Package ‘mutSignatures’

October 13, 2022

Type Package

Title Decipher Mutational Signatures from Somatic Mutational Catalogs

Version 2.1.1

Date 2020-11-01

Author Damiano Fantini, Vania Vidimar, Joshua J Meeks

Maintainer Damiano Fantini <damiano.fantini@gmail.com>

Description Cancer cells accumulate DNA mutations as result of DNA damage and DNA repair processes. This computational framework is aimed at deciphering DNA mutational signatures operating in cancer. The framework includes modules that support raw data import and processing, mutational signature extraction, and results interpretation and visualization. The framework accepts widely used file formats storing information about DNA variants, such as Variant Call Format files. The framework performs Non-Negative Matrix Factorization to extract mutational signatures explaining the observed set of DNA mutations. Bootstrapping is performed as part of the analysis. The framework supports parallelization and is optimized for use on multi-core systems. The software was described by Fantini D et al (2020) <doi:10.1038/s41598-020-75062-0> and is based on a custom R-based implementation of the original MATLAB WTSI framework by Alexandrov LB et al (2013) <doi:10.1016/j.celrep.2012.12.008>.

License GPL-2

Depends R (>= 3.5), foreach

Imports graphics, stats, cluster, doParallel, ggplot2, pracma, proxy, methods

Suggests dplyr, reshape2, kableExtra, gridExtra, knitr, rmarkdown

VignetteBuilder knitr

Encoding UTF-8

LazyData true

URL https://www.data-pulse.com/dev_site/mutsignatures/

RoxygenNote 7.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2020-11-09 07:30:02 UTC
R topics documented:

- `mutSignatures-package` ........................................... 3
- `as.data.frame.mutationCounts-method` .......................... 4
- `as.data.frame.mutSignatures-method` ............................ 4
- `as.data.frame.mutSignExposures-method` ....................... 5
- `as.list.mutationSignatures-method` ............................ 5
- `as.list.mutFrameworkParams-method` ........................... 6
- `as.matrix.mutationCounts-method` ............................. 6
- `as.mutation.counts` ........................................... 7
- `as.mutation.signatures` ....................................... 7
- `as.mutsign.exposures` ........................................... 8
- `attachContext` .................................................. 8
- `attachMutType` ................................................ 9
- `cbind2` .......................................................... 11
- `coerceObj` ...................................................... 12
- `countMutTypes` ................................................ 12
- `decipherMutationalProcesses` ................................ 14
- `extractXvarlinkData` .......................................... 15
- `filterSNV` ...................................................... 16
- `frequencize` .................................................... 17
- `getCosmicSignatures` ......................................... 18
- `getCounts` ...................................................... 19
- `getFwkParam` .................................................. 19
- `getMutationTypes` ............................................ 20
- `getSampleIdentifiers` ........................................ 20
- `getSignatureIdentifiers` .................................... 21
- `importVCFfiles` ............................................... 21
- `matchSignatures` .............................................. 22
- `msigPlot` ...................................................... 23
- `mutationCounts-class` ....................................... 24
- `mutationSignatures-class` .................................. 25
- `mutFrameworkParams-class` .................................. 25
- `mutSigData` .................................................. 26
- `mutSignExposures-class` ..................................... 27
- `plotMutTypeProfile` .......................................... 28
- `plotSignExposures` ........................................... 29
- `prelimProcessAssess` ......................................... 30
- `processVCFdata` ............................................... 31
- `removeMismatchMut` ........................................... 32
- `resolveMutSignatures` ........................................ 33
- `revCompl` ...................................................... 34
- `setFwkParam` .................................................. 36
- `setMutClusterParams` ......................................... 36
- `show.mutationCounts-method` ................................. 38
- `show.mutationSignatures-method` .......................... 39
- `show.mutFrameworkParams-method` .......................... 39
- `show.mutSignExposures-method` .............................. 40
**mutSignatures-package**

Decipher Mutational Signatures from Somatic Mutational Catalogs.

**Description**

Cancer cells accumulate DNA mutations as result of DNA damage and DNA repair processes. mutSignatures is a computational framework that is aimed at deciphering DNA mutational signatures operating in cancer. The input is a numeric matrix of DNA mutation counts detected in a panel of cancer samples. The framework performs Non-negative Matrix Factorization to extract mutational signatures explaining the observed set of DNA mutations. The framework relies on parallelization and is optimized for use on multi-core systems. This framework was described by Fantini D et al (2020) [https://www.nature.com/articles/s41598-020-75062-0/](https://www.nature.com/articles/s41598-020-75062-0/) and is built upon a custom R-based implementation of the original MATLAB WTSI framework by Alexandrov LB et al (2013) [https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3588146/](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3588146/). The mutSignatures framework has been described in peer-reviewed publications, including Fantini D et al (2018) [https://www.nature.com/articles/s41388-017-0099-6/](https://www.nature.com/articles/s41388-017-0099-6/) and Fantini D et al (2019) [https://www.sciencedirect.com/science/article/abs/pii/S1078143918303818/](https://www.sciencedirect.com/science/article/abs/pii/S1078143918303818/). The framework includes three modules that support raw data import and pre-processing, mutation counts deconvolution, and data visualization.

**References**

More info, examples and vignettes:

1. **GitHub Repo**: [https://github.com/dami82/mutSignatures/](https://github.com/dami82/mutSignatures/)
2. **More info and examples** about the mutSignatures R library: [https://www.data-pulse.com/dev_site/mutsignatures/](https://www.data-pulse.com/dev_site/mutsignatures/)
3. **2020 Sci Rep paper** describing the latest version of mutSignatures: [https://www.nature.com/articles/s41598-020-75062-0/](https://www.nature.com/articles/s41598-020-75062-0/)
4. **Oncogene paper**: Mutational Signatures operative in bladder cancer: [https://www.nature.com/articles/s41388-017-0099-6/](https://www.nature.com/articles/s41388-017-0099-6/)
as.data.frame,mutationCounts-method

Convert a mutationCounts object to data.frame.

Description

Coerce a mutationCounts-class object to data.frame by applying the coerceObj method.

Usage

## S4 method for signature 'mutationCounts'
as.data.frame(x)

Arguments

x  

a mutationCounts object

as.data.frame,mutationSignatures-method

Convert a mutationSignatures object to data.frame.

Description

Coerce a mutationSignatures-class object to data.frame by applying the coerceObj method.

Usage

## S4 method for signature 'mutationSignatures'
as.data.frame(x)

Arguments

x  

a mutationSignatures object
as.data.frame, mutSignExposures-method

Convert and/or transpose a mutSignExposures object to data.frame.

Description

Coerce a mutSignExposures-class object to data.frame by applying the coerceObj method. The data.frame can be returned in a transposed or non-transposed format.

Usage

```r
## S4 method for signature 'mutSignExposures'
as.data.frame(x, row.names = NULL, optional = NULL, ...)
```

Arguments

- `x`: a mutSignExposures object
- `row.names`: NULL, not used
- `optional`: NULL, not used
- `...`: additional parameters to be passed to coerceObj, such as transpose (logical)

as.list, mutationSignatures-method

Convert a mutationSignatures object to list.

Description

Coerce a mutationSignatures-class object to list by applying the coerceObj method.

Usage

```r
## S4 method for signature 'mutationSignatures'
as.list(x)
```

Arguments

- `x`: a mutationSignatures object
as.list, mutFrameworkParams-method

Convert a mutFrameworkParams object to list.

Description

Coerce a mutFrameworkParams-class object to list by applying the coerceObj method.

Usage

```r
## S4 method for signature 'mutFrameworkParams'
as.list(x)
```

Arguments

- `x` a mutFrameworkParams object

as.matrix, mutationCounts-method

Convert a mutationCounts object to matrix.

Description

Coerce a mutationCounts-class object to matrix by applying the coerceObj method.

Usage

```r
## S4 method for signature 'mutationCounts'
as.matrix(x)
```

Arguments

- `x` a mutationCounts object
as.mutation.counts  

Method as.mutation.counts.

Description

Cast a data.frame into a mutationCounts-class object.

Usage

as.mutation.counts(x, rownames = NULL, colnames = NULL)

## S4 method for signature 'data.frame'
as.mutation.counts(x, rownames = NULL, colnames = NULL)

Arguments

x  
an object to extract Signature Identifiers from, i.e. a mutSignExposures-class
rownames  
character vector to overwrite data row names. Can be NULL if rownames(x) is
not NULL.
colnames  
character vector to overwrite data column names. Can be NULL if colnames(x) is
not NULL.

as.mutation.signatures  

Method as.mutation.signatures.

Description

Cast a data.frame into a mutationCounts-class object.

Usage

as.mutation.signatures(x)

## S4 method for signature 'data.frame'
as.mutation.signatures(x)

Arguments

x  
a data.frame to be converted to a mutationCounts-class object.
as.mutsign.exposures  Method as.mutsign.exposures.

Description

Cast a data.frame into a mutSignExposures-class object.

Usage

as.mutsign.exposures(x, samplesAsCols = TRUE)
## S4 method for signature 'data.frame,logical'
as.mutsign.exposures(x, samplesAsCols = TRUE)

Arguments

x  a data.frame to be converted to a mutSignExposures-class object.
samplesAsCols  logical, are samples listed as columns in the input data.frame. If FALSE, samples are expected to be listed as rows in the input data.frame

attachContext  Attach Nucleotide Context.

Description

Retrieve the nucleotide context around each DNA variant based on the genomic coordinates of the variant and a reference BSGenome database.

Usage

attachContext(
  mutData,
  BSGenomeDb,
  chr_colName = "chr",
  start_colName = "start_position",
  end_colName = "end_position",
  nucl_contextN = 3,
  context_colName = "context"
)
**Arguments**

- **mutData**
  data.frame storing mutation data

- **BSGenomeDb**
  a BSGenomeDb-class object, storing info about the genome of interest

- **chr_colName**
  string, name of the column storing seqNames. Defaults to "chr"

- **start_colName**
  string, name of the column storing start positions. Defaults to "start_position"

- **end_colName**
  string, name of the column storing end positions. Defaults to "end_position"

- **nucl_contextN**
  integer, the span of nucleotides to be retrieved around the variant. Defaults to 3

- **context_colName**
  string, name of the column that will be storing the nucleotide context. Defaults to 'context'

**Details**

This function is part of the user-interface set of tools included in mutSignatures. This is an exported function.

**Value**

a modified data.frame including the nucleotide context in a new column

**Author(s)**

Damiano Fantini, <damiano.fantini@gmail.com>

**References**

More information and examples about mutational signature analysis can be found here:

1. [GitHub Repo](https://github.com/dami82/mutSignatures/)
2. [More info and examples](https://www.data-pulse.com/dev_site/mutsignatures/) about the mutSignatures R library:
3. [Sci Rep paper](https://www.nature.com/articles/s41598-020-75062-0/), introducing mutS:
4. [Oncogene paper](https://www.nature.com/articles/s41388-017-0099-6), Mutational Signatures Operative in Bladder Cancer:

---

**Description**

Modify a data.frame carrying information about DNA mutation, and add a new column that stores formatted multi-nucleotide types.
Usage

attachMutType(
  mutData,
  ref_colName = "reference_allele",
  var_colName = "variant_allele",
  var2_colName = NULL,
  context_colName = "context",
  format = 1,
  mutType_dict = "alexa",
  mutType_colName = "mutType"
)

Arguments

mutData       data.frame including information about DNA mutations
ref_colName   string, pointing to the column with information about the sequence of the "reference_allele"
var_colName   string, pointing to the column with information about the sequence of the "variant_allele"
var2_colName  string (optional), pointing to the column with information about the sequence of a second "variant_allele". Can be NULL
context_colName string, pointing to the column with information about the nucleotidic "context"
format        integer, indicates the desired mutation type format: (1) N[R>V]N; (2) NN.R>V; (3) R."V[NRN][NVN]
mutType_dict  string, indicates the dictionary to be used for simplifying reverse-complement identical mutation types. It is recommended to use the standard dictionary from COSMIC, by selecting the default value, i.e. "alexa".
mutType_colName string, column name of the new column added to the data.frame where mutTypes are stored.

Details

This function is part of the user-interface set of tools included in mutSignatures. This is an exported function.

Value

a data.frame including a new column with mutation Types.

Author(s)

Damiano Fantini, <damiano.fantini@gmail.com>
References

More information and examples about mutational signature analysis can be found here:

1. GitHub Repo: https://github.com/dami82/mutSignatures/
2. More info and examples about the mutSignatures R library: https://www.data-pulse.com/dev_site/mutsignatures/
4. Oncogene paper, Mutational Signatures Operative in Bladder Cancer: https://www.nature.com/articles/s41388-017-0099-6

Examples

```r
A <- data.frame(REF = c("A", "T", "G"),
                 VAR = c("G", "C", "C"),
                 CTX = c("TAG", "GTG", "CGA"),
                 stringsAsFactors = FALSE)
mutSignatures::attachMutType(mutData = A, ref_colName = "REF",
                             var_colName = "VAR", context_colName = "CTX")
```

---

**cbind2**

Combine two `mutationSignatures-class` objects.

**Description**

Combine two `mutationSignatures-class` objects.

**Usage**

```r
## S4 method for signature 'mutationSignatures,mutationSignatures'
cbind2(x, y)
```

**Arguments**

- `x` the first `mutSignExposures-class` object to combine
- `y` the first `mutSignExposures-class` object to combine

**Details**

A variant of this method accepting more than 2 object to combine together is under preparation and be available soon...
Description

Cast an object to a different format, by extracting and returning the most appropriate information. Note that data.frames can be coerced to one of the classes defined in the mutSignatures package using coerceObj.

Usage

coerceObj(x, to, ...)

## S4 method for signature 'mutFrameworkParams,character'
coerceObj(x, to)

## S4 method for signature 'mutationSignatures,character'
coerceObj(x, to)

## S4 method for signature 'mutationCounts,character'
coerceObj(x, to, ...)

## S4 method for signature 'mutSignExposures,character'
coerceObj(x, to, ...)

## S4 method for signature 'data.frame,character'
coerceObj(x, to, ...)

Arguments

x          an object to coerce to a different format
to         string, indicates the expected format (such as list or data.frame)
...         additional parameters passed to the functions used for the coercion

countMutTypes  Count Mutation Types.

Description

Analyze a table (data.frame) including mutation counts. Count and aggregate Count Mutation Types. If multiple samples are included in the same table, results are aggregated by samples.

Usage

countMutTypes(mutTable, mutType_colName = "mutType", sample_colName = NULL)
countMutTypes

Arguments

- **mutTable**
  - data.frame including mutation types and an optional sample ID column
- **mutType_colName**
  - string, name of the column storing mutTypes
- **sample_colName**
  - string, name of the column storing sample identifiers. Can be NULL

Details

This function is part of the user-interface set of tools included in mutSignatures. This is an exported function.

Value

- a mutationCounts-class object

Author(s)

Damiano Fantini, <damiano.fantini@gmail.com>

References

More information and examples about mutational signature analysis can be found here:

1. **GitHub Repo**: [https://github.com/dami82/mutSignatures/](https://github.com/dami82/mutSignatures/)
2. **More info and examples** about the mutSignatures R library: [https://www.data-pulse.com/dev_site/mutsignatures/](https://www.data-pulse.com/dev_site/mutsignatures/)
3. **Sci Rep paper**, introducing mutS: [https://www.nature.com/articles/s41598-020-75062-0/](https://www.nature.com/articles/s41598-020-75062-0/)
4. **Oncogene paper**, Mutational Signatures Operative in Bladder Cancer: [https://www.nature.com/articles/s41388-017-0099-6](https://www.nature.com/articles/s41388-017-0099-6)

Examples

```r
x <- mutSignatures:::getTestRunArgs("countMutTypes")
x
y <- mutSignatures::countMutTypes(mutTable = x, 
                              mutType_colName = "mutation", 
                              sample_colName = "sample")
y
```
decipherMutationalProcesses

*Decipher Mutational Processes Contributing to a Collection of Genomic Mutations.*

**Description**

Decipher Mutational ProCancer cells accumulate DNA mutations as a result of DNA damage and DNA repair processes. This computational framework allows to decipher mutational processes from cancer-derived somatic mutational catalogs.

**Usage**

`decipherMutationalProcesses(input, params)`

**Arguments**

- `input` a mutationCounts-class object, including a mutation counts data.
- `params` a mutFrameworkParams-class object including all the parameters required for running the mutational signature analysis.

**Details**

This is one of the core functions included in the original mutSignatures R library, and in the WTSI MATLAB framework. This is the main user interface for the mutSignatures analysis.

**Value**

list including all results of the analysis. The extracted signatures (processes) are included in the "processes" element of the list. The relative contribution of each signature in each sample is summarized in the "exposures" element.

**Author(s)**

Damiano Fantini, <damiano.fantini@gmail.com>

**References**

More information and examples about mutational signature analysis can be found here:

1. GitHub Repo: [https://github.com/dami82/mutSignatures/](https://github.com/dami82/mutSignatures/)
2. More info and examples about the mutSignatures R library: [https://www.data-pulse.com/dev_site/mutsignatures/](https://www.data-pulse.com/dev_site/mutsignatures/)
3. Sci Rep paper, introducing mutS: [https://www.nature.com/articles/s41598-020-75062-0/](https://www.nature.com/articles/s41598-020-75062-0/)
4. Oncogene paper, Mutational Signatures Operative in Bladder Cancer: [https://www.nature.com/articles/s41388-017-0099-6](https://www.nature.com/articles/s41388-017-0099-6)
5. WTSI framework: [https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3588146/](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3588146/)
extractXvarlinkData

Examples

```r
library(mutSignatures)
x <- mutSignatures:::getTestRunArgs("decipherMutationalProcesses")
$x$muts
y <- mutSignatures::decipherMutationalProcesses(input = x$muts,
params = x$params)
y$Results$signatures
```

extractXvarlinkData  
Extract Variants from XvarlinkData.

Description

Extract Variants from data stored as XvarlinkData.

Usage

```r
extractXvarlinkData(xvarLink_data)
```

Arguments

- `xvarLink_data`  
  character vector, including mutation data embedded in XvarlinkData

Details

This function is part of the user-interface set of tools included in mutSignatures. This is an exported function.

Value

a data.frame including mutations as well as corresponding reference nucleotides.

Author(s)

Damiano Fantini, <damiano.fantini@gmail.com>

References

More information and examples about mutational signature analysis can be found here:

1. **GitHub Repo**: [https://github.com/dami82/mutSignatures/](https://github.com/dami82/mutSignatures/)
2. **More info and examples** about the mutSignatures R library: [https://www.data-pulse.com/dev_site/mutsignatures/](https://www.data-pulse.com/dev_site/mutsignatures/)
3. **Sci Rep paper**, introducing mutS: [https://www.nature.com/articles/s41598-020-75062-0/](https://www.nature.com/articles/s41598-020-75062-0/)
4. **Oncogene paper**, Mutational Signatures Operative in Bladder Cancer: [https://www.nature.com/articles/s41388-017-0099-6](https://www.nature.com/articles/s41388-017-0099-6)
filterSNV

Filter Single Nucleotide Variants.

Description

Remove entries corresponding to non-SNV, such as insertions and deletions.

Usage

filterSNV(dataSet, seq_colNames)

Arguments

dataSet: data.frame including variant information
seq_colNames: character vector with the names of the columns storing variant data

Details

This function is part of the user-interface set of tools included in mutSignatures. This is an exported function.

Value

a filtered data.frame only including SNVs

Author(s)

Damiano Fantini, damiano.fantini@gmail.com

References

More information and examples about mutational signature analysis can be found here:

1. GitHub Repo: https://github.com/dami82/mutSignatures/
2. More info and examples about the mutSignatures R library: https://www.data-pulse.com/dev_site/mutsignatures/
4. Oncogene paper, Mutational Signatures Operative in Bladder Cancer: https://www.nature.com/articles/s41388-017-0099-6
frequencize

Examples

```r
x <- mutSignatures:::getTestRunArgs("filterSNV")
nrow(x)

y <- mutSignatures::filterSNV(dataSet = x,
    seq_colNames = c("REF", "ALT"))
nrow(y)
```

---

**frequencize**

*Convert Mutation Counts to PerMille Frequencies.*

**Description**

Convert Mutation Counts to frequencies. Typically, a permille frequency is returned. In other words, the resulting number indicates the expected mutation count if the genome had a total of 1000 mutations. This way, the MutSignatures analysis will be less biased toward the hyper-mutator samples.

**Usage**

```r
frequencize(countMatrix, permille = TRUE)
```

**Arguments**

- `countMatrix`: numeric matrix of mutation counts
- `permille`: logical, shall the permille conversion be used instead of the standard frequency

**Details**

This function is part of the user-interface set of tools included in mutSignatures. This is an exported function.

**Value**

list including `colSums` (mutation burden of each sample) and `freqs` (matrix of frequencies)

**Author(s)**

Damiano Fantini, <damiano.fantini@gmail.com>
References

More information and examples about mutational signature analysis can be found here:

1. GitHub Repo: https://github.com/dami82/mutSignatures/
2. More info and examples about the mutSignatures R library: https://www.data-pulse.com/dev_site/mutsignatures/
4. Oncogene paper, Mutational Signatures Operative in Bladder Cancer: https://www.nature.com/articles/s41388-017-0099-6

Examples

```r
A <- cbind(c(7, 100, 90, 1000), c(1, 3, 5, 9))
fA <- mutSignatures$frequencize(A)
fA$freqs
```

---

**getCosmicSignatures**

Obtain COSMIC mutational Signatures.

**Description**

Obtain latest mutational Signature definitions from COSMIC. For more info, please visit: https://cancer.sanger.ac.uk/cosmic/

**Usage**

```r
getCosmicSignatures(forceUseMirror = FALSE, asMutSign = TRUE)
```

**Arguments**

- `forceUseMirror` logical, shall signatures be downloaded from a mirror. Set to TRUE if the COSMIC server goes down.
- `asMutSign` logical, shall data be returned as a mutSignatures-class object. Defaults to TRUE

**Details**

This function is part of the user-interface set of tools included in mutSignatures. This is an exported function.

**Value**

an object storing COSMIC mutational signature data

**Author(s)**

Damiano Fantini, <damiano.fantini@gmail.com>
getCounts

References

More information and examples about mutational signature analysis can be found here:

1. GitHub Repo: https://github.com/dami82/mutSignatures/
2. More info and examples about the mutSignatures R library: https://www.data-pulse.com/dev_site/mutsignatures/
4. Oncogene paper, Mutational Signatures Operative in Bladder Cancer: https://www.nature.com/articles/s41388-017-0099-6

getCounts

Method getCounts.

Description

Retrieve mutation counts from an object.

Usage

getCounts(x)

# S4 method for signature 'mutationCounts'
getCounts(x)

Arguments

x an object to extract Mutation counts from, i.e. a mutationCounts-class object

getFwkParam

Method getFwkParam.

Description

Retrieve the list of parameters used for running a Mutation Signature Analysis.

Usage

getFwkParam(x, label)

# S4 method for signature 'mutFrameworkParams,character'
getFwkParam(x, label)

Arguments

x a mutFrameworkParams-class object
label string, corresponding to the parameter name to extract
getMutationTypes \hspace{1cm} \textit{Method getMutationTypes.}

\textbf{Description}

Retrieve the list of mutation types from an object.

\textbf{Usage}

getMutationTypes(x)

\texttt{## S4 method for signature 'mutationSignatures' getMutationTypes(x)}

\texttt{## S4 method for signature 'mutationCounts' getMutationTypes(x)}

\textbf{Arguments}

\texttt{x} \hspace{1cm} \text{an object to extract Mutation types from, i.e. a mutationSignatures-class or a mutationCounts-class object}

getSampleIdentifiers \hspace{1cm} \textit{Method getSampleIdentifiers.}

\textbf{Description}

Retrieve the list of sample identifiers from an object.

\textbf{Usage}

getSampleIdentifiers(x)

\texttt{## S4 method for signature 'mutationCounts' getSampleIdentifiers(x = "mutationCounts")}

\texttt{## S4 method for signature 'mutSignExposures' getSampleIdentifiers(x)}

\textbf{Arguments}

\texttt{x} \hspace{1cm} \text{an object to extract Mutation types from, i.e. a mutationCounts-class or a mutSignExposures-class object}
**getSignatureIdentifiers**

*Method getSignatureIdentifiers.*

---

**Description**

Retrieve the list of signature identifiers from an object.

**Usage**

```r
getSignatureIdentifiers(x)
```

```r
## S4 method for signature 'mutSignExposures'
getSignatureIdentifiers(x)
```

```r
## S4 method for signature 'mutationSignatures'
getSignatureIdentifiers(x)
```

**Arguments**

`x`  
an object to extract Signature Identifiers from, i.e. a mutSignExposures-class or a mutationSignatures-class object

---

**importVCFfiles**  
*Import Mutation data from VCF files.*

---

**Description**

Import Mutation data from VCF files. The columns are expected in the following order: c("CHROM", "POS", "ID", "REF", "ALT", "QUAL", "FILTER", "INFO", "FORMAT"). Optional columns can be present to inform about sample ID or other info.

**Usage**

```r
importVCFfiles(vcfFiles, sampleNames = NULL)
```

**Arguments**

`vcfFiles`  
character vector, includes the names of the VCF files to be analyzed

`sampleNames`  
character vector with alternative sample names (otherwise, VCF file names will be used to identify each sample).

**Details**

This function is part of the user-interface set of tools included in mutSignatures. This is an exported function.
Value

a concatenated data.frame with all variants found in the input VCF files. Sample ID is stored in the "SAMPLEID" column.

Author(s)

Damiano Fantini, <damiano.fantini@gmail.com>

References

More information and examples about mutational signature analysis can be found here:

1. GitHub Repo: https://github.com/dami82/mutSignatures/
2. More info and examples about the mutSignatures R library: https://www.data-pulse.com/dev_site/mutsignatures/
4. Oncogene paper, Mutational Signatures Operative in Bladder Cancer: https://www.nature.com/articles/s41388-017-0099-6

matchSignatures

Match Mutational Signatures.

Description

Analyze the similarity between mutational signatures from different analyses/runs. This function can be helpful to match de novo extracted signatures with previously described signatures (such as COSMIC), or to reveal signatures that can be identified with alternative NMF algorithms, or that may be due to an algorithm bias.

Usage

matchSignatures(
  mutSign,
  reference = NULL,
  method = "cosine",
  threshold = 0.5,
  plot = "TRUE"
)

Arguments

mutSign a mutationSignatures object
reference a mutationSignatures object. If NULL, COSMIC signatures will be retrieved
method distance method used to compute similarity (1 - distance)
threshold signal (similarity) upper threshold for maxing the signal
plot logical, shall a heatmap be plotted
**Details**

This function is part of the user-interface set of tools included in mutSignatures. This is an exported function.

**Value**

list, including distance matrix and a heatmap plot

**Author(s)**

Damiano Fantini, <damiano.fantini@gmail.com>

**References**

More information and examples about mutational signature analysis can be found here:

1. **GitHub Repo**: https://github.com/dami82/mutSignatures/
2. **More info and examples** about the mutSignatures R library: https://www.data-pulse.com/dev_site/mutsignatures/
4. **Oncogene paper**, Mutational Signatures Operative in Bladder Cancer: https://www.nature.com/articles/s41388-017-0099-6

---

**msigPlot**

*Method msigPlot.*

**Description**

Generate standard plots using data from mutsignature-class objects.

**Usage**

```r
msigPlot(x, ...)

## S4 method for signature 'mutationSignatures'
msigPlot(x, ...)

## S4 method for signature 'mutationCounts'
msigPlot(x, ...)

## S4 method for signature 'mutSignExposures'
msigPlot(x, ...)
```
mutationCounts-class

Class mutationCounts.

Description

Class mutationCounts defines objects storing Mutation COunts data.

Usage

## S4 method for signature 'mutationCounts'
initialize(.Object, x, muts, samples)

Arguments

/Object/ the mutationCounts object being built
x data.frame including mutation count values for each biological sample
muts data.frame including information about mutation types
samples data.frame including information about sample identifiers (unique names)

Slots

counts data.frame including information about mutation counts
mutTypes data.frame including information about mutation types
sampleId data.frame including information about sample identifiers

Author(s)

Damiano Fantini <damiano.fantini@gmail.com>
mutationSignatures-class

Class mutationSignatures.

Description

Class mutationSignatures defines objects storing Mutational Signatures data.

Usage

## S4 method for signature 'mutationSignatures'
initialize(.Object, x, muts, signNames)

Arguments

- .Object: the mutationSignatures object being built
- x: data.frame including frequency data of multiple mutation signatures
- muts: data.frame including information about mutation types
- signNames: data.frame including information about mutation signature names (unique identifiers)

Slots

- mutationFreq: data.frame including information about mutation frequencies
- mutTypes: data.frame including information about mutation types
- signatureId: data.frame including information about mutation signature Identifiers

Author(s)

Damiano Fantini <damiano.fantini@gmail.com>

mutFrameworkParams-class

Class mutFrameworkParams.

Description

Class mutFrameworkParams defines objects including the set of parameters used for running a Mutational Signature Analysis.

Usage

## S4 method for signature 'mutFrameworkParams'
initialize(.Object, params)
mutSigData

Arguments

.params Object the mutFrameworkParams object being built
.params list including values for a set of mutFramework params

Slots

.params list including the set of parameters used for running a Mutational Signature Analysis

Author(s)

Damiano Fantini <damiano.fantini@gmail.com>

Description

A series of objects, including collections of DNA mutations from 50 Bladder cancer samples, as well as mutational signatures extracted from the same samples. Mutation catalogs were obtained from a TCGA bladder cancer dataset (data available from the BROAD Institute). Original sample IDs were shuffled and then re-encoded. Data are available in different formats, and can be used as input for running mutational signature analyses.

Usage

data("mutSigData")

Format

A list with 6 elements. Each element is a different type of mutSignatures input/data:

- **inputA** data.frame with 10401 rows and 4 columns. DNA mutation data mimicking a TCGA dataset downloaded using TCGAretriever/cBio
- **inputB** data.frame with 13523 rows and 12 columns. DNA mutation data mimicking a TCGA MAF file
- **inputC** data.frame with 13523 rows and 11 columns. DNA mutation data mimicking a VCF file decorated with a SAMPLEID column
- **inputC.txt** data.frame with 13523 rows and 11 columns. DNA mutation data mimicking a VCF file decorated with a SAMPLEID column
- **inputD** data.frame with 13487 rows and 56 columns. DNA mutation data mimicking a set of VCF files casted into a 2D matrix (samples as columns)
- **inputS** list including data for silhouette plot generation (used in the vignette)
- **blcaMUTS** data.frame with 96 rows and 50 columns. A table of DNA mutation counts (rows are mutation types; columns are samples)
- **blcaSIGS** data.frame with 96 rows and 8 columns. Set of 8 mutational signatures (rows are mutation types; columns are signatures)
- **.addOn** list of add-on functions (executed only upon request, not evaluated; these may require manual installation of external libraries from Bioconductor or GitHub)
mutSignExposures-class

Details

Examples and more information are available in the vignette, as well as at the following URL: https://www.data-pulse.com/dev_site/mutsignatures/

Source

BLCA data were downloaded from http://gdac.broadinstitute.org/ and then further processed, modified, and formatted.

Examples

data(mutSigData)
print(mutSigData$input.A[1:6,])

mutSignExposures-class

Class mutSignExposures.

Description

Class mutSignExposures defines objects storing information about Exposures of biological samples to Mutational Signatures.

Usage

## S4 method for signature 'mutSignExposures'
initialize(.Object, x, samples, signNames)

Arguments

/Object/ the mutSignExposures object being built
.x/ data.frame including numeric values of exposures to mutational signatures
.samples/ data.frame including information about biological sample identifiers (unique names)
.signNames/ data.frame including information about mutational signature identifiers

Slots

.exposures/ data.frame including information about exposures
.sampleId/ data.frame including information about sample identifiers
.signatureId/ data.frame including information about signature identifiers

Author(s)

Damiano Fantini <damiano.fantini@gmail.com>


**plotMutTypeProfile**  

*Plot Mutation Signature Profiles.*

**Description**

Build a barplot to visualize the relative abundance of mutation counts in a mutational signature or biological sample of interest.

**Usage**

```r
plotMutTypeProfile(
  mutCounts,
  mutLabs,
  freq = TRUE,
  ylim = "auto",
  ylab = "Fraction of Variants",
  xlab = "Sequence Motifs",
  xaxis_cex = 0.475,
  cols = c("#4eb3d3", 
             
            "#040404",
            
            "#b30000",
            
            "#bdbdbd",
            
            "#41ab5d",
            
            "#dd3497"),
  main = "MutType Profile"
)
```

**Arguments**

- **mutCounts**  
  data.frame including mutation types counts or frequencies, such as a data.frame of mutation counts from samples, or mutation type frequencies from a mutational signature.

- **mutLabs**  
  character vector, labels to be used for the mutation types

- **freq**  
  logical, shall frequency be plotted rather than counts. Defaults to TRUE

- **ylim**  
  values used for ylim. Defaults to "auto" (ylim automatically set)

- **ylab**  
  string, used as y-axis title. Defaults to "Fraction of Variants"

- **xlab**  
  string, used as x-axis title. Defaults to "Sequence Motifs"

- **xaxis_cex**  
  numeric, cex value for the xaxis

- **cols**  
  character vector, indicates the colors to be used for the bars. It typically requires 6 colors.

- **main**  
  string, title of the plot. Defaults to "MutType Profile"

**Details**

This function is part of the user-interface set of tools included in mutSignatures. This is an exported function.

**Value**

NULL. A plot is printed to the active device.
plotSignExposures

Author(s)

Damiano Fantini, <damiano.fantini@gmail.com>

References

More information and examples about mutational signature analysis can be found here:

1. GitHub Repo: [https://github.com/dami82/mutSignatures/](https://github.com/dami82/mutSignatures/)
2. More info and examples about the mutSignatures R library: [https://www.data-pulse.com/dev_site/mutsignatures/](https://www.data-pulse.com/dev_site/mutsignatures/)
3. Sci Rep paper, introducing mutS: [https://www.nature.com/articles/s41598-020-75062-0/](https://www.nature.com/articles/s41598-020-75062-0/)
4. Oncogene paper, Mutational Signatures Operative in Bladder Cancer: [https://www.nature.com/articles/s41388-017-0099-6](https://www.nature.com/articles/s41388-017-0099-6)

---

Plot Signature Exposure Profiles.

Description

Build a barplot to visualize exposures to mutation signatures.

Usage

```r
plotSignExposures(mutCount, top = 50)
```

Arguments

- `mutCount`: a data.frame including mutation Counts
- `top`: integer, max number of samples to include in the plot

Details

This function is part of the user-interface set of tools included in mutSignatures. This is an exported function.

Value

a plot (ggplot2 object)

Author(s)

Damiano Fantini, <damiano.fantini@gmail.com>
References

More information and examples about mutational signature analysis can be found here:

1. More info and examples about the mutSignatures R library: https://www.data-pulse.com/dev_site/mutsignatures/
2. Sci Rep paper, introducing mutS: https://www.nature.com/articles/s41598-020-75062-0/

---

prelimProcessAssess  Run a Preliminary Process Assess analysis.

Description

This function is an attempt to analyze the relationship between error and k. In other words, the goal of prelimProcessAssess is to visualize the reduction in the error/residuals.

Usage

prelimProcessAssess(
  input,
  maxProcess = 6,
  approach = "counts",
  plot = TRUE,
  verbose = TRUE
)

Arguments

input  a mutationCounts-class object
maxProcess  integer, maximum k to test
approach  sting, "counts" or "freq"
plot  logical, shall a plot be printed to the active device
verbose  logical, info about the ongoing analysis be messaged/printed to console

Details

This function is part of the user-interface set of tools included in mutSignatures. This is an exported function.

Value

a data.frame showing the estimated total error with respect to the range of k values

Author(s)

Damiano Fantini, <damiano.fantini@gmail.com>
References

More information and examples about mutational signature analysis can be found here:

1. GitHub Repo: https://github.com/dami82/mutSignatures/
2. More info and examples about the mutSignatures R library: https://www.data-pulse.com/dev_site/mutsignatures/
4. Oncogene paper, Mutational Signatures Operative in Bladder Cancer: https://www.nature.com/articles/s41388-017-0099-6

processVCFData

Process VCF data.

Description

Check, annotate, and process variants imported from a list of VCF files, so that it can be used to run a mutational signature analysis

Usage

processVCFData(
  vcfData,  # data.frame, includes mutation data from 2 or more samples
  BSGenomeDb,  # a BSGenomeDb-class object storing the genomic sequences and coordinates
  chr_colName = "CHROM",  # string, name of the column including the chromosome (seq) name. Defaults to "CHROM"
  pos_colName = "POS",  # string, name of the column including the genomic coordinates/position. Defaults to "POS"
  ref_colName = "REF",  # string, name of the column including the reference nucleotide. Defaults to "REF"
  alt_colName = "ALT",  # string, name of the column including the variant nucleotide. Defaults to "ALT"
  sample_colName = NULL,  # string, name of the column including the sample ID. Can be NULL
  nucl_contextN = 3,  # integer, span (in nucleotides) of the context around the variants. Defaults to 3
  verbose = TRUE  # logical, shall information about the ongoing analysis be printed to console
)

Arguments

vcfData BSGenomeDb chr_colName pos_colName ref_colName alt_colName sample_colName nucl_contextN verbose
removeMismatchMut

Details
This function is part of the user-interface set of tools included in mutSignatures. This is an exported function.

Value
a data.frame including processed variants from VCF files

Author(s)
Damiano Fantini, damiano.fantini@gmail.com

References
More information and examples about mutational signature analysis can be found here:
1. GitHub Repo: https://github.com/dami82/mutSignatures/
2. More info and examples about the mutSignatures R library: https://www.data-pulse.com/dev_site/mutsignatures/
4. Oncogene paper, Mutational Signatures Operative in Bladder Cancer: https://www.nature.com/articles/s41388-017-0099-6

removeMismatchMut
Remove Mismatched Mutations.

Description
Remove mutation types that do not match the expected nucleotidic context.

Usage
removeMismatchMut(
  mutData,
  refMut_colName = "mutation",
  context_colName = "context",
  refMut_format = "N>N"
)

Arguments
mutData data.frame including mutation data, as well as the nucleotide context around the mutated position
refMut_colName string, name of the column storing REF and VAR data. Defaults to "N>N"
context_colName string, name of the column storing nucleotide context around the variant.
refMut_format string, format of mutation types. Defaults to "N>N"
Details

This function is part of the user-interface set of tools included in mutSignatures. This is an exported function.

Value

filtered data.frame

Author(s)

Damiano Fantini, <damiano.fantini@gmail.com>

References

More information and examples about mutational signature analysis can be found here:

1. GitHub Repo: https://github.com/dami82/mutSignatures/
2. More info and examples about the mutSignatures R library: https://www.data-pulse.com/dev_site/mutsignatures/
4. Oncogene paper, Mutational Signatures Operative in Bladder Cancer: https://www.nature.com/articles/s41388-017-0099-6

Examples

```r
x <- mutSignatures:::getTestRunArgs("removeMismatchMut")
y <- mutSignatures:::removeMismatchMut(x,
  refMut_colName = "REF",
  context_colName = "context",
  refMut_format = "N")
y
```

---

**resolveMutSignatures** Resolve Mutation Signatures.

Description

If Mutation signatures are known (such as COSMIC signatures), we can estimate the contribution of each signature in different samples. This functions used a matrix of mutation counts and a matrix of mutation signatures, and estimates Exposures to Mutational Signature of each sample.

Usage

```r
resolveMutSignatures(mutCountData, signFreqData, byFreq = TRUE)
```
Arguments

- `mutCountData`: object storing mutation counts
- `signFreqData`: object storing mutation signatures
- `byFreq`: logical, shall exposures be estimated on per_mille normalized counts

Details

This function is part of the user-interface set of tools included in mutSignatures. This is an exported function.

Value

A list of objects including data about exposures to mutational signatures

Author(s)

Damiano Fantini, <damiano.fantini@gmail.com>

References

More information and examples about mutational signature analysis can be found here:

1. GitHub Repo: https://github.com/dami82_mutSignatures/
2. More info and examples about the mutSignatures R library: https://www.data-pulse.com/dev_site/mutsignatures/
4. Oncogene paper, Mutational Signatures Operative in Bladder Cancer: https://www.nature.com/articles/s41388-017-0099-6

Examples

```r
x <- mutSignatures:::getTestRunArgs("resolveMutSignatures")
y <- mutSignatures::resolveMutSignatures(mutCountData = x$muts, signFreqData = x$sigs)
y
```

Description

The `revCompl` function computes the reverse complement of a DNA sequence. It can return either the reverse complement sequence or only the reverse sequence (or only the complement) as specified by the `byFreq` argument.
revCompl

Usage

revCompl(DNAseq, rev = TRUE, compl = TRUE)

Arguments

DNAseq character vector of DNA sequences
rev logical, shall the reverse sequence be computed
compl logical, shall the complementary sequence be computed

Details

This function is part of the user-interface set of tools included in mutSignatures. This is an exported function.

Value

a character vector including transformed DNA sequences

Author(s)

Damiano Fantini, <damiano.fantini@gmail.com>

References

More information and examples about mutational signature analysis can be found here:

1. GitHub Repo: https://github.com/dami82/mutSignatures/
2. More info and examples about the mutSignatures R library: https://www.data-pulse.com/dev_site/mutsignatures/
4. Oncogene paper, Mutational Signatures Operative in Bladder Cancer: https://www.nature.com/articles/s41388-017-0099-6

Examples

A <- c("TAACCG", "CTCGA", "CNNA")
mutSignatures::revCompl(A)
setFwkParam Method setFwkParam.

Description
Set or update one of the parameters in a mutFrameworkParams-class object. Individual parameters can be set or updated, by passing the parameter label, and the corresponding parameter value.

Usage
setFwkParam(x, label, value)

## S4 method for signature 'mutFrameworkParams,character'
setFwkParam(x, label, value)

Arguments
- **x**: an object to extract Signature Identifiers from, i.e. a mutSignExposures-class
- **label**: string corresponding to the parameter label to be updated
- **value**: new value (string or numeric) of the parameter to be updated

setMutClusterParams Set Parameters for Extracting Mutational Signatures.

Description
Create an object including all parameters required for running the mutSignatures framework.

Usage
setMutClusterParams(
    num_processesToExtract = 2,
    num_totIterations = 10,
    num_parallelCores = 1,
    thresh_removeWeakMutTypes = 0.01,
    thresh_removeLastPercent = 0.07,
    distanceFunction = "cosine",
    num_totReplicates = 100,
    eps = 2.2204e-16,
    stopconv = 20000,
    niter = 1e+06,
    guided = TRUE,
    debug = FALSE,
    approach = "freq",
    stopRule = "DF",
)
```r
setMutClusterParams

  algorithm = "brunet",
  logIterations = "lite",
  seed = 12345

Arguments

num_processesToExtract
  integer, number of mutational signatures to extract
num_totIterations
  integer, total number of iterations (bootstrapping)
num_parallelCores
  integer, number of cores to use for the analysis
thresh_removeWeakMutTypes
  numeric, threshold for filtering out under-represented mutation types
thresh_removeLastPercent
  numeric, threshold for removing outlier iteration results
distanceFunction
  string, method for calculating distances. Default method is "cosine"
um_totReplicates
  integer, number of replicates while checking stability
eps
  numeric, close-to-zero positive numeric value for replacing zeros and preventing
  negative values to appear in the matrix during NMF
stopconv
  integer, max number of stable iterations before termination. Defaults to 20000.
niter
  integer, max number of iterations to run. Defaults to 100000
guided
  logical, shall clustering be guided to improve aggregation upon bootstrapping
debug
  logical, shall the analysis be run in DEBUG mode
approach
  string, indicating whether to model absolute counts ("counts") or per_mille fre-
  quency ("freq"). Defaults to "freq".
stopRule
  string, use the sub-optimal termination rule ("AL") from the WTSI package
  (actually, iterations won’t terminate, so niter will most certainly reached) or
  our efficient termination rule ("DF"). Defaults to "DF". The "AL" option is
  implemented for compatibility reasons, but not recommended.
algorithm
  string, algorithm to be used for NMF. Set to "brunet", or "alexa" for using the
  standard algorithm (Brunet’s), otherwise the alternative "chihjen" algorithm will
  be used.
logIterations
  string indicating if storing and returning all intermediates, or only final results.
  Defaults to "lite", i.e. returns full output and limited intermediates. Alternati-
  vely, set to "full".
seed
  integer, seed to set for reproducibility

Value

Object including all parameters for running the analysis
```
Author(s)
Damiano Fantini, <damiano.fantini@gmail.com>

References
More information and examples about mutational signature analysis can be found here:

1. GitHub Repo: https://github.com/dami82/mutSignatures/
2. More info and examples about the mutSignatures R library: https://www.data-pulse.com/dev_site/mutsignatures/
4. Oncogene paper, Mutational Signatures Operative in Bladder Cancer: https://www.nature.com/articles/s41388-017-0099-6
5. WTSI framework: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3588146/

Examples
library(mutSignatures)
# defaults params
A <- setMutClusterParams()
A
# A second example, set num_processes
B <- setMutClusterParams(num_processesToExtract = 5)
B

definitions and examples about mutational signature analysis can be found here:

1. GitHub Repo: https://github.com/dami82/mutSignatures/
2. More info and examples about the mutSignatures R library: https://www.data-pulse.com/dev_site/mutsignatures/
4. Oncogene paper, Mutational Signatures Operative in Bladder Cancer: https://www.nature.com/articles/s41388-017-0099-6
5. WTSI framework: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3588146/

Examples
library(mutSignatures)
# defaults params
A <- setMutClusterParams()
A
# A second example, set num_processes
B <- setMutClusterParams(num_processesToExtract = 5)
B

show.mutationCounts-method

Show method of the mutationCounts Class.

Description
Show method of the mutationCounts Class.
Print method of the mutationCounts Class.

Usage
## S4 method for signature 'mutationCounts'
show(object)

## S4 method for signature 'mutationCounts'
print(x)

Arguments
object the mutationCounts object being shown
x the mutationCounts object being printed
**Description**

Show method of the mutationSignatures Class.
Print method of the mutationSignatures Class.

**Usage**
```
## S4 method for signature 'mutationSignatures'
show(object)

## S4 method for signature 'mutationSignatures'
print(x)
```

**Arguments**

- `object` the mutationSignatures object being shown
- `x` the mutationSignatures object being printed

**Description**

Show method of the mutFrameworkParams Class.
Print method of the mutFrameworkParams Class.

**Usage**
```
## S4 method for signature 'mutFrameworkParams'
show(object)

## S4 method for signature 'mutFrameworkParams'
print(x)
```

**Arguments**

- `object` the mutFrameworkParams object being shown
- `x` the mutFrameworkParams object being printed
show,mutSignExposures-method

*Show method of the mutSignExposures Class.*

**Description**

Show method of the mutSignExposures Class.
Print method of the mutSignExposures Class.

**Usage**

```r
## S4 method for signature 'mutSignExposures'
show(object)
```

```r
## S4 method for signature 'mutSignExposures'
print(x)
```

**Arguments**

- `object`: the mutSignExposures object being shown
- `x`: the mutSignExposures object being printed

**silhouetteMLB**

*Silhouette Analysis.*

**Description**

Analyze the clustering quality and generate a Silhouette Plot.

**Usage**

```
silhouetteMLB(data, fac, method = "cosine", plot = TRUE)
```

**Arguments**

- `data`: numeric matrix
- `fac`: clustering factor
- `method`: method to be used as distance function. Defaults to c(“cosine”)  
- `plot`: logical, shall a barplot showing the cluster silhouettes be printed

**Value**

numeric vector including the silhouette values of the data points in the input matrix
simplifySignatures

Author(s)

Damiano Fantini, <damiano.fantini@gmail.com>

References

More information and examples about mutational signature analysis can be found here:

1. GitHub Repo: https://github.com/dami82/mutSignatures/
2. More info and examples about the mutSignatures R library: https://www.data-pulse.com/dev_site/mutsignatures/
4. Oncogene paper, Mutational Signatures Operative in Bladder Cancer: https://www.nature.com/articles/s41388-017-0099-6
5. Silhouette analysis in R: http://www.biotechworld.it/bioinf/2017/01/20/translating-matlabs-silhouette

Examples

library(mutSignatures)
x <- mutSignatures:::getTestRunArgs("silhouetteMLB")
y <- silhouetteMLB(data = x$data, fac = x$fac)
y

simplifySignatures  Simplify Mutational Signatures.

Description

This function is useful when working with non-standard mutation types, such as tetra-nucleotide mutation types or mutation types with long/complex context. The goal of this function is to aggregate together mutations that can be simplified because of a common mutation core. For example, mutation types AA[A>T]A, TA[A>T]A, CA[A>T]A, and GA[A>T]A can be simplified to the core tri-nucleotide mutation A[A>T]A. This function identifies mergeable mutation types, and aggregates the corresponding counts/freqs.

Usage

simplifySignatures(x, asMutationSignatures = TRUE)

Arguments

x  a mutationSignatures-class object
asMutationSignatures  logical, shall the results be returned as a mutationSignatures-class object
sortByMutations

Details
This function is part of the user-interface set of tools included in mutSignatures. This is an exported function.

Value
object including simplified mutational signatures data

Author(s)
Damiano Fantini, <damiano.fantini@gmail.com>

References
More information and examples about mutational signature analysis can be found here:

1. GitHub Repo: https://github.com/dami82/mutSignatures/
2. More info and examples about the mutSignatures R library: https://www.data-pulse.com/dev_site/mutsignatures/
4. Oncogene paper, Mutational Signatures Operative in Bladder Cancer: https://www.nature.com/articles/s41388-017-0099-6

Examples
```
A <- data.frame(Sig1=1:5, Sig2=5:1, Sig3=1:5)
A <- A/apply(A, 2, sum)
A <- mutSignatures::as.mutation.signatures(A)
mutSignatures::simplifySignatures(x = A)
```

sortByMutations

Sort Data by Mutation Type.

Description
Reorder a mutationSignatures, mutationCounts, data.frame, or matrix object by sorting entries by mutation type.

Usage
```
sortByMutations(x)
```

Arguments

x an object storing mutation count data
Details

This function is part of the user-interface set of tools included in mutSignatures. This is an exported function.

Value

an object of the same class as x, with entries sorted according to mutation types.

Author(s)

Damiano Fantini, <damiano.fantini@gmail.com>

References

More information and examples about mutational signature analysis can be found here:

1. GitHub Repo: https://github.com/dami82/mutSignatures/
2. More info and examples about the mutSignatures R library: https://www.data-pulse.com/dev_site/mutsignatures/
4. Oncogene paper, Mutational Signatures Operative in Bladder Cancer: https://www.nature.com/articles/s41388-017-0099-6

Examples

```r
A <- data.frame(S1=1:5, S2=5:1, S3=1:5)
mutSignatures::sortByMutations(A)
```

```r
# molten data.frame

<table>
<thead>
<tr>
<th>rowLab</th>
<th>colLab</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>sample</td>
<td>feature</td>
<td>count</td>
</tr>
</tbody>
</table>
```

Description

Prepare a molten data.frame starting from a mutation count matrix. Mutation types (rows) are counted for each sample (cols). The results are returned in a 3-column data.frame.

Usage

```r
table2df(dataMatrix, rowLab = "sample", colLab = "feature", valueLab = "count")
```

Arguments

dataMatrix a numeric matrix including mutation counts
rowLab string, name for the column that will be storing row IDs, typically sample IDs
collab string, name for the column that will be storing column IDs, typically sample IDs
valueLab string, name for the column that will be storing mutation count values
Details
This function is part of the user-interface set of tools included in mutSignatures. This is an exported function.

Value
data.frame storing mutation counts by sample

Author(s)
Damiano Fantini, <damiano.fantini@gmail.com>

References
More information and examples about mutational signature analysis can be found here:
1. More info and examples about the mutSignatures R library: https://www.data-pulse.com/dev_site/mutsignatures/
2. Sci Rep paper, introducing mutS: https://www.nature.com/articles/s41598-020-75062-0/
3. Oncogene paper, Mutational Signatures Operative in Bladder Cancer: https://www.nature.com/articles/s41388-017-0099-6

Examples
A <- cbind("A>G"=c(5,10),"A>T"=c(3,20),"A>C"=c(15,0))
rownames(A) = c("Smpl1", "Smpl2")
mutSignatures::table2df(A)

---

Description
Subset a mutationCounts-class object.

Usage
## S4 method for signature 'mutationCounts,numeric,ANY,ANY'
x[i]

Arguments
x a mutationCounts-class object to subset
i numeric, indeces of the elements to be extracted
[.,mutationSignatures,numeric,ANY,ANY-method

Subset a mutationSignatures-class object.

Description

Subset a mutationSignatures-class object.

Usage

## S4 method for signature 'mutationSignatures,numeric,ANY,ANY'

x[i]

Arguments

x  
a mutationSignatures-class object to subset

i  
numeric, indeces of the elements to be extracted

[.,mutSignExposures,numeric,ANY,ANY-method

Subset a mutSignExposures-class object.

Description

Subset a mutSignExposures-class object.

Usage

## S4 method for signature 'mutSignExposures,numeric,ANY,ANY'

x[i]

Arguments

x  
a mutSignExposures-class object to subset

i  
numeric, indeces of the elements to be extracted
Index

* datasets
  mutSigData, 26
  [,mutSignExposures,numeric,ANY,ANY-method, 45
  [,mutSignExposures,numeric-method
  ([,mutSignExposures,numeric,ANY,ANY-method), 45
  [,mutationCounts,numeric,ANY,ANY-method, 44
  [,mutationCounts,numeric-method
  ([,mutationCounts,numeric,ANY,ANY-method), 44
  [,mutationSignatures,numeric,ANY,ANY-method, 45
  [,mutationSignatures,numeric-method
  ([,mutationSignatures,numeric,ANY,ANY-method), 45
  as.data.frame,mutationCounts-method, 4
  as.data.frame,mutationSignatures-method, 4
  as.data.frame,mutSignExposures-method, 5
  as.list,mutationSignatures-method, 5
  as.list,mutFrameworkParams-method, 6
  as.matrix,mutationCounts-method, 6
  as.mutation.counts, 7
  as.mutation.counts,data.frame,ANY,ANY-method
  (as.mutation.counts), 7
  as.mutation.counts,data.frame-method
  (as.mutation.counts), 7
  as.mutation.signatures, 7
  as.mutation.signatures,data.frame-method
  (as.mutation.signatures), 7
  as.mutsign.exposures, 8
  as.mutsign.exposures,data.frame,logical-method
  (as.mutsign.exposures), 8
  attachContext, 8
  attachMutType, 9
  cbind2, 11
  cbind2,mutationSignatures,mutationSignatures-method
  (cbind2), 11
  coerceObj, 12
  coerceObj,data.frame,character-method
  (coerceObj), 12
  coerceObj,mutationCounts,character-method
  (coerceObj), 12
  coerceObj,mutationSignatures,character-method
  (coerceObj), 12
  coerceObj,mutFrameworkParams,character-method
  (coerceObj), 12
  coerceObj,mutSignExposures,character-method
  (coerceObj), 12
  countMutTypes, 12
  decipherMutationalProcesses, 14

  extractXvarlinkData, 15
  filterSNV, 16
  frequencize, 17
  getCosmicSignatures, 18
  getCounts, 19
  getCounts,mutationCounts-method
  (getCounts), 19
  getFwkParam, 19
  getFwkParam,mutFrameworkParams,character-method
  (getFwkParam), 19
  getMutationTypes, 20
  getMutationTypes,mutationCounts-method
  (getMutationTypes), 20
  getMutationTypes,mutationSignatures-method
  (getMutationTypes), 20
  getSampleIdentifiers, 20
  getSampleIdentifiers,mutationCounts-method
  (getSampleIdentifiers), 20
  getSampleIdentifiers,mutSignExposures-method
  (getSampleIdentifiers), 20
INDEX

getSignatureIdentifiers, 21
getSignatureIdentifiers, mutationSignatures-method
  (getSignatureIdentifiers), 21
getSignatureIdentifiers, mutSignExposures-method
  (getSignatureIdentifiers), 21
importVCFfiles, 21
initialize, mutationCounts-method
  (mutationCounts-class), 24
initialize, mutationSignatures-method
  (mutationSignatures-class), 25
initialize, mutFrameworkParams-method
  (mutFrameworkParams-class), 25
initialize, mutSignExposures-method
  (mutSignExposures-class), 27
matchSignatures, 22
msigPlot, 23
msigPlot, mutationCounts-method
  (msigPlot), 23
msigPlot, mutationSignatures-method
  (msigPlot), 23
msigPlot, mutSignExposures-method
  (msigPlot), 23
mutationCounts-class, 24
mutationSignatures-class, 25
mutFrameworkParams-class, 25
mutSigData, 26
mutSignatures-package, 3
mutSignExposures-class, 27
plotMutTypeProfile, 28
plotSignExposures, 29
prelimProcessAssess, 30
print, mutationCounts-method
  (show, mutationCounts-method), 38
print, mutationSignatures-method
  (show, mutationSignatures-method), 39
print, mutFrameworkParams-method
  (show, mutFrameworkParams-method), 39
print, mutSignExposures-method
  (show, mutSignExposures-method), 40
processVCFdata, 31
removeMismatchMut, 32
resolveMutSignatures, 33
setFwkParam, 36
setFwkParam, mutFrameworkParams, character, ANY-method
  (setFwkParam), 36
setFwkParam, mutFrameworkParams, character-method
  (setFwkParam), 36
setMutClusterParams, 36
show, mutationCounts-method, 38
show, mutationSignatures-method, 39
show, mutFrameworkParams-method, 39
show, mutSignExposures-method, 40
silhouetteMLB, 40
simplifySignatures, 41
sortByMutations, 42
table2df, 43