Package 'panelcn.mops'

November 6, 2025

Type Package

Title CNV detection tool for targeted NGS panel data

Version 1.33.0

Date 2025-07-28

Description CNV detection tool for targeted NGS panel data.
Extension of the cn.mops package.

License LGPL (>= 2.0)

LazyData TRUE

Imports GenomicRanges, Rsamtools, IRanges, S4Vectors, Seqinfo, grDevices

Depends R (>= 3.5), cn.mops, methods, utils, stats, graphics

biocViews Sequencing, CopyNumberVariation, CellBiology, GenomicVariation, VariantDetection, Genetics

RoxygenNote 6.0.1

NeedsCompilation no

Suggests knitr, rmarkdown, RUnit, BiocGenerics

VignetteBuilder knitr

git_url https://git.bioconductor.org/packages/panelcn.mops

git_branch devel

git_last_commit 5e38d91

git_last_commit_date 2025-10-29

Repository Bioconductor 3.23

Date/Publication 2025-11-05

Author Verena Haunschmid [aut], Gundula Povysil [aut, cre]

Maintainer Gundula Povysil <povysil@bioinf.jku.at>

Contents

control . . . 2
countBamListInGRanges . . . 3
countWindows . . . 3
createResultTable . . . 4
getWindows . . . 5
panelcn.mops . . . 5
plotBoxplot . . . 7
read.width . . . 8
resultlist . . . 8
runPanelcnMops . . . 9
splitROIs . . . 10
test . . . 11

Index **12**

`control` *GRanges object of countWindows with read counts for control samples as elementMetadata.*

Description

The object was created using the function countBamListInGRanges with the enclosed countWindows object, a subset of BAM files provided by the 1000 Genomes Project and the read.width parameter set to 150.

Details

Control data included in panelcn.mops

Author(s)

Gundula Povysil

Examples

```
data(panelcn.mops)
control
```

countBamListInGRanges *Get read counts for a list of BAM files and given count windows*

Description

Get read counts for a list of BAM files and given count windows

Usage

```
countBamListInGRanges(bam.files, countWindows, read.width = 150, ...)
```

Arguments

bam.files	list with absolute or relative paths to BAM files
countWindows	data.frame with contents of a BED file as returned by getWindows
read.width	read.width parameter for countBamInGRanges or FALSE if actual read width should be extracted from BAM file
...	additional parameters

Value

a GRanges object over the countWindows with read counts for each sample as elementMetadata

Examples

```
bed <- system.file("extdata/Genes_part.bed", package = "panelcn.mops")
countWindows <- getWindows(bed)
## Not run:
testbam <- "SAMPLE1.bam"
test <- countBamListInGRanges(countWindows = countWindows,
                                bam.files = testbam, read.width = 150)

## End(Not run)
```

countWindows *result object of getWindows - a data.frame with the contents of the provided BED file with an additional gene name and exon name column*

Description

Data included in panelcn.mops

Author(s)

Gundula Povysil

Examples

```
data(panelcn.mops)
countWindows
```

`createResultTable` *Creates a user readable result table for the test samples of the genes of interest*

Description

Creates a user readable result table for the test samples of the genes of interest

Usage

```
createResultTable(resultlist, XandCB, countWindows, selectedGenes = NULL,
  sampleNames)
```

Arguments

`resultlist`	result object of runPanelcnMops
`XandCB`	GRanges object of combined read counts of test samples and control samples as returned by getRCRanges or countBamListInGRanges
`countWindows`	data.frame with contents of a BED file as returned by getWindows
`selectedGenes`	vector of names of genes of interest that should be displayed or NULL if all genes are of interest. Default = NULL
`sampleNames`	names of the test samples (basename of the BAM files)

Value

a data.frame containing the results for the test samples within the genes of interest

Examples

```
data(panelcn.mops)
XandCB <- test
elementMetadata(XandCB) <- cbind(elementMetadata(XandCB),
                                 elementMetadata(control))
sampleNames <- colnames(elementMetadata(test))
selectedGenes <- "ATM"
resulttable <- createResultTable(resultlist = resultlist, XandCB = XandCB,
                                    countWindows = countWindows,
                                    selectedGenes = selectedGenes,
                                    sampleNames = sampleNames)
```

getWindows *Convert BED file into data.frame of count windows*

Description

Convert BED file into data.frame of count windows

Usage

```
getWindows(filename, chr = FALSE)
```

Arguments

filename	filename of the BED file with absolute or relative path (structure of BED file without header: chromosome, exon start, exon end, exon name)
chr	indicates whether naming contains chr prefix

Value

a data.frame with the contents of the BED file with an additional gene name and exon name column

Examples

```
bed <- list.files(system.file("extdata", package = "panelcn.mops"),
                        pattern = ".bed$", full.names = TRUE)
countWindows <- getWindows(bed)
```

panelcn.mops *Core copy number detection algorithm for targeted NGS panel data*

Description

This function performs the cn.mops algorithm for copy number detection in NGS data adjusted to targeted NGS panel data including the second quality control.

Usage

```
panelcn.mops(input, testi = 1, geneInd = NULL, classes = c("CN0", "CN1",
  "CN2", "CN3", "CN4"), I = c(0.025, 0.5, 1, 1.5, 2), priorImpact = 1,
  cyc = 20, normType = "quant", sizeFactor = "quant", qu = 0.25,
  quSizeFactor = 0.75, norm = 1, minReadCount = 5, maxControls = 25,
  corrThresh = 0.99, useMedian = FALSE, returnPosterior = TRUE)
```

Arguments

input	either an instance of "GRanges" or a raw data matrix, where columns are interpreted as samples and rows as genomic regions. An entry is the read count of a sample in the genomic region.
testi	positive integer that gives the index of the test sample in input. Default = 1
geneInd	vector of indices of rows input that are within target genes. These regions are not considered for chosing correlated reference samples. If NULL, all regions are considered for the correlation. Default = NULL
classes	vector of characters of the same length as the parameter vector "I". One vector element must be named "CN2". The names reflect the labels of the copy number classes. Default = c("CN0","CN1","CN2","CN3","CN4").
I	vector of positive real values containing the expected fold change of the copy number classes. Length of this vector must be equal to the length of the "classes" parameter vector. For human copy number polymorphisms the default is c(0.025,0.5,1,1.5,2).
priorImpact	positive real value that reflects how strong the prior assumption affects the result. The higher the value the more samples will be assumed to have copy number 2. Default = 1.
cyc	positive integer that sets the number of cycles for the algorithm. Usually after less than 15 cycles convergence is reached. Default = 20.
normType	type of the normalization technique. Each samples' read counts are scaled such that the total number of reads are comparable across samples. Options are "mean", "median", "poisson", "quant", and "mode". Default = "quant".
sizeFactor	parameter for calculating the size factors for normalization. Options are "mean", "median", "quant", and "mode". Default = "quant".
qu	Quantile of the normType if normType is set to "quant". Real value between 0 and 1. Default = 0.25.
quSizeFactor	Quantile of the sizeFactor if sizeFactor is set to "quant". 0.75 corresponds to "upper quartile normalization". Real value between 0 and 1. Default = 0.75.
norm	the normalization strategy to be used. If set to 0 the read counts are not normalized and cn.mops does not model different coverages. If set to 1 the read counts are normalized. If set to 2 the read counts are not normalized and cn.mops models different coverages. Default = 1.
minReadCount	if all samples are below this value the algorithm will return the prior knowledge. This prevents that the algorithm from being applied to segments with very low coverage. Default = 5.
maxControls	integer reflecting the maximal numbers of controls to use. If set to 0 all highly correlated controls are used. Default = 25
corrThresh	threshold for selecting highly correlated controls. Default = 0.99
useMedian	flag indicating whether "median" instead of "mean" of a segment should be used for the CNV call. Default = FALSE.
returnPosterior	flag that decides whether the posterior probabilities should be returned. The posterior probabilities have a dimension of samples times copy number states times genomic regions and therefore consume a lot of memory. Default = TRUE.

Value

an instance of "CNVDetectionResult".

Examples

```
data(panelcn.mops)
XandCB <- test
elementMetadata(XandCB) <- cbind(elementMetadata(XandCB),
                                 elementMetadata(control))
result <- panelcn.mops(XandCB)
```

`plotBoxplot` *Create box plot of normalized read counts*

Description

Create box plot of normalized read counts

Usage

```
plotBoxplot(result, sampleName, countWindows, selectedGenes = NULL,
  showGene = 1, showLegend = TRUE, exonRange = NULL, ylimup = 1.15,
  thresh = 0)
```

Arguments

`result`	result object of panelcn.mops
`sampleName`	name of the test sample that should be displayed
`countWindows`	data.frame with contents of a BED file as returned by getWindows
`selectedGenes`	vector of names of genes of interest that should be displayed or NULL if all genes are of interest. Default = NULL
`showGene`	integer indicating which of the genes of interest to plot
`showLegend`	flag to indicate whether to display a legend with the names of the test samples. Default = TRUE
`exonRange`	vector of 2 positive integers to limit box plot to a certain range of exons or NULL
`ylimup`	numeric, maximum RC is multiplied by this value to calculate second value of ylim. Default = 1.15
`thresh`	numeric threshold for plotting fold change areas E.g. thresh = 0.4 plots a green rectangle above (1 + 0.4)*median for each boxplot and a red rectangle below (1 - 0.4)*median. Default of zero does not plot any colored areas.

Value

generates a boxplot of the normalized read counts

Examples

```
data(panelcn.mops)
sampleNames <- colnames(elementMetadata(test))
selectedGenes <- "ATM"
plotBoxplot(result = resultlist[[1]], sampleName = sampleNames[1],
            countWindows = countWindows, selectedGenes = selectedGenes,
            showGene = 1)
```

`read.width` *read width used for calculating RCs of test and control*

Description

Data included in panelcn.mops

Author(s)

Gundula Povysil

Examples

```
data(panelcn.mops)
read.width
```

`resultlist` *result object of runPanelcnMops - a list of instances of "CNVDetectionResult"*

Description

Result data included in panelcn.mops

Author(s)

Gundula Povysil

Examples

```
data(panelcn.mops)
resultlist
```

`runPanelcnMops` *Full copy number detection for targeted NGS panel data for multiple samples*

Description

This function performs first quality control and runs panelcn.mops for CNV detection on all test samples.

Usage

```
runPanelcnMops(XandCB, testiv = c(1), countWindows, selectedGenes = NULL,
  I = c(0.025, 0.57, 1, 1.46, 2), normType = "quant",
  sizeFactor = "quant", qu = 0.25, quSizeFactor = 0.75, norm = 1,
  priorImpact = 1, minMedianRC = 30, maxControls = 25,
  corrThresh = 0.99, sex = "mixed")
```

Arguments

`XandCB`	GRanges object of combined read counts of test samples and control samples as returned by countBamListInGRanges
`testiv`	vector of indices of test samples in XandCB. Default = c(1)
`countWindows`	data.frame with contents of a BED file as returned by getWindows
`selectedGenes`	vector of names of genes of interest or NULL if all genes are of interest. Default = NULL
`I`	vector of positive real values containing the expected fold change of the copy number classes. Length of this vector must be equal to the length of the "classes" parameter vector. For targeted NGS panel data the default is c(0.025,0.57,1,1.46,2)
`normType`	type of the normalization technique. Each samples' read counts are scaled such that the total number of reads are comparable across samples. Options are "mean","median","poisson", "quant", and "mode" Default = "quant"
`sizeFactor`	parameter for calculating the size factors for normalization. Options are "mean","median", "quant", and "mode". Default = "quant"
`qu`	Quantile of the normType if normType is set to "quant". Real value between 0 and 1. Default = 0.25
`quSizeFactor`	Quantile of the sizeFactor if sizeFactor is set to "quant". 0.75 corresponds to "upper quartile normalization". Real value between 0 and 1. Default = 0.75
`norm`	the normalization strategy to be used. If set to 0 the read counts are not normalized and cn.mops does not model different coverages. If set to 1 the read counts are normalized. If set to 2 the read counts are not normalized and cn.mops models different coverages. Default = 1.
`priorImpact`	positive real value that reflects how strong the prior assumption affects the result. The higher the value the more samples will be assumed to have copy number 2. Default = 1

minMedianRC	segments with median read counts over all samples < minMedianRC are excluded from the analysis
maxControls	integer reflecting the maximal numbers of controls to use. If set to 0 all highly correlated controls are used. Default = 25
corrThresh	threshold for selecting highly correlated controls. Default = 0.99
sex	either "mixed", "male", or "female" reflecting the sex of all samples (test and control)

Value

list of instances of "CNVDetectionResult"

Examples

```
data(panelcn.mops)
XandCB <- test
elementMetadata(XandCB) <- cbind(elementMetadata(XandCB),
                                 elementMetadata(control))
resultlist <- runPanelcnMops(XandCB, countWindows = countWindows)
```

---

splitROIs *Split (larger) ROIs into multiple smaller (overlapping) bins and create new BED file*

Description

Split (larger) ROIs into multiple smaller (overlapping) bins and create new BED file

Usage

```
splitROIs(oldBedFile, newBedFile, limit = 0, bin = 100, shift = 50,
  chr = FALSE)
```

Arguments

oldBedFile	filename of the BED file with absolute or relative path (structure of BED file without header: chromosome, exon start, exon end, exon name)
newBedFile	filename of the new BED file that should be created
limit	ROIs larger than limit will be split
bin	size of bins (in bp) the ROIs will be split into
shift	no. of bp between start positions of adjacent bins
chr	indicates whether naming contains chr prefix

Value

generates a new BED file with (larger) ROIs split into smaller bins

Examples

```
bed <- list.files(system.file("extdata", package = "panelcn.mops"),
                    pattern = ".bed$", full.names = TRUE)
splitROIs(bed, "newBed.bed")
```

`test` *GRanges object of countWindows with read counts for a test sample as elementMetadata.*

Description

The object was created using the function countBamListInGRanges with the enclosed countWindows object, a subset of a BAM file provided by the 1000 Genomes Project and the read.width parameter set to 150.

Details

Test data included in panelcn.mops

Author(s)

Gundula Povysil

Examples

```
data(panelcn.mops)
test
```

Index

∗ **data**
control, 2
countWindows, 3
read.width, 8
resultlist, 8
test, 11

control, 2
countBamListInGRanges, 3
countWindows, 3
createResultTable, 4

getWindows, 5

panelcn.mops, 5
plotBoxplot, 7

read.width, 8
resultlist, 8
runPanelcnMops, 9

splitROIs, 10

test, 11