Package 'RNAshapeQC'

November 12, 2026

Type Package

Title RNA Coverage-Shape-Based Quality Control Metrics

Version 1.0.0

Description RNAshapeQC provides coverage-shape-based quality control (QC) metrics for mRNA-seq and total RNA-seq data. It supports per-gene pileup construction from BAM files as well as toy datasets for quick-start examples. The package implements protocol-specific metrics, including decay rate (DR), degradation score (DS), mean coverage depth (MCD), window coefficient of variation (wCV), area under the curve (AUC), and shape-based sample-level indices. RNAshapeQC also includes HPC-friendly functions for per-gene batch processing and cross-study pileup generation. This package enables interpretable, protocol-specific QC assessments for diverse RNA-seq workflows.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.4)

Imports GenomicRanges, IRanges, Rsamtools, foreach, zoo, ggplot2, dplyr, magrittr, tidyr, ComplexHeatmap, circlize, dendextend, DescTools, ggpubr, MASS, SummarizedExperiment, BiocParallel

Suggests doParallel, knitr, rmarkdown, BiocStyle, testthat (>= 3.0.0), PupillometryR, matrixStats, RColorBrewer, scales, tibble, patchwork, ConsensusClusterPlus

biocViews RNASeq, QualityControl, Coverage, Transcriptomics, Sequencing

URL https://github.com/hyochoi/RNAshapeQC

BugReports https://github.com/hyochoi/RNAshapeQC/issues

VignetteBuilder knitr

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

git_branch RELEASE_3_23

git_last_commit 5d1a8e4

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Miyeon Yeon [aut, cre, cph] (ORCID:
<https://orcid.org/0000-0003-1618-0643>),
Won-Young Choi [aut, cph] (ORCID:
<https://orcid.org/0009-0003-8276-2235>),
Jin Young Lee [ctb] (ORCID: <https://orcid.org/0000-0002-5366-7488>),
Katherine A. Hoadley [aut] (ORCID:
<https://orcid.org/0000-0002-1216-477X>),
D. Neil Hayes [aut, fnd, cph] (ORCID:
<https://orcid.org/0000-0001-6203-7771>),
Hyo Young Choi [aut, cph] (ORCID:
<https://orcid.org/0000-0002-7627-8493>)

Maintainer Miyeon Yeon <myeon@uthsc.edu>

Contents

.build_pileupExon . . . 3
combine_vecObj . . . 4
compute_DIIwt . . . 5
compute_DR . . . 6
compute_MCD . . . 7
compute_SOI . . . 8
compute_wCV . . . 9
construct_pileup . . . 10
extract_RData . . . 11
filter_lowExpGenes . . . 12
gen_DR . . . 12
gen_MCD . . . 13
gen_wCV . . . 14
get_DIIhc . . . 15
get_DIIwt . . . 16
get_DR . . . 17
get_MCD . . . 18
get_pileupExon . . . 19
get_SOI . . . 19
get_wCV . . . 20
plot_DIIwt . . . 21
plot_GBC . . . 22
plot_GBCos . . . 24
plot_SOI . . . 25
TOY_mrna_mat . . . 26
TOY_mrna_se . . . 26
TOY_total_mat . . . 27

TOY_total_se 28

Index **29**

`.build_pileupExon` *Core helper to build exon-only pileup*

Description

Core helper to build exon-only pileup

Usage

```
.build_pileupExon(pileupPath, cases = NULL, study = NULL)
```

Arguments

`pileupPath`	file paths of coverage pileupData including .RData file names
`cases`	a vector of specific samples among all samples in pileup. If NULL, all samples are selected. Default is NULL.
`study`	a character of study abbreviation in the pileupList. Default is NULL.

Value

a numeric matrix of exon-only coverage (rows: exon positions, columns: samples).

References

Choi, H.Y., Jo, H., Zhao, X. et al. SCISSOR: a framework for identifying structural changes in RNA transcripts. Nat Commun 12, 286 (2021).

Examples

```
## Requires a base-level coverage pileup .RData file
try(
    .build_pileupExon(pileupPath="path/to/pileup.RData"), silent=TRUE
)
```

combine_vecObj *Combine vectors as a matrix from objects*

Description

Combine vectors as a matrix from objects

Usage

```
combine_vecObj(
  filePath,
  objName = NULL,
  header = NULL,
  skip = NULL,
  txtCol = NULL,
  margin,
  rowNames,
  colNames,
  nCores = 2
)
```

Arguments

filePath	file paths including .RData file names
objName	a character of object name
header	logical; whether the text files have a header line.
skip	integer; number of lines to skip before reading data from '.txt' files.
txtCol	integer; column index in the text file that contains the numeric vector to be extracted.
margin	1 and 2 return for gene- and sample-level vectors, respectively.
rowNames	a vector of gene names
colNames	a vector of sample names
nCores	the number of cores for parallel computing. Default is 2.

Value

a gene x sample matrix

Examples

```
## API illustration only
invisible(NULL)
```

compute_DIIwt *Core helper to compute a degraded/intact index using gene weight*

Description

Core helper to compute a degraded/intact index using gene weight

Usage

```
compute_DIIwt(DR, alpha = 2, cutoff = 3, TPM, thr = 5, pct = 40, genelength)
```

Arguments

DR	a the number of genes x the number of samples matrix of decay rates
alpha	a positive numeric exponent factor to weight the magnitude of decay rates. Default is 2.
cutoff	numeric threshold on projection depth used to classify samples.
TPM	a numeric matrix of TPM values with the same genes in rows and the same samples in columns as DR.
thr	threshold. Default is 5.
pct	percent. Default is 40.
genelength	a gene length (bp) vector with names as gene IDs.

Value

a matrix of with decay rate with filtered genes; a matrix including a vector of DII; a data frame of gene info; and a scale factor.

Examples

```
data("TOY_mrna_mat")

compute_DIIwt(
    DR         = TOY_mrna_mat$DR,
    TPM        = TOY_mrna_mat$TPM,
    genelength = TOY_mrna_mat$gene_length
)
```

compute_DR *Core helper to compute decay rate*

Description

Core helper to compute decay rate

Usage

```
compute_DR(
  pileupData,
  exonRanges,
  sampleInfo,
  cases = NULL,
  logshiftVal = 10,
  plotNormalization = FALSE
)
```

Arguments

pileupData	exon-only coverage pileup matrix for a single gene.
exonRanges	GRanges object specifying exon coordinates for the gene.
sampleInfo	a sample information table including sample id. The number of rows is equal to the number of samples.
cases	optional character vector specifying a subset of samples. used for handling missing coverage.
logshiftVal	numeric; passed to process_pileup().
plotNormalization	logical; passed to process_pileup().

Details

The arguments pileupData, exonRanges, logshiftVal, and plotNormalization are passed directly to process_pileup(); see its documentation for details.

Value

a numeric vector of decay rates, one value per sample.

References

Choi, H.Y., Jo, H., Zhao, X. et al. SCISSOR: a framework for identifying structural changes in RNA transcripts. Nat Commun 12, 286 (2021).

Examples

```
## API illustration only
## Exon-only pileup matrix (rows: exon positions, columns: samples)
## Typically obtained via get_pileupExon()
pileupData <- matrix(c(10, 12, 8, 9, 5, 6, 4, 5), nrow=2, byrow=TRUE)
colnames(pileupData) <- c("S1", "S2", "S3", "S4")

sampleInfo <- data.frame(SampleID=colnames(pileupData))

exonRanges <- list(
    Gene        = "KEAP1",
    cRanges     = data.frame(e.start=c(1), e.end=c(50001), row.names="exon1"),
    regions     = "chr19:10600000-10650000:+",
    new.regions = "chr19:10600000-10650000:+",
    strand      = "+"
)

compute_DR(
    pileupData = pileupData,
    exonRanges = exonRanges,
    sampleInfo = sampleInfo
)
```

`compute_MCD` *Core helper to compute a mean coverage depth*

Description

Core helper to compute a mean coverage depth

Usage

```
compute_MCD(pileupData, sampleInfo, cases = NULL)
```

Arguments

`pileupData`	exon-only coverage pileup matrix for a single gene.
`sampleInfo`	a sample information table including sample id. The number of rows is equal to the number of samples.
`cases`	optional character vector specifying a subset of samples. used for handling missing coverage.

Value

a numeric vector of mean coverage depth, one value per sample.

Examples

```
## API illustration only
## Exon-only pileup matrix (rows: exon positions, columns: samples)
## Typically obtained via get_pileupExon()
pileupData <- matrix(c(10, 12, 8, 9, 5, 6, 4, 5), nrow=2, byrow=TRUE)
colnames(pileupData) <- c("S1", "S2", "S3", "S4")

sampleInfo <- data.frame(SampleID=colnames(pileupData))

compute_MCD(pileupData=pileupData, sampleInfo=sampleInfo)
```

---

compute_SOI *Core helper to compute a suboptimal/optimal index*

Description

Core helper to compute a suboptimal/optimal index

Usage

```
compute_SOI(MCD, wCV, rstPct = 20, obsPct = 50, cutoff = 3)
```

Arguments

MCD	a mean coverage depth is a the number of genes x the number of samples matrix.
wCV	a window coefficient of variation is a the number of genes x the number of samples matrix.
rstPct	restricted percent (one-side) to restrict genes by log transformed MC. Default is 20.
obsPct	span includes the percent of observations in each local regression. Default is 50.
cutoff	numeric threshold on projection depth used to classify samples.

Value

a matrix including a vector of SOI; a coordinate matrix of smoothed data; and a range of MCD.

Examples

```
data("TOY_total_mat")

compute_SOI(MCD=TOY_total_mat$MCD, wCV=TOY_total_mat$wCV)
```

compute_wCV *Core helper to compute a window coefficient of variation*

Description

Core helper to compute a window coefficient of variation

Usage

```
compute_wCV(
  pileupData,
  sampleInfo,
  rnum = 100,
  method = 1,
  winSize = 20,
  egPct = 10,
  cases = NULL
)
```

Arguments

pileupData	exon-only coverage pileup matrix for a single gene.
sampleInfo	a sample information table including sample id. The number of rows is equal to the number of samples.
rnum	the number of regions for uniformly dividing the x-axis. Default is 100.
method	1 and 2 return the raw read depth and the interpolated read depth at the normalized genomic position, respectively. Default is 1.
winSize	window size of the rolling window. Default is 20.
egPct	edge percent (one-side) to calculate the trimmed mean. Default is 10.
cases	optional character vector specifying a subset of samples. used for handling missing coverage.

Value

a numeric vector of window coefficients of variation, one value per sample.

Examples

```
## API illustration only
## Exon-only pileup matrix (rows: exon positions, columns: samples)
## Typically obtained via get_pileupExon()
pileupData <- matrix(c(10, 12, 8, 9, 5, 6, 4, 5), nrow=2, byrow=TRUE)
colnames(pileupData) <- c("S1", "S2", "S3", "S4")

sampleInfo <- data.frame(SampleID=colnames(pileupData))
```

```
compute_wCV(
    pileupData = pileupData,
    sampleInfo = sampleInfo,
    rnum       = 10,
    winSize    = 2
)
```

`construct_pileup` *Construct a per-gene pileup from BAM files (for single-study or multi-study)*

Description

Construct a per-gene pileup from BAM files (for single-study or multi-study)

Usage

```
construct_pileup(
  Gene,
  studylist,
  regionsFile,
  regionsFormat = c("auto", "SCISSOR_gaf", "gencode.regions"),
  geneCol = 1,
  regionsCol = NULL,
  bamFilesList,
  caseIDList,
  max_depth = 1e+05,
  strand.specific = FALSE,
  nCores = 2,
  outFile = NULL
)
```

Arguments

`Gene`	a character of gene name
`studylist`	a character vector of study IDs or abbreviation/name
`regionsFile`	either a file path or a data.frame specifying gene regions. If a file path is given, it is read using `read.table()` with automatic handling of header/non-header cases.
`regionsFormat`	character; one of `"auto"`, `"SCISSOR_gaf"`, or `"gencode.regions"`. In `"auto"` mode, the function attempts to detect whether the file uses SCISSOR-style columns `gene_name` and `regions`, falling back to `"gencode.regions"` otherwise.
`geneCol`	integer; column index for the gene identifier when `regionsFormat="gencode.regions"`. Default is 1.
`regionsCol`	integer; column index for the regions string when `regionsFormat="gencode.regions"`. Default is `NULL`, which is interpreted as 2.

bamFilesList	named list of character vectors of BAM file paths
caseIDList	named list of character vectors of sample IDs corresponding to bamFilesList
max_depth	integer; max depth parameter for Rsamtools::PileupParam. Default is 100000.
strand.specific	Logical; whether to use strand-specific pileup. If TRUE, only the "+" strand is retained (when strand information is available). Default is FALSE.
nCores	the number of cores for parallel computing. Default is 2.
outFile	a directory with a file name to save outputs. Default is NULL.

Value

a pileup matrix, regions, and ranges of genomic positions

Examples

```
## API illustration only
invisible(NULL)
```

---

extract_RData *Extract an object from .RData*

Description

Extract an object from .RData

Usage

```
extract_RData(file, object)
```

Arguments

file	.RData file
object	object name

Value

the object

References

https://stackoverflow.com/questions/65964064/programmatically-extract-an-object-from-collection-of-rdata-files

Examples

```
tmp <- tempfile(fileext=".RData")
x <- 1
save(x, file=tmp)
extract_RData(tmp, "x")
```

filter_lowExpGenes *Filter low expressed genes*

Description

Filter low expressed genes

Usage

```
filter_lowExpGenes(genelist, TPM, thr = 5, pct = 40)
```

Arguments

genelist	a vector of gene names
TPM	a gene expression counts matrix transformed by TPM
thr	threshold. Default is 5.
pct	percent. Default is 40.

Value

a vector of filtered gene names

Examples

```
data("TOY_mrna_mat")
filter_lowExpGenes(genelist=TOY_mrna_mat$genes, TPM=TOY_mrna_mat$TPM)
```

gen_DR *Get a decay rate for genes and samples (for a single gene)*

Description

Get a decay rate for genes and samples (for a single gene)

Usage

```
gen_DR(Gene, pileupPath, sampleInfo, cases = NULL, Study = NULL, outFile)
```

Arguments

Gene	a character of gene name
pileupPath	file paths of coverage pileupData including .RData file names
sampleInfo	a sample information table including sample id. The number of rows is equal to the number of samples.
cases	a vector of specific samples among all samples in pileup. If NULL, all samples are selected. Default is NULL.
Study	a character of study abbreviation in the pileupList. Default is NULL.
outFile	a directory with a file name to save outputs

Value

Invisibly returns NULL; results are saved to `outFile`.

References

Choi, H.Y., Jo, H., Zhao, X. et al. SCISSOR: a framework for identifying structural changes in RNA transcripts. Nat Commun 12, 286 (2021).

Examples

```
## API illustration only
invisible(NULL)
```

---

gen_MCD *Get a mean coverage depth for genes and samples (for a single gene)*

Description

Get a mean coverage depth for genes and samples (for a single gene)

Usage

```
gen_MCD(Gene, pileupPath, sampleInfo, cases = NULL, Study = NULL, outFile)
```

Arguments

Gene	a character of gene name
pileupPath	file paths of coverage pileupData including .RData file names
sampleInfo	a sample information table including sample id. The number of rows is equal to the number of samples.
cases	a vector of specific samples among all samples in pileup. If NULL, all samples are selected. Default is NULL.
Study	a character of study abbreviation in the pileupList. Default is NULL.
outFile	a directory with a file name to save outputs

Value

Invisibly returns NULL; results are saved to `outFile`.

Examples

```
## API illustration only
invisible(NULL)
```

`gen_wCV` *Get a window coefficient of variation for genes and samples (for a single gene)*

Description

Get a window coefficient of variation for genes and samples (for a single gene)

Usage

```
gen_wCV(
  Gene,
  pileupPath,
  sampleInfo,
  rnum = 100,
  method = 1,
  winSize = 20,
  egPct = 10,
  cases = NULL,
  Study = NULL,
  outFile
)
```

Arguments

`Gene`	a character of gene name
`pileupPath`	file paths of coverage pileupData including .RData file names
`sampleInfo`	a sample information table including sample id. The number of rows is equal to the number of samples.
`rnum`	the number of regions for uniformly dividing the x-axis for gene length normalization. Default is 100.
`method`	1 and 2 return the raw read depth and the interpolated read depth at the normalized genomic position, respectively. Default is 1.
`winSize`	window size of the rolling window. Default is 20.
`egPct`	edge percent (one-side) to calculate the trimmed mean. Default is 10.
`cases`	a vector of specific samples among all samples in pileup. If NULL, all samples are selected. Default is NULL.
`Study`	a character of study abbreviation in the pileupList. Default is NULL.
`outFile`	a directory with a file name to save outputs

Value

Invisibly returns NULL; results are saved to `outFile`.

Examples

```
## API illustration only
invisible(NULL)
```

`get_DIIhc` *Get a degraded/intact index for samples using hierarchical clustering*

Description

Get a degraded/intact index for samples using hierarchical clustering

Usage

```
get_DIIhc(DR, topPct = 5)
```

Arguments

`DR`	a the number of genes x the number of samples matrix of decay rates
`topPct`	top percentages of decay rates defined as degrateGrp=1. Default is 5.

Value

a matrix of binary converted decay rates; hierarchical clustering outputs of samples; and a vector of DII per sample.

Examples

```
data("TOY_mrna_mat")
get_DIIhc(DR=TOY_mrna_mat$DR)
```

get_DIIwt *Get a degraded/intact index for samples using gene weight*

Description

Get a degraded/intact index for samples using gene weight

Usage

```
get_DIIwt(
  DR,
  alpha = 2,
  cutoff = 3,
  TPM = NULL,
  thr = 5,
  pct = 40,
  genelength = NULL,
  assay.DR = "DR",
  assay.TPM = "TPM"
)
```

Arguments

DR	a the number of genes x the number of samples matrix of decay rates
alpha	a positive numeric exponent factor to weight the magnitude of decay rates. Default is 2.
cutoff	numeric threshold on projection depth used to classify samples.
TPM	a numeric matrix of TPM values with the same genes in rows and the same samples in columns as DR.
thr	threshold. Default is 5.
pct	percent. Default is 40.
genelength	a gene length (bp) vector with names as gene IDs.
assay.DR	character string specifying the assay name containing the DR matrix in a SummarizedExperiment object.
assay.TPM	character string specifying the assay name containing the TPM matrix in a SummarizedExperiment object.

Value

a matrix of with decay rate with filtered genes; a matrix including a vector of DII; a data frame of gene info; and a scale factor.

Examples

```
data("TOY_mrna_se")

get_DIIwt(TOY_mrna_se)
```

`get_DR` *Get a decay rate for genes and samples (for a genelist)*

Description

Get a decay rate for genes and samples (for a genelist)

Usage

```
get_DR(genelist, pileupPath, sampleInfo, cases = NULL, nCores = 2)
```

Arguments

`genelist`	a vector of gene names
`pileupPath`	file paths of coverage pileupData including .RData file names
`sampleInfo`	a sample information table including sample id. The number of rows is equal to the number of samples.
`cases`	a vector of specific samples among all samples in pileup. If NULL, all samples are selected. Default is NULL.
`nCores`	the number of cores for parallel computing. Default is 2.

Value

DR is a the number of genes x the number of samples matrix.

References

Choi, H.Y., Jo, H., Zhao, X. et al. SCISSOR: a framework for identifying structural changes in RNA transcripts. Nat Commun 12, 286 (2021).

Examples

```
## NOTE:
## This example demonstrates the function interface only.
## Meaningful results require coverage pileup files generated
## from BAM files (see vignette for a full workflow).
data("TOY_mrna_mat")

## Interface-only example (no meaningful output is produced)
try(
    get_DR(
        genelist   = TOY_mrna_mat$genes,
```

```
        pileupPath = rep(NA, length(TOY_mrna_mat$genes)),
        sampleInfo = data.frame(SampleID=TOY_mrna_mat$samples),
        nCores     = 2
    ),
    silent=TRUE
)
```

---

get_MCD *Get a mean coverage depth for genes and samples (for a genelist)*

Description

Get a mean coverage depth for genes and samples (for a genelist)

Usage

```
get_MCD(genelist, pileupPath, sampleInfo, cases = NULL, nCores = 2)
```

Arguments

genelist	a vector of gene names
pileupPath	file paths of coverage pileupData including .RData file names
sampleInfo	a sample information table including sample id. The number of rows is equal to the number of samples.
cases	a vector of specific samples among all samples in pileup. If NULL, all samples are selected. Default is NULL.
nCores	the number of cores for parallel computing. Default is 2.

Value

MCD is a the number of genes x the number of samples matrix.

Examples

```
## NOTE:
## This example demonstrates the function interface only.
## Meaningful results require coverage pileup files generated
## from BAM files (see vignette for a full workflow).
data("TOY_total_mat")

## Interface-only example (no meaningful output is produced)
try(
    get_MCD(
        genelist   = TOY_total_mat$genes,
        pileupPath = rep(NA, length(TOY_total_mat$genes)),
        sampleInfo = data.frame(SampleID=TOY_total_mat$samples),
        nCores     = 2
    ),
    silent=TRUE
)
```

get_pileupExon *Get a focused pileup of exon location (for single-study)*

Description

Get a focused pileup of exon location (for single-study)

Usage

```
get_pileupExon(g, pileupPath, cases = NULL)
```

Arguments

g	the gene order in genelist
pileupPath	file paths of coverage pileupData including .RData file names
cases	a vector of specific samples among all samples in pileup. If NULL, all samples are selected. Default is NULL.

Value

a focused pileup is a the number of exon locations x the number of samples matrix for the g-th gene.

Examples

```
## Requires a base-level coverage pileup .RData file
try(
   get_pileupExon(g=1, pileupPath=c("path/to/pileup1.RData", "path/to/pileup2.RData")), silent=TRUE
)
```

get_SOI *Get a suboptimal/optimal index for samples*

Description

Get a suboptimal/optimal index for samples

Usage

```
get_SOI(
  MCD,
  wCV = NULL,
  rstPct = 20,
  obsPct = 50,
  cutoff = 3,
  assay.MCD = "MCD",
  assay.wCV = "wCV"
)
```

Arguments

MCD	a mean coverage depth is a the number of genes x the number of samples matrix.
wCV	a window coefficient of variation is a the number of genes x the number of samples matrix.
rstPct	restricted percent (one-side) to restrict genes by log transformed MC. Default is 20.
obsPct	span includes the percent of observations in each local regression. Default is 50.
cutoff	numeric threshold on projection depth used to classify samples.
assay.MCD	character string specifying the assay name containing the MCD matrix in a SummarizedExperiment object.
assay.wCV	character string specifying the assay name containing the wCV matrix in a SummarizedExperiment object.

Value

a matrix including a vector of SOI; a coordinate matrix of smoothed data; and a range of MCD.

Examples

```
data("TOY_total_se")

get_SOI(TOY_total_se)
```

---

get_wCV *Get a window coefficient of variation for genes and samples (for a genelist)*

Description

Get a window coefficient of variation for genes and samples (for a genelist)

Usage

```
get_wCV(
  genelist,
  pileupPath,
  sampleInfo,
  rnum = 100,
  method = 1,
  winSize = 20,
  egPct = 10,
  cases = NULL,
  nCores = 2
)
```

Arguments

genelist	a vector of gene names
pileupPath	file paths of coverage pileupData including .RData file names
sampleInfo	a sample information table including sample id. The number of rows is equal to the number of samples.
rnum	the number of regions for uniformly dividing the x-axis for gene length normalization. Default is 100.
method	1 and 2 return the raw read depth and the interpolated read depth at the normalized genomic position, respectively. Default is 1.
winSize	window size of the rolling window. Default is 20.
egPct	edge percent (one-side) to calculate the trimmed mean. Default is 10.
cases	a vector of specific samples among all samples in pileup. If NULL, all samples are selected. Default is NULL.
nCores	the number of cores for parallel computing. Default is 2.

Value

wCV is a the number of genes x the number of samples matrix.

Examples

```
## NOTE:
## This example demonstrates the function interface only.
## Meaningful results require coverage pileup files generated
## from BAM files (see vignette for a full workflow).
data("TOY_total_mat")

## Interface-only example (no meaningful output is produced)
try(
    get_wCV(
        genelist   = TOY_total_mat$genes,
        pileupPath = rep(NA, length(TOY_total_mat$genes)),
        sampleInfo = data.frame(SampleID=TOY_total_mat$samples),
        nCores     = 2
    ),
    silent=TRUE
)
```

---

plot_DIIwt *Plot degraded/intact index outputs*

Description

Plot degraded/intact index outputs

Usage

```
plot_DIIwt(DR, DIIresult, cutoff = 3, outFile = NULL)
```

Arguments

DR	a the number of genes x the number of samples matrix of decay rates
DIIresult	outputs from get_DII function
cutoff	numeric threshold on projection depth used to classify samples.
outFile	a directory with a file name to save outputs. Default is NULL.

Value

figures for the distribution of DII by PD; and the heatmap of DR.

References

https://jtr13.github.io/cc21fall2/raincloud-plot-101-density-plot-or-boxplotwhy-not-do-both.html

Examples

```
data("TOY_mrna_se")

res <- get_DIIwt(TOY_mrna_se)

try(
    plot_DIIwt(
        DR        = TOY_mrna_se$DR,
        DIIresult = res,
        outFile   = tempfile(fileext=".png")
    ),
    silent=TRUE
)
```

---

plot_GBC *Plot gene body coverage*

Description

Plot gene body coverage

Usage

```
plot_GBC(
  pileupPath,
  geneNames,
  rnum = 100,
  method = 1,
```

```
  scale = TRUE,
  stat = 2,
  plot = TRUE,
  sampleInfo
)
```

Arguments

pileupPath	file paths of coverage pileupData including .RData file names
geneNames	gene names per file. If NULL, Gene i with the same length of pileupPath be set. Default is NULL.
rnum	the number of regions for uniformly dividing the x-axis. Default is 100.
method	1 and 2 return the raw read depth and the interpolated read depth at the normalized genomic position, respectively. Default is 1.
scale	TRUE/FALSE returns the scaled/unscaled normalized transcript coverage. Default is TRUE.
stat	1 and 2 return median and mean normalized coverage curves per sample, respectively. Default is 1.
plot	TRUE/FALSE turns on/off the normalized transcript coverage plot. Default is TRUE.
sampleInfo	a sample information table including sample id. The number of rows is equal to the number of samples.

Value

a matrix and a plot, or a matrix for the gene body coverage where plot is TRUE or FALSE, respectively.

Examples

```
## Interface-only example
try(
    plot_GBC(
        pileupPath = NA,
        geneNames  = "GENE1",
        sampleInfo = data.frame(
            SampleID       = c("S1", "S2"),
            CODING_BASES   = c(1, 1),
            INTRONIC_BASES = c(1, 1)
        ),
        plot       = FALSE
    ),
    silent=TRUE
)
```

plot_GBCos *Plot gene body coverage with optimal samples*

Description

Plot gene body coverage with optimal samples

Usage

```
plot_GBCos(stat = 2, plot = TRUE, sampleInfo, GBCresult, auc.vec)
```

Arguments

stat	1 and 2 return median and mean normalized coverage curves per sample, respectively. Default is 1.
plot	TRUE/FALSE turns on/off the normalized transcript coverage plot. Default is TRUE.
sampleInfo	a sample information table including sample id. The number of rows is equal to the number of samples.
GBCresult	results of the gene body coverage with all samples
auc.vec	a vector with SOI per sample

Value

a matrix and a plot, or a matrix for the gene body coverage where plot is TRUE or FALSE, respectively.

Examples

```
## Interface-only example
GBCresult <- list(
    GBP=data.frame(
        region      = 1:2,
        sample      = c("S1", "S2"),
        scale.geom  = c(1, 1),
        RatioIntron = c(1, 1)
    )
)

auc.vec <- data.frame(
    Sample = c("S1", "S2"),
    PD     = c(0, 0),
    SOI    = c("Optimal", "Optimal")
)

try(
    plot_GBCos(
        sampleInfo = data.frame(SampleID=c("S1", "S2")),
```

```
        GBCresult  = GBCresult,
        auc.vec    = auc.vec,
        plot       = FALSE
    ),
    silent=TRUE
)
```

---

plot_SOI *Plot suboptimal/optimal index outputs*

Description

Plot suboptimal/optimal index outputs

Usage

```
plot_SOI(SOIresult, cutoff = 3, outFile = NULL)
```

Arguments

SOIresult	outputs from get_SOI function
cutoff	numeric threshold on projection depth used to classify samples.
outFile	a directory with a file name to save outputs. Default is NULL.

Value

figures for the distribution of SOI by PD; and the relation of wCV and MCD.

References

https://jtr13.github.io/cc21fall2/raincloud-plot-101-density-plot-or-boxplotwhy-not-do-both.html

Examples

```
data("TOY_total_se")

res <- get_SOI(TOY_total_se)

plot_SOI(SOIresult=res, outFile=tempfile(fileext=".png"))
```

TOY_mrna_mat *Toy mRNA-seq-like dataset for RNAshapeQC (matrix input)*

Description

A small synthetic dataset mimicking mRNA-seq coverage-based quality control (QC) inputs. It is used in the vignette to demonstrate degradation-based metrics such as decay rate (DR), degradation score (DS), and the degraded/intact index (DII).

Format

A list with 6 components:

DR A numeric matrix of decay rates; genes in rows and samples in columns. In this toy dataset, it is a 100 (genes) x 10 (samples) matrix with row names like `"Gene001"` and column names like `"T01"`.

genes A character vector of length 100 containing gene IDs used as row names in `DR` and `TPM`.

samples A character vector of length 10 containing sample IDs used as column names in `DR` and `TPM`.

protocol A single character string indicating the protocol used, here `"mRNA-seq"`.

TPM A numeric matrix of TPM values; same dimension and dimnames as `DR`.

genelength A gene length (bp) vector with names matching the row names of `DR`.

Details

All values are synthetic and were generated solely for demonstration and testing. They do not correspond to any real samples or cohorts.

Examples

```
data("TOY_mrna_mat")
str(TOY_mrna_mat)
```

TOY_mrna_se *Toy mRNA-seq-like dataset for RNAshapeQC (SE input)*

Description

A small synthetic `SummarizedExperiment` object mimicking mRNA-seq coverage-based quality control (QC) inputs. It is used in the vignette to demonstrate degradation-based metrics such as decay rate (DR), degradation score (DS), and the degraded/intact index (DII).

Format

A SummarizedExperiment object with:

assays Two matrices:

- **DR** A numeric matrix of decay rates (genes x samples).
- **TPM** A numeric matrix of TPM expression values with the same dimension and dimnames as DR.

rowData A DataFrame containing gene-level metadata, including gene_length (bp).

Details

The dataset contains 100 synthetic genes and 10 synthetic samples. All values were generated solely for demonstration and testing purposes and do not correspond to real biological data.

Examples

```
data(TOY_mrna_se)
TOY_mrna_se
```

TOY_total_mat *Toy total RNA-seq-like dataset for RNAshapeQC (matrix input)*

Description

A small synthetic dataset mimicking total RNA-seq coverage-based quality control (QC) inputs. It is used in the vignette to demonstrate coverage-shape metrics such as mean coverage depth (MCD), window coefficient of variation (wCV), and the suboptimal/optimal index (SOI).

Format

A list with 5 components:

MCD A numeric matrix of mean coverage depth; genes in rows and samples in columns. In this toy dataset, it is a 100 (genes) x 10 (samples) matrix with row names like "Gene001" and column names like "A01".

wCV A numeric matrix of window coefficients of variation; same dimension and dimnames as MCD.

genes A character vector of length 100 containing gene IDs used as row names in MCD and wCV.

samples A character vector of length 10 containing sample IDs used as column names in MCD and wCV.

protocol A single character string indicating the protocol used, here "total RNA-seq".

Details

All values are synthetic and were generated solely for demonstration and testing. They do not correspond to any real samples or cohorts.

Examples

```
data("TOY_total_mat")
str(TOY_total_mat)
```

`TOY_total_se` *Toy total RNA-seq-like dataset for RNAshapeQC (SE input)*

Description

A small synthetic `SummarizedExperiment` object mimicking total RNA-seq coverage-based quality control (QC) inputs. It is used in the vignette to demonstrate coverage-shape metrics such as mean coverage depth (MCD), window coefficient of variation (wCV), and the suboptimal/optimal index (SOI).

Format

A `SummarizedExperiment` object with:

assays Two matrices:

- **MCD** A numeric matrix of mean coverage depth (genes x samples).
- **wCV** A numeric matrix of window coefficients of variation with the same dimension and dimnames as `MCD`.

Details

The dataset contains 100 synthetic genes and 10 synthetic samples. All values were generated solely for demonstration and testing purposes and do not correspond to real biological data.

Examples

```
data(TOY_total_se)
TOY_total_se
```

Index

∗ **datasets**
    TOY_mrna_mat, 26
    TOY_mrna_se, 26
    TOY_total_mat, 27
    TOY_total_se, 28
.build_pileupExon, 3

combine_vecObj, 4
compute_DIIwt, 5
compute_DR, 6
compute_MCD, 7
compute_SOI, 8
compute_wCV, 9
construct_pileup, 10

extract_RData, 11

filter_lowExpGenes, 12

gen_DR, 12
gen_MCD, 13
gen_wCV, 14
get_DIIhc, 15
get_DIIwt, 16
get_DR, 17
get_MCD, 18
get_pileupExon, 19
get_SOI, 19
get_wCV, 20

plot_DIIwt, 21
plot_GBC, 22
plot_GBCos, 24
plot_SOI, 25

TOY_mrna_mat, 26
TOY_mrna_se, 26
TOY_total_mat, 27
TOY_total_se, 28