Package 'PSCBS'

March 21, 2012

Title Analysis of Parent-Specific DNA Copy Numbers

Version 0.23.0

Date 2012-03-20

Author A.B. Olshen, H. Bengtsson, P. Neuvial, P.T. Spellman, R.A. Olshen, V.E. Seshan

Description Segments allele-specific DNA copy number data to detect regions with abnormal copy number within each parental chromosome

Maintainer Henrik Bengtsson <henrikb@braju.com>

Depends R.methodsS3 (>= 1.2.2), R.oo (>= 1.9.3), R.utils (>= 1.12.0)

Imports DNAcopy (>= 1.24.0), utils, R.rsp (>= 0.7.5)

Suggests DNAcopy (>= 1.24.0), aroma.light (>= 1.22.0), matrixStats (>= 0.4.4), R.oo (>= 1.9.3), R.cache (>= 0.6.1), digest (>= 0.5.1),Hmisc (>= 3.8-3), ggplot2 (>= 0.8.9)

License GPL (>= 2)

LazyLoad yes

biocViews aCGH, CopyNumberVariants, SNP, Microarray, OneChannel,TwoChannel, Genetics

Repository CRAN

Date/Publication 2012-03-21 07:59:23

R topics documented:

PSCBS-package *Package PSCBS*

Description

Segments allele-specific DNA copy number data to detect regions with abnormal copy number within each parental chromosome.

This package should be considered to be in an alpha or beta phase. You should expect the API to be changing over time.

Requirements

This package requires external packages R.methodsS3 (>= 1.2.1), R.oo (>= 1.8.3), R.utils (>= 1.11.0), DNAcopy (>= 1.24.0), utils, R.rsp (>= 0.7.3), and also suggests DNAcopy (>= 1.24.0), aroma.light (>= 1.22.0), matrixStats (>= 0.4.3), R.oo (>= 1.8.3), R.cache (>= 0.6.1), digest (>= 0.5.1), Hmisc (>= 3.8-3), ggplot2 (>= 0.8.9).

Installation and updates

To install this package, use `install.packages("PSCBS")`.

To get started

To get started, see:

1. `segmentByCBS()` - segments total copy-numbers, or any other unimodal genomic signals, using the CBS method [3,4].
2. `segmentByPairedPSCBS()` - segments allele-specific tumor signal from a tumor with a matched normal using the Paired PSCBS method [1,2].

How to cite

Please use [1] and [2] to cite when using Paired PSCBS, and [3] and [4] when using CBS.

License

GPL (>= 2).

Author(s)

A.B. Olshen, H. Bengtsson, P. Neuvial, P.T. Spellman, R.A. Olshen, V.E. Seshan.

References

[1] A.B. Olshen, H. Bengtsson, P. Neuvial, P.T. Spellman, R.A. Olshen, V.E. Seshan, *Parent-specific copy number in paired tumor-normal studies using circular binary segmentation*, Bioinformatics, 2011
[2] H. Bengtsson, P. Neuvial and T.P. Speed, *TumorBoost: Normalization of allele-specific tumor copy numbers from a single pair of tumor-normal genotyping microarrays*, BMC Bioinformatics, 2010
[3] A.B. Olshen, E.S. Venkatraman (aka Venkatraman E. Seshan), R. Lucito and M. Wigler, *Circular binary segmentation for the analysis of array-based DNA copy number data*, Biostatistics, 2004
[4] E.S. Venkatraman and A.B. Olshen, *A faster circular binary segmentation algorithm for the analysis of array CGH data*, Bioinformatics, 2007

`AbstractCBS` *The AbstractCBS class*

Description

Package: PSCBS
Class AbstractCBS

```
list
~~|
~~+--AbstractCBS
```

Directly known subclasses:
CBS, *PairedPSCBS*, *PSCBS*

public abstract static class **AbstractCBS**
extends list

All CBS-style segmentation results extend this class, e.g. `CBS` and `PairedPSCBS`.

Usage

```
AbstractCBS(fit=list(), sampleName=fit$sampleName, ...)
```

Arguments

`fit`	A `list` structure containing the segmentation results.
`sampleName`	A `character` string.
`...`	Not used.

Fields and Methods

Methods:

append	Appends one segmentation result to another.
extractCNs	-
getChromosomes	Gets the set of chromosomes.
getLocusData	Gets the locus-level data.
getSegments	Gets the segments.
getSegmentSizes	-
load	Loads an AbstractCBS object from file.
mergeThreeSegments	Merge a segment and its two flanking segments.
nbrOfChangePoints	Gets the number of change points.
nbrOfChromosomes	Gets the number of chromosomes.
nbrOfLoci	Gets the number of loci.
nbrOfSegments	Gets the number of segments.
plotTracks	Plots the segmentation result along the genome.
sampleCNs	-
save	Saves an AbstractCBS object to file.

Methods inherited from list:
all.equal, as.data.frame, attachLocally, callHooks, relist, within

Author(s)

Henrik Bengtsson (http://www.braju.com/R/)

`callSegmentationOutliers`
Calls single-locus outliers along the genome

Description

Calls single-locus outliers along the genome that have a signal that differ significantly from the neighboring loci.

Usage

```
## Default S3 method:
callSegmentationOutliers(y, chromosome=0, x=NULL, method="DNAcopy::smooth.CNA", ..., verbose=FALSE)
```

Arguments

y	A numeric vector of J genomic signals to be segmented.
chromosome	(Optional) An integer scalar (or a vector of length J contain a unique value). Only used for annotation purposes.
x	Optional numeric vector of J genomic locations. If NULL, index locations 1:J are used.
method	A character string specifying the Method used for calling outliers.
...	Additional arguments passed to internal outlier detection method.
verbose	See Verbose.

Value

Returns logical vector of length J.

Missing and non-finite values

Signals as well as genomic positions may contain missing values, i.e. NAs or NaNs. By definition, these cannot be outliers.

Author(s)

Henrik Bengtsson (http://www.braju.com/R/)

See Also

Internally smooth.CNA is utilized to identify the outliers.

CBS *The CBS class*

Description

A CBS object holds results from the Circular Binary Segmentation (CBS) method for *one* sample for one or more chromosomes.

Package: PSCBS
Class CBS

```
list
~~|
~~+--AbstractCBS
~~~~~~~|
~~~~~~~+--CBS
```

Directly known subclasses:

public abstract static class **CBS**
extends *AbstractCBS*

Usage

```
CBS(...)
```

Arguments

...	Arguments passed to the constructor of AbstractCBS.

Fields and Methods

Methods:

append	Appends one segmentation result to another.
as	-
estimateStandardDeviation	Estimates the whole-genome standard deviation of the signals.
getSegmentSizes	-
plotTracks	Plots copy numbers along the genome.
pruneBySdUndo	Prune the CBS profile by dropping change points that are too small.
segmentByCBS	-
writeSegments	Writes the table of segments to file.

Methods inherited from AbstractCBS:
all.equal, append, as.data.frame, drawChangePoints, dropChangePoint, dropChangePoints, dropRegion, dropRegions, extractChromosome, extractChromosomes, extractCNs, extractRegions, extractSegments, getChromosomes, getLocusData, getSampleName, getSegments, getSegmentSizes, load, mergeThreeSegments, mergeTwoSegments, nbrOfChangePoints, nbrOfChromosomes, nbrOfLoci, nbrOfSegments, plotTracks, print, pruneByHClust, renameChromosomes, resegment, sampleCNs, sampleName, sampleName<-, save, setLocusData, setSampleName, setSegments, tileChromosomes, updateMeans

Methods inherited from list:
all.equal, as.data.frame, attachLocally, callHooks, relist, within

Difference to DNAcopy object

A CBS object is similar to DNAcopy objects with the major difference that a CBS object holds only one sample, whereas a DNAcopy object can hold more than one sample.

See also

The segmentByCBS() method returns an object of this class.

Author(s)

Henrik Bengtsson

PairedPSCBS *The PairedPSCBS class*

Description

Package: PSCBS
Class PairedPSCBS

```
list
~~|
~~+--AbstractCBS
~~~~~~~|
~~~~~~~+--PSCBS
~~~~~~~~~~~~|
~~~~~~~~~~~~+--PairedPSCBS
```

Directly known subclasses:

public abstract static class **PairedPSCBS**
extends *PSCBS*

A PairedPSCBS is an object containing the results from the Paired PSCBS method.

Usage

```
PairedPSCBS(fit=list(), ...)
```

Arguments

`fit`	A `list` structure containing the Paired PSCBS results.
`...`	Not used.

Fields and Methods

Methods:

`callAB`	Calls segments that are in allelic balance.
`callCN`	Calls segments that are copy neutral.
`callCopyNeutral`	-
`callGainNeutralLoss`	-
`callGNL`	Calls segments that are gained, copy neutral, or lost.
`callLOH`	Calls segments that are in LOH.
`callROH`	Calls segments that are in ROH.
`estimateDeltaAB`	Estimate a threshold for calling allelic balance from DH.
`estimateDeltaLOH`	Estimate a threshold for calling LOH from DH.
`estimateKappa`	Estimate global background in segmented copy numbers.
`extractCNs`	-
`getSegmentSizes`	-
`plotTracks`	Plots parental specific copy numbers along the genome.

segmentByPairedPSCBS -

Methods inherited from PSCBS:
append, as.data.frame, drawChangePoints, extractChromosomes, getLocusData, isSegmentSplitter, writeSegments

Methods inherited from AbstractCBS:
all.equal, append, as.data.frame, drawChangePoints, dropChangePoint, dropChangePoints, dropRegion, dropRegions, extractChromosome, extractChromosomes, extractCNs, extractRegions, extractSegments, getChromosomes, getLocusData, getSampleName, getSegments, getSegmentSizes, load, mergeThreeSegments, mergeTwoSegments, nbrOfChangePoints, nbrOfChromosomes, nbrOfLoci, nbrOfSegments, plotTracks, print, pruneByHClust, renameChromosomes, resegment, sampleCNs, sampleName, sampleName<-, save, setLocusData, setSampleName, setSegments, tileChromosomes, updateMeans

Methods inherited from list:
all.equal, as.data.frame, attachLocally, callHooks, relist, within

Author(s)

Henrik Bengtsson (http://www.braju.com/R/)

See Also

The segmentByPairedPSCBS() method returns an object of this class.

PSCBS *The PSCBS class*

Description

Package: PSCBS
Class PSCBS

```
list
~~|
~~+--AbstractCBS
~~~~~~~|
~~~~~~~+--PSCBS
```

Directly known subclasses:
PairedPSCBS

public abstract static class **PSCBS**
extends *AbstractCBS*

A PSCBS is an object containing results from parent-specific copy-number (PSCN) segmentation.

Usage

```
PSCBS(fit=list(), ...)
```

Arguments

`fit`	A `list` structure containing the PSCN segmentation results.
`...`	Not used.

Fields and Methods

Methods:

`append`	Appends one segmentation result to another.
`writeSegments`	Writes the table of segments to file.

Methods inherited from AbstractCBS:
all.equal, append, as.data.frame, drawChangePoints, dropChangePoint, dropChangePoints, dropRegion, dropRegions, extractChromosome, extractChromosomes, extractCNs, extractRegions, extractSegments, getChromosomes, getLocusData, getSampleName, getSegments, getSegmentSizes, load, mergeThreeSegments, mergeTwoSegments, nbrOfChangePoints, nbrOfChromosomes, nbrOfLoci, nbrOfSegments, plotTracks, print, pruneByHClust, renameChromosomes, resegment, sampleCNs, sampleName, sampleName<-, save, setLocusData, setSampleName, setSegments, tileChromosomes, updateMeans

Methods inherited from list:
all.equal, as.data.frame, attachLocally, callHooks, relist, within

Author(s)

Henrik Bengtsson (http://www.braju.com/R/)

See Also

`PairedPSCBS`.

---

segmentByCBS *Segment genomic signals using the CBS method*

Description

Segment genomic signals using the CBS method of the **DNAcopy** package. This is a convenient low-level wrapper for the `DNAcopy::segment()` method. It is intended to be applied to a sample at the time. For more details on the Circular Binary Segmentation (CBS) method see [1,2].

Usage

```
## Default S3 method:
segmentByCBS(y, chromosome=0, x=NULL, index=seq(along = y), w=NULL, undo=Inf, ..., joinSegments=TRUE, k
```

Arguments

y	A numeric vector of J genomic signals to be segmented.
chromosome	Optional numeric vector of length J, specifying the chromosome of each loci. If a scalar, it is expanded to a vector of length J.
x	Optional numeric vector of J genomic locations. If NULL, index locations 1:J are used.
index	An optional integer vector of length J specifying the genomewide indices of the loci.
w	Optional numeric vector in [0,1] of J weights.
undo	A non-negative numeric. If less than +Inf, then arguments undo.splits="sdundo" and undo.SD=undo are passed to DNAcopy::segment().
...	Additional arguments passed to the DNAcopy::segment() segmentation function.
joinSegments	If TRUE, there are no gaps between neighboring segments. If FALSE, the boundaries of a segment are defined by the support that the loci in the segments provides, i.e. there exist a locus at each end point of each segment. This also means that there is a gap between any neighboring segments, unless the change point is in the middle of multiple loci with the same position. The latter is what DNAcopy::segment() returns.
knownSegments	Optional data.frame specifying *non-overlapping* known segments. These segments must not share loci. See findLargeGaps() and gapsToSegments().
seed	An (optional) integer specifying the random seed to be set before calling the segmentation method. The random seed is set to its original state when exiting. If NULL, it is not set.
verbose	See Verbose.

Details

Internally segment of **DNAcopy** is used to segment the signals. This segmentation method support weighted segmentation.

Value

Returns a CBS object.

Reproducibility

The DNAcopy::segment() implementation of CBS uses approximation through random sampling for some estimates. Because of this, repeated calls using the same signals may result in slightly different results, unless the random seed is set/fixed.

Missing and non-finite values

Signals may contain missing values (NA or NaN), but not infinite values (+/-Inf). Loci with missing-value signals are preserved and keep in the result.

Likewise, genomic positions may contain missing values. However, if they do, such loci are silently excluded before performing the segmentation, and are not kept in the results. The mapping between the input locus-level data and ditto of the result can be inferred from the `index` column of the locus-level data of the result.

None of the input data may have infinite values, i.e. -Inf or +Inf. If so, an informative error is thrown.

Author(s)

Henrik Bengtsson (http://www.braju.com/R/)

References

[1] A.B. Olshen, E.S. Venkatraman (aka Venkatraman E. Seshan), R. Lucito and M. Wigler, *Circular binary segmentation for the analysis of array-based DNA copy number data*, Biostatistics, 2004
[2] E.S. Venkatraman and A.B. Olshen, *A faster circular binary segmentation algorithm for the analysis of array CGH data*, Bioinformatics, 2007

See Also

To segment allele-specific tumor copy-number signals from a tumor with a matched normal, see segmentByPairedPSCBS().

Examples

```
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
# Simulating copy-number data
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
set.seed(0xBEEF)

# Number of loci
J <- 1000

mu <- double(J)
mu[200:300] <- mu[200:300] + 1
mu[350:400] <- NA # centromere
mu[650:800] <- mu[650:800] - 1
eps <- rnorm(J, sd=1/2)
y <- mu + eps
x <- sort(runif(length(y), max=length(y))) * 1e5
w <- runif(J)
w[650:800] <- 0.001


# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
# Segmentation
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
fit <- segmentByCBS(y, x=x)
print(fit)
plotTracks(fit)



xlab <- "Position (Mb)"
ylim <- c(-3,3)
xMb <- x/1e6
plot(xMb,y, pch=20, col="#aaaaaa", xlab=xlab, ylim=ylim)
drawLevels(fit, col="red", lwd=2, xScale=1e-6)


# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
# TESTS
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
fit <- segmentByCBS(y, x=x, seed=0xBEEF)
print(fit)
##   id chromosome       start      end nbrOfLoci    mean
## 1  y          0    55167.82 20774251       201  0.0164
## 2  y          0 20774250.85 29320105        99  1.0474
## 3  y          0 29320104.86 65874675       349 -0.0227
## 4  y          0 65874675.06 81348129       151 -1.0813
## 5  y          0 81348129.20 99910827       200 -0.0612


# Test #1: Reverse the ordering and segment
fitR <- segmentByCBS(rev(y), x=rev(x), seed=0xBEEF)
# Sanity check
stopifnot(all.equal(getSegments(fitR), getSegments(fit)))
# Sanity check
stopifnot(all.equal(rev(getLocusData(fitR)$index), getLocusData(fit)$index))

# Test #2: Reverse, but preserve ordering of 'data' object
fitRP <- segmentByCBS(rev(y), x=rev(x), preserveOrder=TRUE)
stopifnot(all.equal(getSegments(fitRP), getSegments(fit)))


# (Test #3: Change points inbetween data points at the same locus)
x[650:654] <- x[649]
fitC <- segmentByCBS(rev(y), x=rev(x), preserveOrder=TRUE, seed=0xBEEF)

# Test #4: Allow for some missing values in signals
y[450] <- NA
fitD <- segmentByCBS(y, x=x, seed=0xBEEF)


# Test #5: Allow for some missing genomic annotations
x[495] <- NA
fitD <- segmentByCBS(y, x=x, seed=0xBEEF)
```

```
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
# MISC.
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
# Emulate a centromere
x[650:699] <- NA
fit <- segmentByCBS(y, x=x, seed=0xBEEF)
xMb <- x/1e6
plot(xMb,y, pch=20, col="#aaaaaa", xlab=xlab, ylim=ylim)
drawLevels(fit, col="red", lwd=2, xScale=1e-6)

fitC <- segmentByCBS(y, x=x, joinSegments=FALSE, seed=0xBEEF)
drawLevels(fitC, col="blue", lwd=2, xScale=1e-6)



# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
# Multiple chromosomes
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
# Appending CBS results
fit1 <- segmentByCBS(y, chromosome=1, x=x)
fit2 <- segmentByCBS(y, chromosome=2, x=x)
fit <- append(fit1, fit2)
print(fit)
plotTracks(fit, subset=NULL, lwd=2, Clim=c(-3,3))


# Segmenting multiple chromosomes at once
chromosomeWG <- rep(1:2, each=J)
xWG <- rep(x, times=2)
yWG <- rep(y, times=2)
fitWG <- segmentByCBS(yWG, chromosome=chromosomeWG, x=xWG)
print(fitWG)
plotTracks(fitWG, subset=NULL, lwd=2, Clim=c(-3,3))

# Assert same results
fit$data[,"index"] <- getLocusData(fitWG)[,"index"] # Ignore 'index'
stopifnot(all.equal(getLocusData(fitWG), getLocusData(fit)))
stopifnot(all.equal(getSegments(fitWG), getSegments(fit)))
```

`segmentByPairedPSCBS` *Segment total copy numbers and allele B fractions using the Paired PSCBS method*

Description

Segment total copy numbers and allele B fractions using the Paired PSCBS method [1]. This method requires matched normals. This is a low-level segmentation method. It is intended to be applied to one tumor-normal sample at the time.

Usage

```
## Default S3 method:
segmentByPairedPSCBS(CT, betaT, betaN=NULL, muN=NULL, chromosome=0, x=NULL, alphaTCN=0.009, alphaDH=0.
```

Arguments

CT	A numeric vector of J tumor total copy number (TCN) ratios in [0,+Inf) (due to noise, small negative values are also allowed). The TCN ratios are typically scaled such that copy-neutral diploid loci have a mean of two.
betaT	A numeric vector of J tumor allele B fractions (BAFs) in [0,1] (due to noise, values may be slightly outside as well) or NA for non-polymorphic loci.
betaN	A numeric vector of J matched normal BAFs in [0,1] (due to noise, values may be slightly outside as well) or NA for non-polymorphic loci.
muN	An optional numeric vector of J genotype calls in {0,1/2,1} for AA, AB, and BB, respectively, and NA for non-polymorphic loci. If not given, they are estimated from the normal BAFs using callNaiveGenotypes as described in [2].
chromosome	(Optional) An integer scalar (or a vector of length J), which can be used to specify which chromosome each locus belongs to in case multiple chromosomes are segments. This argument is also used for annotation purposes.
x	Optional numeric vector of J genomic locations. If NULL, index locations 1:J are used.
alphaTCN, alphaDH	The significance levels for segmenting total copy numbers (TCNs) and decrease-in-heterozygosity signals (DHs), respectively.
undoTCN, undoDH	Non-negative numerics. If less than +Inf, then a cleanup of segmentions post segmentation is done. See argument undo of segmentByCBS() for more details.
...	Additional arguments passed to segmentByCBS().
flavor	A character specifying what type of segmentation and calling algorithm to be used.
tbn	If TRUE, betaT is normalized before segmentation using the TumorBoost method [2], otherwise not.
joinSegments	If TRUE, there are no gaps between neighboring segments. If FALSE, the boundaries of a segment are defined by the support that the loci in the segments provides, i.e. there exist a locus at each end point of each segment. This also means that there is a gap between any neighboring segments, unless the change point is in the middle of multiple loci with the same position. The latter is what DNAcopy::segment() returns.
knownSegments	Optional data.frame specifying *non-overlapping* known segments. These segments must not share loci. See findLargeGaps() and gapsToSegments().
seed	An (optional) integer specifying the random seed to be set before calling the segmentation method. The random seed is set to its original state when exiting. If NULL, it is not set.
verbose	See Verbose.

Details

Internally segmentByCBS() is used for segmentation. The Paired PSCBS segmentation method does *not* support weights.

Value

Returns the segmentation results as a PairedPSCBS object.

Reproducibility

The "DNAcopy::segment" implementation of CBS uses approximation through random sampling for some estimates. Because of this, repeated calls using the same signals may result in slightly different results, unless the random seed is set/fixed.

Whole-genome segmentation is preferred

Although it is possible to segment each chromosome independently using Paired PSCBS, we strongly recommend to segment whole-genome (TCN,BAF) data at once. The reason for this is that downstream CN-state calling methods, such as the AB and the LOH callers, performs much better on whole-genome data. In fact, they may fail to provide valid calls if done chromsome by chromosome.

Missing and non-finite values

The total copy number signals as well as any optional positions must not contain missing values, i.e. NAs or NaNs. If there are any, an informative error is thrown. Allele B fractions may contain missing values, because such are interpreted as representing non-polymorphic loci.

None of the input signals may have infinite values, i.e. -Inf or +Inf. If so, an informative error is thrown.

Paired PSCBS with only genotypes

If allele B fractions for the matched normal (betaN) are not available, but genotypes (muN) are, then it is possible to run a version of Paired PSCBS where TumorBoost normalization of the tumor allele B fractions is skipped. In order for this to work, argument tbn must be set to FALSE.

Author(s)

Henrik Bengtsson (http://www.braju.com/R/)

References

[1] A.B. Olshen, H. Bengtsson, P. Neuvial, P.T. Spellman, R.A. Olshen, V.E. Seshan, *Parent-specific copy number in paired tumor-normal studies using circular binary segmentation*, Bioinformatics, 2011

[2] H. Bengtsson, P. Neuvial and T.P. Speed, *TumorBoost: Normalization of allele-specific tumor copy numbers from a single pair of tumor-normal genotyping microarrays*, BMC Bioinformatics, 2010

See Also

Internally, callNaiveGenotypes is used to call naive genotypes, normalizeTumorBoost is used for TumorBoost normalization, and segmentByCBS() is used to segment TCN and DH separately.

To segment total copy-numbers, or any other unimodal signals, see segmentByCBS().

Examples

```
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
# Load SNP microarray data
# (note to package developers: this example data set may
#  be replaced in a future release of the package)
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
pathname <- system.file("data-ex/PairedPSCBS,exData,chr01.Rbin", package="PSCBS")
data <- R.utils::loadObject(pathname)
str(data)


# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
# Paired PSCBS segmentation
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
# Drop single-locus outliers
dataS <- dropSegmentationOutliers(data)

# Speed up example by segmenting fewer loci
dataS <- dataS[seq(from=1, to=nrow(data), by=5),]

str(dataS)

R.oo::attachLocally(dataS)

# Paired PSCBS segmentation
fit <- segmentByPairedPSCBS(CT, betaT=betaT, betaN=betaN,
                            chromosome=chromosome, x=x,
                            seed=0xBEEF, verbose=-10)
print(fit)

# Plot results
plotTracks(fit)


# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
# Bootstrap segment level estimates
# (used by the AB caller, which, if skipped here,
#  will do it automatically)
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
fit <- bootstrapTCNandDHByRegion(fit, verbose=-10)
print(fit)
plotTracks(fit)


# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
# Calling segments in allelic balance (AB)
# NOTE: Ideally, this should be done on whole-genome data
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
# Explicitly estimate the threshold in DH for calling AB
# (which be done by default by the caller, if skipped here)
deltaAB <- estimateDeltaAB(fit, flavor="qq(DH)", verbose=-10)
print(deltaAB)
## [1] 0.1657131

fit <- callAB(fit, delta=deltaAB, verbose=-10)
print(fit)
plotTracks(fit)

# Even if not explicitly specified, the estimated
# threshold parameter is returned by the caller
stopifnot(fit$params$deltaAB == deltaAB)


# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
# Calling segments in loss-of-heterozygosity (LOH)
# NOTE: Ideally, this should be done on whole-genome data
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
# Explicitly estimate the threshold in C1 for calling LOH
# (which be done by default by the caller, if skipped here)
deltaLOH <- estimateDeltaLOH(fit, flavor="minC1|nonAB", verbose=-10)
print(deltaLOH)
## [1] 0.625175

fit <- callLOH(fit, delta=deltaLOH, verbose=-10)
print(fit)
plotTracks(fit)

# Even if not explicitly specified, the estimated
# threshold parameter is returned by the caller
stopifnot(fit$params$deltaLOH == deltaLOH)
```

Index