Package 'signeR'

June 14, 2016

Type Package

Title signeR

Version 0.5.0

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

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

Description The signeR package provides an empirical Bayesian approach to mutational signature discovery. It is designed to analyze single nucleotide variaton (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, BSgenome, class, grDevices, GenomicRanges, nloptr, NMF, plot3D, R.methodsS3, R.oo, tensorA, VariantAnnotation

LinkingTo Rcpp, RcppArmadillo

SystemRequirements C++11

NeedsCompilation yes

ByteCompile TRUE

biocViews GenomicVariation, SomaticMutation, StatisticalMethod, Visualization

Suggests knitr, rtracklayer, base64

VignetteBuilder knitr

R topics documented:

signeR-package *signeR*

Description

The signeR package provides an empirical Bayesian approach to mutational signature discovery. It is designed to analyze single nucleotide variaton (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 focus on the characterization and analysis of mutational processes. Its functionalities can be divided in 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, in order 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@cipe.accamargo.org.br>

References

here be references

Examples

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

Classify *Classify unknown samples*

Description

Classify: Assign unknown samples to previously defined groups.

Usage

```
## S3 method for class 'SignExp'
Classify(this, labels, method="knn", k=3, plotfile="Classification_barplot.pdf", ...)
```

Arguments

this	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).
k	number of nearest neighbors considered for classification, used only if method="kNN". Default is 3.
plotfile	file that will be generated with classification graphic output.
...	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 group label. Contains the assignment frequencies of each sample to each group.

Examples

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

## Not run:
    my_labels <- c("a","a",NA,"b","b",NA)
   Class <- Classify(signatures$SignExposures, labels=my_labels, plotfile="Sample_classification.pdf")

## End(Not run)
# see also
vignette(package="signeR")
```

DiffExp *Differential Exposure Analysis*

Description

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

Usage

```
## S3 method for class 'SignExp'
DiffExp(this, labels, method='kw', contrast="all", quant=0.5, cutoff=0.05, plotfile="Diffexp_boxplot
```

Arguments

this	a SignExp object returned by signeR function.
labels	sample labels used to define sample groups.
method	algorithm used to compare each signature exposures among sample groups. Default is "kw", 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 evaluates the null hypothesis of exposure levels beeing constant in all groups. Instead, if this parameter contains a list of group labels, the algorithm will evaluate the null hypothesis of exposure levels beeing constant among those groups.
quant	the p-values quantile which, after log-transform, will be used as DES (Differental Exposure Score). Deafult is 0.5, which means the median log-transformed p-value will be considered as DES.
cutoff	threshold for p-values quantile for signatures to be considered as showing differential exposure.
plotfile	Output file to export p-values boxplot.
pal	Color palette.
...	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.

Examples

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

## Not run:
   diff_exposure <- DiffExp(signatures$SignExposures, labels=my_labels, plotfile="Diffexp_boxplot.pdf")

## End(Not run)

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

generateMatrix *count matrix and opportinity matrix generators*

Description

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

Usage

```
genCountMatrixFromVcf(bsgenome, vcfobj)
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.
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 a SNV change with a 3bp context.

Examples

```
## Not run:
    library(BSgenome.Hsapiens.UCSC.hg19)
    library(rtracklayer)
    library(VariantAnnotation)

    vcfobj <- readVcf("/path/to/a/file.vcf", "hg19")
    mut <- genCountMatrixFromVcf(BSgenome.Hsapiens.UCSC.hg19, vcfobj)

    target_regions <- import(con="/path/to/a/target.bed", format="bed")
   opp <- genOpportunityFromGenome(BSgenome.Hsapiens.UCSC.hg19, target_regions, nsamples=nrows(mut))

## End(Not run)

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

plots *signeR plot functions*

Description

SignPlot: Plot the mutations signatures in a barchart.
SignHeat: Plot the mutations signatures in a heatmap.
ExposureBoxplot: ...
Paths : Plot the convergence of the Gibbs sampler.
BICboxplot : Plot the BICs.

Usage

```
## S3 method for class 'SignExp'
SignPlot(this, file="Signature_plot.pdf", pal="bcr1", threshold=0, plots_per_page=4, gap=1, reord=NA
## S3 method for class 'SignExp'
SignHeat(this, file="Signature_heatmap.pdf", nbins=20, ...)
## S3 method for class 'SignExp'
ExposureBoxplot(this, file="Exposure_boxplot.pdf", pal='terrain', threshold=0, plots_per_page=4, ...
## S3 method for class 'SignExp'
Paths(this, file_sufix="plot.png", plots_per_page=4, ...)
BICboxplot(signeRout, file="Model_selection_BICs.pdf")
```

Arguments

this	a SignExp object returned by signeR function. e.g.: sig$SignExposures
signeRout	the value returned by the signeR function.
file	Output pdf file of the plots.
pal	Color palette used.
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 in this number of bins for plotting, each bin corresponding to a different color.
file_sufix	The suffix of the output file.
...	.

Value

The plot result is saved in the file defined by the file argument.

Examples

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

## Not run:
    SignPlot(signatures$SignExposures)
    Paths(signatures$SignExposures)

## End(Not run)

vignette(package="signeR")
```

---

signeR *signeR*

Description

Generates the signatures.

Usage

```
signeR(M, Mheader = TRUE, samples = "rows", Opport = NA,
    Oppheader = FALSE, nsig = NA, nlim = c(NA, NA),
    try_all = FALSE, ap = NA, bp = NA, ae = NA, be = NA,
    lp = NA, le = NA, var.ap = 10, var.ae = 10,
    testing_burn = 1000, testing_eval = 1000, main_burn =
    10000, main_eval = 2000, start = "lee", estimate_hyper
    = FALSE)
```

Arguments

M	mutation counts matrix of samples x features.
Mheader	If M have 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 have header defined.
nsig	Number of signatures, which can be provided or estimated by the algorithm.
nlim	Define a interval to search for the optimal number of signatures.
try_all	If true, all possible values for nsig will be tested
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
testing_burn	number of burning iterations of the Gibbs sampler used to estimate the number of signatures in data.
testing_eval	number of iterations of the Gibbs sampler used to estimate the number of signatures in data.
main_burn	number of burning iterations of the final Gibbs sampler.
main_eval	number of iterations of the final Gibbs sampler.
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".
estimate_hyper	if TRUE, algorithm estimates optimal values of ap,bp,ae,be,lp,le. Start values can still be provided.

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 contain 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")
```

SignExp *SignExp class*

Description

Keep samples for signature and exposure matrices.

Value

Object fields:

`.Sign`	tensor of signature matrix samples.
`.Exp`	tensor 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 tensor has been normalized.

Index