Package ‘MesKit’

March 31, 2025

Type Package

Title A tool kit for dissecting cancer evolution from multi-region
derived tumor biopsies via somatic alterations

Version 1.16.0

Description MesKit provides commonly used analysis and visualization modules based on mutational data generated by multi-region sequencing (MRS). This package allows to depict mutational profiles, measure heterogeneity within or between tumors from the same patient, track evolutionary dynamics, as well as characterize mutational patterns on different levels. Shiny application was also developed for a need of GUI-based analysis. As a handy tool, MesKit can facilitate the interpretation of tumor heterogeneity and the understanding of evolutionary relationship between regions in MRS study.

License GPL-3

Encoding UTF-8

LazyData TRUE

Depends R (>= 4.0.0)

Imports methods, data.table, Biostrings, dplyr, tidyr (>= 1.0.0), ape (>= 5.4.1), ggrepel, pracma, ggridges, AnnotationDbi, IRanges, circlize, cowplot, mclust, phangorn, ComplexHeatmap (>= 1.9.3), ggplot2, RColorBrewer, grDevices, stats, utils, S4Vectors

RoxygenNote 7.1.1

VignetteBuilder knitr

Suggests shiny, knitr, rmarkdown, BSgenome.Hsapiens.UCSC.hg19 (>= 1.4.0), org.Hs.eg.db, clusterProfiler, TxDb.Hsapiens.UCSC.hg19.knownGene

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

git_branch RELEASE_3_20

git_last_commit d0b7a3c

git_last_commit_date 2024-10-29

Repository Bioconductor 3.20

Date/Publication 2025-03-31

Author Mengni Liu [aut, cre] (<https://orcid.org/0000-0001-9938-9973>),
Jianyu Chen [aut, ctb] (<https://orcid.org/0000-0003-4491-9265>),
Xin Wang [aut, ctb] (<https://orcid.org/0000-0002-6072-599X>)

Maintainer Mengni Liu <niinleslie@gmail.com>

Contents

calFst *calFst*

Description

Genetic divergence between regions of subclonal sSNVs using the Weir and Cockerham method

Usage

```
calFst(
  maf,
  patient.id = NULL,
  min.vaf = 0,
  min.total.depth = 2,
  use.adjVAF = FALSE,
  plot = TRUE,
  withinTumor = FALSE,
  use.circle = TRUE,
  title = NULL,
  number.cex = 8,
  number.col = "#C77960",
  use.tumorSampleLabel = FALSE,
  ...
)
```

Arguments

`maf` A Maf or MafList object generated by `readMaf` function.

`patient.id` Select the specific patients. Default NULL, all patients are included.

`min.vaf` Specify The minimum VAF to filter variants. Default 0.

`min.total.depth` The minimum total allele depth for filtering variants. Default 2.

`use.adjVAF` Use adjusted VAF in analysis when adjusted VAF or CCF is available. Default FALSE.

`plot` Logical (Default: TRUE).

`withinTumor` Logical (Default: FALSE). Whether calculate between-region heterogeneity within tumors.

`use.circle` Logical (Default: TRUE). Whether use "circle" in the plot. as visualization method of correlation matrix

`title` The title of the plot. Default "Nei's distance"

`number.cex` The size of text shown in correlation plot. Default 8.

`number.col` The color of text shown in correlation plot. Default "#C77960".

`use.tumorSampleLabel` Logical (Default: FALSE). Rename the 'Tumor_Sample_Barcode' by 'Tumor_Sample_Label'.

`...` Other options passed to `subMaf`

Value

A list contains Fst value of MRS and Hudson estimator of each sample-pair, respectively.

References

Sun R, Hu Z, Sottoriva A, et al. Between-region genetic divergence reflects the mode and tempo of tumor evolution. Nat Genet. 2017;49(7):1015-1024.

Bhatia G, Patterson N, Sankararaman S, Price AL. Estimating and interpreting FST: the impact of rare variants. Genomic Res. 2013;23(9):1514-1521.

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
calFst(maf)
```

calJSI *compareJSI*

Description

The Jaccard similarity index (JSI) is applied to distinguish monoclonal versus polyclonal seeding in metastases.

Usage

```
calJSI(
  maf,
  patient.id = NULL,
  pairByTumor = FALSE,
  min.ccf = 0,
  plot = FALSE,
  use.circle = TRUE,
  title = NULL,
  number.cex = 8,
  number.col = "#C77960",
  use.tumorSampleLabel = FALSE,
  ...
)
```

Arguments

`maf`	Maf or MafList object generated by `readMaf` function.
`patient.id`	Select the specific patients. Default NULL, all patients are included.
`pairByTumor`	Compare JSI between different tumors. Default FALSE.
`min.ccf`	The minimum value of CCF. Default 0.
`plot`	Logical (Default: FALSE).
`use.circle`	Logical (Default: TRUE). Whether to use "circle" as visualization method of correlation matrix.
`title`	Title of the plot Default "Jaccard similarity".
`number.cex`	The size of text shown in correlation plot. Default 8.

`number.col` The color of text shown in correlation plot. Default "#C77960".

`use.tumorSampleLabel` Logical (Default: FALSE). Rename the 'Tumor_Sample_Barcode' by 'Tumor_Sample_Label'.

`...` Other options passed to `subMaf`

Value

Correlation matrix and heatmap via Jaccard similarity coefficient method

References

Hu, Z., Li, Z., Ma, Z. et al. Multi-cancer analysis of clonality and the timing of systemic spread in paired primary tumors and metastases. Nat Genet (2020).

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
calJSI(maf)
```

`calNeiDist` *calNeiDist*

Description

Nei's distance of CCF for sample/tumor pair.

Usage

```
calNeiDist(
  maf,
  patient.id = NULL,
  withinTumor = FALSE,
  min.ccf = 0,
  plot = TRUE,
  use.circle = TRUE,
  title = NULL,
  number.cex = 8,
  number.col = "#C77960",
  use.tumorSampleLabel = FALSE,
  ...
)
```

Arguments

`maf` A Maf or MafList object generated by `readMaf` function.

`patient.id` Select the specific patients. Default NULL, all patients are included.

`withinTumor` Calculate between-region heterogeneity within tumor. (Default: FALSE).

`min.ccf` Specify the minimum CCF. Default 0.

`plot` Logical (Default: TRUE).

`use.circle` Logical (Default: TRUE). Whether to use "circle" as visualization method of correlation matrix.

`title` The title of the plot. Default "Nei's distance"

`number.cex` The size of text shown in correlation plot. Default 8.

`number.col` The color of text shown in correlation plot. Default "#C77960".

`use.tumorSampleLabel`
Logical (Default: FALSE). Rename the 'Tumor_Sample_Barcode' by 'Tumor_Sample_Label'.

`...` Other options passed to `subMaf`

Value

Nei's genetic distance matrix and heatmap of sample-pairs from the same patient

References

Lee JK, Wang J, Sa JK, et al. Spatiotemporal genomic architecture informs precision oncology in glioblastoma. Nat Genet. 2017;49(4):594-599.

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
calNeiDist(maf)
```

ccfAUC *ccfAUC*

Description

The tumor heterogeneity was estimated as the area under the curve (AUC) of the cumulative density function from all cancer cell fractions per tumor

Usage

```
ccfAUC(
  maf,
  patient.id = NULL,
  min.ccf = 0,
  withinTumor = FALSE,
  plot.density = TRUE,
  use.tumorSampleLabel = FALSE,
  ...
)
```

Arguments

`maf`	A Maf or MafList object generated by `readMaf` function.
`patient.id`	Select the specific patients. Default NULL, all patients are included.
`min.ccf`	The minimum value of CCF. Default 0.
`withinTumor`	Calculate between-region heterogeneity within tumor. Default FALSE.
`plot.density`	Whether to show the density plot. Default TRUE.
`use.tumorSampleLabel`	Logical (Default: FALSE). Rename the 'Tumor_Sample_Barcode' by 'Tumor_Sample_Label'.
`...`	Other options passed to `subMaf`

Value

A list containing AUC of CCF and a graph

References

Charoentong P, Finotello F, et al. Pan-cancer Immunogenomic Analyses Reveal Genotype-Immunophenotype Relationships and Predictors of Response to Checkpoint Blockade. Cell reports 2017, 18:248-262.

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
ccfAUC(maf)
```

`classifyMut` *classifyMut*

Description

classifyMut

Usage

```
classifyMut(maf, patient.id = NULL, class = "SP", classByTumor = FALSE, ...)
```

Arguments

`maf`	Maf or MafList object generated by `readMaf` function. Classify SSNVs/Indels into Shared/P-shared/Private, Clonal/Subclonl or Shared-Clonal/P-shared-Clonal/Private-Clonal/Shared-Subclonal/P-shared-SubClonal/Private-SubClonal
`patient.id`	Select the specific patients. Default NULL, all patients are included
`class`	The class which would be represented. Default: "SP" (Shared pattern: Public/Shared/Private), other options: "CS" (Clonal status: Clonal/Subclonl) and "SPCS".
`classByTumor`	Logical (Default: FALSE). Classify mutations based on "Tumor_ID".
`...`	Other options passed to `subMaf`

Value

A data.frame with classification of mutations for each patient

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
classifyMut(maf, class = "SP")
```

---

cna2gene *cna2gene*

Description

cna2gene

Usage

```
cna2gene(seg, txdb, min.overlap.len = 50, geneList = NULL)
```

Arguments

seg — seg object generated by readSegment function.

txdb — A TxDb object. i.e., TxDb.Hsapiens.UCSC.hg19.knownGene. Default NULL.

min.overlap.len — The minimum insertion size of segment and gene. Default 50.

geneList — The list of genes used to limit the annotation.Default NULL.

Value

seg object

Examples

```
segFile <- system.file("extdata", "CRC_HZ.seg.txt", package = "MesKit")
gisticAmpGenesFile <- system.file("extdata", "COREAD_amp_genes.conf_99.txt", package = "MesKit")
gisticDelGenesFile <- system.file("extdata", "COREAD_del_genes.conf_99.txt", package = "MesKit")
gisticAllLesionsFile <- system.file("extdata", "COREAD_all_lesions.conf_99.txt", package = "MesKit")
seg <- readSegment(segFile = segFile,
                   gisticAmpGenesFile = gisticAmpGenesFile,
                    gisticDelGenesFile = gisticDelGenesFile,
                   gisticAllLesionsFile = gisticAllLesionsFile)

library(TxDb.Hsapiens.UCSC.hg19.knownGene)
library(org.Hs.eg.db)
cna2gene(seg, txdb = TxDb.Hsapiens.UCSC.hg19.knownGene)
```

compareCCF *compareCCF*

Description

Compare the CCF between samples/tumor pairs This function requires CCF for clustering

Usage

```
compareCCF(
  maf,
  patient.id = NULL,
  min.ccf = 0,
  pairByTumor = FALSE,
  use.tumorSampleLabel = FALSE,
  ...
)
```

Arguments

maf Maf or MafList object generated by readMaf function.

patient.id Select the specific patients. Default NULL, all patients are included.

min.ccf The minimum value of CCF. Default 0.

pairByTumor Pair by tumor types in each patients. Default FALSE.

use.tumorSampleLabel
Logical (Default: FALSE). Rename the 'Tumor_Sample_Barcode' by 'Tumor_Sample_Label'.

... Other options passed to subMaf

Value

a result list of CCF comparing between samples/tumor pairs

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
compareCCF(maf)
```

`compareTree` *compareTree*

Description

Compares two phylogenetic trees and returns a detailed report of several distance methods

Usage

```
compareTree(
  phyloTree1,
  phyloTree2,
  plot = FALSE,
  min.ratio = 1/20,
  show.bootstrap = FALSE,
  use.tumorSampleLabel = FALSE
)
```

Arguments

`phyloTree1` A phyloTree object generated by `getPhyloTree` function.

`phyloTree2` A phyloTree object generated by `getPhyloTree` function.

`plot` Logical (Default: FALSE). If TRUE, two trees will be plotted on the same device and their similarities will be shown.

`min.ratio` Double, Default 1/20. If min.ratio is not NULL, all edge length which are smaller than min.ratio*the longest edge length will be reset as min.ratio*longest edge length.

`show.bootstrap` Logical (Default: FALSE). Whether to add bootstrap value on internal nodes.

`use.tumorSampleLabel`
Logical (Default: FALSE). Rename the 'Tumor_Sample_Barcode' by 'Tumor_Sample_Label'.

Value

A vector containing the following tree distance methods by R package phangorn Symmetric.difference Robinson-Foulds distance KF-branch distance the branch score distance (Kuhner & Felsenstein 1994) Path.difference difference in the path length, counted as the number of branches Weighted.path.difference difference in the path length, counted using branches lengths

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")


phyloTree1 <- getPhyloTree(maf$V402, method = "NJ")
phyloTree2 <- getPhyloTree(maf$V402, method = "MP")
compareTree(phyloTree1, phyloTree2)
compareTree(phyloTree1, phyloTree2, plot = TRUE)
```

fitSignatures *fitSignatures*

Description

Find nonnegative linear combination of mutation signatures to reconstruct matrix and calculate cosine similarity based on somatic SNVs.

Usage

```
fitSignatures(
  tri_matrix = NULL,
  patient.id = NULL,
  signaturesRef = "cosmic_v2",
  associated = NULL,
  min.mut.count = 15,
  signature.cutoff = 0.1
)
```

Arguments

`tri_matrix`	A matrix or a list of matrix generated by `triMatrix` function.
`patient.id`	Select the specific patients. Default NULL, all patients are included
`signaturesRef`	Signature reference,Users can upload their own reference. Default "cosmic_v2". Option: "exome_cosmic_v3","nature2013".
`associated`	Associated Vector of associated signatures. If given, will narrow the signatures reference to only the ones listed. Default NULL.
`min.mut.count`	The threshold for the variants in a branch. Default 15.
`signature.cutoff`	Discard any signature relative contributions with a weight less than this amount. Default 0.1.

Value

A list of data frames, each one contains treeMSOutput, containing information about each set/branch's mutational signature.

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")

## Load a reference genome.
library(BSgenome.Hsapiens.UCSC.hg19)

phyloTree <- getPhyloTree(maf, patient.id = 'V402')
tri_matrix <- triMatrix(phyloTree)
fitSignatures(tri_matrix)
```

getBinaryMatrix *getBinaryMatrix*

Description

getBinaryMatrix

Usage

```
getBinaryMatrix(object)

## S4 method for signature 'phyloTree'
getBinaryMatrix(object)
```

Arguments

`object`	An object of phyloTree

Value

Binary matrix of phyloTree

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
phyloTree <- getPhyloTree(maf)
getBinaryMatrix(phyloTree$V402)
```

getBootstrapValue *getBootstrapValue*

Description

getBootstrapValue

Usage

```
getBootstrapValue(object)

## S4 method for signature 'phyloTree'
getBootstrapValue(object)
```

Arguments

`object`	An object of phyloTree

Value

Bootstrap value of phyloTree

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
phyloTree <- getPhyloTree(maf)
getBootstrapValue(phyloTree$V402)
```

getBranchType *getBranchType*

Description

getBranchType

Usage

```
getBranchType(object)

## S4 method for signature 'phyloTree'
getBranchType(object)
```

Arguments

`object`	An object of phyloTree

Value

Branch type of phyloTree

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
phyloTree <- getPhyloTree(maf)
getBranchType(phyloTree$V402)
```

getCCFMatrix *getCCFMatrix*

Description

getCCFMatrix

Usage

```
getCCFMatrix(object)

## S4 method for signature 'phyloTree'
getCCFMatrix(object)
```

Arguments

object	An object of phyloTree

Value

CCF matrix of phyloTree

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
phyloTree <- getPhyloTree(maf)
getCCFMatrix(phyloTree$V402)
```

getMafData *getMafData*

Description

getMafData

Usage

```
getMafData(object)

## S4 method for signature 'Maf'
getMafData(object)
```

Arguments

object	An object of Maf

Value

Maf data

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
getMafData(maf$V402)
```

getMafPatient *getMafPatient*

Description

getMafPatient

Usage

```
getMafPatient(object)

## S4 method for signature 'Maf'
getMafPatient(object)
```

Arguments

`object`	An object of Maf

Value

Human reference genome versions of Maf

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
getMafPatient(maf$V402)
```

getMafRef *getMafRef*

Description

getMafRef

Usage

```
getMafRef(object)

## S4 method for signature 'Maf'
getMafRef(object)
```

Arguments

object	An object of Maf

Value

Human reference genome versions of Maf

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
getMafRef(maf$V402)
```

getMutBranches *getMutBranches*

Description

getMutBranches

Usage

```
getMutBranches(object)

## S4 method for signature 'phyloTree'
getMutBranches(object)
```

Arguments

object	An object of phyloTree

Value

Branches mutation of phyloTree

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
phyloTree <- getPhyloTree(maf)
getMutBranches(phyloTree$V402)
```

`getNonSyn_vc` *getNonSyn_vc*

Description

getNonSyn_vc

Usage

```
getNonSyn_vc(object)

## S4 method for signature 'Maf'
getNonSyn_vc(object)
```

Arguments

`object`	An object of Maf

Value

A list of Variant classifications which are considered as non-silent.

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
getNonSyn_vc(maf$V402)
```

`getPhyloTree` *getPhyloTree*

Description

getPhyloTree

Usage

```
getPhyloTree(
  maf,
  patient.id = NULL,
  method = "NJ",
  min.vaf = 0,
  min.ccf = 0,
  bootstrap.rep.num = 100,
  ...
)
```

Arguments

`maf` Maf or MafList object generated by `readMaf` function

`patient.id` Select the specific patients. Default NULL, all patients are included.

`method` Approach to construct phylogenetic trees. Choose one of "NJ"(Neibor-Joining), "MP"(maximum parsimony), "ML"(maximum likelihood), "FASTME.ols" or "FASTME.bal".

`min.vaf` The minimum value of vaf. Default 0.

`min.ccf` The minimum value of CCF. Default 0

`bootstrap.rep.num` Bootstrap iterations. Default 100.

`...` Other options passed to `subMaf`

Value

PhyloTree or phyloTreeList object

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
phyloTree <- getPhyloTree(maf)
```

`getPhyloTreePatient` *getPhyloTreePatient*

Description

getPhyloTreePatient

Usage

```
getPhyloTreePatient(object)

## S4 method for signature 'phyloTree'
getPhyloTreePatient(object)
```

Arguments

`object`	An object of phyloTree

Value

patientID of phyloTree

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
phyloTree <- getPhyloTree(maf)
getPhyloTreePatient(phyloTree$V402)
```

`getPhyloTreeRef` *getPhyloTreeRef*

Description

getPhyloTreeRef

Usage

```
getPhyloTreeRef(object)

## S4 method for signature 'phyloTree'
getPhyloTreeRef(object)

## S4 method for signature 'phyloTree'
getPhyloTreeTsbLabel(object)
```

Arguments

`object`	An object of phyloTree

Value

Reference genome versions of phyloTree

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
phyloTree <- getPhyloTree(maf)
getPhyloTreeRef(phyloTree$V402)
```

getPhyloTreeTsbLabel *getPhyloTreeRef*

Description

getPhyloTreeRef

Usage

```
getPhyloTreeTsbLabel(object)
```

Arguments

`object`	An object of phyloTree

Value

relationship between Tumor_Sample_Barcode and Tumor_Sample_Label

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
phyloTree <- getPhyloTree(maf)
getPhyloTreeTsbLabel(phyloTree$V402)
```

getSampleInfo *getSampleInfo*

Description

getSampleInfo

Usage

```
getSampleInfo(object)

## S4 method for signature 'Maf'
getSampleInfo(object)
```

Arguments

`object`	An object of Maf

Value

Sample information

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
getSampleInfo(maf$V402)
```

---

getTree *getTree*

Description

getTree

Usage

```
getTree(object)

## S4 method for signature 'phyloTree'
getTree(object)
```

Arguments

`object`	An object of phyloTree

Value

Tree object of phyloTree

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
phyloTree <- getPhyloTree(maf)
getTree(phyloTree$V402)
```

---

getTreeMethod *getTreeMethod*

Description

getTreeMethod

Usage

```
getTreeMethod(object)

## S4 method for signature 'phyloTree'
getTreeMethod(object)
```

Arguments

`object` An object of phyloTree

Value

Tree construction method of phyloTree

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
phyloTree <- getPhyloTree(maf)
getTreeMethod(phyloTree$V402)
```

`Maf-class` *Maf class*

Description

Maf class.

Slots

`data` data.table of MAF file containing somatic mutations.

`sample.info` data.frame of sample information per patient.

`nonSyn.vc` list of variant classifications which are considered as non-silent. Default NULL, use Variant Classifications with "Frame_Shift_Del", "Frame_Shift_Ins", "Splice_Site", "Translation_Start_Site", "Nonsense_Mutation", "Nonstop_Mutation", "In_Frame_Del", "In_Frame_Ins", "Missense_Mutation"

`ref.build` human reference genome version. Default 'hg19'. Optional: 'hg18' or 'hg38'.

`MafList-class` *MafList class*

Description

S4 class for storing a list of Maf objects.

Slots

`.Data` a list of `Maf` objects.

Constructor

`MafList (...)` combine multiple Maf objects supplied in ... into a MafList object.

mathScore *mathScore*

Description

calculates MATH score of each tumor sample or based on Mutant-Allele Tumor Heterogeneity (MATH) approach.

Usage

```
mathScore(
  maf,
  patient.id = NULL,
  withinTumor = FALSE,
  min.vaf = 0,
  use.adjVAF = FALSE,
  segFile = NULL,
  use.tumorSampleLabel = FALSE,
  ...
)
```

Arguments

`maf`	Maf or MafList object generated by `readMaf` function.
`patient.id`	Select the specific patients. Default NULL, all patients are included.
`withinTumor`	Calculate between-region heterogeneity within tumor. Default: FALSE.
`min.vaf`	Specify The minimum VAF to filter variants. Default: 0.
`use.adjVAF`	Use adjusted VAF in analysis when adjusted VAF or CCF is available. Default: FALSE.
`segFile`	The segment file.
`use.tumorSampleLabel`	Logical (Default: FALSE). Rename the 'Tumor_Sample_Barcode' by 'Tumor_Sample_Label'.
`...`	Other options passed to `subMaf`

Value

A data.frame of MATH scores

References

Mroz, Edmund A. et al. Intra-Tumor Genetic Heterogeneity and Mortality in Head and Neck Cancer: Analysis of Data from The Cancer Genome Atlas. Ed. Andrew H. Beck. PLoS Medicine 12.2 (2015): e1001786.

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
mathScore(maf)
```

`mutCluster` *mutCluster*

Description

Cluster mutations based on variant allele frequencies (VAFs) or cancer cell fractions (CCFs).

Usage

```
mutCluster(
  maf,
  patient.id = NULL,
  use.ccf = FALSE,
  segFile = NULL,
  withinTumor = FALSE,
  use.tumorSampleLabel = FALSE,
  ...
)
```

Arguments

`maf`	Maf or MafList object generated by `readMaf` function.
`patient.id`	Select the specific patients. Default NULL, all patients are included.
`use.ccf`	Cluster CCF. Default FALSE.
`segFile`	The segment file.
`withinTumor`	Cluster Tumor average CCF within tumors in each patients. Default FALSE.
`use.tumorSampleLabel`	Logical (Default: FALSE). Rename the 'Tumor_Sample_Barcode' by 'Tumor_Sample_Label'.
`...`	Other options passed to `subMaf`

Value

clustering plots of vaf

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
mutCluster(maf, patient.id = 'V402')
```

mutHeatmap *mutHeatmap*

Description

plot binary or CCF heatmap of somatic mutations.

Usage

```
mutHeatmap(
  maf,
  patient.id = NULL,
  min.vaf = 0,
  min.ccf = 0,
  use.adjVAF = FALSE,
  use.ccf = FALSE,
  geneList = NULL,
  plot.geneList = FALSE,
  show.geneList = TRUE,
  mut.threshold = 50,
  sample.text.size = 9,
  legend.title.size = 10,
  gene.text.size = 9,
  sampleOrder = NULL,
  use.tumorSampleLabel = FALSE,
  classByTumor = FALSE,
  ...
)
```

Arguments

maf	Maf or MafList object generated by readMaf function.
patient.id	Select the specific patients. Default NULL, all patients are included.
min.vaf	The minimum value of VAF. Default 0. Option: on the scale of 0 to 1.
min.ccf	The minimum value of CCF. Default 0. Option: on the scale of 0 to 1.
use.adjVAF	Use adjusted VAF in analysis when adjusted VAF or CCF is available. Default FALSE.
use.ccf	Logical. If FALSE (Default: FALSE), print a binary heatmap of mutations. Otherwise, print a cancer cell frequency (CCF) heatmap.
geneList	List of genes to restrict the analysis. Default NULL.
plot.geneList	Logical (Default: FALSE). If TRUE, plot heatmap with genes on geneList when geneList is not NULL.
show.geneList	Show the names of gene on the geneList. Default TRUE.
mut.threshold	show.gene and show.geneList will be FALSE when patient have more mutations than threshold. Default 150.
sample.text.size	Size of sample name.Default 9.

`legend.title.size`
Size of legend title.Default 10.

`gene.text.size` Size of gene text. Default 9.

`sampleOrder` A named list which contains the sample order used in plotting the heatmap. Default NULL.

`use.tumorSampleLabel`
Logical (Default: FALSE). Rename the 'Tumor_Sample_Barcode' by 'Tumor_Sample_Label'.

`classByTumor` Logical Default: FALSE. Classify mutations based on "Tumor_ID".

`...` Other options passed to `subMaf`

Value

heatmap of somatic mutations

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
mutHeatmap(maf)
```

`mutTrunkBranch` *mutTrunkBranch*

Description

Summarize and conduct paired Fisher test of mutations of trunk/branches in a phylogenetic tree.

Usage

```
mutTrunkBranch(
  phyloTree,
  patient.id = NULL,
  CT = FALSE,
  pvalue = 0.05,
  plot = TRUE
)
```

Arguments

`phyloTree` phyloTree or phyloTreeList object generated by `getPhyloTree` function.

`patient.id` Select the specific patients. Default NULL, all patients are included

`CT` Distinction between C>T at CpG and C>T at other sites. (Default: FALSE).

`pvalue` Confidence level of the interval for Fisher test. Default 0.05.

`plot` Logical. (Default: TRUE).

Value

a list of box plots based on mutational categories

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")

## Load a reference genome.
library(BSgenome.Hsapiens.UCSC.hg19)

phyloTree <- getPhyloTree(maf, patient.id = 'V402')
mutTrunkBranch(phyloTree, plot = TRUE)
```

phyloTree-class *phyloTree class*

Description

S4 class for storing informations about phylogenetic tree.

Slots

`patientID` patient ID.

`tree` a object of class "phylo".

`bootstrap.value` a numeric vector of bootstrap values.

`method` approach to construct a phylogenetic tree.

`binary.matrix` a presense/absent binary matrix of mutations.

`ccf.matrix` a ccf matrix of mutations.

`mut.branches` a data.frame of mutations per trunk/branch.

`branch.type` a data.frame of trunk/branch types based on shared pattern.

`ref.build` human reference genome version. Default: 'hg19'. Optional: 'hg18' or 'hg38'.

`tsb.label` store relationship between Tumor_Sample_Barcode and Tumor_Sample_Label if Tumor_Sample_Label is provided in clinical data.

phyloTreeList-class *phyloTreeList class*

Description

S4 class for storing a list of phyloTree objects.

Slots

`.Data` a list of `phyloTree` objects.

Constructor

`phyloTreeList (...)` combine multiple phyloTree objects supplied in ... into a phyloTreeList object.

`plotCNA` *plotCNA*

Description

plotCNA

Usage

```
plotCNA(
  seg,
  patient.id = NULL,
  sampleOrder = NULL,
  chrSilent = NULL,
  refBuild = "hg19",
  sample.text.size = 11,
  chrom.text.size = 3,
  legend.text.size = 9,
  legend.title.size = 11,
  annot.text.size = 3,
  sample.bar.height = 0.5,
  chrom.bar.height = 0.5,
  showRownames = TRUE,
  removeEmptyChr = TRUE,
  showCytoband = FALSE,
  showGene = FALSE,
  use.tumorSampleLabel = FALSE
)
```

Arguments

`seg` Object generated by `readSegment` function.

`patient.id` Select the specific patients. Default NULL, all patients are included.

`sampleOrder` A named list which contains the sample order used in plotting the final profile. Default NULL.

`chrSilent` Chromosomes excluded in the analysis. e.g, 1, 2, 3. Default NULL.

`refBuild` Human reference genome versions of hg18, hg19 or hg38 by UCSC. Default "hg19".

`sample.text.size` Fontsize of sample name. Default 11.

`chrom.text.size` Fontsize of chromosome text. Default 3.

`legend.text.size` Fontsize of legend text. Default 9.

`legend.title.size` Fontsize of legend title. Default 11.

`annot.text.size` Fontsize of cytoband or gene symbols. Default 3.

`sample.bar.height`
Bar height of each sample. Default 0.5.

`chrom.bar.height`
Bar height of each chromosome. Default 0.5.

`showRownames` Logical (Default: TRUE). Show sample names of rows.

`removeEmptyChr` Remove empty chromosomes that do not exist in all samples. Default TRUE.

`showCytoband` Logical (Default: FALSE). Show cytobands on the plot. Only when the seg object is created with GISTIC results, this parameter can be TRUE.

`showGene` Logical (Default: FALSE). Show gene symbols on the plot. Only when the seg object is created with txdb, this parameter can be TRUE.

`use.tumorSampleLabel`
Logical (Default: FALSE). Rename the 'Tumor_Sample_Barcode' with 'Tumor_Sample_Label'.

Value

a heatmap plot of CNA profile

Examples

```
segFile <- system.file("extdata", "CRC_HZ.seg.txt", package = "MesKit")
seg <- readSegment(segFile = segFile)
plotCNA(seg)

## showCytoband
gisticAmpGenesFile <- system.file("extdata", "COREAD_amp_genes.conf_99.txt", package = "MesKit")
gisticDelGenesFile <- system.file("extdata", "COREAD_del_genes.conf_99.txt", package = "MesKit")
gisticAllLesionsFile <- system.file("extdata", "COREAD_all_lesions.conf_99.txt", package = "MesKit")
seg <- readSegment(segFile = segFile,
                   gisticAmpGenesFile = gisticAmpGenesFile,
                    gisticDelGenesFile = gisticDelGenesFile,
                   gisticAllLesionsFile = gisticAllLesionsFile)
plotCNA(seg, showCytoband = TRUE)
```

`plotMutProfile` *plotMutProfile*

Description

plotMutProfile

Usage

```
plotMutProfile(
  maf,
  patient.id = NULL,
  class = "SP",
  classByTumor = FALSE,
  topGenesCount = 10,
  geneList = NULL,
  sample.text.size = 11,
  gene.text.size = 11,
  legend.text.size = 11,
  legend.title.size = 11,
  bgCol = "#f0f0f0",
  patientsCol = NULL,
  removeEmptyCols = TRUE,
  removeEmptyRows = TRUE,
  showColnames = TRUE,
  sampleOrder = NULL,
  use.tumorSampleLabel = FALSE,
  ...
)
```

Arguments

`maf` Maf or MafList object generated by `readMaf` function.

`patient.id` Select or reorder the patients. Default NULL, all patients are included. Classify SSNVs/Indels into Shared/P-shared/Private, Clonal/Subclonl or Shared-Clonal/P-shared-Clonal/Private-Clonal/Shared-Subclonal/P-shared-SubClonal/Private-SubClonal

`class` The class which would be represented. Default "SP" (Shared pattern: Public/Shared/Private), other options: "CS" (Clonal status: Clonal/Subclonl) and "SPCS".

`classByTumor` Logical (Default: FALSE). Define shared pattern of mutations based on tumor types (TRUE) or samples (FALSE)

`topGenesCount` The number of genes print, Default 10.

`geneList` A list of genes to restrict the analysis. Default NULL.

`sample.text.size` Fontsize of sample name. Default 11.

`gene.text.size` Fontsize of gene text. Default 11.

`legend.text.size` Fontsize of legend text. Default 11.

`legend.title.size` Fontsize of legend title. Default 11.

`bgCol` Background grid color. Default "#f0f0f0".

`patientsCol` A list containing customized colors for distinct patients. Default NULL.

`removeEmptyCols` Logical (Default: TRUE). Whether remove the samples without alterations.

`removeEmptyRows` Logical (Default: TRUE). Whether remove the genes without alterations.

`showColnames` Logical (Default: TRUE). Show sample names of columns.

`sampleOrder` A named list which contains the sample order used in plotting the final profile. Default NULL.

`use.tumorSampleLabel` Logical (Default: FALSE). Rename the ’Tumor_Sample_Barcode’ with ’Tumor_Sample_Label’.

`...` Other options passed to `subMaf`

Value

Mutational profile

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
plotMutProfile(maf, class = "SP")
```

`plotMutSigProfile` *plotMutSigProfile*

Description

plotMutSigProfile

Usage

```
plotMutSigProfile(
  sig_input,
  patient.id = NULL,
  mode = NULL,
  contribution.type = "relative",
  use.tumorSampleLabel = FALSE
)
```

Arguments

`sig_input` Result generated by function `fitSignatures` or `triMatrix`.

`patient.id` Select the specific patients. Default NULL, all patients are included.

`mode` Type of mutation spectrum. Default NULL. Options:'Original','Reconstructed' or 'Difference'

`contribution.type` Type of Signature contribution. Default 'realative'. Options:'realative' or 'absolute'.

`use.tumorSampleLabel` Logical (Default: FALSE). Rename the 'Tumor_Sample_Barcode' by 'Tumor_Sample_Label'.

Value

Mutational signature profile of patients

Examples

```
## input from fitSignatures
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
phyloTree <- getPhyloTree(maf, patient.id = 'V402')

## Load a reference genome.
library(BSgenome.Hsapiens.UCSC.hg19)

tri_matrix <- triMatrix(phyloTree)
fit_out <- fitSignatures(tri_matrix)
plotMutSigProfile(fit_out)
## input from treeMatrix
plotMutSigProfile(tri_matrix)
```

plotPhyloTree *plotPhyloTree*

Description

plotPhyloTree

Usage

```
plotPhyloTree(
  phyloTree,
  patient.id = NULL,
  branchCol = "mutType",
  show.bootstrap = TRUE,
  min.ratio = 1/20,
  signaturesRef = "cosmic_v2",
  min.mut.count = 15,
  use.tumorSampleLabel = FALSE,
  show.scale.bar = FALSE,
  scale.bar.x = NULL,
  scale.bar.y = NULL
)
```

Arguments

`phyloTree`	phyloTree or phyloTreeList object generated by `getPhyloTree` function.
`patient.id`	Select the specific patients. Default NULL, all patients are included.
`branchCol`	Specify the colors of branches Default 'mutType'. Other options: "mutSig" for coloring branches by branch mutation signature;
`show.bootstrap`	Logical (Default: TRUE). Whether to add bootstrap value on internal nodes.
`min.ratio`	Double. Default 1/20. If min.ratio is not NULL, all edge length of a phylogenetic tree should be greater than min.ratio*the longest edge length. If not, the edge length will be reset as min.ratio*longest edge length.

`signaturesRef` Signature reference,Users can upload their own reference. Default "cosmic_v2". Option:"exome_cosmic_v3","nature2013".

`min.mut.count` The threshold for the variants in a branch. Default 15.

`use.tumorSampleLabel` Logical (Default: FALSE). Rename the 'Tumor_Sample_Barcode' with 'Tumor_Sample_Label'.

`show.scale.bar` Logical (Default: FALSE). Whether to show scale bar.This function adds a horizontal bar giving the scale of the branch lengths to a plot on the current graphical device.

`scale.bar.x` The x location of scale bar.

`scale.bar.y` The y location of scale bar.

Value

return a list of phylotree graph .

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")

phyloTree <- getPhyloTree(maf, patient.id = 'V402')
plotPhyloTree(phyloTree)
```

readMaf *readMaf*

Description

Read tab delimited MAF (can be plain text or *.gz compressed) file along with sample information file.

Usage

```
readMaf(
  mafFile,
  clinicalFile,
  ccfFile = NULL,
  adjusted.VAF = FALSE,
  nonSyn.vc = NULL,
  use.indel.ccf = FALSE,
  ccf.conf.level = 0.95,
  refBuild = "hg19"
)
```

Arguments

`mafFile` A tab delimited MAF file (plain text or *.gz compressed). Required.

`clinicalFile` A clinical data file includes Tumor_Sample_Barcode, Tumor_ID, Patient_ID. Tumor_Sample_Label is optional. Default NULL.

`ccfFile` A CCF file of somatic mutations. Default NULL.

`adjusted.VAF` Whether adjusted VAF is included in mafFile. Default FALSE.

`nonSyn.vc` List of Variant classifications which are considered as non-silent. Default NULL, use Variant Classifications with "Frame_Shift_Del","Frame_Shift_Ins","Splice_Site","Translation_St

`use.indel.ccf` Whether include indels in ccfFile. Default FALSE.

`ccf.conf.level` The confidence level of CCF to identify clonal or subclonal. Only works when "CCF_std" or "CCF_CI_high" is provided in ccfFile. Default 0.95.

`refBuild` Human reference genome version. Default 'hg19'. Optional: 'hg18' or 'hg38'.

Value

an object of Maf or MafList.

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File,clinicalFile = clin.File, refBuild="hg19")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
```

readSegment *readSegment*

Description

readSegment

Usage

```
readSegment(
  segFile,
  gisticAmpGenesFile = NULL,
  gisticDelGenesFile = NULL,
  gisticAllLesionsFile = NULL,
  gistic.qval = 0.25,
  min.seg.size = 500,
  txdb = NULL,
  min.overlap.len = 50,
  verbose = TRUE,
  ...
)
```

Arguments

`segFile` The segment file.

`gisticAmpGenesFile` Amplification Genes file generated by GISTIC. Default NULL.

`gisticDelGenesFile` Deletion Genes file generated by GISTIC. Default NULL.

`gisticAllLesionsFile` Information of all lesions generated by GISTIC. Default NULL.

`gistic.qval` The threshold of gistic Q value. Default 0.25.

`min.seg.size` The smallest size of segments. Default 500.

`txdb` A TxDb object. i.e., TxDb.Hsapiens.UCSC.hg19.knownGene. Default NULL.

`min.overlap.len` The minimum insertion size of segment and gene. Default 50.

`verbose` Whether to display details in the console. Default TRUE.

`...` ... Other options passed to `cna2gene`.

Value

a list of segmentation data frame

Examples

```
segFile <- system.file("extdata", "CRC_HZ.seg.txt", package = "MesKit")
gisticAmpGenesFile <- system.file("extdata", "COREAD_amp_genes.conf_99.txt", package = "MesKit")
gisticDelGenesFile <- system.file("extdata", "COREAD_del_genes.conf_99.txt", package = "MesKit")
gisticAllLesionsFile <- system.file("extdata", "COREAD_all_lesions.conf_99.txt", package = "MesKit")
seg <- readSegment(segFile = segFile,
                   gisticAmpGenesFile = gisticAmpGenesFile,
                   gisticDelGenesFile = gisticDelGenesFile,
                   gisticAllLesionsFile = gisticAllLesionsFile)
```

`runMesKit` *Run the default MesKit app for analysis locally*

Description

`runMesKit` run MesKit locally

Usage

```
runMesKit()
```

Value

a shiny app window

Author(s)

Mengni Liu

Examples

```
runMesKit()
```

subMaf *Subset Maf object*

Description

Subset Maf object

Usage

```
subMaf(
  maf,
  mafObj = FALSE,
  patient.id = NULL,
  geneList = NULL,
  chrSilent = NULL,
  mutType = "All",
  use.indel = TRUE,
  min.vaf = 0,
  max.vaf = 1,
  min.average.vaf = 0,
  min.ccf = 0,
  min.ref.depth = 0,
  min.alt.depth = 0,
  min.total.depth = 0,
  clonalStatus = NULL,
  use.adjVAF = FALSE,
  use.tumorSampleLabel = FALSE
)
```

Arguments

`maf` Maf or MafList object generated by `readMaf` function.

`mafObj` return Maf class. (Default: FALSE).

`patient.id` Select the specific patients. Default NULL, all patients are included.

`geneList` A list of genes to restrict the analysis. Default NULL.

`chrSilent` Chromosomes excluded in the analysis. e.g, 1, 2, X, Y. Default NULL.

`mutType` Select Proper variant classification you need. Default "All". Option: "nonSyn".

`use.indel` Logical value. Whether to use INDELs besides somatic SNVs. (Default: TRUE).

`min.vaf` The minimum VAF for filtering variants. Default 0.

`max.vaf` The maximum VAF for filtering variants. Default 1.

`min.average.vaf` The minimum tumor average VAF for filtering variants. Default 0.

`min.ccf` The minimum CCF for filtering variants. Default NULL.

`min.ref.depth` The minimum reference allele depth for filtering variants. Default 0.

`min.alt.depth` The minimum alteratation allele depth for filtering variants. Default 0.

`min.total.depth`
The minimum total allele depth for filtering variants. Default 0.

`clonalStatus` Subset by clonal status. Default NULL. Option: "Clonal","Subclonal".

`use.adjVAF` Use adjusted VAF in analysis when adjusted VAF or CCF is available. Default FALSE.

`use.tumorSampleLabel`
Logical (Default: FALSE). Rename the 'Tumor_Sample_Barcode' by 'Tumor_Sample_Label'.

Value

Maf object or Maf data.

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
maf_data <- subMaf(maf)
```

`testNeutral` *testNeutral*

Description

Evaluate whether a tumor follows neutral evolution or under strong selection during the growth based on variant frequency distribution (VAF) of subclonal mutations. The subclonal mutant allele frequencies of a follow a simple power-law distribution predicted by neutral growth.

Usage

```
testNeutral(
  maf,
  patient.id = NULL,
  withinTumor = FALSE,
  min.total.depth = 2,
  min.vaf = 0.1,
  max.vaf = 0.3,
  R2.threshold = 0.98,
  min.mut.count = 20,
  plot = TRUE,
  use.tumorSampleLabel = FALSE,
  ...
)
```

Arguments

`maf` Maf or MafList object generated by `readMaf` function.

`patient.id` Select the specific patients. Default NULL, all patients are included.

`withinTumor` Test neutral within tumros in each patients. Default FALSE.

`min.total.depth` The minimun total depth of coverage. Defalut 2

`min.vaf` The minimum value of adjusted VAF value. Default 0.1

`max.vaf` The maximum value of adjusted VAF value. Default 0.3

`R2.threshold` The threshod of R2 to decide whether a tumor follows neutral evolution. Default 0.98

`min.mut.count` The minimun number of subclonal mutations used to fit model. Default 20

`plot` Logical, whether to print model fitting plot of each sample. Default TRUE.

`use.tumorSampleLabel` Let Tumor_Sample_Barcode replace Tumor_Sample_Label if Tumor Label is provided in clinical data. Default FALSE.

`...` Other options passed to `subMaf`

Value

the neutrality metrics and model fitting plots

References

Williams, M., Werner, B. et al. Identification of neutral tumor evolution across cancer types. Nat Genet 48, 238-244 (2016)

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")
testNeutral(maf)
```

`triMatrix` *triMatrix*

Description

Calculate the frequency of 96 trinucleotide mutation based on somatic SNVs.

Usage

```
triMatrix(phyloTree, patient.id = NULL, level = 2)
```

Arguments

`phyloTree`	phyloTree or phyloTreeList object generated by `getPhyloTree` function.
`patient.id`	Select the specific patients. Default NULL, all patients are included
`level`	Calculate the frequency of 96 trinucleotide mutatio on different levels. 1: patient-level, 2: tumor-level, 3: sample-level, 4: branch-level, 5: shared pattern (public/shared/private) of each tumor. 6: trunk/branch-level. Default 2.

Value

The frequency of 96 trinucleotide mutation.

Examples

```
maf.File <- system.file("extdata/", "CRC_HZ.maf", package = "MesKit")
clin.File <- system.file("extdata/", "CRC_HZ.clin.txt", package = "MesKit")
ccf.File <- system.file("extdata/", "CRC_HZ.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, clinicalFile = clin.File, ccfFile=ccf.File, refBuild="hg19")

## Load a reference genome.
library(BSgenome.Hsapiens.UCSC.hg19)

phyloTree <- getPhyloTree(maf, patient.id = 'V402')
triMatrix(phyloTree)
```

Index