Package 'copa'

November 10, 2026

Title Functions to perform cancer outlier profile analysis.

Version 1.80.0

Date 2006-01-26

Author James W. MacDonald

Maintainer James W. MacDonald <jmacdon@u.washington.edu>

Description COPA is a method to find genes that undergo recurrent fusion in a given cancer type by finding pairs of genes that have mutually exclusive outlier profiles.

Depends Biobase, methods

Suggests colonCA

License Artistic-2.0

LazyLoad yes

biocViews OneChannel, TwoChannel, DifferentialExpression, Visualization

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

git_branch RELEASE_3_23

git_last_commit 606f635

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-10

Contents

copa-package 2
copa 3
copaFilter 4
copaPerm 5
getans 7
perm.mat 7
plotCopa 8

pSum 9
scatterPlotCopa 10
summaryCopa 11
tableCopa 12

Index **13**

`copa-package` *copa - A package to compute 'Cancer Outlier Profile Analysis'*

Description

This package is used to compute copa scores, p-values based on permutation, and plots of paired genes.

Details

Package:	copa
Type:	Package
Version:	1.1.2
Date:	2006-01-26
License:	Artistic

There are two main functions; `copa`, which is used to compute the COPA score for a set of microarrays, and `permCopa`, which is used to calculate permutation based p-values and estimate false discovery rate (FDR).

Author(s)

James W. MacDonald

Maintainer: James W. MacDonald <jmacdon@u.washington.edu>

References

Tomlins, SA, et al. Recurrent fusion of TMPRSS2 and ETS transcription factor genes in prostate cancer. Science. 2005 Oct 28;310(5748):644-8.

copa *Calculate COPA Scores from a Set of Microarrays*

Description

This function calculates COPA scores from a set of microarrays. Input can be an `ExpressionSet`, or a matrix or `data.frame`.

Usage

```
copa(object, cl, cutoff = 5, max.overlap = 0, norm.count = 0, pct = 0.95)
```

Arguments

`object`	An `ExpressionSet`, or a matrix or `data.frame`.
`cl`	A vector of classlabels indicating sample status (normal = 1, tumor = 2).
`cutoff`	The cutoff to determine 'outlier' status. See details for more information.
`max.overlap`	The maximum number of samples that can be considered 'outliers' when comparing two genes. The default is 0, indicating that there can be no overlap. See details for more information.
`norm.count`	The number of normal samples that can be considered 'outliers'. The default is 0, meaning that no normals may be outliers.
`pct`	The percentile to use for pre-filtering the data. A preliminary step is to compute the number of outlier samples for each gene. All genes with a number of outlier samples less than the (default 95th) percentile will be removed from further consideration.

Details

Cancer Outlier Profile Analysis is a method that is intended to find pairs of genes that may be involved in recurrent gene fusion with a third (unknown) gene. The underlying idea here is that in certain cancers it may be common for the promoter region of one gene to become fused to certain oncogenes. For instance, Tomlins et. al. showed that the promoter region of TMPRSS2 fused to either ERG or ETV1 in the majority of prostate cancer tumors tested.

Since this fusion should only happen with one oncogene in a given sample, we look for pairs of genes where some samples have much higher expression values, but the samples for gene 'A' are mutually exclusive from the samples for gene 'B'.

The cutoff argument for this function is used to determine how high the centered and scaled expression value has to be in order to be considered an outlier. The max.overlap argument allows one to relax the requirement of mutual exclusivity, although in practice this is probably not advisable.

Note that this function computes all row-wise comparisons, which gets very large very quickly. The function will throw a warning for any data set containing > 1000 rows and query the user to see if he/she really wants to proceed. The number of genes to be considered can be adjusted by increasing/decreasing the 'pct' argument.

Value

ord.prs	A matrix with two columns containing the ordered row numbers from the original matrix of gene expression values.
pr.sums	A numeric vector with the number of mutually exclusive outliers for each gene pair. This is the criterion for ranking the gene pairs; the assumption being that a pair of genes with more mutually exclusive outliers will be more interesting than a pair with relatively fewer outliers.
mat	A matrix containing the filtered gene expression values.
cl	The classlabel vector passed to copa
cutoff	The cutoff used
max.overlap	The value of max.overlap used
norm.count	The value of norm.count used
pct	The percentile used in the pre-filtering step

Author(s)

James W. MacDonald

References

Tomlins, SA, et al. Recurrent fusion of TMPRSS2 and ETS transcription factor genes in prostate cancer. Science. 2005 Oct 28;310(5748):644-8.

Examples

```
library(Biobase)
data(sample.ExpressionSet)
cl <- abs(3 - as.numeric(pData(sample.ExpressionSet)[,2]))
tmp <- copa(sample.ExpressionSet, cl)
```

---

copaFilter *Pre-filter Genes for COPA Analysis*

Description

This function is used to pre-filter genes prior to doing a COPA analysis. The filtering is based on the nth percentile of the outlier samples for each gene. This function is an internal function and not intended to be called by the end user.

Usage

```
## S4 method for signature 'matrix'
copaFilter(object, cl, cutoff, norm.count, pct)
## S4 method for signature 'data.frame'
copaFilter(object, cl, cutoff, norm.count, pct)
## S4 method for signature 'ExpressionSet'
copaFilter(object, cl, cutoff, norm.count, pct)
```

Arguments

object	An ExpressionSet, or a matrix or data.frame.
cl	A vector of classlabels indicating sample status (normal = 1, tumor = 2).
cutoff	The cutoff to determine 'outlier' status. See details for more information.
norm.count	The number of normal samples that can be considered 'outliers'. The default is 0, meaning that no normals may be outliers.
pct	The percentile to use for pre-filtering the data. A preliminary step is to compute the number of outlier samples for each gene. All genes with a number of outlier samples less than the (default 95th) percentile will be removed from further consideration.

Value

mat	A matrix containing the gene expression values for the filtered genes.

Author(s)

James W. MacDonald

References

Tomlins, SA, et al. Recurrent fusion of TMPRSS2 and ETS transcription factor genes in prostate cancer. Science. 2005 Oct 28;310(5748):644-8.

---

copaPerm *Measure Significance of COPA by Permutation*

Description

This function can be used to determine the significance of the results that one gets from running copa on a particular dataset, based on permuting the class assignments.

Usage

```
copaPerm(object, copa, outlier.num, gene.pairs, B = 100, pval = FALSE, verbose = TRUE)
```

Arguments

object	An ExpressionSet, or a matrix or data.frame.
copa	An object of class 'copa', produced by running copa on a set of microarray data.
outlier.num	The number of outliers to test for. See details for more information
gene.pairs	The number of gene pairs to test for. See details for more information
B	The number of permutations to perform. Defaults to 100. This may be too many for interactive use.

pval	Boolean. Output an estimated p-value and false discovery rate? Defaults to FALSE. This result will only be reasonable for large numbers of permutations (500 - 1000). See details.
verbose	Boolean. Print out the permutation number at each of 100, 200, etc. Defaults to TRUE

Details

Running copa on a set of microarray data will result in the output of an object of class 'copa', which is a list containing (among other things) an ordered vector that lists the number of mutually exclusive outlier samples for various gene pairs. This vector is ordered from smallest to largest following the assumption that the gene pairs with the most mutually exclusive outliers are probably more likely to be involved in some sort of recurrent fusion.

One can see how many pairs of genes resulted in a given number of outliers by calling tableCopa. One may then want to determine how significant a certain number of pairs is (e.g., how likely is it to get that many pairs if there is no recurrent fusion occuring). The most straightforward way to estimate the significance of a given result is to repeatedly permute the classlabels and see how many times one gets a result as large or larger than what was observed.

Technically speaking, to get a reasonable estimate of significance and a false discovery rate, one would need to permute 500 - 1000 times. However, this can take an inordinate amount of time (best left for an overnight run). To get a quick idea of significance, one could simply permute maybe 10 times (with pval = FALSE) to see how likely it is to get a certain number of outliers.

Value

out	A vector listing the number of gene pairs with at least as many outliers as 'num.outlier'.
p.value	A permuted p-value, only output if pval = TRUE. Note that the size of the p-value is determined by both the number of outliers >= 'num.outlier' as well as the number of permutations, so too few permutations may result in a p-value that doesn't look very significant even if it is.
fdr	The expected number of gene pairs with at least as many outliers as 'num.outlier'. This can be converted to a %FDR by dividing by the observed value.

Author(s)

James W. MacDonald

References

Tomlins, SA, et al. Recurrent fusion of TMPRSS2 and ETS transcription factor genes in prostate cancer. Science. 2005 Oct 28;310(5748):644-8.

getans *Interactive Function*

Description

A function to query the end user. This is an internal function and not intended to be called directly by the end user.

Usage

```
getans(msg, allowed = c("y", "n"))
```

Arguments

msg	The query.
allowed	Allowed responses

Value

The response is returned.

Author(s)

James W. MacDonald

perm.mat *Produce a Matrix of Permuted Classlabels*

Description

This function makes a matrix of permuted classlabels. This is not intended to be called directly by end users.

Usage

```
perm.mat(B, ids)
```

Arguments

B	The number of permutations
ids	A vector of classlabels

Value

A matrix of permuted classlabels.

Author(s)

James W. MacDonald

`plotCopa` *Plot Gene Pairs fom the Results of Running copa*

Description

This function can be used to visualize pairs of genes that may be involved in recurrent gene fusion in cancer.

Usage

```
plotCopa(copa, idx, lib = NULL,  sort = TRUE, col = NULL, legend = NULL)
```

Arguments

`copa`	An object of class 'copa', resulting from a call to the `copa` function.
`idx`	A numeric vector listing the gene pairs to plot (e.g., idx = 1:3 will plot the first three gene pairs).
`lib`	If the underlying data are Affymetrix expression values, one can specify an annotation package and the plot labels will be extracted from the xxxSYMBOL environment. If `NULL`, the `row.names` of the gene expression matrix will be used.
`sort`	Boolean. Should the data be sorted before plotting? Defaults to `TRUE`.
`col`	A vector of color names or numbers to be used for coloring the different samples in the resulting barplot.
`legend`	A vector of terms describing the two sample types (e.g., 'Normal' and 'Tumor'). Defaults to `NULL`

Details

Note that this function will output all the gene pairs in the idx vector without pausing. This can be controlled by either setting par(ask = TRUE), or by redirecting the output to a file (using e.g., `pdf`, `ps`, etc.).

Value

This function is called solely for outputting plots. No values are returned.

Author(s)

James W. MacDonald

References

Tomlins, SA, et al. Recurrent fusion of TMPRSS2 and ETS transcription factor genes in prostate cancer. Science. 2005 Oct 28;310(5748):644-8.

Examples

```
if(interactive()){
library(Biobase)
data(sample.ExpressionSet)
cl <- abs(3 - as.numeric(pData(sample.ExpressionSet)[,2]))
tmp <- copa(sample.ExpressionSet, cl)
plotCopa(tmp, 1, col = c("red", "blue"))
}
```

---

pSum *Compute all pairwise sums*

Description

A function that computes all pairwise sums for a vector of numbers. This is an internal function and is not intended for use by end-users.

Usage

```
pSum(a)
```

Arguments

a	A numeric vector

Value

out	A square matrix (of dimension length(a) X length(a)) containing all pairwise sums.

Author(s)

James W. MacDonald

scatterPlotCopa *Create scatterplots of interesting gene pairs*

Description

This function allows one to create scatterplots of gene pairs that may be involved in recurrent gene fusion in cancer.

Usage

```
scatterPlotCopa(copa, idx, lib = NULL)
```

Arguments

copa	An object of class 'copa', resulting from a call to the copa function
idx	A numeric vector listing the gene pairs to plot (e.g., idx = 1:3 will plot the first three gene pairs).
lib	If the underlying data are Affymetrix expression values, one can specify an annotation package and the plot labels will be extracted from the xxxSYMBOL environment. If NULL, the row.names of the gene expression matrix will be used.

Details

Note that this function will output all the gene pairs in the idx vector without pausing. This can be controlled by either setting par(ask = TRUE), or by redirecting the output to a file (using e.g., pdf, ps, etc.).

Value

This function is called solely for outputting plots. No values are returned.

Author(s)

James W. MacDonald

References

Tomlins, SA, et al. Recurrent fusion of TMPRSS2 and ETS transcription factor genes in prostate cancer. Science. 2005 Oct 28;310(5748):644-8.

Examples

```
if(interactive()){
library(Biobase)
data(sample.ExpressionSet)
cl <- abs(3 - as.numeric(pData(sample.ExpressionSet)[,2]))
tmp <- copa(sample.ExpressionSet, cl)
scatterPlotCopa(tmp, 1)
}
```

---

summaryCopa *Create Summary Showing Top Gene Pairs*

Description

This function can be used to output a data.frame containing the ID and optionally the gene symbol for the top gene pairs, based on the number of outliers.

Usage

```
summaryCopa(copa, pairnum, lib = NULL)
```

Arguments

copa	An object of class 'copa', resulting from a call to the copa function.
pairnum	The maximum number of outlier pairs to be output. A table can be output first using tableCopa
lib	For Affymetrix data that have an annotation package, this can be specified and the table will then also contain the gene symbol

Value

The output from this function is a data.frame with the number of outliers, the manufacturer identifiers, and optionally, the gene symbol for the genes.

Author(s)

James W. MacDonald <jmacdon@u.washington.edu>

References

Tomlins, SA, et al. Recurrent fusion of TMPRSS2 and ETS transcription factor genes in prostate cancer. Science. 2005 Oct 28;310(5748):644-8.

Examples

```
if(interactive()){
library(Biobase)
data(sample.ExpressionSet)
cl <- abs(3 - as.numeric(pData(sample.ExpressionSet)[,2]))
tmp <- copa(sample.ExpressionSet, cl)
summaryCopa(tmp, 6)
}
```

---

tableCopa *Summarize copa results*

Description

This function will output a table showing the number of gene pairs at each number of outliers.

Usage

```
tableCopa(copa)
```

Arguments

copa	A 'copa' object, the result of a call to copa

Value

This function simply prints a table to the screen, useful for summarizing the output from a call to copa.

Author(s)

James W. MacDonald

Examples

```
library(Biobase)
data(sample.ExpressionSet)
cl <- abs(3 - as.numeric(pData(sample.ExpressionSet)[,2]))
tmp <- copa(sample.ExpressionSet, cl)
tableCopa(tmp)
```

Index

∗ **hplot**
    plotCopa, 8
    scatterPlotCopa, 10
∗ **internal**
    copaFilter, 4
    getans, 7
    perm.mat, 7
    pSum, 9
∗ **manip**
    copaPerm, 5
    summaryCopa, 11
    tableCopa, 12
∗ **package**
    copa-package, 2
∗ **univar**
    copa, 3

copa, 3
copa-package, 2
copaFilter, 4
copaFilter, data.frame-method *(*copaFilter*)*, 4
copaFilter, ExpressionSet-method *(*copaFilter*)*, 4
copaFilter, matrix-method *(*copaFilter*)*, 4
copaFilter-methods *(*copaFilter*)*, 4
copaPerm, 5

do.copaFilter *(*copaFilter*)*, 4

getans, 7

perm.mat, 7
plotCopa, 8
pSum, 9

scatterPlotCopa, 10
summaryCopa, 11

tableCopa, 12