

Package ‘FIREVAT’

February 19, 2019

Type Package

Title FIREVAT, FInding REliable Variants without ArTifacts

Description FIREVAT is a variant filtering tool for cancer sequencing data, which uses mutational signatures to identify sequencing artifacts and low-quality variants.

Version 0.1.0

Authors Andy Jinseok Lee, Hyunbin Kim

Maintainer Andy Jinseok Lee <jinseok.lee@ncc.re.kr>, Hyunbin Kim <khb7840@ncc.re.kr>

Imports data.table,
stringi,
bedr,
GA,
jsonlite,
yaml,
MutationalPatterns,
deconstructSigs,
lsa,
BSgenome.Hsapiens.UCSC.hg19,
BSgenome.Hsapiens.UCSC.hg38,
ggpubr,
caTools,
ggrepel,
gridExtra,
ggplot2,
rmarkdown,
gtable

URL <https://github.com/cgab-ncc/FIREVAT>

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Suggests knitr

VignetteBuilder knitr

R topics documented:

ComputeZScore	3
ComputeZScoreEquiValue	3
DecimalCeiling	4
EnumerateTriNucCounts	4
FilterVCFFromConfig	5
GetCOSMICMutSigs	5
GetCOSMICMutSigsEtiologiesColors	6
GetCOSMICMutSigsNames	6
GetOptimizedSignatures	6
GetPCAWGMutSigs	7
GetPCAWGMutSigsEtiologiesColors	7
GetPCAWGMutSigsNames	8
InitializeVCFFromConfig	8
MakeFilterFromConfig	9
MutaliskParseVCFObj	9
ParseConfigFile	10
PCAWG.All.Sequencing.Artifact.Signatures	10
PCAWG.Known.Sequencing.Artifact.Signatures	11
PCAWG.Likely.Sequencing.Artifact.Signatures	11
PCAWG.Possible.Sequencing.Artifact.Signatures	11
PCAWG.Target.Mutational.Signatures	12
PlotMutaliskResults	12
PlotMutationTypes	13
PlotOptimizationIterations	14
PlotSigsProbs	15
PlotTable	16
PlotTriNucSpectrum	16
PlotVCFStatsBoxPlots	17
PlotVCFStatsHistograms	18
PrepareReportDatafromConfig	19
PrepareReportPlotsfromConfig	20
ReadOptimizationIterationReport	21
ReadVCF	21
ReportFIREVATResults	22
RunFIREVAT	22
RunMutalisk	23
RunMutaliskHelper	24
TriNuc.Mutation.Type.Hex.Colors	25
UpdateFilterFromConfig	26
WriteVCF	26

ComputeZScore	<i>ComputeZScore</i>
---------------	----------------------

Description

Returns a z-score of x given a distribution of values

Usage

ComputeZScore(values, x)

Arguments

- | | |
|--------|------------------|
| values | a numeric vector |
| x | a numeric value |

Value

a numeric value corresponding to the z-score of x

ComputeZScoreEquiValue	<i>ComputeZScoreEquiValue</i>
------------------------	-------------------------------

Description

Returns a numeric value that is equivalent to the specified z.score in the distribution of 'values'

Usage

ComputeZScoreEquiValue(z.score, values)

Arguments

- | | |
|---------|----------------|
| z.score | numeric value |
| values | numeric vector |

Value

a numeric value corresponding to the specified z.score in the 'values' distribution

DecimalCeiling	<i>DecimalCeiling</i>
----------------	-----------------------

Description

Returns the ceiling of a decimal value e.g. value = 0.15, decimal = 0.1 returns 0.2

Usage

```
DecimalCeiling(value, decimal)
```

Arguments

value	numeric value (decimal)
decimal	numeric value (e.g. 0.1, 0.001)

Value

a numeric value

EnumerateTriNucCounts	<i>EnumerateTriNucCounts</i>
-----------------------	------------------------------

Description

Returns C>A, C>G, C>T, T>A, T>C, T>G counts

Usage

```
EnumerateTriNucCounts(spectrum)
```

Arguments

spectrum	a numeric vector with 96 numeric values
----------	---

Details

Please note that this function assumes that 'spectrum' is sorted (i.e. 1:16 → C>A; 17:32 → C>G; 33:48 → C>T; 49:64 → T>A; 65:80 → T>C; 81:96 → T>G)

Value

a numeric vector of length 6 corresponding to the counts of each trinucleotide change (C>A, C>G, C>T, T>A, T>C, T>G)

FilterVCFFromConfig	<i>FilterVCFFromConfig</i>
---------------------	----------------------------

Description

Filter vcf based on the filter Filtering parameters are saved in config.obj Split vcf.obj into vcf.obj.filtered & vcf.obj.artifact based on vcf.filter

Usage

```
FilterVCFFromConfig(vcf.obj, vcf.filter, config.obj, verbose = FALSE)
```

Arguments

vcf.obj	A list from ReadVCF
vcf.filter	A list from MakeMuTect2Filter
config.obj	A list from ParseConfigFile
verbose	If true, provides process detail

Value

A list with the following elements

- 1) Mutations which passed filteringvcf.obj.filtered = vcf.obj (list with data, header, genome)
- 2) Mutations which did not pass filteringvcf.obj.artifact = vcf.obj (list with data, header, genome)

GetCOSMICMutSigs	<i>GetCOSMICMutSigs</i>
------------------	-------------------------

Description

Returns a data.frame of the COSMIC mutational signature reference file from the data directory

Usage

```
GetCOSMICMutSigs()
```

Value

a data.frame of the COSMIC reference mutational signatures

```
GetCOSMICMutSigsEtiologiesColors
  GetCOSMICMutSigsNames
```

Description

Returns all COSMIC mutational signature etiologies and colors

Usage

```
GetCOSMICMutSigsEtiologiesColors()
```

Value

data.frame with following columns: signature, group and color.

```
GetCOSMICMutSigsNames  GetCOSMICMutSigsNames
```

Description

Returns all COSMIC mutational signature names

Usage

```
GetCOSMICMutSigsNames()
```

Value

a character vector

```
GetOptimizedSignatures
  GetOptimizedSignatures
```

Description

This function fetches the last row from the optimization iteration log and returns the target and artifactual mutational signatures for the type of mutations ('refined' or 'artifactual')

Usage

```
GetOptimizedSignatures(data, mutations.type = "refined",
  signatures = "all")
```

Arguments

- data A list of main data from [RunFIREVAT](#)
- mutations.type A string for type of mutations ('refined' or 'artifact')
- signatures A string ('all', 'target', 'artifact')

Value

A data.frame with the columns 'signature' and 'weight'

GetPCAWGMutSigs	<i>GetPCAWGMutSigs</i>
-----------------	------------------------

Description

Returns the PCAWG mutational signatures data

Usage

GetPCAWGMutSigs()

Value

a data.frame of the PCAWG mutatioanl signatures

GetPCAWGMutSigsEtiologiesColors	<i>GetPCAWGMutSigsEtiologiesColors</i>
---------------------------------	--

Description

Returns the PCAWG mutational signatures etiologies and colors

Usage

GetPCAWGMutSigsEtiologiesColors()

Value

a data.frame with the columns 'signature', 'group', 'color'

GetPCAWGMutSigsNames	<i>GetPCAWGMutSigsNames</i>
----------------------	-----------------------------

Description

Returns the PCAWG mutational signatures names

Usage

GetPCAWGMutSigsNames()

Value

a character vector of the PCAWG mutational signatures names

InitializeVCFFromConfig	<i>InitializeVCFFromConfig</i>
-------------------------	--------------------------------

Description

Initialize VCF with FIREVAT config file This functions selects point mutations and appends filter values to vcf.obj\$data

Usage

InitializeVCFFromConfig(vcf.obj, config.obj, verbose = TRUE)

Arguments

vcf.obj	A list from ReadVCF
config.obj	A list from ParseConfigFile
verbose	If true, provides process detail

Value

A list with the following elements

- vcf.obj.filteredvcf.obj (high-quality vcf)
- vcf.obj.artifactvcf.obj (low-quality vcf)

MakeFilterFromConfig	<i>MakeFilterFromConfig</i>
----------------------	-----------------------------

Description

Creates a vcf filter from config.obj

Usage

```
MakeFilterFromConfig(config.obj)
```

Arguments

config.obj	A list from ParseConfigFile (any filter with "use_in_filter" value declared as FALSE is not considered)
------------	---

Value

A list with the filter parameters

MutaliskParseVCFObj	<i>MutaliskParseVCFObj</i>
---------------------	----------------------------

Description

Parses a vcf.obj and prepares it to run Mutalisk.

Usage

```
MutaliskParseVCFObj(vcf.obj)
```

Arguments

vcf.obj	A list from ReadVCF
---------	---------------------

Value

A data.frame

ParseConfigFile

ParseConfigFile

Description

This function returns config.obj from JSON or YAML config file. - Check if the config file is in JSON format or YAML format - Return config.obj

Usage

```
ParseConfigFile(config.path, verbose = TRUE)
```

Arguments

config.path	A string for config file path
verbose	If true, provides process detail

Value

config.obj: list of parameters

Examples

```
## Not run:  
ParseConfigFile("example.variant.caller.json")  
ParseConfigFile("example.variant.caller.json", verbose=False)  
  
## End(Not run)
```

PCAWG.All.Sequencing.Artifact.Signatures*Constant*

Description

PCAWG mutational signatures reported to be associated with sequencing artifacts

Usage

```
PCAWG.All.Sequencing.Artifact.Signatures
```

Format

An object of class character of length 17.

PCAWG.Known.Sequencing.Artifact.Signatures
Constant

Description

PCAWG mutational signatures reported to be associated with sequencing artifacts

Usage

PCAWG.Known.Sequencing.Artifact.Signatures

Format

An object of class character of length 1.

PCAWG.Likely.Sequencing.Artifact.Signatures
Constant

Description

PCAWG mutational signatures reported to be associated with sequencing artifacts

Usage

PCAWG.Likely.Sequencing.Artifact.Signatures

Format

An object of class character of length 5.

PCAWG.Possible.Sequencing.Artifact.Signatures
Constant

Description

PCAWG mutational signatures reported to be associated with sequencing artifacts

Usage

PCAWG.Possible.Sequencing.Artifact.Signatures

Format

An object of class character of length 11.

PCAWG.Target.Mutational.Signatures
<i>Constant</i>

Description

PCAWG target mutational signatures reported to be unrelated to sequencing artifacts

Usage

```
PCAWG.Target.Mutational.Signatures
```

Format

An object of class character of length 49.

PlotMutaliskResults	<i>PlotMutaliskResults</i>
---------------------	----------------------------

Description

Plots Mutalisk results

Usage

```
PlotMutaliskResults(mutalisk.results, signatures, trinuc.max.y,  
  trinuc.min.y, mut.type.max.y, title)
```

Arguments

- mutalisk.results A list obtained from [RunMutalisk](#)
- signatures A character vector of mutational signatures names
- trinuc.max.y A numeric value (maximum y-axis value)
- trinuc.min.y A numeric value (minimum y-axis value)
- mut.type.max.y A numeric value
- title A string value

Value

A ggplot object

Examples

```
## Not run:
df.ref.mut.sigs <- GetPCAWGMutSigs()
target.mut.sigs <- GetPCAWGMutSigsNames()
vcf.obj <- ReadVCF(vcf.file = "../data/sample/P-233-CT.final.vcf")
mutalisk.results <- RunMusalisk(vcf.obj = vcf.obj,
                               df.ref.mut.sigs = df.ref.mut.sigs,
                               target.mut.sigs = target.mut.sigs)
p <- PlotMusaliskResults(musalisk.results = mutalisk.results)
print(p)

## End(Not run)
```

PlotMutationTypes	<i>PlotMutationTypes</i>
-------------------	--------------------------

Description

Plots a horizontal barplot of mutation types

Usage

```
PlotMutationTypes(mutation.types = c("C>A", "C>G", "C>T", "T>A", "T>C",
  "T>G"), mutation.types.values, mutation.types.colors, max.y.val, title,
  convert.to.percentage = T, show.legend = T, font.size.small = 8,
  font.size.med = 14, plot.margin = unit(c(0.5, 0.5, 0.5, 0.5), "cm"))
```

Arguments

<code>mutation.types</code>	Mutation types; Default = c("C>A", "C>G", "C>T", "T>A", "T>C", "T>G")
<code>mutation.types.values</code>	Mutation count for each mutation type
<code>mutation.types.colors</code>	A color vector for indicating mutation types
<code>max.y.val</code>	y axis maximum value
<code>title</code>	Plot title
<code>convert.to.percentage</code>	if True convert y values to percentage (x 100); Default = T
<code>show.legend</code>	If True, show legend; Default = T
<code>font.size.small</code>	Small font size; Default = 8
<code>font.size.med</code>	Medium font size; Default = 14
<code>plot.margin</code>	Margin vector for drawing plot; Default = unit(c(0.5, 0.5, 0.5, 0.5), "cm"))

Value

A ggplot object

Examples

```
## Not run:
p <- PlotMutationTypes(mutation.types = c("C>A", "C>G", "C>T", "T>A", "T>C", "T>G"),
  mutation.types.values = c(0.3, 0.3, 0.1, 0.1, 0.1, 0.1),
  mutation.types.colors = TriNuc.Mutation.Type.Hex.Colors,
  max.y.val = 0.5,
  convert.to.percentage = T,
  show.legend = T,
  font.size.small = 8,
  font.size.med = 14,
  plot.margin = unit(c(0.5, 0.5, 0.5, 0.5), "cm"))

print(p)

## End(Not run)
```

PlotOptimizationIterations

PlotOptimizationIterations

Description

Plots multiple scatter plots into one figure

Usage

```
PlotOptimizationIterations(df, columns.to.plot, x.axis.var, x.axis.title,
  x.max, save.file, title, y.axis.title = "", y.max = 1,
  point.size = 1, connect.dots = T, plot.legend = T,
  legend.ncol = 1, font.size.med = 14, font.size.large = 16,
  plot.margin = unit(c(0.5, 0.5, 0.5, 0.5), "cm"))
```

Arguments

df	A data.frame (from reading "FIREVAT_Optimization_Logs.tsv")
columns.to.plot	A character vector (of column names to plot)
x.axis.var	x axis variable
x.axis.title	x axis title
x.max	x axis maximum value
save.file	Filename (including full path) to which the plot will be saved
title	Plot title
y.axis.title	y axis title; Default = ""

<code>y.max</code>	y axis maximum value; Default = 1
<code>point.size</code>	Point size; Default = 1
<code>connect.dots</code>	If True draws dots for each iteration; Default = True
<code>plot.legend</code>	If True write legend of plot; Default = T
<code>legend.ncol</code>	legend.n Default = 1
<code>font.size.med</code>	Medium font size; Default = 14
<code>font.size.large</code>	Large font size; Default = 16
<code>plot.margin</code>	Margin vector for plot; Default = <code>unit(c(0.5, 0.5, 0.5, 0.5), "cm")</code>

Value

A ggplot object

PlotSigsProbs	<i>PlotSigsProbs</i>
---------------	----------------------

Description

Plots a horizontal barplot of identified mutational signatures

Usage

```
PlotSigsProbs(df.identified.mut.sigs, df.ref.sigs.groups.colors, title,
  convert.to.percentage = T, font.size.small = 8, font.size.med = 14,
  plot.margin = unit(c(0.5, 0.5, 0.5, 0.5), "cm"))
```

Arguments

<code>df.identified.mut.sigs</code>	A data.frame of identified mutational signatures
<code>df.ref.sigs.groups.colors</code>	A data.frame with 'signature', 'group', and 'color' columns
<code>title</code>	Plot title
<code>convert.to.percentage</code>	If true, convert y values to percentage (x 100); Default = T,
<code>font.size.small</code>	Small font size; Default = 8,
<code>font.size.med</code>	Medium font size; Default = 14,
<code>plot.margin</code>	Margin vector for drawing plot; Default = <code>unit(c(0.5, 0.5, 0.5, 0.5), "cm")</code>

Value

A ggplot object

Examples

```
## Not run:
g <- PlotSigsProbs(sigs = c(mutalisk.results$identified.mut.sigs),
  sigs.probs = c(mutalisk.results$identified.mut.sigs.probs),
  df.ref.sigs.groups.colors = GetPCAWGMutSigsEtiologiesColors())

print(g)

## End(Not run)
```

PlotTable	<i>PlotTable</i>
-----------	------------------

Description

Plots basic statistics table

Usage

```
PlotTable(df, padding = 20, font.size = 14)
```

Arguments

- df = A data.frame where the first column is header and the second column is data value
- padding Padding size; Default = 20
- font.size Font size; Default = 14

Value

A plot

PlotTriNucSpectrum	<i>PlotTriNucSpectrum</i>
--------------------	---------------------------

Description

Plots the spectrum of 96 trinucleotide distribution (C>A, C>G, C>T, T>A, T>C, T>G) Please note that this function assumes that both sub.types and spectrum are sorted in the following order: C>A, C>G, C>T, T>A, T>C, T>G

Usage

```
PlotTriNucSpectrum(sub.types, spectrum, max.y.val, min.y.val, y.axis.title,
  draw.top.strip = T, draw.x.axis.labels = T, draw.y.axis.labels = T,
  draw.y.axis.title = T, font.size.small = 8, font.size.med = 14,
  plot.margin.top = 0.5, plot.margin.bottom = 0.5,
  plot.margin.left = 0.5, plot.margin.right = 0.5, title)
```


Arguments

sub.types	A character vector (types of 96 trinucleotide substitutions)
spectrum	A numeric vector (96 elements)
max.y.val	y axis maximum value
min.y.val	y axis minimum value
y.axis.title	y axis title
draw.top.strip	If True then draws top strip; Default = T
draw.x.axis.labels	If True then draws x axis labels; Default = T
draw.y.axis.labels	If True then draws y axis labels; Default = T
draw.y.axis.title	If True then draws y axis title; Default = T
font.size.small	Small font size; Default = 8
font.size.med	Medium font size; Default = 14
plot.margin.top	Top margin; Default = 0.5
plot.margin.bottom	Bottom margin; Default = 0.5
plot.margin.left	Left margin; Default = 0.5
plot.margin.right	Right margin; Default = 0.5
title	Plot title

Value

A ggplot object

PlotVCFStatsBoxPlots *PlotVCFStatsBoxPlots*

Description

Plots multiple (original, refined, artifact vcf) boxplots for single filter parameter

Usage

```
PlotVCFStatsBoxPlots(original.vcf.stat.values, refined.vcf.stat.values,
  artifact.vcf.stat.values, xlab, axis.font.size = 10,
  label.font.size = 10, title.font.size = 12)
```

Arguments

<code>original.vcf.stat.values</code>	A numeric vector corresponding to the original vcf.obj values of single filter parameter
<code>refined.vcf.stat.values</code>	A numeric vector corresponding to the refined vcf.obj values of single filter parameter
<code>artifact.vcf.stat.values</code>	A numeric vector corresponding to the artifact vcf.obj values of single filter parameter
<code>xlab</code>	A string value (x-axis label)
<code>axis.font.size</code>	An integer value (axis font size)
<code>label.font.size</code>	An integer value (label font size)
<code>title.font.size</code>	An integer value (title font size)

Value

A ggboxplot

PlotVCFStatsHistograms

PlotVCFStatsHistograms

Description

Plots multiple VCF stats histograms into one figure

Usage

```
PlotVCFStatsHistograms(plot.values, x.axis.labels, stat.y.max.vals,
  stat.x.max.vals, sample.id, save.file, title, cutoff.values,
  plot.boxplot = F, plot.cutoff.line.color = "#D4012E",
  plot.cutoff.value.lines = F, bin.width = 1, ncol = 4, nrow = 3,
  font.size.med = 10, font.size.large = 12, plot.margin = unit(c(0.5,
  0.5, 0.5, 0.5), "cm"))
```

Arguments

<code>plot.values</code>	A list of multiple numeric vectors
<code>x.axis.labels</code>	A character vector of x axis labels
<code>stat.y.max.vals</code>	A numeric vector of max y-axis values

stat.x.max.vals	A numeric vector of max x-axis values
sample.id	A string value of sample ID
save.file	A string value of file to which the resulting plot will be saved
title	A string value of plot title
cutoff.values	A numeric vector of cutoff values
plot.boxplot	A boolean value (default = False)
plot.cutoff.line.color	A hex string value (default = "#D4012E")
plot.cutoff.value.lines	A boolean value (default = False)
bin.width	An integer value (default = 1; histogram bin width)
ncol	An integer value (default = 4; ggarrange ncol)
nrow	An integer value (default = 3; ggarrange nrow)
font.size.med	An integer value (default = 10)
font.size.large	An integer value (default = 12)
plot.margin	A list (default = unit(c(0.5, 0.5, 0.5, 0.5), "cm"))

Value

A list with the following elements

- f = A ggarrange object
- graphs = A list of length 3; each element is a ggplot histogram

```
PrepareReportDatafromConfig
```

PrepareReportDatafromConfig

Description

Prepares report data (table data)

Usage

```
PrepareReportDatafromConfig(x.solution.decimal, data)
```

Arguments

x.solution.decimal	A numeric vector
data	A list of main data from RunFIREVAT

Value

A list with following items:

- `df.parameters`A data.frame of FIREVAT Genetic Algorithm (GA) Parameters
- `df.filter.cutoffs`A data.frame of variant filtering cutoffs
- `df.results`A data.frame of objective values
- `df.mutalisk.results`A data.frame of mutational signature analysis results from Mutalisk

```
PrepareReportPlotsfromConfig
```

```
PrepareReportPlotsfromConfig
```

Description

Prepares report plots

Usage

```
PrepareReportPlotsfromConfig(x.solution.decimal, data)
```

Arguments

<code>x.solution.decimal</code>	A numeric vector
<code>data</code>	A list of main data from RunFIREVAT

Value

A list with following items:

- `optimization.iter.plot`A plot for the trend of objective value during iteration
- `refined.muts.seq.art.iter.plot`A plot for showing sequencing artifact weights in refined variants
- `artifactual.muts.seq.art.iter.plot`A plot for showing sequencing artifact types in artifactual variants
- `raw.muts.mutalisk.plots`A plot for mutalisk results of original variants
- `refined.muts.mutalisk.plots`A plot for mutalisk results of refined variants
- `artifactual.muts.mutalisk.plots`A plot for mutalisk results of artifactual variants
- `original.vcf.stats.plots`A histogram for original VCF stats
- `refined.vcf.stats.plots`A histogram for refined VCF stats
- `artifact.vcf.stats.plots`A histogram for artifactual VCF stats
- `vcf.stats.boxplots`A plot for showing comparison results among VCF stats of original, refined and artifactual VCFs

ReadOptimizationIterationReport	
	<i>ReadOptimizationIterationReport</i>

Description

Read optimization iteration report

Usage

ReadOptimizationIterationReport(data)

Arguments

data A list of main data from [RunFIREVAT](#)

Value

A data.frame of FIREVAT optimization logs

ReadVCF	<i>ReadVCF</i>
---------	----------------

Description

Reads a .vcf file

Usage

ReadVCF(vcf.file, genome = "hg19")

Arguments

vcf.file (full path of a .vcf file)
genome ('hg19' or 'hg38')

Value

A list with elements 'data', 'header', 'genome'

ReportFIREVATResults	<i>ReportFIREVATResults</i>
----------------------	-----------------------------

Description

Reports FIREVAT results in html format (generated from Rmd)

Usage

```
ReportFIREVATResults(x.solution.decimal, data, verbose = T,
  report.format = "html")
```

Arguments

x.solution.decimal	A numeric vector
data	A list of main data from RunFIREVAT
verbose	If true, provides process detail
report.format	The format of FIREVAT report. In current version, we only support "html"

RunFIREVAT	<i>RunFIREVAT</i>
------------	-------------------

Description

Runs FIREVAT using configuration data.Filters point mutations in the specified vcf.file based on mutational signature decomposition and outputs the refined and artifact vcf as well as metadata related to the refinement process.

Usage

```
RunFIREVAT(vcf.file, vcf.file.genome, config.file, df.ref.mut.sigs,
  target.mut.sigs, sequencing.artifact.mut.sigs, num.cores, output.dir,
  pop.size = 200, max.iter = 200, run = 50, pmutation = 0.25,
  verbose = TRUE)
```

Arguments

vcf.file	String value corresponding to input .vcf file (full path)
vcf.file.genome	Genome assembly of the input .vcf file
config.file	String value corresponding to input configuration file (refer to ...)
df.ref.mut.sigs	A data.frame of the reference mutational signatures

target.mut.sigs	A character vector of the target mutational signatures (from reference mutational signatures)
sequencing.artifact.mut.sigs	A character vector of the sequencing artifact mutational signatures (from reference mutational signatures)
num.cores	Number of cores to allocate
output.dir	String value of the desired output directory
pop.size	Integer value of the Genetic Algorithm 'population size' parameter
max.iter	Integer value of the Genetic Algorithm 'maximum iterations' parameter
run	Integer value of the Genetic Algorithm 'run' parameter
pmutation	Float value of the Genetic Algorithm 'mutation probability' parameter
verbose	If true, provides process detail

Value

A list with the following elements

- x.solution.decimalA numeric vector of optimized parameter values
- data

RunMutalisk

RunMutalisk

Description

Identifies mutational signatures using Mutalisk

Usage

```
RunMutalisk(vcf.obj, df.ref.mut.sigs, target.mut.sigs,
  random.sampling.candidate.mut.sigs = c(), method = "random.sampling",
  n.sample = 20, n.iter = 10, verbose = T)
```

Arguments

vcf.obj	A list (from firevat_vcf::ReadVCF)
df.ref.mut.sigs	A data.frame of reference mutational signatures
target.mut.sigs	A character vector of target mutational signatures names to identify from
random.sampling.candidate.mut.sigs	A character vector of mutational signatures names that gets appended to the list of candidate mutational signatures so that these are always considered.

method	A string value (must be either 'random.sampling' or 'all'). The method 'random.sampling' samples (without replacement) 'n.sample' number of signatures 'n.iter' number of times and runs the candidate signatures one last time. The method 'all' uses all target.mut.sigs
n.sample	An integer value ('random.sampling' method parameter) Number of signatures to choose for each iteration of random sampling).
n.iter	An integer value ('random.sampling' method parameter). Number of iterations to perform random sampling.
verbose	If true, provides process details

Value

A list with the following elements

- num.point.mutations An integer value - count of total point mutations
- sub.types A character vector of length 96
- sub.types.spectrum A numeric vector of length 96
- num.mut.sigs An integer value (count of unique mutational signatures identified)
- identified.mut.sigs A character vector where each element is a mutational signature identified
- identified.mut.sigs.probs A numeric vector where each element is the weight of mutational signature identified. The ordering follows identified.mut.sigs
- identified.mut.sigs.spectrum A numeric vector of length 96
- residuals A numeric vector of length 96
- rss A numeric value (residual sum of squares)
- cos.sim.score A numeric value (cosine similarity score between observed mutational spectrum and reconstructed mutational signatures)
- all.models.sigs A list where each element is a model; a model is a list of signatures identified
- all.models.sigs.probs A list where each element is a model; a model is a list of contribution probabilities
- all.models.cos.sim.scores A list where each element is a model; a model is a list of cosine similarity socres

RunMutaliskHelper

RunMutaliskHelper

Description

Helper function for RunMutalisk

Usage

```
RunMutaliskHelper(vcf.trinucleotide.data, df.ref.mut.sigs, target.mut.sigs)
```


Arguments

`vcf.trinucleotide.data`
 A data.frame (from `firevat_mutalisk::MutaliskParseVCFObj`)

`df.ref.mut.sigs`
 A data.frame of reference mutational signatures

`target.mut.sigs`
 A character vector of target mutational signatures names

Value

A list with the following elements

- `num.point.mutations` An integer value - count of total point mutations
- `sub.types` A character vector of length 96
- `sub.types.spectrum` A numeric vector of length 96
- `num.mut.sigs` An integer value (count of unique mutational signatures identified)
- `identified.mut.sigs` A character vector where each element is a mutational signature identified
- `identified.mut.sigs.probs` A numeric vector where each element is the weight of mutational signature identified. The ordering follows `identified.mut.sigs`
- `identified.mut.sigs.spectrum` A numeric vector of length 96
- `residuals` A numeric vector of length 96
- `rss` A numeric value (residual sum of squares)
- `cos.sim.score` A numeric value (cosine similarity score between observed mutational spectrum and reconstructed mutational signatures)
- `all.models.sigs` A list where each element is a model; a model is a list of signatures identified
- `all.models.sigs.probs` A list where each element is a model; a model is a list of contribution probabilities
- `all.models.cos.sim.scores` A list where each element is a model; a model is a list of cosine similarity scores

TriNuc.Mutation.Type.Hex.Colors
Constant

Description

Hex codes for the mutation types (for plotting purposes)

Usage

TriNuc.Mutation.Type.Hex.Colors

Format

An object of class character of length 6.

UpdateFilterFromConfig
<i>UpdateFilterFromConfig</i>

Description

Update filter based on optim parameter values

Usage

UpdateFilterFromConfig(vcf.filter, param.values)

Arguments

- vcf.filter A list from MakeFilterFromConfig
- param.values A numeric vector contains filtering value (same length with length(vcf.config.filter))

Value

Updated vcf.filter (list)

WriteVCF	<i>WriteVCF</i>
----------	-----------------

Description

Writes a vcf.obj to a .vcf file

Usage

WriteVCF(vcf.obj, save.file)

Arguments

- vcf.obj (from the function ReadVCF)
- save.file (full path including filename)

Index

*Topic datasets

[PCAWG.All.Sequencing.Artifact.Signatures](#), [11](#)
[10](#) [PCAWG.Possible.Sequencing.Artifact.Signatures](#),
[PCAWG.Known.Sequencing.Artifact.Signatures](#), [11](#)
[11](#) [PCAWG.Target.Mutational.Signatures](#), [12](#)
[PCAWG.Likely.Sequencing.Artifact.Signatures](#), [PlotMuta](#)[liskResults](#), [12](#)
[11](#) [PlotMutationTypes](#), [13](#)
[PCAWG.Possible.Sequencing.Artifact.Signatures](#), [ReadOptimizationIterations](#), [14](#)
[11](#) [PlotSigsProbs](#), [15](#)
[PCAWG.Target.Mutational.Signatures](#), [PlotTable](#), [16](#)
[12](#) [PlotTriNucSpectrum](#), [16](#)
[TriNuc.Mutation.Type.Hex.Colors](#), [PlotVCFStatsBoxPlots](#), [17](#)
[25](#) [PlotVCFStatsHistograms](#), [18](#)
[ComputeZScore](#), [13](#) [PrepareReportDatafromConfig](#), [19](#)
[ComputeZScoreEquiValue](#), [3](#) [PrepareReportPlotsfromConfig](#), [20](#)
[DecimalCeiling](#), [4](#) [ReadOptimizationIterationReport](#), [21](#)
[EnumerateTriNucCounts](#), [4](#) [ReadVCF](#), [21](#)
[FilterVCFFromConfig](#), [5](#) [ReportFIREVATResults](#), [22](#)
[GetCOSMICMutSigs](#), [5](#) [RunFIREVAT](#), [7](#), [19–22](#), [22](#)
[GetCOSMICMutSigsEtiologiesColors](#), [6](#) [RunMuta](#)[lisk](#), [12](#), [23](#)
[GetCOSMICMutSigsNames](#), [6](#) [RunMuta](#)[liskHelper](#), [24](#)
[GetOptimizedSignatures](#), [6](#) [TriNuc.Mutation.Type.Hex.Colors](#), [25](#)
[GetPCAWGMutSigs](#), [7](#) [UpdateFilterFromConfig](#), [26](#)
[GetPCAWGMutSigsEtiologiesColors](#), [7](#) [WriteVCF](#), [26](#)
[GetPCAWGMutSigsNames](#), [8](#)
[InitializeVCFFromConfig](#), [8](#)
[MakeFilterFromConfig](#), [9](#)
[Muta](#)[liskParseVCFObj](#), [9](#)
[ParseConfigFile](#), [10](#)
[PCAWG.All.Sequencing.Artifact.Signatures](#),
[10](#)
[PCAWG.Known.Sequencing.Artifact.Signatures](#),
[11](#)