

cghMCR

April 19, 2009

cghMCR-class

Class "cghMCR" is a S4 class for the identification of minimum common regions of gains or losses across samples

Description

Objects of this class provides the functionalities to detecting chromosome regions that show gains or losses across different samples

Objects from the Class

Objects can be created by calls of the form `new ("cghMCR", ...)`. A constructor `cghMCR` may be used to instantiate object of this class

Slots

DNASeg: Object of class "data.frame" containing segmentation data derived from segmentation analysis using `segment` or `getSegments`

gapAllowed: Object of class `gapAllowed` is an integer specifying low threshold of base pair number to separate two adjacent segments, below which the two segments will be joined as an altered span

alteredLow: Object of class `alteredLow` is a positive number between 0 and 1 specifying the lower reshold percental value. Only segments with values falling below this threshold are considered as altered span

alteredHigh: Object of class `alteredHigh` is a positive number between 0 and 1 specifying the upper reshold percental value. Only segments with values falling over this threshold are considered as altered span

recurrence: Object of class `recurrence` is an integer between 1 and 100 that specifies the rate of occurrence for a gain or loss that are observed across sample. Only gains or losses with occurrence rate grater than the threshold values are declared as MCRs

spanLimit: Object of class `spanLimit` is an integer that defines the leangh of altered spans that can be considered as locus. It is not of any use at this time

Methods

MCR `signature(object = "cghMCR")`: identifies minimum common regions of gains/losses across samples

Note

The function is a contribution of The Center for Applied Cancer Science of Dana-Farber Cancer Institute

Author(s)

Jianhua Zhang

See Also

[cghMCR](#)

Examples

```
require("cghMCR")
data("sampleData")
segments <- getSegments(sampleData)
cghmcr <- cghMCR(segments, gapAllowed = 500, alteredLow = 0.20,
                 alteredHigh = 0.80, recurrence = 50)
```

cghMCR

The constructor for the cghMCR class

Description

Instantiates a cghMCR object using the parameters passed.

Usage

```
cghMCR(segments, gapAllowed = 500, alteredLow = 0.03, alteredHigh =
0.97, spanLimit = 2e+07, recurrence = 75, thresholdType = c("quantile",
"value"))
```

Arguments

segments	segments is a data frame extracted from the "output" element of the object returned by segment of the package DNACopy or getSegments
gapAllowed	gapAllowed is an integer specifying low threshold of base pair number to separate two adjacent segments, below which the two segments will be joined as an altered span
alteredLow	alteredLow is a positive number between 0 and 1 specifying the lower resh-old percentual value. Only segments with values falling below this threshold are considered as altered span
alteredHigh	alteredHigh is a positive number between 0 and 1 specifying the upper resh-old percentual value. Only segments with values falling over this threshold are considered as altered span
recurrence	recurrence is an integer between 1 and 100 that specifies the rate of occurrence for a gain or loss that are observed across sample. Only gains/losses with occurrence rate grater than the threshold values are declared as MCRs

`spanLimit` `spanLimit` is an integer that defines the length of altered spans that can be considered as locus. It is not of any use at this time

`thresholdType` `thresholdType` is a character string that can be either "quantile" or "value" indicating whether `alteredLow` or `alteredHigh` is quantile or actual value

Details

The function is just a constructor of the [cghMCR](#) class for the instantiation of a cghMCR object

Value

An object of the cghMCR class

Note

The function is a contribution of The Center for Applied Cancer Science of Dana-Farber Cancer Institute

Author(s)

Jianhua Zhang

References

References on S4 class

See Also

[cghMCR](#)

Examples

```
require("cghMCR")
data("sampleData")
segments <- getSegments(sampleData)
cghmcr <- cghMCR(segments, gapAllowed = 500, alteredLow = 0.20,
                 alteredHigh = 0.80, recurrence = 50)
```

`mergeMCRProbes` *A function that appends probe ids to a data frame containing MCRs*

Description

This function takes a data frame generated by [MCR](#) and then appends probe ids corresponding to each MCR as a column to the data frame.

Usage

```
mergeMCRProbes(mcr, rawData)
```

Arguments

<code>mcr</code>	<code>mcr</code> is a data frame generated by MCR that contains MCRs identified and other related data
<code>rawData</code>	<code>rawData</code> is a data frame with at least three columns. The first column should be probe ids, second the chromosome number the probes corresponding to, and the third the starting or ending chromosomal locations of the probes

Details

The `mcr` data frame passed must have the first column for chromosome numbers, the 7th column for the starting positions of the MCRs, and the 8th column for the ending positions of the MCRs.

Value

A data frame with MCRs and the corresponding probe ids and other data.

Note

The function is a contribution of The Center for Applied Cancer Science of Dana-Farber Cancer Institute

Author(s)

Jianhua Zhang

See Also

[MCR](#)

Examples

```
require("cghMCR")
data("sampleData")
segments <- getSegments(sampleData)
cghmcr <- cghMCR(segments, gapAllowed = 500, alteredLow = 0.20,
                 alteredHigh = 0.80, recurrence = 50)
mcrs <- MCR(cghmcr)
mcrs <- mergeMCRProbes(mcrs, segments[["data"]])
```

<code>plot.DNAcopy</code>	<i>A function to plot the original data along with the segments identified using segment of DNAcopy.</i>
---------------------------	--

Description

This function takes a DNAcopy object generated by running [getSegments](#) and then plots the original data along with the segments identified.

Usage

```
plot.DNAcopy(x, ..., save = FALSE, layout)
```

Arguments

x	x is a DNACopy object generated by getSegments
...	... may contain an optional boolean for save to indicate whether to show the plot on the screen (FALSE. Default) or return the name of the plot (TRUE) to allow the users to save the plot
save	save a boolean indicating wheather the plot will be draw on the screen only (FALSE) or the name of the png file of the plot be returned (TRUE)
layout	layout a numeric vector of two elements defining the values used for mfrow of the function par for the layout of figures on a plot

Details

This function only works with the DNACopy returned by [getSegments](#). It will work with the DNACopy returned by segment of the DNACopy package when the a column containing the probe ids is added as the first column of the "data" element of the object.

Value

This function returns invisible() or the name of the plot depending on whether save is set to FALSE of TRUE.

Note

The function is a contribution of The Center for Applied Cancer Science of Dana-Farber Cancer Institute

Author(s)

Jianhua Zhang

References

The DNACopy package

See Also

[getSegments](#)

Examples

```
if(interactive()){
  require("cghMCR")
  data("sampleData")
  segments <- getSegments(sampleData)
  plot(segments)
}
```

Index

*Topic **classes**

`cghMCR-class`, 1

*Topic **hplot**

`plot.DNAcopy`, 4

*Topic **manip**

`mergeMCRProbes`, 3

*Topic **methods**

`cghMCR`, 2

`cghMCR`, 1, 2, 2, 3

`cghMCR-class`, 1

`getSegments`, 1, 2, 4, 5

`getSegments (cghMCR-class)`, 1

`getSegments, marrayNorm-method`

`(cghMCR-class)`, 1

`getSegments, marrayRaw-method`

`(cghMCR-class)`, 1

`MCR`, 3, 4

`MCR (cghMCR-class)`, 1

`MCR, cghMCR-method (cghMCR-class)`,

1

`mergeMCRProbes`, 3

`plot.DNAcopy`, 4

`sampleData (cghMCR)`, 2