Package 'MutSigTools'

Vinod Singh, Neil Coleman and Subhajyoti De

2019-02-22

Title: MutSigTools

Type: Package

Version: 1.0.0

Description: MutSigTools implements a wide range of generic utilities for processing and analysis of mutation signatures.

Author: Vinod Kumar Singh, Neil Coleman and Subhajyoti De

Maintainer: The package maintainer vs580@scarletmail.rutgers.edu

License: MIT + file LICENSE

Encoding: UTF-8

LazyData: true

Depends: R (>= 3.4.0), bedr, gplots, factoextra, data.table, tools, stats, GenomicRanges, Biostrings, deconstructSigs.

RoxygenNote: 6.1.1

Suggests:devtools, knitr, rmarkdown.

VignetteBuilder: knitr.

URL: https://github.com/sjdlabgroup/MutSigTools

BugReports: https://github.com/sjdlabgroup/MutSigTools/issues

R topics documented:

1. Introduction: . . . 2
2. Availability and Installation . . . 2
3. Functions . . . 2
   - 3.1 vcfToSNV . . . 2
   - 3.2 contextSNV . . . 3
   - 3.3 processSNV . . . 4
   - 3.4 addSignature . . . 5
   - 3.5 deleteSignature . . . 5
   - 3.6 renameSignature . . . 6
   - 3.7 mergeSignature . . . 6
   - 3.8 clusterSignature . . . 7
   - 3.9 signatureHeatmap . . . 8
   - 3.10 signaturePCA . . . 8
   - 3.11 signature_tSNE . . . 9
   - 3.12 confidenceSig . . . 9
   - 3.13 persistSig . . . 10
   - 3.14 enrichSig . . . 11
   - 3.15 caseControlSig . . . 12

1. Introduction:

The patterns of mutation accumulation in tumor genomes provide insights into past exposure to mutagens, mechanism of DNA damage, repair defects, and the extent of genomic instability during development, aging, and cancer. A number of mutation signatures were initially identified using non-negative matrix-factorization-based approaches in analyzing cancer genomic data, providing insights into etiology of the malignant diseases Alexandrov et al. [2013], Alexandrov and Stratton [2014]. Subsequently additional mutation signatures have been identified using experimental and computational approaches Boot et al. [2018] MutSigTools implements wide range of utilities for processing, analysis, and interpretation of somatic mutations and associated mutation signatures.

2. Availability and Installation

The development version of `MutSigTools` package is available at https://github.com/sjdlabgroup/MutSigTools and can be installed as

```
# install.packages("devtools")
devtools::install_github("sjdlabgroup/MutSigTools",build_vignettes = TRUE )
```

3. Functions

3.1 vcfToSNV

Description

Reads a `vcf` formatted file to catalog mutations in a data frame of class `snv` for downstream analysis. A `snv` dataframe consists of `sample, chr, pos, ref, alt and/or freq` columns.

Usage

```
vcfToSNV(vcf, allelefreq = FALSE)
```

Arguments

- `vcf`: The `path` of a standard `vcf file`.
- `allelefreq`: A logical input, TRUE if mutation allele frequency is required in output object. `Default: FALSE`

Details

Read `vcf` formatted file to catalog mutations in a data frame of class `snv` for downstream analysis for each input sample.

Value

1. `snv` is a dataframe with columns `sample, chr, pos, ref, alt, and/or freq`. Where,
    - `sample` : Name of the sample name.
    - `chr` : chromosome number
    - `pos` : position of mutation
    - `ref` : Reference base (mutated from)
    - `alt` : mutated base (mutated to)
    - `freq` : Alt Allele Frequency (optional)

Note: `vcf file name` with no extension will be the `sample name`.

Examples

```
> vcf_file=system.file("extdata", "test.vcf", package = "MutSigTools", mustWork = TRUE)
> snv=vcfToSNV(vcf=vcf_file, allelefreq=TRUE)
> head(snv)
     sample  chr   pos ref alt     freq
1 CRF004556 chr1 15118   A   G 1.000000
2 CRF004556 chr1 15211   T   G 0.935484
3 CRF004556 chr1 15237   C   T 0.153846
4 CRF004556 chr1 16497   A   G 0.833333
5 CRF004556 chr1 16534   C   T 1.000000
6 CRF004556 chr1 16571   G   A 1.000000
```

3.2 contextSNV

Description

Group mutations according to their occurrence within or outside user-annotated genomic segments.

Usage

```
contextSNV(snv, file = "empty", mode = "include",  bed_file_start_cord  =  0)
```

Arguments

- snv: A dataframe having `sample, chr, pos, ref, alt and/or freq` as its columns. This snv dataframe can be created by the vcfToSNV function of this package.
- `file`: Path of the `BED file`, having genomic segments and their context information.
- `mode`: Default: `include`: Select mutations lying inside the given genomic contexts (obtained from BED file). `exclude`: will give mutations outside the given context.

- `bed_file_start_cord`: Default: `0`: If BED file has 0-based coordinate system. `1`: If BED file has 1-based coordinate system..

Details

When the `include` mode is used, group mutations according to their occurrence in user-specified annotated genomic segments; such segments could represent specific genomic or epigenomic contexts (e.g. actively transcribed coding regions). Alternately, in the `exclude` mode, mutations in undesired regions (e.g.black-listed genomic regions) can be removed.

Value

`snv` dataframe with column names of `sample, chr, pos, ref, alt, freq`. Where, sample name will be concatenated by the its genomic context. Example: if a mutation within a sample of name `sample1`, lies in the context segment of label `C1` given in the input `bed file` , the sample name of mutation is modified to `sample1_C1`. The mutations that don't lie in any context segment given in the `bed file` are discarded.

Examples

```
> BED_file=system.file("extdata", "context_testFile.bed", package = "MutSigTools", mustWork = TRUE)
> data(snv_sample)   # load 'snv' dataframe object
> context_snv=contextSNV(snv=snv_sample,BED_file, mode='include', bed_file_start_cord=0)
> head(context_snv)
        sample  chr   pos ref alt
1 CRF004556_E7 chr1 15118   A   G
2 CRF004556_E7 chr1 15211   T   G
3 CRF004556_E7 chr1 15237   C   T
4 CRF004556_E7 chr1 16497   A   G
5 CRF004556_E7 chr1 16534   C   T
6 CRF004556_E7 chr1 16571   G   A
```

3.3 processSNV

Description

Estimate frequency of point mutations in tri-nucleotide motif contexts for one or more input samples.

Usage

```
processSNV(snv, bsg = BSgenome.Hsapiens.UCSC.hg19::Hsapiens)
```

Arguments

- `snv`: A dataframe having sample, chr, pos, ref, alt and/or freq as its columns. The `snv` dataframe can be created by `vcfToSNV` function.
- `bsg`: An boject of class BSGenome. `Default:` BSgenome.Hsapiens.UCSC.hg19::Hsapiens

Details

Estimation of mutation frequency in pre-determined, user-specified nucleotide motif contexts for input samples. The `BSgenome` object contains the reference genome information, and should be specified for alternative assembly of human genomes or nonhuman reference genomes. The output returns a data frame containing

estimated mutation frequencies in the different genomic contexts, as provided in the `contextfreq` object for each input sample.

Value

A data frame of class `contextfreq` containing mutation frequency in user-specified nucleotide contexts.

Examples

```
> data(snv_sample)
> context.freq=processSNV(snv=snv_sample, bsg = BSgenome.Hsapiens.UCSC.hg19::Hsapiens)
> context.freq[,1:6]   # view first 6 mutation frequency contexts of 96
          A[C>A]A A[C>A]C A[C>A]G A[C>A]T C[C>A]A C[C>A]C
CRF004556    1920    1412     653    1180    1845    1439
```

3.4 addSignature

Description

Add new signature(s) to an existing set of `mutation signatures`, and return the updated signature set.

Usage

```
addSignature(sigmatrix2, sigmatrix1)
```

Arguments

- `sigmatrix1`: An object of class `mutsig` describing the existing set of signatures.
- `sigmatrix2`: An object of class `mutsig` describing the new signature(s).

Value

The list of mutational signatures with the new signature added.

Examples

```
> newSigMatrix=addSignature(sigmatrix2, sigmatrix1)
```

3.5 deleteSignature

Description

Delete selected signatures from an existing set of mutation signatures.

Usage

```
deleteSignature(sigmatrix, del_sig)
```

Arguments

- `sigmatrix`: An object of class `mutsig` describing the existing set of signatures.
- `del_sig`: Signature(s) to be removed.

Details

Deletion of one or more mutation signatures from an existing set of mutation signatures upon which the updated signature set is returned.

Value

An object of mutsig class with the deleted signatures.

Examples

```
> reducedSigMatrix=deleteSignature(sigmatrix=signatures.cosmic, c(2,5)) # delete signature 2 and 5.
```

3.6 renameSignature

Description

Rename selected signatures from an existing set of signatures.

Usage

```
renameSignature(sigmatrix,selectSig,renameSig)
```

Arguments

- `sigmatrix`: An object of class mutsig describing the existing set of signatures.
- `selectSig`: A vector containing name(s) of the signature(s) to be removed.
- `renameSig`: A vector containing updated name(s) of the signatures.

Details

Renaming of selected mutation signatures to an existing signature set, upon which the updated signature set is returned.

Value

An object of `mutsig` class.

Examples

```
> renamedSigMatrix=renameSignature(sigmatrix=signatures.cosmic, selectSig=sig=c(2,5),
> renameSig=resig=c('two', 'five')) # rename signatures 2 and 5.
```

3.7 mergeSignature

Description

This function creates a new mutational signature by taking a linear combination of `k` other signatures s_{11}, s_{12},...,s_{kn} in a matrix of mutational signatures. In order to use this function, a weight vector w_1, w_2,...,w_k must be specified. The new mutational signature is created by multiplying the component signature values by the corresponding weights in the weight vector. The formula for the entries of the new signature is $s_{11}w_1+s_{21}w_2+\ldots+s_{k1}w_k$, $s_{12}w_1+s_{22}w_2+\ldots+s_{k2}w_k$,...,$s_{1n}w_1+s_{2n}w_2+\ldots+s_{kn}w_k$. The function returns this

new signature plus all of the signatures in the original dataset that were not merged into the new matrix of signatures.

Usage

```
mergeSignature(sigmatrix,sig,weights)
```

Arguments

- `sigmatrix`: A `mutsig` object of mutational signatures.
- `sig`: A vector that describes the signatures to be merged
- `weights`: A vector that describes the relative weights of each of the component signatures in the merged mutational signature.

Value

A new matrix of signatures consisting of the signatures in the original matrix of mutational signatures that were not merged as well as the mutational signature that is a result of merging the input signatures. For example, if there were 10 signatures in the original signature matrix and signatures 1-6 were set to be merged, then the output value would be a matrix of 5 signatures(signature 7,8,9,10 and the result of merging signatures 1-6).

Examples

```
> weights=rep(1/27,27)
> mergeSignature(signatures.cosmic,1:27,weights)
> # another example
> weights=rep(1/6,6)
> mergeSignature(signatures.cosmic,1:6,weights)
```

3.8 clusterSignature

Performs a `hierarchical clustering` analysis of an input mutational signature matrix using the Euclidean distance metric.

Usage

```
clusterSignature(sigmatrix)
```

Arguments

- `sigmatrix`: An input matrix of mutational signatures.

Value

A hierarchical clustering object formed from the input matrix of mutational signatures. This can then be plotted to show how the mutational signatures in the input matrix are related to one another.

Examples

```
> hclust_obj=clusterSignature(sigmatrix=signatures.cosmic)
```

3.9 signatureHeatmap

Description

Construct a `heatmap` showing relative weights of tri-nucleotide motif for different mutation signatures.

Usage

```
signatureHeatmap(sigmat, pngfile='signatures_heatmap' )
```

Arguments

- `sigmat`: An object of class `mutsig` describing the existing set of signatures.
- `pngfile`: The name of `png` formatted image file to be created.

Details

Create a heatmap to show relative weights of trinucleotide motifs for different mutation signatures in a signature-set.

Value

A png formatted image file.

Examples

```
> sigmat=signatures.cosmic
> signatureHeatmap(sigmat, pngfile='signatures_heatmap')
```

3.10 signaturePCA

Description

Constructs three PCA plots showing variations among the signatures in terms of the weights of the principal components for different mutation signatures, as well as the eigenvalues and cosine similarity among expected and actual signatures.

Usage

```
signaturePCA(sigmat, pngfile )
```

Arguments

- `sigmat`: An object of class `mutsig` describing a set of signatures.
- `pngfile`: Name of `png` formatted image file to be created.

Value

Create three `PCA plots` with the following extensions to show difference among mutational signatures.

- `*_Eigen.png` : The eigenvalues of the PCA and the relative amount of the variation each eigenvalue.
- `*_var.png` : The different single-nucleotide alteration components in terms of the two dimensions.
- `*_PCA.png` : The different signatures in terms of the 2 non-dimensional vectors derived by the PCA analysis and the cosine similarity between the estimate of each mutation signature using PCA and the actual mutation signature(in order to determine how accurately the 2 dimensions calculated by PCA represent the mutational signatures).

Examples

```
> sigmat=signatures.cosmic
> signaturePCA(sigmat, pngfile="signaturePCA")
```

3.11 signature_tSNE

Description

Constructs tSNE plot of mutation signatures.

Usage

```
signature_tSNE(sigmat, pngfile )
```

Arguments

- `sigmat`: An object of class `mutsig` describing a set of signatures.
- `pngfile`: Name of `png` formatted image file to be created.

Value

Create a `tSNE plots` plot to show similarity among mutational signatures.

- `*_tSNE.png` : The tSNE plot file of the mutation signatures.

Examples

```
> sigmat=signatures.cosmic
> signature_tSNE(sigmat, pngfile="signature_tSNE")
```

3.12 confidenceSig

Description

Provides an interval of uncertainty for estimated weights of known mutation signatures in the catalog of mutations from a set of samples.

Usage

```
confidenceSig(contextfreq.sample, subsample=0.8, iter=1000 , signatures.ref=signatures.cosmic,
lbound=0.1, ubound=0.9,  replace=FALSE)
```

Arguments

- `contextfreq.sample`: A sample from the dataframe of class `contextfreq` containing mutation frequency in tri-nucleotide contexts.
- `subsample`: Proportion of mutations included during each subsampling. `Default: 0.8` (80 percent)
- `iter`: Number of iterations of subsampling. `Default: 1000`
- `signatures.ref`: An object of class mutsig comprising the set of signatures. ('signatures.nature2013' or 'signatures.cosmic' or 'signatures.cosmic.2019'), Default: signatures.cosmic
- `lbound`: Lower bound of the interval of uncertainty for estimated weights of the signatures. `Default: 0.1` (10 percent)
- `ubound`: Upper bound of the interval of uncertainty for estimated weights of the signatures. `Default: 0.9` (90 percent)
- `replace`: should sampling be with replacement? TRUE or FALSE. `Default: FALSE`

Details

Provides an interval of uncertainty for estimated weights of known mutation signatures in the catalog of mutations from a set of samples. First, based on the catalog of mutations in a sample, weights of the mutation signatures are estimated. Next, mutations are sub-sampled iteratively without replacement, each time estimating the weights of the mutation signatures. The intervals of uncertainty for weights of each mutation signatures are determined by aggregating observations from a given number of iterations.

Value

An object containing the following information:

- `observed.weights` : A data frame containing estimated weights of known mutation signatures based on all mutations in a sample.
- `median.weights` : A data frame containing estimated median of the weights of known mutation signatures based on iteratively subsampled mutations.
- `ubound.weights` : A data frame containing upper-bound values of the weights of known mutation signatures based on iteratively subsampled mutations.
- `lbound.weights`: : A data frame containing lower-bound values of the weights of known mutation signatures based on iteratively subsampled mutations.

Examples

```
> data(contextfreq.sample_test)
> robust_sig_object=confidenceSig(contextfreq.sample=contextfreq.sample_test, subsample=0.8, iter=50,
signatures.ref=signatures.cosmic, lbound=0.1, ubound=0.9, replace=FALSE)
```

3.13 persistSig

Determine the burden of different mutation signatures across different allele frequency ranges. The function should be used with caution; Local copy number and purity-corrected variant allele frequencies should be provided in the column 6 of the `snv` object.

Usage

```
persistSig(snv, th_vec_lw, th_vec_up, bsg = BSgenome.Hsapiens.UCSC.hg19::Hsapiens)
```

Arguments

- snv: A dataframe having `sample, chr, pos, ref, alt and/or freq` as its columns.This dataframe may have more than one sample.
- `th_vec_lw`: A vector of lower limits of frequency ranges.
- `th_vec_up`: A vector of upper limits of frequency ranges. `signatures.ref`: An object of class mutsig comprising the set of signatures. ('signatures.nature2013' or 'signatures.cosmic' or 'signatures.cosmic.2019'), Default: signatures.cosmic

Value

A list of samples having mutational signature corresponding to each allele frequency range

Examples

```
> data(snv_sample)   # load 'snv' dataframe object
> mut_sig_per_freq_range=persistSig(snv=snv_sample,th_vec_lw=c(0,0.4), th_vec_up=c(0.1,1),
bsg = BSgenome.Hsapiens.UCSC.hg19::Hsapiens)  # list of samples having mutational signature for each al

> mut_sig_per_freq_range$CRF004556[,1:6] # view some signatures at different allele freq ranges
          sig.1 sig.2     sig.3 sig.4      sig.5      sig.6
0-0.1 0.02644092     0 0.6242417     0 0.00000000 0.00000000
0.4-1 0.19941897     0 0.2126304     0 0.09984891 0.08473515
```

3.14 enrichSig

Description

Determine over-represented mutation signatures in individual case sample(s) relative to that in the panel of control samples. This is suitable when case samples are heterogeneous, and only some of them might have excess of certain mutation signatures of interest relative to the control population.

Usage

```
enrichSig(contextfreq.cases, contextfreq.controls, signatures.ref=signatures.cosmic,
threshold=0.05)
```

Arguments

- `contextfreq.cases`: A data frame of class `contextfreq` containing mutation frequency in tri-nucleotide contexts in case samples
- `contextfreq.controls`: A data frame of class `contextfreq` containing mutation frequency in tri-nucleotide contexts in control samples.
- `signatures.ref`: An object of class mutsig comprising the set of signatures. ('signatures.nature2013' or 'signatures.cosmic' or 'signatures.cosmic.2019'), Default: signatures.cosmic
- `threshold`: Threshold for uncorrected percentile score. `Default: 0.05`

Details

Determine over-represented mutation signatures in individual case sample(s), highlighting those that are significantly enriched. The extent of enrichment is indicated using a percentile score, with low scores indicating

high enrichment for specific mutation signatures in a case sample relative to that in the panel of control samples.

Value

An object of class `enrichSig.obj` providing the following information:

- `n.case`: Number of case samples.
- `n.control`: Number of control samples.
- `case.weights`: A data frame containing estimated weights of known mutation signatures in the case samples.
- `control.weights`: A data frame containing estimated weights of known mutation signatures in the control samples.
- `case.percentile`: Percentile scores corresponding to the extent of enrichment of known mutation signatures in case sample(s) relative to that in the control samples.

Details

Determine over-represented mutation signatures in individual case sample(s), highlighting those that are significantly enriched. The extent of enrichment is indicated using a percentile score, with low scores indicating high enrichment for specific mutation signatures in a case sample relative to that in the panel of control samples.

Examples

```
> data(contextfreq.cases_test)
> data(contextfreq.controls_test)
> enrich_obj=enrichSig(contextfreq.case=contextfreq.cases_test,
contextfreq.controls=contextfreq.controls_test,
signatures.ref=signatures.cosmic, threshold=0.05)
```

3.15 caseControlSig

Description

Identifies signatures with significantly higher mutation burden in case samples over control samples.

Usage

```
caseControlSig(contextfreq.cases, contextfreq.controls, signatures.ref=signatures.cosmic,
threshold=0.05, adjust="fdr")
```

Arguments

- `contextfreq.cases`: A data frame of class `contextfreq` containing mutation frequency in tri-nucleotide contexts in case samples
- `contextfreq.controls`: A data frame of class `contextfreq` containing mutation frequency in tri-nucleotide contexts in control samples.
- `signatures.ref`: An object of class mutsig comprising the set of signatures. ('signatures.nature2013' or 'signatures.cosmic' or 'signatures.cosmic.2019'), Default: signatures.cosmic
- `threshold`: Threshold for uncorrected percentile score. `Default: 0.05`
- `adjust`: Method for p-value correction for multiple testing. Options are as provided in the function `p.adjust`. The default method is the `FDR` method.

Details

Identifies signatures with significantly higher mutation burden in case samples compared to control samples using the Wilcoxon Rank sum test. This is suitable when the samples are homogeneous within respective groups, and signatures significantly over-represented in cases relative to the controls are of interest.

Value

An object providing the following information:

- `n.case` : Number of case samples.
- `n.control`: Number of control samples.
- `case.weights`: A data frame containing estimated weights of known mutation signatures in the case samples.
- `control.weights`: A data frame containing estimated weights of known mutation signatures in the control samples.
- `p.value` : Indicates statistical significance of higher mutation burden of signatures in cases over controls.
- `adjust.p.value`: P-values adjusted for multiple testing correction.

Examples

```
> data(contextfreq.cases_test)
> data(contextfreq.controls_test)
> signif_signatures_obj=caseControlSig(contextfreq.cases=contextfreq.cases_test,
contextfreq.controls=contextfreq.controls_test, signatures.ref=signatures.cosmic,
threshold=0.05, adjust="fdr")
```

References

Ludmil B Alexandrov and Michael R Stratton. Mutational signatures: the patterns of somatic mutations hidden in cancer genomes. *Current opinion in genetics & development*, 24:52–60, 2014.

Ludmil B Alexandrov, Serena Nik-Zainal, David C Wedge, Samuel AJR Aparicio, Sam Behjati, Andrew V Biankin, Graham R Bignell, Niccolo Bolli, Ake Borg, Anne-Lise Børresen-Dale, et al. Signatures of mutational processes in human cancer. *Nature*, 500(7463):415, 2013.

Arnoud Boot, Mi Ni Huang, Alvin WT Ng, Szu-Chi Ho, Jing Quan Lim, Yoshiiku Kawakami, Kazuaki Chayama, Bin Tean Teh, Hidewaki Nakagawa, and Steven G Rozen. In-depth characterization of the cisplatin mutational signature in human cell lines and in esophageal and liver tumors. *Genome research*, 28 (5):654–665, 2018.

Index

BSgenome
    BSgenome, 4
    BSgenome.Hsapiens.UCSC.hg19, 4

Dataframes
    `contextfreq`, 5
    sigmatrix, 5
    snv, 2

Signatures
    mutation signatures, 5
    signatures.cosmic, 9

Variation among Signatures
    heatmap, 8
    hierarchical clustering, 7
    signature_tSNE, 9
    signaturePCA, 8