Package 'ASCAT'

October 30, 2024

Type Package

Title Allele-Specific Copy Number Analysis of Tumors

Version 3.2.0

Date 2024-30-10

Author Peter Van Loo [aut, cre], Gro Nielsen [ctb], Jiqiu Cheng [ctb], Stefan Dentro [ctb], Edith Ross [ctb], Kerstin Haase [ctb], Jonas Demeulemeester [ctb], Maxime Tarabichi [ctb], Tom Lesluyes [ctb], Zhihui Zhang [ctb]

Description
R package of ASCAT as published in <https://pubmed.ncbi.nlm.nih.gov/20837533>.

Depends R (>= 2.13.0)

Imports data.table,
doParallel,
foreach,
GenomicRanges,
graphics,
grDevices,
IRanges,
RColorBrewer,
S4Vectors,
splines,
stats,
utils

Suggests ggplot2,
knitr,
plyr,
rmarkdown

License GPL-3

Encoding UTF-8

VignetteBuilder knitr

LazyLoad yes

RoxygenNote 7.3.2

R topics documented:

`ascat.asmultipcf` *Allele-specific segmentation of multiple samples*

Description

This segmentation function should only be used if part of the breakpoints are expected to be shared between samples, e.g. due to a common ancestry.

Usage

```
ascat.asmultipcf(
  ASCATobj,
  ascat.gg = NULL,
  penalty = 70,
  out.dir = ".",
  wsample = NULL,
  selectAlg = "exact",
  refine = TRUE,
  seed = as.integer(Sys.time())
)
```

Arguments

`ASCATobj`	an ASCAT object
`ascat.gg`	germline genotypes (NULL if germline data is available)
`penalty`	penalty of introducing an additional ASPCF breakpoint (expert parameter, don't adapt unless you know what you are doing)
`out.dir`	directory in which output files will be written. Can be set to NA to not write PCFed files.

wsample	Vector of length length(ASCATobj$samples). Can be used to assign different weights to samples, for example to account for differences in sequencing quality. (Default = NULL)
selectAlg	Set to "exact" to run the exact algorithm, or "fast" to run the heuristic algorithm. (Default = "exact")
refine	Logical. Should breakpoints be refined on a per sample base? Otherwise each breakpoint is assumed to be present in each sample. (Default = TRUE)
seed	A seed to be set when subsampling SNPs for X in males (optional, default=as.integer(Sys.time())).

Details

This function saves the results in in [sample].LogR.PCFed.txt and [sample].BAF.PCFed.txt

Value

output: ascat data structure containing:
1. Tumor_LogR data matrix
2. Tumor_BAF data matrix
3. Tumor_LogR_segmented: matrix of LogR segmented values
4. Tumor_BAF_segmented: list of BAF segmented values; each element in the list is a matrix containing the segmented values for one sample (only for probes that are germline homozygous)
5. Germline_LogR data matrix
6. Germline_BAF data matrix
7. SNPpos: position of all SNPs
8. ch: a list containing vectors with the indices for each chromosome (e.g. Tumor_LogR[ch[[13]],] will output the Tumor_LogR data of chromosome 13
9. chr: a list containing vectors with the indices for each distinct part that can be segmented separately (e.g. chromosome arm, stretch of DNA between gaps in the array design)

`ascat.aspcf` *ascat.aspcf*

Description

run ASPCF segmentation

Usage

```
ascat.aspcf(
  ASCATobj,
  selectsamples = 1:length(ASCATobj$samples),
  ascat.gg = NULL,
  penalty = 70,
  out.dir = ".",
  out.prefix = "",
  seed = as.integer(Sys.time())
)
```

Arguments

ASCATobj	an ASCAT object
selectsamples	a vector containing the sample number(s) to PCF. Default = all
ascat.gg	germline genotypes (NULL if germline data is available)
penalty	penalty of introducing an additional ASPCF breakpoint (expert parameter, don't adapt unless you know what you're doing)
out.dir	directory in which output files will be written. Can be set to NA to not write PCFed files.
out.prefix	prefix for output file names
seed	A seed to be set when subsampling SNPs for X in males (optional, default=as.integer(Sys.time())).

Details

This function can be easily parallelised by controlling the selectsamples parameter
it saves the results in LogR_PCFed[sample]_[segment].txt and BAF_PCFed[sample]_[segment].txt

Value

output: ascat data structure containing:
1. Tumor_LogR data matrix
2. Tumor_BAF data matrix
3. Tumor_LogR_segmented: matrix of LogR segmented values
4. Tumor_BAF_segmented: list of BAF segmented values; each element in the list is a matrix containing the segmented values for one sample (only for probes that are not germline homozygous)
5. Germline_LogR data matrix
6. Germline_BAF data matrix
7. SNPpos: position of all SNPs
8. ch: a list containing vectors with the indices for each chromosome (e.g. Tumor_LogR[ch[[13]],] will output the Tumor_LogR data of chromosome 13
9. chr: a list containing vectors with the indices for each distinct part that can be segmented separately (e.g. chromosome arm, stretch of DNA between gaps in the array design)

---

ascat.correctLogR *ascat.correctLogR*

Description

Corrects logR of the tumour sample(s) with genomic GC content (replication timing is optional)

Usage

```
ascat.correctLogR(ASCATobj, GCcontentfile = NULL, replictimingfile = NULL)
```

Arguments

ASCATobj	an ASCAT object
GCcontentfile	File containing the GC content around every SNP for increasing window sizes
replictimingfile	File containing replication timing at every SNP for various cell lines (optional)

Details

Note that probes not present in the GC content file will be lost from the results

Value

ASCAT object with corrected tumour logR

`ascat.GCcorrect` *ascat.GCcorrect*

Description

Function kept for backward compatibility, please use ascat.correctLogR instead

Usage

```
ascat.GCcorrect(ASCATobj, GCcontentfile = NULL)
```

Arguments

`ASCATobj`	an ASCAT object
`GCcontentfile`	File containing the GC content around every SNP for increasing window sizes

---

`ascat.getAlleleCounts` *Obtain allele counts for a given set of loci through external program alleleCounter.*

Description

Obtain allele counts for a given set of loci through external program alleleCounter.

Usage

```
ascat.getAlleleCounts(
  seq.file,
  output.file,
  loci.file,
  min.base.qual = 20,
  min.map.qual = 35,
  allelecounter.exe = "alleleCounter",
  additional_allelecounter_flags = NA
)
```

Arguments

`seq.file`	A BAM/CRAM alignment file on which the counter should be run.
`output.file`	The file where output should go.
`loci.file`	A file with SNP loci.
`min.base.qual`	The minimum base quality required for it to be counted (optional, default=20).
`min.map.qual`	The minimum mapping quality required for it to be counted (optional, default=35).
`allelecounter.exe`	A pointer to where the alleleCounter executable can be found (optional, default points to $PATH).
`additional_allelecounter_flags`	Additional flags passed on to alleleCounter, e.g., -r <FASTA> for parsing CRAMs (optional, default=NA).

Author(s)

sd11, tl, jd

`ascat.getBAFsAndLogRs` *Obtain BAF and LogR from the allele counts.*

Description

Obtain BAF and LogR from the allele counts.

Usage

```
ascat.getBAFsAndLogRs(
  samplename,
  tumourAlleleCountsFile.prefix,
  normalAlleleCountsFile.prefix,
  tumourLogR_file,
  tumourBAF_file,
  normalLogR_file,
  normalBAF_file,
  alleles.prefix,
  gender,
  genomeVersion,
  chrom_names = c(1:22, "X"),
  minCounts = 20,
  BED_file = NA,
  probloci_file = NA,
  tumour_only_mode = FALSE,
  loci_binsize = 1,
  seed = as.integer(Sys.time())
)
```

Arguments

samplename	String, name of the sample.
tumourAlleleCountsFile.prefix	Prefix of the allele counts files for the tumour (e.g. "Tumour_alleleFrequencies_chr").
normalAlleleCountsFile.prefix	Prefix of the allele counts files for the normal (e.g. "Normal_alleleFrequencies_chr").
tumourLogR_file	File where LogR from the tumour will be written.
tumourBAF_file	File where BAF from the tumour will be written.
normalLogR_file	File where LogR from the normal will be written.
normalBAF_file	File where BAF from the normal will be written.
alleles.prefix	Prefix path to the allele data (e.g. "G1000_alleles_chr")
gender	Gender information, either "XX" (=female) or "XY" (=male).
genomeVersion	Genome version, available options are "hg19", "hg38" or "CHM13".
chrom_names	A vector with allowed chromosome names (optional, default=c(1:22, "X")). Do not set it to paste0("chr", c(1:22, "X")) if data is "chr"-based.
minCounts	Minimum depth, in normal samples, required for a SNP to be considered (optional, default=20).
BED_file	A BED file for only looking at SNPs within specific intervals (optional, default=NA).
probloci_file	A file (chromosome <tab> position; no header) containing specific loci to ignore (optional, default=NA).
tumour_only_mode	Should the BAF and LogR be computed from tumour-only (optional, default = FALSE)
loci_binsize	Size of the bins for long-read sequencing data (optional, default = 1)
seed	A seed to be set when randomising the alleles (optional, default=as.integer(Sys.time())).

Author(s)

dw9, sd11, tl, jd, rc

---

ascat.loadData *ascat.loadData*

Description

Function to read in SNP array data

Usage

```
ascat.loadData(
  Tumor_LogR_file,
  Tumor_BAF_file,
  Germline_LogR_file = NULL,
  Germline_BAF_file = NULL,
  chrs = c(1:22, "X", "Y"),
  gender = NULL,
  sexchromosomes = c("X", "Y"),
  genomeVersion = NULL,
  isTargetedSeq = FALSE
)
```

Arguments

Tumor_LogR_file	file containing logR of tumour sample(s)
Tumor_BAF_file	file containing BAF of tumour sample(s)
Germline_LogR_file	file containing logR of germline sample(s), NULL
Germline_BAF_file	file containing BAF of germline sample(s), NULL
chrs	a vector containing the names for the chromosomes (e.g. c(1:22, "X"))
gender	a vector of gender for each cases ("XX" or "XY"). Default = all female ("XX")
sexchromosomes	a vector containing the names for the sex chromosomes. Default = c("X", "Y")
genomeVersion	a string ('hg19', 'hg38' or 'CHM13') so nonPAR coordinates on X can be stored, NULL
isTargetedSeq	a boolean indicating whether data come from a targeted sequencing experiment. Default = F

Details

germline data files can be NULL - in that case these are not read in

Value

ascat data structure containing:
1. Tumor_LogR data matrix
2. Tumor_BAF data matrix
3. Tumor_LogR_segmented: placeholder, NULL
4. Tumor_BAF_segmented: placeholder, NULL
5. Germline_LogR data matrix
6. Germline_BAF data matrix
7. SNPpos: position of all SNPs
8. ch: a list containing vectors with the indices for each chromosome (e.g. Tumor_LogR[ch[[13]],] will output the Tumor_LogR data of chromosome 13
9. chr: a list containing vectors with the indices for each distinct part that can be segmented separately (e.g. chromosome arm, stretch of DNA between gaps in the array design)
10. chrs: a vector containing chromosome names
11. samples: a vector containing sample name(s)

12. gender: a vector of gender for each cases ("XX" or "XY"). Default = NULL: all female ("XX")
13. sexchromosomes: a vector containingg names of sex chromosomes
14. X_nonPAR: a vector of two values (start and stop) to define where the nonPAR region is on X
15. isTargetedSeq: boolean indicating whether data come from a targeted sequencing experiment
16. failedarrays: placeholder, NULL

`ascat.metrics` *Function to extract different metrics from ASCAT profiles.*

Description

Function to extract different metrics from ASCAT profiles.

Usage

```
ascat.metrics(ASCAT_input_object, ASCAT_output_object)
```

Arguments

`ASCAT_input_object`
: R object generated by the ascat.aspcf function and given to the ascat.runAscat function.

`ASCAT_output_object`
: R object generated by the ascat.runAscat function.

Value

A dataframe (one sample per line) with the following metrics (as columns):
sex - Sex information as provided.
tumour_mapd - Median Absolute Pairwise Difference (MAPD) in tumour logR track.
normal_mapd - Median Absolute Pairwise Difference (MAPD) in normal logR track (should be NA without matched normals and 0 for sequencing data).
GC_correction_before - logR/GC correlation before correction.
GC_correction_after - logR/GC correlation after correction.
RT_correction_before - logR/RT correlation before correction.
RT_correction_after - logR/RT correlation after correction.
n_het_SNP - Number of heterozygous SNPs.
n_segs_logR - Number of segments in the logR track.
n_segs_BAF - Number of segments in the BAF track.
n_segs_logRBAF_diff - Difference between number of segments in the logR versus BAF track.
frac_homo - Fraction of homozygous (<0.1 | >0.9) probes in tumour.
purity - Purity estimate.
ploidy - Ploidy estimate.
goodness_of_fit - Goodness of fit.
size_intermediate_segments - Total size of (unrounded) segments in the X.45-X.55 range.
size_odd_segments - Total size of segments with an odd (1/3/5/+) CN (either nMajor or nMinor).
n_segs - Number of copy-number segments.
segs_size - Total size of all segments.
n_segs_1kSNP - Number of segments per 1k heterozygous SNPs.
homdel_segs - Number of segments with homozygous deletion.

homdel_largest - largest segment with homozygous deletion.
homdel_size - Total size of segments with homozygous deletion.
homdel_fraction - Fraction of the genome with homozygous deletion.
LOH - Fraction of the genome with LOH (ignoring sex chromosomes).
mode_minA - Mode of the minor allele (ignoring sex chromosomes).
mode_majA - Mode of the major allele (ignoring sex chromosomes).
WGD - Whole genome doubling event (ignoring sex chromosomes).
GI - Genomic instability score (ignoring sex chromosomes).

Author(s)

tl

`ascat.plotAdjustedAscatProfile`
ascat.plotAdjustedAscatProfile

Description

Function plotting the "adjusted" (with realistic chromosome sizes) rounded/unrounded ASCAT profiles over all chromosomes.

Usage

```
ascat.plotAdjustedAscatProfile(
  ASCAT_output_object,
  REF,
  y_limit = 5,
  plot_unrounded = FALSE,
  png_prefix = ""
)
```

Arguments

`ASCAT_output_object`	R object generated by the ascat.runAscat function.
`REF`	Can be "hg19", "hg38" or "CHM13" for standard human genome or a data.frame with three columns: chrom, start and end.
`y_limit`	Optional parameter determining the size of the y axis in the profile (default=5).
`plot_unrounded`	Optional parameter to define whether rounded (default) or unrounded profile (set to TRUE) should be plotted.
`png_prefix`	Optional parameter to add a prefix to png name (can be also used to set a path).

Value

Plot showing the adjusted (rounded/unrounded) ASCAT profile of the sample

ascat.plotAscatProfile *ascat.plotAscatProfile*

Description

Function plotting the rounded ASCAT profiles over all chromosomes

Usage

```
ascat.plotAscatProfile(
  n1all,
  n2all,
  heteroprobes,
  ploidy,
  rho,
  goodnessOfFit,
  nonaberrant,
  y_limit = 5,
  ch,
  lrr,
  bafsegmented,
  chrs
)
```

Arguments

n1all	copy number major allele
n2all	copy number minor allele
heteroprobes	probes with heterozygous germline
ploidy	ploidy of the sample
rho	purity of the sample
goodnessOfFit	estimated goodness of fit
nonaberrant	boolean flag denoting non-aberrated samples
y_limit	Optional parameter determining the size of the y axis in the nonrounded plot and ASCAT profile. Default=5
ch	a list containing c vectors, where c is the number of chromosomes and every vector contains all probe numbers per chromosome
lrr	(unsegmented) log R, in genomic sequence (all probes), with probe IDs
bafsegmented	B Allele Frequency, segmented, in genomic sequence (only probes heterozygous in germline), with probe IDs
chrs	a vector containing the names for the chromosomes (e.g. c(1:22, "X"))

Value

plot showing the ASCAT profile of the sample

ascat.plotGenotypes *ascat.plotGenotypes*

Description

ascat.plotGenotypes

Usage

```
ascat.plotGenotypes(ASCATobj, title, Tumor_BAF_noNA, Hom, ch_noNA)
```

Arguments

ASCATobj	an ASCAT object
title	main title of the plot
Tumor_BAF_noNA	B-allele frequencies of the tumour sample with removed NA values
Hom	Boolean vector denoting homozygous SNPs
ch_noNA	vector of probes per chromosome (NA values excluded)

Value

plot showing classified BAF per sample, with unused SNPs in green, germline homozygous SNPs in blue and all others in red

ascat.plotNonRounded *ascat.plotNonRounded*

Description

Function plotting the unrounded ASCAT copy number over all chromosomes

Usage

```
ascat.plotNonRounded(
  ploidy,
  rho,
  goodnessOfFit,
  nonaberrant,
  nAfull,
  nBfull,
  y_limit = 5,
  bafsegmented,
  ch,
  lrr,
  chrs
)
```

Arguments

ploidy	ploidy of the sample
rho	purity of the sample
goodnessOfFit	estimated goodness of fit
nonaberrant	boolean flag denoting non-aberrated samples
nAfull	copy number major allele
nBfull	copy number minor allele
y_limit	Optional parameter determining the size of the y axis in the nonrounded plot and ASCAT profile. Default=5
bafsegmented	B Allele Frequency, segmented, in genomic sequence (only probes heterozygous in germline), with probe IDs
ch	a list containing c vectors, where c is the number of chromosomes and every vector contains all probe numbers per chromosome
lrr	(unsegmented) log R, in genomic sequence (all probes), with probe IDs
chrs	a vector containing the names for the chromosomes (e.g. c(1:22, "X"))

Value

plot showing the nonrounded copy number profile, using base plotting function

---

ascat.plotRawData *ascat.plotRawData*

Description

Plots SNP array data

Usage

```
ascat.plotRawData(
  ASCATobj,
  img.dir = ".",
  img.prefix = "",
  logr.y_values = c(-2, 2)
)
```

Arguments

ASCATobj	an ASCAT object (e.g. data structure from ascat.loadData)
img.dir	directory in which figures will be written
img.prefix	prefix for figure names
logr.y_values	define Y min and max values for logR track (optional; default: c(-2, 2))

Value

Produces png files showing the logR and BAF values for tumour and germline samples

ascat.plotSegmentedData *ascat.plotSegmentedData*

Description

plots the SNP array data before and after segmentation

Usage

```
ascat.plotSegmentedData(
  ASCATobj,
  img.dir = ".",
  img.prefix = "",
  logr.y_values = c(-2, 2)
)
```

Arguments

ASCATobj	an ASCAT object (e.g. from ascat.aspcf)
img.dir	directory in which figures will be written
img.prefix	prefix for figure names
logr.y_values	define Y min and max values for logR track (optional; default: c(-2, 2))

Value

png files showing raw and segmented tumour logR and BAF

---

ascat.plotSunrise *ascat.plotSunrise*

Description

ascat.plotSunrise

Usage

```
ascat.plotSunrise(d, psi_opt1, rho_opt1, minim = TRUE)
```

Arguments

d	distance matrix for a range of ploidy and tumour percentage values
psi_opt1	optimal ploidy
rho_opt1	optimal purity
minim	when set to true, optimal regions in the sunrise plot are depicted in blue; if set to false, colours are inverted and red corresponds to optimal values (default: TRUE)

Value

plot visualising range of ploidy and tumour percentage values

ascat.predictGermlineGenotypes
ascat.predictGermlineGenotypes

Description

predicts the germline genotypes of samples for which no matched germline sample is available

Usage

```
ascat.predictGermlineGenotypes(
  ASCATobj,
  platform = "AffySNP6",
  img.dir = ".",
  img.prefix = ""
)
```

Arguments

ASCATobj	an ASCAT object
platform	used array platform
img.dir	directory in which figures will be written
img.prefix	prefix for figure names

Details

Currently possible values for platform:
AffySNP6 (default)
Custom10k
IlluminaASA
IlluminaGSAv3
Illumina109k
IlluminaCytoSNP
IlluminaCytoSNP850k
Illumina610k
Illumina660k
Illumina700k
Illumina1M
Illumina2.5M
IlluminaOmni5
IlluminaGDACyto-8
Affy10k
Affy100k
Affy250k_sty
Affy250k_nsp
AffyOncoScan
AffyCytoScanHD
HumanCNV370quad
HumanCore12
HumanCoreExome24

HumanOmniExpress12
IlluminaOmniExpressExome
WGS_hg38_50X

Value

predicted germline genotypes

`ascat.prepareHTS` *Extract both logR and BAF values from sequencing data*

Description

Method derived from the Battenberg package (https://github.com/Wedge-lab/battenberg).

Usage

```
ascat.prepareHTS(
  tumourseqfile,
  normalseqfile = NA,
  tumourname,
  normalname = NA,
  allelecounter_exe,
  alleles.prefix,
  loci.prefix,
  gender,
  genomeVersion,
  nthreads = 1,
  tumourLogR_file = NA,
  tumourBAF_file = NA,
  normalLogR_file = NA,
  normalBAF_file = NA,
  minCounts = 10,
  BED_file = NA,
  probloci_file = NA,
  chrom_names = c(1:22, "X"),
  min_base_qual = 20,
  min_map_qual = 35,
  additional_allelecounter_flags = NA,
  skip_allele_counting_tumour = FALSE,
  skip_allele_counting_normal = FALSE,
  loci_binsize = 1,
  seed = as.integer(Sys.time())
)
```

Arguments

`tumourseqfile`	Full path to the tumour BAM/CRAM file.
`normalseqfile`	Full path to the normal BAM/CRAM file.
`tumourname`	Identifier to be used for tumour output files.

`normalname` Identifier to be used for normal output files.

`allelecounter_exe` Path to the allele counter executable.

`alleles.prefix` Prefix path to the allele data (e.g. "G1000_alleles_chr").

`loci.prefix` Prefix path to the loci data (e.g. "G1000_loci_chr").

`gender` Gender information, either "XX" (=female) or "XY" (=male).

`genomeVersion` Genome version, available options are "hg19", "hg38" or "CHM13".

`nthreads` The number of parallel processes for getting allele counts (optional, default=1).

`tumourLogR_file` Path to the tumour logR output (optional, paste0(tumourname, "_tumourLogR.txt")).

`tumourBAF_file` Path to the tumour BAF output (optional, paste0(tumourname, "_tumourBAF.txt")).

`normalLogR_file` Path to the normal logR output (optional, paste0(tumourname, "_normalLogR.txt")).

`normalBAF_file` Path to the normal BAF output (optional, paste0(tumourname, "_normalBAF.txt")).

`minCounts` Minimum depth required in the normal for a SNP to be considered (optional, default=10).

`BED_file` A BED file for only looking at SNPs within specific intervals (optional, default=NA).

`probloci_file` A file (chromosome <tab> position; no header) containing specific loci to ignore (optional, default=NA).

`chrom_names` A vector containing the names of chromosomes to be considered (optional, default=c(1:22, "X")).

`min_base_qual` Minimum base quality required for a read to be counted (optional, default=20).

`min_map_qual` Minimum mapping quality required for a read to be counted (optional, default=35).

`additional_allelecounter_flags` Additional flags passed on to alleleCounter, e.g., -r <FASTA> for parsing CRAMs (optional, default=NA).

`skip_allele_counting_tumour` Flag, set to TRUE if tumour allele counting is already complete (files are expected in the working directory on disk; optional, default=FALSE).

`skip_allele_counting_normal` Flag, set to TRUE if normal allele counting is already complete (files are expected in the working directory on disk; optional, default=FALSE).

`loci_binsize` Size of the bins for long-read sequencing data (optional, default=1).

`seed` A seed to be set when randomising the alleles (optional, default=as.integer(Sys.time())).

Author(s)

sd11, tl

`ascat.prepareTargetedSeq` *Method to extract a curated list of SNPs covered by a targeted sequencing experiment.*

Description

From a complete set of loci (alleles.prefix), this method will keep SNPs falling into the targeted design (based on BED_file) and check allele counts in normal samples (listed in Worksheet). The cleaned list of loci/allele files will be located under Workdir/alleleData/Cleaned/.

Usage

```
ascat.prepareTargetedSeq(
  Worksheet,
  Workdir,
  alleles.prefix,
  BED_file,
  allelecounter_exe,
  genomeVersion,
  nthreads = 1,
  minCounts = 10,
  is_chr_based = FALSE,
  chrom_names = c(1:22, "X"),
  min_base_qual = 20,
  min_map_qual = 35,
  additional_allelecounter_flags = NA,
  plotQC = TRUE
)
```

Arguments

`Worksheet`	A tab-separated file with the following columns: Patient_ID, Normal_ID, Normal_file and Gender (additional columns can be provided but will not be used). Must contain one single normal per patient. Normal_file can either be paths to BAMs/CRAMs or paths to pre-computed (zipped) alleleCounts (e.g. "sample_alleleCounts_chr"). Gender must either be XX (females) or XY (males).
`Workdir`	The folder where output should go (will be created if it doesn't exist).
`alleles.prefix`	Prefix path to the allele data (e.g. "G1000_alleles_chr").
`BED_file`	A BED file for only looking at SNPs within specific intervals. Must fit with the design used for targeted sequencing.
`allelecounter_exe`	Path to the allele counter executable.
`genomeVersion`	Genome version, available options are 'hg19', 'hg38' or 'CHM13'.
`nthreads`	The number of parallel processes to speed up the process (optional, default=1).
`minCounts`	Minimum depth required in the normal for a SNP to be considered (optional, default=10).
`is_chr_based`	A boolean indicating whether data is "chr"-based (e.g. 'chr1' instead of '1'; optional, default=FALSE).

chrom_names	A vector containing the names of chromosomes to be considered (optional, default=c(1:22, "X")). Do not set it to paste0("chr", c(1:22, "X")) if data is "chr"-based.
min_base_qual	Minimum base quality required for a read to be counted (optional, default=20).
min_map_qual	Minimum mapping quality required for a read to be counted (optional, default=35).
additional_allelecounter_flags	Additional flags passed on to alleleCounter, e.g., -r <FASTA> for parsing CRAMs (optional, default=NA).
plotQC	A boolean to generate QC reports as PNGs (optional, default=TRUE).

---

ascat.runAscat *ascat.runAscat*

Description

ASCAT main function, calculating the allele-specific copy numbers

Usage

```
ascat.runAscat(
  ASCATobj,
  gamma = 0.55,
  pdfPlot = FALSE,
  y_limit = 5,
  circos = NA,
  min_ploidy = 1.5,
  max_ploidy = 5.5,
  min_purity = 0.1,
  max_purity = 1.05,
  rho_manual = NA,
  psi_manual = NA,
  img.dir = ".",
  img.prefix = "",
  write_segments = FALSE
)
```

Arguments

ASCATobj	an ASCAT object from ascat.aspcf
gamma	technology parameter, compaction of Log R profiles (expected decrease in case of deletion in diploid sample, 100% aberrant cells; 1 in ideal case, 0.55 of Illumina 109K arrays)
pdfPlot	Optional flag if nonrounded plots and ASCAT profile in pdf format are desired. Default=F
y_limit	Optional parameter determining the size of the y axis in the nonrounded plot and ASCAT profile. Default=5
circos	Optional file to output the non-rounded values in Circos track format. Default=NA

`min_ploidy`	optional numerical parameter determining the minimum boundary of the ploidy solution search space (expert parameter, don't adapt unless you know what you're doing). Default=1.5
`max_ploidy`	optional numerical parameter determining the maximum boundary of the ploidy solution search space (expert parameter, don't adapt unless you know what you're doing). Default=5.5
`min_purity`	optional numerical parameter determining the minimum boundary of the purity solution search space (expert parameter, don't adapt unless you know what you're doing). Default=0.1
`max_purity`	optional numerical parameter determining the maximum boundary of the purity solution search space (expert parameter, don't adapt unless you know what you're doing). Default=1.05
`rho_manual`	optional argument to override ASCAT optimization and supply rho parameter (expert parameter, don't adapt unless you know what you're doing).
`psi_manual`	optional argument to override ASCAT optimization and supply psi parameter (expert parameter, don't adapt unless you know what you're doing).
`img.dir`	directory in which figures will be written
`img.prefix`	prefix for figure names
`write_segments`	Optional flag to output segments in text files (.segments_raw.txt and .segments.txt under img.dir). Default=F

Details

Note: for copy number only probes, nA contains the copy number value and nB = 0.

Value

an ASCAT output object, containing:
1. nA: copy number of the A allele
2. nB: copy number of the B allele
3. purity: the tumour purity of all arrays
4. aberrantcellfraction: the aberrant cell fraction (=tumour purity) of all arrays
5. ploidy: the ploidy of all arrays
6. failedarrays: arrays on which ASCAT analysis failed
7. nonaberrantarrays: arrays on which ASCAT analysis indicates that they show virtually no aberrations
8. segments: an array containing the copy number segments of each sample (not including failed arrays)
9. segments_raw: an array containing the copy number segments of each sample without any rounding applied
10. distance_matrix: distances for a range of ploidy and tumor percentage values

---

`ascat.synchroniseFiles`

Synchronise SNPs across files

Description

Synchronise SNPs across files

Usage

```
ascat.synchroniseFiles(
  samplename,
  tumourLogR_file,
  tumourBAF_file,
  normalLogR_file,
  normalBAF_file
)
```

Arguments

samplename	String, name of the sample.
tumourLogR_file	File where LogR from the tumour will be read and overwritten.
tumourBAF_file	File where BAF from the tumour will be read and overwritten.
normalLogR_file	File where LogR from the normal will be read and overwritten.
normalBAF_file	File where BAF from the normal will be read and overwritten.

Author(s)

tl

Index