Package 'rCGH'

May 11, 2024

Type Package

Title Comprehensive Pipeline for Analyzing and Visualizing Array-Based CGH Data

Version 1.34.0

Date 2018-05-12

Maintainer Frederic Commo <fredcommo@gmail.com>

URL https://github.com/fredcommo/rCGH

Description A comprehensive pipeline for analyzing and interactively visualizing genomic profiles generated through commercial or custom aCGH arrays. As inputs, rCGH supports Agilent dual-color Feature Extraction files (.txt), from 44 to 400K, Affymetrix SNP6.0 and cytoScanHD probeset.txt, cychp.txt, and cnchp.txt files exported from ChAS or Affymetrix Power Tools. rCGH also supports custom arrays, provided data complies with the expected format. This package takes over all the steps required for individual genomic profiles analysis, from reading files to profiles segmentation and gene annotations. This package also provides several visualization functions (static or interactive) which facilitate individual profiles interpretation.
Input files can be in compressed format, e.g. .bz2 or .gz.

License Artistic-2.0

biocViews aCGH,CopyNumberVariation,Preprocessing,FeatureExtraction

Depends R (>= 3.4),methods,stats,utils,graphics

Imports plyr,DNAcopy,lattice,ggplot2,grid,shiny (>= 0.11.1), limma,affy,mclust,TxDb.Hsapiens.UCSC.hg18.knownGene, TxDb.Hsapiens.UCSC.hg19.knownGene,TxDb.Hsapiens.UCSC.hg38.knownGene, org.Hs.eg.db,GenomicFeatures,GenomeInfoDb,GenomicRanges,AnnotationDbi, parallel,IRanges,grDevices,aCGH

Suggests BiocStyle, knitr, BiocGenerics, RUnit

VignetteBuilder knitr

LazyData true

NeedsCompilation no

Author Frederic Commo [aut, cre]

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

git_branch RELEASE_3_19

git_last_commit 2d56ebd

git_last_commit_date 2024-04-30

Repository Bioconductor 3.19

Date/Publication 2024-05-10

Contents

rCGH-package *Comprehensive Pipeline for Analyzing and Visualizing Array-Based CGH Data*

Description

A comprehensive pipeline for analyzing and interactively visualizing genomic profiles generated through commercial or custom aCGH arrays. As inputs, rCGH supports Agilent dual-color Feature Extraction files (.txt), from 44 to 400K, Affymetrix SNP6.0 and cytoScanHD probeset.txt, cychp.txt, and cnchp.txt files exported from ChAS or Affymetrix Power Tools.
rCGH also supports custom arrays, provided data is in a suitable format. This package takes over all the steps required for individual genomic profiles analysis, from reading files to segmenting and annotating genes. This package provides several visualization functions (static or interactive) which facilitate individual profiles interpretation. Input files can be in compressed format, e.g. .bz2 or .gz.

Author(s)

Frederic Commo <frederic.commo@gustaveroussy.fr>

See Also

readAgilent, readAffySNP6, readAffyCytoScan, readGeneric

Examples

```
filePath <- system.file("extdata", "Affy_cytoScan.cyhd.CN5.CNCHP.txt.bz2",
    package = "rCGH")
cgh <- readAffyCytoScan(filePath, sampleName = "AffyScHD")

cgh <- adjustSignal(cgh, nCores=1)
cgh <- segmentCGH(cgh, nCores=1)
cgh <- EMnormalize(cgh)

# Static visalizations
plotDensity(cgh)
multiplot(cgh)

## Not run:
# Interactive visalizations
view(cgh)

## End(Not run)
```

adjustSignal *Array-based CGH Preprocessing*

Description

This function performs several preprocessing steps: local regressions (loessFit) are used to correct cy3/cy5 and GC% bias, when Agilent dual-color hybridization are used only.
In case of Affymetrix cychp.txt (or cnchp.txt) data are used, these steps have been already done in ChAS or Affymetrix Power Tools.

Usage

```
## S4 method for signature 'rCGH'
adjustSignal(object, Scale=TRUE, Cy=TRUE, GC=TRUE, Ref="cy3",
    suppOutliers=TRUE, nCores=NULL, verbose=TRUE)
```

Arguments

object	: An object of class "rCGH"
Scale	: logical. If TRUE (default), Log2Ratios are standardized.
Cy	: logical. If TRUE (default), cy3/cy5 bias is corrected using a local regression (loessFit). For Agilent dual-color hybridization only. Notice that, in case of Affymetrix files (cychp.txt or cnchp.txt), this argument is automatically set to FALSE, since this step is managed when files are exported from ChAS or APT.
GC	: logical. If TRUE (default), the GC% bias is corrected using a local regression (loessFit). For Agilent dual-color hybridization only. Notice that, in case of Affymetrix files (cychp.txt or cnchp.txt), this argument is automatically set to FALSE, since this step is managed when files are exported from ChAS or APT.
Ref	: string. Which cyanine was used as the reference. Possible values are "cy3" (default) and "cy5". For Agilent dual-color hybridization only.
suppOutliers	: logical. If TRUE (default), outliers are removed using an Expectation-Maximization algorithm (EM). See details below.
nCores	: numeric. The number of cores to use in order to speed up the computation. When NULL (default), half of the available cores are used. If a value greater than the number of available cores is passed, this latter will be used. See detectCores.
verbose	: logical. When TRUE (default), progress is printed.

Details

When suppOutliers is TRUE (default), the Log2Ratios are splitted with respect to chromosomes. The main regions within each chromosome are identified using EM. Within each region *r_i*, x[r_i] values are redifined according to the corresponding model parameters. as:

$$x[r_i]\ N(mu_i, sigma_i)$$

Notice that this step may substantially increase the computation time.

Value

An object of class "rCGH"

Author(s)

Frederic Commo

See Also

detectCores, mclapply

Examples

```
filePath <- system.file("extdata", "Affy_cytoScan.cyhd.CN5.CNCHP.txt.bz2",
    package = "rCGH")
cgh <- readAffyCytoScan(filePath, sampleName = "AffyScHD")
cgh <- adjustSignal(cgh, nCores=1)
getParam(cgh)
```

---

agilentDB *aCGH Agilent Probes GC Fraction*

Description

A dataset containing the Agilent probe Ids, and their GC content.
These data allow rCGH to support Agilent aCGH arrays, from 44K to 400K. See the source section.
This information is used to correct the GC% bias. For Agilent data only.

Usage

```
agilentDB
```

Format

A data frame with 411056 rows and 2 columns:

- ProbeID: Official Agilent probe ids.
- GC: The GC content in each probe sequence, expressed as the GC fraction.

Value

A dataset

Note

Probe sequences are not used and have been removed after computing the GC fractions as GC = sum(G or C)/length(sequence), for each sequence in file.

Author(s)

Frederic Commo

Source

These data derive from the official Agilent Sequence List, SurePrint_G3_Human_CGH_Microarray 2x400K_021850_D_SequenceList_20111015.txt, freely available at: Agilent SureDesign
Access date: 3-2-2015
Notice that a User ID and Password are required to sign in.

`AllAccessors` *"rCGH" Accessor Functions*

Description

Methods for extracting information from an object of class "rCGH".
Each of the below methods are simply convenience functions which extract the corresponding slots (as the name of each method suggests) from an object of class "rCGH".

Usage

```
## S4 method for signature 'rCGH'
getInfo(object, item = NULL)
## S4 method for signature 'rCGH'
getCNset(object)
## S4 method for signature 'rCGH'
getParam(object)
## S4 method for signature 'rCGH'
getSegTable(object, minLen = NULL)
```

Arguments

`object`	: An object of class "rCGH"
`item`	: character. Can be one, or a vector of items. When `NULL`, the full available information is returned. If `item` is specified, and exists, the corresponding value(s) only is(are) returned.
`minLen`	: numeric. The mininal length for a segment, in Kb. When `NULL` (default), the segmentation table is exported, as it has been computed with `segmentCGH`, segments shorter than the specified value are re-merged otherwise.

Value

- getInfo(object, item = NULL): character.
- getCNset(object): a data frame.
- getParam(object): a list of parameters.
- getSegTable(object, minLen = NULL): a data frame.

Methods

"rCGH"
- getInfo(object, item = NULL): returns the values of the specified items, all the information otherwise.
- getCNset(object): returns the full by-probe dataset.
- getParam(object): returns the analysis parameters.
- getSegTable(object, minLen = NULL): returns the segmentation table - one row per segment.

Author(s)

Frederic Commo

See Also

setInfo, segmentCGH

Examples

```
filePath <- system.file("extdata", "Agilent4x180K.txt.bz2", package = "rCGH")
cgh <- readAgilent(filePath, sampleName = "Agilent4x180K", labName = "myLab")

# Getting all the information
getInfo(cgh)

# Getting specific items
getInfo(cgh, c("sampleName", "labName"))
```

---

byGeneTable *Converting a Segmentation Table Into a By-Gene Table*

Description

This function creates a by-gene table by listing all the genes contained in each of the segments in the segmentation table.
Gene annotations (symbol, location,...), segmented Log2Ratios, and segment legnth are reported in the final table.
A supplementary score is the `relativeLog`: the magnitude, in Log2, from the closest centromere.

Usage

```
byGeneTable(segTable, symbol = NULL,
    genome = c("hg19", "hg18", "hg38"), columns = NA, verbose = TRUE)
```

Arguments

segTable	: data frame. A segmentation table exported from an object of class "rCGH"
symbol	: character. A valid HUGO symbol. When NULL the full gene table is returned, the corresponding gene information only o/w.
genome	: string. The genome build to use. Supported genomes are hg18, hg19 (default), and hg38.
columns	: string. what supplementary genes annotations to export. Allowed annotations are those supported by the select method from the AnnotationDbi bioconductor package. When NA (default), 'SYMBOL','ENTREZID', 'GENENAME' and 'MAP' are exported.
verbose	: logical. When TRUE progress is printed.

Details

For gene annotations, Hg19/GRCh37 annotations downloaded from *NCBI* are considered.

Value

An object of class "rCGH"

Author(s)

Frederic Commo

See Also

getSegTable

Examples

```
filePath <- system.file("extdata", "Affy_cytoScan.cyhd.CN5.CNCHP.txt.bz2",
    package = "rCGH")
cgh <- readAffyCytoScan(filePath, sampleName = "AffyScHD")
cgh <- adjustSignal(cgh, nCores=1)
cgh <- segmentCGH(cgh, nCores=1)
cgh <- EMnormalize(cgh)
st <- getSegTable(cgh)
bygene <- byGeneTable(st)
head(bygene)
```

EMnormalize *Genomic Profile Centralization*

Description

This function analyses the Log2Ratios as a mixture of several gaussian populations, using an Expectation-Maximization algorithm (EM).
The peakThresh argument specifies what proportion of the main density peak is allowed for choosing a neutral 2-copies population. The mean of the chosen population is used for centralizing the profile.
See Mclust.

Usage

```
## S4 method for signature 'rCGH'
EMnormalize(object, G = 2:6, priorScale = 5,
    peakThresh = 0.5, mergeVal = 0.1, Title = NA, verbose = TRUE)
```

Arguments

object	: An object of class "rCGH"
G	: numeric. The number of groups to test during the gaussian mixture estimation. Default is from 2 to 6.
priorScale	: numeric. A scale value passed to priorControl. Default is 5.
peakThresh	: numeric. The proportion of the highest peak to consider as a peak selection threshold. Default is 0.5.
mergeVal	: numeric. Populations with means closer than mergeVal will be pooled together, default is 0.1. Set mergeVal to zero to not pool closed sub-populations.
Title	: character string. A title for the density plot. If NA (default), the sample name (when exists in object info) will be used.
verbose	: logical. When TRUE (default), progress is printed.

Details

Depending on peakThresh, the mean of the highest density, or a lower value, is chosen for centering the genomic profile. To do so, L2R are modeled for each segment s_i, with respect to n_i (the number of probes included in segment i), mu_i and sd_i. The mixture of L2Rs is then analysed as a mixture of gaussian populations.
When a peakThresh value is specified, heights of density peaks are compared: the lowest peak mean among the peaks respecting the criteria: peakHeight > max(peaks)*peakThresh, is chosen for centralizing the data. See References

Value

An object of same class as the input.

Author(s)

Frederic Commo

References

Commo et al. Impact of centralization on aCGH-based genomic profiles for precision medicine in oncology. Ann Oncol. 2014

See Also

plotDensity, mclust

Examples

```
filePath <- system.file("extdata", "Affy_cytoScan.cyhd.CN5.CNCHP.txt.bz2",
    package = "rCGH")
cgh <- readAffyCytoScan(filePath, sampleName = "AffyScHD")
cgh <- adjustSignal(cgh, nCores=1)
cgh <- segmentCGH(cgh, nCores=1)
cgh <- EMnormalize(cgh)
getParam(cgh)
```

---

hg18 *Hg18 Chromosome Lengths and Centromere Locations*

Description

A data set containing lengths and centromere locations for each of the 24 chromosomes, according to Hg18.

Usage

```
hg18
```

Format

A data set with 24 rows and 5 columns:

- chrom: chromosome number.
- length: chromosome length.
- centromerStart: centromere start position.
- centromerEnd: centromere start position.
- cumlen: cumulative length (where the previous chromosome ends).

Value

a data set.

Author(s)

Frederic Commo

Source

These data derived from the Hg18 gap UCSC table, freely available at: UCSC
Access date: 10-10-2015
Within the browser, select:
group: All Tables
database: hg18
table: gap

Examples

```
# For users convenience, we provide a prebuilt dataset
# containing the Hg18 chr lengths, and centromeres location.
hg18

# The same dataset can be obtained as follow:
## Not run:
library(BSgenome)
library(rtracklayer)

getChrLength <- function(genome){
    genome <- sprintf("BSgenome.Hsapiens.UCSC.
    g <- getBSgenome(genome, masked=FALSE)
    data.frame(chrom=1:24, length=seqlengths(g)[1:24])
}
.chrAsNum <- function(tbl){
    tbl$chrom <- gsub("chr", "", tbl$chrom)
    tbl$chrom[tbl$chrom=="X"] <- 23
    tbl$chrom[tbl$chrom=="Y"] <- 24
    tbl$chrom <- as.numeric(tbl$chrom)
    tbl[order(tbl$chrom),]
}
getCentromeres <- function(genome){
    mySession <- try(browserSession("UCSC"), silent=TRUE)
    # In case it fails, use another mirror
    if(inherits(mySession, "try-error"))
        mySession <- browserSession("UCSC",
                                    url="http://genome-euro.ucsc.edu/cgi-bin/")
    genome(mySession) <- genome
    obj <- ucscTableQuery(mySession, table="gap")
    tbl <- getTable(obj)
    tbl <- tbl[tbl$type=="centromere", c("chrom", "chromStart", "chromEnd")]
    colnames(tbl)[2:3] <- c("centromerStart", "centromerEnd")
    .chrAsNum(tbl)
}
makeHg <- function(genome){
    chrL <- getChrLength(genome)
    ctm <- getCentromeres(genome)
```

```
    tbl <- merge(chrL, ctm, by="chrom")
    cumlen <- c(0, cumsum(as.numeric(tbl$length))[-nrow(tbl)])
    cbind.data.frame(tbl, cumlen=cumlen)
}
hg18 <- makeHg("hg18")
hg18

## End(Not run)
```

---

hg19 *Hg19 Chromosome Lengths and Centromere Locations*

Description

A data set containing lengths and centromere locations for each of the 24 chromosomes, according to Hg19.

Usage

```
hg19
```

Format

A data set with 24 rows and 5 columns:

- chrom: chromosome number.
- length: chromosome length.
- centromerStart: centromere start position.
- centromerEnd: centromere start position.
- cumlen: cumulative length (where the previous chromosome ends).

Value

a data set.

Author(s)

Frederic Commo

Source

These data derived from the Hg19 gap UCSC table, freely available at: UCSC
Access date: 1-31-2014
Within the browser, select:
group: All Tables
database: hg19
table: gap

Examples

```
# For users convenience, we provide a prebuilt dataset
# containing the Hg19 chr lengths, and centromeres location.
hg19

# The same dataset can be obtained as follow:
## Not run:
library(BSgenome)
library(rtracklayer)

getChrLength <- function(genome = "BSgenome.Hsapiens.UCSC.hg19"){
    g <- getBSgenome(genome, masked=FALSE)
    data.frame(chrom=1:24, length=seqlengths(g)[1:24])
}
.chrAsNum <- function(tbl){
    tbl$chrom <- gsub("chr", "", tbl$chrom)
    tbl$chrom[tbl$chrom=="X"] <- 23
    tbl$chrom[tbl$chrom=="Y"] <- 24
    tbl$chrom <- as.numeric(tbl$chrom)
    tbl[order(tbl$chrom),]
}
getCentromeres <- function( genome="hg19" ){
    mySession <- try(browserSession("UCSC"), silent=TRUE)
    # In case of failure, try another mirror
    if(inherits(mySession, "try-error"))
        mySession <- browserSession("UCSC",
                                    url="http://genome-euro.ucsc.edu/cgi-bin/")
    genome(mySession) <- genome
    obj <- ucscTableQuery(mySession, table="gap")
    tbl <- getTable(obj)
    tbl <- tbl[tbl$type=="centromere", c("chrom", "chromStart", "chromEnd")]
    colnames(tbl)[2:3] <- c("centromerStart", "centromerEnd")
    .chrAsNum(tbl)
}
makeHg19 <- function(){
    tbl <- merge(getChrLength(), getCentromeres(), by="chrom")
    cumlen <- c(0, cumsum(as.numeric(tbl$length))[-nrow(tbl)])
    cbind.data.frame(tbl, cumlen=cumlen)
}
hg19 <- makeHg19()
hg19

## End(Not run)
```

---

hg38 *Hg38 Chromosome Lengths and Centromere Locations*

Description

A data set containing lengths and centromere locations for each of the 24 chromosomes, according to Hg38.

Usage

```
hg38
```

Format

A data set with 24 rows and 5 columns:

- chrom: chromosome number.
- length: chromosome length.
- centromerStart: centromere start position.
- centromerEnd: centromere start position.
- cumlen: cumulative length (where the previous chromosome ends).

Value

a data set.

Author(s)

Frederic Commo

Source

These data derived from the Hg38 gap UCSC table, freely available at: UCSC
Access date: 10-10-2015
Within the browser, select:
group: All Tables
database: hg38
table: gap

Examples

```
# For users convenience, we provide a prebuilt dataset
# containing the Hg38 chr lengths, and centromeres location.
hg38

# The same dataset can be obtained as follow:
## Not run:
library(BSgenome)
library(rtracklayer)

getChrLength <- function(genome){
    genome <- sprintf("BSgenome.Hsapiens.UCSC.
    g <- getBSgenome(genome, masked=FALSE)
    data.frame(chrom=1:24, length=seqlengths(g)[1:24])
}
.chrAsNum <- function(tbl){
    tbl$chrom <- gsub("chr", "", tbl$chrom)
    tbl$chrom[tbl$chrom=="X"] <- 23
    tbl$chrom[tbl$chrom=="Y"] <- 24
```

```
    tbl$chrom <- as.numeric(tbl$chrom)
    tbl[order(tbl$chrom),]
}
getCentromeres <- function(genome){
    mySession <- try(browserSession("UCSC"), silent=TRUE)
    # In case it fails, use another mirror
    if(inherits(mySession, "try-error"))
        mySession <- browserSession("UCSC",
                                    url="http://genome-euro.ucsc.edu/cgi-bin/")
    genome(mySession) <- genome
    obj <- ucscTableQuery(mySession, table="gap")
    tbl <- getTable(obj)
    if(!"centromere"
        return(NULL)
    tbl <- tbl[tbl$type=="centromere", c("chrom", "chromStart", "chromEnd")]
    colnames(tbl)[2:3] <- c("centromerStart", "centromerEnd")
    .chrAsNum(tbl)
}
makeHg <- function(genome){
    chrL <- getChrLength(genome)
    ctm <- getCentromeres(genome)
    # Notice that, in case of Hg38, centromeres locations are in Hg19.
    if(is.null(ctm))
        ctm <- getCentromeres("hg19")
    tbl <- merge(chrL, ctm, by="chrom")
    cumlen <- c(0, cumsum(as.numeric(tbl$length))[-nrow(tbl)])
    cbind.data.frame(tbl, cumlen=cumlen)
}
hg38 <- makeHg("hg38")
hg38

## End(Not run)
```

`multiplot` *Static Genomic Profile and LOH Visualization*

Description

This function display a static view of the genomic profile and the allelic difference stored in an object of class "rCGH".
If no allelic difference is available, the genomic profile only is displayed.

Usage

```
## S4 method for signature 'rCGH'
multiplot(object, symbol=NULL, gain=.5,
    loss=(-.5), minLen=10, pCol = "grey50", GLcol = c("blue", "red3"),
    L=matrix(seq(1, 12)), p=c(1/2, 1/4, 1/4), Title=NULL, ylim=NULL)
```

Arguments

object	: An object of class "rCGH"
symbol	: character. A valid HUGO symbol (case insensitive).
gain	: numeric. A gain threshold value (in $Log2(Ratio)$). Segments greater, or equal to, this value will be colored, as specified by GLcol.
loss	: numeric. A loss threshold value (in $Log2(Ratio)$). Segments lower, or equal to, this value will be colored, as specified by GLcol.
minLen	: numeric. The mininal length for a segment, expressed in Kb. When NULL (default), segments are reported as they have been computed by segmentCGH. Segments shorter than the specified value are re-merged otherwise.
pCol	: string. The probe points color. Default is "grey50".
GLcol	: vector. A vector of 2 colors: the gained and lost segments colors, respectively. Default is "blue" for gains and "red3" for losses.
L	: matrix. A matrix defining the layout. Default is 12 lines.
p	: numeric. The proportion of each plot within the plot window. Default is 1/2, 1/4, 1/4, which corresponds to 6-4-4 lines for the genomic profile in Log2R, in copy number, and the LOH plot, respectiviely, and given a 12-line layout.
Title	: character string. A title for the plot. If NULL (default), the sample name (when exists) is used.
ylim	: numeric. A vector of two values specifying the y-axis range. See plotProfile.

Value

None.

Note

If no allelic difference is available, the genomic profile only is displayed.

Author(s)

Frederic Commo

See Also

plotDensity, plotProfile, plotLOH, view

Examples

```
filePath <- system.file("extdata", "Affy_cytoScan.cyhd.CN5.CNCHP.txt.bz2",
    package = "rCGH")
cgh <- readAffyCytoScan(filePath, sampleName = "AffyScHD")
cgh <- adjustSignal(cgh, nCores=1)
cgh <- segmentCGH(cgh, nCores=1)
cgh <- EMnormalize(cgh)

# Static visalizations
```

```
multiplot(cgh, symbol = "erbb2")
```

---

plotDensity *Visualizing the Log2Ratios Density and Centralization Decision*

Description

This function display the distribution of the Log2Ratios, as well as how the "EMnormalize" step estimates the mixture of gaussian populations, and choose a centralization value.

Usage

```
## S4 method for signature 'rCGH'
plotDensity(object, breaks=NULL, Title=NULL,...)
```

Arguments

object	: An object of class "rCGH"
breaks	: The number of breaks to use. See hist. When NULL (default), breaks is arbitrarily defined from the number of values to draw.
Title	: character string. A title for the density plot. If NULL (default), the sample name (when exists) will be used.
...	: Other graphical parameters supported by par.

Value

None.

Author(s)

Frederic Commo

See Also

plotProfile, plotLOH, multiplot, view

Examples

```
filePath <- system.file("extdata", "Affy_cytoScan.cyhd.CN5.CNCHP.txt.bz2",
    package = "rCGH")
cgh <- readAffyCytoScan(filePath, sampleName = "AffyScHD")
cgh <- adjustSignal(cgh, nCores=1)
cgh <- segmentCGH(cgh, nCores=1)
cgh <- EMnormalize(cgh)
plotDensity(cgh)
```

plotLOH *Allelic Differences Visualization*

Description

This function display a static view of the allele differences, when available.

Usage

```
## S4 method for signature 'rCGH'
plotLOH(object, Title=NULL)
```

Arguments

object	: An object of class "rCGH"
Title	: character string. A title for the density plot. If NULL (default), the sample name (when exists) is used.

Value

None.

Author(s)

Frederic Commo

See Also

plotDensity, plotProfile, multiplot, view

Examples

```
filePath <- system.file("extdata", "Affy_cytoScan.cyhd.CN5.CNCHP.txt.bz2",
    package = "rCGH")
cgh <- readAffyCytoScan(filePath, sampleName = "AffyScHD")
cgh <- adjustSignal(cgh, nCores=1)
cgh <- segmentCGH(cgh, nCores=1)
cgh <- EMnormalize(cgh)

# Static visalizations
plotLOH(cgh)
```

plotProfile *Static Genomic Profile Visualization*

Description

This function display a static view of the genomic profile stored in an object of class "rCGH".

Usage

```
## S4 method for signature 'rCGH'
plotProfile(object, showCopy=FALSE, symbol=NULL,
    gain=.5, loss=(-.5), minLen = 10, pCol = "grey50",
    GLcol = c("blue", "red3"), Title=NULL, ylim=NULL)
```

Arguments

object	: An object of class "rCGH"
showCopy	: logical. To show the estimated copy numbers instead of the Log2Ratios. default is FALSE.
symbol	: character. A valid HUGO symbol (case insensitive).
gain	: numeric. A gain threshold value (in $Log2(Ratio)$) from where gained segments will be shown, in blue.
loss	: numeric. A loss threshold value (in $Log2(Ratio)$) from where lossed segments will be shown, in red.
minLen	: numeric. The mininal length for a segment, in Kb. When NULL (default), segments are reported as they have been computed with segmentCGH, segments shorter than the specified value are re-merged otherwise.
pCol	: string. The probe points color. DEfault is "grey50".
GLcol	: vector. A vector of 2 colors: the gained and lost segments colors, respectively. Default is "blue" for gains and "red3" for losses.
Title	: character string. A title for the density plot. If NULL (default), the sample name (when exists) is used.
ylim	: numeric. A vector of two values specifying a range for the y-axis. If NULL (default), the range of Log2Ratio is used.

Value

None.

Author(s)

Frederic Commo

See Also

plotDensity, plotLOH, multiplot, view

Examples

```
filePath <- system.file("extdata", "Affy_cytoScan.cyhd.CN5.CNCHP.txt.bz2",
    package = "rCGH")
cgh <- readAffyCytoScan(filePath, sampleName = "AffyScHD")
cgh <- adjustSignal(cgh, nCores=1)
cgh <- segmentCGH(cgh, nCores=1)
cgh <- EMnormalize(cgh)

# Static visalization using Log2Ratios
plotProfile(cgh, symbol = "erbb2")

# Static visalization using estimated copy numbers
plotProfile(cgh, showCopy = TRUE, symbol = "erbb2")
```

---

rCGH-Agilent-class *Class* "rCGH-Agilent"

Description

An instance of class "rCGH-Agilent", which inherits from the superclass "rCGH".
Slots described below are used to store sample information, analysis parameters, and segmentation results. All are accessible through specific "Accessors" functions.

Objects from the Class

Objects can be created by calls of the form new("rCGH-Agilent", ...).

Slots

info: Object of class "character": where sample information can be stored. See getInfo and setInfo.

cnSet: Object of class "data.frame": the full data set. See getCNset.

param: Object of class "list": the analysis parameters stored for traceability. getParam.

segTable: Object of class "data.frame": the segmentation table. getSegTable.

Extends

Class "rCGH", directly.

Methods

No methods defined with class "rCGH-Agilent" in the signature.

Author(s)

Frederic Commo

See Also

"rCGH", "rCGH-cytoScan", "rCGH-SNP6", "rCGH-generic"

Examples

```
showClass("rCGH-Agilent")
```

---

rCGH-class *Class* "rCGH"

Description

Class "rCGH" is a superclass living on top of "rCGH-Agilent", "rCGH-SNP6", "rCGH-cytoScan", "rCGH-oncoScan", and "rCGH-generic". These objects inherit most of the properties of the superclass, and allow specific parameterizations used during the analysis process.
Objects are created by platform-specific `read` functions: "readAgilent", "readAffySNP6", and "readAffyCytoScan", each corresponding to their matched file format.
A supplementary "readGeneric" allows the user to create a "rCGH" object from custom arrays.
Slots described below are used to store sample information and analysis parameters, as well as segmentation results. All are accessible through specific "Accessors" functions.

Objects from the Class

Objects can be created by calls of the form `new("rCGH", ...)`.
Slots content are updated at each different analysis step, and are accessible through specific `get` functions.

Slots

`info`**:** Object of class "`character`": where sample information can be stored. See "getInfo" and "setInfo".

`cnSet`**:** Object of class "`data.frame`": the full data set. See "getCNset".

`param`**:** Object of class "`list`": the analysis parameters stored for traceability. See "getParam".

`segTable`**:** Object of class "`data.frame`": the segmentation table. See "getSegTable".

Methods

show `signature(object = "rCGH")`: ...

Author(s)

Frederic Commo

See Also

"rCGH-Agilent", "rCGH-SNP6", "rCGH-cytoScan", "rCGH-generic"

Examples

```
showClass("rCGH")
```

---

rCGH-cytoScan-class *Class* "rCGH-cytoScan"

Description

An instance of class "rCGH-cytoScan", which inherits from the superclass "rCGH".
Slots described below are used to store sample information, analysis parameters, and segmentation results. All are accessible through specific "Accessors" functions.

Objects from the Class

Objects can be created by calls of the form new("rCGH-cytoScan", ...).

Slots

info: Object of class "character": where sample information can be stored. See getInfo and setInfo.

cnSet: Object of class "data.frame": the full data set. See getCNset.

param: Object of class "list": the analysis parameters stored for traceability. getParam.

segTable: Object of class "data.frame": the segmentation table. getSegTable.

Extends

Class "rCGH", directly.

Methods

No methods defined with class "rCGH-cytoScan" in the signature.

Author(s)

Frederic Commo

See Also

"rCGH", "rCGH-Agilent", "rCGH-SNP6", "rCGH-generic"

Examples

```
showClass("rCGH-cytoScan")
```

rCGH-generic-class *Class* "rCGH-generic"

Description

An instance of class "rCGH-generic", which inherits from the superclass "rCGH".
Slots described below are used to store sample information, analysis parameters, and segmentation results. All are accessible through specific "Accessors" functions.

Objects from the Class

Objects can be created by calls of the form new("rCGH-generic", ...).

Slots

info: Object of class "character": where sample information can be stored. See getInfo and setInfo.

cnSet: Object of class "data.frame": the full data set. See getCNset.

param: Object of class "list": the analysis parameters stored for traceability. getParam.

segTable: Object of class "data.frame": the segmentation table. getSegTable.

Extends

Class "rCGH", directly.

Methods

No methods defined with class "rCGH-generic" in the signature.

Author(s)

Frederic Commo

See Also

"rCGH", "rCGH-Agilent", "rCGH-SNP6", "rCGH-cytoScan"

Examples

```
showClass("rCGH-generic")
```

rCGH-oncoScan-class *Class* "rCGH-oncoScan"

Description

An instance of class "rCGH-oncoScan", which inherits from the superclass "rCGH".
Slots described below are used to store sample information, analysis parameters, and segmentation results. All are accessible through specific "Accessors" functions.

Objects from the Class

Objects can be created by calls of the form new("rCGH-oncoScan", ...).

Slots

info: Object of class "character": where sample information can be stored. See getInfo and setInfo.

cnSet: Object of class "data.frame": the full data set. See getCNset.

param: Object of class "list": the analysis parameters stored for traceability. getParam.

segTable: Object of class "data.frame": the segmentation table. getSegTable.

Extends

Class "rCGH", directly.

Methods

No methods defined with class "rCGH-oncoScan" in the signature.

Author(s)

Frederic Commo

See Also

"rCGH", "rCGH-Agilent", "rCGH-SNP6", "rCGH-cytoScan", "rCGH-generic"

Examples

```
showClass("rCGH-oncoScan")
```

rCGH-SNP6-class *Class* "rCGH-SNP6"

Description

An instance of class "rCGH-SNP6", which inherits from the superclass "rCGH".
Slots described below are used to store sample information, analysis parameters, and segmentation results. All are accessible through specific "Accessors" functions.

Objects from the Class

Objects can be created by calls of the form new("rCGH-SNP6", ...).

Slots

info: Object of class "character": where sample information can be stored. See getInfo and setInfo.

cnSet: Object of class "data.frame": the full data set. See getCNset.

param: Object of class "list": the analysis parameters stored for traceability. getParam.

segTable: Object of class "data.frame": the segmentation table. getSegTable.

Extends

Class "rCGH", directly.

Methods

No methods defined with class "rCGH-SNP6" in the signature.

Author(s)

Frederic Commo

See Also

"rCGH", "rCGH-Agilent", "rCGH-cytoScan"

Examples

```
showClass("rCGH-SNP6")
```

readAffyCytoScan *Affymetrix CytoScanHD "rCGH-cytoScan" Constructor*

Description

A constructor function which takes an Affymetrix cytoScanHD cychp.txt (or cnchp.txt) file as input, possibly in a compressed format (.bz2 or .gz).
These files are exported from Chromosome Analysis Suite (ChAS) or Affymetrix Power Tools (see References section).

Usage

```
readAffyCytoScan(filePath, sampleName=NA, labName=NA,
    useProbes=c("snp", "cn", "all"), genome = c("hg19", "hg18", "hg38"),
    ploidy = 2, verbose=TRUE)
```

Arguments

filePath	: string. A path to an Affymetrix cytoScanHD cychp.txt (or cnchp.txt) file.
sampleName	: string. A sample Id. Optional.
labName	: string. A lab Id. Optional.
useProbes	: character. What probes to consider. Possible choices are SNP probes only ("snp", default), CN probes only ("cn"), or all the probes ("all").
genome	: string. The genome build to use. Supported genomes are hg18, hg19 (default), and hg38.
ploidy	: numeric. A priori plody value, when known, to adjust the estimation of copy numbers. Default is 2.
verbose	: logical. When TRUE (default), progress is printed.

Details

When available in the file preambule, several array information will be stored in Object@info: scanning date, grid version,...
Any other useful item can be stored using setInfo.

Value

An object of class "rCGH"

Author(s)

Frederic Commo

References

Affymetrix Power Tools

See Also

readAgilent, readAffySNP6, readGeneric, readAffyOncoScan, setInfo, getInfo

Examples

```
filePath <- system.file("extdata", "Affy_cytoScan.cyhd.CN5.CNCHP.txt.bz2",
    package = "rCGH")
cgh <- readAffyCytoScan(filePath, sampleName = "AffyScHD")
cgh
```

---

readAffyOncoScan *Affymetrix OncoScan "rCGH-oncoScan" Constructor*

Description

A constructor function which takes an Affymetrix oncoScan tabulated file as input, possibly in a compressed format (.bz2 or .gz).
This can be either a 'ProbeSets,CopyNumber.tsv' alone, or merged with its corresponding 'ProbeSets,AllelicData.tsv' file. See the details section.

Usage

```
readAffyOncoScan(filePath, sampleName=NA, labName=NA,
    genome = c("hg19", "hg18", "hg38"),
    ploidy = 2, verbose=TRUE)
```

Arguments

filePath	: string. A path to an Affymetrix .tsv file. See details.
sampleName	: string. A sample Id. Optional.
labName	: string. A lab Id. Optional.
genome	: string. The genome build to use. Supported genomes are hg18, hg19 (default), and hg38.
ploidy	: numeric. A priori plody value, when known, to adjust the estimation of copy numbers. Default is 2.
verbose	: logical. When TRUE (default), progress is printed.

Details

The Affymetrix Power Tools apt-copynumber-onco-ssa script produces 2 files: 'ProbeSets,CopyNumber.tsv' and 'ProbeSets,AllelicData.tsv'. Merging these 2 files may produce a unique file containing both probes Log2Ratio and AllelicDifference.

Value

An object of class "rCGH"

Author(s)

Frederic Commo

References

Affymetrix Power Tools

See Also

readAgilent, readAffySNP6, readGeneric, readAffyCytoScan, setInfo, getInfo

Examples

```
# Just a toy file
filePath <- system.file("extdata", "oncoscan.tsv.bz2", package = "rCGH")
cgh <- readAffyOncoScan(filePath, sampleName = "AffyOncoScan")
cgh
```

---

readAffySNP6 *Affymetrix SNP6 "rCGH-SNP6" Constructor*

Description

A constructor function which takes an Affymetrix SNP6 cychp.txt (or cnchp.txt) file as input, possibly in a compressed format (.bz2 or .gz).
These files are exported from Chromosome Analysis Suite (ChAS) or Affymetrix Power Tools (APT) (see the References section).

Usage

```
readAffySNP6(filePath, sampleName = NA, labName = NA,
    useProbes=c("snp", "cn", "all"), genome = c("hg19", "hg18", "hg38"),
    ploidy = 2, verbose = TRUE)
```

Arguments

filePath	: string. A path to an Affymetrix SNP6 cychp.txt (or cnchp.txt) file.
sampleName	: string. A sample Id. Optional.
labName	: string. A lab Id. Optional.
useProbes	: character. What probes to consider. Possible choices are SNP probes only ("snp", default), CN probes only ("cn"), or all the probes ("all").
genome	: string. The genome build to use. Supported genomes are hg18, hg19 (default), and hg38.
ploidy	: numeric. A priori ploidy value, when known, to adjust the estimation of copy numbers. Default is 2.
verbose	: logical. When TRUE (default), progress is printed.

Details

When available in the file preambule, several array information will be stored in `Object@info`: scanning date, grid version,...
Any other useful item can be stored using `setInfo`.

Value

An object of class `"rCGH"`

Author(s)

Frederic Commo

References

Affymetrix Power Tools

See Also

`readAgilent`, `readAffyCytoScan`, `readGeneric`, `readAffyOncoScan`, `setInfo`, `getInfo`

Examples

```
filePath <- system.file("extdata", "Affy_snp6_cnchp.txt.bz2", package = "rCGH")
cgh <- readAffySNP6(filePath, sampleName = "AffySNP6")
cgh
```

---

readAgilent *Agilent Dual-Color Hybridization "rCGH-Agilent" Constructor.*

Description

A constructor function taking as input an Agilent FE .txt file, exported from Feature Extraction, possibly in a compressed format (.bz2 or .gz).
Agilent from 44 to 400K are supported.

Usage

```
readAgilent(filePath, sampleName = NA,
    labName = NA, supFlags = TRUE, genome = c("hg19", "hg18", "hg38"),
    ploidy = 2, verbose = TRUE)
```

Arguments

filePath	: string. A path to an Agilent FE (.txt) file.
sampleName	: string. A sample Id. Optional.
labName	: string. A lab Id. Optional.
supFlags	: should the flagged probes be suppressed. Default is TRUE.
genome	: string. The genome build to use. Supported genomes are hg18, hg19 (default), and hg38.
ploidy	: numeric. A priori ploidy value, when known, to adjust the estimation of copy numbers. Default is 2.
verbose	: logical. if TRUE (default), progress is printed.

Details

When available in the file preambule, several array information will be stored in Object@info: scanning date, grid version,...
Any other useful item can be stored using setInfo.

Value

An object of class "rCGH"

Author(s)

Frederic Commo

See Also

readAffyCytoScan, readAffySNP6, readGeneric, readAffyOncoScan, setInfo, getInfo

Examples

```
filePath <- system.file("extdata", "Agilent4x180K.txt.bz2", package = "rCGH")
cgh <- readAgilent(filePath, sampleName = "Agilent4x180K", labName = "myLab")
cgh
```

---

readGeneric *Generic rCGH object "rCGH-generic" Constructor*

Description

A constructor function which takes a tabulated .txt file as input, possibly in a compressed format (.bz2 or .gz). Notice that precise column names are mandatory, see the details section.

Usage

```
readGeneric(filePath, sampleName=NA, labName=NA,
    genome = c("hg19", "hg18", "hg38"), ploidy = 2, verbose=TRUE)
```

Arguments

filePath	: string. A path to an Generic .txt file.
sampleName	: string. A sample Id. Optional.
labName	: string. A lab Id. Optional.
genome	: string. The genome build to use. Supported genomes are hg18, hg19 (default), and hg38.
ploidy	: numeric. A priori ploidy value, when known, to adjust the estimation of copy numbers. Default is 2.
verbose	: logical. When TRUE (default), progress is printed.

Details

This generic constructor does not expect any preamble. Mandatory columns are:

ProbeName: Character strings. Typicaly the probe ids.

ChrNum: numeric. The chromosome numbers. In case Chr X and Y are used and named as "X" and "Y", these notations will be converted into 23 and 24, respectively.

ChrStart: numeric. The chromosomal probes locations.

Log2Ratio: numeric. The corresponding Log2Ratios.

Value

An object of class "rCGH"

Author(s)

Frederic Commo

See Also

readAgilent, readAffySNP6, readAffyCytoScan, readAffyOncoScan, setInfo, getInfo

Examples

```
filePath <- system.file("extdata", "generic.txt.bz2", package = "rCGH")
cgh <- readGeneric(filePath, sampleName = "demo")
cgh
```

recenter<-methods *Recentering a Genomic Profile*

Description

This function allows the user to recenter a genomic profile stored in an object of class "rCGH". Peaks are indexed from 1 to k, from left to right, as they appear on the plotDensity after the EMnormalize step.

Usage

```
## S4 replacement method for signature 'rCGH'
recenter(object) <- value
```

Arguments

object	: An object of class "rCGH"
value	: numeric. What peak number to choose to recenter the genomic profile.

Value

An object of class "rCGH"

Note

When a profile is recentered, the stored workflow parameters are updated. see getParam.

Author(s)

Frederic Commo

See Also

EMnormalize, plotDensity

Examples

```
filePath <- system.file("extdata", "Affy_cytoScan.cyhd.CN5.CNCHP.txt.bz2",
    package = "rCGH")
cgh <- readAffyCytoScan(filePath, sampleName = "AffyScHD")
cgh <- adjustSignal(cgh, nCores=1)
cgh <- segmentCGH(cgh, nCores=1)
cgh <- EMnormalize(cgh)

# Default peak choice center the profile on the 1st peak
plotDensity(cgh)

# Recentering on the 2nd density peak
recenter(cgh) <- 2
```

```
plotDensity(cgh)
```

---

segmentCGH *Genomic Profile Segmentation*

Description

A function for performing the Log2Ratio segmentation on an object of class "rCGH". See the details section below.

Usage

```
## S4 method for signature 'rCGH'
segmentCGH(object, Smooth=TRUE, UndoSD = NULL,
minLen = 10, nCores=NULL, verbose = TRUE)
```

Arguments

object	: An object of class "rCGH"
Smooth	: logical. Should the LRR be smoothed before being segmented. See DNAcopy for details.
UndoSD	: numeric. When not specified (default is NULL), the UndoSD value is estimated from the Log2Ratios. See DNAcopy for details.
minLen	: numeric. The minimal length for a segment, in Kb. Shorter segments will be merged to the closest adjacent one. Default value is 10(Kb).
nCores	: numeric. The number of cores to use in order to speed up the computation. When NULL (default), half of the available cores are used. See mclapply.
verbose	: logical. if TRUE (default), progress is printed.

Details

This function is a wrapper for the DNAcopy, CNA and segment functions, which allows parallelization and data-driven parameterization.
In addition to the usual DNAcopy output, the segmentation table contains the probes Log2Ratio standard deviation for each segment, as well as there length, in Kb.

Value

An object of class "rCGH"

Author(s)

Frederic Commo

References

Venkatraman ES1, Olshen AB. A faster circular binary segmentation algorithm for the analysis of array CGH data.Bioinformatics. 2007 Mar 15;23(6):657-63.

See Also

CNA, segment, mclapply

Examples

```
filePath <- system.file("extdata", "Affy_cytoScan.cyhd.CN5.CNCHP.txt.bz2",
    package = "rCGH")
cgh <- readAffyCytoScan(filePath, sampleName = "AffyScHD")
cgh <- adjustSignal(cgh, nCores=1)
cgh <- segmentCGH(cgh, nCores=1)
st <- getSegTable(cgh)
head(st)
```

---

setInfo<-methods *Adding Information In An Object Of Class "rCGH"*

Description

This function allows the user to store any type of supplementaty information in an object of class "rCGH".

Usage

```
## S4 replacement method for signature 'rCGH'
setInfo(object, item = NULL) <- value
```

Arguments

object	: An object of class "rCGH"
item	: character. An item name to store. Default is NULL.
value	: any. A value to store.

Value

An object of class "rCGH"

Warning: When either item or value is NULL, an error is returned.

Author(s)

Frederic Commo

See Also

getInfo

Examples

```
filePath <- system.file("extdata", "Affy_cytoScan.cyhd.CN5.CNCHP.txt.bz2",
    package = "rCGH")
cgh <- readAffyCytoScan(filePath, sampleName = "AffyScHD")

# When supplementary information is added,
# numerical, logical, or strings are supported
setInfo(cgh, "someItem1") <- 35
setInfo(cgh, "someItem2") <- TRUE
setInfo(cgh, "someItem3") <- "someComment"
getInfo(cgh)

# or to get back specific items
getInfo(cgh, c("someItem1", "someItem3"))
```

---

show-methods show "rCGH"

Description

A method for visualizing an object of class "rCGH"

Methods

signature(object = "rCGH")

Author(s)

Frederic Commo

---

view *Interactive Genomic Profile Visualization*

Description

This function is build on top of shiny, and provides an iteractive way for visualizing a genomic profile, and exploring the list of genes.
From a command panel, the user can interact with the graph in different ways. See details.

Usage

```
## S4 method for signature 'rCGH'
view(object, browser = TRUE, ...)
```

Arguments

object	: An object of class "rCGH"
browser	: logical. When TRUE (default), the system's default web browser will be launched automatically.
...	: Optional parameters used by runApp.

Details

The left command panel allows the user several actions:

- displaying a specific gene by calling its HUGO symbol.
- showing all or one unique chromosome.
- merging segments shorter than a specified value, in Kb.
- recentering the entire profile.
- rescaling the y-axis.
- specifying the Log2Ratio cut offs for defining gains and losses.
- specifying a segment lenght cut off, in Mb.
- exporting the genomic plot.
- exporting the genes list.

Some actions, such as showing one unique chromosome or specifying cut offs (gain, loss, segment length), automatically update the gene table available in the *"Genes table"* tab.

Value

None.

Author(s)

Frederic Commo

See Also

plotProfile, plotLOH, multiplot, runApp

Examples

```
filePath <- system.file("extdata", "Affy_cytoScan.cyhd.CN5.CNCHP.txt.bz2",
    package = "rCGH")
cgh <- readAffyCytoScan(filePath, sampleName = "AffyScHD")
cgh <- adjustSignal(cgh, nCores=1)
cgh <- segmentCGH(cgh, nCores=1)
cgh <- EMnormalize(cgh)

## Not run:
# Interactive visalizations
view(cgh)
```

```
## End(Not run)
```

Index