Package ‘MinimumDistance’

March 22, 2017

Title  A Package for De Novo CNV Detection in Case-Parent Trios
Version 1.18.0
Date 2016-03-05
Author Robert B Scharpf and Ingo Ruczinski
Maintainer Robert B Scharpf <rscharpf@jhu.edu>
Description Analysis of de novo copy number variants in trios from high-dimensional genotyping platforms.
License Artistic-2.0
Depends R (>= 3.3), VanillaICE (>= 1.31.3)
Imports methods, BiocGenerics, Biobase, S4Vectors (>= 0.9.25), IRanges, GenomeInfoDb, GenomicRanges (>= 1.17.16), SummarizedExperiment (>= 0.2.0), oligoClasses, DNAcopy, ff, foreach, matrixStats, lattice, data.table, grid, stats, utils
Suggests human610quadv1bCrlmm (>= 1.0.3), BSgenome.Hsapiens.UCSC.hg18, BSgenome.Hsapiens.UCSC.hg19, SNPchip, RUnit
Enhances snow, doSNOW
LazyLoad yes
biocViews Microarray, SNP, CopyNumberVariation
RoxygenNote 5.0.1
NeedsCompilation no
R topics documented:

acf2 .......................................................... 3
calculateMindist ............................................ 3
coerce .......................................................... 4
denovoHemizygous ........................................... 5
DNAcopyParam ................................................. 6
exampleTrioSetList .......................................... 6
filterExperiment ............................................... 7
FilterParamMD .................................................. 7
FilterParamMD-class ......................................... 8
mad2 ........................................................... 9
MAP ........................................................... 9
MAP2 .......................................................... 10
mdLegend ......................................................... 11
MDRanges-class ............................................... 11
md_exp ......................................................... 12
md_gr .......................................................... 13
mindist ........................................................ 13
MinDistExperiment .......................................... 14
MinDistExperiment-class ................................... 14
MinDistGRanges .............................................. 15
MinDistGRanges-class ....................................... 16
MinDistParam .................................................. 17
MinDistParam-class ......................................... 18
MinDistPosterior-class ...................................... 19
MinimumDistance ........................................... 19
nMAD .......................................................... 20
offspringNames ............................................... 20
ParentOffspring-class ....................................... 21
ParentOffspringList-class .................................. 22
Pedigree ......................................................... 23
Pedigree-class ................................................ 24
pedigreeGrid ................................................... 24
pedigreeName ................................................. 25
pedigreeViewports ........................................... 26
PennParam ....................................................... 26
plotDenovo ...................................................... 27
range,ILimit-method ........................................ 28
segment2 ....................................................... 28
TrioSet ........................................................ 29
TrioSet-class ................................................... 30
TrioSetList ..................................................... 31
TrioSetList-class ............................................. 32
TrioSetListLD .................................................. 34

Index 35
acf2

Function for computing autocorrelations

Description
By default, this function returns the lag-10 autocorrelations of a numeric vector and omits missing values.

Usage
acf2(x, lag.max = 10, type = c("correlation", "covariance", "partial"),
plot = FALSE, na.action = na.omit, demean = TRUE, ...)

Arguments
- x: a numeric vector
- lag.max: see acf
- type: see acf
- plot: logical, as in acf
- na.action: ignored. Missing values are automatically omitted.
- demean: logical, as in acf
- ...: additional arguments passed to acf

See Also
acf

Examples
x <- rnorm(100)
x[5] <- NA
acf2(x)

calculateMindist

Deprecated function to calculate the minimum distance

Description
The 'minimum distance' is the minimum signed absolute difference of the parental log R ratios and the offspring log R ratios. Specifically, let IO-Fl denote the absolute difference in the log R ratios comparing offspring to father and IO-Ml the absolute difference in the log R ratios comparing offspring to mother. The minimum distance at a marker is the signed minimum of IO-Ml and IO-Fl. After segmentation of the minimum distance, non-zero segments can indicate a de novo difference in the log R ratio of the offspring and either parent. For example, a positive minimum distance suggests that the log R ratio from the offspring is greater than the log R ratio of either parent.
Usage

```r
calculateMindist(object, ...)  
## S4 method for signature 'TrioSet'
calculateMindist(object, verbose = TRUE, ...)
## S4 method for signature 'TrioSetList'
calculateMindist(object)
## S4 method for signature 'list'
calculateMindist(object, outdir = ldPath(), ...)
## S4 method for signature 'arrayORff_array'
calculateMindist(object, outdir, center, ...)
## S4 method for signature 'matrix'
calculateMindist(object, ...)
```

Arguments

- `object`: see `showMethods("calculateMindist")`
- `...`: Ignored
- `verbose`: logical. Whether to display messages indicating progress.
- `outdir`: character string indicating path to save output
- `center`: ignored

Description

Coercion methods in `MinimumDistance` package

Coerces a `TrioSetList` to a `TrioSet`

Usage

```r
as(from, to)
## S4 method for signature 'TrioSetList'
stack(x, ...)
```

Arguments

- `x`: a `TrioSetList`
- `...`: ignored
- `from`: see `showMethods("coerce")`
- `to`: see `showMethods("coerce")`

Value

- a `TrioSet`
**denovoHemizygous**

Filter the genomic intervals for denovo copy number states

**Description**

This function filters the genomic intervals for denovo events.

**Usage**

```r
denovoHemizygous(object, filters = FilterParamMD(state = "221"))
denovoHomozygous(object, filters = FilterParamMD(state = "220"))
denoovo(object, filters = FilterParamMD(state = c("220", "221", "224")))
denovoDuplication(object, filters = FilterParamMD(state = "224"))
```

Arguments

- **object**: see `showMethods(denovo)` for a list of defined methods
- **filters**: an object of class `FilterParamMD`

**Details**

The function `denovo` filters genomic intervals for states '220', '221', and '224', corresponding to denovo homozygous deletion, denovo hemizygous deletion, and denovo duplication, respectively.

- `denovoHemizygous` filters genomic intervals for state '221'.
- `denovoHomozygous` filters genomic intervals for state '220'.
- `denovoDuplication` filters genomic intervals for state '224'.

**See Also**

- `FilterParamMD-class`
DNAcopyParam  Constructor for DNAcopyParam class

Description

Creates an instance of a parameter class for circular binary segmentation of the minimum distance and the log R ratios. Parameters in this object are passed to the segment function in the package DNAcopy.

Usage

DNAcopyParam(alpha = 0.01, min.width = 2L, undo.splits = c("none", "prune", "sundo"), undo.SD = 3)

Arguments

alpha  see segment
min.width  see segment
undo.splits  see segment
undo.SD  see segment

See Also

segment

Examples

segment_params <- DNAcopyParam(alpha = 0.01)
params <- MinDistParam(dnacopy = segment_params)

dna

exampleTrioSetList  An example TrioSetList object

Description

A dataset containing log R ratios and B allele frequencies for two chromosomes, organized as a TrioSetList. Each element in the list class is a TrioSet object. Both TrioSetList and TrioSet classes are deprecated; the example data will be removed in a future release.

Format

a TrioSetList
**filterExperiment**  
*Methods for filtering MinDistExperiment objects*

**Description**
Filter a MinDistExperiment object to exclude markers with missing values in the low-level summaries, exclude markers that lie in segments (granges argument) with small minimum distance values (unlikely to be de novo)

**Usage**
```
filterExperiment(object, granges, param)
## S4 method for signature 'MinDistExperiment,GRanges'
filterExperiment(object, granges, param)
## S4 method for signature 'MinDistExperiment,GRangesList'
filterExperiment(object, granges, param)
## S4 method for signature 'MinDistExperiment,MinDistGRanges'
filterExperiment(object, granges, param)
```

**Arguments**
- `object` A MinDistExperiment
- `granges` A GRanges, GRangesList, or MinDistGRanges object
- `param` a MinDistParam object

**Value**
a MinDistExperiment

**FilterParamMD**  
*Parameters for filtering results from the segmentation and copy number inference*

**Description**
A container for criteria used to filter the segmentation results post-hoc. Options including filtering on the posterior call, the posterior probability of the posterior call, the minimum number of markers spanned by the segment, the minimum width of the segment, and chromosome. Convenience functions are available for commonly used filters.

**Usage**
```
FilterParamMD(state = trioStateNames(), seqnames = paste0("chr", 1:22), ...)
```
FilterParamMD-class

A class for filtering genomic intervals called by MinimumDistance

Description

Options for filtering include the number of markers spanned by a segment, the posterior probability of the maximum a posteriori estimate of the trio copy number state, and the trio copy number state.

Usage

### S4 method for signature 'FilterParamMD'

```r
show(object)
```

Arguments

- **object**: a `FilterParamMD` object

See Also

denovo
mad2

Deprecated wrapper for computing the median absolute deviation of low-level summaries

Description

Deprecated wrapper for computing the median absolute deviation of low-level summaries

Usage

mad2(object, byrow = FALSE, ...)

## S4 method for signature 'list'
mad2(object, byrow, pedigree, ...)

## S4 method for signature 'TrioSetList'
mad2(object, byrow = FALSE, ...)

## S4 method for signature 'matrix'
mad2(object, byrow, pedigree, ...)

## S4 method for signature 'array'
mad2(object, byrow, pedigree, ...)

Arguments

- **object**: see showMethods("mad2")
- **byrow**: logical if TRUE, compute the median absolute deviation of the rows of a matrix
- **...**: additional arguments to mad
- **pedigree**: an object of class Pedigree

MAP

Computes the maximum a posteriori trio copy number state for the segmented minimum distance

Description

This function is deprecated and will be defunct in a future release. The replacement function is MAP2.

Usage

MAP(object, ranges, id, TAUP = 1e+10, tauMAX = 1 - 5e-08, cnStates = c(-2, -0.4, 0, 0, 0.4, 1), pr.nonmendelian = 1.5e-06, mdThr = 0.9, ...)
Arguments

- **object**: see `showMethods(MAP)`
- **ranges**: A GRanges object
- **id**: character string for sample identifier
- **TAUP**: scalar for transition probabilities
- **tauMAX**: the maximum probability that the current state is the same as the previous state
- **cnStates**: character vector for hidden Markov model state labels
- **pr.nonmendelian**: numeric: the a priori probability of a non-Mendelian copy number alteration
- **mdThr**: a length-one numeric vector. A minimum distance below this threshold in absolute value will not be evaluated for copy number alterations.
- **...**: Ignored.

---

\textit{MAP2} \hspace{1cm} \textit{Computes maximum a posteriori estimate for the trio copy number state}

Description

Computes maximum a posteriori estimate for the trio copy number state

Usage

\begin{verbatim}
MAP2(object, mdgr, param = MinDistParam(), ...)
\end{verbatim}

## S4 method for signature 'MinDistExperiment,MinDistGRanges'
MAP2(object, mdgr, param = MinDistParam(), ...)

## S4 method for signature 'MinDistExperiment,GRangesList'
MAP2(object, mdgr, param = MinDistParam(), ...)

## S4 method for signature 'MinDistExperiment,GRanges'
MAP2(object, mdgr, param = MinDistParam(), ...)

Arguments

- **object**: An object of class MinDistExperiment
- **mdgr**: An object of class MinDistGRanges, GRangesList, or GRanges.
- **param**: An object of class MinDistParam.
- **...**: ignored

Value

An object of class MinDistPosterior
Examples

library(oligoClasses)
library(VanillaICE)
## A MinDistExperiment object:
data(md_exp)
## Segmented data
data(md_gr)
e_param <- EmissionParam(temper=1, p_outlier=1/100)
param <- MinDistParam(thin=1L, emission=e_param)
## Not run:
md_g <- MAP2(md_exp, md_gr, param)
## End(Not run)

mdLegend

Text summary of information encapsulated in a MDRanges object for a particular interval

Description

Text summary of information encapsulated in a MDRanges object for a particular interval

Usage

mdLegend(g)

Arguments

g a MDRanges object

MDRanges-class

A GRanges-derived class

Description

Contains maximum a posteriori estimates for each genomic interval

Usage

MDRanges(..., posteriors)

Arguments

... additional arguments to GRanges constructor
posteriors a DataFrame

Examples

MDRanges()
**Description**

This dataset contains log R ratios and B allele frequencies from a parent-offspring trio (three individuals). Only markers from chromosomes 7 and 22 are included in this object. The `MinDistExperiment` class extends `RangedSummarizedExperiment`, and so many of the methods defined for `RangedSummarizedExperiment` such as `findOverlaps` are available through inheritance.

**Format**

a `MinDistExperiment`

**Examples**

```r
## Not run:
library(oligoClasses)
library(VanillaICE)
library(data.table)
library(BSgenome.Hsapiens.UCSC.hg18)

extdir <- system.file("extdata", package="VanillaICE")
features <- suppressWarnings(fread(file.path(extdir, "SNP_info.csv")))
fgr <- GRanges(paste0("chr", features$Chr), IRanges(features$Position, width=1), isSnp=features[["Intensity Only"]]==0)
fgr <- SnpGRanges(fgr)
names(fgr) <- features[["Name"]]
sl <- seqLevels(BSgenome.Hsapiens.UCSC.hg18)
seqlevels(fgr) <- sl[sl %in% seqlevels(fgr)]
seqinfo(fgr) <- seqinfo(BSgenome.Hsapiens.UCSC.hg18)[seqlevels(fgr),]
fgr <- sort(fgr)
files <- list.files(extdir, full.names=TRUE, recursive=TRUE, pattern="FinalReport")

## parse files
parsedDir <- "ParsedFiles"
if(!file.exists(parsedDir)) dir.create(parsedDir)
views <- ArrayViews(rowRanges=fgr, sourcePaths=files, parsedPath=parsedDir)
dat <- fread(files[1])

select_columns <- match(c("SNP Name", "Allele1 - AB", "Allele2 - AB", "Log R Ratio", "B Allele Freq"), names(dat))
index_genome <- match(names(fgr), dat[["SNP Name"]])
scan_params <- CopyNumScanParams(index_genome=index_genome, select=select_columns, cnvar="Log R Ratio", bafvar="B Allele Freq", gtvar=c("Allele1 - AB", "Allele2 - AB"))
invisible(sapply(views, parseSourceFile, param=scan_params))
ped_hapmap <- ParentOffspring(id = "hapmap", father="12287_03", mother="12287_02", offspring="12287_01", parsedPath=parsedPath(views))
ped_list <- ParentOffspringList(pedigrees=list(
  ParentOffspring(id = "hapmap", father="12287_03", mother="12287_02", offspring="12287_01", parsedPath=parsedPath(views)),
```
ParentOffspring(id = "cleft",
    father="22169_03",
    mother="22169_02",
    offspring="22169_01",
    parsedPath=parsedPath(views))))

```r
df <- read.csv(file.path(extdir, "sample_data.csv"), stringsAsFactors=FALSE)
df$id <- setNames(gsub(" ", "", df$IndividualID), df$File)
colnames(views) <- ind_id[seqlevels(views, force=TRUE) <- "chr22"
params <- MinDistParam()
md_gr <- segment2(md_exp, params)
```

## End(Not run)

**md_gr**

An example MinDistGRanges object

**Description**

Prior to inferring de novo trio copy number states, the log R ratios are segmented independently for each individual in a ParentOffsping class. The segmentation results are recorded in separate GRanges objects for the parents. For segmentation of the offspring log R ratios and the minimum distance, the segments are stored in separate GRangesList objects. For convenience, these GRanges, GRangesList, and pedigree information are bound in a single container referred to as a MinDistGRanges object. The example MinDistGRanges object provided in this package was obtained from the segmentation of the data stored in the example MinDistExperiment object.

**Format**

- a MinDistGRanges object

**mindist**

Getter and setter for the minimum distance statistic

**Description**

Getter and setter for the minimum distance statistic

**Usage**

```r
mindist(object)
```

```r
mindist(object) <- value
```

**Arguments**

- object: see showMethods("mindist")
- value: a matrix of the minimum distance
Description

Constructor for MinDistExperiment class

Usage

MinDistExperiment(object = ArrayViews(), pedigree = ParentOffspring(), ...)

## S4 method for signature 'ArrayViews,ParentOffspring'
MinDistExperiment(object = ArrayViews(),
pedigree = ParentOffspring(), ...)

Arguments

object see showMethods(MinDistExperiment)
pedigree a ParentOffspring object
... ignored

Value

an object of class MinDistExperiment

Description

Class and methods for MinDistExperiment

Usage

## S4 method for signature 'MinDistExperiment'
show(object)

## S4 method for signature 'MinDistExperiment'
pedigree(object)

## S4 replacement method for signature 'MinDistExperiment'
pedigree(object) <- value

## S4 method for signature 'MinDistExperiment'
mindist(object)

## S4 replacement method for signature 'MinDistExperiment,ANY'
MinDistGRanges

mindist(object) <- value

## S4 method for signature 'MinDistExperiment,ANY,ANY,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'MinDistExperiment'
offspring(object)

## S4 method for signature 'MinDistExperiment'
father(object)

## S4 method for signature 'MinDistExperiment'
mother(object)

## S4 method for signature 'MinDistExperiment'
segment2(object, param = MinDistParam())

## S4 method for signature 'RangedSummarizedExperiment'
father(object)

## S4 method for signature 'RangedSummarizedExperiment'
mother(object)

## S4 method for signature 'RangedSummarizedExperiment'
offspring(object)

Arguments

object a MinDistExperiment object
value a ParentOffspring object
x a MinDistExperiment object
i a numeric-vector for indexing the rows (optional)
j a numeric-vector for indexing the columns (optional)
... additional arguments propagated to subsetting methods for RangedSummarizedExperiment

drop logical. Whether to simplify a one-row or one-column matrix to a vector. In most cases, this should always be FALSE.

param a MinDistParam object

Slots

mindist a matrix
pedigree a ParentOffspring object

MinDistGRanges Constructor for MinDistGRanges class
**Description**

The MinDistGRanges class contains the segmentation of the father, mother, offspring, and the minimum distance for each possible parent-offspring trio. For the parents, the segmentation results are expected to be in GRanges format. To accommodate multiple-offspring families, both the offspring segments and minimum distance segments should be of class GRangesList where the length of the list corresponds to the number of offspring.

**Usage**

```r
MinDistGRanges(mindist = GRangesList(), offspring = GRangesList(),
father = GRanges(), mother = GRanges(), pedigree = ParentOffspring())
```

**Arguments**

- `mindist`: a GRangesList object
- `offspring`: a GRangesList object
- `father`: a GRanges object
- `mother`: a GRanges object
- `pedigree`: a ParentOffspring object

**Examples**

```r
MinDistGRanges()
```

---

### MinDistGRanges-class

A container for storing segmentation data for members in a ParentOffspring family

---

**Description**

A container for storing segmentation data for members in a ParentOffspring family

**Usage**

```r
## S4 method for signature 'MinDistGRanges'
names(x)

## S4 method for signature 'MinDistGRanges'
mindist(object)

## S4 replacement method for signature 'MinDistGRanges,GRangesList'
mindist(object) <- value

## S4 method for signature 'MinDistGRanges'
offspring(object)

## S4 method for signature 'MinDistGRanges'
mother(object)

## S4 method for signature 'MinDistGRanges'
```
The MinDistParam class contains parameters used for the segmentation implemented in the DNAcopy package, parameters extracted from the PennCNV HMM such as parent-offspring transmission probabilities (see citation below), and initial values / parameters for computing emission probabilities.

Usage

\[
\text{MinDistParam}(nMAD = 0.75, \text{dnacopy} = \text{DNAcopyParam()}, \text{penncnv} = \text{PennParam()}, \\
\text{emission} = \text{EmissionParam()}, \text{thin} = 10L)
\]
Arguments

nMAD a length-one numeric vector indicating the minimal number of median absolute deviations of the mean segmented minimum distance from zero. For non-zero segments (# median absolute deviations > nMAD), maximum a posteriori estimates of the parent-offspring copy number states are computed. Segments with minimum distance values near zero are not called as they are less likely to correspond to regions with de novo copy number alterations.

dnacopy an object of class DNAcopyParam.

penncnv probabilities/parameters of the PennCNV hidden Markov model

emission an object of class EmissionParam

thin a length-one vector indicating whether to thin the data. This is primarily for internal use in conjunction with the filterExperiment function.

See Also

segment

MinDistParam-class  
Class and methods for parameters of minimum distance algorithm

Description

Contains parameters used for circular binary segmentation (package DNAcopy), parameters in the PennCNV hidden Markov model, and parameters used for computing emission probabilities.

Usage

### S4 method for signature 'MinDistParam'

nMAD(object)

### S4 replacement method for signature 'MinDistParam,numeric'

nMAD(object) <- value

### S4 method for signature 'MinDistParam'

show(object)

### S4 method for signature 'DNAcopyParam'

show(object)

Arguments

object a MinDistParam object

value a length-one numeric vector.
Slots

- `nMAD`: a length-one numeric vector
- `dnacopy`: an object of class `DNAcopyParam`
- `penncnv`: an object of class `PennParam`
- `emission`: an object of class `EmissionParam`
- `thin`: a length-one non-negative integer

MinDistPosterior-class

*Container for the segmentation results from a MinDistExperiment*

Description

MinDistPosterior is a GRangesList-derived container for the segmentation and maximum a posteriori trio copy number states.

Usage

```r
## S4 method for signature 'MinDistPosterior,ANY,ANY,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'MinDistPosterior'
show(object)
```

Arguments

- `x`: a `MinDistPosterior` object
- `i`: an index for subsetting rows
- `j`: an index for subsetting columns
- `...`: additional arguments passed to subsetting matrices
- `drop`: logical – whether to coerce single-row matrices to vectors
- `object`: a `MinDistPosterior` object

See Also

- `denovo`

MinimumDistance

*De novo copy number alterations in parent-offspring trios*

Description

De novo copy number alterations in parent-offspring trios
nMAD  
Settter and getter for number of median absolute deviations the mean minimum distance of a genomic interval is from zero

Description
Setter and getter for number of median absolute deviations the mean minimum distance of a genomic interval is from zero

Usage
nMAD(object)

nMAD(object) <- value

Arguments
object  see showMethods("nMAD")
value  a length-one numeric vector

offspringNames  
Deprecated functions and methods

Description
These functions will be defunct in a future release.

Usage
offspringNames(object)

offspringNames(object) <- value
trios(object)
pedigree(object)
pedigree(object) <- value

Arguments
object  see showMethods("offspringNames")
value  a character vector of offspring identifiers
Description

Container for registering sample identifiers with membership in a pedigree. For representing multiple pedigrees, see `ParentOffspringList`.

Accessor for the sample identifiers for the members in a pedigree

Constructor for `ParentOffspring` class

Usage

```r
offspring(object)
mother(object)
father(object)
ParentOffspring(id = character(), father = character(),
    mother = character(), offspring = character(), parsedPath = character())
```

Arguments

- **object** a `ParentOffspring` object
- **id** length-one character vector providing a family-level id
- **father** length-one character vector providing sample ids for father
- **mother** length-one character vector providing sample ids for mother
- **offspring** character vector providing sample ids for offspring (can have length greater than one if there is more than one offspring)
- **parsedPath** character vector providing path to low-level data
- **x** a `ParentOffspring` object
Slots

- **id** length-one character vector providing a family-level id
- **father** length-one character vector providing sample ids for father
- **mother** length-one character vector providing sample ids for mother
- **offspring** character vector providing sample ids for offspring (can have length greater than one if there is more than one offspring)
- **parsedPath** character vector providing path to parsed files of the marker-level summaries

See Also

ParentOffspringList-class

Examples

ParentOffspring()

ParentOffspringList-class

A list of ParentOffspring objects

Description

Each element of the list is an element of class ParentOffspring. Constructor for ParentOffspringList class

Usage

ParentOffspringList(pedigrees = list(), id)

## S4 method for signature 'ParentOffspringList'
pedigreeName(object)

## S4 method for signature 'ParentOffspringList'
show(object)

## S4 method for signature 'ParentOffspringList,ANY,ANY'
x[[i, j, ..., drop = FALSE]]

## S4 method for signature 'ParentOffspringList,ANY,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'ParentOffspringList'
length(x)
**Arguments**

- **pedigrees**: a list of `ParentOffspring` objects
- **id**: identifier for a pedigree
- **object**: a `ParentOffspringList` object
- **x**: a `ParentOffspringList` object
- **i**: a numeric vector for subsetting the list (optional)
- **j**: ignored
- **...**: ignored
- **drop**: ignored

**Slots**

- **id**: a character vector of identifiers for the pedigrees. **id** must have the same length as **pedigrees**
- **pedigrees**: A list of `ParentOffspring` objects.

**Examples**

```r
ParentOffspringList()
```

---

**Description**

This function is deprecated and will be removed in a future release.

**Usage**

```r
Pedigree(pedigreeInfo, fatherIds = character(), motherIds = character(),
          offspringIds = character())
```

**Arguments**

- **pedigreeInfo**: a `data.frame` with column names ‘F’ (father), ‘M’ (mother), and ‘O’ (offspring). Elements of the `data.frame` are the sample names.
- **fatherIds**: character vector of identifiers for the father
- **motherIds**: character vector of identifiers for the mother
- **offspringIds**: character vector of identifiers for the offspring

**Examples**

```r
Pedigree()
```
Pedigree-class

Deprecated class for storing pedigree data

Description

 Deprecated class for storing pedigree data

Usage

## S4 method for signature 'Pedigree'
trios(object)

## S4 method for signature 'Pedigree'
offspringNames(object)

## S4 method for signature 'Pedigree'
show(object)

## S4 method for signature 'Pedigree,ANY,ANY,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'Pedigree'
dim(x)

Arguments

object a Pedigree object
x a Pedigree object
i a numeric vector for subsetting (optional)
j ignored
... ignored
drop ignored

Slots

trios a data.frame with colnames 'F', 'M', and 'O' containing sample identifiers for the father (F), mother (M), and offspring (O).

trioIndex a data.frame

---

pedigreeGrid

Plot the log R ratios and BAFs on a grid given by precomputed viewports

Description

 Plot the log R ratios and BAFs on a grid given by precomputed viewports
pedigreeName

Usage

pedigreeGrid(g, vps, figs)

Arguments

g a MDRanges object
vps a list of viewports. See pedigreeViewports.
figs a list of trellis objects created by the function plotDenovo.

See Also

plotDenovo pedigreeViewports

Examples

library(VanillaICE)
require(grid)
##marker-level summaries
data(md_exp)
seqlevels(md_exp, force=TRUE) <- "chr22"
## segmentation results
data(md_gr)
posteriorCalls <- MAP2(md_exp, md_gr, MinDistParam())
g <- denovoHemizygous(posteriorCalls)
g
vps <- MinimumDistance:::pedigreeViewports()
param <- HmmTrellisParam()
p <- plotDenovo(md_exp, g[1], param)
leg <- mdLegend(g[1])
upViewport(0)
pushViewport(vps["legend"])
grid.text(leg, x=unit(0.02, "npc"), y=unit(0.95, "npc"), just=c("left", "top"),
gp=gpar(cex=0.6, fontfamily="mono"))

## combine adjacent denovo hemizygous
##
g2 <- reduce(denovoHemizygous(posteriorCalls), min.gapwidth=500e3)
post <- MAP2(md_exp, g2)
g2 <- denovoHemizygous(post)
p <- plotDenovo(md_exp, g2, param)
p <- pedigreeGrid(g=g2, vps=vps, figs=p)
leg <- mdLegend(g2)
upViewport(0)
pushViewport(vps["legend"])
grid.text(leg, x=unit(0.02, "npc"), y=unit(0.95, "npc"), just=c("left", "top"),
gp=gpar(cex=0.6, fontfamily="mono"))
Usage

pedigreeName(object)

Arguments

object a ParentOffspring or ParentOffspringList object

See Also

ParentOffspring ParentOffspringList

Description

Default viewports for plotting log R ratios, BAFs, chromosome idiogram, and a legend for a case-parent trio

Usage

pedigreeViewports()

See Also

plotDenovo pedigreeGrid

Examples

vps <- pedigreeViewports()

PennParam

Constructor for class PennParam

Description

Parameters for the PennCNV Hidden Markov model

Usage

PennParam(states, referenceState = "222", prLessLikelyCN = 9e-04, prNonMendelian = 1.5e-06, prInitialStateNotDiploid = 4/5, prTransitionToNewState = 0.5, tauNM = 0.01)

## S4 method for signature 'PennParam'
show(object)
plotDenovo

Arguments

states character vector of possible trio states
referenceState the reference (normal) trio copy number state (typically '222')
prLessLikelyCN as defined in Wang et al. 2007, this is the probability of the less likely allele-specific copy numbers for the trio
prNonMendelian the prior probability of a non-Mendelian copy number alteration
prInitialStateNotDiploid initial probability for non-diploid states
prTransitionToNewState probability of transitioning to a new state
tauNM probability of transitioning from a Mendelian given previous event was non-Mendelian (and vice versa).
object a PennParam object

References

Wang et al., Genome Res. 2007 Nov;17(11):1665-74. PMID: 17921354

Description

Plot marker-level summaries for a genomic interval of interest

Usage

plotDenovo(object, g, param)

## S4 method for signature 'MinDistExperiment,MDRanges'
plotDenovo(object, g, param)

Arguments

object see showMethods("plotDenovo")
g a MDRanges object
param a HmmTrellisParam object
range,ILimit-method  

compute the range of an ILimit instance

Description

The range method for class ILimit is used internally in MinimumDistance.

Usage

```r
## S4 method for signature 'ILimit'
range(x, ..., na.rm = FALSE)
```

Arguments

- `x` a ILimit object
- `...` ignored
- `na.rm` logical. If TRUE, missing values are removed.

segment2  

A wrapper for DNAcopy’s segment function

Description

Methods for circular binary segmentation.

Usage

```r
segment2(object, ...)
```

```r
## S4 method for signature 'TrioSet'
segment2(object, md = NULL, segmentParents = TRUE, verbose = TRUE, ...)
```

```r
## S4 method for signature 'matrix'
segment2(object, pos, chrom, id, featureNames, ...)
```

```r
## S4 method for signature 'ff_matrix'
segment2(object, pos, chrom, id, featureNames, ...)
```

```r
## S4 method for signature 'arrayORff_array'
segment2(object, pos, chrom, id, featureNames, segmentParents = TRUE, verbose = TRUE, ...)
```
Arguments

object 
... 
md 
segmentParents 
verbose 
pos 
chrom 
id 
featureNames

see showMethods(segment2)
Additional arguments passed to DNACopy’s segment.
a matrix of the minimum distance
logical. Whether to segment the log R ratios of the parents using circular binary segmentation.
logical. Whether to display messages that indicate progress.
integer vector of physical position of markers in the genome
character or integer vector of chromosome names
character vector of trio identifiers for subsetting object
character vector specifying marker names for subsetting object

See Also

segment
segment

TrioSet

Deprecated constructor for TrioSet class

Description

The TrioSet class has been deprecated and may be removed in a future release.

Usage

TrioSet(pedigreeData = Pedigree(), sample.sheet, row.names = NULL, lrr, baf,
featureData, cdfname, drop = TRUE, mindist = NULL, genome = c("hg19",
"hg18"))

Arguments

pedigreeData 
sample.sheet 
row.names 
lrr 
baf 
featureData 
cdfname 
drop 
mindist 
genome

an object of class Pedigree
a data.frame containing metadata on the trios
a character vector providing row identifiers for the sample.sheet argument that match the names of the trios in the pedigreeData argument.
a matrix of log R ratios
a matrix of B allele frequencies
a GenomeAnnotatedDataFrame object for the SNPs/nonpolymorphic markers
character string indicating the annotation package used to extract physical position and chromosome of markers
logical. When FALSE, the dimnames on the log R ratio and BAF arrays is set to NULL
can be either NULL or a matrix of the minimum distance
character string providing the UCSC genome build
Value

TrioSet

Examples

```r
path <- system.file("extdata", package="MinimumDistance")
load(file.path(path, "logRratio.rda"))
load(file.path(path, "baf.rda"))
load(file.path(path, "pedigreeInfo.rda"))
trioSet <- TrioSet(lrr=logRratio,
                   baf=baf,
                   pedigree=Pedigree(pedigreeInfo),
                   cdfname="human610quadv1Crlmm",
                   genome="hg18")
```

TrioSet-class

Deprecated class for storing low-level genomic data for trios

Description

This class is deprecated and will be defunct in a future release.

Usage

```r
## S4 method for signature 'TrioSet'
pedigree(object)

## S4 method for signature 'TrioSet'
show(object)

## S4 method for signature 'TrioSet'
mindist(object)

## S4 replacement method for signature 'TrioSet,matrix'
mindist(object) <- value

## S4 method for signature 'TrioSet'
dim(x)

## S4 method for signature 'TrioSet'
trios(object)

## S4 method for signature 'TrioSet,ANY,ANY,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'TrioSet,GRanges'
MAP(object, ranges,
     transition_param = TransitionParam(), emission_param = EmissionParam(),
     mdThr = 0.9, ...)

## S4 method for signature 'ff_array,ANY,ANY,ANY'
x[i, j, ..., drop = FALSE]
```
The `TrioSetList` class has been deprecated and may be removed in a future release. Use `MinDistExperiment` instead.

### Slots

- `fatherPhenoData` AnnotatedDataFrame containing covariates for the father
- `motherPhenoData` AnnotatedDataFrame containing covariates for the mother
- `pedigree` an object of class `Pedigree`
- `mindist` a numeric matrix of the minimum distance for each trio, or NULL

### Description

The `TrioSetList` class has been deprecated and may be removed in a future release. Use `MinDistExperiment` instead.

### Usage

```r
TrioSetList(chromosome = integer(), pedigreeData = Pedigree(), sample.sheet, row.names = NULL, lrr, baf, featureData, cdfname, ffname = "", genome)
```

### Arguments

- `chromosome` integer vector of chromosome names
- `pedigreeData` a `Pedigree` object
- `sample.sheet` a `data.frame` containing sample covariates
- `row.names` a character vector
- `lrr` a matrix of log R ratios
- `baf` a matrix of B allele frequencies
TrioSetList-class

Description

This class is deprecated and will be defunct in a future release.

Usage

## S4 method for signature 'TrioSetList,RangedSummarizedExperiment'
coerce(from, to)

## S4 method for signature 'TrioSetList'
pedigree(object)

## S4 method for signature 'TrioSetList'
trios(object)

## S4 method for signature 'TrioSetList'
offspringNames(object)

## S4 method for signature 'TrioSetList,ANY,ANY,ANY'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'TrioSetList,ANY,ANY'
x[[i, j, ..., exact = TRUE]]

## S4 method for signature 'TrioSetList'
show(object)

## S4 method for signature 'TrioSetList'
length(x)

## S4 method for signature 'TrioSetList'
x$name

## S4 method for signature 'TrioSetList,GRanges'
MAP(object, ranges, id, TAUP = 1e+10,
    tauMAX = 1 - 5e-08, cnStates = c(-2, -0.4, 0, 0, 0.4, 1),
    pr.nonmendelian = 1.5e-06, mdThr = 0.9, ...)

## S4 method for signature 'TrioSetList'
segment2(object, md = NULL, segmentParents = TRUE,
    verbose = TRUE, ...)
## S4 method for signature 'list'
segment2(object, pos, chrom, id = NULL, featureNames,
    segmentParents = TRUE, verbose = TRUE, genome, ...)

### Arguments
- **from**: a TrioSetList
- **to**: a RangedSummarizedExperiment
- **object**: a TrioSetList object
- **x**: a TrioSetList
- **i**: a numeric vector for subsetting the chromosomes (optional)
- **j**: a numeric vector for subsetting trios (optional)
- **...**: additional arguments passed to subsetting methods for matrices and data frames
- **drop**: logical. Whether to simplify matrices to numeric vectors. This should be left as FALSE.
- **exact**: ignored
- **name**: character string of a variable name in the phenoData
- **ranges**: a GRanges object
- **id**: a character vector of trio identifiers
- **TAUP**: length-one numeric vector. Larger values decrease the probability of transitioning to a different state.
- **tauMAX**: the maximum allowed transition probability
- **cnStates**: a length-six numeric vector providing initial values for the mean copy number for each of the 6 states
- **pr.nonmendelian**: a length-one numeric vector indicating the probability of a non-Mendelian copy number alteration in the offspring
- **mdThr**: a length-one numeric vector indicating the minimum value of the mean minimum distance. Segments with absolute mean value less than mdThr are not called.
- **md**: a list of minimum distance matrices. Length of list should be the same as the length of the TrioSetList object.
- **segmentParents**: logical. Whether to segment the parental log R ratios.
- **verbose**: logical. Whether to display messages indicating progress.
- **pos**: a list of the genomic positions (integers)
- **chrom**: list of chromosome names
- **featureNames**: a list of the marker names
- **genome**: a character vector indicating the UCSC genome build used for the annotation (i.e., `‘hg18’` or `‘hg19’`).

### Slots
- **fatherPhenoData**: AnnotatedDataFrame containing covariates for the father
- **motherPhenoData**: AnnotatedDataFrame containing covariates for the mother
- **pedigree**: an object of class Pedigree
TrioSetListLD

Deprecated TrioSetList constructor for large data

Description

The TrioSetListLD constructor uses *ff* objects to handle large datasets. This function is defunct. Use MinDistExperiment instead.

Usage

```r
TrioSetListLD(path, fnames, ext = "", samplesheet, row.names, pedigreeData, featureData, annotationPkg, outdir = ldPath(), ffprefix = "", genome = c("hg19", "hg18"))
```

Arguments

- **path**: Path to plain-text files containing log R ratios and B allele frequencies. Files should contain data for a single sample.
- **fnames**: Character string providing filenames.
- **ext**: Character string indicating whether the fnames has a file extension (e.g., ".txt")
- **samplesheet**: (Optional) `data.frame` containing phenotypic / experimental covariates on the samples. Note that if `samplesheet` is provided, `row.names` must be specified.
- **row.names**: Character vector indicating the sample id for each row in `samplesheet`. `row.names` should be unique and, ideally, correspond to `fnames`
- **pedigreeData**: An object of class Pedigree.
- **featureData**: A `GenomeAnnotatedDataFrame`
- **annotationPkg**: Character string indicating the annotation package used to extract information on the features (chromosome, physical position, and whether the feature is polymorphic ("isSnp")).
- **outdir**: Character string indicating the path for storing *ff* objects. Ignored if the *ff* package is not loaded.
- **ffprefix**: Character string indicating the prefix used to name *ff* objects. Ignored if the *ff* package is not loaded.
- **genome**: character string indicating UCSC genome build. Only "hg19" is allowed for annotation packages that support a single build. Supported builds for most platforms are "hg18" and "hg19".

Value

A TrioSetList object

See Also

- TrioSetList
Index

[,,,MinDistExperiment,ANY,ANY,ANY-method (MinDistExperiment-class), 14
[,,,MinDistPosterior,ANY,ANY,ANY-method (MinDistPosterior-class), 19
[,,ParentOffspringList,ANY,ANY,ANY-method (ParentOffspringList-class), 22
[,,ParentOffspringList,ANY-method (ParentOffspringList-class), 22
[,,Pedigree,ANY,ANY,ANY-method (Pedigree-class), 24
[,,Pedigree,ANY-method (Pedigree-class), 24
[,,TrioSet,ANY,ANY,ANY-method (TrioSet-class), 30
[,,TrioSet,ANY-method (TrioSet-class), 30
[,,TrioSetList,ANY,ANY,ANY-method (TrioSetList-class), 32
[,,TrioSetList,ANY-method (TrioSetList-class), 32
[,,,ff_array,ANY,ANY,ANY-method (TrioSet-class), 30
[,,,ff_array,ANY-method (TrioSet-class), 30
[,,ParentOffspringList,ANY,ANY-method (ParentOffspringList-class), 22
[,,,TrioSetList,ANY,ANY-method (TrioSetList-class), 32
[,,,TrioSetList-method (TrioSetList-class), 32

acf, 3
acf2, 3
as (coerce), 4

calculateMindist, 3
calculateMindist,arrayORff_array-method (calculateMindist), 3
calculateMindist,list-method (calculateMindist), 3
calculateMindist,matrix-method (calculateMindist), 3
calculateMindist,TrioSet-method (calculateMindist), 3
calculateMindist,TrioSetList-method (calculateMindist), 3
calculateMindist,MinDistExperiment-method (calculateMindist), 3
calculateMindist,MinDistPosterior-method (calculateMindist), 3
calculateMindist,ParentOffspring-method (calculateMindist), 3
calculateMindist,Pedigree-method (calculateMindist), 3
calculateMindist,TrioSet-method (calculateMindist), 3
calculateMindist,TrioSetList-method (calculateMindist), 3
calculateMindist,MinDistExperiment-method (coerce), 4
calculateMindist,MinDistPosterior-method (coerce), 4
calculateMindist,ParentOffspring-method (coerce), 4
calculateMindist,Pedigree-method (coerce), 4
calculateMindist,TrioSet-method (coerce), 4
calculateMindist,TrioSetList-method (coerce), 4
calculateMindist,RangedSummarizedExperiment-method (TrioSetList-class), 32
calculateMindist,RangedSummarizedExperiment-method (TrioSet-class), 30
calculateMindist,RangedSummarizedExperiment-method (MinDistExperiment-class), 14
calculateMindist,RangedSummarizedExperiment-method (MinDistPosterior-class), 19

acf, 3
acf2, 3
as (coerce), 4

calculateMindist, 3
calculateMindist,arrayORff_array-method (calculateMindist), 3
calculateMindist,list-method (calculateMindist), 3
calculateMindist,matrix-method (calculateMindist), 3
calculateMindist,TrioSet-method (calculateMindist), 3
calculateMindist,MinDistExperiment-method (calculateMindist), 3
calculateMindist,MinDistPosterior-method (calculateMindist), 3
calculateMindist,ParentOffspring-method (calculateMindist), 3
calculateMindist,Pedigree-method (calculateMindist), 3
calculateMindist,TrioSet-method (calculateMindist), 3
calculateMindist,TrioSetList-method (calculateMindist), 3
calculateMindist,MinDistExperiment-method (coerce), 4
calculateMindist,MinDistPosterior-method (coerce), 4
calculateMindist,ParentOffspring-method (coerce), 4
calculateMindist,Pedigree-method (coerce), 4
calculateMindist,TrioSet-method (coerce), 4
calculateMindist,TrioSetList-method (coerce), 4
calculateMindist,RangedSummarizedExperiment-method (TrioSetList-class), 32
calculateMindist,RangedSummarizedExperiment-method (TrioSet-class), 30
calculateMindist,RangedSummarizedExperiment-method (MinDistExperiment-class), 14
calculateMindist,RangedSummarizedExperiment-method (MinDistPosterior-class), 19

denovo, 19
denovo (denovoHemizygous), 5
denovo,MinDistPosterior-method (denovoHemizygous), 5
denovoDuplication (denovoHemizygous), 5
denovoDuplication,MinDistPosterior-method (denovoHemizygous), 5
denovoHemizygous, 5
denovoHemizygous,MinDistPosterior-method (denovoHemizygous), 5
denovoHemizygous,MinDistPosterior-method (denovoHemizygous), 5
denovoHemizygous,Homozygous,MinDistPosterior-method (denovoHemizygous), 5
denovoHomozygous,MinDistPosterior-method (denovoHemizygous), 5
denovoHomozygous,MinDistPosterior-method (denovoHemizygous), 5
denovoHomozygous,Pedigree-method (Pedigree-class), 24
dim,MinDistExperiment-method (MinDistExperiment-class), 14
dim,MinDistGRanges-method (MinDistGRanges-class), 16
dim,ParentOffspring-method (ParentOffspring-class), 21
dim,RangedSummarizedExperiment-method (MinDistExperiment-class), 14

dim,ParentOffspring-method (ParentOffspring-class), 21

dim,MinDistExperiment-method (MinDistExperiment-class), 14

dim,MinDistGRanges-method (MinDistGRanges-class), 16

dim,ParentOffspring-method (ParentOffspring-class), 21

dim,RangedSummarizedExperiment-method (MinDistExperiment-class), 14

DNAcopyParam, 6
denovo, 19
denovo (denovoHemizygous), 5
denovo,MinDistPosterior-method (denovoHemizygous), 5
denovoDuplication (denovoHemizygous), 5
denovoDuplication,MinDistPosterior-method (denovoHemizygous), 5
denovoHemizygous, 5
denovoHemizygous,MinDistPosterior-method (denovoHemizygous), 5
denovoHemizygous,MinDistPosterior-method (denovoHemizygous), 5
denovoHemizygous,Homozygous,MinDistPosterior-method (denovoHemizygous), 5
denovoHomozygous,MinDistPosterior-method (denovoHemizygous), 5
denovoHomozygous,MinDistPosterior-method (denovoHemizygous), 5
denovoHomozygous,Pedigree-method (Pedigree-class), 24
dim,MinDistExperiment-method (MinDistExperiment-class), 14
dim,MinDistGRanges-method (MinDistGRanges-class), 16
dim,ParentOffspring-method (ParentOffspring-class), 21
filterExperiment, 7
filterExperiment, MinDistExperiment, GRanges-method
(filterExperiment), 7
filterExperiment, MinDistExperiment, GRangesList-method
(filterExperiment), 7
filterExperiment, MinDistExperiment, MinDistGRanges-method
(filterExperiment), 7
FilterParam, 8
FilterParamMD, 7
FilterParamMD-class, 8
length, ParentOffspringList-method
(ParentOffspringList-class), 22
length, TrioSetList-method
(TrioSetList-class), 32
mad, 9
mad2, 9
mad2, array-method (mad2), 9
mad2, list-method (mad2), 9
mad2, matrix-method (mad2), 9
mad2, TrioSetList-method (mad2), 9
MAP, 9
MAP, TrioSet, GRanges-method
(TrioSet-class), 30
MAP, TrioSetList, GRanges-method
(TrioSetList-class), 32
MAP2, 10
MAP2, MinDistExperiment, GRanges-method
(MAP2), 10
MAP2, MinDistExperiment, GRangesList-method
(MAP2), 10
MAP2, MinDistExperiment, MinDistGRanges-method
(MAP2), 10
mad_exp, 12
mad_gr, 13
mdLegend, 11
MDRanges (MDRanges-class), 11
MDRanges-class, 11
mindist, 13
mindist, MinDistExperiment-method
(MinDistExperiment-class), 14
mindist, MinDistGRanges-method
(MinDistGRanges-class), 16
mindist, TrioSet-method (TrioSet-class), 30
mindist<- (mindist), 13
mindist<-, MinDistExperiment, ANY-method
(MinDistExperiment-class), 14
mindist<-, MinDistGRanges, GRangesList-method
(MinDistGRanges-class), 16
mindist<-, TrioSet, matrix-method
(TrioSet-class), 30
MinDistExperiment, 14
MinDistExperiment, ArrayViews, ParentOffspring-method
(MinDistExperiment), 14
MinDistExperiment-class, 14
MinDistGRanges, 15
MinDistGRanges-class, 16
MinDistParam, 17
MinDistParam-class, 18
MinDistPosterior-class, 19
MinimumDistance, 19
MinimumDistance-package
(MinimumDistance), 19
mother (ParentOffspring-class), 21
mother, MinDistExperiment-method
(MinDistExperiment-class), 14
mother, MinDistGRanges-method
(MinDistGRanges-class), 16
mother, ParentOffspring-method
(ParentOffspring-class), 21
mother, RangedSummarizedExperiment-method
(MinDistExperiment-class), 14
names, MinDistGRanges-method
(MinDistGRanges-class), 16
names, ParentOffspring-method
(ParentOffspring-class), 21
nMAD, 20
nMAD, MinDistParam-method
(MinDistParam-class), 18
nMAD<- (nMAD), 20
nMAD<-, MinDistParam, numeric-method
(MinDistParam-class), 18
offspring (ParentOffspring-class), 21
offspring, GRangesList-method
(MinDistGRanges-class), 16
offspring, MinDistExperiment-method
(MinDistExperiment-class), 14
offspring, MinDistGRanges-method
(MinDistGRanges-class), 16
offspring, ParentOffspring-method
(ParentOffspring-class), 21
offspring, RangedSummarizedExperiment-method
(MinDistExperiment-class), 14
offspringNames, 20
offspringNames, Pedigree-method
(Pedigree-class), 24
offspringNames, TrioSetList-method
(TrioSetList-class), 32
offspringNames<- (offspringNames), 20
ParentOffspring, 22, 26
INDEX

ParentOffspring
  (ParentOffspring-class), 21
ParentOffspring-class, 21
ParentOffspringList, 21, 26
ParentOffspringList
  (ParentOffspringList-class), 22
ParentOffspringList-class, 22
Pedigree, 23
pedigree (offspringNames), 20
pedigree, MinDistExperiment-method
  (MinDistExperiment-class), 14
pedigree, MinDistGRanges-method
  (MinDistGRanges-class), 16
pedigree, TrioSet-method
  (TrioSet-class), 30
pedigree, TrioSetList-method
  (TrioSetList-class), 32
Pedigree-class, 24
pedigree <- (offspringNames), 20
pedigree<-, MinDistExperiment-method
  (MinDistExperiment-class), 14
pedigreeGrid, 24, 26
pedigreeName, 25
pedigreeName, ParentOffspring-method
  (ParentOffspring-class), 21
pedigreeName, ParentOffspringList-method
  (ParentOffspringList-class), 22
pedigreeViewports, 25, 26
PennParam, 26
plotDenovo, 25, 26, 27
plotDenovo, MinDistExperiment, MDRanges-method
  (plotDenovo), 27
range, ILimit-method, 28

segment, 6, 18, 29
segment2, 28
segment2, arrayORff_array-method
  (segment2), 28
segment2, ff_matrix-method (segment2), 28
segment2, list-method
  (TrioSetList-class), 32
segment2, matrix-method (segment2), 28
segment2, MinDistExperiment-method
  (MinDistExperiment-class), 14
segment2, TrioSet-method (segment2), 28
segment2, TrioSetList-method
  (TrioSetList-class), 32
show, DNAcopyParam-method
  (MinDistParam-class), 18
show, FilterParamMD-method
  (FilterParamMD-class), 8
show, MinDistExperiment-method
  (MinDistExperiment-class), 14
show, MinDistGRanges-method
  (MinDistGRanges-class), 16
show, MinDistParam-method
  (MinDistParam-class), 18
show, MinDistPosterior-method
  (MinDistPosterior-class), 19
show, ParentOffspring-method
  (ParentOffspring-class), 21
show, ParentOffspringList-method
  (ParentOffspringList-class), 22
show, Pedigree-method (Pedigree-class), 24
show, PennParam-method (PennParam), 26
show, TrioSet-method (TrioSet-class), 30
show, TrioSetList-method
  (TrioSetList-class), 32
stack, TrioSetList-method (coerce), 4
stack, TrioSetList-method (coerce), 4
trios (offspringNames), 20
trios, Pedigree-method (Pedigree-class), 24
trios, TrioSet-method (TrioSet-class), 30
trios, TrioSetList-method
  (TrioSetList-class), 32
TrioSet, 29
TrioSet-class, 30
TrioSetList, 31, 34
trioSetList (exampleTrioSetList), 6
TrioSetList-class, 32
TrioSetListLD, 34