Package ‘cfTools’

March 20, 2023

Type Package

Title Informatics Tools for Cell-Free DNA Study

Version 0.99.0

Description The cfTools R package provides methods for cell-free DNA (cfDNA) methylation data analysis to facilitate cfDNA-based studies. Given the methylation sequencing data of a cfDNA sample, for each cancer marker or tissue marker, we deconvolve the tumor-derived or tissue-specific reads from all reads falling in the marker region. Our read-based deconvolution algorithm exploits the pervasiveness of DNA methylation for signal enhancement, therefore can sensitively identify a trace amount of tumor-specific or tissue-specific cfDNA in plasma. cfTools provides functions for (1) cancer detection: sensitively detect tumor-derived cfDNA and estimate the tumor-derived cfDNA fraction (tumor burden); (2) tissue deconvolution: infer the tissue type composition and the cfDNA fraction of multiple tissue types for a plasma cfDNA sample. These functions can serve as foundations for more advanced cfDNA-based studies, including cancer diagnosis and disease monitoring.

License file LICENSE

Encoding UTF-8

Suggests BiocStyle,
knitr,
rmarkdown,
testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.2.3

Imports Rcpp,
utils,
GenomicRanges

StagedInstall no

SystemRequirements Python 3+, NumPy, SciPy

biocViews Software, BiomedicalInformatics, Epigenetics, Sequencing, MethylSeq, DNAMethylation, DifferentialMethylation

VignetteBuilder knitr

LinkingTo Rcpp,
BH

R topics documented:

`CancerDetector` *Cancer Detector*

Description

Detect tumor-derived cfDNA and estimate the tumor burden.

Usage

```
CancerDetector(readsBinningFile, tissueMarkersFile, python = "python3")
```

Arguments

`readsBinningFile`
a file of the fragment-level methylation states of reads that mapped to the markers.

`tissueMarkersFile`
a file of paired shape parameters of beta distributions for markers.

`python` a path to Python 3. Default is "python3".

Value

cfDNA tumor burden and normal cfDNA fraction.

Examples

```
## input files
demo.dir <- system.file("extdata", package="cfTools")
readsBinningFile <- file.path(demo.dir, "CancerDetector.reads.txt")
tissueMarkersFile <- file.path(demo.dir, "CancerDetector.markers.txt")

CancerDetector(readsBinningFile, tissueMarkersFile)
```

cfDeconvolve *cfDNA methylation read deconvolution*

Description

Infer the tissue-type composition of plasma cfDNA.

Usage

```
cfDeconvolve(
  readsBinningFile,
  tissueMarkersFile,
  numTissues,
  emAlgorithmType = "em.global.unknown",
  likelihoodRatioThreshold = 2,
  emMaxIterations = 100
)
```

Arguments

`readsBinningFile`
: a file of the fragment-level methylation states of reads that mapped to the markers. Either in plain text or compressed form.

`tissueMarkersFile`
: a file of paired shape parameters of beta distributions for markers.

`numTissues`
: a number of tissue types.

`emAlgorithmType`
: a read-based tissue deconvolution EM algorithm type: em.global.unknown (default), em.global.known, em.local.unknown, em.local.known.

`likelihoodRatioThreshold`
: a positive float number. Default is 2.

`emMaxIterations`
: a number of EM algorithm maximum iteration. Default is 100.

Value

a data frame containing the cfDNA fractions of different tissue types and an unknown class.

Examples

```
## input files
demo.dir <- system.file("extdata", package="cfTools")
readsBinningFile <- file.path(demo.dir, "cfDeconvolve.reads.txt")
tissueMarkersFile <- file.path(demo.dir, "cfDeconvolve.markers.txt")
numTissues <- 7
emAlgorithmType <- "em.global.unknown"
likelihoodRatioThreshold <- 2

cfDeconvolve(readsBinningFile, tissueMarkersFile, numTissues,
emAlgorithmType, likelihoodRatioThreshold)
```

CollapseCpGs *Generate fragment-level methylation states of CpGs*

Description

Collapse the methylation states of all CpGs corresponding to the same fragment onto one line in output.

Usage

```
CollapseCpGs(CpG_OT, CpG_OB, output.dir = "", id = "", python = "python3")
```

Arguments

`CpG_OT`	a file of methylation information for CpG on the original top strand (OT), which is one of the outputs from ‘bismark methylation extractor‘.
`CpG_OB`	a file of methylation information for CpG on the original bottom strand (OB), which is one of the outputs from ‘bismark methylation extractor‘.
`output.dir`	a path to the output directory. Default is "", which means the output will not be written into a file.
`id`	an ID name for the input data. Default is "", which means the output will not be written into a file.
`python`	a path to Python 3. Default is "python3".

Value

a data frame in BED file format and/or written to an output BED file.

Examples

```
## input files
demo.dir <- system.file("extdata", package="cfTools")
CpG_OT <- file.path(demo.dir, "CpG_OT_demo.txt.gz")
CpG_OB <- file.path(demo.dir, "CpG_OB_demo.txt.gz")

output <- CollapseCpGs(CpG_OT, CpG_OB)
```

CollapsePEReads *Generate fragment-level information for paired-end sequencing reads*

Description

Collapse BED file (the output of ‘bedtools bamtobed‘) to fragment-level for paired-end sequencing reads.

Usage

```
CollapsePEReads(bed_file, output.dir = "", id = "", python = "python3")
```

Arguments

`bed_file`	a (sorted) BED file of paired-end reads.
`output.dir`	a path to the output directory. Default is "", which means the output will not be written into a file.
`id`	an ID name for the input data. Default is "", which means the output will not be written into a file.
`python`	a path to Python 3. Default is "python3".

Value

a data frame in BED file format and/or written to an output BED file.

Examples

```
## input files
demo.dir <- system.file("extdata", package="cfTools")
PEReads <- file.path(demo.dir, "demo.sorted.bed.gz")

output <- CollapsePEReads(PEReads)
```

`GenerateFragMeth` *Generate fragment-level information about methylation states*

Description

Join two data frames containing the fragment information and the methylation states on each fragment into one data frame.

Usage

```
GenerateFragMeth(
  frag_bed,
  meth_bed,
  output.dir = "",
  id = "",
  python = "python3"
)
```

Arguments

`frag_bed`	a BED file containing information for every fragment, which is the output of CollapsePEReads().
`meth_bed`	a BED file containing methylation states on every fragment, which is the output of CollapseCpGs().
`output.dir`	a path to the output directory. Default is "", which means the output will not be written into a file.
`id`	an ID name for the input data. Default is "", which means the output will not be written into a file.
`python`	a path to Python 3. Default is "python3".

Value

a data frame in BED file format and/or written to an output BED file.

Examples

```
## input files
demo.dir <- system.file("extdata", package="cfTools")
frag_bed <- read.delim(file.path(demo.dir, "demo.refo_frag.bed"), colClasses = "character")
meth_bed <- read.delim(file.path(demo.dir, "demo.refo_meth.bed"), colClasses = "character")

output <- GenerateFragMeth(frag_bed, meth_bed)
```

GenerateMarkerParam *Generate the methylation pattern of markers*

Description

Output paired shape parameters of beta distributions for methylation markers.

Usage

```
GenerateMarkerParam(x, sample.types, marker.names, output.file = "")
```

Arguments

x	a data frame of methylation levels (e.g., beta values), where each row is a sample and each column is a marker.
sample.types	a vector of sample types (e.g., tumor or normal, tissue types) corresponding to the rows of the data frame.
marker.names	a vector of marker names corresponding to the columns of the data frame.
output.file	a character string naming the output file. Default is "", which means the output will not be written into a file.

Value

a data frame containing the paired shape parameters of beta distributions for markers and/or written to an output file.

Examples

```
## input files
demo.dir <- system.file("extdata", package="cfTools")
methLevel <- read.csv(file.path(demo.dir, "beta_matrix.csv"), row.names=1)
sampleTypes <- read.csv(file.path(demo.dir, "sample_type.csv"), row.names=1)$Sample.Type
markerNames <- read.csv(file.path(demo.dir, "marker_index.csv"), row.names=1)$Marker.Index

output <- GenerateMarkerParam(methLevel, sampleTypes, markerNames)
```

Index