Package 'MutaliskR'

March 26, 2021

Type Package

Title MutaliskR

Description MutaliskR is a general-purpose mutational signature identification tool that uses maximum likelihood estimation to perform linear regression.

Version 0.4.5

Authors Andy Jinseok Lee

Maintainer Andy Jinseok Lee <andyjinseoklee@gmail.com>

Depends R (>= 3.5.0)

Imports ggplot2,
ggpubr,
svglite,
dplyr,
plyr,
stringr,
bedr,
GenomicRanges,
DNAcopy,
lsa,
scales,
doParallel,
foreach,
parallel

biocViews CancerGenomics,
MutationalSignatures,
SomaticMutation

URL https://github.com/Honglab-Research/MutaliskR

Encoding UTF-8

LazyData true

License MIT + file LICENSE

RoxygenNote 7.1.1

R topics documented:

`AssignIndelPCAWGClassification` *Assigns PCAWG classification of an indel*

Description

This function returns PCAWG classification data of an indel.

Usage

```
AssignIndelPCAWGClassification(x)
```

Arguments

`x`	One row of data.frame the function PreprocessIndel.

Value

A data.frame with the following columns:

`PCAWG_Indel_Class`
: PCAWG indel mutation type (e.g. 'DEL_C_1_1').

`Microhomology_Class`
: PCAWG microhomology mutation type (e.g. 'DEL_MH_2_1').

`Microhomology_Size`
: Microhomology size.

`Microhomology_Seq`
: Microhomology sequence.

`Microhomology_Direction`
: Microhomology direction.

`EstimateMaximumLikelihood` *Performs maximum likelihood estimation*

Description

This function performs maximum likelihood estimation of a given identification setup.

Usage

```
EstimateMaximumLikelihood(
  mutation.frequencies,
  signature.weights,
  num.mutations,
  min.probability = 0.01
)
```

Arguments

`mutation.frequencies`	A numeric vector corresponding to the frequency of each mutation type
`signature.weights`	A matrix of signature weights.
`num.mutations`	An integer value that specifies the number of mutations
`min.probability`	A float value that specifies the minimum probability to attribute to a signature.

Value

A list with the following items:

`Cosine_Similarity`	Cosine similarity score between mutation.frequncies and Reconstructed_Spectrum
`Par`	A numeric vector of contribution weights of identified signatures.
`Par_Normalized`	A normalized (0 to 1) numeric vector of contribution weights of identified signatures.
`Reconstructed_Spectrum`	A numeric vector of reconstructed spectrum.
`Reconstructed_Spectrum_Normalized`	A normalized (0 to 1) numeric vector of reconstructed spectrum.
`Residuals`	A numeric vector of residual spectrum.
`Residuals_Normalized`	A normalized (0 to 1) numeric vector of residual spectrum.
`RSS`	Residual sum of squares (of Residuals).
`RSS_Normalized`	Residual sum of squares (of Residuals_Normalized).

GetComplementBase *Returns complement base*

Description

This function returns the complement base of a nucleotide

Usage

```
GetComplementBase(nucleotide)
```

Arguments

nucleotide	Nucleotide base.

Value

"A", "C", "G", or "T"

GetComplementBases *Returns complement bases*

Description

This function returns the complement bases of a nucleotide sequence

Usage

```
GetComplementBases(nucleotides)
```

Arguments

nucleotides	Nucleotide bases (e.g. "AG").

Value

e.g. "TC"

`GetCosmicDbsSignaturesData` *Returns the COSMIC v3.1 DBS reference signature data*

Description

This function returns the COSMIC v3.1 doublet base substitution (DBS) mutational signatures data.

Usage

```
GetCosmicDbsSignaturesData()
```

Value

A data.frame of the PCAWG DBS mutational signatures.

`GetCosmicDbsSignaturesNames` *Return the COSMIC v3.1 DBS reference signature names*

Description

This function returns the COSMIC v3.1 DBS mutational signatures names.

Usage

```
GetCosmicDbsSignaturesNames()
```

Value

Returns a character vector of all COSMIC v3.1 DBS mutational signature names.

`GetCosmicIndelSignaturesData` *Returns the COSMIC v3.1 indel reference signature data*

Description

This function returns the COSMIC v3.1 small insertion and deletion (indel) mutational signatures data.

Usage

```
GetCosmicIndelSignaturesData()
```

Value

A data.frame of the COSMIC v3.1 indel mutational signatures.

`GetCosmicIndelSignaturesNames` *Returns the COSMIC v3.1 indel signature names*

Description

This function returns the COSMIC v3.1 indel mutational signatures names.

Usage

```
GetCosmicIndelSignaturesNames()
```

Value

Returns a character vector of all COSMIC v3.1 indel mutational signature names.

`GetCosmicSbsSignaturesData` *Returns the COSMIC v3.1 SBS reference signature data*

Description

This function returns the COSMIC v3.1 single base substitution (SBS) mutational signatures data.

Usage

```
GetCosmicSbsSignaturesData()
```

Value

A data.frame of the COSMIC v3.1 SBS mutational signatures.

`GetCosmicSbsSignaturesNames` *Return the COSMIC v3.1 SBS reference signature names*

Description

This function returns the COSMIC v3.1 SBS mutational signatures names.

Usage

```
GetCosmicSbsSignaturesNames()
```

Value

Returns a character vector of all COSMIC v3.1 SBS mutational signature names.

GetDbsSignaturesColors

Returns the DBS signatures colors and etiologies

Description

This function returns the DBS mutational signatures colors and etiologies.

Usage

```
GetDbsSignaturesColors()
```

Value

A data.frame of the DBS mutational signatures colors and etiologies.

GetDbsSignaturesPlotTheme

Fetches DBS plotting theme

Description

This function returns the DBS plotting theme elements.

Usage

```
GetDbsSignaturesPlotTheme(
  x.axis.labels = PLOT.DBS.X.AXIS.LABELS,
  x.axis.text.size = 10,
  x.axis.text.angle = 90,
  x.axis.text.hjust = 1,
  x.axis.text.vjust = 0.5,
  y.axis.text.size = 12,
  y.axis.title.size = 12,
  y.axis.min = 0,
  y.axis.max = 1,
  y.axis.title = "Mutation Fraction",
  groups = DBS.MUTATION.TYPES.GROUPS,
  strip.labels = PLOT.DBS.STRIP.LABELS,
  strip.colors = PLOT.DBS.STRIP.COLORS,
  strip.text.colors = PLOT.DBS.STRIP.TEXT.COLORS,
  strip.text.size = 12,
  legend.title = "Mutation Type",
  legend.title.size = 12,
  legend.text.size = 10,
  legend.labels = PLOT.DBS.LEGEND.LABELS
)
```

Arguments

x.axis.labels	A character vector used to label the x-axis.
x.axis.text.size	x-axis text size.
x.axis.text.angle	x-axis text angle.
y.axis.text.size	y-axis text size.
y.axis.title.size	y-axis title size.
y.axis.min	y-axis lower-bound (minimum range for limits).
y.axis.max	y-axis upper-bound (maximum range for limits).
y.axis.title	y-axis title text.
groups	A character vector of group variable values.
strip.labels	A character vector used to label the strip.
strip.colors	A character vector of hex values used to color the strip.
strip.text.colors	A character vector of hex values used to color the strip text.
strip.text.size	Strip (facet) text size.
legend.title	Legend title text.
legend.title.size	Legend title text size.
legend.labels	A character vector used to label the legend items.

Value

A data.frame of the DBS signature plot theme elements.

GetDeletionMicrohomologySize

Returns microhomology size of a deletion

Description

This function computes and returns the microhomology size of a small deletion.

Usage

```
GetDeletionMicrohomologySize(upstream.seq, downstream.seq, mutated.seq)
```

Arguments

upstream.seq	Upstream nucleotide sequence.
downstream.seq	Downstream nucleotide sequence.
mutated.seq	Indel nucleotide sequence

Value

A data.frame with the following columns:

`Microhomology_Seq`
: Microhomology nucleotide sequence.

`Microhomology_Size`
: Size (number of bases) of microhomology.

`Microhomology_Direction`
: Direction of microhomology.

`GetHsapiensEnsemblData` *Returns the hsapiens ensembl data*

Description

This function returns the hsapiens ensembl data.

Usage

```
GetHsapiensEnsemblData(version)
```

Arguments

`version`	Either "GRCh37" or "GRCh38".

`GetIndelPCAWGClassification` *Returns PCAWG classification of an indel*

Description

This function classifies an indel according to the PCAWG classification.

Usage

```
GetIndelPCAWGClassification(type, mutated.seq, upstream.seq, downstream.seq)
```

Arguments

`type`	Indel type. Either 'INS' or 'DEL'.
`mutated.seq`	Indel nucleotide sequence.
`upstream.seq`	Upstream nucleotide sequence.
`downstream.seq`	Downstream nucleotide sequence.

Value

A data.frame with the following columns:

`PCAWG_Indel_Class`
: PCAWG indel mutation type (e.g. 'DEL_C_1_0').

`Microhomology_Class`
: PCAWG microhomology mutation type (e.g. 'DEL_MH_2_1').

`Microhomology_Size`
: Microhomology size.

`Microhomology_Seq`
: Microhomology sequence.

`Microhomology_Direction`
: Microhomology direction.

`GetIndelRepeatSize` *Returns indel repeat size*

Description

This function computes and returns the repeat size of an indel variant.

Usage

```
GetIndelRepeatSize(downstream.seq, mutated.seq, type)
```

Arguments

`downstream.seq`	Downstream nucleotide sequence.
`mutated.seq`	Indel nucleotide sequence.
`type`	Indel type. Either 'INS' or 'DEL'.

Value

Repeat size (integer).

`GetIndelSignaturesColors` *Returns the indel signatures colors and etiologies*

Description

This function returns the indel mutational signatures colors and etiologies.

Usage

```
GetIndelSignaturesColors()
```

Value

A data.frame of the indel mutational signatures colors and etiologies.

GetIndelSignaturesPlotTheme *Fetches INDEL plotting theme*

Description

This function returns the INDEL plotting theme elements.

Usage

```
GetIndelSignaturesPlotTheme(
  x.axis.labels = PLOT.INDEL.X.AXIS.LABELS,
  x.axis.text.size = 10,
  x.axis.text.angle = 0,
  y.axis.text.size = 12,
  x.axis.text.hjust = 0.5,
  x.axis.text.vjust = 0.5,
  y.axis.title.size = 12,
  y.axis.min = 0,
  y.axis.max = 1,
  y.axis.title = "Mutation Fraction",
  groups = INDEL.MUTATION.TYPES.GROUPS,
  strip.labels = PLOT.INDEL.STRIP.LABELS,
  strip.colors = PLOT.INDEL.STRIP.COLORS,
  strip.text.colors = PLOT.INDEL.STRIP.TEXT.COLORS,
  strip.text.size = 12,
  legend.title = "Mutation Type",
  legend.title.size = 12,
  legend.text.size = 10,
  legend.labels = PLOT.INDEL.LEGEND.LABELS
)
```

Arguments

x.axis.labels	A character vector used to label the x-axis.
x.axis.text.size	x-axis text size.
x.axis.text.angle	x-axis text angle.
y.axis.text.size	y-axis text size.
y.axis.title.size	y-axis title size.
y.axis.min	y-axis lower-bound (minimum range for limits).
y.axis.max	y-axis upper-bound (maximum range for limits).
y.axis.title	y-axis title text.
groups	A character vector of group variable values.
strip.labels	A character vector used to label the strip.
strip.colors	A character vector of hex values used to color the strip.

`strip.text.colors`
: A character vector of hex values used to color the strip text.

`strip.text.size`
: Strip (facet) text size.

`legend.title`
: Legend title text.

`legend.title.size`
: Legend title text size.

`legend.labels`
: A character vector used to label the legend items.

Value

A data.frame of the INDEL signature plot theme elements.

GetKucabSbsSignaturesData

Returns the Kucab SBS reference signature data

Description

This function returns the Kucab single base subsitution (SBS) mutational signatures data.

Usage

```
GetKucabSbsSignaturesData()
```

Value

A data.frame of the Kucab SBS mutational signatures.

GetKucabSbsSignaturesNames

Return the Kucab SBS reference signature names

Description

This function returns the Kucab SBS mutational signatures names.

Usage

```
GetKucabSbsSignaturesNames()
```

`GetMmusculusEnsemblData` *Returns the mmusculus ensembl data*

Description

This function returns the mmusculus ensembl data.

Usage

```
GetMmusculusEnsemblData(version)
```

Arguments

`version`	Either "NCBIM37" or "GRCm38".

`GetPcawgDbsSignaturesData` *Returns the PCAWG DBS reference signature data*

Description

This function returns the PCAWG doublet base subsitution (DBS) mutational signatures data.

Usage

```
GetPcawgDbsSignaturesData()
```

Value

A data.frame of the PCAWG DBS mutational signatures.

`GetPcawgDbsSignaturesNames` *Return the DBS reference signature names*

Description

This function returns the PCAWG DBS mutational signatures names.

Usage

```
GetPcawgDbsSignaturesNames()
```

Value

Returns a character vector of all PCAWG DBS mutational signature names.

GetPcawgIndelSignaturesData

Returns the PCAWG indel reference signature data

Description

This function returns the PCAWG small insertion and deletion (indel) mutational signatures data.

Usage

```
GetPcawgIndelSignaturesData(version)
```

Arguments

version	Either "SigProfiler" or "SignatureAnalyzer".

Value

A data.frame of the PCAWG indel mutational signatures.

GetPcawgIndelSignaturesNames

Returns the PCAWG indel signature names

Description

This function returns the PCAWG indel mutational signatures names.

Usage

```
GetPcawgIndelSignaturesNames(version)
```

Arguments

version	Either "SigProfiler" or "SignatureAnalyzer".

Value

Returns a character vector of all PCAWG indel mutational signature names.

`GetPcawgPlatinumSbsSignaturesData` *Returns the PCAWG Platinum SBS reference signature data*

Description

This function returns the PCAWG Platinum single base subsitution (SBS) mutational signatures data (from Petljak et al., Cell 2019).

Usage

```
GetPcawgPlatinumSbsSignaturesData()
```

Value

A data.frame of the PCAWG Platinum SBS mutational signatures.

`GetPcawgPlatinumSbsSignaturesNames` *Return the PCAWG Platinum SBS reference signature names*

Description

This function returns the PCAWG Platinum SBS mutational signatures names.

Usage

```
GetPcawgPlatinumSbsSignaturesNames()
```

Value

Returns a character vector of all PCAWG Platinum SBS mutational signature names.

`GetPcawgSbsPentanucleotideSignaturesData` *Returns the PCAWG SBS penta-nucleotide reference signature data*

Description

This function returns the PCAWG single base substitution (SBS) pentanucleotide context mutational signatures data.

Usage

```
GetPcawgSbsPentanucleotideSignaturesData()
```

Value

A data.frame of the PCAWG SBS penta-nucleotide mutational signatures.

GetPcawgSbsPentanucleotideSignaturesNames

Return the PCAWG SBS penta-nucleotide reference signature names

Description

This function returns the PCAWG SBS penta-nucleotide mutational signatures names.

Usage

```
GetPcawgSbsPentanucleotideSignaturesNames()
```

Value

Returns a character vector of all PCAWG SBS penta-nucleotide mutational signature names.

GetPcawgSbsSignaturesData

Returns the PCAWG SBS reference signature data

Description

This function returns the PCAWG single base subsitution (SBS) mutational signatures data.

Usage

```
GetPcawgSbsSignaturesData(sequencing.type)
```

Arguments

`sequencing.type`
: Either 'WES' for whole-exome sequencing or 'WGS' for whole-genome sequencing.

Value

A data.frame of the PCAWG SBS mutational signatures.

GetPcawgSbsSignaturesNames *Return the PCAWG SBS reference signature names*

Description

This function returns the PCAWG SBS mutational signatures names.

Usage

```
GetPcawgSbsSignaturesNames(sequencing.type)
```

Arguments

sequencing.type	Either 'WES' for whole-exome sequencing or 'WGS' for whole-genome sequencing.

Value

Returns a character vector of all PCAWG SBS mutational signature names.

GetPetljakSbsSignaturesData *Returns the Petljak SBS reference signature data*

Description

This function returns the Petljak single base subsitution (SBS) mutational signatures data.

Usage

```
GetPetljakSbsSignaturesData()
```

Value

A data.frame of the Petljak SBS mutational signatures.

GetPetljakSbsSignaturesNames

Return the Petljak SBS reference signature names

Description

This function returns the Petljak SBS mutational signatures names.

Usage

```
GetPetljakSbsSignaturesNames(sequencing.type)
```

Value

Returns a character vector of all Petljak SBS mutational signature names.

GetPleguezuelosManzanoIndelSignaturesData

Returns the Pleguezuelos-Manzano indel reference signature data

Description

This function returns the Pleguezuelos-Manzano small insertion and deletion (indel) mutational signatures data.

Usage

```
GetPleguezuelosManzanoIndelSignaturesData()
```

Value

A data.frame of the Pleguezuelos-Manzano indel mutational signatures.

GetPleguezuelosManzanoIndelSignaturesNames

Returns the Pleguezuelos-Manzano indel signature names

Description

This function returns the Pleguezuelos-Manzano indel mutational signatures names.

Usage

```
GetPleguezuelosManzanoIndelSignaturesNames()
```

Value

Returns a character vector of all Pleguezuelos-Manzano indel mutational signature names.

GetPleguezuelosManzanoSbsSignaturesData

Returns the Pleguezuelos-Manzano SBS reference signature data

Description

This function returns the Pleguezuelos-Manzano single base subsitution (SBS) mutational signatures data.

Usage

```
GetPleguezuelosManzanoSbsSignaturesData()
```

Value

A data.frame of the Petljak Pleguezuelos-Manzano mutational signatures.

GetPleguezuelosManzanoSbsSignaturesNames

Return the Pleguezuelos-Manzano SBS reference signature names

Description

This function returns the Pleguezuelos-Manzano SBS mutational signatures names.

Usage

```
GetPleguezuelosManzanoSbsSignaturesNames()
```

GetSbsSignaturesColors

Returns the SBS signatures colors and etiologies

Description

This function returns the SBS mutational signatures colors and etiologies.

Usage

```
GetSbsSignaturesColors()
```

Value

A data.frame of the SBS mutational signatures colors and etiologies.

`GetSbsSignaturesPlotTheme` *Fetches SBS plotting theme*

Description

This function returns the SBS plotting theme elements.

Usage

```
GetSbsSignaturesPlotTheme(
  x.axis.labels = PLOT.SBS.X.AXIS.LABELS,
  x.axis.text.size = 10,
  x.axis.text.angle = 90,
  x.axis.text.hjust = 0.5,
  x.axis.text.vjust = 0.5,
  y.axis.text.size = 12,
  y.axis.title.size = 12,
  y.axis.min = 0,
  y.axis.max = 1,
  y.axis.title = "Mutation Fraction",
  groups = SBS.MUTATION.TYPES.GROUPS,
  strip.labels = PLOT.SBS.STRIP.LABELS,
  strip.colors = PLOT.SBS.STRIP.COLORS,
  strip.text.colors = PLOT.SBS.STRIP.TEXT.COLORS,
  strip.text.size = 12,
  legend.title = "Mutation Type",
  legend.title.size = 12,
  legend.text.size = 10,
  legend.labels = PLOT.SBS.LEGEND.LABELS
)
```

Arguments

`x.axis.labels`	A character vector used to label the x-axis.
`x.axis.text.size`	x-axis text size.
`x.axis.text.angle`	x-axis text angle.
`y.axis.text.size`	y-axis text size.
`y.axis.title.size`	y-axis title size.
`y.axis.min`	y-axis lower-bound (minimum range for limits).
`y.axis.max`	y-axis upper-bound (maximum range for limits).
`y.axis.title`	y-axis title text.
`groups`	A character vector of group variable values.
`strip.labels`	A character vector used to label the strip.
`strip.colors`	A character vector of hex values used to color the strip.

`strip.text.colors`
: A character vector of hex values used to color the strip text.

`strip.text.size`
: Strip (facet) text size.

`legend.title`
: Legend title text.

`legend.title.size`
: Legend title text size.

`legend.labels`
: A character vector used to label the legend items.

Value

A data.frame of the SBS signature plot theme elements.

`IdentifyDbsSignatures` *Identify doublet base subsitution mutational signatures*

Description

This function identifies DBS mutational signatures.

Usage

```
IdentifyDbsSignatures(
  input,
  bsg,
  sample.id = "Sample",
  reference = GetPcawgDbsSignaturesData(),
  target.signatures = GetPcawgDbsSignaturesNames(),
  plot.theme = GetDbsSignaturesPlotTheme(),
  analyze.variants.column.gene,
  analyze.variants.column.group,
  analyze.variants = FALSE,
  n.cores = 2,
  combn.m = 3,
  n.max.signatures = 7,
  min.probability = 0.01,
  zeta.value = 1e-10,
  save = TRUE,
  save.dir = NULL
)
```

Arguments

`input`
: Either VCF file path or data.frame. If the input is a data.frame, it must include the following columns: Chr, Pos, Ref, Alt.

`bsg`
: BSgenome object.

`sample.id`
: Sample ID that will be used to name output files (default: 'Sample').

`reference`
: A data.frame with the following columns: Mutation_Type, and names of DBS signatures (default: a data.frame returned from `GetPcawgDbsSignaturesData`).

target.signatures
: Signatures to be considered for identification (default: an array returned from GetPcawgDbsSignaturesNames).

plot.theme
: A data.frame returned from GetDbsSignaturesPlotTheme.

analyze.variants.column.gene
: Name of column in the data.frame corresponding to the gene name or ID (e.g. "Gene.refGene" if using ANNOVAR data).

analyze.variants.column.group
: Name of column in the data.frame corresponding to the variant group for plotting purposes (e.g. "Func.refGene" if using ANNOVAR data).

n.cores
: Number of cores to use.

combn.m
: Number of signatures to consider in each step. 'm' parameter in combn function (default: 3).

n.max.signatures
: Maximum number of signatures to identify. Recommended: n.max.signatures >= initial.exploration.combn.m (default: 7).

min.probability
: Minimum probability to attribute to a signature (default: 0.01).

zeta.value
: A float value that is added to the data frequency (default: 1e-10).

save
: Save resulting files if TRUE, otherwise do not save (default: TRUE).

save.dir
: Save directory path (default: NULL).

Value

A list with the following elements:

results: a data.frame with the following columns:

Mutations_Count
: Number of mutations.

Signatures
: Identified mutational signatures separated by comma.

Signatures_Count
: Number of identified mutational signatures.

Signatures_Weights
: Normalized (0 to 1) weights of identified mutational signatures separated by comma.

Mutation_Types
: Mutation types separated by comma.

Mutation_Types_Groups
: Mutation type groups separated by comma.

Observed_Spectrum
: Normalized spectrum (frequency) of observed mutations separated by comma.

Reconstructed_Spectrum
: Normalized spectrum (frequency) of MLE reconstructed mutations separated by comma.

Residual_Spectrum
: Normalized spectrum (frequency) of residual mutations separated by comma.

Cosine_Similarity
: Cosine similarity score between Observed_Spectrum and Reconstructed_Spectrum.

RSS
: Raw residual sum of squares (derived from Residual_Spectrum).

RSS_Normalized	Normalized residual sum of squares (derived from Residual_Spectrum).
BIC	Bayesian information criterion of the identified model.

results.variants: a data.frame with the following columns:

`IdentifyIndelSignatures` *Identify indel mutational signatures*

Description

This function identifies indel mutational signatures.

Usage

```
IdentifyIndelSignatures(
  input,
  bsg,
  sample.id = "Sample",
  reference = GetPcawgIndelSignaturesData(version = "SigProfiler"),
  target.signatures = GetPcawgIndelSignaturesNames(version = "SigProfiler"),
  plot.theme = GetIndelSignaturesPlotTheme(),
  analyze.variants.column.gene,
  analyze.variants.column.group,
  analyze.variants = FALSE,
  n.cores = 2,
  max.len = 25,
  padding.len = 80,
  combn.m = 3,
  n.max.signatures = 7,
  min.probability = 0.01,
  zeta.value = 1e-10,
  save = TRUE,
  save.dir = NULL
)
```

Arguments

input	Either VCF file path or data.frame. If the input is a data.frame, it must include the following columns: Chr, Pos, Ref, Alt.
bsg	BSgenome object.
sample.id	Sample ID that will be used to name output files (default: 'Sample').
reference	A data.frame with the following columns: Mutation_Type, and names of indel signatures (default: a data.fram returned from `GetPcawgIndelSignaturesData(version = "SigProfiler")`).
target.signatures	Signatures to be considered for identification (default: an array returned from `GetPcawgIndelSignaturesNames(version = "SigProfiler")`).
plot.theme	A data.frame returned from `GetIndelSignaturesPlotTheme`.

`analyze.variants.column.gene`
: Name of column in the data.frame corresponding to the gene name or ID (e.g. "Gene.refGene" if using ANNOVAR data).

`analyze.variants.column.group`
: Name of column in the data.frame corresponding to the variant group for plotting purposes (e.g. "Func.refGene" if using ANNOVAR data).

`analyze.variants`
: A boolean value that indicates whether variant-level signature analysis should be performed (default: FALSE).

`n.cores`
: Number of cores to use.

`max.len`
: Maximum number of bases allowed for a small insertion and deletion (indels bigger than this will be excluded; default: 25).

`padding.len`
: Number of bases to use for upstream and downstream sequences (default: 80).

`combn.m`
: Number of signatures to consider in each step. 'm' parameter in combn function (default: 3).

`n.max.signatures`
: Maximum number of signatures to identify. Recommended: n.max.signatures >= initial.exploration.combn.m (default: 7).

`min.probability`
: Minimum probability to attribute to a signature (default: 0.01).

`zeta.value`
: A float value that is added to the data frequency (default: 1e-10).

`save`
: Save resulting files if TRUE, otherwise do not save (default: TRUE).

`save.dir`
: Save directory path (default: NULL).

Value

A list with the following elements:

results: a data.frame with the following columns:

`Mutations_Count`
: Number of mutations.

`Signatures`
: Identified mutational signatures separated by comma.

`Signatures_Count`
: Number of identified mutational signatures.

`Signatures_Weights`
: Normalized (0 to 1) weights of identified mutational signatures separated by comma.

`Mutation_Types`
: Mutation types separated by comma.

`Mutation_Types_Groups`
: Mutation type groups separated by comma.

`Observed_Spectrum`
: Normalized spectrum (frequency) of observed mutations separated by comma.

`Reconstructed_Spectrum`
: Normalized spectrum (frequency) of MLE reconstructed mutations separated by comma.

`Residual_Spectrum`
: Normalized spectrum (frequency) of residual mutations separated by comma.

`Cosine_Similarity`
: Cosine similarity score between Observed_Spectrum and Reconstructed_Spectrum.

RSS	Raw residual sum of squares (derived from Residual_Spectrum).
RSS_Normalized	Normalized residual sum of squares (derived from Residual_Spectrum).
BIC	Bayesian information criterion of the identified model.

results.variants: a data.frame with the following columns:

`IdentifySbsSignatures` *Identify single-base substitution mutational signatures*

Description

This function identifies SBS mutational signatures.

Usage

```
IdentifySbsSignatures(
  input,
  bsg,
  sample.id = "Sample",
  reference = GetPcawgSbsSignaturesData(sequencing.type = "WGS"),
  target.signatures = GetPcawgSbsSignaturesNames(sequencing.type = "WGS"),
  plot.theme = GetSbsSignaturesPlotTheme(),
  analyze.variants.column.gene,
  analyze.variants.column.group,
  analyze.variants = FALSE,
  context.length = 3,
  n.cores = 2,
  combn.m = 3,
  n.max.signatures = 7,
  min.probability = 0.01,
  zeta.value = 1e-10,
  save = TRUE,
  save.dir = NULL
)
```

Arguments

input	Either VCF file path or data.frame. If the input is a data.frame, it must include the following columns: Chr, Pos, Ref, Alt.
bsg	BSgenome object.
sample.id	Sample ID that will be used to name output files (default: 'Sample').
reference	A data.frame with the following columns: Mutation_Type, and names of SBS signatures (default: a data.frame returned from `GetPcawgSbsSignaturesData(sequencing.type = "WGS")`).
target.signatures	Signatures to be considered for identification (default: an array returned from .{GetPcawgSbsSignaturesNames(sequencing.type = "WGS")).

Value

A list with the following elements:

results: a data.frame with the following columns:

`Mutations_Count`
: Number of mutations.

`Signatures`
: Identified mutational signatures separated by comma.

`Signatures_Count`
: Number of identified mutational signatures.

`Signatures_Weights`
: Normalized (0 to 1) weights of identified mutational signatures separated by comma.

`Mutation_Types`
: Mutation types separated by comma.

`Mutation_Types_Groups`
: Mutation type groups separated by comma.

`Observed_Spectrum`
: Normalized spectrum (frequency) of observed mutations separated by comma.

`Reconstructed_Spectrum`
: Normalized spectrum (frequency) of MLE reconstructed mutations separated by comma.

`Residual_Spectrum`
: Normalized spectrum (frequency) of residual mutations separated by comma.

`Cosine_Similarity`
: Cosine similarity score between Observed_Spectrum and Reconstructed_Spectrum.

`RSS`
: Raw residual sum of squares (derived from Residual_Spectrum).

`RSS_Normalized`
: Normalized residual sum of squares (derived from Residual_Spectrum).

`BIC`
: Bayesian information criterion of the identified model.

results.variants: a data.frame with the following columns:

IdentifySignatures *Identify contribution weights of underlying mutational signatures*

Description

This function identifies which mutational signatures are underlying a given data.frame and contribution weights of each mutational signature.

Usage

```
IdentifySignatures(
  data,
  reference,
  target.signatures,
  n.cores = 2,
  combn.m = 3,
  n.max.signatures = 7,
  min.probability = 0.01,
  zeta.value = 1e-10
)
```

Arguments

data	A data.frame prepared by either PrepareSbsVcfFile, PrepareDbsVcfFile, PrepareIdVcfFile, or PrepareSvVcfFile depending on signature.type.
reference	A data.frame where the first column is the list of mutation types and where 2nd to k-th column headers are the names of the signatures.
target.signatures	A character vector that lists all desired signatures to consider for identification.
n.cores	Number of cores to use (default: 2).
combn.m	An integer value that specifies the number of signatures to consider in each step. 'm' parameter in combn function.
n.max.signatures	An integer value that specifies the maximum number of signatures to consider. It is recommended that n.max.signatures >= initial.exploration.combn.m.
min.probability	A float value that specifies the minimum probability to attribute to a signature.
zeta.value	A float value that is added to the data frequency.

Value

A data.frame with the following columns:

Mutations_Count	Number of mutations.
Signatures	Identified mutational signatures separated by comma.
Signatures_Count	Number of identified mutational signatures.
Signatures_Weights	Normalized (0 to 1) weights of identified mutational signatures separated by comma.
Mutation_Types	Mutation types separated by comma.
Mutation_Types_Groups	Mutation type groups separated by comma.
Observed_Spectrum	Normalized spectrum (frequency) of observed mutations separated by comma.
Reconstructed_Spectrum	Normalized spectrum (frequency) of MLE reconstructed mutations separated by comma.
Residual_Spectrum	Normalized spectrum (frequency) of residual mutations separated by comma.
Cosine_Similarity	Cosine similarity score between Observed_Spectrum and Reconstructed_Spectrum.
RSS	Raw residual sum of squares (derived from Residual_Spectrum).
RSS_Normalized	Normalized residual sum of squares (derived from Residual_Spectrum).
BIC	Bayesian information criterion of the identified model.

IdentifyVariantSignatures

Identifies mutational signatures associated with each variant

Description

This function identifies mutational signatures associated with each variant.

Usage

```
IdentifyVariantSignatures(prepared.data, identified.model, reference)
```

Arguments

prepared.data	A data.frame returned from PrepareSbsDataFrame.
identified.model	A data.frame returned from IdentifySignatures.
reference	A data.frame with the following columns: Mutation_Type, and names of SBS signatures.

Value

A data.frame with the following columns:

Chr	Chromosome.
Pos	Genomic position.
Ref	Reference allele.
Alt	Alternate allele.
Mutation_Type	Mutation type.
Variant_Gene	Variant gene.
Variant_Group	Variant group.
Signature	Signature.
Probability	Probability the mutation is the result of the signature.

LinearRegressionOptimFn

Computes and returns the residual sum of squares

Description

This function estimates the residual sum of squares.

Usage

```
LinearRegressionOptimFn(par, mutation.frequencies, signature.weights)
```

Arguments

`par`	A numeric vector of parameters to optimze.
`mutation.frequencies`	A numeric vector corresponding to the frequency of each mutation type.
`signature.weights`	A matrix of signature weights.

Value

Residual sum of squares.

`PlotIdentifiedModel` *Plots identified model*

Description

This function plots an identified model.

Usage

```
PlotIdentifiedModel(identified.model, reference, plot.theme)
```

Arguments

`identified.model`	A row of the data.frame returned from the function IdentifySignatures.
`reference`	A data.frame with the following columns: Mutation_Type, and names of signatures.
`plot.theme`	A data.frame returned from `GetSbsSignaturesPlotTheme`, `GetDbsSignaturesPlotTheme` or `GetIndelSignaturesPlotTheme`.

Value

A list with the following items:

`plot.merged`	Merged (plot.observed, plot.reconstructed.spectrum, plot.residual.spectrum, plot.legend) plot.
`plot.observed`	Observed mutation frequencies plot.
`plot.reconstructed.spectrum`	Reconstructed mutation frequencies plot.
`plot.residual.spectrum`	Residual mutation frequencies plot.
`plot.legend`	Legend plot.

`PlotIdentifiedModels` *Plots identified models*

Description

This function plots identified models for a group of samples IDs.

Usage

```
PlotIdentifiedModels(
  df.models,
  df.signatures.colors,
  signature.group.order = c(),
  sample.ids.order = c(),
  apply.facet = FALSE,
  df.sample.ids.groups = c(),
  sample.ids.groups.order = c(),
  multiply.by.mutations.count = FALSE,
  y.axis.max = NULL,
  plot.title = "",
  title.size = 12,
  x.axis.text.size = 8,
  y.axis.text.size = 10,
  y.axis.title.size = 12,
  legend.title.size = 12,
  legend.text.size = 10
)
```

Arguments

`df.models`	A data.frame of all best models appended by rbind with Sample_ID added as a column.
`df.signatures.colors`	A data.frame returned from `GetSbsSignaturesColors`, `GetDbsSignaturesColors`, or `GetIndelSignaturesColors`.
`signature.group.order`	A character vector specifying the order of the signature groups.
`x.axis.text.size`	A numeric value that sets the x.axis.text size.
`y.axis.text.size`	A numeric value that sets the y.axis.text size.
`y.axis.title.size`	A numeric value that sets the y.axis.title size.
`legend.title.size`	A numeric value that sets the legend title size.
`legend.text.size`	A numeric value that sets the legend text size.

Value

A ggplot object.

PlotIdentifiedModelSignatureWeights

Plots signature weights of an identified model

Description

This function plots signature weights of an identified model.

Usage

```
PlotIdentifiedModelSignatureWeights(
  identified.model,
  df.signatures.colors,
  plot.theme
)
```

Arguments

identified.model	A row of the data.frame returned from the function IdentifySignatures.
df.signature.colors	A data.frame with the following columns: Mutation_Type, and names of signatures.

Value

ggplot object.

PlotIdentifiedModelVariantSignatures

Plots a stacked barplot of signatures probabilities of variants

Description

This function plots a stacked barplot of signature probabilities of variants.

Usage

```
PlotIdentifiedModelVariantSignatures(df.sigs.probs, df.signatures.colors)
```

Value

A list with the following items:

plots.merged	Merged rainfall plots.
plots	Rainfall plot for each chromosome.

PlotSpectrum *Plots spectrum*

Description

This function plots spectrum of mutation frequencies.

Usage

```
PlotSpectrum(y, plot.theme)
```

Arguments

y	A numeric vector corresponding to the y-axis values.
plot.theme	Plot theme (returned from GetSbsSignaturesPlotTheme).

Value

A list with the following items:

plot	plot.
legend	legend plot.

PrepareAnnovarFile *Fetches the closest Gene.refGene from an ANNOVAR file*

Description

This function prepares an ANNOVAR file by fetching the closest Gene.refGene from an ANNOVAR file. If the genomic distance information cannot be parsed, the original 'Gene.refGene' value is returned.

Usage

```
PrepareAnnovarFile(annovar.file)
```

Arguments

annovar.file	ANNOVAR file including path.

Value

A data.frame with the following items:

Chr	Chromosome name.
Pos	Genomic position.
Ref	Nucleotide(s).
Alt	Nucleotide(s).
Gene.refGene	Closest Gene.refGene.
Func.refGene	Functional consequence of variant.

PrepareDbsDataFrame *Prepares DBS data.frame*

Description

This function prepares a DBS data.frame and returns relevant data to run IdentifySignatures function.

Usage

```
PrepareDbsDataFrame(df, bsg, analyze.variants)
```

Arguments

df	A data.frame with the following columns: Chr, Pos, Ref, Alt.
bsg	BSgenome object.
analyze.variants	A boolean value that indicates whether variant-level signature analysis should be performed.

Value

A list with the following items:

df.dbs	A data.frame of DBS data.
df.dbs.frequencies	A data.frame of DBS mutation tyep frequencies.

PrepareDbsVcfFile *Prepares DBS VCF file*

Description

This function prepares a DBS VCF file and returns relevant data to run IdentifySignatures function.

Usage

```
PrepareDbsVcfFile(vcf.file, bsg)
```

Arguments

vcf.file	VCF file including path.
bsg	BSgenome object.

Value

A list with the following items:

df.dbs	A data.frame of DBS data.
df.dbs.frequencies	A data.frame of DBS mutation type frequencies.

PrepareIndelDataFrame *Prepares indel data.frame*

Description

This function prepares an indel data.frame and returns relevant data to run IdentifySignatures function.

Usage

```
PrepareIndelDataFrame(
  df,
  bsg,
  analyze.variants,
  max.len = 25,
  padding.len = 80
)
```

Arguments

df	A data.frame with the following columns: Chr, Pos, Ref, Alt.
bsg	BSgenome object.
analyze.variants	A boolean value that indicates whether variant-level signature analysis should be performed.
max.len	Maximum length of an indel to include. Indels longer than this length will be excluded.
padding.len	Number of bases to pad upstream and downstream of each indel.

Value

A list with the following items:

df.indel	A data.frame of the indel data.
df.indel.frequencies	A data.frame of the PCAWG indel class frequencies.

PrepareIndelVcfFile *Prepares indel VCF file*

Description

This function prepares an indel VCF file and returns relevant data to run IdentifySignatures function.

Usage

```
PrepareIndelVcfFile(vcf.file, bsg, max.len = 25, padding.len = 80)
```

Arguments

vcf.file	VCF file including path.
bsg	BSgenome object.
max.len	Maximum length of an indel to include. Indels longer than this length will be excluded.
padding.len	Number of bases to pad upstream and downstream of each indel.

Value

A list with the following items:

df.indel	A data.frame of the indel data.
df.indel.frequencies	A data.frame of the PCAWG indel class frequencies.

PrepareSbsDataFrame *Prepares SBS data.frame*

Description

This function prepares a SBS data.frame and returns relevant data to run IdentifySignatures function.

Usage

```
PrepareSbsDataFrame(df, bsg, reference, context.length, analyze.variants)
```

Arguments

df	A data.frame with the following columns: Chr, Pos, Ref, Alt.
bsg	BSgenome object.
reference	A data.frame with the following columns: Mutation_Type, and names of SBS signatures (default: a data.frame returned from GetPcawgSbsSignaturesData).
context.length	Number of context nucleotides.
analyze.variants	A boolean value that indicates whether variant-level signature analysis should be performed (default: FALSE).

Value

A list with the following items:

df.sbs	A data.frame of SBS data.
df.sbs.frequencies	A data.frame of SBS mutation tyep frequencies.

PrepareSbsVcfFile *Prepares SBS VCF file*

Description

This function prepares a SBS VCF file and returns relevant data to run IdentifySignatures function.

Usage

```
PrepareSbsVcfFile(vcf.file, bsg, reference, context.length)
```

Arguments

vcf.file	VCF file including path.
bsg	BSgenome object.
reference	A data.frame with the following columns: Mutation_Type, and names of SBS signatures (default: a data.frame returned from GetPcawgSbsSignaturesData).
context.length	Number of context nucleotides.

Value

A list with the following items:

df.sbs	A data.frame of SBS data.
df.sbs.frequencies	A data.frame of SBS mutation type frequencies.

PrepareVcfFile *Prepares a VCF file*

Description

This function prepares a VCF file.

Usage

```
PrepareVcfFile(vcf.file)
```

Arguments

vcf.file	VCF file including path.

Value

A data.frame with the following items:

Chr	Chromosome name.
Pos	Genomic position.
Ref	Genomic position.
Alt	Genomic position.

PreprocessDbs *Preprocess doublet base substitutions*

Description

This function preprocesses a data.frame of DBS.

Usage

```
PreprocessDbs(df, bsg, analyze.variants)
```

Arguments

df	A data.frame with the following columns: Chr, Pos, Ref, Alt.
bsg	BSgenome object.
analyze.variants	A boolean value that indicates whether variant-level signature analysis should be performed.

Value

A data.frame with the following columns:

Chr	Chromosome name.
Pos	Genomic position.
Ref	Reference nucleotide sequence.
Alt	Alternate nucleotide sequence.
Mutation_Type	One of the 96 mutation sub types (e.g. AC>CA).

PreprocessIndels *Preprocess indels*

Description

This function preprocesses a data.frame of indels.

Usage

```
PreprocessIndels(df, bsg, padding.len, analyze.variants)
```

Arguments

df	A data.frame with the following columns: Chr, Pos, Ref, Alt.
bsg	BSgenome object.
padding.len	Number of bases to fetch upstream and downstream of each indel.
analyze.variants	A boolean value that indicates whether variant-level signature analysis should be performed.

Value

A data.frame with the following columns:

`Chr`	Chromosome name.
`Pos`	Genomic position.
`Ref`	Reference nucleotide sequence.
`Alt`	Alternate nucleotide sequence.
`Type`	Either 'INS' (insertion) or 'DEL' (deletion).
`Len`	Length (number of bases) of this indel.
`Mutated_Seq`	Nucleotide sequence of this indel.
`Upstream_Seq`	Upstream nucleotide sequence of this indel.
`Upstream_Start`	Upstream nucleotide sequence start position.
`Upstream_End`	Upstream nucleotide sequence end position.
`Downstream_Seq`	Downstream nucleotide sequence of this indel.
`Downstream_Start`	Downstream nucleotide sequence start position..
`Downstream_End`	Downstream nucleotide sequence end position.

`PreprocessSbs` *Preprocess single base substitutions*

Description

This function preprocesses a data.frame of SBS.

Usage

```
PreprocessSbs(df, bsg, context.length, analyze.variants)
```

Arguments

`df`	A data.frame with the following columns: Chr, Pos, Ref, Alt.
`bsg`	BSgenome object.
`context.length`	Number of context nucleotides.
`analyze.variants`	A boolean value that indicates whether variant-level signature analysis should be performed.

Value

A data.frame with the following columns:

`Chr`	Chromosome name.
`Pos`	Genomic position.
`Ref`	Reference nucleotide sequence.
`Alt`	Alternate nucleotide sequence.

Type	One of =the six mutation types (e.g. C>A).
Sub_Type	One of the 32 trinucleotide subtypes (e.g. ACA).
Somatic_Mutation_Type	One of the 96 mutation sub types (e.g. A[C>A]A).
Upstream_Seq	Upstream nucleotide sequence of this SBS.
Downstream_Seq	Downstream nucleotide sequence of this SBS.

`PrintLog` *Prints log*

Description

This function prints a log message.

Usage

```
PrintLog(message, type = "INFO")
```

Arguments

message	String value message to print along with log type and date.
type	String value that represents type of this message. 'INFO' by default.

`SortDataFrameForStackedBarPlot` *Sorts a data.frame before plotting a stacked barplot*

Description

This function sorts a data.frame before plotting a stacked barplot.

Usage

```
SortDataFrameForStackedBarPlot(df, ordered.features, x.axis.var)
```

Arguments

df	A data.frame.
ordered.features	An ordered character vector of columns to sort in df.
x.axis.var	String value for x axis variable.

Value

A sorted data.frame.

UnwrapIdentifiedModel *Unwraps an identified model*

Description

This function returns an unwrapped list of an identified model.

Usage

```
UnwrapIdentifiedModel(identified.model)
```

Arguments

identified.model	A data.frame returned from IdentifySbsSignatures, IdentifyDbsSignatures, or IdentifyIndelSignatures.

Value

A data.frame.

WrapIdentifiedModel *Wraps an identified model*

Description

This function returns a (wrapped) data.frame of an identified model

Usage

```
WrapIdentifiedModel(
  mutations.count,
  signatures,
  signatures.weights,
  mutation.types,
  mutation.types.groups,
  observed.spectrum,
  reconstructed.spectrum,
  residual.spectrum,
  cosine.similarity,
  rss,
  rss.normalized,
  bic
)
```

Arguments

`mutations.count`	Mutations count.
`signatures`	A character vector of signature names.
`signatures.weights`	A numeric vector of signature weights.
`mutation.types`	A character vector of mutation types.
`mutation.types.groups`	A character vector of mutation type groups.
`observed.spectrum`	A numeric vector of observed spectrum.
`reconstructed.spectrum`	A numeric vector of reconstructed spectrum.
`residual.spectrum`	A numeric vector of residual spectrum.
`cosine.similarity`	Cosine similiarity score.
`rss`	Residual sum of squares.
`rss.normalized`	Normalized residual sum of squares.
`bic`	Bayesian information criterion.

Value

A data.frame with the following columns:

Index