoData* (*AnnotatedDataFrame of end point data*). Please refer to *Biobase* for details on how to create such an ExpressionSet.
geneSet	a GeneSetCollection class with minimum of setName and geneIDs for each GeneSet. Please refer to *GSEABase* for how to create such a GeneSetCollection class. The default is NULL which will perform no gene set enrichment analysis.
promise.pattern	a data frame defining the association pattern of interest. The column names must be *stat.coef*, *stat.func*, and *endpt.vars*. The *stat.coef* column gives the coefficients for combining the statistics of association of genomic variables with individual endpoint variable into the ultimate PROMISE statistic. If *proj0=TRUE*, the *stat.coeff* is ignored. The *stat.func* column gives the name of the R routine that computes the test statistic of association of the endpoint variables. Two R routines (*jung.rstat* and *spearman.rstat*)are provided. Users can provide their own routine accordingly. The *endpt.vars* column gives the name(s) of variable(s) in the endpoint data file needed to compute each term of the PROMISE statistic. A common without a space should be used to separate multiple variables that correspond to the same term in the association pattern definition.
strat.var	the name or numeric value of stratum variable in *exprSet* for stratified analysis. The default is NULL which performs an unstratified analysis.

proj0	indicator of whether projection to 0 is performed. It takes two valid values: TRUE or FALSE. If *proj0=TRUE*, PROMISE statistics is the sum of squares of individual statitics and the *stat.coeff* in *promise.pattern* is ignored. The default is FALSE.
seed	initial seed of random number generator. The default is 13.
nbperm	indicator of fast permuation using negative binomial strategy, taking two valid values: FALSE or TRUE. The default is FALSE.
max.ntail	number of sucess if *nbperm = T*. Further permutation will not be performed for gene(s) or gene set(s) which *max.ntail* permutated statistics are greater or equal to the observed statistics, The default is 100
nperms	number of permutations. The default is 10,000.

Value

$generes	individual genes' test statistics and p-values for each individual endpoint and PROMISE analysis. If *nbperm=T*, the last column contains number of permuations for each gene.
$setres	gene set's test statistics and p-values for each individual endpoint and PROMISE analysis. If *nbperm=T*, the last column contains number of permuations for gene set. If *geneSet* is NULL, the value of this component is also *NULL*.

Author(s)

Stan Pounds <stanley.pounds@stjude.org>; Xueyuan Cao <xueyuan.cao@stjude.org>

References

Pounds S, Cheng C, Cao X, Crews KR, Plunkett W, Gandhi V, Rubnitz J, Ribeiro RC, Downing JR, and Lamba J (2009) PROMISE: a tool to identify genomic features with a specific biologically interesting pattern of associations with multiple endpoint variables. Bioinformatics 25: 2013-2019

See Also

jung.rstat avg.abs.genestat promise.genestat spearman.rstat promise.pattern

Examples

```
## load sampExprSet, sampGeneSet, phPatt.
data(sampExprSet)
data(sampGeneSet)
data(phPatt)

## Perform PROMISE procedure without GSEA
test1 <- PROMISE(exprSet=sampExprSet,
                 geneSet=NULL,
                 promise.pattern=phPatt,
                 strat.var=NULL,
                 proj0=FALSE,
                 nbperm=FALSE,
                 max.ntail=10,
                 seed=13,
                 nperms=100)
```

```
## Perform PROMISE procedure with GSEA and using fast permuation
test2 <- PROMISE(exprSet=sampExprSet,
                 geneSet=sampGeneSet,
                 promise.pattern=phPatt,
                 strat.var=NULL,
                 proj0=TRUE,
                 nbperm=TRUE,
                 max.ntail=10,
                 seed=13,
                 nperms=100)
```

---

promise.genestat *Function to Calculate PROMISE Statistics*

Description

a function to calculate individual gene and PROMISE statistics for a defined pattern of association

Usage

```
promise.genestat(Y, ph.data, ph.pattern, strat = NULL, proj0=FALSE)
```

Arguments

Y	a data frame with row corresponding to probe set and column corresponding to subjects, the order of column should match order of row in *ph.data*.
ph.data	a data frame with rows corresponding to subjects and columns corresponding to endpoint variables.
ph.pattern	a data frame with column headers: *stat.coef*, *stat.func*, *endpt.vars*. The *stat.coeff* column gives the coefficients for combining the statistics of association of genomic variable with individual endpoint variable into the ultimate PROMISE statistic. If *proj0=TRUE*, the *stat.coeff* is ignored. The *stat.func* column gives the name of the R routine that computes the test statistic of association of the end point variables. *jung.rstat* and *spearman.rstat* are provided. Users can provide their own routines accordingly. The *endpt.vars* column gives the name(s) of variable(s) in *ph.data* needed to compute each term of the PROMISE statistic. A **comma** without a space should be used to separate multiple variables that correspond to the same term in the association pattern definition.
strat	a vector of stratum to calculate stratified statistics. The default is NULL.
proj0	indicator of whether projection to 0 is performed. It takes two valid values: TRUE or FALSE. If proj0=TRUE, PROMISE statistics is the sum of squares of individual statitics. The default is FALSE.

Value

a matrix of statistics. Each row gives gene's statistics of each individual endpoint and the PROMISE statistics defined in *ph.pattern*.

Note

a function internally called by PROMISE.

Author(s)

Stan Pounds <stanley.pounds@stjude.org>; Xueyuan Cao <xueyuan.cao@stjude.org>

References

Pounds S, Cheng C, Cao X, Crews KR, Plunkett W, Gandhi V, Rubnitz J, Ribeiro RC, Downing JR, and Lamba J (2009) PROMISE: a tool to identify genomic features with a specific biologically interesting pattern of associations with multiple endpoint variables. Bioinformatics 25: 2013-2019

See Also

PROMISE

Examples

```
## load sampExprSet, phPatt.
data(sampExprSet)
data(phPatt)

Y <- exprs(sampExprSet)
ph.data <- pData(phenoData(sampExprSet))

test <- promise.genestat(Y, ph.data, phPatt, strat=ph.data[, 5])
test2 <- promise.genestat(Y, ph.data, phPatt, strat=ph.data[, 5], proj0=TRUE)
```

---

promise.pattern *PROMISE pattern*

Description

PROMISE pattern is a data frame of association pattern definition, consisting of three columns.

Format

PROMISE pattern: The column names must be *stat.coef*, *stat.func*, and *endpt.vars*.

stat.coef column gives the coefficients for combining the statistics of association of genomic variable with individual endpoint variable into the ultimate PROMISE statistic.

stat.func column gives the name of the R routine that computes the test statistic of association of the endpoint variables. Two R routines (*jung.rstat* and *spearman.rstat*)are provided in current release. Users can provide their own routine accordingly.

endpt.vars column gives the name(s) of variable(s) in the endpoint data frame needed to compute each term of the PROMISE statistic. If more than one variables involve in one term, they should be separated by a **comma** without space.

Author(s)

Stan Pounds <stanley.pounds@stjude.org>; Xueyuan Cao <xueyuan.cao@stjude.org>

References

Pounds S, Cheng C, Cao X, Crews KR, Plunkett W, Gandhi V, Rubnitz J, Ribeiro RC, Downing JR, and Lamba J (2009) PROMISE: a tool to identify genomic features with a specific biologically interesting pattern of associations with multiple endpoint variables. Bioinformatics 25: 2013-2019

See Also

PROMISE

---

sampExprSet *An Example Expression Set*

Description

This hypothetical expression set *sampExpSet* belongs to an *ExpressionSet* class. It contains 100 genomic features (probe_1 to probe_100) for 50 subjects (array_1 to array_50) and phenotype data of drugLevel, residualDisease, obsTime, obsCensor and strat. The expression values can be accessed by *exprs(sampExprSet)*. The phenotype data can be accessed by *pData(phenoData(sampExprSet))*

Usage

```
data(sampExprSet)
```

---

sampGeneSet *An Example Gene Set Collection*

Description

This hypothetical gene set *sampGeneSet* belongs to a *GeneSetCollection* class. It contains 10 gene sets (*GeneSet* class).

Usage

```
data(sampGeneSet)
```

spearman.rstat *Function to Calculate Spearman Correlation Statistics*

Description

A function to calculate Spearman rank correlation of each gene in an array data with a continuous variable

Usage

```
spearman.rstat(Y, x, strat = NULL)
```

Arguments

Y	a numeric data frame. Each row gives values of one genomic variable.
x	a vector of continuous variable.
strat	a vector of stratum to calculate stratified correlation statistics, default = NULL.

Value

Return a vector of Spearman rank correlation statistics.

Author(s)

Stan Pounds <stanley.pounds@stjude.org>; Xueyuan Cao <xueyuan.cao@stjude.org>

References

Spearman C. (1904) The proof and measurement of association between two things. Amer. J. Psychol. 15: 72-101

See Also

PROMISE

Examples

```
## load sampExprSet.
data(sampExprSet)

## extract expression matrix from sampExprSet
Y <- exprs(sampExprSet)

## extract end point data from sampExprSet
x <- pData(phenoData(sampExprSet))$drugLevel
strat <- pData(phenoData(sampExprSet))$strat

## Calculte Spearman correlation statistics
test <- spearman.rstat(Y, x, strat = strat)
```

Index