Package 'bams'

July 2, 2014

Maintainer Toby Dylan Hocking <toby@sg.cs.titech.ac.jp>

Author Toby Dylan Hocking

Version 1.6

License GPL-3

Title Breakpoint annotation model smoothing

Description Code and data to compare several change-point detection models on DNA copy number profiles.

Suggests GLAD, DNAcopy, cghFLasso, flsa, cghseg, grid, gada, plyr,ggplot2 (>= 0.9.0), RColorBrewer, reshape2, lattice, proto,changepoint

Depends R (>= 2.10)

NeedsCompilation no

Repository CRAN

Date/Publication 2013-03-13 11:10:56

R topics documented:

`bams-package` *Breakpoint annotation model smoothing*

Description

Code and data to compare several change-point detection models on DNA copy number profiles.

Details

Package:	bams
Maintainer:	Toby Dylan Hocking <toby@sg.cs.titech.ac.jp>
Author:	Toby Dylan Hocking
Version:	1.6
License:	GPL-3
Title:	Breakpoint annotation model smoothing
Suggests:	GLAD, DNAcopy, cghFLasso, flsa, cghseg, grid, gada, plyr, ggplot2 (>= 0.9.0), RColorBrewer, reshape2, lattic
Depends:	R (>= 2.10)

Author(s)

Toby Dylan Hocking

`article.smoothers` *article smoothers*

Description

Smoothing functions used in the article.

Usage

```
article.smoothers
```

dnacopy.smoothvec *dnacopy smoothvec*

Description

Smooth a profile using DNAcopy.

Usage

```
dnacopy.smoothvec(profile, var, vals, ...)
```

Arguments

profile	A profile data.frame.
var	Smoothness variable.
vals	Smoothness values.
...	Other arguments, passed to segment.

Value

Matrix of smoothed profiles: nparam x nprobes.

Author(s)

Toby Dylan Hocking

each.chrom *each chrom*

Description

Apply a smoothing function independently to each chromosome of a profile.

Usage

```
each.chrom(profile, FUN)
```

Arguments

profile	Profile data.frame.
FUN	Function that will take a profile data.frame for one chromosome and return a smoothing matrix for that chromosome: nparam x nprobes.

Value

Matrix of smoothed profiles for the entire profile: nparam x nprobes.

Author(s)

Toby Dylan Hocking

`fit.gada` *fit gada*

Description

Run the first fitting steps of the gada algorithm.

Usage

```
fit.gada(pro)
```

Arguments

`pro`	Profile data.frame.

Author(s)

Toby Dylan Hocking

`gada.results` *gada results*

Description

Recover a matrix of smoothed signals from the gada results.

Usage

```
gada.results(pro, fit.list)
```

Arguments

`pro`	Profile data.frame.
`fit.list`	List of gada results. Each gada result is a list, one element for each chromosome.

Author(s)

Toby Dylan Hocking

geom_tallrect *geom tallrect*

Description

ggplot2 geom with xmin and xmax aesthetics that covers the entire y range.

Usage

```
geom_tallrect(mapping = NULL, data = NULL, stat = "identity",
    position = "identity", ...)
```

Arguments

mapping	
data	
stat	
position	
...	

Author(s)

Toby Dylan Hocking

neuroblastomaDetailed *Detailed annotations of the neuroblastoma data*

Description

An annotation is the number of breakpoints that an expert expects of a segmentation model in a certain region, after visual inspection of the scatterplot of data.

Usage

```
data(neuroblastomaDetailed)
```

Format

A data frame with 4359 observations on the following 5 variables.

`profile.id` a factor with levels corresponding to the profile.id column of `neuroblastoma$profiles`.

`chromosome` idem for `neuroblastoma$chromosome`.

`min` first position of the annotated region in base pairs.

`max` idem for the last position.

`annotation` factor indicating the number of breakpoints in this region: `>0breakpoints` means at least one breakpoint, `1breakpoint` means exactly 1 breakpoint, `normal` means exactly 0 breakpoints.

Details

The neuroblastoma data are a set of 575 DNA copy number profiles of neuroblastoma tumors, available as `data(neuroblastoma,package="neuroblastoma")`. That package provides the "original" set of up to 6 annotated regions per profile as `neuroblastoma$annotations`. There is at most 1 annotation per chromosome, and 2 types of annotations: `breakpoint` means 1 or more breakpoints and `normal` means exactly 0 breakpoints. These data were made by Gudrun Schleiermacher and Isabelle Janoueix-Lerosey, by typing 0 or 1 in a spreadsheet after visual inspection of the profiles.

This package provides a different set of annotations of the same data. We say they are detailed since there is often more than 1 annotation per chromosome, and there is another type of annotation: `1breakpoint` means there is exactly 1 breakpoint in that region. These annotations were created by Toby Dylan Hocking and Valentina Boeva using GUIs which allow drawing regions on the plotted data.

Source

http://cbio.ensmp.fr/~thocking/neuroblastoma/annotations.csv

`pick.best.index` *pick best index*

Description

Minimizer for local models, described in article section 2.3 "Picking the optimal model"

Usage

```
pick.best.index(err)
```

Arguments

`err`	Vector of errors to minimize.

Value

Integer index of the minimal error.

Author(s)

Toby Dylan Hocking

Examples

```
stopifnot(pick.best.index(rep(0,100))==50)

err <- rep(1,100)
err[5] <- 0
stopifnot(pick.best.index(err)==5)

## should pick the middle
err <- rep(1,100)
err[40:60] <- 0
stopifnot(pick.best.index(err)==50)

## should pick the biggest
err <- rep(1,100)
err[1:60] <- 0
stopifnot(pick.best.index(err)==60)

## should pick the smallest
err <- rep(1,100)
err[50:100] <- 0
stopifnot(pick.best.index(err)==50)
```

`run.cghseg` *run cghseg*

Description

Run cghseg maximum likelihood DP segmentation and pick the model using picker.

Usage

```
run.cghseg(profile, picker)
```

Arguments

`profile`	Profile data.frame.
`picker`	Function that gets arguments const.lines (a data.frame with one line for each segment in the maximum likelihood segmentation for a chrom), smoothed.mat (matrix of smoothed profile for a chrom: kmax x nprobes), Y (vector of logratio measurements for a chrom), kmax (maximum number of segments to consider), n (number of probes), and should return the chosen smoothing vector for the chrom.

Value

Smoothing matrix nparam x nprobes.

Author(s)

Toby Dylan Hocking

`run.pelt` *run pelt*

Description

Smooth a profile using the PELT algorithm.

Usage

```
run.pelt(profile, penalty = "SIC", values = 0, FUN = cpt.mean,
    format = NULL)
```

Arguments

`profile`	A profile data.frame.
`penalty`	character specifying the penalty to use.
`values`	vector of penalty parameters to try.
`FUN`	PELT function to use.
`format`	if character, use sprintf(format, values) for values.

Value

Matrix of smoothed profiles: nparam x nprobes.

Author(s)

Toby Dylan Hocking

`runglad` *runglad*

Description

Run glad to smooth a profile.

Usage

```
runglad(profile, ...)
```

Arguments

profile	Profile data.frame.
...	Smoothing parameter for glad.

Value

Smoothing matrix nparam x nprobes.

Author(s)

Toby Dylan Hocking

---

seg.profile *seg profile*

Description

Run several smoothers on a profile, saving the detected breakpoint locations to disk.

Usage

```
seg.profile(profile, smooth.funs = smoothers, tosave = c("seconds",
    "parameters", "breakpoints"), db = file.path(Sys.getenv("HOME"),
    "seg"))
```

Arguments

profile	Profile data.frame.
smooth.funs	List of smoothing functions to apply to the profile.
tosave	Variables to save to the db directory.
db	Location to save gzipped result files.

Value

Nothing, the results are saved to files.

Author(s)

Toby Dylan Hocking

smoothers *smoothers*

Description

This is a list of functions, each of which must return a matrix of smoothed profiles. The first argument of each function is a data.frame that represents a copy number profile, with at least columns: position logratio chromosome. We assume that positions are already sorted in ascending order p_1 < p_2. The second argument to each of these functions should be a vector of smoothing parameters, and there should be a default value. The matrix returned has 1 row for each parameter, and 1 column for each position.

Usage

```
smoothers
```

Index