Package 'decompTumor2Sig'

July 27, 2018

Type Package

Title Decomposition of individual tumors into mutational signatures

Depends R(>= 3.3), ggplot2

Imports methods, Matrix, quadprog(>= 1.5-5), vcfR, GenomicRanges, stats, GenomicFeatures, Biostrings, BiocGenerics, S4Vectors, plyr, utils, graphics, BSgenome.Hsapiens.UCSC.hg19, TxDb.Hsapiens.UCSC.hg19.knownGene, VariantAnnotation, SummarizedExperiment

Suggests pmsignature, knitr, rmarkdown, BiocStyle

Version 1.3.0

Date 2018-07-26

Author Rosario M. Piro, Sandra Krueger

Maintainer Rosario M. Piro <rmpiro@gmail.com> or <r.piro@fu-berlin.de>

Description Uses quadratic programming to decompose the mutation catalog from an individual tumor sample into a set of given mutational signatures (either Alexandrov-model signatures or Shiraishi-model signatures), computing weights that reflect the contributions of the signatures to the mutation load of the tumor.

License GPL-2

URL http://rmpiro.net/decompTumor2Sig/

biocViews Software, SNP, Sequencing, DNASeq, GenomicVariation, SomaticMutation, BiomedicalInformatics, Genetics, BiologicalQuestion, StatisticalMethod

VignetteBuilder knitr

NeedsCompilation no

R topics documented:

`decompTumor2Sig-package` *decompTumor2Sig*

Description

The decompTumor2Sig package uses quadratic programming to decompose the somatic mutation catalog from an individual tumor sample (or multiple individual tumor samples) into a set of given mutational signatures (either of the "Alexandrov model" by Alexandrov et al, Nature 500(7463):415-421, 2013), or the "Shiraishi model" by Shiraishi et al, PLoS Genet 11(12):e1005657, 2015), thus computing weights (or "exposures") that reflect the contributions of the signatures to the mutation load of the tumor.

The package additionally provides helper functions to extract genomes (mutation catalogs) and signatures from objects provided by the `pmsignature` package (Shiraishi et al, 2015) or to read them from files.

Details

Package:	decompTumor2Sig
Type:	Package
Version:	1.3.0
Date:	2018-07-26
License:	GPL (>=2)

The package provides the following functions:

decomposeTumorGenomes():	determines the weights/contributions of a set of SIGNATURES to each of a set of individual tumor GENOMES.
loadGenomesFromVCF():	loads a genome or set of genomes from a Variant Call Format (VCF) file.
loadGenomesFromMPF():	loads a genome or set of genomes from a Mutation Position Format (MPF) file.
convertGenomesFromVRanges():	convert a genome or set of genomes from a `VariantAnnotation::VRanges` object.

getGenomesFromMutationFeatureData():	extracts the GENOMES from a `MutationFeatureData` object as provided by, for example, `pmsignature::readMPFile`.
loadShiraishiSignatures():	loads Shiraishi signatures from flat files.
getSignatureListFromEstimatedParameters():	extracts a set of SIGNATURES from an `EstimatedParameters` object as computed by `pmsignature::getPMSignature`.
loadAlexandrovSignatures():	loads Alexandrov signatures in the COSMIC format from a flat file or URL.
convertAlexandrov2Shiraishi():	converts a set of Alexandrov signatures to Shiraishi signatures.
computeExplainedVariance():	determines the variance explained by estimated signature contributions/exposures.
plotExplainedVariance():	plots the variance of the genome's mutation load that can be explained with an increasing number of signatures.
plotDecomposedContribution():	plot a the decomposition/exposures of a genome to the mutational signatures.
plotMutationDistribution():	plot a single signature or mutation frequency data for a single genome.
composeGenomesFromExposures():	(re-)construct tumor genome mutation frequencies from the signatures and their corrsponding exposures/contributions.
evaluateDecompositionQuality():	evaluates the quality of a decomposition by comparing the re-composed (=re-constructed) tumor genome mutation frequencies to those actually observed in the tumor genome.
determineSignatureDistances():	for a given target signature compute its distances to each of a set of other signatures.
mapSignatureSets():	find a mapping from one signature set to another.
downgradeShiraishiSignatures():	downgrades Shiraishi signatures by removing flanking bases and/or the transcription direction.

Author(s)

Rosario M. Piro, Sandra Krueger
Freie Universitaet Berlin
Maintainer: Rosario M. Piro
E-Mail: <r.piro@fu-berlin.de> or <rmpiro@gmail.com>

References

Krueger, Piro (2017) Identification of Mutational Signatures Active in Individual Tumors. NETTAB 2017 - Methods, Tools & Platforms for Personalized Medicine in the Big Data Era, October 16-18, 2017, Palermo, Italy. PeerJ Preprints 5:e3257v1, 2017.
http://rmpiro.net/decompTumor2Sig/

composeGenomesFromExposures
composeGenomesFromExposures

Description

(Re-)compose/construct tumor genome characteristics, or mutation frequencies, from the signatures and their corresponding exposures, or contributions. The (re-)composition is performed by computing the weighted sum of the mutational signatures, where the weights are to the exposures (=contributions) of the corresponding signatures. This can, for example, be used to verify that a decomposition obtained from `decomposeTumorGenomes` is meaningful.

Usage

```
composeGenomesFromExposures(exposures, signatures)
```

Arguments

`exposures`	(Mandatory) A single vector or list of vectors containing the estimated signature contributions/exposures as provided by the function `decomposeTumorGenomes`. A list of vectors is used if the (re-)composition shall be performed for multiple genomes. The number of elements of each exposure vector must correspond to the number of signatures (see below).
`signatures`	(Mandatory) The list of signatures (vectors, data frames or matrices) for which the exposures were obtained. Each of the list objects represents one mutational signature. Vectors are used for Alexandrov signatures, data frames or matrices for Shiraishi signatures.

Value

A list of "predicted" genomes, i.e., their mutational patterns computed as weighted sums of the mutational signatures, where the weights correspond to the exposures/contributions of the corresponding signatures.

Author(s)

Rosario M. Piro
Freie Universitaet Berlin
Maintainer: Rosario M. Piro
E-Mail: <r.piro@fu-berlin.de> or <rmpiro@gmail.com>

References

Krueger, Piro (2017) Identification of Mutational Signatures Active in Individual Tumors. NETTAB 2017 - Methods, Tools & Platforms for Personalized Medicine in the Big Data Era, October 16-18, 2017, Palermo, Italy. PeerJ Preprints 5:e3257v1, 2017.
http://rmpiro.net/decompTumor2Sig/

See Also

decompTumor2Sig
decomposeTumorGenomes

Examples

```
### get Alexandrov signatures from COSMIC
signatures <- loadAlexandrovSignatures()

### load preprocessed breast cancer genomes (object 'genomes') from
### Nik-Zainal et al (PMID: 22608084)
gfile <- system.file("extdata",
         "Nik-Zainal_PMID_22608084-genomes-Alexandrov_3bases.Rdata",
         package="decompTumor2Sig")
load(gfile)

### compute exposures
exposures <- decomposeTumorGenomes(genomes, signatures, verbose=FALSE)

### re-compose (predict) tumor genome features from exposures
predGenomes <- composeGenomesFromExposures(exposures, signatures)
```

`computeExplainedVariance`
computeExplainedVariance

Description

For a single genome or a set of genomes, the function computes the explained variance(s) of the estimated signature contributions/exposures.

Usage

```
computeExplainedVariance(exposures, signatures, genomes)
```

Arguments

`exposures`	(Mandatory) A single vector or list of vectors containing the estimated signature contributions/exposures as provided by the function `decomposeTumorGenomes`. A list of vectors is used if the explained variance shall be computed for multiple genomes. The number of exposure vectors must correspond to the number of genomes (see below). The number of elements of each exposure vector must correspond to the number of signatures (see below).
`signatures`	(Mandatory) The list of signatures (vectors, data frames or matrices) for which the exposures were obtained. Each of the list objects represents one mutational signature. Vectors are used for Alexandrov signatures, data frames or matrices for Shiraishi signatures.
`genomes`	(Mandatory) Can be either a vector, a data frame or a matrix (for an individual tumor genome), or a list of one of these object types (for multiple tumors). Each tumor genome must be of the same form as the '`signatures`' (see above).

Value

A numeric vector of explained variances, one for each genome.

Author(s)

Rosario M. Piro
Freie Universitaet Berlin
Maintainer: Rosario M. Piro
E-Mail: <r.piro@fu-berlin.de> or <rmpiro@gmail.com>

References

Krueger, Piro (2017) Identification of Mutational Signatures Active in Individual Tumors. NETTAB 2017 - Methods, Tools & Platforms for Personalized Medicine in the Big Data Era, October 16-18, 2017, Palermo, Italy. PeerJ Preprints 5:e3257v1, 2017.
http://rmpiro.net/decompTumor2Sig/

See Also

decompTumor2Sig
decomposeTumorGenomes
plotExplainedVariance

Examples

```
### get Alexandrov signatures from COSMIC
signatures <- loadAlexandrovSignatures()

### load preprocessed breast cancer genomes (object 'genomes') from
### Nik-Zainal et al (PMID: 22608084)
gfile <- system.file("extdata",
         "Nik-Zainal_PMID_22608084-genomes-Alexandrov_3bases.Rdata",
         package="decompTumor2Sig")
load(gfile)

### compute exposures
exposures <- decomposeTumorGenomes(genomes, signatures, verbose=FALSE)

### compute explained variance for the tumor genomes
computeExplainedVariance(exposures, signatures, genomes)
```

`convertAlexandrov2Shiraishi` *convertAlexandrov2Shiraishi*

Description

Converts a set Alexandrov signatures (loaded with `loadAlexandrovSignatures`) to the Shiraishi format, summing the respective frequencies of base changes, and upstream and downstream flanking bases. The resulting Shiraishi signatures don't provide information on the transcription strand, as this is not part of the Alexandrov signatures.

[Attention: this conversion is experimental and the applicability of Shiraishi signatures derived from Alexandrov signatures has not been explored!]

Usage

```
convertAlexandrov2Shiraishi(signatures)
```

Arguments

signatures	(Mandatory) A list of Alexandrov signatures with named elements as produced by loadAlexandrovSignatures.

Value

A list of Shiraishi signatures that can be used for decomposeTumorGenomes.

Author(s)

Rosario M. Piro, Sandra Krueger
Freie Universitaet Berlin
Maintainer: Rosario M. Piro
E-Mail: <r.piro@fu-berlin.de> or <rmpiro@gmail.com>

References

Krueger, Piro (2017) Identification of Mutational Signatures Active in Individual Tumors. NETTAB 2017 - Methods, Tools & Platforms for Personalized Medicine in the Big Data Era, October 16-18, 2017, Palermo, Italy. PeerJ Preprints 5:e3257v1, 2017.
http://rmpiro.net/decompTumor2Sig/

See Also

decompTumor2Sig
loadAlexandrovSignatures
loadShiraishiSignatures

Examples

```
### get Alexandrov signatures from COSMIC
signAlexandrov <- loadAlexandrovSignatures()

### convert them to the Shiraishi model
signShiraishi <- convertAlexandrov2Shiraishi(signAlexandrov)
```

---

convertGenomesFromVRanges

convertGenomesFromVRanges

Description

Converts teh SNVs of a single tumor genome (sample) or a set of genomes from a VRanges object (package "VariantAnnotation") and determines the mutation frequencies according to a specific model of mutational signatures (Alexandrov or Shiraishi), such that the resulting format can be used as genomes input for decomposeTumorGenomes.

Usage

```
convertGenomesFromVRanges(vranges, numBases=5, type="Shiraishi", trDir=TRUE,
refGenome=BSgenome.Hsapiens.UCSC.hg19::BSgenome.Hsapiens.UCSC.hg19,
transcriptAnno=TxDb.Hsapiens.UCSC.hg19.knownGene::TxDb.Hsapiens.UCSC.hg19.knownGene,
verbose=TRUE)
```

Arguments

vranges	(Mandatory) The VRanges object which specifies the mutations.
numBases	(Mandatory) Total number of bases (mutated base and flanking bases) to be used for sequence patterns. Default: 5
type	(Mandatory) Signature model or type ('Alexandrov' or 'Shiraishi'). Default: Shiraishi
trDir	(Mandatory) Specifies whether the transcription direction is taken into account in the signature model. If so, only mutations within genes can be considered. Default: TRUE
refGenome	(Mandatory) The reference genome (BSgenome) needed to extract sequence patterns. Default: BSgenome object for hg19.
transcriptAnno	(Optional) Transcript annotation (TxDb object) used to determine the transcription direction. This is required only if trDir is TRUE. Default: TxDb object for hg19.
verbose	(Optional) Print information about reading and processing the mutation data. Default: TRUE

Value

A list containing the genomes in terms of frequencies of the mutated sequence patterns. This list of genomes can be used for decomposeTumorGenomes.

Author(s)

Rosario M. Piro
Freie Universitaet Berlin
Maintainer: Rosario M. Piro
E-Mail: <r.piro@fu-berlin.de> or <rmpiro@gmail.com>

References

Krueger, Piro (2017) Identification of Mutational Signatures Active in Individual Tumors. NETTAB 2017 - Methods, Tools & Platforms for Personalized Medicine in the Big Data Era, October 16-18, 2017, Palermo, Italy. PeerJ Preprints 5:e3257v1, 2017.
http://rmpiro.net/decompTumor2Sig/

See Also

decompTumor2Sig
decomposeTumorGenomes
loadGenomesFromVCF

Examples

```
### load the reference genome and the transcript annotation database
refGenome <- BSgenome.Hsapiens.UCSC.hg19::BSgenome.Hsapiens.UCSC.hg19
transcriptAnno <-
  TxDb.Hsapiens.UCSC.hg19.knownGene::TxDb.Hsapiens.UCSC.hg19.knownGene

### take the breast cancer genomes from Nik-Zainal et al (PMID: 22608084)
gfile <- system.file("extdata",
         "Nik-Zainal_PMID_22608084-VCF-convertedfromMPF.vcf.gz",
         package="decompTumor2Sig")

### get the corresponding VRanges object (using the VariantAnnotation
### package)
library(VariantAnnotation)
vr <- readVcfAsVRanges(gfile, genome="hg19")

### convert the VRanges object to the decompTumor2Sig format
genomes <- convertGenomesFromVRanges(vr, numBases=5, type="Shiraishi",
         trDir=TRUE, refGenome=refGenome, transcriptAnno=transcriptAnno,
         verbose=FALSE)
```

---

decomposeTumorGenomes *decomposeTumorGenomes*

Description

Takes a set of mutational signatures and mutation features from one or more tumor genomes and computes weights/contributions for each of the signatures in each individual genome. Alternatively, the function can determine for each genome only a subset of signatures and their contributions which are sufficient to exceed a user-given minimum threshold for the explained variance of the genome's mutation load.

Usage

```
decomposeTumorGenomes(genomes, signatures, minExplainedVariance=NULL,
                      minNumSignatures=2, maxNumSignatures=NULL,
                      greedySearch=FALSE,
                      constrainToMaxContribution=FALSE, tolerance=0.1,
                      verbose=FALSE)
```

Arguments

genomes	(Mandatory) Can be either a vector, a data frame or a matrix (for an individual tumor genome), or a list of one of these object types (for multiple tumors). Each tumor genome must be of the same form as the 'signatures' (see below).
signatures	(Mandatory) A list of vectors, data frames or matrices. Each of the objects represents one mutational signature. Vectors are used for Alexandrov signatures, data frames or matrices for Shiraishi signatures.
minExplainedVariance	(Optional) If NULL (default), exactly maxNumSignatures (see below; default: all) will be taken for decomposing each genome. If a numeric value between 0 and 1 is specified for minExplainedVariance, for each genome the function

will select the smallest number of signatures which is sufficient to explain at least the specified fraction of the variance of the genome's mutation load. E.g., if `minExplainedVariance`=0.99 the smallest subset of signatures that explains at least 99% of the variance is taken. Please note: depending on the number of signatures, this may take quite a while because for each number K of signatures, all possible subsets composed of K signatures will be tested to identify the subset that explains the highest part of the variance. If not enough variance is explained, K will be incremented by one. Notes: 1) to speed up the search, the parameters `minNumSignatures`, `maxNumSignatures` and `greedySearch` can be used (see below); 2) for genomes for which none of the possible subsets of signatures explains enough variance, the returned exposure vector will be set to `NULL`.

`minNumSignatures`
: (Optional) Used if `minExplainedVariance` is specified (see above). To find the smallest subset of signatures which explain the variance, at least `minNumSignatures` will be taken. This can be used to reduce the search space in a time-consuming search over a large number of signatures.

`maxNumSignatures`
: (Optional) Used if `minExplainedVariance` is specified (see above). To find the smallest subset of signatures which explain the variance, at most `maxNumSignatures` will be taken. This can be used to reduce the search space in a time-consuming search over a large number of signatures. If `maxNumSignatures` is `NULL` (default), all signatures will be taken as the maximum.

`greedySearch`
: (Optional) Used if `minExplainedVariance` is specified (see above). If `greedySearch` is `TRUE` then not all possible combinations of `minNumSignatures` to `maxNumSignatures` signatures will be checked. Instead, first all possible combinations for exactly `minNumSignatures` will be checked to select the best starting set, then iteratively the next best signature will be added (maximum increase in explained variability) until `minExplainedVariance` of the variance can be explained (or `maxNumSignatures` is exceeded). NOTE: this is highly recommended for large sets of signatures (>15)!

`constrainToMaxContribution`
: (Optional) [Note: this is experimental and is usually not needed!] If `TRUE`, the maximum contribution that can be attributed to a signature will be constraint by the variant feature counts (e.g., specific flanking bases) observed in the individual tumor genome. If, for example, 30% of all observed variants have a specific feature and 60% of the variants produced by a mutational process/signature will manifest the feature, then the signature can have contributed up to 0.3/0.6 (=0.5 or 50%) of the observed variants. The lowest possible contribution over all signature features will be taken as the allowed maximum contribution of the signature. This allowed maximum will additionally be increased by the value specified as `tolerance` (see below). For the illustrated example and `tolerance`=0.1 a contribution of up to 0.5+0.1 = 0.6 (or 60%) of the signature would be allowed.

`tolerance`
: (Optional) If `constrainToMaxContribution` is `TRUE`, the maximum contribution computed for a signature is increased by this value (see above). If the parameter `constrainToMaxContribution` is `FALSE`, the tolerance value is ignored. Default: 0.1.

`verbose`
: (Optional) If `TRUE` some information about the processed genome and used number of signatures will be printed.

Value

A list of signature weight vectors (also called 'exposures'), one for each tumor genome. E.g., the first vector element of the first list object is the weight/contribution of the first signature in the first tumor genome. IMPORTANT: If `minExplainedVariance` is specified, then the exposures of a genome will NOT be returned if the minimum explained variance is not reached within the requested minimum and maximum numbers of signatures (`minNumSignatures` and `maxNumSignatures`)! The corresponding exposure vector will be set to `NULL`.

Author(s)

Rosario M. Piro, Sandra Krueger
Freie Universitaet Berlin
Maintainer: Rosario M. Piro
E-Mail: <r.piro@fu-berlin.de> or <rmpiro@gmail.com>

References

Krueger, Piro (2017) Identification of Mutational Signatures Active in Individual Tumors. NETTAB 2017 - Methods, Tools & Platforms for Personalized Medicine in the Big Data Era, October 16-18, 2017, Palermo, Italy. PeerJ Preprints 5:e3257v1, 2017.
http://rmpiro.net/decompTumor2Sig/

See Also

decompTumor2Sig

Examples

```
### get Alexandrov signatures from COSMIC
signatures <- loadAlexandrovSignatures()

### load reference genome
refGenome <- BSgenome.Hsapiens.UCSC.hg19::BSgenome.Hsapiens.UCSC.hg19

### load breast cancer genomes from Nik-Zainal et al (PMID: 22608084)
gfile <- system.file("extdata",
         "Nik-Zainal_PMID_22608084-VCF-convertedfromMPF.vcf.gz",
         package="decompTumor2Sig")
genomes <- loadGenomesFromVCF(gfile, numBases=3, type="Alexandrov",
         trDir=FALSE, refGenome=refGenome, verbose=FALSE)

### compute exposures
exposures <- decomposeTumorGenomes(genomes, signatures, verbose=FALSE)

### (for further examples on searching subsets, please see the vignette)
```

`determineSignatureDistances`
determineSignatureDistances

Description

Determines all similarities between a given target signature (of type Alexandrov or Shiraishi) with a set of other signatures (of the same type). This can help to compare signatures that have been determined in different ways or from different datasets. Different distance measures can be used (see below).

Usage

```
determineSignatureDistances(target, signatures, method="euclidean")
```

Arguments

`target`	(Mandatory) A single signature of the Alexandrov (vector) or Shiraishi type (data frame or matrix).
`signatures`	(Mandatory) The list of signatures for which the distances to the `target` need to be computed. The signatures must be of the same type as `target`.
`method`	(Optional) The distance measure to be used. This can be one of the following: `"frobenius"` for Frobenius distance between matrices (only for Shiraishi signatures); `"rss"` for the residual sum of squares (squared error); or any distance measure available for the function `dist()` of the `stats` package. Default: `"euclidean"`.

Value

A signature-named vector containing all distances. This vector has the same order as the signature list, so it is not sorted according to distance.

Author(s)

Rosario M. Piro
Freie Universitaet Berlin
Maintainer: Rosario M. Piro
E-Mail: <r.piro@fu-berlin.de> or <rmpiro@gmail.com>

References

Krueger, Piro (2017) Identification of Mutational Signatures Active in Individual Tumors. NETTAB 2017 - Methods, Tools & Platforms for Personalized Medicine in the Big Data Era, October 16-18, 2017, Palermo, Italy. PeerJ Preprints 5:e3257v1, 2017.
http://rmpiro.net/decompTumor2Sig/

See Also

decompTumor2Sig
mapSignatureSets

Examples

```
### get Alexandrov signatures from COSMIC
signAlexandrov <- loadAlexandrovSignatures()

### convert them to Shiraishi signatures
```

```
signAlex2Shi <- convertAlexandrov2Shiraishi(signAlexandrov)

### define an arbitrary signature just for testing
### (similar to signature 1)
testSig <- matrix(c(0.1,   0, 0.7, 0.1, 0.1,   0,
                    0.3, 0.2, 0.3, 0.2,   0,   0,
                    0.2, 0.1, 0.5, 0.2,   0,   0), nrow=3, byrow=TRUE)

### compute distances of the test signature to the converted
### Alexandrov signatures from COSMIC
determineSignatureDistances(testSig, signAlex2Shi, method="frobenius")
```

`downgradeShiraishiSignatures` *downgradeShiraishiSignatures*

Description

Downgrades/trims Shiraishi signatures by dropping flanking bases (reducing the length of the sequence pattern and/or the transcription direction. This can be easily down because the flanking bases and the transcription direction are considered as independent features according to the Shiraishi model of mutational signatures.

Usage

```
downgradeShiraishiSignatures(signatures, numBases=NULL, removeTrDir=FALSE)
```

Arguments

`signatures`	(Mandatory) A list of Shiraishi signatures that need to be downgraded/trimmed.
`numBases`	(Conditionally optional) The total number of bases (mutated base plus flanking bases around the mutated base) that should be kept. All further flanking bases farther away from the mutated bases are dropped. If specified, `numBases` must be odd an smaller than the `signatures`' current number of bases. If `NULL`, no flanking bases will be dropped. At least one of `numBases` or `removeTrDir` must be specified.
`removeTrDir`	(Conditionally optional) Logical value that specifies whether information on the transcript direction should be dropped (if present at all). At least one of `numBases` or `removeTrDir` must be specified.

Value

A list of Shiraishi signatures that have been accordingly downgraded.

Author(s)

Rosario M. Piro, Sandra Krueger
Freie Universitaet Berlin
Maintainer: Rosario M. Piro
E-Mail: <r.piro@fu-berlin.de> or <rmpiro@gmail.com>

References

Krueger, Piro (2017) Identification of Mutational Signatures Active in Individual Tumors. NETTAB 2017 - Methods, Tools & Platforms for Personalized Medicine in the Big Data Era, October 16-18, 2017, Palermo, Italy. PeerJ Preprints 5:e3257v1, 2017.
http://rmpiro.net/decompTumor2Sig/

See Also

decompTumor2Sig

Examples

```
### Load 15 Shiraishi signatures obtained from 435 tumor genomes from
### Alexandrov et al. (number of bases: 5, transcription direction: yes)
sfile <- system.file("extdata",
         "Alexandrov_PMID_23945592_435_tumors-pmsignature-15sig.Rdata",
         package="decompTumor2Sig")
load(sfile)

### downgrade the signatures to include only 3 bases and drop the
### trancription direction
downgradeShiraishiSignatures(signatures, numBases=3, removeTrDir=TRUE)
```

`evaluateDecompositionQuality` *evaluateDecompositionQuality*

Description

Evaluates the quality of a decomposition with `decomposeTumorGenomes` by comparing the re-composed (=re-constructed) tumor genome mutation frequencies to those actually observed in the tumor genome. Tumor genome mutation frequencies are reconstructed using `composeGenomesFromExposures` and the results can optionally be plotted.

Usage

```
evaluateDecompositionQuality(exposure, signatures, genome, plot=FALSE)
```

Arguments

`exposure`	(Mandatory) A single vector containing the estimated signature contributions/exposures of a single tumor as provided by the function `decomposeTumorGenomes`. The number of elements of the exposure vector must correspond to the number of signatures (see below).
`signatures`	(Mandatory) The list of signatures (vectors, data frames or matrices) for which the exposures were obtained. Each of the list objects represents one mutational signature. Vectors are used for Alexandrov signatures, data frames or matrices for Shiraishi signatures.
`genome`	(Mandatory) A single tumor genome in form of mutation frequencies specified either in the Alexandrov or the Shiraishi format (must match the format used for `signatures`, see above).

`plot` (Optional) If `FALSE` (default), the numerical results (see below) will be returned. If `TRUE`, the reconstructed mutation frequencies will be plotted against the original, observed mutation frequencies and the numerical results will be integrated as text labels in the plot.

Value

A named list object containing measurements for the Pearson corrleation coefficient between the reconstructed and observerd mutation frequencies, and the explained variance; or alternatively, a plot with these measurements (see option `plot` above).

Author(s)

Rosario M. Piro
Freie Universitaet Berlin
Maintainer: Rosario M. Piro
E-Mail: <r.piro@fu-berlin.de> or <rmpiro@gmail.com>

References

Krueger, Piro (2017) Identification of Mutational Signatures Active in Individual Tumors. NETTAB 2017 - Methods, Tools & Platforms for Personalized Medicine in the Big Data Era, October 16-18, 2017, Palermo, Italy. PeerJ Preprints 5:e3257v1, 2017.
http://rmpiro.net/decompTumor2Sig/

See Also

decompTumor2Sig
decomposeTumorGenomes
composeGenomesFromExposures
computeExplainedVariance

Examples

```
### get Alexandrov signatures from COSMIC
signatures <- loadAlexandrovSignatures()

### load preprocessed breast cancer genomes (object 'genomes') from
### Nik-Zainal et al (PMID: 22608084)
gfile <- system.file("extdata",
         "Nik-Zainal_PMID_22608084-genomes-Alexandrov_3bases.Rdata",
         package="decompTumor2Sig")
load(gfile)

### compute exposures
exposures <- decomposeTumorGenomes(genomes, signatures, verbose=FALSE)

### evaluate the decomposition by comparing to the original data
evaluateDecompositionQuality(exposures[[1]], signatures, genomes[[1]])
```

getGenomesFromMutationFeatureData
getGenomesFromMutationFeatureData

Description

Takes a MutationFeatureData object (mutation count data) as read by the 'pmsignature' package (e.g., by pmsignature::readMPFile) and extracts the mutation counts. For passing them to decomposeTumorGenomes, the mutation counts must be normalized to mutation fractions, which is done by default. IMPORTANT: set normalize to FALSE only if you are interested in full integer counts, but do not pass unnormalized counts to decomposeTumorGenomes!

Usage

```
getGenomesFromMutationFeatureData(countData, normalize=TRUE)
```

Arguments

countData	(Mandatory) A MutationFeatureData object as constructed, for example, by pmsignature::readMPFile.
normalize	(Optional) Boolean value to specify whether to normalize the mutation count data to mutation fractions between 0 and 1. This is the default and NECESSARY in case you want to pass the return value to decomposeTumorGenomes. Set normalize to FALSE only if you are interested in full integer counts, but do not pass unnormalized counts to decomposeTumorGenomes!

Value

A list of (normalized) mutation counts, one object per genome. The format is the same table used by the corresponding Shiraishi signatures.

Author(s)

Rosario M. Piro, Sandra Krueger
Freie Universitaet Berlin
Maintainer: Rosario M. Piro
E-Mail: <r.piro@fu-berlin.de> or <rmpiro@gmail.com>

References

Krueger, Piro (2017) Identification of Mutational Signatures Active in Individual Tumors. NETTAB 2017 - Methods, Tools & Platforms for Personalized Medicine in the Big Data Era, October 16-18, 2017, Palermo, Italy. PeerJ Preprints 5:e3257v1, 2017.
http://rmpiro.net/decompTumor2Sig/

See Also

decompTumor2Sig

Examples

```
### get breast cancer genomes from
### Nik-Zainal et al (PMID: 22608084) in the format produced by
### pmsignature (PMID: 26630308)
pmsigdata <- system.file("extdata",
         "Nik-Zainal_PMID_22608084-pmsignature-G.Rdata",
         package="decompTumor2Sig")
load(pmsigdata)

### extract the genomes from the pmsignature G object
genomes <- getGenomesFromMutationFeatureData(G, normalize=TRUE)
```

`getSignatureListFromEstimatedParameters` *getSignatureListFromEstimatedParameters*

Description

Takes an `EstimatedParameters` object (signatures data) as computed by the 'pmsignature' package (by `pmsignature::getPMSignature`) and extracts the signature information. This can then be passed to `decomposeTumorGenomes`.

Usage

```
getSignatureListFromEstimatedParameters(Param)
```

Arguments

`Param`	(Mandatory) An `EstimatedParameters` object as the one produced by the `pmsignature` package's signature contruction method `pmsignature::getPMSignature`.

Value

A list of Shiraishi signatures, one object per signature. The format is a table with mutliple rows, the first for the base substitution, then 2*N rows for N flanking bases in each direction, and finally an (optional) row for the transcription strand, if it has been taken into account. Please see the `pmsignature` package or the `decompTumor2Sig` vignette for more information.

Author(s)

Rosario M. Piro, Sandra Krueger
Freie Universitaet Berlin
Maintainer: Rosario M. Piro
E-Mail: <r.piro@fu-berlin.de> or <rmpiro@gmail.com>

References

Krueger, Piro (2017) Identification of Mutational Signatures Active in Individual Tumors. NETTAB 2017 - Methods, Tools & Platforms for Personalized Medicine in the Big Data Era, October 16-18, 2017, Palermo, Italy. PeerJ Preprints 5:e3257v1, 2017.
http://rmpiro.net/decompTumor2Sig/

See Also

decompTumor2Sig
loadShiraishiSignatures

Examples

```
### load signatures for breast cancer genomes from
### Nik-Zainal et al (PMID: 22608084) in the format produced by
### pmsignature (PMID: 26630308)
pmsigdata <- system.file("extdata",
         "Nik-Zainal_PMID_22608084-pmsignature-Param.Rdata",
         package="decompTumor2Sig")
load(pmsigdata)

### extract the signatures from the pmsignature Param object

signatures <- getSignatureListFromEstimatedParameters(Param)
```

---

loadAlexandrovSignatures

loadAlexandrovSignatures

Description

Loads a set Alexandrov signatures from a flat file or URL. Signatures must be specified in the tab-separated format used by the COSMIC website; see http://cancer.sanger.ac.uk/cosmic/signatures -> Download signatures.

Example:

Subst.	Trinucleotide	Mutation Type	Signature 1	Signature 2	...
C>A	ACA	A[C>A]A	0.011098326166	0.000682708227	...
C>A	ACC	A[C>A]C	0.009149340734	0.000619107232	...
C>A	ACG	A[C>A]G	0.001490070468	0.000099278956	...
C>A	ACT	A[C>A]T	0.006233885236	0.000323891363	...
[...]					
T>G	TTG	T[T>G]G	0.002031076880	0.000206615168	...
T>G	TTT	T[T>G]T	0.004030128160	0.000023598204	...

Usage

```
loadAlexandrovSignatures(file)
```

Arguments

`file`	(Mandatory) Can be a single file name or an URL for download. Default (COSMIC): "http://cancer.sanger.ac.uk/cancergenome/assets/signatures_probabilities.txt"

Value

A list of Alexandrov signatures that can be used for `decomposeTumorGenomes`.

Author(s)

Rosario M. Piro, Sandra Krueger
Freie Universitaet Berlin
Maintainer: Rosario M. Piro
E-Mail: <r.piro@fu-berlin.de> or <rmpiro@gmail.com>

References

Krueger, Piro (2017) Identification of Mutational Signatures Active in Individual Tumors. NETTAB 2017 - Methods, Tools & Platforms for Personalized Medicine in the Big Data Era, October 16-18, 2017, Palermo, Italy. PeerJ Preprints 5:e3257v1, 2017.
http://rmpiro.net/decompTumor2Sig/

See Also

decompTumor2Sig
loadShiraishiSignatures

Examples

```
### get Alexandrov signatures from COSMIC
signatures <- loadAlexandrovSignatures()
```

`loadGenomesFromMPF` *loadGenomesFromMPF*

Description

Loads a single tumor genome (sample) or a set of genomes from an MPF file (Mutation Position Format) and determines the mutation frequencies according to a specific model of mutational signatures (Alexandrov or Shiraishi).

An MPF file has the following format (one line per mutation and patient/sample):

[sampleID]<tab>[chrom]<tab>[position]<tab>[ref_bases]<tab>[alt_bases]

Usage

```
loadGenomesFromMPF(file, numBases=5, type="Shiraishi", trDir=TRUE,
refGenome=BSgenome.Hsapiens.UCSC.hg19::BSgenome.Hsapiens.UCSC.hg19,
transcriptAnno=TxDb.Hsapiens.UCSC.hg19.knownGene::TxDb.Hsapiens.UCSC.hg19.knownGene,
verbose=TRUE)
```

Arguments

file	(Mandatory) The name of the MPF file (can be compressed with gzip).
numBases	(Mandatory) Total number of bases (mutated base and flanking bases) to be used for sequence patterns. Default: 5
type	(Mandatory) Signature model or type ('Alexandrov' or 'Shiraishi'). Default: Shiraishi
trDir	(Mandatory) Specifies whether the transcription direction is taken into account in the signature model. If so, only mutations within genes can be considered. Default: TRUE
refGenome	(Mandatory) The reference genome (BSgenome) needed to extract sequence patterns. Default: BSgenome object for hg19.
transcriptAnno	(Optional) Transcript annotation (TxDb object) used to determine the transcription direction. This is required only if trDir is TRUE. Default: TxDb object for hg19.
verbose	(Optional) Print information about reading and processing the mutation data. Default: TRUE

Value

A list containing the genomes in terms of frequencies of the mutated sequence patterns. This list of genomes can be used for decomposeTumorGenomes.

Author(s)

Rosario M. Piro
Freie Universitaet Berlin
Maintainer: Rosario M. Piro
E-Mail: <r.piro@fu-berlin.de> or <rmpiro@gmail.com>

References

Krueger, Piro (2017) Identification of Mutational Signatures Active in Individual Tumors. NETTAB 2017 - Methods, Tools & Platforms for Personalized Medicine in the Big Data Era, October 16-18, 2017, Palermo, Italy. PeerJ Preprints 5:e3257v1, 2017.
http://rmpiro.net/decompTumor2Sig/

See Also

decompTumor2Sig
decomposeTumorGenomes
loadGenomesFromVCF
getGenomesFromMutationFeatureData

Examples

```
### load reference genome and transcript annotation (if direction is needed)
refGenome <- BSgenome.Hsapiens.UCSC.hg19::BSgenome.Hsapiens.UCSC.hg19
transcriptAnno <-
  TxDb.Hsapiens.UCSC.hg19.knownGene::TxDb.Hsapiens.UCSC.hg19.knownGene

### load breast cancer genomes from Nik-Zainal et al (PMID: 22608084)
```

```
gfile <- system.file("extdata", "Nik-Zainal_PMID_22608084-MPF.txt.gz",
         package="decompTumor2Sig")
genomes <- loadGenomesFromMPF(gfile, numBases=5, type="Shiraishi",
         trDir=TRUE, refGenome=refGenome, transcriptAnno=transcriptAnno,
         verbose=FALSE)
```

`loadGenomesFromVCF` *loadGenomesFromVCF*

Description

Loads a single tumor genome (sample) or a set of genomes from a VCF file and determines the mutation frequencies according to a specific model of mutational signatures (Alexandrov or Shiraishi).

Usage

```
loadGenomesFromVCF(file, numBases=5, type="Shiraishi", trDir=TRUE,
refGenome=BSgenome.Hsapiens.UCSC.hg19::BSgenome.Hsapiens.UCSC.hg19,
transcriptAnno=TxDb.Hsapiens.UCSC.hg19.knownGene::TxDb.Hsapiens.UCSC.hg19.knownGene,
verbose=TRUE)
```

Arguments

`file`	(Mandatory) The name of the VCF file (can be compressed with `gzip`).
`numBases`	(Mandatory) Total number of bases (mutated base and flanking bases) to be used for sequence patterns. Default: 5
`type`	(Mandatory) Signature model or type ('`Alexandrov`' or '`Shiraishi`'). Default: `Shiraishi`
`trDir`	(Mandatory) Specifies whether the transcription direction is taken into account in the signature model. If so, only mutations within genes can be considered. Default: `TRUE`
`refGenome`	(Mandatory) The reference genome (`BSgenome`) needed to extract sequence patterns. Default: `BSgenome` object for hg19.
`transcriptAnno`	(Optional) Transcript annotation (`TxDb` object) used to determine the transcription direction. This is required only if `trDir` is `TRUE`. Default: `TxDb` object for hg19.
`verbose`	(Optional) Print information about reading and processing the mutation data. Default: `TRUE`

Value

A list containing the genomes in terms of frequencies of the mutated sequence patterns. This list of genomes can be used for `decomposeTumorGenomes`.

Author(s)

Rosario M. Piro
Freie Universitaet Berlin
Maintainer: Rosario M. Piro
E-Mail: <r.piro@fu-berlin.de> or <rmpiro@gmail.com>

References

Krueger, Piro (2017) Identification of Mutational Signatures Active in Individual Tumors. NETTAB 2017 - Methods, Tools & Platforms for Personalized Medicine in the Big Data Era, October 16-18, 2017, Palermo, Italy. PeerJ Preprints 5:e3257v1, 2017.
http://rmpiro.net/decompTumor2Sig/

See Also

decompTumor2Sig
decomposeTumorGenomes
loadGenomesFromVCF
getGenomesFromMutationFeatureData

Examples

```
### load reference genome and transcript annotation (if direction is needed)
refGenome <- BSgenome.Hsapiens.UCSC.hg19::BSgenome.Hsapiens.UCSC.hg19
transcriptAnno <-
  TxDb.Hsapiens.UCSC.hg19.knownGene::TxDb.Hsapiens.UCSC.hg19.knownGene

### load breast cancer genomes from Nik-Zainal et al (PMID: 22608084)
gfile <- system.file("extdata",
         "Nik-Zainal_PMID_22608084-VCF-convertedfromMPF.vcf.gz",
         package="decompTumor2Sig")
genomes <- loadGenomesFromVCF(gfile, numBases=5, type="Shiraishi",
         trDir=TRUE, refGenome=refGenome, transcriptAnno=transcriptAnno,
         verbose=FALSE)
```

`loadShiraishiSignatures` *loadShiraishiSignatures*

Description

Loads one or more Shiraishi signatures from flat files (one file per signature). The signatures must be specified as matrices without headers and row names.

Format (see Shiraishi et al. PLoS Genetics 11(12):e1005657, 2015):

First line: Frequencies of the base changes C>A, C>G, C>T, T>A, T>C, and T>G

Following 2k lines (for k up- and downstream flanking bases): Frequencies of the bases A, C, G, and T, followed by two 0 values

Final line (only if transcription direction is considered): Frequencies of occurences on the transcription strand, and on the opposite strand, followed by four 0 values.

Example:

1.8874e-14	0.10974	0.045918	0.11308	0.07429	0.65697
3.8079e-01	0.12215	0.191456	0.30561	0.00000	0.00000
1.5311e-01	0.34214	0.179774	0.32497	0.00000	0.00000
1.2378e-01	0.10243	0.163461	0.61032	0.00000	0.00000
3.4891e-01	0.15346	0.156687	0.34094	0.00000	0.00000
5.6435e-01	0.43565	0.000000	0.00000	0.00000	0.00000

Usage

```
loadShiraishiSignatures(files)
```

Arguments

`files`	(Mandatory) Can be a single file name, a vector of file names, or a list of file names.

Value

A list of Shiraishi signatures that can be used for `decomposeTumorGenomes`.

Author(s)

Rosario M. Piro, Sandra Krueger
Freie Universitaet Berlin
Maintainer: Rosario M. Piro
E-Mail: <r.piro@fu-berlin.de> or <rmpiro@gmail.com>

References

Krueger, Piro (2017) Identification of Mutational Signatures Active in Individual Tumors. NETTAB 2017 - Methods, Tools & Platforms for Personalized Medicine in the Big Data Era, October 16-18, 2017, Palermo, Italy. PeerJ Preprints 5:e3257v1, 2017.
http://rmpiro.net/decompTumor2Sig/

See Also

decompTumor2Sig
loadAlexandrovSignatures
getSignatureListFromEstimatedParameters

Examples

```
### load four Shiraishi signatures for breast cancer genomes from
### Nik-Zainal et al (PMID: 22608084) from flat files
sigfiles <- system.file("extdata",
        paste0("Nik-Zainal_PMID_22608084-pmsignature-sig",1:4,".tsv"),
        package="decompTumor2Sig")

signatures <- loadShiraishiSignatures(sigfiles)
```

mapSignatureSets *mapSignatureSets*

Description

Find a mapping from one set of signatures to another. Both Alexandrov and Shiraishi signatures can be handled, but both sets must be of the same type. The mapping can either be a unique (one-to-one) mapping or indentify best matching while allowing multiple signatures to be mapped to the same target signature if it is the best match for more than one signature.

Usage

```
mapSignatureSets(fromSignatures, toSignatures, method="euclidean",
                 unique=FALSE)
```

Arguments

fromSignatures	(Mandatory) A set (list) of signatures of the Alexandrov (vector) or Shiraishi type (data frame or matrix), that has to be mapped to the signatures of a second set (toSignatures).
toSignatures	(Mandatory) The set (list) of signatures to which the set of fromSignatures has to be mapped.
method	(Optional) The distance measure to be used. This can be one of the following: "frobenius" for Frobenius distance between matrices (only for Shiraishi signatures); "rss" for the residual sum of squares (squared error); or any distance measure available for the function dist() of the stats package. Default: "euclidean".
unique	(Optional) If set to FALSE (default), then for each signature of fromSignatures the best match (minimum distance) from toSignatures is selected. The selected signatures need not be unique, i.e., one signature of toSignatures may be the best match for multiple signatures of fromSignatures. If set to TRUE, i.e., if a unique (one-to-one) mapping is required, an interative approach is performed: in each step, the best matching pair from fromSignatures and toSignatures is mapped and then removed from the list of signatures that remain to be mapped, such that they cannot be selected again.

Value

A vector having as elements the mapped signatures of fromSignatures, and as names the signatures of toSignatures with which they have been associated.

Author(s)

Rosario M. Piro
Freie Universitaet Berlin
Maintainer: Rosario M. Piro
E-Mail: <r.piro@fu-berlin.de> or <rmpiro@gmail.com>

References

Krueger, Piro (2017) Identification of Mutational Signatures Active in Individual Tumors. NETTAB 2017 - Methods, Tools & Platforms for Personalized Medicine in the Big Data Era, October 16-18, 2017, Palermo, Italy. PeerJ Preprints 5:e3257v1, 2017.
http://rmpiro.net/decompTumor2Sig/

See Also

decompTumor2Sig
determineSignatureDistances

Examples

```
### get Alexandrov signatures from COSMIC
signAlexandrov <- loadAlexandrovSignatures()

### convert them to Shiraishi signatures
signAlex2Shi <- convertAlexandrov2Shiraishi(signAlexandrov)

### define a small set of arbitrary signatures just for testing
### (similar to signatures 1, 5 and 13, respectively)
test1 <- matrix(c(0.1,   0, 0.7, 0.1, 0.1,   0,
                  0.3, 0.2, 0.3, 0.2,   0,   0,
                  0.2, 0.1, 0.5, 0.2,   0,   0), nrow=3, byrow=TRUE)
test2 <- matrix(c(0.1, 0.1, 0.3, 0.1, 0.3, 0.1,
                  0.3, 0.25, 0.2, 0.25, 0,   0,
                  0.3, 0.2, 0.2, 0.3,   0,   0), nrow=3, byrow=TRUE)
test3 <- matrix(c(0.1, 0.7, 0.2,   0,   0,   0,
                    0,   0,   0, 1.0,   0,   0,
                  0.5, 0.1,   0, 0.4,   0,   0), nrow=3, byrow=TRUE)
fromSig <- list(sig1=test1, sig2=test2, sig3=test3)

### compute distances of the test signature to the converted
### Alexandrov signatures from COSMIC
mapSignatureSets(fromSig, signAlex2Shi, method="frobenius", unique=TRUE)
```

`plotDecomposedContribution` *plotDecomposedContribution*

Description

Plots a the decomposition/exposures of a genome to the mutational signatures (mutational processes), that is, the contributions of the signatures to the mutations observed in a tumor genome.

These decompositions can be obtained running `decomposeTumorGenomes()`.

Usage

```
plotDecomposedContribution(decomposition, signatures=NULL, removeNA=TRUE)
```

Arguments

decomposition	(Mandatory) A decomposition vector (exposure vector) obtained for a single tumor genome.
signatures	(Optional) A list object containing the signatures used to compute the decomposition. If specified, the signature labels used in the plot will be taken from the element names of the list; otherwise signatures will be named from sign_1 to sign_N.
removeNA	(Optional) If TRUE (default), signatures with an NA as exposure will not be included on the x-axis of the the plot. Exposures can NA if they have been determined with a greedy search.

Value

No return value. The function creates a plot of the decomposed tumor genome (i.e., contributions of the single signatures).

Author(s)

Rosario M. Piro
Freie Universitaet Berlin
Maintainer: Rosario M. Piro
E-Mail: <r.piro@fu-berlin.de> or <rmpiro@gmail.com>

References

Krueger, Piro (2017) Identification of Mutational Signatures Active in Individual Tumors. NETTAB 2017 - Methods, Tools & Platforms for Personalized Medicine in the Big Data Era, October 16-18, 2017, Palermo, Italy. PeerJ Preprints 5:e3257v1, 2017.
http://rmpiro.net/decompTumor2Sig/

See Also

decompTumor2Sig
decomposeTumorGenomes

Examples

```
### get Alexandrov signatures from COSMIC
signatures <- loadAlexandrovSignatures()

### load preprocessed breast cancer genomes (object 'genomes') from
### Nik-Zainal et al (PMID: 22608084)
gfile <- system.file("extdata",
         "Nik-Zainal_PMID_22608084-genomes-Alexandrov_3bases.Rdata",
         package="decompTumor2Sig")
load(gfile)

### compute exposures
exposures <- decomposeTumorGenomes(genomes, signatures, verbose=FALSE)

### plot signature composition of the first genome
plotDecomposedContribution(exposures[[1]], signatures=NULL)
```

plotExplainedVariance *plotExplainedVariance*

Description

For a single genome and a given set of signatures, the function plots the variance of the genome's mutation load that can be explained with an increasing number of signatures (increasing subset of signatures). For each number K of signatures, the highest variance explained by any possible subset of K signatures will be plotted. This can help to evaluate what minimum threshlod for the explained variance should be used to decompose tumor genomes with the function `decomposeTumorGenomes`.

Usage

```
plotExplainedVariance(genome, signatures, minExplainedVariance=NULL,
                      minNumSignatures=2, maxNumSignatures=NULL)
```

Arguments

`genome`	(Mandatory) The mutation load of a single genome in Alexandrov- of Shiraishi-format, i.e. as vector or matrix. The format must be the same as the one used for the `signatures` (see below).
`signatures`	(Mandatory) The list of signatures (vectors, data frames or matrices) which are to be evaluated. Each of the list objects represents one mutational signature. Vectors are used for Alexandrov signatures, data frames or matrices for Shiraishi signatures.
`minExplainedVariance`	(Optional) If a numeric value between 0 and 1 is specified, the plot highlights the smallest subset of signatures that is sufficient to explain at least the specified fraction of the variance of the genome's mutation load. If, for example, `minExplainedVariance` is 0.99 the smallest subset of signatures that explains at least 99% of the variance will be highlighted.
`minNumSignatures`	(Optional) The plot will be generated only for K>=`minNumSignatures`.
`maxNumSignatures`	(Optional) The plot will be generated only for K<=`minNumSignatures`.

Value

No return value. The function creates a plot of the explained variance as a function of the number of signatures.

Author(s)

Rosario M. Piro
Freie Universitaet Berlin
Maintainer: Rosario M. Piro
E-Mail: <r.piro@fu-berlin.de> or <rmpiro@gmail.com>

References

Krueger, Piro (2017) Identification of Mutational Signatures Active in Individual Tumors. NETTAB 2017 - Methods, Tools & Platforms for Personalized Medicine in the Big Data Era, October 16-18, 2017, Palermo, Italy. PeerJ Preprints 5:e3257v1, 2017.
http://rmpiro.net/decompTumor2Sig/

See Also

decompTumor2Sig
decomposeTumorGenomes
computeExplainedVariance

Examples

```
### get 15 pre-processed Shiraishi signatures computed (object 'signatures')
### from 435 tumor genomes Alexandrov et al (PMID: 23945592)
### using the pmsignature package
sfile <- system.file("extdata",
         "Alexandrov_PMID_23945592_435_tumors-pmsignature-15sig.Rdata",
         package="decompTumor2Sig")
load(sfile)

### load preprocessed breast cancer genomes (object 'genomes') from
### Nik-Zainal et al (PMID: 22608084)
gfile <- system.file("extdata",
         "Nik-Zainal_PMID_22608084-genomes-Shiraishi_5bases_trDir.Rdata",
         package="decompTumor2Sig")
load(gfile)

### plot the explained variance for 2 to 6 signatures of the first genome
plotExplainedVariance(genomes[[1]], signatures,
         minExplainedVariance=0.98, minNumSignatures=2, maxNumSignatures=6)
```

---

plotMutationDistribution

plotMutationDistribution

Description

Plots a single signature or mutation frequency data for a single genome. This works for signatures or genome data of both the Shiraishi and the Alexandrov type.

[IMPORTANT: The function requires the 'pmsignature' package to be installed (Shiraishi et al. PLoS Genet 11(12):e1005657, 2015)!]

Usage

```
plotMutationDistribution(mutData)
```

Arguments

mutData	(Mandatory) The signature or genome mutation frequency data to be plotted. This can either be a matrix (Shiraishi-type model) or a numeric vector (Alexandrov-type model).

Value

No return value. The creates a plot on the standard graphical output.

Author(s)

Rosario M. Piro
Freie Universitaet Berlin
Maintainer: Rosario M. Piro
E-Mail: <r.piro@fu-berlin.de> or <rmpiro@gmail.com>

References

Krueger, Piro (2017) Identification of Mutational Signatures Active in Individual Tumors. NETTAB 2017 - Methods, Tools & Platforms for Personalized Medicine in the Big Data Era, October 16-18, 2017, Palermo, Italy. PeerJ Preprints 5:e3257v1, 2017.
http://rmpiro.net/decompTumor2Sig/

See Also

decompTumor2Sig

Examples

```
### Attention: using plotMutationDistribution requires the package
### pmsignature to be installed!


### get Alexandrov signatures from COSMIC
signatures <- loadAlexandrovSignatures()

### plot the first Alexandrov signature

plotMutationDistribution(signatures[[1]])


### load four Shiraishi signatures for breast cancer genomes from
### Nik-Zainal et al (PMID: 22608084) from flat files
sigfiles <- system.file("extdata",
         paste0("Nik-Zainal_PMID_22608084-pmsignature-sig",1:4,".tsv"),
         package="decompTumor2Sig")
signatures <- loadShiraishiSignatures(sigfiles)

### plot the first Shiraishi signature

plotMutationDistribution(signatures[[1]])
```

Index