Package 'signeR'

November 26, 2024

Type Package

Title Empirical Bayesian approach to mutational signature discovery

Version 2.8.0

Author Rafael Rosales, Rodrigo Drummond, Renan Valieris, Alexandre Defelicibus, Israel Tojal da Silva

Maintainer Renan Valieris <renan.valieris@accamargo.org.br>

Description The signeR package provides an empirical Bayesian approach to mutational signature discovery. It is designed to analyze single nucleotide variation (SNV) counts in cancer genomes, but can also be applied to other features as well. Functionalities to characterize signatures or genome samples according to exposure patterns are also provided.

License GPL-3

Imports BiocGenerics, Biostrings, class, grDevices, GenomeInfoDb, GenomicRanges, IRanges, nloptr, methods, stats, utils, PMCMRplus, parallel, pvclust, ppclust, clue, survival, maxstat, survivalAnalysis, future, VGAM, MASS, kknn, glmnet, e1071, randomForest, ada, future.apply, ggplot2, pROC, pheatmap, RColorBrewer, listenv, reshape2, scales, survminer, dplyr, ggpubr, cowplot, tibble, readr, shiny, shinydashboard, shinycssloaders, shinyWidgets, bsplus, DT, magrittr, tidyr, BiocFileCache, proxy, rtracklayer, BSgenome

Depends R (>= 3.0.2), VariantAnnotation, NMF

LinkingTo Rcpp, RcppArmadillo (>= 0.7.100)

SystemRequirements C++11

URL https://github.com/TojalLab/signeR

LazyData true

NeedsCompilation yes

ByteCompile TRUE

biocViews GenomicVariation, SomaticMutation, StatisticalMethod, Visualization

Suggests knitr, BSgenome.Hsapiens.UCSC.hg19, BSgenome.Hsapiens.UCSC.hg38, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.2.3

Encoding UTF-8

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

git_branch RELEASE_3_20

git_last_commit 99cef7c

git_last_commit_date 2024-10-29

Repository Bioconductor 3.20

Date/Publication 2024-11-25

Contents

`signeR-package` *Empirical Bayesian approach to mutational signature discovery*

Description

The signeR package provides an empirical Bayesian approach to mutational signature discovery. It is designed to analyze single nucleotide variation (SNV) counts in cancer genomes, but can also be applied to other features as well. Functionalities to characterize signatures or genome samples according to exposure patterns are also provided.

Details

signeR package focuses on the characterization and analysis of mutational processes. Its functionalities can be divided into three steps. Firstly, it provides tools to process VCF files and generate matrices of SNV mutation counts and mutational opportunities, both divided according to a 3bp context (mutation site and its neighboring bases). Secondly, the main part of the package takes those matrices as input and applies a Bayesian approach to estimate the number of underlying signatures and their mutational profiles. Thirdly, the package provides tools to correlate the activities of those signatures with other relevant information, e.g. clinical data, to infer conclusions about the analyzed genome samples, which can be useful for clinical applications.

Author(s)

Rodrigo Drummond, Rafael Rosales, Renan Valieris, Israel Tojal da Silva

Maintainer: Renan Valieris <renan.valieris@accamargo.org.br>

References

This work has been submitted to Bioinformatics under the title "signeR: An empirical Bayesian approach to mutational signature discovery".

L. B. Alexandrov, S. Nik-Zainal, D. C. Wedge, P. J. Campbell, and M. R. Stratton. Deciphering Signatures of Mutational Processes Operative in Human Cancer. Cell Reports, 3(1):246-259, Jan. 2013. doi:10.1016/j.celrep.2012.12.008.

A. Fischer, C. J. Illingworth, P. J. Campbell, and V. Mustonen. EMu: probabilistic inference of mutational processes and their localization in the cancer genome. Genome biology, 14(4):R39, Apr. 2013. doi:10.1186/gb-2013-14-4-r39.

Examples

```
vignette(package="signeR")
```

---

cosmic_data *COSMIC Mutational Signatures*

Description

COSMIC Mutational Signatures Data Files (SBS) v3.2.

Usage

```
data("cosmic_data")
```

Format

A data frame with 96 observations on the following 75 variables.

`Substitution.Type` a character vector

`Trinucleotide` a character vector

`Somatic.Mutation.Type` a character vector

`SBS1` a numeric vector

SBS2 a numeric vector

SBS3 a numeric vector

SBS4 a numeric vector

SBS5 a numeric vector

SBS6 a numeric vector

SBS7a a numeric vector

SBS7b a numeric vector

SBS7c a numeric vector

SBS7d a numeric vector

SBS8 a numeric vector

SBS9 a numeric vector

SBS10a a numeric vector

SBS10b a numeric vector

SBS11 a numeric vector

SBS12 a numeric vector

SBS13 a numeric vector

SBS14 a numeric vector

SBS15 a numeric vector

SBS16 a numeric vector

SBS17a a numeric vector

SBS17b a numeric vector

SBS18 a numeric vector

SBS19 a numeric vector

SBS20 a numeric vector

SBS21 a numeric vector

SBS22 a numeric vector

SBS23 a numeric vector

SBS24 a numeric vector

SBS25 a numeric vector

SBS26 a numeric vector

SBS27 a numeric vector

SBS28 a numeric vector

SBS29 a numeric vector

SBS30 a numeric vector

SBS31 a numeric vector

SBS32 a numeric vector

SBS33 a numeric vector

SBS34 a numeric vector

SBS35 a numeric vector

SBS36 a numeric vector

`SBS37` a numeric vector

`SBS38` a numeric vector

`SBS39` a numeric vector

`SBS40` a numeric vector

`SBS41` a numeric vector

`SBS42` a numeric vector

`SBS43` a numeric vector

`SBS44` a numeric vector

`SBS45` a numeric vector

`SBS46` a numeric vector

`SBS47` a numeric vector

`SBS48` a numeric vector

`SBS49` a numeric vector

`SBS50` a numeric vector

`SBS51` a numeric vector

`SBS52` a numeric vector

`SBS53` a numeric vector

`SBS54` a numeric vector

`SBS55` a numeric vector

`SBS56` a numeric vector

`SBS57` a numeric vector

`SBS58` a numeric vector

`SBS59` a numeric vector

`SBS60` a numeric vector

`SBS84` a numeric vector

`SBS85` a numeric vector

`SBS86` a numeric vector

`SBS87` a numeric vector

`SBS88` a numeric vector

`SBS89` a numeric vector

`SBS90` a numeric vector

Source

https://cancer.sanger.ac.uk/signatures/documents/453/COSMIC_v3.2_SBS_GRCh38.txt

DiffExp *Differential Exposure Analysis*

Description

DiffExp : Identify signatures with significantly different activities among sample groups.

Usage

```
## S4 method for signature 'SignExp,character'
DiffExp(signexp_obj, labels, max_instances=200,
    method=kruskal.test, contrast="all", quant=0.5, cutoff=0.05,
    p.adj= "BH",plot_to_file=FALSE, file="Diffexp_boxplot.pdf",
    colored=TRUE, relative = FALSE, ...)
```

Arguments

signexp_obj	a SignExp object returned by signeR function.
labels	sample labels used to define sample groups.
max_instances	Maximum number of the exposure matrix instances to be analyzed. If the number of available E instances is bigger than this parameter, a subset of those will be randomly selected for analysis.
method	algorithm used to compare each signature exposure among sample groups. Default is kruskal.test, which leads to the use of Kruskal-Wallis Rank Sum Test.
contrast	defines which sample groups will be considered in the analysis. Default is "all", which leads the algorithm to evaluate the null hypothesis of exposure levels being constant in all groups. Instead, if this parameter contains a list of group labels, the algorithm will evaluate the null hypothesis of exposure levels being constant among those groups.
quant	the p-values quantile which, after log-transform, will be used as DES (Differential Exposure Score). Default is 0.5, which means the median log-transformed p-value will be considered as DES.
p.adj	correction method for p-values adjust at the post-hoc tests performed when there are more than two group labels. See p.adjust for options.
cutoff	threshold for p-values quantile for signatures to be considered as showing differential exposure.
plot_to_file	Whether to save the plot to the file parameter. Default is FALSE.
file	Output file to export p-values boxplot.
colored	Boolean variable, if TRUE boxplots of differentially exposed signatures will be colored in green, cutoff line will be colored in red and line segments showing the transformed p-value quantile used for DE evaluation will be colored in blue. Otherwise the plot will be black & white.
relative	Whether tests should be performed on absolute or relative signature contributions to each sample mutation. Default is FALSE (absolute contributions will be tested).
...	additional parameters for test algorithm defined by the method parameter.

Value

A list with the following items:

Pvquant	boolean array with one entry for each signature, indicating whether it shows differential exposure.
Pvalues	matrix containing all computed p-values, with one row for each signature.
MostExposed	for each differentially exposed signature, this array contains the label of the group where it showed higher levels of exposure. Contains NA for signatures not showing differential exposure.
Differences	List of matrices, exported only when there are more than two groups in the analysis and any signature is found to be differentially active. Each matrix corresponds to one of the highlighted signatures and show the results of comparisons among groups, with the significant ones marked as TRUE.

Examples

```
# assuming signatures is the return value of signeR()

# labels vector, one for each sample
my_labels <- c("a","a","b","b")

diff_exposure <- DiffExp(signatures$SignExposures,labels=my_labels)

# see also
vignette(package="signeR")
```

ExposureClassify *Classify samples by exposure levels*

Description

Assign unlabeled samples to previously defined groups.

Usage

```
## S4 method for signature 'SignExp,character'
ExposureClassify(signexp_obj, labels,
    method="knn", max_instances=200, k=3, weights=NA, plot_to_file=FALSE,
   file="Classification_barplot.pdf", colors=NA_character_, min_agree=0.75,...)
```

Arguments

signexp_obj	A SignExp object returned by signeR function.
labels	Sample labels. Every sample labeled as NA will be classified according to its mutational profile and the profiles of labeled samples.
method	Classification algorithm used. Default is k-Nearest Neighbors (kNN). Any other algorithm may be used, as long as it is customized to satisfy the following conditions: Input: a matrix of labeled samples, with one sample per line and one feature per column; a matrix of unlabeled samples to classify, with the same structure; an array of labels, with one entry for each labeled sample. Output: an array of assigned labels, one for each unlabeled sample.

max_instances	Maximum number of the exposure matrix instances to be analyzed. If the number of available E instances is bigger than this parameter, a subset of those will be randomly selected for analysis.
k	Number of nearest neighbors considered for classification, used only if method="kNN". Default is 3.
weights	Vector of weights applied to the signatures when performing classification. Default is NA, which leads all the signatures to have weight=1.
plot_to_file	Whether to save the plot to the file parameter. Default is FALSE.
file	File that will be generated with classification graphic output.
colors	Array of color names, one for each sample class. Colors will be recycled if the length of this array is less than the number of classes.
min_agree	Minimum frequency of agreement among individual classifications. Samples showing a frequency of agreement below this value are considered as "undefined". Default is 0.75.
...	additional parameters for classification algorithm (defined by "method" above).

Value

A list with the following items:

class	The assigned classes for each unlabeled sample.
freq	Classification agreement for each unlabeled sample: the relative frequency of assignment of each sample to the group specified in "class".
‘	
allfreqs	Matrix with one column for each unlabeled sample and one row for each class label. Contains the assignment frequencies of each sample to each class.
probs	As above, a matrix with unlabeled samples in columns and class labels in rows. Contains the average probability, among repeated exposure classifications, of each sample belonging to each class.

Examples

```
# assuming signatures is the return value of signeR()


my_labels <- c("a","a","a","a",NA,"b","b","b","b",NA)
Class <- ExposureClassify(signatures$SignExposures, labels=my_labels)

# see also
vignette(package="signeR")
```

ExposureClassifyCV *k-fold cross-validation of sample classifcation by exposure levels*

Description

Splits labeled samples in k groups (deafult k=8), keeping the proportion of classes stable among groups. Classify samples in each group according to the k-1 remaining ones. Gather results and evaluate global classification performance.

Usage

```
## S4 method for signature 'SignExp,character'
ExposureClassifyCV(signexp_obj, labels, method="knn",
    max_instances=200, k=3, weights=NA, plot_to_file=FALSE,
    file="Classification_CV_barplot.pdf", colors=NA_character_,
    min_agree=0.75, fold=8, ...)
```

Arguments

signexp_obj	A SignExp object returned by signeR function.
labels	Sample labels. Unlabeled samples (NA labels) will be ignored.
method	Classification algorithm used. Default is k-Nearest Neighbors (kNN). Any other algorithm may be used, as long as it is customized to satisfy the following conditions: Input: a matrix of labeled samples, with one sample per line and one feature per column; a matrix of unlabeled samples to classify, with the same structure; an array of labels, with one entry for each labeled sample. Output: an array of assigned labels, one for each unlabeled sample.
max_instances	Maximum number of the exposure matrix instances to be analyzed. If the number of available E instances is bigger than this parameter, a subset of those will be randomly selected for analysis.
k	Number of nearest neighbors considered for classification, used only if method="kNN". Default is 3.
weights	Vector of weights applied to the signatures when performing classification. Default is NA, which leads all the signatures to have weight=1.
plot_to_file	Whether to save the plot to the file parameter. Default is FALSE.
file	File that will be generated with cross validation graphic output.
colors	Array of color names, one for each sample class. Colors will be recycled if the length of this array is less than the number of classes.
min_agree	Minimum frequency of agreement among individual classifications. Samples showing a frequency of agreement below this value are considered as "undefined". Default is 0.75.
fold	Number of subsets in which labeled samples will be split
...	additional parameters for classification algorithm (defined by "method" above).

Value

A list with the following items:

`confusion_matrix`
: Contingency table of attributed sample classes against original labels.

`class`
: The assigned classes for each sample.

`freq`
: Classification agreement for each sample: the relative frequency of assignment of each sample to the group specified in "class".

`allfreqs`
: Matrix with one column for each sample and one row for each class label. Contains the assignment frequencies of each sample to each class.

`probs`
: As above, a matrix with samples in columns and class labels in rows. Contains the average probability, among repeated exposure classifications, of each sample belonging to each class.

Examples

```
# assuming signatures is the return value of signeR()


my_labels <- c("a","a","a","a","a","b","b","b","b","b")
ClassCV <- ExposureClassifyCV(signatures$SignExposures, labels=my_labels,fold=5)

# see also
vignette(package="signeR")
```

`ExposureCorrelation` *Exposure correlation analysis (given a known sample feature)*

Description

ExposureCorrelation : Identify signatures which are significantly correlated with a provided (numeric) sample feature.

Usage

```
## S4 method for signature 'SignExp,numeric'
ExposureCorrelation(Exposures, feature,
    method="spearman", max_instances=200, cutoff_pvalue=0.05, quant=0.5,
    plot_to_file=FALSE, file="ExposureCorrelation_plot.pdf",
    colors=TRUE,...)
```

Arguments

`Exposures`
: a SignExp object returned by signeR function or a matrix of exposures (with signatures in rows and a column for each sample).

`feature`
: numeric feature associated with each sample, such as age, weight or the expression of a gene.

`method`
: a character string indicating which correlation coefficient should be used for the test. Options are "pearson", "kendall", or "spearman" (default).

max_instances	Maximum number of the exposure matrix instances to be analyzed. If the number of available E instances is bigger than this parameter, a subset of those will be randomly selected for analysis.
cutoff_pvalue	threshold for p-values quantile for signatures to be considered as showing significant correlation.
quant	the p-values quantile which, after log-transform, will be used for selecting significantly correlated signatures. Default is 0.5, which means the median p-value will be considered.
plot_to_file	Whether to save the plot to the file parameter. Default is FALSE.
file	Output file to export p-values boxplot and scatterplots showing the correlations of exposures and the provided feature.
colors	Boolean variable, if TRUE p-values boxplots of significantly correlated signatures will be colored in green, cutoff line will be colored in red and line segments showing the transformed p-value quantile used for significance evaluation will be colored in blue. Otherwise the plot will be black & white.
...	additional parameters for test algorithm defined by the method parameter.

Value

A list with the following items:

Significance	boolean array with one entry for each signature, indicating whether it shows significant correlation with the provided feature.
Correlation_quantiles	vector of correlation quantiles, with one entry for each signature.
Pvalues_quantiles	vector of p-values quantiles used for significance evaluation.
Correlations	matrix containing all computed correlations, with one row for each signature.
Pvalues	matrix containing all computed p-values, with one row for each signature.

Examples

```
# assuming signatures is the return value of signeR()

# feature vector, with one value for each sample
my_feature <- rnorm(30,100,20)+signatures$SignExposures@Exp[1,,1]

Exp_corr <- ExposureCorrelation(signatures$SignExposures,feature=my_feature)

# see also
vignette(package="signeR")
```

ExposureGLM *Exposure Generalized Linear Model*

Description

Fits a GLM to exposure data, with a given sample feature as the target of the model.

Usage

```
## S4 method for signature 'SignExp,numeric'
ExposureGLM(Exposures, feature, max_instances=200,
     cutoff_pvalue=0.05,quant=0.5, plot_to_file=FALSE,
     file="ExposureGLM_plot.pdf",colors=TRUE,...)
```

Arguments

Exposures	A SignExp object returned by signeR function or a matrix of exposures (with signatures in rows and a column for each sample).
feature	numeric feature associated with each sample, such as age, weight or the expression of a gene.
max_instances	Maximum number of the exposure matrix instances to be analyzed. If the number of available E instances is bigger than this parameter, a subset of those will be randomly selected for analysis.
cutoff_pvalue	threshold for p-values quantile for signatures to be considered as significant on the model.
quant	p-values quantile used to evaluate if signatures are significant. Default is 0.5, meaning that median p-values are adopted.
plot_to_file	Whether to save plots to the file parameter. Default is FALSE.
file	Output file to export p-values boxplot and scatterplots showing the correlations of exposures and the provided feature.
colors	Boolean variable, if TRUE p-values boxplots of significantly correlated signatures will be colored in green, cutoff line will be colored in red and line segments showing the transformed p-value quantile used for significance evaluation will be colored in blue. Otherwise the plot will be black & white.
...	additional parameters for test algorithm defined by the method parameter.

Value

A list with the following items:

Significance	boolean array with one entry for each signature, indicating whether it shows a significant contribution to the model.
Stats	matrix of model statistics, with one line for each signature.
Pvalues	vector of p-values used for significance evaluation.

Examples

```
# assuming signatures is the return value of signeR()


my_feature <- rnorm(30,100,20)+signatures$SignExposures@Exp[1,,1]
EGlm <- ExposureGLM(signatures$SignExposures, feature=my_feature)

# see also
vignette(package="signeR")
```

ExposureSurvival *Exposure survival analysis*

Description

ExposureSurvival: Given survival data, identify signatures that are significantly related to differences in hazards.

Usage

```
## S4 method for signature 'SignExp,Surv'
ExposureSurvival(signexp_obj, surv, max_instances=200,
    method=logrank, quant=0.5, cutoff_pvalue=0.05, cutoff_hr=NA,
    plot_to_file=FALSE, file="ExposureSurvival_plot.pdf",
    colors=TRUE, ...)
```

Arguments

signexp_obj	a SignExp object returned by signeR function.
surv	a Surv object from package survival or a matrix with columns "time" and "status" (the last indicates whether 1:an event occurred or 0:there was a loss of follow up).
max_instances	Maximum number of the exposure matrix instances to be analyzed. If the number of available E instances is bigger than this parameter, a subset of those will be randomly selected for analysis.
method	a character string indicating which approach should be used for the test. Options are "logrank" (default) or "cox" (fit a Cox proportional hazards model to data).
quant	the quantile of p-values and hazard ratios which will be used for selecting survival significant signatures. Default is 0.5, which means the median p-value and hazard ratio will be considered.
cutoff_pvalue	threshold for p-values quantile for signatures to be considered as significant.
cutoff_hr	threshold for hazard ratio quantile for signatures to be considered as significant.
plot_to_file	Whether to save the plot to the file parameter. Default is FALSE.
file	Output file to export p-values boxplots and Kaplan-Meier curves.
colors	Boolean variable, if TRUE p-values boxplots of significant signatures will be colored in green, cutoff line will be colored in red and line segments showing the transformed p-value quantile used for significance evaluation will be colored in blue. Otherwise the plot will be black & white.
...	additional parameters for test algorithm defined by the method parameter.

Value

A list with the following items:

Significance	boolean array with one entry for each signature, indicating whether its levels of exposure are significant to survival.
Correlation_quantiles	vector of correlation quantiles, with one entry for each signature.

pvalues	vector of p-values used for significance evaluation.
limits	vector containing one cut value for the exposures of each signature, such that splitting the samples according to this value leads to maximal differences in survival among generated groups.
Groups	matrix containing one line for each signature, defining a division of the samples into two groups according to their exposures, such that survival differences between the groups are maximal.

Examples

```
# assuming signatures is the return value of signeR()

# feature vector, with one value for each sample
library(survival)
my_surv <- Surv(rnorm(30,730,100),sample(c(0:1),30,replace=TRUE))

Exp_corr <- ExposureSurvival(signatures$SignExposures, surv = my_surv)

# see also
vignette(package="signeR")
```

ExposureSurvModel *Exposure Cox model*

Description

ExposureSurvModel: Given survival data, fits a multivariate Cox proportional hazards model to exposure data.

Usage

```
## S4 method for signature 'SignExp,Surv'
ExposureSurvModel(Exposures, surv, addata,
    max_instances=200, quant=0.5, cutoff_pvalue=0.05, cutoff_hr=NA,
    plot_to_file=FALSE, file="ExposureSurvival_plot.pdf", colors=TRUE, ...)
```

Arguments

Exposures	A SignExp object returned by signeR function or a matrix of exposures (with signatures in rows and a column for each sample).
surv	a Surv object from package survival or a matrix with columns "time" and "status" (the last indicates whether 1:an event occurred or 0:there was a loss of follow up).
addata	a data frame with additional data (one sample per row) that will be used in the Cox model along with exposure data.
max_instances	Maximum number of the exposure matrix instances to be analyzed. If the number of available E instances is bigger than this parameter, a subset of those will be randomly selected for analysis.
quant	the quantile of p-values and hazard ratios which will be used for selecting survival significant signatures. Default is 0.5, which means the median p-value and hazard ratio will be considered.

cutoff_pvalue	threshold for p-values quantile for signatures to be considered as significant.
cutoff_hr	threshold for hazard ratio quantile for signatures to be considered as significant.
plot_to_file	Whether to save the plot to the file parameter. Default is FALSE.
file	Output file to export p-values boxplots and Kaplan-Meier curves.
colors	Boolean variable, if TRUE p-values boxplots of significant signatures will be colored in green, cutoff line will be colored in red and line segments showing the transformed p-value quantile used for significance evaluation will be colored in blue. Otherwise the plot will be black & white.
...	additional parameters for test algorithm defined by the method parameter.

Value

A list with the following items:

Significance	boolean array with one entry for each signature, indicating whether its levels of exposure are significant to survival.
Stats	data frame containing hazard ratios and pvalues for signatures (one per line) on fitted Cox models.

Examples

```
# assuming signatures is the return value of signeR()

# feature vector, with one value for each sample
library(survival)
my_surv <- Surv(rnorm(30,730,100),sample(c(0:1), 30, replace = TRUE))

Exp_corr <- ExposureSurvModel(signatures$SignExposures, surv = my_surv)

# see also
vignette(package="signeR")
```

FuzzyClustExp *Fuzzy Clustering of exposure data*

Description

FuzzyClustExp : Performs fuzzy C-means clustering of samples, based on exposures. The number of clusters is defined by optimizing the PBMF index of obtained clustering.

Usage

```
## S4 method for signature 'SignExp,numeric'
FuzzyClustExp(signexp_obj, max_instances=200, Clim,
                method.dist="euclidean", method.clust="fcm", relative=FALSE,
                m=2, plot_to_file=FALSE, file="FuzzyClustExp.pdf",colored=TRUE)
```

Arguments

signexp_obj	a SignExp object returned by signeR function.
max_instances	Maximum number of the exposure matrix instances to be analyzed. If the number of available E instances is bigger than this parameter, a subset of those will be randomly selected for analysis.
Clim	number of groups range, a vector with minimum and maximum accepted number of groups. The algorithm will maximize the PBMF-index within this range.
method.dist	used distance metric
method.clust	clustering method. Either "fcm", default, for fuzzy C-means or "km" for k-means.
relative	Whether to normalize exposures of each sample so that they sum up to one. Default is FALSE, thus clustering samples by the absolute contributions of signatures to mutation counts. Otherwise, clustering will be based on relative contributions.
m	Expoent used in PBMF-index
plot_to_file	Whether to save a heatmap of results to the file parameter. Default is FALSE.
file	Output file to export a heatmap with the levels of pertinence of samples to found groups.
colored	Whether plots will be in color or B&W. Default is TRUE.

Value

A list with the following items: Meanfuzzy=Meanfuzzy, AllFuzzy=Fuzzy[[1]], Centroids=Fuzzy[[2]]

Meanfuzzy	Final clustering: mean levels of pertinence of samples to found groups.
AllFuzzy	All levels of pertinence of samples to found groups in repeated runs of the clustering algorithm.
Centroids	All centroids of found groups in repeated runs of the clustering algorithm.

Examples

```
# assuming signatures is the return value of signeR()

# Limits to number of groups:
cl <- c(2,4)

FuzClust <- FuzzyClustExp(signatures$SignExposures, Clim = cl)

# see also
vignette(package="signeR")
```

generateMatrix *count matrix and opportunity matrix generators*

Description

genCountMatrixFromVcf : generate a count matrix from a VCF file.
genCountMatrixFromMAF : generate a count matrix from an MAF file.
genOpportunityFromGenome : generate an opportunity matrix from a target regions set.

Usage

```
genCountMatrixFromVcf(bsgenome, vcfobj)
genCountMatrixFromMAF(bsgenome, maf_file)
genOpportunityFromGenome(bsgenome, target_regions, nsamples=1)
```

Arguments

bsgenome	A BSgenome object, equivalent to the genome used for the variant call.
vcfobj	A VCF object. See VCF-class from the VariantAnnotation package.
maf_file	Path to a MAF file.
target_regions	A GRanges object, describing the target region analyzed by the variant caller.
nsamples	Number of samples to generate the matrix, should be the same number as rows of the count matrix.

Value

A matrix of samples x (96 features).
Each feature is an SNV change with a 3bp context.

Examples

```
library(rtracklayer)
library(VariantAnnotation)

# input files, variant call and target
vcf_file <- system.file("extdata","example.vcf", package="signeR")
bed_file <- system.file("extdata","example.bed", package="signeR")
maf_file <- system.file("extdata","example.maf", package="signeR")

# BSgenome, will depend on your variant call
library(BSgenome.Hsapiens.UCSC.hg19)

vcfobj <- readVcf(vcf_file, "hg19")
mut <- genCountMatrixFromVcf(BSgenome.Hsapiens.UCSC.hg19, vcfobj)

target_regions <- import(con=bed_file, format="bed")
opp <- genOpportunityFromGenome(BSgenome.Hsapiens.UCSC.hg19,
    target_regions, nsamples=nrow(mut))

mut <- genCountMatrixFromMAF(BSgenome.Hsapiens.UCSC.hg19, maf_file)

# see also
vignette(package="signeR")
```

HClustExp *Hierarchical Clustering of exposure data*

Description

HClustExp: Performs hierarchical clustering of samples, based on exposures.

Usage

```
## S4 method for signature 'SignExp,numeric'
HClustExp(signexp_obj, Med_exp=NA,
        max_instances=200, method.dist="euclidean", method.hclust="average",
        use.cor=FALSE, relative=FALSE, plot_to_file=FALSE,
        file="HClustExp_dendrogram.pdf", colored=TRUE)
```

Arguments

signexp_obj	a SignExp object returned by signeR function.
Med_exp	optional matrix with (median) exposures.
max_instances	Maximum number of the exposure matrix instances to be analyzed. If the number of available E instances is bigger than this parameter, a subset of those will be randomly selected for analysis.
method.dist	used distance metric
method.hclust	clustering method.
use.cor	used in pv.distance
relative	Whether to normalize exposures of each sample so that they sum up to one. Default is FALSE, thus clustering samples by the absolute contributions of signatures to mutation counts. Otherwise, clustering will be based on relative contributions.
plot_to_file	Whether to save a heatmap of results to the file parameter. Default is FALSE.
file	Output file to export a heatmap with the levels of pertinence of samples to found groups.
colored	Whether plots will be in color or B&W. Default is TRUE.

Value

A pvclust object, as described in package pvclust.

Examples

```
# assuming signatures is the return value of signeR()


HClust <- HClustExp(signatures$SignExposures)

# see also
vignette(package="signeR")
```

methods *SignExp class methods*

Description

setSamples: Define sample names for a SignExp object, according to the "names" argument.

setMutations: Define mutation names for a SignExp object, according to the "mutations" argument.

Normalize: Normalize a SignExp object so that the entries of each signature sum up to one.

Reorder_signatures: Change the order of the signatures in a SignExp object. The new signature order will be defined by the "ord" argument.

Reorder_samples: Change samples order, according to ord parameter.

Reorder_mutations: Change mutations order, according to ord parameter.

Average_sign: Exports an approximation of the signatures obtained by the averages of the samples for the signature matrix P.

Median_sign: Exports an approximation of the signatures obtained by the medians of the samples for signature matrix P.

Average_exp: Exports an approximation of the exposures obtained by the averages of the samples for exposure matrix E.

Median_exp: Exports an approximation of the exposures obtained by the medians of the samples for exposure matrix E.

Usage

```
## S4 method for signature 'SignExp'
setSamples(signexp_obj, names)
## S4 method for signature 'SignExp'
setMutations(signexp_obj, mutations)
## S4 method for signature 'SignExp'
Normalize(signexp_obj)
## S4 method for signature 'SignExp,numeric'
Reorder_signatures(signexp_obj, ord)
## S4 method for signature 'SignExp,numeric'
Reorder_samples(signexp_obj, ord)
## S4 method for signature 'SignExp,numeric'
Reorder_mutations(signexp_obj, ord)
## S4 method for signature 'SignExp'
Average_sign(signexp_obj, normalize=TRUE)
## S4 method for signature 'SignExp'
Median_sign(signexp_obj, normalize=TRUE)
```

```
## S4 method for signature 'SignExp'
Average_exp(signexp_obj, normalize=TRUE)
## S4 method for signature 'SignExp'
Median_exp(signexp_obj, normalize=TRUE)
```

Arguments

signexp_obj	a SignExp object returned by signeR function. e.g.: sig$SignExposures
names	Vector of sample names.
mutations	Vector of mutations, e.g. "C>A:TCG".
normalize	Whether the signatures should be normalized before extracting approximations. Default is TRUE.
ord	Vector with the new signature order.

Value

setSamples, setMutations, Normalize and Reorder_* returns a modified SignExp object.
Average_sign, Median_sign, Average_exp and Median_exp return a matrix with the corresponding approximation.

Examples

```
# each function needs the SignExposures object
# which is part of the result of the signeR() call
signexp  <- Normalize(signatures$SignExposures)
signexp  <- Reorder_signatures(signatures$SignExposures,ord=c(2,1))
matrix_p <- Median_sign(signatures$SignExposures)
# etc ...

# see also
vignette(package="signeR")
```

---

plots *signeR plot functions*

Description

BICboxplot: Plot the measured values of the Bayesian Information Criterion (BICs) for tested model dimensions.

Paths: Plot the convergence of the Gibbs sampler for signatures and exposures on separate charts.

SignPlot: Plot the mutational signatures in a bar chart, with error bars according to the variation of individual entries along the generated Gibbs samples.

SignHeat: Plot the mutation signatures in a heatmap.

ExposureBarplot: Barplot of estimated exposure values, showing the contribution of the signatures to the mutation counts of each genome sample.

ExposureBoxplot: Boxplot of exposure values, showing their variation along the generated Gibbs samples.

ExposureHeat: Plot a heatmap of the exposures, along with a dendrogram of the samples grouped by exposure levels.

Usage

```
BICboxplot(signeRout, plot_to_file=FALSE, file="Model_selection_BICs.pdf")
## S4 method for signature 'SignExp'
Paths(signexp_obj, plot_to_file=FALSE,
    file_suffix="plot.pdf", plots_per_page=4, ...)
## S4 method for signature 'SignExp'
SignPlot(signexp_obj, plot_to_file=FALSE,
    file="Signature_plot.pdf", pal="bcr1", threshold=0, plots_per_page=4,
    gap=1, reord=NA, ...)
## S4 method for signature 'SignExp'
SignHeat(signexp_obj, plot_to_file=FALSE,
    file="Signature_heatmap.pdf", nbins=50, pal="roh", ...)
## S4 method for signature 'SignExp'
ExposureBarplot(signexp_obj, plot_to_file=FALSE,
    file="Exposure_barplot.pdf", col='tan2', threshold=0, relative=FALSE,
    title="", show_samples=NA, ...)
## S4 method for signature 'SignExp'
ExposureBoxplot(signexp_obj, plot_to_file=FALSE,
    file="Exposure_boxplot.pdf", col='tan2', threshold=0, show_samples=NA,
    plots_per_page=4, reord=NA, ...)
## S4 method for signature 'SignExp'
ExposureHeat(signexp_obj, plot_to_file=FALSE,
    file="Exposure_heatmap.pdf", nbins=50, pal="roh", distmethod="euclidean",
        clustermethod="complete", show_samples=NA, ...)
```

Arguments

signexp_obj	A SignExp object returned by signeR function. e.g.: sig$SignExposures
signeRout	The list returned by the signeR function.
plot_to_file	Whether to save the plot to the file parameter. Default is FALSE.
file	Output pdf file of the plots.
pal	Color palette used. Options are: "brew","lba","bcr1", "bcr2","bw","rdh","roh","blh" or "bph".
threshold	Entries below this value will be rounded to 0. Default is 0 (all entries are kept).
plots_per_page	How many plots in a single page, default is 4.
gap	Distance between consecutive bars on the plot.
reord	Order of signatures for plotting. Should be a permutation of 1:nsig, where nsig is the number of signatures. By default, signatures are ordered by the total exposure, in decreasing order.

nbins	The range of signature entries is divided into this number of bins for plotting, each bin corresponding to a different color.
file_suffix	The suffix of the output file.
col	Single color name for boxplots.
distmethod	Distance measure used for grouping samples. Default is "euclidean", see the documentation of the dist function for other options.
clustermethod	Agglomeration method used for grouping samples. Default is "complete", see the documentation of the hclust function for other options.
relative	Whether to normalize exposures of each sample so that they sum up to one. Default is FALSE, thus generating a plot of absolute contributions of signatures to mutation counts. Otherwise, relative contributions will be displayed.
title	Main title added to the plot. Default is no title.
show_samples	Whether sample names will be shown in the plot. Default is NA, which leads to sample names being displayed only when there are less than 30 samples. However, even if show_samples=TRUE, due to display limitations sample names are not shown if there are more than 50 samples.
...	.

Value

The plot result is exported to the current graphic device. If plot_to_file=TRUE, the plot is saved in the file defined by the file argument.

Examples

```
# each plot function needs the SignExposures object
# which is part of the result of the signeR() call
SignPlot(signatures$SignExposures)
Paths(signatures$SignExposures)
# etc ...

# BICboxplot needs the returned list itself
BICboxplot(signatures)

# see also
vignette(package="signeR")
```

---

signeR *signeR*

Description

Generates the signatures.

Usage

```
signeR(M, Mheader = TRUE, samples = "rows", Opport = NA,
    Oppheader = FALSE,  P = NA, fixedP = FALSE,
    nsig = NA, nlim = c(NA, NA),
    try_all = FALSE, BICsignificance = FALSE, critical_p = 0.05,
    ap = NA, bp = NA, ae = NA, be = NA,
    lp = NA, le = NA, var.ap = 10, var.ae = 10,
    start = "lee", testing_burn = 1000, testing_eval = 1000,
    main_burn = 10000, main_eval = 2000,
    estimate_hyper = FALSE, EMit_lim=100, EM_eval = 100,
    parallelization = "multisession")
```

Arguments

M	mutation counts matrix of samples x features.
Mheader	if M has colnames defined use TRUE, if FALSE a default order will be assumed.
samples	if the samples are row-wise or column-wise in M, default is "row".
Opport	context count matrix of samples x features in the target genome or region.
Oppheader	if Opport has header defined.
P	Previously known matrix of signatures. If provided, can be fixed along algorithm iterations or just used as an initial value (see next parameter)
fixedP	If TRUE, provided P matrix will be fixed along iterations.
nsig	number of signatures, which can be provided or estimated by the algorithm.
nlim	define an interval to search for the optimal number of signatures.
try_all	if TRUE, all possible values for nsig will be tested
BICsignificance	if TRUE, BICs will be considered different only if their distribution is significantly different. In case of ties in BICs comparison, signer will adopt the model with fewer signatures.
critical_p	level of significance for BICs distribution to be considered different
ap	shape parameter of the gamma distribution used to generate the entries of a matrix of rate parameters of the gamma distributions which generate signatures.
bp	rate parameter of the gamma distribution used to generate the entries of a matrix of rate parameters of the gamma distributions which generate signatures.
ae	shape parameter of the gamma distribution used to generate the entries of a matrix of rate parameters of the gamma distributions which generate exposures.
be	rate parameter of the gamma distribution used to generate the entries of a matrix of rate parameters of the gamma distributions which generate exposures.
lp	parameter of the exponential distribution used to generate the entries of a matrix of shape parameters of the gamma distributions which generate signatures.
le	parameter of the exponential distribution used to generate the entries of a matrix of shape parameters of the gamma distributions which generate exposures.
var.ap	variance of the gamma distribution used to generate proposals for shape parameters of signatures
var.ae	variance of the gamma distribution used to generate proposals for shape parameters of exposures

start	NMF algorithm used to generate initial values for signatures and exposures,options: "brunet","KL","lee","Frobenius","offset","nsNMF", "ls-nmf","pe-nmf","siNMF","snmf/r" or "snmf/l".
testing_burn	number of burning iterations of the Gibbs sampler used to estimate the number of signatures in data. Corresponds to R0 at Algorithm 1 on signeR paper.
testing_eval	number of iterations of the Gibbs sampler used to estimate the number of signatures in data. Corresponds to R2 at Algorithm 1 on signeR paper.
EM_eval	number of samples generated at each iteration of the EM algorithm. Corresponds to R1 at Algorithm 1 on signeR paper.
main_burn	number of burning iterations of the final Gibbs sampler.
main_eval	number of iterations of the final Gibbs sampler.
estimate_hyper	if TRUE, algorithm estimates optimal values of ap,bp,ae,be,lp,le. Start values can still be provided.
EMit_lim	limit of EM iterations for the estimation of hyper-hyperparameters ap,bp,ae,be,lp,le. Default is 100. Corresponds to U at Algorithm 1 on signeR paper.
parallelization	strategy of computation parallelization, see future::plan help

Value

signeR output is a list with the following items:

Nsign	selected number of signatures.
tested_n	array containing the numbers of signatures tested by the algorithm.
Test_BICs	list of measured BIC values when testing different numbers of signatures.
Phat	Estimated signatures, median of P samples.
Ehat	Estimated exposures, median of E samples.
SignExposures	SignExp object which contains the set of samples for the model parameters.
Bics	measured BIC values on the final run of the sampler.
HyperParam	evolution of estimated hyperparameters when testing different numbers of signatures.

Examples

```
vignette(package="signeR")
```

`signeRFlow` *Launch signeRFlow R Shiny web app*

Description

Launch signeRFlow R Shiny web app locally

Usage

```
signeRFlow()
```

SignExp *SignExp class*

Description

Keep samples for signature and exposure matrices.

Value

Object fields:

`@Sign`	array of signature matrix samples.
`@Exp`	array of exposure matrix samples.
`@sigSums`	Signature sums for each sample, organized by row. Normalizing factors.
`@samples`	Genome sample IDs.
`@mutations`	mutation names.
`@normalized`	boolean variable, indicating whether Sign array has been normalized.

tcga_similarities *TCGA Cosmic similarities*

Description

TCGA Cosmic similarities calculated by signeR.

Usage

```
data("tcga_similarities")
```

Format

A data frame with 112 observations on the following 80 variables.

`sigs` a character vector

`project` a character vector

`SBS1` a numeric vector

`SBS10a` a numeric vector

`SBS10b` a numeric vector

`SBS10c` a numeric vector

`SBS10d` a numeric vector

`SBS11` a numeric vector

`SBS12` a numeric vector

`SBS13` a numeric vector

`SBS14` a numeric vector

`SBS15` a numeric vector

SBS16 a numeric vector

SBS17a a numeric vector

SBS17b a numeric vector

SBS18 a numeric vector

SBS19 a numeric vector

SBS2 a numeric vector

SBS20 a numeric vector

SBS21 a numeric vector

SBS22 a numeric vector

SBS23 a numeric vector

SBS24 a numeric vector

SBS25 a numeric vector

SBS26 a numeric vector

SBS27 a numeric vector

SBS28 a numeric vector

SBS29 a numeric vector

SBS3 a numeric vector

SBS30 a numeric vector

SBS31 a numeric vector

SBS32 a numeric vector

SBS33 a numeric vector

SBS34 a numeric vector

SBS35 a numeric vector

SBS36 a numeric vector

SBS37 a numeric vector

SBS38 a numeric vector

SBS39 a numeric vector

SBS4 a numeric vector

SBS40 a numeric vector

SBS41 a numeric vector

SBS42 a numeric vector

SBS43 a numeric vector

SBS44 a numeric vector

SBS45 a numeric vector

SBS46 a numeric vector

SBS47 a numeric vector

SBS48 a numeric vector

SBS49 a numeric vector

SBS5 a numeric vector

SBS50 a numeric vector

SBS51 a numeric vector
SBS52 a numeric vector
SBS53 a numeric vector
SBS54 a numeric vector
SBS55 a numeric vector
SBS56 a numeric vector
SBS57 a numeric vector
SBS58 a numeric vector
SBS59 a numeric vector
SBS6 a numeric vector
SBS60 a numeric vector
SBS7a a numeric vector
SBS7b a numeric vector
SBS7c a numeric vector
SBS7d a numeric vector
SBS8 a numeric vector
SBS84 a numeric vector
SBS85 a numeric vector
SBS86 a numeric vector
SBS87 a numeric vector
SBS88 a numeric vector
SBS89 a numeric vector
SBS9 a numeric vector
SBS90 a numeric vector
SBS91 a numeric vector
SBS92 a numeric vector
SBS93 a numeric vector
SBS94 a numeric vector

`tcga_tumors` *TCGA tumors used on TCGA Explorer*

Description

List of TCGA tumors used on TCGA Explorer

Usage

```
data("tcga_tumors")
```

Format

A data frame with 37 observations on the following 2 variables.

`projectID` a character vector
`projectName` a character vector

Index