Package 'LACHESIS'

April 7, 2026

Title Functions used to analyze early tumor evolution from whole genome sequencing data

Version 0.99.5

Description This package provides modalities to analyze tumor evolution from whole genome sequencing data. In particular, it provides estimates of mutation densities at genomic segments and uses these to time the origin of the tumor.

License GPL (>= 3)

Copyright Verena Körber, Anand Mayakonda, Maximilia Eggle (2025)

Encoding UTF-8

Roxygen list(markdown = TRUE)

Depends R (>= 4.3)

Imports data.table, vcfR, tidyr, stats, utils, graphics, grDevices, ggplot2, gridExtra, survival, survminer, RColorBrewer, Biostrings

Suggests BSgenome.Hsapiens.UCSC.hg19, BiocStyle, Cairo, rmarkdown, knitr, R.utils, tinytest, GenomeInfoDb, GenomicRanges, IRanges, MutationalPatterns, magick

biocViews Software, StatisticalMethod, TimeCourse, Sequencing, WholeGenome, Survival, SomaticMutation

VignetteBuilder knitr

RoxygenNote 7.3.3

URL https://github.com/VerenaK90/LACHESIS

BugReports https://github.com/VerenaK90/LACHESIS/issues

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

git_branch devel

git_last_commit 1828e01

git_last_commit_date 2026-03-16

Repository Bioconductor 3.23

Date/Publication 2026-04-06

Author Verena Körber [aut, cre] (ORCID:
<https://orcid.org/0009-0005-3888-2648>),
Anand Mayakonda [aut],
Maximilia Eggle [aut]

Maintainer Verena Körber <verena.korber@ndcls.ox.ac.uk>

Contents

`classifyLACHESIS` *Classify a tumor's start of clonal outgrowth during tumorigenesis as "Early MRCA" or "Late MRCA" (favorable/ unfavorable prognosis) depending on the mutation density at its MRCA*

Description

Takes SNV density timing as computed by `LACHESIS` as input and classifies the tumors in the cohort.

Usage

```
classifyLACHESIS(
  lachesis,
  mrca.cutpoint = NULL,
  infer.cutpoint = FALSE,
  entity = "neuroblastoma",
  lach.col.multi = "#176A02",
  lach.col.zero = "#4FB12B",
  surv.time = "OS.time",
```

```
  surv.event = "OS",
  surv.time.scale = 1,
  output.dir = NULL
)
```

Arguments

lachesis	output generated from LACHESIS.
mrca.cutpoint	optional; value based on SNV_densities_cohort.pdf observation, will be used as inferred from a test data set if not specified by user.
infer.cutpoint	logical; should the MRCA cutpoint be inferred from the data?
entity	optional; the tumor entity if classifying according to a pre-defined threshold. Currently, only "neuroblastoma" is supported.
lach.col.multi	optional, bar color for multi-copy SSNV densities.
lach.col.zero	optional, bar color for single-copy SSNV densities.
surv.time	column name containing survival time; defaults to OS.time.
surv.event	column name containing event; defaults to OS.
surv.time.scale	numeric value by which survival time is to be divided (e.g., 365 for converting days into years, 30 for months), defaults to 1.
output.dir	link to directory in which output is to be stored.

Value

data.table with binary assignment early/ late

Examples

```
# An example file with sample annotations and meta data
input.files <- system.file("extdata", "Sample_template.txt",
    package = "LACHESIS"
)
input.files <- data.table::fread(input.files)

# cnv and snv files for example tumors
nbe11 <- list.files(system.file("extdata/NBE11/", package = "LACHESIS"),
    full.names = TRUE
)
nbe15 <- list.files(system.file("extdata/NBE15/", package = "LACHESIS"),
    full.names = TRUE
)
nbe26 <- list.files(system.file("extdata/NBE26/", package = "LACHESIS"),
    full.names = TRUE
)

cnv.file <- c(nbe11[1], nbe15[1], nbe26[1])
snv.file <- c(nbe11[2], nbe15[2], nbe26[2])
```

```
input.files$cnv.file <- cnv.file
input.files$snv.file <- snv.file

# Make an example input file with paths to cnv and snv file along with other
# meta data
lachesis_input <- tempfile(
    pattern = "lachesis", tmpdir = tempdir(),
    fileext = ".tsv"
)
data.table::fwrite(x = input.files, file = lachesis_input, sep = "\t")

# Example with template file with paths to multiple cnv/snv files as an input
lachesis <- LACHESIS(input.files = lachesis_input)
classifyLACHESIS(lachesis)
```

`clonalMutationCounter` *Count clonal mutations on one or several chromosomal copies*

Description

This function counts the number of clonal mutations residing on a single or multiple copies per genomic segment. Segments of equal copy number and B-allele count are merged per chromosome.

Usage

```
clonalMutationCounter(
  nbObj = NULL,
  min.cn = 1,
  max.cn = 4,
  chromosomes = seq_len(22)
)
```

Arguments

`nbObj`	combined SNV and CNV information as generated by `nbImport`.
`min.cn`	minimal copy number.
`max.cn`	maximal copy number.
`chromosomes`	the chromosomes to be evaluated.

Value

a data table reporting the length of each segment, the number of clonal mutations on all A-allele copies, the number of clonal mutations on all B-allele copies and the total number of clonal mutations (including clonal mutations on a single copy only).

Examples

```
snvs <- system.file("extdata", "NBE15",
    "snvs_NBE15_somatic_snvs_conf_8_to_10.vcf",
    package = "LACHESIS"
)
s_data <- readVCF(vcf = snvs, vcf.source = "dkfz")
aceseq_cn <- system.file("extdata", "NBE15",
    "NBE15_comb_pro_extra2.51_1.txt",
    package = "LACHESIS"
)
c_data <- readCNV(aceseq_cn)
nb <- nbImport(cnv = c_data, snv = s_data, purity = 1, ploidy = 2.51)
cl_muts <- clonalMutationCounter(nb)
```

`estimateClonality` *Assigning clonality status to every SNV*

Description

Assigns clonality status to every SNV based on the variant allele frequency distribution. The function uses maximum a posteriori assignment of single variants to either "subclonal", "clonal", "early clonal" or "late clonal" (if distinguishable). The likelihood is computed according to a binomial distribution; prior probabilities are empirically determined based on the relative SNV burden per clonal state.

Usage

```
estimateClonality(
  nbObj = NULL,
  mrcaObj = NULL,
  ID = NULL,
  purity = NULL,
  driver.file = NULL,
  ref.build = "hg19"
)
```

Arguments

`nbObj`	combined SNV and CNV information as generated by `nbImport`.
`mrcaObj`	clonal SNV counts stratified by copy number as generated by `MRCA`.
`ID`	sample name.
`purity`	tumor cell content.
`driver.file`	optional, path to file with "chrom", "snv_start", "ref", "alt", "gene" column containing known driver SNVs.
`ref.build`	Reference genome. Default `hg19`. Can be `hg18`, `hg19` or `hg38`.

Value

a data.table with per-SNV clonality assignment

Examples

```
# Example using variants associated with specific SBS mutational signatures
# from vcf file
snvs <- system.file("extdata", "NBE15",
    "snvs_NBE15_somatic_snvs_conf_8_to_10.vcf",
    package = "LACHESIS"
)
s_data <- readVCF(vcf = snvs, vcf.source = "dkfz")
aceseq_cn <- system.file("extdata", "NBE15",
    "NBE15_comb_pro_extra2.51_1.txt",
    package = "LACHESIS"
)
c_data <- readCNV(aceseq_cn)
sig.filepath <- system.file("extdata",
    "NBE15_Decomposed_MutationType_Probabilities.txt",
    package = "LACHESIS"
)
nb <- nbImport(
    cnv = c_data, snv = s_data, purity = 1, ploidy = 2.51,
    sig.assign = TRUE, ID = "NBE15", sig.file = sig.filepath
)
cl_muts <- clonalMutationCounter(nb)
norm_muts <- normalizeCounts(cl_muts)
mrca <- MRCA(norm_muts)
estimateClonality(nbObj = nb, mrcaObj = mrca, ID = "NBE15", purity = 1)
```

LACHESIS *Run MRCA density estimation for a set of tumors*

Description

Takes a set of SNV and CNV files as input and outputs per-tumor SNV densities. Input can either be a tab-delimited file containing the sample specifications or vectors giving direct paths to the sample files. CNV file requires columns for the chromosome number, start and end of the segment, and either the total copy number or the number of A- and B-alleles

Usage

```
LACHESIS(
  input.files = NULL,
  ids = NULL,
  vcf.tumor.ids = NULL,
  cnv.files = NULL,
  snv.files = NULL,
```

```
  vcf.source = NULL,
  purity = NULL,
  ploidy = NULL,
  cnv.chr.col = NULL,
  cnv.start.col = NULL,
  cnv.end.col = NULL,
  cnv.A.col = NULL,
  cnv.B.col = NULL,
  cnv.tcn.col = NULL,
  age = NULL,
  OS.time = NULL,
  OS = NULL,
  EFS.time = NULL,
  EFS = NULL,
  output.dir = NULL,
  ignore.XY = TRUE,
  min.cn = 1,
  max.cn = 4,
  merge.tolerance = 10^5,
  min.vaf = 0.01,
  min.depth = 30,
  vcf.info.af = "AF",
  vcf.info.dp = "DP",
  min.seg.size = 10^7,
  fp.mean = 0,
  fp.sd = 0,
  excl.chr = NULL,
  ref.build = "hg19",
  filter.value = "PASS",
  sig.assign = FALSE,
  sig.file = NULL,
  assign.method = "sample",
  sig.select = NULL,
  min.p = NULL,
  driver.file = NULL,
  ...
)
```

Arguments

`input.files`	a tab-delimited sample-specification file, it must contain the sample name, the path to the SNV file, path to CNV file, and optionally purity, ploidy, cnv.chr.col, cnv.start.col, cnv.end.col, cnv.A.col, cnv.B.col, cnv.tcn.col. A template for this spreadsheet can be downloaded from ...
`ids`	vector of sample names, will be ignored if `input.files` is specified.
`vcf.tumor.ids`	vector of sample names as given in the vcf file; will be ignored if `input.files` is specified.

cnv.files	vector of cnv files in same order as ids; should be in tab-delimited format, will be ignored if input.files is specified.
snv.files	vector of snv files in same order as ids; should be in vcf format, will be ignored if input.files is specified.
vcf.source	Tool used for generating VCF file. Can be strelka or mutect or dkfz or sentieon or sage.
purity	vector tumor cell content in same order as ids; will be ignored if input.files is specified.
ploidy	average copy number in the tumor sample in same order as ids; will be ignored if input.files is specified.
cnv.chr.col	column index of chromosome number in cnv.files.
cnv.start.col	column index of first position of the segment.
cnv.end.col	column index of last position of the segment.
cnv.A.col	column index of the number of A alleles. If A and B are not provided, allele configuration are assumed as 1:1 for disomic, 2:1 for trisomic and 3:1 for tetrasomic regions.
cnv.B.col	column index of the number of B alleles. If A and B are not provided, allele configuration are assumed as 1:1 for disomic, 2:1 for trisomic and 3:1 for tetrasomic regions.
cnv.tcn.col	column index of the total copy number. Is computed to A + B if not provided.
age	optional, the age at diagnosis.
OS.time	optional, overall survival time.
OS	optional, overall survival indicator variable.
EFS.time	optional, event-free survival time.
EFS	optional, event-free survival indicator variable.
output.dir	link to directory in which output is to be stored.
ignore.XY	Ignore allosomes. Default TRUE.
min.cn	minimum copy number to be included in the analysis. Default 2.
max.cn	maximum copy number to be included in the analysis. Default 4.
merge.tolerance	the maximum distance below which adjacent segments with equal copy number are merged. Defaults to 10^5 bp.
min.vaf	Remove variants with vcf below threshold. Default 0.01.
min.depth	Minimum required depth for a variant to be considered. Default 30.
vcf.info.af	The string encoding the allele frequency field in the FORMAT column of the .vcf file. Defaults to AFand will be ignored if vcf.source != sentieon.
vcf.info.dp	The string encoding the read depth field in the FORMAT column of the .vcf file. Defaults to DPand will be ignored if vcf.source != sentieon.
min.seg.size	the minimal segment length to be included in the quantification.
fp.mean	optional, the average false positive rate of clonal mutations (e.g., due to incomplete tissue sampling). Defaults to 0.

fp.sd	optional, the standard deviation of the false positive rate of clonal mutations (e.g., due to incomplete tissue sampling). Defaults to 0.
excl.chr	a vector of chromosomes that should be excluded from the quantification. e.g., due to reporter constructs in animal models.
ref.build	Reference genome. Default hg19. Can be hg18, hg19 or hg38.
filter.value	The FILTER column value for variants that passed the filtering, defaults to PASS.
sig.assign	Logical. If TRUE, each variant will be assigned to the most likely mutational signature.
sig.file	The path to the output file from SigProfilerAssignment, typically named "Decomposed_MutationType_Probabilities.txt". If NULL and sig.assign = TRUE, signatures will be assigned using functions from MutationalPatterns.
assign.method	Method to assign signatures: "max" to assign the signature with the highest probability, "sample" to randomly assign based on signature probabilities.
sig.select	A character vector of specific signatures to include in the analysis (e.g., c("SBS1", "SBS5", "SBS40") to focus on clock-like mutational processes).
min.p	Numeric. The minimum probability threshold from the SigAssignment output that a variant must meet to be considered as matching a specific signature.
driver.file	optional, path to file with "chrom", "snv_start", "ref", "alt", "gene" column containing known driver SNVs.
...	further arguments and parameters passed to LACHESIS functions.

Value

a data.table

See Also

MRCA clonalMutationCounter normalizeCounts

Examples

```
# An example file with sample annotations and meta data
input.files <- system.file("extdata", "Sample_template.txt",
    package = "LACHESIS"
)
input.files <- data.table::fread(input.files)

# cnv and snv files for example tumors
nbe11 <- list.files(system.file("extdata/NBE11/", package = "LACHESIS"),
    full.names = TRUE
)
nbe15 <- list.files(system.file("extdata/NBE15/", package = "LACHESIS"),
    full.names = TRUE
)
nbe26 <- list.files(system.file("extdata/NBE26/", package = "LACHESIS"),
    full.names = TRUE
```

```
)

cnv.file <- c(nbe11[1], nbe15[1], nbe26[1])
snv.file <- c(nbe11[2], nbe15[2], nbe26[2])

input.files$cnv.file <- cnv.file
input.files$snv.file <- snv.file

# Make an example input file with paths to cnv and snv file along with other
# meta data
lachesis_input <- tempfile(
    pattern = "lachesis", tmpdir = tempdir(),
    fileext = ".tsv"
)
data.table::fwrite(x = input.files, file = lachesis_input, sep = "\t")

# Example with template file with paths to multiple cnv/snv files as an input
lachesis <- LACHESIS(input.files = lachesis_input)

# Example with a single sample input
strelka_vcf <- system.file("extdata", "strelka2.somatic.snvs.vcf.gz",
    package = "LACHESIS"
)
aceseq_cn <- system.file("extdata",
    "ACESeq/NBE11_comb_pro_extra2.59_0.83.txt",
    package = "LACHESIS"
)
lachesis <- LACHESIS(
    ids = "NBE11", cnv.files = aceseq_cn,
    snv.files = strelka_vcf, vcf.source = "strelka", purity = 0.83,
    ploidy = 2.59
)

# Example with multiple sample and data frame input
nbe11_vcf <- system.file("extdata",
    "NBE11/snvs_NBE11_somatic_snvs_conf_8_to_10.vcf",
    package = "LACHESIS"
)
nbe11_cn <- read.delim(
    system.file("extdata",
        "NBE11/NBE11_comb_pro_extra2.59_0.83.txt",
        package = "LACHESIS"
    ),
    sep = "\t",
    header = TRUE
)
nbe15_vcf <- system.file("extdata",
    "NBE15/snvs_NBE15_somatic_snvs_conf_8_to_10.vcf",
    package = "LACHESIS"
)
nbe15_cn <- read.delim(
    system.file("extdata",
        "NBE15/NBE15_comb_pro_extra2.51_1.txt",
```

```
        package = "LACHESIS"
    ),
    sep = "\t",
    header = TRUE
)
lachesis <- LACHESIS(
    ids = c("NBE11", "NBE15"), cnv.files =
        list(nbe11_cn, nbe15_cn), snv.files = c(nbe11_vcf, nbe15_vcf),
    vcf.source = c("dkfz", "dkfz"), purity = c(0.83, 1), ploidy = c(2.59, 2.51),
    cnv.chr.col = c(1, 1), cnv.start.col = c(2, 2), cnv.end.col = c(3, 3),
    cnv.A.col = c(34, 34), cnv.B.col = c(35, 35), cnv.tcn.col = c(37, 37)
)
```

MRCA *Compute mutation densities at ECA and MRCA*

Description

This function takes the normalized clonal mutation counts obtained with normalizeCountsto estimate mutation densities at MRCA and an earlier common ancestor, ECA.

Usage

```
MRCA(
  normObj = NULL,
  min.seg.size = 10^7,
  fp.mean = 0,
  fp.sd = 0,
  excl.chr = NULL
)
```

Arguments

normObj	normalized clonal SNV counts stratified by copy number as generated by normalizeCounts.
min.seg.size	the minimal segment length to be included in the quantification
fp.mean	optional, the average false positive rate of clonal mutations (e.g., due to incomplete tissue sampling). Defaults to 0.
fp.sd	optional, the standard deviation of the false positive rate of clonal mutations (e.g., due to incomplete tissue sampling). Defaults to 0.
excl.chr	a vector of chromosomes that should be excluded from the quantification. e.g., due to reporter constructs in animal models.

Value

a data table reporting the assignment of individual segments to ECA or MRCA. Mutation densities at ECA and MRCA, and the bootstrapped 95% CIs are stored as attributes. The columns in the data table report the following information:

`chrom`	Chromsoome
`TCN`	Total copy number
`A`	Number of A alleles
`B`	Number of B alleles
`Seglength`	Number of bps with the given copy number configuration on this chromosome
`n_mut_A`	Normalized number of mutations present on all A alleles
`n_mut_B`	Normalized number of mutations present on all B alleles
`n_mut_total_clonal`	Normalized number of mutations per single copy of the segment
`density_A_mean`	Normalized mean density of mutations present on all A alleles (1/Mb)
`density_B_mean`	Normalized mean density of mutations present on all B alleles (1/Mb)
`density_total_mean`	Normalized mean density of mutations per single copy of the segment (1/Mb)
`density_total_lower`	Lower bound (95% CI) of normalized density of mutations per single copy of the segment (1/Mb)
`density_total_upper`	Upper bound (95% CI) of normalized density of mutations per single copy of the segment (1/Mb)
`density_A_lower`	Lower bound (95% CI) of normalized density of mutations on all A alleles of the segment (1/Mb)
`density_A_upper`	Upper bound (95% CI) of normalized density of mutations on all A alleles of the segment (1/Mb)
`density_B_lower`	Lower bound (95% CI) of normalized density of mutations on all B alleles of the segment (1/Mb)
`density_B_upper`	Upper bound (95% CI) of normalized density of mutations on all B alleles of the segment (1/Mb)
`p_total_to_mrca`	Probability that the density of mutations per single copy of the segment agrees with the mutation density at MRCA.
`p_A_to_to_mrca`	Probability that the density of mutations on all A alleles of the segment agrees with the mutation density at MRCA.
`p_B_to_to_mrca`	Probability that the density of mutations on all B alleles of the segment agrees with the mutation density at MRCA.

p_adj_total_to_mrca
: Probability that the density of mutations on all alleles of the segment agrees with the mutation density at MRCA, adjusted for multiple sampling (Holm correction).

p_adj_A_to_mrca
: Probability that the density of mutations on all A alleles of the segment agrees with the mutation density at MRCA, adjusted for multiple sampling (Holm correction).

p_adj_B_to_mrca
: Probability that the density of mutations on all B alleles of the segment agrees with the mutation density at MRCA, adjusted for multiple sampling (Holm correction).

MRCA_qual
: Quality control. PASS, if the density of mutations on single copies agrees with the density at the MRCA.

p_total_to_eca
: Probability that the density of mutations per single copy of the segment agrees with the mutation density at ECA.

p_A_to_to_eca
: Probability that the density of mutations on all A alleles of the segment agrees with the mutation density at ECA.

p_B_to_to_eca
: Probability that the density of mutations on all B alleles of the segment agrees with the mutation density at ECA.

p_adj_total_to_eca
: Probability that the density of mutations on all alleles of the segment agrees with the mutation density at ECA, adjusted for multiple sampling (Holm correction).

p_adj_A_to_eca
: Probability that the density of mutations on all A alleles of the segment agrees with the mutation density at ECA, adjusted for multiple sampling (Holm correction).

p_adj_B_to_eca
: Probability that the density of mutations on all B alleles of the segment agrees with the mutation density at ECA, adjusted for multiple sampling (Holm correction).

A_time
: Time of A allele gain (can be "ECA", "MRCA", "ECA/MRCA" if assignment is unclear, or "not mapped to ECA or MRCA" if density does not agree with either ECA or MRCA).

B_time
: Time of B allele gain (can be "ECA", "MRCA", "ECA/MRCA" if assignment is unclear, or "not mapped to ECA or MRCA" if density does not agree with either ECA or MRCA).

Examples

```
snvs <- system.file("extdata", "NBE15",
    "snvs_NBE15_somatic_snvs_conf_8_to_10.vcf",
    package = "LACHESIS"
)
s_data <- readVCF(vcf = snvs, vcf.source = "dkfz")
aceseq_cn <- system.file("extdata", "NBE15",
    "NBE15_comb_pro_extra2.51_1.txt",
    package = "LACHESIS"
)
```

```
c_data <- readCNV(aceseq_cn)
nb <- nbImport(cnv = c_data, snv = s_data, purity = 1, ploidy = 2.51)
cl_muts <- clonalMutationCounter(nb)
norm_muts <- normalizeCounts(cl_muts)
mrca <- MRCA(norm_muts)
```

---

nbImport *Combine CNVs and SNVs*

Description

Merges CNVs and SNVs into a single data.table. Each variant is assigned to its corresponding copy number segment and status.

Usage

```
nbImport(
  cnv = NULL,
  snv = NULL,
  purity = NULL,
  ploidy = NULL,
  sig.assign = FALSE,
  assign.method = "sample",
  ID = NULL,
  sig.file = NULL,
  sig.select = NULL,
  min.p = NULL,
  ref.build = "hg19",
  cosmic.version = "COSMIC_v3.2",
  ...
)
```

Arguments

cnv	CNV data from readCNV.
snv	SNV data from readVCF.
purity	tumor cell content.
ploidy	average copy number in the tumor sample.
sig.assign	Logical. If TRUE, each variant will be assigned to a mutational signature.
assign.method	Method to assign signatures: max to assign the signature with the highest probability, sample to randomly assign based on signature probabilities.
ID	sample name.
sig.file	The path to the output file from SigProfilerAssignment, typically named "Decomposed_MutationType_Probabilities.txt". If NULL and sig.assign = TRUE, signatures will be assigned using functions from MutationalPatterns.

sig.select	A character vector of specific signatures to include in the analysis (e.g., c("SBS1", "SBS5", "SBS40") to focus on clock-like mutational processes).
min.p	Numeric. The minimum probability threshold from the SigAssignment output that a variant must meet to be considered as matching a specific signature.
ref.build	Reference genome. Default hg19. Can be hg18, hg19 or hg38.
cosmic.version	COSMIC mutational signature reference. Can be "COSMIC", "COSMIC_v3.1", "COSMIC_v3.2"
...	further arguments and parameters passed to other LACHESIS functions.

Value

a data.table

See Also

plotNB

Examples

```
# Example using all variants from vcf file
snvs <- system.file("extdata", "NBE15",
    "snvs_NBE15_somatic_snvs_conf_8_to_10.vcf",
    package = "LACHESIS"
)
s_data <- readVCF(vcf = snvs, vcf.source = "dkfz")
aceseq_cn <- system.file("extdata", "NBE15",
    "NBE15_comb_pro_extra2.51_1.txt",
    package = "LACHESIS"
)
c_data <- readCNV(aceseq_cn)
nb <- nbImport(cnv = c_data, snv = s_data, purity = 1, ploidy = 2.51)

# Example using variants associated with specific SBS mutational
# signatures from vcf file
snvs <- system.file("extdata", "NBE15",
    "snvs_NBE15_somatic_snvs_conf_8_to_10.vcf",
    package = "LACHESIS"
)
s_data <- readVCF(vcf = snvs, vcf.source = "dkfz")
aceseq_cn <- system.file("extdata", "NBE15",
    "NBE15_comb_pro_extra2.51_1.txt",
    package = "LACHESIS"
)
c_data <- readCNV(aceseq_cn)
sig.filepath <- system.file("extdata",
    "NBE15_Decomposed_MutationType_Probabilities.txt",
    package = "LACHESIS"
)
nb <- nbImport(
    cnv = c_data, snv = s_data, purity = 1, ploidy = 2.51,
    sig.assign = TRUE, ID = "NBE15", sig.file = sig.filepath,
```

```
    sig.select = c("SBS1", "SBS5", "SBS40a", "SBS18")
)
nb.2 <- nbImport(
    cnv = c_data, snv = s_data, purity = 1, ploidy = 2.51,
    sig.assign = TRUE, ID = "NBE15",
    sig.select = c("SBS1", "SBS5", "SBS40", "SBS18")
)
```

`normalizeCounts` *Normalize clonal mutation counts*

Description

This function normalizes the clonal mutation counts obtained with `clonalMutationCounter`. The normalized counts correspond to the number of mutations accumulated between conception/gastrulation and MRCA/copy number gain. They can hence be interpreted as "molecular time".

Usage

```
normalizeCounts(countObj = NULL)
```

Arguments

`countObj`	clonal SNV counts stratified by copy number as generated by `clonalMutationCounter`.

Value

a data table reporting the normalized mutation counts and densities per segment, stratified by copy number

Examples

```
snvs <- system.file("extdata", "NBE15",
    "snvs_NBE15_somatic_snvs_conf_8_to_10.vcf",
    package = "LACHESIS"
)
s_data <- readVCF(vcf = snvs, vcf.source = "dkfz")
aceseq_cn <- system.file("extdata", "NBE15",
    "NBE15_comb_pro_extra2.51_1.txt",
    package = "LACHESIS"
)
c_data <- readCNV(aceseq_cn)
nb <- nbImport(cnv = c_data, snv = s_data, purity = 1, ploidy = 2.51)
cl_muts <- clonalMutationCounter(nb)
norm_muts <- normalizeCounts(cl_muts)
```

plotClinicalCorrelations *Correlate SNV density at ECA/MRCA with clinical parameters such as age, OS, etc.*

Description

Takes SNV densities as computed by LACHESIS as input and correlates them with clinical data such as age at diagnosis, survival data etc.

Usage

```
plotClinicalCorrelations(
  lachesis = NULL,
  clin.par = "Age",
  clin.suppress.outliers = FALSE,
  clin.log.densities = FALSE,
  lach.col.multi = "#176A02",
  lach.col.zero = "#4FB12B",
  output.file = NULL
)
```

Arguments

lachesis	output generated from LACHESIS.
clin.par	the clinical parameter used for correlation. Default Age.
clin.suppress.outliers	shall outliers (defined as the 2.5% tumors with lowest and highest densities) be plotted? Default FALSE.
clin.log.densities	plot logarithmic densities. Default FALSE.
lach.col.multi	optional, color for multi-copy SSNV densities.
lach.col.zero	optional, color for single-copy SSNV densities.
output.file	optional; file path to output.

Value

graph with SNV density at ECA/ MRCA compared to clinical parameters

Examples

```
# An example file with sample annotations and meta data
input.files <- system.file("extdata", "Sample_template.txt",
    package = "LACHESIS"
)
input.files <- data.table::fread(input.files)
```

```
# cnv and snv files for example tumors
nbe11 <- list.files(system.file("extdata/NBE11/", package = "LACHESIS"),
    full.names = TRUE
)
nbe15 <- list.files(system.file("extdata/NBE15/", package = "LACHESIS"),
    full.names = TRUE
)
nbe26 <- list.files(system.file("extdata/NBE26/", package = "LACHESIS"),
    full.names = TRUE
)

cnv.file <- c(nbe11[1], nbe15[1], nbe26[1])
snv.file <- c(nbe11[2], nbe15[2], nbe26[2])

input.files$cnv.file <- cnv.file
input.files$snv.file <- snv.file

# Make an example input file with paths to cnv and snv file along with other
# meta data
lachesis_input <- tempfile(
    pattern = "lachesis", tmpdir = tempdir(),
    fileext = ".tsv"
)
data.table::fwrite(x = input.files, file = lachesis_input, sep = "\t")

# Example with template file with paths to multiple cnv/snv files as an input
lachesis <- LACHESIS(input.files = lachesis_input)
plotClinicalCorrelations(lachesis)
```

`plotClonality` *Plotting assigned clonality status for every SNV by chromosome*

Description

Visualizes results from `estimateClonality`.

Usage

```
plotClonality(
  snvClonality = snvClonality,
  nbObj = NULL,
  sig.assign = FALSE,
  output.file = NULL,
  ...
)
```

Arguments

snvClonality	output generated from estimateClonality.
nbObj	output generated from nbImport.
sig.assign	Logical. If TRUE, clonality status distribution will be plotted for each SBS signature.
output.file	optional, will save the mutational signatures stratified by Clonality.
...	further arguments and parameters passed to other LACHESIS functions.

Value

graphs with clonality status of SNVs per chromosome and if specified, stratified by signature

Examples

```
# Example using variants associated with specific SBS mutational signatures
# from vcf file
snvs <- system.file("extdata", "NBE15",
    "snvs_NBE15_somatic_snvs_conf_8_to_10.vcf",
    package = "LACHESIS"
)
s_data <- readVCF(vcf = snvs, vcf.source = "dkfz")
aceseq_cn <- system.file("extdata", "NBE15",
    "NBE15_comb_pro_extra2.51_1.txt",
    package = "LACHESIS"
)
c_data <- readCNV(aceseq_cn)
sig.filepath <- system.file("extdata",
    "NBE15_Decomposed_MutationType_Probabilities.txt",
    package = "LACHESIS"
)
nb <- nbImport(
    cnv = c_data, snv = s_data, purity = 1, ploidy = 2.51,
    sig.assign = TRUE, ID = "NBE15", sig.file = sig.filepath
)
cl_muts <- clonalMutationCounter(nb)
norm_muts <- normalizeCounts(cl_muts)
mrca <- MRCA(norm_muts)
snvClonality <- estimateClonality(
    nbObj = nb, mrcaObj = mrca,
    ID = "NBE15", purity = 1
)
plotClonality(snvClonality, nbObj = nb, sig.assign = TRUE)
```

plotDiseaseTrajectories *Plot the disease trajectories in a cohort.*

Description

Takes SNV densities as computed by LACHESIS as input and plots for each case the estimated age at ECA, MRCA and the age at diagnosis.

Usage

```
plotDiseaseTrajectories(
  lachesis = NULL,
  mut.snv.rate = 3.2,
  lach.col.eca = "#176A02",
  lach.col.mrca = "#4FB12B",
  corr.time.scale = 1,
  output.file = NULL
)
```

Arguments

lachesis	output generated from LACHESIS.
mut.snv.rate	optional; rate of accumulated SNVs per day in a diploid genome (i.e. 3.2 SNVs/day in neuroblastoma)
lach.col.eca	optional, color for ECA.
lach.col.mrca	optional, color for MRCA.
corr.time.scale	numeric value by which survival time is to be divided to convert into months (e.g., 30 for converting days into months), defaults to 1.
output.file	optional; file path to output.

Value

graph with SNV densities and estimated times at ECA/ MRCA and diagnosis.

Examples

```
# An example file with sample annotations and meta data
input.files <- system.file("extdata", "Sample_template.txt",
    package = "LACHESIS"
)
input.files <- data.table::fread(input.files)

# cnv and snv files for example tumors
nbe11 <- list.files(system.file("extdata/NBE11/", package = "LACHESIS"),
    full.names = TRUE
```

```
)
nbe15 <- list.files(system.file("extdata/NBE15/", package = "LACHESIS"),
    full.names = TRUE
)
nbe26 <- list.files(system.file("extdata/NBE26/", package = "LACHESIS"),
    full.names = TRUE
)

cnv.file <- c(nbe11[1], nbe15[1], nbe26[1])
snv.file <- c(nbe11[2], nbe15[2], nbe26[2])

input.files$cnv.file <- cnv.file
input.files$snv.file <- snv.file

# Make an example input file with paths to cnv and snv file along with other
# meta data
lachesis_input <- tempfile(
    pattern = "lachesis", tmpdir = tempdir(),
    fileext = ".tsv"
)
data.table::fwrite(x = input.files, file = lachesis_input, sep = "\t")

# Example with template file with paths to multiple cnv/snv files as an input
lachesis <- LACHESIS(input.files = lachesis_input)
plotDiseaseTrajectories(lachesis, corr.time.scale = 31)
```

`plotLachesis` *Plot SNV densities at ECA and MRCA*

Description

Visualizes results from LACHESIS. Top plot, histograms of mean mutation densities; bottom plots, cumulative distribution of mean mutation densities with 95% confidence intervals.

Usage

```
plotLachesis(
  lachesis = NULL,
  lach.suppress.outliers = FALSE,
  lach.log.densities = FALSE,
  lach.col.multi = "#176A02",
  lach.border = NULL,
  binwidth = NULL,
  lach.col.zero = "#4FB12B",
  output.file = NULL,
  ...
)
```

Arguments

lachesis	output generated from LACHESIS
lach.suppress.outliers	whether outliers (defined as the 2.5% tumors with lowest and highest densities) are to be plot. Default TRUE.
lach.log.densities	plot logarithmic densities. Default FALSE.
lach.col.multi	optional, bar color for multi-copy SSNV densities.
lach.border	optional, border color for the bars.
binwidth	optional, the binwidth in the histogram.
lach.col.zero	optional, bar color for single-copy SSNV densities.
output.file	optional, the file to which the plot will be stored.
...	further arguments and parameters passed to other LACHESIS functions.

Value

graph with cohort overview of SNV densities at ECA/ MRCA

Examples

```
# An example file with sample annotations and meta data
input.files <- system.file("extdata", "Sample_template.txt",
    package = "LACHESIS"
)
input.files <- data.table::fread(input.files)

# cnv and snv files for example tumors
nbe11 <- list.files(system.file("extdata/NBE11/", package = "LACHESIS"),
    full.names = TRUE
)
nbe15 <- list.files(system.file("extdata/NBE15/", package = "LACHESIS"),
    full.names = TRUE
)
nbe26 <- list.files(system.file("extdata/NBE26/", package = "LACHESIS"),
    full.names = TRUE
)

cnv.file <- c(nbe11[1], nbe15[1], nbe26[1])
snv.file <- c(nbe11[2], nbe15[2], nbe26[2])

input.files$cnv.file <- cnv.file
input.files$snv.file <- snv.file

# Make an example input file with paths to cnv and snv file along with other
# meta data
lachesis_input <- tempfile(
    pattern = "lachesis", tmpdir = tempdir(),
    fileext = ".tsv"
)
```

```
data.table::fwrite(x = input.files, file = lachesis_input, sep = "\t")

# Example with template file with paths to multiple cnv/snv files as an input
lachesis <- LACHESIS(input.files = lachesis_input)
plotLachesis(lachesis)
```

`plotMutationDensities` *Plot normalized mutation density at copy number gain and MRCA per segment*

Description

Visualizes results from MRCA. Top plot, histograms of mean mutation densities; bottom plots, timeline of early tumor evolution, showing mutation densities (mean and 95% CI) of individual chromosomal gains and mutation densities at ECA and MRCA.

Usage

```
plotMutationDensities(
  mrcaObj = NULL,
  samp.name = NULL,
  min.seg.size = 10^7,
  ref.build = "hg19",
  mut.col.zero = "#4FB12B",
  mut.col.multi = "#176A02",
  mut.border = NULL,
  mut.show.density = TRUE,
  mut.breaks = NULL,
  mut.xaxis = NULL,
  mut.show.realtime = FALSE,
  mut.snv.rate = 3.2,
  output.file = NULL,
  ...
)
```

Arguments

`mrcaObj`	output generated from MRCA
`samp.name`	sample name, optional
`min.seg.size`	minimal segment size to plot
`ref.build`	Reference genome. Default `hg19`. Can be `hg18`, `hg19` or `hg38`.
`mut.col.zero`	optional, the bar color for densities of mutations present on single copies.
`mut.col.multi`	optional, the bar color for densities of mutations present on multiple copies.
`mut.border`	optional, the line color

`mut.show.density`
: optional; if `TRUE`, the density distribution of mutation densities on single copies will be shown in the histogram of mutation densities on multiple copies.

`mut.breaks`
: optional; the number of bins in the histogram.

`mut.xaxis`
: optional; cutoff value for x-axis in evolutionary timeline plot in SNVs/Mb

`mut.show.realtime`
: logical; if `TRUE`, displays weeks post-conception on the evolutionary timeline.

`mut.snv.rate`
: optional; rate of accumulated SNVs per day in a diploid genome (i.e. 3.2 SNVs/day in neuroblastoma)

`output.file`
: optional; will save the plot.

`...`
: further arguments and parameters passed to other LACHESIS functions.

Value

graphs with mutation densitiy at ECA and MRCA as well as evolutionary timeline plot

Examples

```
snvs <- system.file("extdata", "NBE15",
    "snvs_NBE15_somatic_snvs_conf_8_to_10.vcf",
    package = "LACHESIS"
)
s_data <- readVCF(vcf = snvs, vcf.source = "dkfz")
aceseq_cn <- system.file("extdata", "NBE15",
    "NBE15_comb_pro_extra2.51_1.txt",
    package = "LACHESIS"
)
c_data <- readCNV(aceseq_cn)
nb <- nbImport(cnv = c_data, snv = s_data, purity = 1, ploidy = 2.51)
cl_muts <- clonalMutationCounter(nb)
norm_muts <- normalizeCounts(cl_muts)
mrca <- MRCA(norm_muts)
plotMutationDensities(mrca)
```

---

plotNB *Plot VAF distribution per copy number*

Description

Visualizes results from `nbImport`. Top plot, measured copy numbers along the genome; bottom plots, VAF histograms of SNVs stratified by copy number and minor/major allele count.

Usage

```
plotNB(
  nb = NULL,
  snvClonality = NULL,
  ref.build = "hg19",
  min.cn = 2,
  max.cn = 4,
  nb.col.abline = "gray70",
  nb.col.cn.2 = "#7f8c8d",
  nb.col.cn = "#16a085",
  nb.col.hist = "#34495e",
  nb.border = NA,
  nb.breaks = 100,
  samp.name = NULL,
  output.file = NULL,
  sig.show = FALSE,
  ...
)
```

Arguments

nb	output generated from nbImport.
snvClonality	output generated from estimateClonality.
ref.build	Reference genome. Default hg19. Can be hg18, hg19 or hg38.
min.cn	maximum copy number to be included in the plotting. Defaults to 2.
max.cn	maximum copy number to be included in the plotting. Defaults to 4.
nb.col.abline	optional, the color code for the lines depicting clonality in the VAF histograms.
nb.col.cn.2	optional, the color code for tcn = 2 in the CNV plot.
nb.col.cn	optional, the color code for other copy numbers in the CNV plot.
nb.col.hist	optional, the color code for bars in the VAF histograms.
nb.border	optional, the line color in the VAF histograms.
nb.breaks	optional, the number of bins in the histograms.
samp.name	Sample name. Optional. Default NULL
output.file	optional, will save the plot.
sig.show	plot stratified VAF histogram with assigned mutational signatures.
...	further arguments and parameters passed to other LACHESIS functions.

Value

copy number plot, VAF histograms stratified by copynumber and clonality; if specified, VAF histograms stratified by copynumber and signature

Examples

```
# Example using all variants from vcf file
snvs <- system.file("extdata", "NBE15",
    "snvs_NBE15_somatic_snvs_conf_8_to_10.vcf",
    package = "LACHESIS"
)
s_data <- readVCF(vcf = snvs, vcf.source = "dkfz")
aceseq_cn <- system.file("extdata", "NBE15",
    "NBE15_comb_pro_extra2.51_1.txt",
    package = "LACHESIS"
)
c_data <- readCNV(aceseq_cn)
nb <- nbImport(cnv = c_data, snv = s_data, purity = 1, ploidy = 2.51)
cl_muts <- clonalMutationCounter(nb)
norm_muts <- normalizeCounts(cl_muts)
mrca <- MRCA(norm_muts)
snvClonality <- estimateClonality(
    nbObj = nb, mrcaObj = mrca, ID = "NBE15",
    purity = 1
)
plotNB(nb = nb, snvClonality = snvClonality)

# Example using variants associated with specific SBS mutational
# signatures from vcf file
snvs <- system.file("extdata", "NBE15",
    "snvs_NBE15_somatic_snvs_conf_8_to_10.vcf",
    package = "LACHESIS"
)
s_data <- readVCF(vcf = snvs, vcf.source = "dkfz")
aceseq_cn <- system.file("extdata", "NBE15",
    "NBE15_comb_pro_extra2.51_1.txt",
    package = "LACHESIS"
)
c_data <- readCNV(aceseq_cn)
sig.filepath <- system.file("extdata",
    "NBE15_Decomposed_MutationType_Probabilities.txt",
    package = "LACHESIS"
)
nb <- nbImport(
    cnv = c_data, snv = s_data, purity = 1, ploidy = 2.51,
    sig.assign = TRUE, ID = "NBE15", sig.file = sig.filepath
)
cl_muts <- clonalMutationCounter(nb)
norm_muts <- normalizeCounts(cl_muts)
mrca <- MRCA(norm_muts)
snvClonality <- estimateClonality(
    nbObj = nb, mrcaObj = mrca, ID = "NBE15",
    purity = 1
)
plotNB(nb = nb, snvClonality = snvClonality, sig.show = TRUE)
```

plotSurvival *Correlate SNV density timing at MRCA with Survival*

Description

Takes SNV density timing as computed by LACHESIS as input and compares survival between tumors with high and low SNV densities

Usage

```
plotSurvival(
  lachesis = NULL,
  mrca.cutpoint = NULL,
  output.dir = NULL,
  surv.time = "OS.time",
  surv.event = "OS",
  surv.palette = c("dodgerblue", "dodgerblue4"),
  surv.time.breaks = NULL,
  surv.time.scale = 1,
  surv.title = "Survival Probability",
  surv.ylab = "Survival",
  surv.xlab = "Time"
)
```

Arguments

lachesis	output generated from LACHESIS.
mrca.cutpoint	optional, MRCA density value to be used for survival stratification, will be computationally inferred to maximize survival differences if not specified by user.
output.dir	link to directory in which output is to be stored.
surv.time	column name containing survival time; defaults to OS.time.
surv.event	column name containing event; defaults to OS.
surv.palette	color palette to be used. Allowed values include "hue" for the default hue color scale; "grey" for grey color palettes; brewer palettes e.g. "RdBu", "Blues", ...; or custom color palette e.g. c("blue", "red").
surv.time.breaks	numeric value controlling time axis breaks; defaults to NULL.
surv.time.scale	numeric value by which survival time is to be divided (e.g., 365 for converting days into years, 30 for months), defaults to 1.
surv.title	main title.
surv.ylab	y-axis label, defaults to Survival.
surv.xlab	x-axis label, defaults to Time.

Value

survival graphs

Examples

```
# An example file with sample annotations and meta data
input.files <- system.file("extdata", "Sample_template.txt",
    package =
        "LACHESIS"
)
input.files <- data.table::fread(input.files)

# cnv and snv files for example tumors
nbe11 <- list.files(system.file("extdata/NBE11/", package = "LACHESIS"),
    full.names = TRUE
)
nbe15 <- list.files(system.file("extdata/NBE15/", package = "LACHESIS"),
    full.names = TRUE
)
nbe26 <- list.files(system.file("extdata/NBE26/", package = "LACHESIS"),
    full.names = TRUE
)

cnv.file <- c(nbe11[1], nbe15[1], nbe26[1])
snv.file <- c(nbe11[2], nbe15[2], nbe26[2])

input.files$cnv.file <- cnv.file
input.files$snv.file <- snv.file

# Make an example input file with paths to cnv and snv file along with other
# meta data
lachesis_input <- tempfile(
    pattern = "lachesis", tmpdir = tempdir(),
    fileext = ".tsv"
)
data.table::fwrite(x = input.files, file = lachesis_input, sep = "\t")

# Example with template file with paths to multiple cnv/snv files as an input
lachesis <- LACHESIS(input.files = lachesis_input)
plotSurvival(lachesis,
    surv.time = "EFS.time", surv.event = "EFS",
    mrca.cutpoint = 0.05
)
```

---

plotVAFdistr *Plot histogram of VAF distribution*

Description

Plot frequency distribution of variant allele frequencies

Usage

```
plotVAFdistr(
  vaf = NULL,
  vaf.interval = 0.05,
  t_sample = NULL,
  vaf.show.counts = FALSE,
  vaf.show.density = TRUE,
  vaf.col = "#34495e",
  vaf.border = "#bdc3c7",
  srtcounts = 45,
  output.file = NULL,
  ...
)
```

Arguments

vaf	output produced by readVCF
vaf.interval	Interval size. Default 0.05
t_sample	Sample name for tumor. Used for plot title. Default NULL
vaf.show.counts	Show counter per break on the histogram. Default FALSE
vaf.show.density	Show additional inset plot of density. Default TRUE
vaf.col	Color to be used to fill the bars, default "#34495e"
vaf.border	Border color, default "#bdc3c7"
srtcounts	Text angle if vaf.show.counts is TRUE. Default 45
output.file	Optional, will save the plot.
...	further arguments and parameters passed to other LACHESIS functions.

Value

VAF histogram

Examples

```
strelka_vcf <- system.file("extdata", "strelka2.somatic.snvs.vcf.gz",
    package = "LACHESIS"
)
s_data <- readVCF(
    vcf = strelka_vcf, vcf.source = "strelka",
    ignore.XY = FALSE
)
plotVAFdistr(s_data)
```

readCNV *Converts a user-specified bed-file with copy number information into a standardized format that can be used as input for downstream analysis*

Description

Convert a user-specified bed-file with copy number information into a standardized format. Perform various quality checks on the file input and return the clean and standardized data-frame. If column identifiers for chromosomal positions and allele-specific copy number information are not provided, the function attempts to identify these columns based on standard nomenclature. If total copy number information is provided but allele-specific information is missing, the function assumes that the number of B alleles is the rounded off of half the total copy number.

Usage

```
readCNV(
  cn.info = NULL,
  chr.col = NULL,
  start.col = NULL,
  end.col = NULL,
  A.col = NULL,
  B.col = NULL,
  tcn.col = NULL,
  merge.tolerance = 10^5,
  ignore.XY = TRUE,
  max.cn = 4,
  tumor.id = NULL
)
```

Arguments

cn.info	Path to the copy number information. Requires columns for the chromosome number, start and end of the segment, and either the total copy number or the number of A- and B-alleles
chr.col	column index of chromosome number
start.col	column index of first position of the segment
end.col	column index of last position of the segment
A.col	column index of the number of A alleles. If A and B are not provided, allele configuration are assumed as 1:1 for disomic, 2:1 for trisomic and 3:1 for tetrasomic regions.
B.col	column index of the number of B alleles. If A and B are not provided, allele configuration are assumed as 1:1 for disomic, 2:1 for trisomic and 3:1 for tetrasomic regions.
tcn.col	column index of the total copy number. Is computed to A + B if not provided.

merge.tolerance	the maximum distance below which adjacent segments with equal copy number are merged. Defaults to 10^5 bp.
ignore.XY	Ignore allosomes. Default TRUE
max.cn	maximum copy number to be included in the analysis. Defaults to 4.
tumor.id	Tumor ID, optional.

Value

A standardized data frame with copy number information per segment. readCNV()

Author(s)

Verena Körber

Examples

```
aceseq_cn <- system.file("extdata",
    "ACESeq/NBE11_comb_pro_extra2.59_0.83.txt",
    package = "LACHESIS"
)
cn_data <- readCNV(aceseq_cn)
ascat_cn <- system.file("extdata",
    "ASCAT/S98.segments.txt",
    package = "LACHESIS"
)
cn_data <- readCNV(ascat_cn)
purple_cn <- system.file("extdata",
    "PURPLE/NB-S-599-T.purple.cnv.somatic.tsv",
    package = "LACHESIS"
)
cn_data <- LACHESIS::readCNV(purple_cn)
```

---

readVCF *Import a VCF file and extract read count*

Description

Import a VCF file and extract read-count and variant allele frequencies. Currently VCF files generated by mutect2, strelka2, dkfz, sentieon and sage are supported.

Usage

```
readVCF(
  vcf = NULL,
  ignore.XY = TRUE,
  vcf.source = "strelka",
  min.vaf = 0.01,
```

```
  min.depth = 30,
  t.sample = NULL,
  info.af = "AF",
  info.dp = "DP",
  filter.value = "PASS",
  filter.biallelic = TRUE,
  filter.indels = TRUE,
  ...
)
```

Arguments

vcf	Input indexed VCF file.
ignore.XY	Ignore allosomes. Default TRUE
vcf.source	Tool used for generating VCF file. Can be strelka or mutect or dkfz or sentieon or sage
min.vaf	Remove variants with vcf below threshold. Default 0.01
min.depth	Minimum required depth for a variant to be considered. Default 30.
t.sample	Sample name for tumor. Must be same as in VCF. Strelka hard codes tumor sample name to "TUMOR"
info.af	The string encoding the allele frequency field in the FORMAT column. Defaults to AFand will be ignored if vcf.source != sentieon.
info.dp	The string encoding the read depth field in the FORMAT column. Defaults to DPand will be ignored if vcf.source != sentieon.
filter.value	The FILTER column value for variants that passed the filtering, defaults to PASS
filter.biallelic	Remove biallelic variants. Default TRUE
filter.indels	Remove indels. Default TRUE
...	further arguments and parameters passed to other LACHESIS functions.

Value

a data.table with chrom, pos, ref, alt, t_ref_count, t_alt_count, t_depth, t_vaf

Examples

```
mutect_vcf <- system.file("extdata", "mutect.somatic.vcf.gz",
    package = "LACHESIS"
)
m_data <- readVCF(
    vcf = mutect_vcf, vcf.source = "mutect",
    filter.value = "."
)
strelka_vcf <- system.file("extdata", "strelka2.somatic.snvs.vcf.gz",
    package = "LACHESIS"
)
s_data <- readVCF(vcf = strelka_vcf, vcf.source = "strelka")
```

```
dkfz_vcf <- system.file("extdata", "NBE15",
    "snvs_NBE15_somatic_snvs_conf_8_to_10.vcf",
    package = "LACHESIS"
)
d_data <- readVCF(vcf = dkfz_vcf, vcf.source = "dkfz")
```

Index