Package 'safe'

November 6, 2026

Title Significance Analysis of Function and Expression

Version 3.53.1

Author William T. Barry

Description
SAFE is a resampling-based method for testing functional categories in gene expression experiments. SAFE can be applied to 2-sample and multi-class comparisons, or simple linear regressions. Other experimental designs can also be accommodated through user-defined functions.

Depends R (>= 2.4.0), AnnotationDbi, Biobase, methods, SparseM

Suggests GO.db, PFAM.db, reactome.db, hgu133a.db, breastCancerUPP, survival, foreach, doRNG, Rgraphviz, GOstats

Maintainer Ludwig Geistlinger <ludwig.geistlinger@gmail.com>

License GPL (>= 2)

biocViews DifferentialExpression, Pathways, GeneSetEnrichment, StatisticalMethod, Software

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

git_branch devel

git_last_commit cbf9c64

git_last_commit_date 2026-05-01

Repository Bioconductor 3.24

Date/Publication 2026-11-05

Contents

safe-package . . . 2
gene.results . . . 2
getCmatrix . . . 3
p53.stat . . . 5
safe . . . 6
SAFE-class . . . 8
safe-internal . . . 9
safe.toptable . . . 10
safedag . . . 10
safeplot . . . 11

Index **14**

safe-package *Significance Analysis of Function and Expression*

Description

SAFE is a resampling-based method for testing functional categories in gene expression experiments. SAFE can be applied to 2-sample and multi-class comparisons, or simple linear regressions. Other experimental designs can also be accommodated through user-defined functions.

Details

Package:	safe
Type:	Package
Version:	3.0
Date:	2012-09-14
License:	GPL (>= 2)
LazyLoad:	yes

For an overview of how to use the package, including the most important functions, please see the included vignette.

Author(s)

William T. Barry

Maintainer: William T. Barry <bbarry@jimmy.harvard.edu>

References

W. T. Barry, A. B. Nobel and F.A. Wright, 2005, *Significance Analysis of functional categories in gene expression studies: a structured permutation approach*, *Bioinformatics* **21**(9) 1943–1949.

See Also

safe,

---

gene.results *Gene-specific results from SAFE*

Description

Provides gene-specific local statistics and resampling-based p-values for every feature in the category of interest. Features are ordered by the degree and direction of differential expression.

Usage

```
gene.results(object, cat.name = NULL, error = "none",
             print.it = TRUE, gene.names = NULL)
```

Arguments

object	Object of class SAFE.
cat.name	Name of the category to be displayed. If omitted, the most significant category is displayed.
error	Specifies a non-resampling based method for adjusting the empirical p-values. A Bonferroni, ("FWER.Bonf"), Holm's step-up ("FWER.Holm"), and Benjamini-Hochberg step down ("FDR.BH") adjustment can be selected. By default ("none") no error rates are computed.
print.it	Logical determining whether results are printed to screen or returned as a list of data.frames for up- and down-regulated genes.
gene.names	Optional character vector of gene names to append to the SAFE output.

Author(s)

William T. Barry: <bbarry@jimmy.harvard.edu>

References

W. T. Barry, A. B. Nobel and F.A. Wright, 2005, *Significance Analysis of functional categories in gene expression studies: a structured permutation approach*, *Bioinformatics***21**(9) 1943–1949.

See also the vignette included with this package.

See Also

safe.

---

getCmatrix *Generation of a C matrix*

Description

This function will construct a matrix of indicator variables for category membership from keyword or gene-indexed lists. Size constraints, the option to prune identical categories, and a vector of present genes can be defined to filter categories and order genes. New to version 3.0.0, annotation can be provided so that each gene, instead of each feature, has equal weight in a category.

Usage

```
getCmatrix(keyword.list = NULL, gene.list = NULL,
           present.genes = NULL, min.size = 2, max.size = Inf,
           by.gene = FALSE, gene.names =  NULL, prefix = "",
           prune = FALSE,
           as.matrix = FALSE, GO.ont = NULL, ...)
```

Arguments

keyword.list	A list containing character vectors for each keyword that specify the gene members.
gene.list	A list containing character vectors for each gene that specify the annotated functional categories.
present.genes	An optional vector used to filter genes in the C matrix. Can be provided as an unordered character vector of gene names that match names(list), or as an ordered vector of presence (1) and absence (0) calls.
min.size	Optional minimum category size to be considered.
max.size	Optional maximum category size to be considered.
by.gene	Optional logical to build 'soft' categories at the gene level, instead of the feature level.
gene.names	Optional character vector of gene names for 'soft' categories.
prefix	Optional character string to preceed category names.
prune	Optional logical to remove duplicate categories.
as.matrix	Optional argument to specify a matrix is returned rather than a matrix.csr.
GO.ont	"CC", "BP", or "MF" specify which Gene Ontology.
...	Any extra arguments will be forwarded to the read.table function when category assignments are given as a file.

Details

Typical usages are

```
getCmatrix(keyword.list, present.genes)
getCmatrix(gene.list, present.genes)
```

Value

C.mat.csr	If as.matrix=F a sparse matrix is returned with the rows corresponding to the genes and columns are categories
row.names	Character vector of gene names
col.names	Character vector of category names
col.synonym	Pipe-delimited Character vector of matching categories when prune=T

Author(s)

William T. Barry: <bbarry@jimmy.harvard.edu>

References

W. T. Barry, A. B. Nobel and F.A. Wright, 2005, *Significance Analysis of functional categories in gene expression studies: a structured permutation approach*, *Bioinformatics* **21**(9) 1943-9.

See also the vignette included with this package.

See Also

safe, safeplot, getPImatrix.

Examples

```
if(interactive()){
 require(hgu133a.db)
 genes <- unlist(as.list(hgu133aSYMBOL))
 RS.list <- list(Genes21 = c("ACTB","RPLP0","MYBL2","BIRC5","BAG1",
                             "GUSB","CD68","BCL2","MMP11","AURKA",
                             "GSTM1","ESR1","TFRC","PGR","CTSL2",
                             "GRB7","ERBB2","MKI67","GAPDH","CCNB1",
                             "SCUBE2"),
                 Genes16 = c("MYBL2","BIRC5","BAG1","CD68","BCL2",
                             "MMP11","AURKA","GSTM1","ESR1","PGR","CTSL2",
                             "GRB7","ERBB2","MKI67","CCNB1","SCUBE2"))
 RS.list <- lapply(RS.list,function(x) return(names(genes[which( match(genes, x, nomatch = 0) > 0)])))
 C1 <- getCmatrix(keyword.list = RS.list)
}
```

`p53.stat` *p53 Mutation Status*

Description

This data set is included for use in the vignette and provides the p53 mutation status (p53+ = 1 and p53- = 0) for each of 251 samples in the Miller et al. breast cancer study data.

Usage

```
data(p53.stat)
```

Format

A data.frame with 2 columns, "samplename" and "p53", and 251 rows.

Source

NCBI's Gene Expression Omnibus, accession GSE3494

References

Miller, L.D., Smeds, J., George, J., Vega, V.B., Vergara, L., Ploner, A., Pawitan, Y.,Hall, P., Klaar, S., Liu, E.T., and Bergh, J. (2005) 'An expression signature for p53status in human breast cancer predicts mutation status, transcriptional effects, and patient survival', *Proc Natl Acad Sci U S A*, 102(38), 13550-13555.

safe *Significance Analysis of Function and Expression*

Description

Performs a significance analysis of function and expression (SAFE) for a gene expression experiment and a set of functional categories specified by the user. SAFE is a two-stage resampling-based method that can be applied to a 2-sample, paired, multi-class, simple linear and right-censored linear regression models. Other experimental designs can also be accommodated through user-defined functions.

Usage

```
safe(X.mat, y.vec, C.mat = NULL, Z.mat = NULL,
     method = "permutation", platform = NULL,
     annotate = NULL, min.size = 2, max.size = Inf,
     by.gene = FALSE, local = "default", global = "default",
     args.local = NULL, args.global = list(one.sided = FALSE),
     Pi.mat = NULL, error = "FDR.BH", parallel=FALSE, alpha = NA,
     epsilon = 10^(-10), print.it = TRUE, ...)
```

Arguments

X.mat	A matrix or data.frame of expression data of size m by n where each row corresponds to a gene feature and each column to a sample. Data should be properly normalized and cannot contain missing values.
y.vec	A numeric, integer or character vector of length n containing the response of interest. For examples of the acceptable forms y.vec can take, see the vignette.
C.mat	A matrix containing the gene category assignments. Each column represents a category and should be named accordingly. For each column, values of 1 (TRUE) and 0 (FALSE) indicate whether the features in the corresponding rows of X.mat are contained in the category. This can also be a list containing a sparse matrix and names as created by getCmatrix.
Z.mat	A data.frame of size n by p, with p covariates as numeric or factors.
method	Type of hypothesis test can be specified as "permutation", "bootstrap.t", and "bootstrap.q". "express" calls the dependent package safeExpress. See vignette for details.
platform	If C.mat is unspecified, a character string of a Bioconductor annotation package can be used to build gene categories. See vignette for details and examples.
annotate	If C.mat is unspecified, a character string to specify the type of gene categories to build from annotation packages. "GO.MF", "GO.BP", "GO.CC", and "GO.ALL" (default) specify one or all Gene Ontologies. "KEGG" specifies pathways, and "PFAM" homologous families from the respective sources.
min.size	Optional minimum category size in building C.mat.
max.size	Optional maximum category size in building C.mat.
by.gene	Logical argument (default = FALSE) specifying whether multiple features to a single gene should be down-weighted.

local	Specifies the gene-specific statistic from the following options: "t.Student", "t.Welch", and "t.paired", for 2-sample designs, "f.ANOVA" for 1-way ANOVAs, and "t.LM" for simple linear regressions. "default" will choose between "t.Student", "f.ANOVA", and "t.LM" based on the form of y.vec. User-defined local statistics can also be used; details are provided in the vignette.
global	Specifies the global statistic for a gene categories. By default, the Wilcoxon rank sum ("Wilcoxon") is used. Else, a Fisher's Exact test statistic ("Fisher"), a Pearson's chi-squared type statistic ("Pearson") or t-statistic for average difference ("AveDiff") is available. User-defined global statistics can also be implemented.
args.local	An optional list to be passed to user-defined local statistics that require additional arguments. By default args.local = NULL.
args.global	An optional list to be passed to global statistics that require additional arguments. For two-sided local statistics, args.global = list(one.sided=F) allows bi-directional differential expression to be considered.
Pi.mat	Either an integer, or a matrix or data.frame containing the permutations. See getPImatrix for the acceptable form of a matrix or data.frame. If Pi.mat is an integer, B, then safe will generate B resamples of X.mat.
error	Specifies the method for computing error rate estimates. By default, Benjamini-Hochberg step down ("FDR.BH") FDR estimates are computed. A Bonferroni ("FWER.Bonf") and Holm's step-up ("FWER.Holm") adjustment can also be specified. Under permutation, "FDR.YB" computes the Yekutieli-Benjamini FDR estimate, and "FWER.WY" computes the Westfall-Young FWER estimate. The user can also specify "none" if no error rates are desired.
parallel	Logical argument (default = FALSE) specifying whether hypothesis test of method should be conducted with parallel processing. Only compatible with error = "none", "FWER.Bonf", or FDR.BH. See vignette for details.
alpha	The threshold for significant results to return. By default, alpha will be 0.05 for nominal p-values (error = "none"), and 0.1 for adjusted p-values.
epsilon	Numeric argument sets the minimum difference for ranking local and global statistics, correcting a numerical precision issue when computing empirical p-values in small data sets (n < 15). The default value is 10^(-10).
print.it	Logical argument (default = TRUE) specifying whether to print progress updates to the log for permutation and bootstrap calculations.
...	Allows arguments from version 2.0 to be ignored.

Details

safe utilizes a general framework for testing differential expression across gene categories that allows it to be used in various experimental designs. Through structured resampling of the data, safe accounts for the unknown correlation among genes, and enables proper estimation of error rates when testing multiple categories. safe also provides statistics and empirical p-values for the gene-specific differential expression.

Value

The function returns an object of class SAFE. See help for SAFE-class for more details.

Author(s)

William T. Barry: <bbarry@jimmy.harvard.edu>

References

W. T. Barry, A. B. Nobel and F.A. Wright, 2005, *Significance Analysis of functional categories in gene expression studies: a structured permutation approach*, *Bioinformatics* **21**(9) 1943–1949.

See also the vignette included with this package.

See Also

safeplot, safe.toptable, gene.results, getCmatrix, getPImatrix.

Examples

```
## Simulate a dataset with 1000 genes and 20 arrays in a 2-sample design.
## The top 100 genes will be differentially expressed at varying levels

g.alt <- 100
g.null <- 900
n <- 20

data<-matrix(rnorm(n*(g.alt+g.null)),g.alt+g.null,n)
data[1:g.alt,1:(n/2)] <- data[1:g.alt,1:(n/2)] +
                         seq(2,2/g.alt,length=g.alt)
dimnames(data) <- list(c(paste("Alt",1:g.alt),
                         paste("Null",1:g.null)),
                       paste("Array",1:n))

## A treatment vector
trt <- rep(c("Trt","Ctr"),each=n/2)

## 2 alt. categories and 18 null categories of size 50

C.matrix <- kronecker(diag(20),rep(1,50))
dimnames(C.matrix) <- list(dimnames(data)[[1]],
    c(paste("TrueCat",1:2),paste("NullCat",1:18)))
dim(C.matrix)

results <- safe(data,trt,C.mat = C.matrix,Pi.mat = 100)
results

## SAFE-plot made for the first category
if (interactive()) {
safeplot(results,"TrueCat 1")
}
```

SAFE-class *Class SAFE*

Description

The class SAFE is the output from the function safe. It is also the input to the plotting function safeplot.

Slots

`local`: Object of class `"character"` indicating the local statistic used.

`local.stat`: Object of class `"numeric"` containing the (unsorted) observed local statistics for genes.

`local.pval`: Object of class `"numeric"` containing the (unsorted) empirical p-values for genes

`global`: Object of class `"character"` indicating the global statistic used.

`global.stat`: Object of class `"numeric"` containing the (unsorted) observed global statistics for categories.

`global.pval`: Object of class `"numeric"` containing the (unsorted) empirical p-values for categories.

`error`: Object of class `"character"` indicating the method used to estimate error rates across multiple comparisons.

`global.error`: Object of class `"numeric"` containing the (unsorted) error rates (adjusted p-values) for categories. If not computed, it will be set to `NA`.

`C.mat`: Object of class `"SparseM"` containing the category assignments.

`alpha`: Object of class `"numeric"` containing the alpha level used in returning significant results.

`method`: Object of class `"character"` indicating the resampling method used in `safe`.

Methods

show (safe.results): Summarizes the test results of significant categories.

[(safe.results): Returns a SAFE object for categories indicated by integer or character strings.

safeplot (safe.results): `safeplot` produces CDFs of the association of expression to phenotype in a category relative to its complement.

Author(s)

William T Barry: `<bbarry@jimmy.harvard.edu >`

See Also

`safe`, `safeplot`.

`safe-internal` *Internal SAFE functions*

Description

The following functions return the appropriate functions within a SAFE analysis and are intended for internal use only.

Details

These are not to be called by the user.

Author(s)

William T. Barry

safe.toptable *Category-specific results from SAFE*

Description

Prints annotated results from SAFE as a data.frame for categories with the strongest association to response/phenotype. This includes (a) category name; (b) category size; (c) global statistic; (d) nominal empirical p-values; (e) adjusted p-values; and (f) descriptions if available.

Usage

```
safe.toptable(safe, number = 10,
              pretty = TRUE, description = TRUE)
```

Arguments

safe	Object of class SAFE.
number	Number of categories to be printed. If omitted, the first 10 categories are reported.
pretty	Logical determining whether p-values smaller than 10^-4 are output in scientific notation, rather than as decimals. By default pretty = TRUE.
description	Logical determining whether category descriptions are appended to printed output. By default description = TRUE.

Author(s)

William T. Barry: <bbarry@jimmy.harvard.edu>

References

W. T. Barry, A. B. Nobel and F.A. Wright, 2005, *Significance Analysis of functional categories in gene expression studies: a structured permutation approach*, *Bioinformatics***21**(9) 1943–1949.

See also the vignette included with this package.

See Also

safe.

safedag *SAFE results displayed in Gene Ontology*

Description

SAFE results are displayed in a directed acyclic graph for the Gene Ontology under investigation. Category-wide significance is displayed by node color.

Usage

```
safedag(object = NULL, ontology = NULL, top = NULL,
        color.cutoffs = c(0.1, 0.01, 0.001), filter = 0,
        max.GOnames = 200)
```

Arguments

`object`	Object of class `SAFE`
`ontology`	Gene Ontology of interest. Character strings of "GO.CC", "GO.BP", and "GO.MF" accepted.
`top`	Optional character string giving the top category name from which to draw a subgraph of the tree
`color.cutoffs`	Numeric vector of length 3 for the cutoffs for coloring significant nodes. Nodes with unadjusted p-values less than `color.cutoff[3]` are drawn in blue; less than `color.cutoff[2]` are drawn in green; less than `color.cutoff[1]` are drawn in red.
`filter`	Optional integer (1,2,3) to only include branches that contain at least one node as significant as the respective color.cutoff.
`max.GOnames`	Maximum size of DAG to include category names as labels.

Details

DAG-plots are suggested as a means for visualizing the extent of differential expression in Gene Ontology categories. The relatedness of significant categories suggests whether similar or disparate biological findings are identified.

Author(s)

William T. Barry: <bbarry@jimmy.harvard.edu>

References

W. T. Barry, A. B. Nobel and F.A. Wright, 2005, *Significance Analysis of functional categories in gene expression studies: a structured permutation approach*, *Bioinformatics* **21**(9) 1943–1949.

See also the vignette included with this package.

See Also

`safe`.

`safeplot` *SAFE plot*

Description

A SAFE plot for a given category displays the empirical distribution function for the ranked (or unranked) local statistics of a given category.

Usage

```
safeplot(safe = NULL, cat.name = "", limits = NULL,
         c.vec = NULL, local.stats = NULL, gene.names = NULL,
         rank = TRUE, x.limits = NULL, c.thresh = 0,
         colors = NULL, x.ticks = NULL, t.cex = 1,
         p.val = NULL, cat.desc = NULL, title = "", ...)
```

Arguments

safe	Object of class SAFE
cat.name	Name of the category to be plotted. If omitted, the most significant category is plotted.
limits	Limits of the shaded region in the plot on the unranked scale
c.vec	Logical vector specifying membership to a gene category
local.stats	Numeric vector of local statistics
gene.names	Optional character vector to replace names(local.stats) in labels
rank	Logical to plotted raned (TRUE) or unranked (FALSE) local statistics on the x-axis
x.limits	Optional limits of the x-axis. By default will be range(local.stats)
c.thresh	Optional threshold for plotting tickmarks for a weighted ("soft") gene category
colors	Optional vector specificy colors for gene labels
x.ticks	Optional location of x-axis tick marks on the ranked scale
t.cex	Text size for gene labels
p.val	Optional numeric value of the category's empirical p-value
cat.desc	Optional character string as a sub-title beneath the category name
title	Optional title to the plot
...	Allows arguments from version 2.0 to be ignored

Details

SAFE-plots display the differential expression in a given category relative to the complementary set of genes. The empirical cumulative distribution is plotted for local statistics in the category, on either a ranked or unranked scale. Tick marks are drawn along the top of the graph to indicate each gene's positions, and labeled when sufficient space permits. In this manner, genes with the most extreme local statistics can be identified as contributing to the category's significance.

Typical usages are

```
safeplot(safe)
safeplot(safe, cat.name)
safeplot(c.vec, local.stats, p.val, limits)
```

Author(s)

William T. Barry: <bbarry@jimmy.harvard.edu>

References

W. T. Barry, A. B. Nobel and F.A. Wright, 2005, *Significance Analysis of functional categories in gene expression studies: a structured permutation approach*, *Bioinformatics* **21**(9) 1943–1949.

See also the vignette included with this package.

See Also

safe.

Examples

```
## Simulate a dataset with 1000 genes and 20 arrays in a 2-sample design.
## The top 100 genes will be differentially expressed at varying levels

g.alt <- 100
g.null <- 900
n <- 20

data<-matrix(rnorm(n*(g.alt+g.null)),g.alt+g.null,n)
data[1:g.alt,1:(n/2)] <- data[1:g.alt,1:(n/2)] +
                         seq(2,2/g.alt,length=g.alt)
dimnames(data) <- list(c(paste("Alt",1:g.alt),
                         paste("Null",1:g.null)),
                       paste("Array",1:n))

## A treatment vector
trt <- rep(c("Trt","Ctr"),each=n/2)

## 2 alt. categories and 18 null categories of size 50

C.matrix <- kronecker(diag(20),rep(1,50))
dimnames(C.matrix) <- list(dimnames(data)[[1]],
    c(paste("TrueCat",1:2),paste("NullCat",1:18)))
dim(C.matrix)

results <- safe(data,trt,C.mat = C.matrix,Pi.mat = 100)
results

## SAFE-plot made for the first category
if (interactive()) {
safeplot(results,"TrueCat 1")
}
```

Index

∗ **datasets**
p53.stat, 5
∗ **hplot**
safedag, 10
safeplot, 11
∗ **htest**
gene.results, 2
getCmatrix, 3
safe, 6
safe.toptable, 10
∗ **internal**
safe-internal, 9
∗ **methods**
SAFE-class, 8
∗ **package**
safe-package, 2
[,SAFE-method *(*SAFE-class*)*, 8

error.FDR.BH *(*safe-internal*)*, 9
error.FDR.YB *(*safe-internal*)*, 9
error.FWER.Bonf *(*safe-internal*)*, 9
error.FWER.Holm *(*safe-internal*)*, 9
error.FWER.WY *(*safe-internal*)*, 9

gene.results, 2, *8*
getCmatrix, 3, *8*
getCOXresiduals *(*safe-internal*)*, 9
getPIcomplete *(*safe-internal*)*, 9
getPImatrix, *4*, *8*
getPImatrix *(*safe-internal*)*, 9
getXYresiduals *(*safe-internal*)*, 9
global.AveDiff *(*safe-internal*)*, 9
global.Fisher *(*safe-internal*)*, 9
global.Kolmogorov *(*safe-internal*)*, 9
global.Pearson *(*safe-internal*)*, 9
global.Wilcoxon *(*safe-internal*)*, 9

local.f.ANOVA *(*safe-internal*)*, 9
local.t.LM *(*safe-internal*)*, 9
local.t.paired *(*safe-internal*)*, 9
local.t.Student *(*safe-internal*)*, 9
local.t.Welch *(*safe-internal*)*, 9

p53.stat, 5

safe, *2–4*, 6, *8–11*, *13*
SAFE-class, 8
safe-internal, 9
safe-package, 2
safe.toptable, *8*, 10
safedag, 10
safeplot, *4*, *8*, *9*, 11
show,SAFE-method *(*SAFE-class*)*, 8
sigfig *(*safe-internal*)*, 9