Package 'MesKit'

January 25, 2021

Type Package

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

Version 1.1.0

Description MesKit provides commonly used analysis and visualization modules based on mutational data generated by multi-region sequencing (MRS). This package allows to decipher ITH, infer metastatic routes as well as uncover the underlying process of mutagenesis. Shiny application was also developed for a need of GUI-based analysis. As a handy tool, MesKit can facilitate the understanding of cancer cell evolution and its relevance to cancer therapeutics.

License MIT + file LICENSE

Encoding UTF-8

LazyData TRUE

Depends R (>= 4.0.0)

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

RoxygenNote 7.1.1

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

VignetteBuilder knitr

biocViews Software, SomaticMutation, GeneticVariability, Genetics, Classification, VariantAnnotation, DataRepresentation, Visualization, Sequencing, GUI

NeedsCompilation no

BugReports https://github.com/Niinleslie/MesKit/issues

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

git_branch master

git_last_commit 975d2fa

git_last_commit_date 2020-10-27

Date/Publication 2021-01-24

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>

R topics documented:

calFst *calFst*

Description

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

Usage

```
calFst(
  maf,
  patient.id = NULL,
  min.vaf = 0.02,
  min.total.depth = 2,
  plot = TRUE,
  withinTumor = FALSE,
  use.circle = TRUE,
  title = NULL,
  number.cex = 8,
  number.col = "#C77960",
  ...
)
```

Arguments

maf	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_adj, default is 0.02
min.total.depth	The minimum total allele depth for filtering variants. Default: 2.
plot	Logical (Default:TRUE). Whether to show the plot, default TRUE
withinTumor	Calculate fst within types in each patients,default is FALSE.
use.circle	Logical (Default:TRUE). Whether to use "circle" as visualization method of correlation matrix
title	The title of the plot. Default is "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".
...	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", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, ccfFile = ccf.File, refBuild="hg19")
calFst(maf)
```

`calNeiDist` *calNeiDist*

Description

Nei's distance of CCF for each sample-pair

Usage

```
calNeiDist(
  maf,
  patient.id = NULL,
  withinTumor = FALSE,
  min.ccf = 0.02,
  plot = TRUE,
  use.circle = TRUE,
  title = NULL,
  number.cex = 8,
  number.col = "#C77960",
  ...
)
```

Arguments

`maf`	Maf or MafList object generated by `readMaf` function.
`patient.id`	Select the specific patients. Default: NULL, all patients are included.
`withinTumor`	Calculate fst within types in each patients,default is FALSE.
`min.ccf`	Specify the minimum CCF, default is 0.08
`plot`	Logical (Default:TRUE). Whether to show the plot. Default TRUE
`use.circle`	Logical (Default:TRUE). Whether to use "circle" as visualization method of correlation matrix
`title`	The title of the plot. Default is "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".
`...`	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", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.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,
  ...
)
```

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
withinTumor	Calculate AUC within types in each patients,default is FALSE.
plot.density	Whether to show the density plot. Default: TRUE
...	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", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.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 is "SP" (Shared pattern: Public/Shared/Private), other options: "CS" (Clonal status: Clonal/Subclonl) and "SPCS".
classByTumor	FALSE(Default). 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", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, ccfFile = ccf.File, refBuild="hg19")
classifyMut(maf, class = "SP")
```

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, ...)
```

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 is FALSE.
...	Other options passed to subMaf

Value

a result list of CCF comparing between samples/tumor pairs

Examples

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

---

compareJSI *compareJSI*

Description

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

Usage

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

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:TRUE).

use.circle	Logical (Default:TRUE). Whether to use "circle" as visualization method of correlation matrix.
title	Title of the plot, default is "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".
...	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", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, ccfFile = ccf.File, refBuild="hg19")
compareJSI(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,
  common.col = "red"
)
```

Arguments

phyloTree1	A phyloTree object generated by getPhyloTree function.
phyloTree2	A phyloTree object generated by getPhyloTree function.
plot	FALSE(Default). 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. Whether to add bootstrap value on internal nodes.Default is TRUE.
common.col	Color of common branches.

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", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, ccfFile = ccf.File, refBuild="hg19")


phyloTree1 <- getPhyloTree(maf$HCC5647, method = "NJ")
phyloTree2 <- getPhyloTree(maf$HCC5647, 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 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", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, ccfFile = ccf.File, refBuild="hg19")

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

phyloTree <- getPhyloTree(maf, patient.id = 'HCC8257')
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", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile = maf.File, ccfFile = ccf.File, refBuild="hg19")
phyloTree <- getPhyloTree(maf)
getBinaryMatrix(phyloTree$HCC5647)
```

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", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile = maf.File, ccfFile = ccf.File, refBuild="hg19")
phyloTree <- getPhyloTree(maf)
getBootstrapValue(phyloTree$HCC5647)
```

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", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile = maf.File, ccfFile = ccf.File, refBuild="hg19")
phyloTree <- getPhyloTree(maf)
getBranchType(phyloTree$HCC5647)
```

---

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", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile = maf.File, ccfFile = ccf.File, refBuild="hg19")
phyloTree <- getPhyloTree(maf)
getCCFMatrix(phyloTree$HCC5647)
```

---

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", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, ccfFile = ccf.File, refBuild="hg19")
getMafData(maf$HCC5647)
```

---

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", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, ccfFile = ccf.File, refBuild="hg19")
getMafPatient(maf$HCC5647)
```

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", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, ccfFile = ccf.File, refBuild="hg19")
getMafRef(maf$HCC5647)
```

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", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile = maf.File, ccfFile = ccf.File, refBuild="hg19")
phyloTree <- getPhyloTree(maf)
getMutBranches(phyloTree$HCC5647)
```

`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", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, ccfFile = ccf.File, refBuild="hg19")
getNonSyn_vc(maf$HCC5647)
```

`getPhyloTree` *getPhyloTree*

Description

getPhyloTree

Usage

```
getPhyloTree(
  maf,
  patient.id = NULL,
  method = "NJ",
  min.vaf = 0.02,
  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 0.02.
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", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile = maf.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", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile = maf.File, ccfFile = ccf.File, refBuild="hg19")
phyloTree <- getPhyloTree(maf)
getPhyloTreePatient(phyloTree$HCC5647)
```

`getPhyloTreeRef` *getPhyloTreeRef*

Description

getPhyloTreeRef

Usage

```
getPhyloTreeRef(object)

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

Arguments

`object`	An object of phyloTree

Value

Reference genome versions of phyloTree

Examples

```
maf.File <- system.file("extdata", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile = maf.File, ccfFile = ccf.File, refBuild="hg19")
phyloTree <- getPhyloTree(maf)
getPhyloTreeRef(phyloTree$HCC5647)
```

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", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, ccfFile = ccf.File, refBuild="hg19")
getSampleInfo(maf$HCC5647)
```

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", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile = maf.File, ccfFile = ccf.File, refBuild="hg19")
phyloTree <- getPhyloTree(maf)
getTree(phyloTree$HCC5647)
```

---

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", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile = maf.File, ccfFile = ccf.File, refBuild="hg19")
phyloTree <- getPhyloTree(maf)
getTreeMethod(phyloTree$HCC5647)
```

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 = 0.02, ...)
```

Arguments

`maf`	Maf or MafList object generated by `readMaf` function.
`patient.id`	Select the specific patients. Default: NULL, all patients are included.
`withinTumor`	Calculate AUC within types in each patients. Default :FALSE.
`min.vaf`	The minimum VAF for filtering variants. Default: 0.02
`...`	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", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, ccfFile = ccf.File, refBuild="hg19")
mathScore(maf)
```

`mutHeatmap` *mutHeatmap*

Description

plot binary or CCF heatmap of somatic mutations.

Usage

```
mutHeatmap(
  maf,
  patient.id = NULL,
  min.vaf = 0.02,
  min.ccf = 0,
  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,
  ...
)
```

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: 0.02. Option: on the scale of 0 to 1.
`min.ccf`	The minimum value of CCF. Default: 0.02. Option: on the scale of 0 to 1.
`use.ccf`	Logical. If FALSE (default), 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`	If TRUE, plot heatmap with genes on geneList when geneList is not NULL.Default FALSE.

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 is 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.
...	Other options passed to subMaf

Value

heatmap of somatic mutations

Examples

```
maf.File <- system.file("extdata", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, ccfFile = ccf.File, refBuild="hg19")
mutHeatmap(maf, patient.id = 'HCC8257')
```

---

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", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, ccfFile = ccf.File, refBuild="hg19")

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

phyloTree <- getPhyloTree(maf, patient.id = 'HCC8257')
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'.

---

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,
  sample.bar.height = 0.5,
  chrom.bar.height = 0.5,
  showRownames = TRUE
)
```

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	Size of sample name.Default 11.
chrom.text.size	Size of chromosome text.Default 3.
legend.text.size	Size of legend text. Default 9.
legend.title.size	Size of legend title.Default 11.
sample.bar.height	Bar height of each sample .Default 0.5.
chrom.bar.height	Bar height of each chromosome .Default 0.5.
showRownames	TRUE(Default). Show sample names of rows.

Value

a heatmap plot of CNA profile

Examples

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

---

plotMutProfile *plotMutProfile*

Description

plotMutProfile

Usage

```
plotMutProfile(
  maf,
  patient.id = NULL,
  class = "SP",
  classByTumor = FALSE,
  topGenesCount = 10,
  geneList = NULL,
  bgCol = "#f0f0f0",
  patientsCol = NULL,
  remove_empty_columns = TRUE,
  remove_empty_rows = TRUE,
  showColnames = TRUE,
  sampleOrder = NULL,
  ...
)
```

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 is "SP" (Shared pattern: Public/Shared/Private), other options: "CS" (Clonal status: Clonal/Subclonl) and "SPCS".
classByTumor	FALSE(Default). Define shared pattern of mutations based on tumor types (TRUE) or samples (FALSE)
topGenesCount	The number of genes print, default is 10
geneList	A list of genes to restrict the analysis. Default NULL.
bgCol	Background grid color. Default: "#f0f0f0"
patientsCol	A list containing customized colors for distinct patients. Default: NULL.
remove_empty_columns	Whether remove the samples without alterations. Only works when plot is TRUE

remove_empty_rows	Whether remove the genes without alterations. Only works when plot is TRUE
showColnames	TRUE(Default). Show sample names of columns.
sampleOrder	A named list which contains the sample order used in plotting the final profile. Default: NULL
...	Other options passed to subMaf

Value

Mutation profile

Examples

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

---

plotMutSigProfile *plotMutSigProfile*

Description

plotMutSigProfile

Usage

```
plotMutSigProfile(sig_input, patient.id = NULL, mode = NULL)
```

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'

Value

Mutation signature profile of patients

Examples

```
## input from fitSignatures
maf.File <- system.file("extdata", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, ccfFile = ccf.File, refBuild="hg19")
phyloTree <- getPhyloTree(maf, patient.id = 'HCC8257')

## 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
)
```

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. Whether to add bootstrap value on internal nodes. Default is TRUE.
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.
	are mapped along the trees as indicated

Value

return a list of phylotree graph .

Examples

```
maf.File <- system.file("extdata", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, ccfFile = ccf.File, refBuild="hg19")

phyloTree <- getPhyloTree(maf, patient.id = 'HCC8257')
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,
  ccfFile = NULL,
  adjusted.VAF = FALSE,
  nonSyn.vc = NULL,
  ccf.conf.level = 0.95,
  refBuild = "hg19"
)
```

Arguments

mafFile	Tab delimited MAF file (plain text or *.gz compressed). Required.
ccfFile	CCF file of somatic mutations. Default NULL.
adjusted.VAF	Let VAF = VAF_adj.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
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/", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata/", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, refBuild="hg19")
maf <- readMaf(mafFile=maf.File, ccfFile=ccf.File, refBuild="hg19")
```

readSegment *readSegment*

Description

readSegment

Usage

```
readSegment(
  segFile = NULL,
  gisticAmpGenesFile = NULL,
  gisticDelGenesFile = NULL,
  gisticAllLesionsFile = NULL,
  gistic.qval = 0.25,
  min.seg.size = 500,
  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 is 0.25
min.seg.size	The smallest size of segments.Default is 500.
verbose	whether to display details in the console. Default True.

Value

a list of segmentation data frame

Examples

```
segFile <- system.file("extdata", "HCC_LDC.seg.txt", package = "MesKit")
gisticAmpGenesFile <- system.file("extdata", "LIHC_amp_genes.conf_99.txt", package = "MesKit")
gisticDelGenesFile <- system.file("extdata", "LIHC_del_genes.conf_99.txt", package = "MesKit")
gisticAllLesionsFile <- system.file("extdata", "LIHC_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,
  geneList = NULL,
  chrSilent = NULL,
  mutType = "All",
  use.indel = TRUE,
  min.vaf = 0,
  max.vaf = 1,
  min.average.vaf = 0,
  min.average.adj.vaf = 0,
  min.ccf = 0,
  min.ref.depth = 0,
  min.alt.depth = 0,
  min.total.depth = 0,
  clonalStatus = NULL,
  use.adjVAF = FALSE
)
```

Arguments

maf	Maf or MafList object generated by readMaf function.
mafObj	return Maf class.Default FALSE.
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.average.adj.vaf	The minimum tumor average ajust 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	Let VAF = VAF_adj, Tumor_Average_VAF = Tumor_Average_VAF_adj.Default: FALSE.

Value

Maf object or Maf data.

Examples

```
maf.File <- system.file("extdata", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.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,
  ...
)
```

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 is 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
...	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", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.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, withinTumor = FALSE)
```

Arguments

phyloTree	phyloTree or phyloTreeList object generated by getPhyloTree function.
patient.id	Select the specific patients. Default: NULL, all patients are included
withinTumor	Exploring signatures within tumor. Default: FALSE.

Value

The frequency of 96 trinucleotide mutation.

Examples

```
maf.File <- system.file("extdata", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, ccfFile = ccf.File, refBuild="hg19")

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

phyloTree <- getPhyloTree(maf, patient.id = 'HCC8257')
triMatrix(phyloTree)
```

vafCluster *vafCluster*

Description

Generate variant allele frequency (VAF) frequency distribution curve.

Usage

```
vafCluster(
  maf,
  patient.id = NULL,
  segFile = NULL,
  min.vaf = 0.02,
  max.vaf = 1,
  withinTumor = FALSE,
  ...
)
```

Arguments

maf	Maf or MafList object generated by readMaf function.
patient.id	Select the specific patients. Default: NULL, all patients are included.
segFile	The segment file.
min.vaf	The minimum value of VAF. Default: 0. Option: on the scale of 0 to 1
max.vaf	The maximum value of VAF. Default: 0. Option: on the scale of 0 to 1
withinTumor	Cluster VAF within tumors in each patients,default is FALSE.
...	Other options passed to subMaf

Value

clustering plots of vaf

Examples

```
maf.File <- system.file("extdata", "HCC_LDC.maf", package = "MesKit")
ccf.File <- system.file("extdata", "HCC_LDC.ccf.tsv", package = "MesKit")
maf <- readMaf(mafFile=maf.File, ccfFile = ccf.File, refBuild="hg19")
vafCluster(maf, patient.id = 'HCC8257')
```

Index