Package 'decompTumor2Sig'

November 11, 2026

Type Package

Title Decomposition of individual tumors into mutational signatures by signature refitting

Depends R(>= 4.0), ggplot2

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

Suggests knitr, rmarkdown, BiocStyle

Version 2.28.1

Date 2026-07-21

Encoding UTF-8

Description Uses quadratic programming for signature refitting, i.e., 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/, https://github.com/rmpiro/decompTumor2Sig

BugReports https://github.com/rmpiro/decompTumor2Sig/issues

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

VignetteBuilder knitr

RoxygenNote 7.1.2

git_url https://git.bioconductor.org/packages/decompTumor2Sig

git_branch RELEASE_3_23

git_last_commit a438ec1

git_last_commit_date 2026-07-21

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Rosario M. Piro [aut, cre],
Sandra Krueger [ctb]

Maintainer Rosario M. Piro <rmpiro@gmail.com>

Contents

decompTumor2Sig-package . . . 3
addBestSignatureToSubset . . . 6
adjustAlexandrovSignature . . . 7
adjustShiraishiSignature . . . 8
adjustSignaturesForRegionSet . . . 8
buildGenomesFromMutationData . . . 10
buildSortedAlexandrovSignaturePatternList . . . 12
compNucFreq . . . 13
composeGenomesFromExposures . . . 14
computeExplainedVariance . . . 15
computeFrobeniusNorm . . . 17
computeRSS . . . 17
convAlx2Shi . . . 18
convertAlexandrov2Shiraishi . . . 19
convertGenomesFromVRanges . . . 20
convertSeqFreqToBaseFreq . . . 22
decomposeTumorGenomes . . . 23
determineSignatureDistances . . . 26
determineTypeNumBasesAndTrDir . . . 27
downgradeShiraishiSignatures . . . 28
evaluateDecompositionQuality . . . 30
getBestDecomp4Ksignatures . . . 31
getCountData . . . 33
getFeatVectList . . . 33
getGenomesFromMutFeatData . . . 34
getNumFlankingBases . . . 35
getNumSignatures . . . 36
getSampleList . . . 37
getSigFeatDist . . . 38
getSigFromEstParam . . . 38
getSignaturesFromEstParam . . . 39
getSigType . . . 40
haveTrDir . . . 41
is.probability.data.frame . . . 42
is.probability.data.frame.list . . . 43
is.probability.matrix . . . 43

is.probability.matrix.list . . . 44
is.probability.object . . . 45
is.probability.vector . . . 46
is.probability.vector.list . . . 46
isAlexandrovSet . . . 47
isBackGround . . . 48
isEstParamObject . . . 49
isExposureSet . . . 50
isMutFeatDataObject . . . 51
isShiraishiSet . . . 52
isSignatureSet . . . 53
mapSignatureSets . . . 54
plotAlexandrovModel . . . 56
plotDecomposedContribution . . . 57
plotExplainedVariance . . . 58
plotMutationDistribution . . . 60
plotShiraishiModel . . . 61
processMultipleSigSets . . . 62
QPforSig . . . 64
readAlexandrovSignatures . . . 65
readGenomesFromMPF . . . 67
readGenomesFromVCF . . . 69
readShiraishiSignatures . . . 71
roundIntegerSum . . . 72
sameSignatureFormat . . . 73
setNames4ShiraishiTable . . . 74
setNumFlankingBases . . . 75
setSigFeatDist . . . 76
setSigType . . . 76
setTrDir . . . 77

Index **79**

`decompTumor2Sig-package`

decompTumor2Sig - Decomposition of individual tumors into mutational signatures by signature refitting

---

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.

Details

Package:	decompTumor2Sig
Type:	Package
Version:	2.13.1
Date:	2022-05-09
License:	GPL (>=2)

The package provides the following functions:

adjustSignaturesForRegionSet():	adjust (normalize) mutational signatures for use with mutation data from a specific, limited subset of genomic regions (e.g., for targetted sequencing).
composeGenomesFromExposures():	(re-)construct tumor genome mutation frequencies from the signatures and their corresponding exposures, or contributions.
computeExplainedVariance():	determine the variance explained by estimated signature contributions (i.e., exposures to signatures).
convertAlexandrov2Shiraishi():	convert a set of Alexandrov signatures to Shiraishi signatures.
convertGenomesFromVRanges():	convert a genome or set of genomes from a `VariantAnnotation::VRanges` object.
decomposeTumorGenomes():	determine the weights/contributions of a set of signatures to each of a set of individual tumor genomes.
determineSignatureDistances():	for a given signature compute its distances to each of a set of target signatures.
downgradeShiraishiSignatures():	downgrade Shiraishi signatures by removing flanking bases and/or the transcription direction.
evaluateDecompositionQuality():	evaluate the quality of a decomposition by comparing the re-composed (=re-constructed) tumor mutation frequencies to those actually observed in the tumor genome.
getGenomesFromMutFeatData():	extract the genomes from a `MutationFeatureData` object as provided by, for example, `pmsignature::readMPFile`.
getSignaturesFromEstParam():	extract a set of signatures from an `EstimatedParameters` object as returned by function `getPMSignature`

	of the `pmsignature` package.
isAlexandrovSet():	checks whether the input list is compatible with the Alexandrov format (probability vectors).
isExposureSet():	checks whether the input list is compatible with exposure output obtained from `decomposeTumorGenomes`.
isShiraishiSet():	checks whether the input list is compatible with the Shiraishi format (matrices or data.frames of probabilities).
isSignatureSet():	checks whether the input list is compatible with either the Alexandrov or Shiraishi format.
mapSignatureSets():	find a mapping from one signature set to another.
plotDecomposedContribution():	plot the decomposition of a genome into mutational signatures (i.e., the contributions of, or exposures to, the signatures).
plotExplainedVariance():	plot the variance of a genome's mutation patterns which can be explained with an increasing number of signatures.
plotMutationDistribution():	plot a single signature or the mutation frequency data for a single genome.
readAlexandrovSignatures():	read Alexandrov signatures in the COSMIC format from a flat file or URL.
readGenomesFromMPF():	read a genome or set of genomes from a Mutation Position Format (MPF) file.
readGenomesFromVCF():	read a genome or set of genomes from a Variant Call Format (VCF) file.
readShiraishiSignatures():	read Shiraishi signatures from flat files.
sameSignatureFormat():	checks whether two input lists are sets of signatures of the same format.

Author(s)

Rosario M. Piro, Politecnico di Milano [aut, cre]
Sandra Krueger, Freie Universitaet Berlin [ctb]
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/

Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

`addBestSignatureToSubset` *addBestSignatureToSubset (internal function)*

Description

Add the best signature to an existing subset (highest increase in expl. var.) to improve the approximate decomposition of a genome.

Usage

```
addBestSignatureToSubset(genome, signatures, subset,
constrainToMaxContribution=FALSE, tolerance=0.1)
```

Arguments

`genome`	Genome for which to improve the decomposition.
`signatures`	The whole set of signatures (from which to choose additional signatures.
`subset`	The current subset that is used for decomposition.
`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.

Value

A list object containing: k=number of signatures; explVar=variance explained by these signatures; sigList=list of the signatures; decomposition=decomposition (exposures) obtained with these signatures.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152

---

adjustAlexandrovSignature

Adjust an Alexandrov signature (internal function)

Description

Adjust an Alexandrov signature by a set of pattern-specific factors.

Usage

```
adjustAlexandrovSignature(signature, factors)
```

Arguments

signature	Single Alexandrov signature to be adjusted
factors	Factors for trinucleotides

Value

Adjusted Alexandrov signature

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signature active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

`adjustShiraishiSignature` *Adjust an Shiraishi signature (internal function)*

Description

Adjust an Shiraishi signature by a set of base-specific factors.

Usage

```
adjustShiraishiSignature(signature, factorsMatrix)
```

Arguments

`signature`	Single Shiraishi signature to be adjusted
`factorsMatrix`	Factors for single nucleotides (Shiraishi-like matrix)

Value

Adjusted Shiraishi signature

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signature active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

`adjustSignaturesForRegionSet` *Adjust (normalize) signatures for a set of genomic regions.*

Description

'adjustSignaturesForRegionSet()' takes a set of signatures that have been orginally defined with respect to the nucleotide frequencies within a specific reference genome or region (e.g., by deriving them from whole genome mutation data) and adjusts or normalizes them to the often different nucleotide frequencies of another specific subset of genomic regions.

Usage

```
adjustSignaturesForRegionSet(signatures,
regionsTarget, regionsOriginal=NULL,
refGenome=BSgenome.Hsapiens.UCSC.hg19::BSgenome.Hsapiens.UCSC.hg19)
```

Arguments

`signatures`	(Mandatory) Signatures to be adjusted to the nucleotide frequencies of the genomic regions defined by the parameter `regions`.
`regionsTarget`	(Mandatory) `GRanges` object defining a subset of the genome (i.e., a set of genomic regions) for which the signatures need to be adjusted (can be set to `NULL` for the whole genome).
`regionsOriginal`	(Optional) `GRanges` object defining the subset of the genome (i.e., set of genomic regions) from which the signatures where originally derived. Default: `NULL` (whole genome).
`refGenome`	(Optional) Reference genome sequence from which to compute the nucleotide frequencies. Default: `BSgenome.Hsapiens.UCSC.hg19::BSgenome.Hsapiens.UCSC.hg19`.

Details

This may be useful, for example, to perform signature refitting (using `decomposeTumorGenomes`) for mutation data from targetted sequencing (e.g., only a subset of genes), whole exome sequencing (only exonic regions), or other limited subsets of the genome with particular nucleotide frequencies.

For Alexandrov-type signatures, the important frequencies are those of the whole sequence patterns (e.g., trinucleotides) whose central base can be mutated. Therefore, adjustment factors for individual mutation types (e.g., A[C>T]G) are computed by comparing the corresponding sequence pattern frequencies (e.g., ACG) between the original reference regions (e.g., whole genome) and the target regions (e.g., target regions of whole exome sequencing).

In the Shiraishi-type signature model, individual bases of the sequence patterns are considered as independent features. Thus, to compute nucleotide frequencies for such signatures, the frequencies of the sequence patterns (computed as for Alexandrov-type signatures) are broken down to single-nucleotide frequencies for the individual positions of the patterns.

In both cases, after the appropriate adjustment of individual features, signatures are re-normalized such that overall probabilites sum up to 1.

Value

A set of adjusted mutational signatures in the same format as those specified for `signatures`.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

See Also

decompTumor2Sig
decomposeTumorGenomes

Examples

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

### get gene annotation for the default reference genome (hg19)
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene::TxDb.Hsapiens.UCSC.hg19.knownGene

### get a GRanges object for gene promoters (-2000 to +200 bases from TSS)
### [taking only the first 1000 for testing purpose]
library(GenomicRanges)
regionsTarget <- promoters(txdb, upstream=2000, downstream=200)[seq(1000)]

### assume these signatures were derived only from mutation data from
### exons on chromosome X [not true; just for illustrative purpose]
filter <- list(tx_chrom = c("chrX"))
regionsOriginal <- exons(txdb, filter=filter)

### adjust signatures according to nucleotide frequencies in the target
### subset of the genome
sign_adj <- adjustSignaturesForRegionSet(signatures, regionsTarget,
                                                     regionsOriginal)
```

---

buildGenomesFromMutationData

buildGenomesFromMutationData (internal function)

Description

Build genome data structures (same as signatures) and fill then with mutation data.

Usage

```
buildGenomesFromMutationData(snvs, numBases, type, trDir,
uniqueTrDir=TRUE, refGenome, transcriptAnno, verbose)
```

Arguments

snvs	SNV matrix (see description above).
numBases	Number of bases for the sequence pattern (odd integer).
type	Type of signature to be used ("Alexandrov", "Shiraishi").
trDir	Logical: use transcription-strand information?
uniqueTrDir	Logical; used only if trDir is also TRUE: if uniqueTrDir is TRUE (default), then only mutations with only one defined transcription strand will be used, mutations for which both strands are valid are ignored. If FALSE, these mutations are accepted and one of the two transcription strands will be arbitrarily taken (the first one encountered in the databse specified for transcriptAnno). The latter was the behavior until version 1.3.5 of decompTumor2Sig and is also the behavior of pmsignature.
refGenome	Reference genome (BSgenome object).
transcriptAnno	Transcription information (TxDb object).
verbose	Logical. Print additional information?

Details

SNVs are specified as a matrix of the following format (adapted from VCF):
#> snvs[1:2,]
CHROM POS REF ALT FORMAT sample1 sample2
#[1,] "2" "947" "C" "T" "GT:PL:GQ:AD:DP" "1/1:84,6,0:6:0,2:2" NA
#[2,] "2" "992" "G" "A" "GT:PL:GQ:AD:DP" "0/1:123,0,33:33:1,3:4" "0/0:..."

Value

A list of genomes: each genome is represented by the observed frequencies of mutation patterns according to the selected signature type.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

`buildSortedAlexandrovSignaturePatternList` *buildSortedAlexandrovSignaturePatternList (internal function)*

Description

Build and sort the pattern list of an Alexandrov signature in the following way: First according to the base change: C>A, C>G, C>T, T>A, T>C, T>G, then within these categories according to the flanking bases: A, C, G, T.

Usage

```
buildSortedAlexandrovSignaturePatternList(numBases, trDir)
```

Arguments

`numBases`	Number of bases for the sequence pattern (odd integer).
`trDir`	Logical: use transcription-strand information?

Value

A sorted list of mutation features (e.g., triplets with base change).

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

compNucFreq *Compute nucleotide frequencies (internal function)*

Description

Compute nucleotide frequencies for a reference genome and a region set.

Usage

```
compNucFreq(refGenome, regions=NULL, numBases=1, mergeByRevComp=TRUE)
```

Arguments

refGenome	Reference genome (BSgenome)
regions	(Optional) Regions (GRanges); default: NULL (whole genome).
numBases	(Optional) Sequence pattern length for which to compute frequencies (e.g., '3' for trinucleotides). Default: 1 (single nucleotides).
mergeByRevComp	(Optional) Reduce redundancy by merging counts for reverse complement sequence patterns. This function allows this only for sequence patterns with odd length (e.g., single nucleotides, trinucleotides). Default: TRUE.

Value

Vector of computed nucleotide frequencies

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

composeGenomesFromExposures
Compose tumor genomes from exposures.

Description

'composeGenomesFromExposures()' re-composes (or predicts) tumor genomes (i.e., their mutation frequencies) from the given mutational 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 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 computed 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`.
`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., the frequencies of their mutational patterns computed as weighted sums of the mutational signatures, where the weights correspond to the contributions of, i.e., exposures to, the corresponding signatures.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

See Also

decompTumor2Sig
decomposeTumorGenomes

Examples

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

### 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

Compute the explained variance.

Description

'computeExplainedVariance()' computes for one or more tumors the variance which is explained by the estimated contributions (exposures) of a set of signatures when compared to the observed genomes.

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. The number of elements of each exposure vector must correspond to the number of signatures.
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.

Value

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

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

See Also

decompTumor2Sig
decomposeTumorGenomes
plotExplainedVariance

Examples

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

### 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)
```

computeFrobeniusNorm *computeFrobeniusNorm (internal function)*

Description

Compute the Frobenius norm of a numeric matrix (if this matrix is the difference between two matrices, then this corresponds to the Frobenius distance between these two matrices).

Usage

```
computeFrobeniusNorm(A)
```

Arguments

A	The numeric matrix.

Value

The Frobenius norm.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

computeRSS *computeRSS (internal function)*

Description

Compute the residual sum of squares (RSS), i.e., the sum of squared errors.

Usage

```
computeRSS(x, y)
```

Arguments

x	The first numeric object (e.g., matrix).
y	The second numeric object (of the same type as x).

Value

The RSS.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

---

convAlx2Shi *convAlx2Shi (internal function)*

Description

Convert a single Alexandrov signature to a Shiraishi signature.

Usage

```
convAlx2Shi(x)
```

Arguments

x	The Alexandrov signature (mutation pattern vector) to be converted.

Value

A Shiraishi signature (mutation pattern matrix).

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

`convertAlexandrov2Shiraishi` *Convert Alexandrov-type signatures to Shiraishi signatures*

Description

'convertAlexandrov2Shiraishi()' converts a set Alexandrov signatures to the Shiraishi model, summing the respective frequencies of base changes, and upstream and downstream flanking bases. In most cases, the resulting Shiraishi signatures don't provide information on the transcription strand, as this is not part of the standard Alexandrov signatures. While the conversion is mainly thought for signatures, it actually works also for mutation frequency data from genomes which have the same format. [Attention: this conversion entails a loss of specificity and the applicability of Shiraishi signatures derived from Alexandrov signatures has not been extensively explored!]

Usage

```
convertAlexandrov2Shiraishi(signatures)
```

Arguments

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

Value

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

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

See Also

decompTumor2Sig
readAlexandrovSignatures
readShiraishiSignatures

Examples

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

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

---

convertGenomesFromVRanges

Convert genomes from a VRanges *object*

Description

'convertGenomesFromVRanges()' converts the 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, enforceUniqueTrDir=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. Must be odd. 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 genomic regions with a defined transcription direction can be considered. Default: TRUE

enforceUniqueTrDir
: (Optional) Used only if trDir is TRUE. If enforceUniqueTrDir is TRUE (default), then mutations which map to a region with multiple overlapping genes with opposing transcription directions will be excluded from the analysis. If FALSE, the transcript direction encountered first in the transcript database (see transcriptAnno) is assigned to the mutation. The latter was the behavior until version 1.3.5 of decompTumor2Sig and is also the behavior of pmsignature. However, it is preferable to exclude these mutations from the count (default) because from mutation data alone it cannot be inferred which of the two genes has the higher transcriptional activity which might potentially be linked to the occurrence of the mutation. (If you are unsure, use the default setting; this option exists mostly for backward compatibility with older versions.)

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
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

See Also

decompTumor2Sig
decomposeTumorGenomes
readGenomesFromVCF
readGenomesFromMPF
getGenomesFromMutFeatData

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, enforceUniqueTrDir=TRUE, refGenome=refGenome,
         transcriptAnno=transcriptAnno, verbose=FALSE)
```

---

convertSeqFreqToBaseFreq

Convert sequence pattern frequencies to base frequencies (internal function)

Description

Convert sequence pattern frequencies (Alexandrov-like) to frequencies for individual bases (Shiraishi-like).

Usage

```
convertSeqFreqToBaseFreq(frequencies, sigType)
```

Arguments

frequencies	Vector for frequencies of individual sequence patterns
sigType	Signature type information as returned by determineTypeNumBasesAndTrDir

Value

Shiraishi signature-like matrix of per-base nucleotide frequencies

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

`decomposeTumorGenomes` *Decompose tumor genomes into mutational signatures*

Description

'decomposeTumorGenomes()' is the core function of this package. It decomposes tumor genomes into a given set of mutational signatures by computing their contributions (exposures) to the mutational load via quadratic programming. The function takes a set of mutational signatures and the mutation features of one or more tumor genomes and computes weights, i.e., contributions for each of the signatures in each individual genome. Alternatively, the function can determine for each genome only a subset of signatures whose contributions are sufficient to exceed a user-given minimum threshold for the explained variance of the genome's mutation patterns.

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`.
`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 patterns.

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 by default 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; 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 this number of signatures 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) If minExplainedVariance is specified 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 minExplainedVariance is NULL, then exactly maxNumSignatures signatures will be used. The default for maxNumSignatures is NULL (all signatures).

greedySearch (Optional) Used only in case minExplainedVariance has been specified. 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 approximate search 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 to 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, Politecnico di Milano
Sandra Krueger, Freie Universitaet Berlin
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

See Also

decompTumor2Sig

Examples

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

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

### read 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 <- readGenomesFromVCF(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` *Determine differences between a given signature and a set of target signatures.*

Description

'determineSignatureDistances()' determines all distances (i.e., differences) between a given signature (of type Alexandrov or Shiraishi) and a set of target 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 details below).

Usage

```
determineSignatureDistances(fromSignature, toSignatures,
                                    method="euclidean")
```

Arguments

`fromSignature`	(Mandatory) A single signature of the Alexandrov (vector) or Shiraishi type (data frame or matrix).
`toSignatures`	(Mandatory) The list of target signatures for which to compute the distances to `fromSignature`. These target signatures must be of the same type and format as `fromSignature`.
`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"`.

Details

Distances that can be used are:

`"frobenius"`	Forbenius distance between real-valued matrices (or Shiraishi signatures) `A` and `B`: `F = sqrt(trace( (A-B) %*% t(A-B) ))`
`"rss"`	Residual sum of squares (i.e., squared error): `rss = sum((A-B)^2)`
`"euclidean"`	(see `?dist` for details)
`"maximum"`	(see `?dist` for details)
`"manhattan"`	(see `?dist` for details)
`"canberra"`	(see `?dist` for details)
`"binary"`	(see `?dist` for details)
`"minkowski"`	(see `?dist` for details)

Value

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

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

See Also

decompTumor2Sig
mapSignatureSets

Examples

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

### 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")
```

`determineTypeNumBasesAndTrDir` *determineTypeNumBasesAndTrDir (internal function)*

Description

For a given signature or genome representation: determine the type (Shiraishi or Alexandrov), the number of bases, and whether transcription-strand information is included.

Usage

```
determineTypeNumBasesAndTrDir(mutData)
```

Arguments

mutData	The mutation frequency data of the signature or genome.

Value

A list object composed of: "type"=type of signature or genome; "numBases"=number of bases of the sequence patterns; and "trDir"=logical value indicating whether transcription-strand information has been considered in the mutation frequency data.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

---

downgradeShiraishiSignatures

Downgrade Shiraishi-type signatures.

Description

'downgradeShiraishiSignatures()' downgrades/trims signatures of the Shiraishi type by discarding flanking bases (reducing the length of the sequence pattern) and/or the transcription direction. The downgrade doesn't pose a problem 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 and smaller than the current number of bases of the signatures. 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
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

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
### transcription direction
downgradeShiraishiSignatures(signatures, numBases=3, removeTrDir=TRUE)
```

`evaluateDecompositionQuality` *Evaluate tumor decomposition quality.*

Description

'evaluateDecompositionQuality()' evaluates the quality of the decomposition into exposures of a single tumor. The function evaluates the quality of a decomposition obtained from the function `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, or exposures, of a single tumor as provided by `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 correlation coefficient between the reconstructed and observed mutation frequencies, and the explained variance; or alternatively, a plot with these measurements (see option `plot` above).

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

See Also

decompTumor2Sig
decomposeTumorGenomes
composeGenomesFromExposures
computeExplainedVariance

Examples

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

### 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]])
```

---

getBestDecomp4Ksignatures

getBestDecomp4Ksignatures (internal function)

Description

Get the best decomposition for a subset of k signatures.

Usage

```
getBestDecomp4Ksignatures(genome, signatures, k,
constrainToMaxContribution=FALSE, tolerance=0.1)
```

Arguments

genome	Genome for which to approximate the decomposition.
signatures	The whole set of signatures (from which to choose a subset signatures.
k	Number of signatures to use (subset size).
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.

Value

A list object containing: k=number of signatures; explVar=variance explained by these signatures; sigList=list of the signatures; decomposition=decomposition (exposures) obtained with these signatures.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

getCountData *getCountData (internal function)*

Description

'getCountData()' extracts the content of the countData slot (numeric values) from a pmsignature::MutationFeatureData object.

Usage

```
getCountData(x)
```

Arguments

x	Object from which to get the slot data.

Value

Slot data.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

getFeatVectList *getFeatVectList (internal function)*

Description

'getFeatVectList()' extracts the content of the featureVectorList slot from a pmsignature::MutationFeatureData object.

Usage

```
getFeatVectList(x)
```

Arguments

x	Object from which to get the slot data.

Value

Slot data.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

---

getGenomesFromMutFeatData

Get genomes (mutation frequencies) from MutationFeatureData.

Description

'getGenomesFromMutFeatData()' takes a MutationFeatureData object (mutation count data) as read by the 'pmsignature' package (e.g., by pmsignature::readMPFile, version 0.3.0) and extracts the mutation counts of the genomes therein. For passing the genomes to decomposeTumorGenomes, the mutation counts must be normalized to mutation frequencies, 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

```
getGenomesFromMutFeatData(mutFeatData, normalize=TRUE)
```

Arguments

mutFeatData	(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 mutation frequencies (or mutation counts if not normalized), one object per genome. The format is either according to the Shiraishi or the Alexandrov model, depending on how the mutation data was loaded with pmsignature.

Author(s)

Rosario M. Piro, Politecnico di Milano
Sandra Krueger, Freie Universitaet Berlin
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

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 <- getGenomesFromMutFeatData(G, normalize=TRUE)
```

---

getNumFlankingBases *getNumFlankingBases (internal function)*

Description

'getNumFlankingBases()' extracts the content of the flankingBasesNum slot (numeric value) from a pmsignature::MutationFeatureData or from a pmsignature::EstimatedParameters object.

Usage

```
getNumFlankingBases(x)
```

Arguments

x	Object from which to get the slot data.

Value

Slot data.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

---

getNumSignatures *getNumSignatures (internal function)*

Description

'getNumSignatures()' extracts the content of the `signatureNum` slot (numeric value) from a `pmsignature::EstimatedParam` object.

Usage

```
getNumSignatures(x)
```

Arguments

x	Object from which to get the slot data.

Value

Slot data.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

`getSampleList` *getSampleList (internal function)*

Description

'getSampleList()' extracts the content of the `sampleList` slot (strings) from a `pmsignature::MutationFeatureData` object.

Usage

```
getSampleList(x)
```

Arguments

x	Object from which to get the slot data.

Value

Slot data.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

getSigFeatDist *getSigFeatDist (internal function)*

Description

getSigFeatDist‘()‘ extracts the content of the `signatureFeatureDistribution` slot from a `pmsignature::EstimatedParam` object.

Usage

```
getSigFeatDist(x)
```

Arguments

`x`	Object from which to get the slot data.

Value

Slot data.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

getSigFromEstParam *getSigFromEstParam (internal function)*

Description

‘getSigFromEstParam()‘ extracts a specific signature from the `signatureFeatureDistribution` slot of a `pmsignature::EstimatedParameters` object.

Usage

```
getSigFromEstParam(x, signum)
```

Arguments

x	Object from which to get the signature.
signum	Number (1..N) of the signature to be extracted.

Value

Signature.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

---

getSignaturesFromEstParam

Get signatures from an `EstimatedParameters` *object.*

Description

'getSignaturesFromEstParam()' takes an `EstimatedParameters` object (signatures data) as computed by the 'pmsignature' package (by `pmsignature::getPMSignature`; version 0.3.0) and extracts the signature information. The signatures can then be passed to `decomposeTumorGenomes`.

Usage

```
getSignaturesFromEstParam(Param)
```

Arguments

Param	(Mandatory) A `pmsignature::EstimatedParameters` object as those produced by the de novo signature construction method `pmsignature::getPMSignature`.

Value

A list of Shiraishi signatures, one object per signature. Please see `readShiraishiSignatures` or the `decompTumor2Sig` vignette for more information on the format of Shiraishi signatures.

Author(s)

Rosario M. Piro, Politecnico di Milano
Sandra Krueger, Freie Universitaet Berlin
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

See Also

decompTumor2Sig
readShiraishiSignatures

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 <- getSignaturesFromEstParam(Param)
```

---

getSigType *getSigType (internal function)*

Description

'getSigType()' extracts the content of the type slot (string) from a pmsignature::EstimatedParameters or a pmsignature::MutationFeatureData object.

Usage

```
getSigType(x)
```

Arguments

x	Object from which to get the slot data.

Value

Slot data.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

---

haveTrDir *haveTrDir (internal function)*

Description

'haveTrDir()' extracts the content of the `transcriptionDirection` slot (logical value) from a `pmsignature::MutationFeatureData` object.

Usage

```
haveTrDir(x)
```

Arguments

`x`	Object from which to get the slot data.

Value

Slot data.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

`is.probability.data.frame`
is.probability.data.frame (internal function)

Description

'is.probability.data.frame()' checks whether the input object is a numeric data.frame of probabilities with a total sum of 1 for every row. Each row must have 6 columns (for Shiraishi format).

Usage

```
is.probability.data.frame(x)
```

Arguments

x	Object to be checked.

Value

Logical value (true or false).

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

is.probability.data.frame.list

is.probability.data.frame.list (internal function)

Description

'is.probability.data.frame.list()' checks whether the input object is a list of numeric data.frame objects. Must have 6 columns (for Shiraishi format).

Usage

```
is.probability.data.frame.list(x)
```

Arguments

x	Object to be checked.

Value

Logical value (true or false).

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

is.probability.matrix

is.probability.matrix (internal function)

Description

'is.probability.matrix()' checks whether the input object is a numeric matrix of probabilities with a total sum of 1 for every row. Each row must have 6 columns (for Shiraishi format).

Usage

```
is.probability.matrix(x)
```

Arguments

x	Object to be checked.

Value

Logical value (true or false).

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

---

is.probability.matrix.list

is.probability.matrix.list (internal function)

Description

'is.probability.matrix.list()' checks whether the input object is a list of numeric matrices. Must have 6 columns (for Shiraishi format).

Usage

```
is.probability.matrix.list(x)
```

Arguments

x	Object to be checked.

Value

Logical value (true or false).

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

`is.probability.object` *is.probability.object (internal function)*

Description

'is.probability.object()' checks whether the input object is a numeric vector, matrix of data.frame of probabilities with a total sum of 1 for every row. Matrices and data.frames must have 6 columns (for Shiraishi format).

Usage

```
is.probability.object(x)
```

Arguments

x	Object to be checked.

Value

Logical value (true or false).

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

is.probability.vector *is.probability.vector (internal function)*

Description

'is.probability.vector()' checks whether the input object is a numeric vector of probabilities with a total sum of 1.

Usage

```
is.probability.vector(x)
```

Arguments

x	Object to be checked.

Value

Logical value (true or false).

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

is.probability.vector.list *is.probability.vector.list (internal function)*

Description

'is.probability.vector.list()' checks whether the input object is a list of numeric vectors.

Usage

```
is.probability.vector.list(x)
```

Arguments

x	Object to be checked.

Value

Logical value (true or false).

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

---

isAlexandrovSet *isAlexandrovSet*

Description

'isAlexandrovSet()' checks whether the input object is a set (list) of numeric objects compatible with the Alexandrov format (probability vectors; sum up to 1). NOTE: These can also be genomes compatible with the Alexandrov format!

Usage

```
isAlexandrovSet(x)
```

Arguments

x	Object to be checked.

Value

Logical value (true or false).

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

See Also

decompTumor2Sig
readAlexandrovSignatures
isSignatureSet

Examples

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

isAlexandrovSet(signAlexandrov)
```

---

isBackGround *isBackGround (internal function)*

Description

'isBackGround()' extracts the content of the isBackGround slot (logical value) from a pmsignature::EstimatedParameter object.

Usage

```
isBackGround(x)
```

Arguments

x	Object from which to get the slot data.

Value

Slot data.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

`isEstParamObject` *isEstParamObject (internal function)*

Description

'isEstParamObject()' checks whether an object is compatible to a `pmsignature::EstimatedParameters` object (version 0.3.0), i.e., whether it contains the same info in the same format.

Usage

```
isEstParamObject(x)
```

Arguments

x	Object for which to verify compatibility.

Value

Logical value (true or false).

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

isExposureSet *isExposureSet*

Description

'isExposureSet()' checks whether the input object is a set (list) of numeric objects compatible with exposure output obtained from `decomposeTumorGenomes`.

Usage

```
isExposureSet(x)
```

Arguments

x	Object to be checked.

Value

Logical value (true or false).

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

See Also

`decompTumor2Sig`
`decomposeTumorGenomes`

Examples

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

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

### read 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 <- readGenomesFromVCF(gfile, numBases=3, type="Alexandrov",
          trDir=FALSE, refGenome=refGenome, verbose=FALSE)

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

isExposureSet(exposures)
```

`isMutFeatDataObject` *isMutFeatDataObject (internal function)*

Description

'isMutFeatDataObject()' checks whether an object is compatible to a `pmsignature::MutationFeatureData` object (version 0.3.0), i.e., whether it contains the same info in the same format.

Usage

```
isMutFeatDataObject(x)
```

Arguments

x	Object for which to verify compatibility.

Value

Logical value (true or false).

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

isShiraishiSet *isShiraishiSet*

Description

'isShiraishiSet()' checks whether the input object is a set (list) of numeric objects compatible with the Shiraishi format (matrices or data.frames of probabilities; 6 columns, each row sums up to 1). NOTE: These can also be genomes compatible with the Shiraishi format!

Usage

```
isShiraishiSet(x)
```

Arguments

x	Object to be checked.

Value

Logical value (true or false).

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

See Also

decompTumor2Sig
isSignatureSet
readShiraishiSignatures

Examples

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

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

```
isShiraishiSet(signShiraishi)
```

`isSignatureSet` *isSignatureSet*

Description

'isSignatureSet()' checks whether the input object is a set (list) of numeric objects compatible with either the Alexandrov format (probability vectors; see `isAlexandrovSet`) or the Shiraishi format (matrices or data.frames of probabilities; see `isShiraishiSet`). NOTE: These can also be genomes compatible with one of the two formats!

Usage

```
isSignatureSet(x)
```

Arguments

x	Object to be checked.

Value

Logical value (true or false).

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

See Also

`decompTumor2Sig`
`isAlexandrovSet`
`isShiraishiSet`

Examples

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

isSignatureSet(signAlexandrov)
```

mapSignatureSets *Map one signature set to another.*

Description

'mapSignatureSets()' determines 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 identify best matches while allowing multiple signatures to be mapped to the same target signature if it is the best match for more than one signature. Different distance measures can be used (see details below).

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 iterative 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.

Details

Distances that can be used are:

`"frobenius"`	Forbenius distance between real-valued matrices (or Shiraishi signatures) `A` and `B`: `F = sqrt(trace( (A-B) %*% t(A-B) ))`
`"rss"`	Residual sum of squares (i.e., squared error): `rss = sum((A-B)^2)`

"euclidean"	(see ?dist for details)
"maximum"	(see ?dist for details)
"manhattan"	(see ?dist for details)
"canberra"	(see ?dist for details)
"binary"	(see ?dist for details)
"minkowski"	(see ?dist for details)

Value

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

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

See Also

decompTumor2Sig
determineSignatureDistances

Examples

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

### 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)
```

`plotAlexandrovModel` *plotAlexandrovModel (internal function)*

Description

'plotAlexandrovModel()' plots a single signature or the mutation frequency data for a single genome of the Alexandrov-type model.

Usage

```
plotAlexandrovModel(mutData, numBases, trDir, colors = NULL,
strip = NULL)
```

Arguments

`mutData`	(Mandatory) The signature or genome mutation frequency data to be plotted.
`numBases`	The number of bases of the sequence pattern.
`trDir`	Logical value specifying whether transcription strand information is present.
`colors`	Vector of six colors to be used for the base change data. If `NULL` (default), the colors are set to those used by the COSMIC website.
`strip`	Background color for strip labels. If `NULL` (default), "papayawhip" will be used.

Value

Returns (or draws) a plot according to the Alexandrov model of mutational signatures.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

plotDecomposedContribution
Plot the decomposition (contributions/exposures) of a tumor genome.

Description

'plotDecomposedContribution()' plots the decomposition of a tumor genome, i.e., the contributions/exposures obtained from `decomposeTumorGenomes` for a set of signatures.

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 signature names will be taken from the exposure object (decomposition) or 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 be NA if they have been determined with a greedy search.

Value

Returns (or draws) a plot of the decomposed tumor genome (i.e., contributions of the single signatures).

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

See Also

decompTumor2Sig
decomposeTumorGenomes

Examples

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

### 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 *Plot the explained variance as a function of the number of signatures*

Description

'plotExplainedVariance()' plots the explained variance of a single tumor genome's mutation patterns as a function of the number of signatures (increasing subsets of signatures) used for decomposition. For each number K of signatures, the highest variance explained by possible subsets of K signatures will be plotted (full or greedy search, see below). This can help to evaluate what minimum threshold for the explained variance can be used to decompose tumor genomes with the function `decomposeTumorGenomes`.

Usage

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

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 which is sufficient to explain at least the specified fraction of the variance of the genome's mutation patterns. 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<=`maxNumSignatures`.

`greedySearch`
: (Optional) If `greedySearch` is set to `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 `maxNumSignatures` is reached). NOTE: while this is only an approximation, it is highly recommended for large sets of signatures (>15)!

Value

Returns (or draws) a plot of the explained variance as a function of the number of signatures.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

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` *Plot mutation frequency data of a mutational signature or tumor genome.*

Description

'plotMutationDistribution()' plots a single signature or the mutation frequency data for a single genome. This works for signatures or genome data of both the Shiraishi and the Alexandrov type.

Usage

```
plotMutationDistribution(mutData, colors = NULL, strip = NULL)
```

Arguments

`mutData`	(Mandatory) The signature or genome mutation frequency data to be plotted. This can either be a matrix (Shiraishi model) or a numeric vector (Alexandrov model).
`colors`	Vector of colors to be used for the base change data. For Alexandrov-type data, this vector must contain six elements (one per base change). For Shiraishi-type data, this vector must contain four elements (one per base). If `NULL` (default), for Alexandrov-type data, the colors are set to those used by the COSMIC website; for Shiraishi-type data, the consensus base colors for sequence logos will be used.
`strip`	Background color for strip labels; used only for Alexandrov-type data. If `NULL` (default), "papayawhip" will be used.

Value

Returns (or draws) a plot according to the Alexandrov or Shiraishi model of mutational signatures.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

See Also

decompTumor2Sig

Examples

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


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

### plot the first Alexandrov signature
plotMutationDistribution(signatures[[1]])

### read 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 <- readShiraishiSignatures(sigfiles)

### plot the first Shiraishi signature
plotMutationDistribution(signatures[[1]])
```

---

plotShiraishiModel *plotShiraishiModel (internal function)*

Description

'plotShiraishiModel()' plots a single signature or the mutation frequency data for a single genome of the Shiraishi-type model.

Usage

```
plotShiraishiModel(mutData, numBases, trDir, colors = NULL)
```

Arguments

mutData	(Mandatory) The signature or genome mutation frequency data to be plotted.
numBases	The number of bases of the sequence pattern.
trDir	Logical value specifying whether transcription strand information is present.
colors	Vector of four colors to be used for the base data. If NULL (default), the colors are consensus colors used for sequence logos.

Value

Returns (or draws) a plot according to the Shiraishi model of mutational signatures.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

---

processMultipleSigSets

processMultipleSigSets (internal function)

Description

Performs the quadratic programming/exposure prediction for multiple subsets (of size k) of mutational signatures and returns information on the best subset (highest explained variance). This function is used by getBestDecomp4Ksignatures and addBestSignatureToSubset.

Usage

```
processMultipleSigSets(genome, signatures, sigCombn, k,
constrainToMaxContribution=FALSE, tolerance=0.1)
```

Arguments

genome	Genome for which to approximate the decomposition.
signatures	The whole set of signatures (from which to choose a subset signatures.
sigCombn	The combinations of subsets of k signatures to use. Has to be the same format as generated by combn.
k	Number of signatures to use (subset size).
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.

Value

A list object containing: k=number of signatures; explVar=variance explained by these signatures; sigList=list of the signatures; decomposition=decomposition (exposures) obtained with these signatures.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

QPforSig *QPforSig (internal function)*

Description

Perform quadratic programming for signatures to determine exposures.

Usage

```
QPforSig(counts, signatures, constrainToMaxContribution=FALSE,
tolerance=0.1)
```

Arguments

counts	The genome's mutation frequencies (either in Alexandrov or Shiraishi format).
signatures	The signatures to be used for decomposition. Must be in the same format as the genome's counts.
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.

Value

The decomposition in form of an exposure vector (same order as signatures).

Author(s)

Rosario M. Piro, Politecnico di Milano
Sandra Krueger, Freie Universitaet Berlin
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

`readAlexandrovSignatures` *Read Alexandrov-type signatures (COSMIC format).*

Description

'readAlexandrovSignatures()' reads a set of Alexandrov-type signatures (COSMIC format) from a flat file or URL. Signatures must be specified in the tab-separated format used by the COSMIC website for signatures version 2 (March 2015), the comma-separated format used for signatures version 3 (May 2019), the Microsoft Excel 2007+ sheet used for version 3.1, or the tab-sperated format used for version 3.2 (see Details below). Excel sheets cannot be read from an URL and must be downloaded first.

Usage

```
readAlexandrovSignatures(file)
```

Arguments

`file`	(Mandatory) Can be a file name or an URL for download. Default: "https://raw.githubusercontent.com/vanallenlab/deconstruct_sigs_py/refs/heads/master/deconstructSigs/d (COSMIC signatures v2).

Details

For details on the accepted signature formats, see below or
http://cancer.sanger.ac.uk/cosmic/signatures_v2 -> "Download signatures" for version 2,
https://www.synapse.org/#!Synapse:syn12009743 for version 3,
https://cancer.sanger.ac.uk/sigs-assets-20/COSMIC_Mutational_Signatures_v3.1.xlsx for version 3.1,
and https://cancer.sanger.ac.uk/signatures/ for version 3.2. For versions 3, 3.1 and 3.2, only Single Base Substitution (SBS) signatures can be used.

COSMIC format for Alexandrov signatures, version 2:

Subst.	Trinucleotide	Mutation Type	Signature 1	Signature 2	...
C>A	ACA	A[C>A]A	0.0110983262	0.0006827082	...
C>A	ACC	A[C>A]C	0.0091493407	0.0006191072	...
C>A	ACG	A[C>A]G	0.0014900705	0.0000992790	...
C>A	ACT	A[C>A]T	0.0062338852	0.0003238914	...
[...]					

T>G	TTG	T[T>G]G	0.0020310769	0.0002066152	...
T>G	TTT	T[T>G]T	0.0040301281	0.0000235982	...

COSMIC/Synapse format for Alexandrov signatures, version 3 and 3.1:

```
Type,SubType,SBS1,SBS2,SBS3,SBS4,SBS5,SBS6, ...
C>A,ACA,8.86E-04,5.80E-07,2.08E-02,4.22E-02,1.20E-02,4.25E-04, ...
C>A,ACC,2.28E-03,1.48E-04,1.65E-02,3.33E-02,9.44E-03,5.24E-04, ...
C>A,ACG,1.77E-04,5.23E-05,1.75E-03,1.56E-02,1.85E-03,5.20E-05, ...
C>A,ACT,1.28E-03,9.78E-05,1.22E-02,2.95E-02,6.61E-03,1.80E-04, ...
[...]
T>G,TTG,5.83E-04,9.54E-05,8.05E-03,2.32E-03,6.94E-03,3.24E-04, ...
T>G,TTT,2.23E-16,2.23E-16,1.05E-02,5.68E-04,1.35E-02,1.01E-03, ...
```

Version 3.1 has assentially the same format as version 3, but is distributed as an Excel spread sheet.

COSMIC/Synapse format for Alexandrov signatures, version 3.2:

Type	SBS1	SBS2	...
A[C>A]A	0.0110983262	0.0006827082	...
A[C>A]C	0.0091493407	0.0006191072	...
A[C>A]G	0.0014900705	0.0000992790	...
A[C>A]T	0.0062338852	0.0003238914	...
[...]			
T[T>G]G	0.0020310769	0.0002066152	...
T[T>G]T	0.0040301281	0.0000235982	...

Value

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

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

See Also

decompTumor2Sig
readShiraishiSignatures

Examples

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

---

readGenomesFromMPF *Read tumor genomes from an MPF file (Mutation Position Format).*

Description

'readGenomesFromMPF()' reads somatic mutations of a single tumor genome (sample) or a set of genomes from an MPF file (Mutation Position Format; see details below) and determines the mutation frequencies according to a specific model of mutational signatures (Alexandrov or Shiraishi).

Usage

```
readGenomesFromMPF(file, numBases=5, type="Shiraishi", trDir=TRUE,
enforceUniqueTrDir=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. Must be odd. 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 genomic regions with a defined transcription direction can be considered. Default: TRUE
enforceUniqueTrDir	(Optional) Used only if trDir is TRUE. If enforceUniqueTrDir is TRUE (default), then mutations which map to a region with multiple overlapping genes with opposing transcription directions will be excluded from the analysis. If FALSE, the transcript direction encountered first in the transcript database (see transcriptAnno) is assigned to the mutation. The latter was the behavior until version 1.3.5 of decompTumor2Sig and is also the behavior of pmsignature.

	However, it is preferable to exclude these mutations from the count (default) because from mutation data alone it cannot be inferred which of the two genes has the higher transcriptional activity which might potentially be linked to the occurrence of the mutation. (If you are unsure, use the default setting; this option exists mostly for backward compatibility with older versions.)
`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`

Details

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]

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
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

See Also

`decompTumor2Sig`
`decomposeTumorGenomes`
`readGenomesFromVCF`
`getGenomesFromMutFeatData`

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

### read breast cancer genomes from Nik-Zainal et al (PMID: 22608084)
gfile <- system.file("extdata", "Nik-Zainal_PMID_22608084-MPF.txt.gz",
         package="decompTumor2Sig")
genomes <- readGenomesFromMPF(gfile, numBases=5, type="Shiraishi",
         trDir=TRUE, enforceUniqueTrDir=TRUE, refGenome=refGenome,
         transcriptAnno=transcriptAnno, verbose=FALSE)
```

`readGenomesFromVCF` *Read tumor genomes from a VCF file (Variant Call Format).*

Description

'readGenomesFromVCF()' reads somatic mutations of a single tumor genome (sample) or a set of genomes from a VCF file (Variant Call Format) and determines the mutation frequencies according to a specific model of mutational signatures (Alexandrov or Shiraishi).

Usage

```
readGenomesFromVCF(file, numBases=5, type="Shiraishi", trDir=TRUE,
enforceUniqueTrDir=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. Must be odd. 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 genomic regions with a defined transcription direction can be considered. Default: `TRUE`
`enforceUniqueTrDir`	(Optional) Used only if `trDir` is `TRUE`. If `enforceUniqueTrDir` is TRUE (default), then mutations which map to a region with multiple overlapping genes with opposing transcription directions will be excluded from the analysis. If `FALSE`, the transcript direction encountered first in the transcript database (see `transcriptAnno`) is assigned to the mutation. The latter was the behavior until version 1.3.5 of `decompTumor2Sig` and is also the behavior of `pmsignature`. However, it is preferable to exclude these mutations from the count (default) because from mutation data alone it cannot be inferred which of the two genes has

	the higher transcriptional activity which might potentially be linked to the occurrence of the mutation. (If you are unsure, use the default setting; this option exists mostly for backward compatibility with older versions.)
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
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

See Also

decompTumor2Sig
decomposeTumorGenomes
readGenomesFromMPF
getGenomesFromMutFeatData

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

### read 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 <- readGenomesFromVCF(gfile, numBases=5, type="Shiraishi",
         trDir=TRUE, enforceUniqueTrDir=TRUE, refGenome=refGenome,
```

```
            transcriptAnno=transcriptAnno, verbose=FALSE)
```

`readShiraishiSignatures` *Read a set of Shiraishi signatures.*

Description

'readShiraishiSignatures()' reads one or more Shiraishi-type signatures from flat files (one file per signature). The signatures must be specified as matrices without headers and row names (see details below).

Usage

```
readShiraishiSignatures(files)
```

Arguments

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

Details

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 occurrences 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

Value

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

Author(s)

Rosario M. Piro, Politecnico di Milano
Sandra Krueger, Freie Universitaet Berlin
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

See Also

decompTumor2Sig
readAlexandrovSignatures
getSignaturesFromEstParam

Examples

```
### read 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 <- readShiraishiSignatures(sigfiles)
```

`roundIntegerSum` *roundIntegerSum (internal function)*

Description

update a numeric vector such that it's composed of integers and its sum reaches a desired total. Positive or negative discrepancies are distributed proportionally between the summands.

Usage

```
roundIntegerSum(vec, targetSum)
```

Arguments

`vec`	Vector of integers.
`targetSum`	The target sum to be reached.

Value

The updated vector with the desired total sum.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

---

sameSignatureFormat *sameSignatureFormat*

Description

'sameSignatureFormat()' checks whether two input object are sets (lists) of numeric objects both compatible with the same signature format (probability vectors for Alexandrov signatures and probability matrices or data.frames for Shiraishi signatures). For Shiraishi signatures also the number of flanking bases and the presence of transcription-strand information are compared. For Alexandrov signatures also the number of triplet changes are compared.

Usage

```
sameSignatureFormat(x, y)
```

Arguments

x	First object to be checked.
y	Second object to be checked.

Value

Logical value (true or false).

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

See Also

decompTumor2Sig
isAlexandrovSet
isShiraishiSet

Examples

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

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

sameSignatureFormat(signAlexandrov, signShiraishi)
```

---

setNames4ShiraishiTable

setNames4ShiraishiTable (internal function)

Description

Set row and column names of a Shiraishi-model genome or signature table.

Usage

```
setNames4ShiraishiTable(table)
```

Arguments

table	The table to be named.

Value

The same table with named columns and rows.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

`setNumFlankingBases` *setNumFlankingBases (internal function)*

Description

'setNumFlankingBases()' serves as wrapper functions to change content of the `flankingBasesNum` slot (numeric value) of a `pmsignature::MutationFeatureData` object.

Usage

```
setNumFlankingBases(x, value)
```

Arguments

`x`	Object of which to change slot data.
`value`	Value to be written to the slot.

Value

The modified object `x`.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

setSigFeatDist *setSigFeatDist (internal function)*

Description

'setSigFeatDist()' serves as wrapper functions to change content of the `signatureFeatureDistribution` slot (numeric values) of a `pmsignature::EstimatedParameters` object.

Usage

```
setSigFeatDist(x, value)
```

Arguments

`x`	Object of which to change slot data.
`value`	Value to be written to the slot.

Value

The modified object `x`.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

setSigType *setSigType (internal function)*

Description

'setSigType()' serves as wrapper functions to change content of the `type` slot (string) of a `pmsignature::EstimatedParamet` object.

Usage

```
setSigType(x, value)
```

Arguments

x	Object of which to change slot data.
value	Value to be written to the slot.

Value

The modified object x.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

---

setTrDir *setTrDir (internal function)*

Description

'setTrDir()' serves as wrapper functions to change content of the transcriptionDirection slot (logical value) of a pmsignature::MutationFeatureData object.

Usage

```
setTrDir(x, value)
```

Arguments

x	Object of which to change slot data.
value	Value to be written to the slot.

Value

The modified object x.

Author(s)

Rosario M. Piro
Politecnico di Milano
Maintainer: Rosario M. Piro
E-Mail: <rmpiro@gmail.com> or <rosariomichael.piro@polimi.it>

References

http://rmpiro.net/decompTumor2Sig/
Krueger, Piro (2019) decompTumor2Sig: Identification of mutational signatures active in individual tumors. BMC Bioinformatics 20(Suppl 4):152.

Index

∗ **internal**
addBestSignatureToSubset, 6
adjustAlexandrovSignature, 7
adjustShiraishiSignature, 8
buildGenomesFromMutationData, 10
buildSortedAlexandrovSignaturePatternList, 12
compNucFreq, 13
computeFrobeniusNorm, 17
computeRSS, 17
convAlx2Shi, 18
convertSeqFreqToBaseFreq, 22
determineTypeNumBasesAndTrDir, 27
getBestDecomp4Ksignatures, 31
getCountData, 33
getFeatVectList, 33
getNumFlankingBases, 35
getNumSignatures, 36
getSampleList, 37
getSigFeatDist, 38
getSigFromEstParam, 38
getSigType, 40
haveTrDir, 41
is.probability.data.frame, 42
is.probability.data.frame.list, 43
is.probability.matrix, 43
is.probability.matrix.list, 44
is.probability.object, 45
is.probability.vector, 46
is.probability.vector.list, 46
isBackGround, 48
isEstParamObject, 49
isMutFeatDataObject, 51
plotAlexandrovModel, 56
plotShiraishiModel, 61
processMultipleSigSets, 62
QPforSig, 64
roundIntegerSum, 72
setNames4ShiraishiTable, 74
setNumFlankingBases, 75
setSigFeatDist, 76
setSigType, 76
setTrDir, 77

addBestSignatureToSubset, 6
adjustAlexandrovSignature, 7
adjustShiraishiSignature, 8
adjustSignaturesForRegionSet, 8

buildGenomesFromMutationData, 10
buildSortedAlexandrovSignaturePatternList, 12

compNucFreq, 13
composeGenomesFromExposures, 14, *31*
computeExplainedVariance, 15, *31*, *59*
computeFrobeniusNorm, 17
computeRSS, 17
convAlx2Shi, 18
convertAlexandrov2Shiraishi, 19
convertGenomesFromVRanges, 20
convertSeqFreqToBaseFreq, 22

decomposeTumorGenomes, *9, 10*, *15, 16*, *21*, 23, *31*, *50*, *58, 59*, *68*, *70*
decompTumor2Sig, *10*, *15, 16*, *20, 21*, *25*, *27*, *29*, *31*, *35*, *40*, *48*, *50*, *52, 53*, *55*, *58, 59*, *61*, *67, 68*, *70*, *72*, *74*
decompTumor2Sig *(*decompTumor2Sig-package*)*, 3
decompTumor2Sig-package, 3
determineSignatureDistances, 26, *55*
determineTypeNumBasesAndTrDir, *22*, 27
downgradeShiraishiSignatures, 28

evaluateDecompositionQuality, 30

getBestDecomp4Ksignatures, 31
getCountData, 33
getFeatVectList, 33

getGenomesFromMutFeatData, *21*, 34, *68*, *70*
getNumFlankingBases, 35
getNumSignatures, 36
getSampleList, 37
getSigFeatDist, 38
getSigFromEstParam, 38
getSignaturesFromEstParam, 39, *72*
getSigType, 40

haveTrDir, 41

is.probability.data.frame, 42
is.probability.data.frame.list, 43
is.probability.matrix, 43
is.probability.matrix.list, 44
is.probability.object, 45
is.probability.vector, 46
is.probability.vector.list, 46
isAlexandrovSet, 47, *53*, *74*
isBackGround, 48
isEstParamObject, 49
isExposureSet, 50
isMutFeatDataObject, 51
isShiraishiSet, 52, *53*, *74*
isSignatureSet, *48*, *52*, 53

mapSignatureSets, *27*, 54

plotAlexandrovModel, 56
plotDecomposedContribution, 57
plotExplainedVariance, *16*, 58
plotMutationDistribution, 60
plotShiraishiModel, 61
processMultipleSigSets, 62

QPforSig, 64

readAlexandrovSignatures, *20*, *48*, 65, *72*
readGenomesFromMPF, *21*, 67, *70*
readGenomesFromVCF, *21*, *68*, 69
readShiraishiSignatures, *20*, *40*, *52*, *67*, 71
roundIntegerSum, 72

sameSignatureFormat, 73
setNames4ShiraishiTable, 74
setNumFlankingBases, 75
setSigFeatDist, 76
setSigType, 76
setTrDir, 77