

# Package ‘cghMCR’

November 8, 2024

**Version** 1.64.0

**Title** Find chromosome regions showing common gains/losses

**Author** J. Zhang and B. Feng

**Maintainer** J. Zhang <jzhang@jimmy.harvard.edu>

**Depends** methods, DNAcopy, CNTools, limma

**Imports** BiocGenerics (>= 0.1.6), stats4

**Description** This package provides functions to identify genomic regions of interests based on segmented copy number data from multiple samples.

**Keyword** arrayCGH

**License** LGPL

**LazyLoad** yes

**biocViews** Microarray, CopyNumberVariation

**git\_url** <https://git.bioconductor.org/packages/cghMCR>

**git\_branch** RELEASE\_3\_20

**git\_last\_commit** acf6ec8

**git\_last\_commit\_date** 2024-10-29

**Repository** Bioconductor 3.20

**Date/Publication** 2024-11-08

## Contents

cghMCR . . . . .	2
cghMCR-class . . . . .	3
colnames-methods . . . . .	4
mergeMCRProbes . . . . .	4
plot.DNAcopy . . . . .	5
SGOL-class . . . . .	7
<b>Index</b>	<b>8</b>

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 <a href="#">getSegments</a>
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 reshld percental 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 reshld percental 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 occurence rate grater than the threshold values are declared as MCRs
spanLimit	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
thresholdType	thresholdType is a character string that can be either "quantile" or "value" indicating wether alteredLow or alteredHigh is quantial 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

```
data("segData")
cghmcr <- cghMCR(segData, gapAllowed = 500,
  alteredLow = 0.20, alteredHigh = 0.80, recurrence = 50)
```

---

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

**DNADData:** Object of class "data.frame" containing raw data derived used for the segmentation analysis from segmentation analysis

**altered:** Object of class "data.frame" containing data for the altered regions

**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 greater than the threshold values are declared as MCRs

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

**thresholdType:** A character string that can be either "quantile", "value" to indicate the type of the value for recurrence

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("CNTools")
data("sampleData")
cghmcr <- cghMCR(sampleData[sampleData[, "ID"] %in%
  sample(unique(sampleData[, "ID"]), 20), ], gapAllowed = 500,
  alteredLow = 0.20, alteredHigh = 0.80, recurrence = 50)
```

---

colnames-methods	<i>Methods for Function colnames/rownames in Package ‘cghMCR’</i>
------------------	---

---

Description

Methods for function colnames or rownames in Package ‘cghMCR’ to extract row or column names from a SGOL object

Methods

**x = "SGOL"** Extract colnum/row names from SGOL object

---

mergeMCRProbes	<i>A function that appends probe ids to a data frame containing MCRs</i>
----------------	--

---

Description

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

Usage

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

**Arguments**

mcr	mcr is a data frame generated by <a href="#">MCR</a> that contains MCRs identified and other related data
rawData	rawData 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**

```
data("segData")
cghmcr <- cghMCR(segData, gapAllowed = 500, alteredLow = 0.20,
                 alteredHigh = 0.80, recurrence = 50)
mcrcs <- MCR(cghmcr)
mcrcs <- mergeMCRProbes(mcrcs, segData[["data"]])
```

---

plot.DNACopy

*A function to plot the original data along with the segments identified using segment of DNACopy.*

---

**Description**

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

**Usage**

```
## S3 method for class 'DNACopy'
plot(x, ..., save = FALSE, layout)
```

**Arguments**

x	x is a DNACopy object generated by <a href="#">getSegments</a>
...	... 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("segData")
  plot(segData)
}
```

---

SGOL-class*Class "SGOL" represents segments of gain or loss across samples*

---

**Description**

Segments of gains or losses along chromosomes can be calculated based on segmentation data derived from the segment function of the DNAcopy package

**Objects from the Class**

Objects can be created by calls of the form `new("SGOL", ...)` or using a constructor `SGOL`.

**Slots**

**gol:** Object of class "matrix" holding the gain or loss data for chromosomal segments

**threshold:** Object of class "vector" of length 2 indicating the lower and upper thresholds below/over which data points will be included in the calculation of SGOL score using the method defined by method

**method:** Object of class "function" giving the method used to calculate SGOL scores. Common methods include sum, median, and mean

**Methods**

**gol** signature(object = "SGOL"): extracts SGOL scores

**method** signature(object = "SGOL"): gets the name of the function used to calculate the SGOL scores

**plot** signature(x = "SGOL", y = "ANY", ...): plots the data

**threshold** signature(object = "SGOL"): gets the threshold used for the calculation

**Author(s)**

Jianhua Zhang

**References**

The SGOL score is a modified version of the GISTIC score published in PNAS 104: 20007-20012

**Examples**

```
showClass("SGOL")
require(CNTools)
```

# Index

- \* **classes**
  - cghMCR-class, [3](#)
  - SGOL-class, [7](#)
- \* **hplot**
  - plot.DNAcopy, [5](#)
- \* **manip**
  - mergeMCRProbes, [4](#)
- \* **methods**
  - cghMCR, [2](#)
  - colnames-methods, [4](#)
  - [, SGOL-method (colnames-methods), [4](#)
  - [-methods (colnames-methods), [4](#)
- cghMCR, [2](#), [2](#), [3](#), [4](#)
- cghMCR-class, [3](#)
- colnames, SGOL-method
  - (colnames-methods), [4](#)
- colnames-methods, [4](#)
- getSegments, [2](#), [5](#), [6](#)
- getSegments (cghMCR-class), [3](#)
- getSegments, marrayNorm-method
  - (cghMCR-class), [3](#)
- getSegments, marrayRaw-method
  - (cghMCR-class), [3](#)
- gol (SGOL-class), [7](#)
- gol, SGOL-method (SGOL-class), [7](#)
- MCR, [4](#), [5](#)
- MCR (cghMCR-class), [3](#)
- MCR, cghMCR-method (cghMCR-class), [3](#)
- mergeMCRProbes, [4](#)
- method, SGOL-method (SGOL-class), [7](#)
- plot (SGOL-class), [7](#)
- plot, ANY-method (SGOL-class), [7](#)
- plot, methods (SGOL-class), [7](#)
- plot, SGOL, ANY-method (SGOL-class), [7](#)
- plot, SGOL-method (SGOL-class), [7](#)
- plot.DNAcopy, [5](#)
- rownames, SGOL-method
  - (colnames-methods), [4](#)
- rownames-methods (colnames-methods), [4](#)
- segData (cghMCR), [2](#)
- SGOL (SGOL-class), [7](#)
- SGOL-class, [7](#)
- threshold, SGOL-method (SGOL-class), [7](#)