Package 'conumee'

December 23, 2024

Title Enhanced copy-number variation analysis using Illumina DNA methylation arrays

Version 1.41.0

Author Volker Hovestadt, Marc Zapatka

Maintainer Volker Hovestadt <conumee@hovestadt.bio>

Address Division of Molecular Genetics, German Cancer Research Center (DKFZ), Heidelberg, Germany

Description This package contains a set of processing and plotting methods for performing copy-number variation (CNV) analysis using Illumina 450k or EPIC methylation arrays.

Imports methods, stats, DNAcopy, rtracklayer, GenomicRanges, IRanges, GenomeInfoDb

Depends R (>= 3.0), minfi, IlluminaHumanMethylation450kanno.ilmn12.hg19, IlluminaHumanMethylation450kmanifest, IlluminaHumanMethylationEPICanno.ilm10b2.hg19, IlluminaHumanMethylationEPICmanifest

Suggests BiocStyle, knitr, rmarkdown, minfiData, RCurl

License GPL (>= 2)

LazyData false

Collate classes.R annotation.R load.R process.R output.R data.R

biocViews CopyNumberVariation, DNAMethylation, MethylationArray, Microarray, Normalization, Preprocessing, QualityControl, Software

VignetteBuilder knitr

RoxygenNote 6.0.1

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

git_branch devel

git_last_commit f47752d

git_last_commit_date 2024-10-29

Repository Bioconductor 3.21

Date/Publication 2024-12-23

Contents

`CNV.analysis-class` *CNV.analysis class*

Description

CNV analysis data of a single sample is stored in this class

Usage

```
## S4 method for signature 'CNV.analysis'
show(object)

## S4 method for signature 'CNV.analysis'
names(x)

## S4 replacement method for signature 'CNV.analysis'
names(x) <- value

## S4 method for signature 'CNV.analysis'
coef(object)
```

Arguments

object	CNV.analysis object
x	CNV.analysis object (defined by show generic).
value	Replacement names.

Details

Use CNV.fit to create. Modified by CNV.bin, CNV.detail and CNV.segment.

Value

CNV.analysis class.

Author(s)

Volker Hovestadt <conumee@hovestadt.bio>

Examples

```
# prepare
library(minfiData)
data(MsetEx)
d <- CNV.load(MsetEx)
anno <- CNV.create_anno()

# create object
x <- CNV.fit(query = d['GroupB_1'], ref = d[c('GroupA_1', 'GroupA_2', 'GroupA_3')], anno)

# modify object
x <- CNV.bin(x)
x <- CNV.detail(x)
x <- CNV.segment(x)

# general information
x
show(x)

# coefficients of linear regression
coef(x)

# show or replace sample name
names(x)
names(x) <- 'Sample 1'

# output plots
CNV.genomeplot(x)
CNV.genomeplot(x, chr = 'chr6')
#CNV.detailplot(x, name = 'MYCN')
#CNV.detailplot_wrap(x)
CNV.write(x, what = 'segments')
```

CNV.anno-class *CNV.anno class*

Description

Annotations required for CNV analysis are stored in this class.

Usage

```
## S4 method for signature 'CNV.anno'
show(object)
```

Arguments

object	CNV.anno object

Details

This class does not contain any sample data. Use CNV.create_anno to create.

Value

CNV.anno class.

Author(s)

Volker Hovestadt <conumee@hovestadt.bio>

Examples

```
# create object
anno <- CNV.create_anno()

# general information
anno
show(anno)
```

CNV.bin *CNV.bin*

Description

Combine single probe intensitiy values into predefined bins.

Usage

```
CNV.bin(object, ...)

## S4 method for signature 'CNV.analysis'
CNV.bin(object)
```

Arguments

object	CNV.analysis object.
...	Additional parameters (CNV.bin generic, currently not used).

Details

The median intensity per bin is calculated. Bins are defined using CNV.create_anno. A value by which all probe and bin intensity values are shifted in subsequent analysis steps is calculated by minimizing the median absolute deviation from all bins to zero (ideally shifting the copy-neutral state to 0).

Value

CNV.analysis object.

Author(s)

Volker Hovestadt <conumee@hovestadt.bio>

Examples

```
# prepare
library(minfiData)
data(MsetEx)
d <- CNV.load(MsetEx)
data(detail_regions)
anno <- CNV.create_anno(detail_regions = detail_regions)

# create object
x <- CNV.fit(query = d['GroupB_1'], ref = d[c('GroupA_1', 'GroupA_2', 'GroupA_3')], anno)

# modify object
x <- CNV.bin(x)
#x <- CNV.detail(x)
```

```
#x <- CNV.segment(x)

# general information
x
show(x)

# coefficients of linear regression
coef(x)

# show or replace sample name
names(x)
names(x) <- 'Sample 1'
```

CNV.check *CNV.check*

Description

Check intensity values.

Usage

```
CNV.check(object)

## S4 method for signature 'CNV.data'
CNV.check(object)
```

Arguments

object	CNV.data object.

Details

This method checks if intensities are positive and not NA. If not, they are set to 1. Warnings are given if intensities are abnormally high or low (> 50000 or < 5000, respectively).

Value

CNV.data object.

Author(s)

Volker Hovestadt <conumee@hovestadt.bio>

CNV.create_anno *CNV.create_anno*

Description

Create annotations for CNV analysis.

Usage

```
CNV.create_anno(bin_minprobes = 15, bin_minsize = 50000,
  bin_maxsize = 5000000, array_type = "450k", chrXY = FALSE,
  exclude_regions = NULL, detail_regions = NULL)
```

Arguments

bin_minprobes	numeric. Minimum number of probes per bin. Bins are interatively merged with neighboring bin until minimum number is reached.
bin_minsize	numeric. Minimum size of a bin.
bin_maxsize	numeric. Maximum size of a bin. Merged bins that are larger are filtered out.
array_type	character. One of 450k, EPIC, or overlap. Defaults to 450k.
chrXY	logical. Should chromosome X and Y be included in the analysis?
exclude_regions	GRanges object or path to bed file containing genomic regions to be excluded.
detail_regions	GRanges object or path to bed file containing genomic regions to be examined in detail.

Details

This function collects all annotations required for CNV analysis using Illumina 450k or EPIC arrays. The output CNV.anno object is not editable. Rerun CNV.create_anno to change parameters.

Value

CNV.anno object.

Author(s)

Volker Hovestadt <conumee@hovestadt.bio>

Examples

```
# create annotation object
anno <- CNV.create_anno()
anno
```

CNV.create_bins *CNV.create_bins*

Description

Split genome into bins of defined size.

Usage

```
CNV.create_bins(hg19.anno, bin_minsize = 50000, hg19.gap, hg19.exclude)
```

Arguments

hg19.anno	foo
bin_minsize	foo
hg19.gap	foo
hg19.exclude	foo

Value

GRanges object.

CNV.data-class *CNV.data class*

Description

Intensities of one or multiple samples are stored in this class.

Usage

```
## S4 method for signature 'CNV.data'
show(object)

## S4 method for signature 'CNV.data,ANY,ANY,ANY'
x[i]

## S4 method for signature 'CNV.data'
names(x)

## S4 replacement method for signature 'CNV.data'
names(x) <- value
```

Arguments

object	CNV.data object
x	CNV.data object (defined by Extract generic).
i	index. logical, numeric or character.
value	Replacement names.

Details

Use CNV.load to create.

Value

CNV.data class.

Author(s)

Volker Hovestadt <conumee@hovestadt.bio>

Examples

```
# create object
library(minfiData)
data(MsetEx)

d <- CNV.load(MsetEx)

# general information
d
show(d)

# show or replace sample names
names(d)
names(d) <- toupper(names(d))

# subset samples
d[1:2]
```

CNV.detail *CNV.detail*

Description

Combine single probe values within detail regions.

Usage

```
CNV.detail(object, ...)

## S4 method for signature 'CNV.analysis'
CNV.detail(object)
```

Arguments

object	CNV.analysis object.
...	Additional parameters (CNV.detail generic, currently not used).

Details

The median intensity per detail region is calculated. Detail regions are defined using CNV.create_anno(detail_bed=)

Value

CNV.analysis object.

Author(s)

Volker Hovestadt <conumee@hovestadt.bio>

Examples

```
# prepare
library(minfiData)
data(MsetEx)
d <- CNV.load(MsetEx)
data(detail_regions)
anno <- CNV.create_anno(detail_regions = detail_regions)

# create object
x <- CNV.fit(query = d['GroupB_1'], ref = d[c('GroupA_1', 'GroupA_2', 'GroupA_3')], anno)

# modify object
x <- CNV.bin(x)
x <- CNV.detail(x)
#x <- CNV.segment(x)

# general information
x
show(x)

# coefficients of linear regression
coef(x)

# show or replace sample name
names(x)
names(x) <- 'Sample 1'
```

CNV.detailplot *CNV.detailplot*

Description

Create CNV plot for detail region.

Usage

```
CNV.detailplot(object, ...)

## S4 method for signature 'CNV.analysis'
CNV.detailplot(object, name, yaxt = "l",
  ylim = c(-1.25, 1.25), set_par = TRUE, cols = c("red", "red",
  "lightgrey", "green", "green"))
```

Arguments

object	CNV.analysis object.
...	Additional parameters (CNV.detailplot generic, currently not used).
name	character. Name of detail region to plot.
yaxt	character. Include y-axis? 'l': left, 'r': right, 'n': no. Defaults to 'l'.
ylim	numeric vector. The y limits of the plot. Defaults to c(-1.25, 1.25).
set_par	logical. Use recommended graphical parameters for oma and mar? Defaults to TRUE. Original parameters are restored afterwards.
cols	character vector. Colors to use for plotting intensity levels of bins. Centered around 0. Defaults to c('red', 'red', 'lightgrey', 'green', 'green').

Details

This method provides the functionality for generating detail regions CNV plots. Probes are shown as dots, bins are shown as lines. See parameters for more information.

Value

NULL.

Author(s)

Volker Hovestadt <conumee@hovestadt.bio>

Examples

```
# prepare
library(minfiData)
data(MsetEx)
d <- CNV.load(MsetEx)
data(detail_regions)
anno <- CNV.create_anno(detail_regions = detail_regions)

# create/modify object
x <- CNV.segment(CNV.detail(CNV.bin(CNV.fit(query = d['GroupB_1'],
    ref = d[c('GroupA_1', 'GroupA_2', 'GroupA_3')], anno))))

# output plots
CNV.genomeplot(x)
CNV.genomeplot(x, chr = 'chr6')
CNV.detailplot(x, name = 'PTEN')
CNV.detailplot_wrap(x)

# output text files
CNV.write(x, what = 'segments')
CNV.write(x, what = 'detail')
CNV.write(x, what = 'bins')
CNV.write(x, what = 'probes')
```

`CNV.detailplot_wrap` *CNV.detailplot_wrap*

Description

Create CNV plot for all detail regions.

Usage

```
CNV.detailplot_wrap(object, ...)

## S4 method for signature 'CNV.analysis'
CNV.detailplot_wrap(object, set_par = TRUE,
  main = NULL, ...)
```

Arguments

`object`	`CNV.analysis` object.
`...`	Additional paramters supplied to `CNV.detailplot`.
`set_par`	logical. Use recommended graphical parameters for `oma` and `mar`? Defaults to `TRUE`. Original parameters are restored afterwards.
`main`	character. Title of the plot. Defaults to sample name.

Details

This method is a wrapper of the CNV.detailplot method to plot all detail regions.

Value

NULL.

Author(s)

Volker Hovestadt <conumee@hovestadt.bio>

Examples

```
# prepare
library(minfiData)
data(MsetEx)
d <- CNV.load(MsetEx)
data(detail_regions)
anno <- CNV.create_anno(detail_regions = detail_regions)

# create/modify object
x <- CNV.segment(CNV.detail(CNV.bin(CNV.fit(query = d['GroupB_1'],
    ref = d[c('GroupA_1', 'GroupA_2', 'GroupA_3')], anno))))

# output plots
CNV.genomeplot(x)
CNV.genomeplot(x, chr = 'chr6')
CNV.detailplot(x, name = 'PTEN')
CNV.detailplot_wrap(x)

# output text files
CNV.write(x, what = 'segments')
CNV.write(x, what = 'detail')
CNV.write(x, what = 'bins')
CNV.write(x, what = 'probes')
```

CNV.fit *CNV.fit*

Description

Normalize query sample intensities by fitting intensities to reference set using a linear regression model.

Usage

```
CNV.fit(query, ref, anno, ...)

## S4 method for signature 'CNV.data,CNV.data,CNV.anno'
CNV.fit(query, ref, anno, name = NULL,
  intercept = TRUE)
```

Arguments

query	CNV.data object of query sample (single sample).
ref	CNV.data object of reference set.
anno	CNV.anno object. Use CNV.create_anno do create.
...	Additional parameters (CNV.fit generic, currently not used).
name	character. Optional parameter to set query sample name.
intercept	logical. Should intercept be considered? Defaults to TRUE.

Details

The log2 ratio of query intensities versus a linear combination of reference set intensities that best reflects query intensities is calculated (as determined by linear regression). The annotations provided to CNV.fit are saved within the returned CNV.analysis object and used for subsequent analysis steps.

Value

CNV.analysis object.

Author(s)

Volker Hovestadt <conumee@hovestadt.bio>

Examples

```
# prepare
library(minfiData)
data(MsetEx)
d <- CNV.load(MsetEx)
data(detail_regions)
anno <- CNV.create_anno(detail_regions = detail_regions)

# create object
x <- CNV.fit(query = d['GroupB_1'], ref = d[c('GroupA_1', 'GroupA_2', 'GroupA_3')], anno)

# modify object
#x <- CNV.bin(x)
#x <- CNV.detail(x)
#x <- CNV.segment(x)

# general information
```

```
x
show(x)

# coefficients of linear regression
coef(x)

# show or replace sample name
names(x)
names(x) <- 'Sample 1'
```

CNV.genomeplot *CNV.genomeplot*

Description

Create CNV plot for the whole genome or chromosomes.

Usage

```
CNV.genomeplot(object, ...)

## S4 method for signature 'CNV.analysis'
CNV.genomeplot(object, chr = "all", chrX = TRUE,
  chrY = TRUE, centromere = TRUE, detail = TRUE, main = NULL,
  ylim = c(-1.25, 1.25), set_par = TRUE, cols = c("red", "red",
  "lightgrey", "green", "green"))
```

Arguments

object	CNV.analysis object.
...	Additional parameters (CNV.detailplot generic, currently not used).
chr	character vector. Which chromomsomes to plot. Defaults to 'all'.
chrX	logical. Plot values for chrX? Defaults to TRUE. Set CNV.create_anno(chrXY = FALSE) if chrX and Y should not be included at all.
chrY	logical. Plot values for chrY? Defaults to TRUE.
centromere	logical. Show dashed lines at centromeres? Defaults to TRUE.
detail	logical. If available, include labels of detail regions? Defaults to TRUE.
main	character. Title of the plot. Defaults to sample name.
ylim	numeric vector. The y limits of the plot. Defaults to c(-1.25, 1.25).
set_par	logical. Use recommended graphical parameters for oma and mar? Defaults to TRUE. Original parameters are restored afterwards.
cols	character vector. Colors to use for plotting intensity levels of bins. Centered around 0. Defaults to c('red', 'red', 'lightgrey', 'green', 'green').

Details

This method provides the functionality for generating CNV plots for the whole genome or defined chromosomes. Bins are shown as dots, segments are shown as lines. See parameters for more information.

Value

NULL.

Author(s)

Volker Hovestadt <conumee@hovestadt.bio>

Examples

```
# prepare
library(minfiData)
data(MsetEx)
d <- CNV.load(MsetEx)
data(detail_regions)
anno <- CNV.create_anno(detail_regions = detail_regions)

# create/modify object
x <- CNV.segment(CNV.detail(CNV.bin(CNV.fit(query = d['GroupB_1'],
    ref = d[c('GroupA_1', 'GroupA_2', 'GroupA_3')], anno))))

# output plots
CNV.genomeplot(x)
CNV.genomeplot(x, chr = 'chr6')
CNV.detailplot(x, name = 'PTEN')
CNV.detailplot_wrap(x)

# output text files
CNV.write(x, what = 'segments')
CNV.write(x, what = 'detail')
CNV.write(x, what = 'bins')
CNV.write(x, what = 'probes')
```

CNV.load *CNV.load*

Description

Prepare combined intensities from various input objects.

Usage

```
CNV.load(input, ...)

## S4 method for signature 'GenomicRatioSet'
CNV.load(input, names = NULL)

## S4 method for signature 'MethylSet'
CNV.load(input, names = NULL)

## S4 method for signature 'data.frame'
CNV.load(input, names = NULL)

## S4 method for signature 'matrix'
CNV.load(input, names = NULL)

## S4 method for signature 'numeric'
CNV.load(input, names = NULL)
```

Arguments

input	Object of MethylSet class (minfi package), data.frame class, matrix class or numeric class.
...	Additional parameters (CNV.load generic, currently not used).
names	Vector specifying sample names. If not supplied, colnames are used. For MethylSet input, the first column of pData(input) matching 'name' (grep) is used.

Details

This method gathers combined intensities of the Methylated and Unmethylated signals for all supplied probes. Probe IDs must be supplied as row names or in a seperate column named 'ID_REF' or 'TargetID'. If column names match 'intensity', only those columns are used. Else, if column names match 'signal' or 'methylated', only those columns are used. Otherwise, all columns are used.

Value

CNV.data object.

Author(s)

Volker Hovestadt <conumee@hovestadt.bio>

Examples

```
library(minfiData)
d <- CNV.load(MsetEx)
d
```

CNV.merge_bins *CNV.merge_bins*

Description

Merge bins containing less than the defined number probes with neighboring bin containing fewer probes.

Usage

```
CNV.merge_bins(hg19.anno, hg19.tile, bin_minprobes = 20, hg19.probes,
  bin_maxsize = 5e+06, verbose = FALSE)
```

Arguments

hg19.anno	foo
hg19.tile	foo
bin_minprobes	foo
hg19.probes	foo
bin_maxsize	foo
verbose	foo

Value

GRanges object.

CNV.process *CNV.process*

Description

Given a case index, control indices, CNV.data, and CNV.anno, along with hints about sex chromosomes, call CN for a sample.

Usage

```
CNV.process(case, controls, CNdata, anno)

## S4 method for signature 'integer,integer,CNV.data,CNV.anno'
CNV.process(case, controls,
  CNdata, anno)
```

Arguments

case	index of the case to process CN for.
controls	indices of the control samples.
CNdata	CNV.data object.
anno	CNV.anno object.

Details

This method wraps most of conumee, and tries to call sex chromosomes properly using chrX/chrY information derived from the source GenomicRatioSet. For female subjects, chrY is dropped.

Value

CNV.analysis object.

Author(s)

Tim Triche, Jr. <tim.triche@gmail.com>

CNV.segment *CNV.segment*

Description

Segment bin values (wrapper of DNAcopy package).

Usage

```
CNV.segment(object, ...)

## S4 method for signature 'CNV.analysis'
CNV.segment(object, alpha = 0.001, nperm = 50000,
  min.width = 5, undo.splits = "sdundo", undo.SD = 2.2, verbose = 0,
  ...)
```

Arguments

object	CNV.analysis object.
...	Additional parameters supplied to the segment method of the DNAcopy package.
alpha	See details. Defaults to 0.001.
nperm	See details. Defaults to 50000.
min.width	See details. Defaults to 5.
undo.splits	See details. Defaults to 'sdundo'.
undo.SD	See details. Defaults to 2.2.
verbose	See details. Defaults to 0.

Details

This method is a wrapper of the CNA, segment, segments.summary and segments.p methods of the DNAcopy package. Please refer to the respective man pages for more detailed information. The default parameters of `CNV.segment` override some of the default parameters of segment and are optimized for 450k data CNV analysis.

Value

`CNV.analysis` object.

Author(s)

Volker Hovestadt <conumee@hovestadt.bio>

Examples

```
# prepare
library(minfiData)
data(MsetEx)
d <- CNV.load(MsetEx)
data(detail_regions)
anno <- CNV.create_anno(detail_regions = detail_regions)

# create object
x <- CNV.fit(query = d['GroupB_1'], ref = d[c('GroupA_1', 'GroupA_2', 'GroupA_3')], anno)

# modify object
x <- CNV.bin(x)
x <- CNV.detail(x)
x <- CNV.segment(x)

# general information
x
show(x)

# coefficients of linear regression
coef(x)

# show or replace sample name
names(x)
names(x) <- 'Sample 1'
```

`CNV.write` *CNV.write*

Description

Output CNV analysis results as table.

Usage

```
CNV.write(object, ...)

## S4 method for signature 'CNV.analysis'
CNV.write(object, file = NULL, what = "segments")
```

Arguments

object	CNV.analysis object.
...	Additional parameters (CNV.write generic, currently not used).
file	Path where output file should be written to. Defaults to NULL: No file is written, table is returned as data.frame object.
what	character. This should be (an unambiguous abbreviation of) one of 'probes', 'bins', 'detail' or 'segments'. Defaults to 'segments'.

Value

if parameter file is not supplied, the table is returned as a data.frame object.

Examples

```
# prepare
library(minfiData)
data(MsetEx)
d <- CNV.load(MsetEx)
data(detail_regions)
anno <- CNV.create_anno(detail_regions = detail_regions)

# create/modify object
x <- CNV.segment(CNV.detail(CNV.bin(CNV.fit(query = d['GroupB_1'],
    ref = d[c('GroupA_1', 'GroupA_2', 'GroupA_3')], anno))))

# output plots
CNV.genomeplot(x)
CNV.genomeplot(x, chr = 'chr6')
CNV.detailplot(x, name = 'PTEN')
CNV.detailplot_wrap(x)

# output text files
CNV.write(x, what = 'segments')
CNV.write(x, what = 'detail')
CNV.write(x, what = 'bins')
CNV.write(x, what = 'probes')
```

`detail_regions` *detail_regions*

Description

Example of genomic regions to be analyzed in detail (e.g. candidate oncogenes/TSGs).

Details

Imported using `rtracklayer`. Raw data stored in `inst/extdata/detail_regions.bed`.

Author(s)

Volker Hovestadt <conumee@hovestadt.bio>

`exclude_regions` *exclude_regions*

Description

Example of genomic regions to exclude (e.g. known polymorphic regions).

Details

Imported using `rtracklayer`. Raw data stored in `inst/extdata/exclude_regions.bed`.

Author(s)

Volker Hovestadt <conumee@hovestadt.bio>

`read.450k.url` *read.450k.url*

Description

Read IDAT files from the web.

Usage

```
read.450k.url(url = NULL, idat = NULL)
```

Arguments

url	URL of the directory in which the IDAT files are located.
idat	Vector of IDAT names. url and idat default to the TCGA example described in the vignette.

Details

This method downloads the provided list of IDAT files to a temporary folder (using the RCurl package). It then uses the ‘read.450k.exp‘ method of the ‘minfi‘ package.

Value

RGChannelSet object.

Author(s)

Volker Hovestadt <conumee@hovestadt.bio>

Examples

```
RGsetTCGA <- read.450k.url()
```

---

tbl_ucsc *tbl_ucsc*

Description

UCSC tables required for creating annotation object.

Details

Imported using rtracklayer::browserSession('UCSC'): chromInfo, gap, cytoBand.

Author(s)

Volker Hovestadt <conumee@hovestadt.bio>

tcgaBRCA.sentrix2name *tcgaBRCA.sentrix2name*

Description

Named vector for Sentrix ID to TCGA ID conversion of breast cancer example data (see README).

Details

Based on https://tcga-data.nci.nih.gov/tcgafiles/ftp_auth/distro_ftpusers/anonymous/tumor/brca/cgcc/

Author(s)

Volker Hovestadt <conumee@hovestadt.bio>

Index