Package ‘MinimumDistance’

March 23, 2024

Title A Package for De Novo CNV Detection in Case-Parent Trios

Version 1.46.0

Date 2017-04-29

Author Robert B Scharpf and Ingo Ruczinski

Description
Analysis of de novo copy number variants in trios from high-dimensional genotyping platforms.

License Artistic-2.0

Depends R (>= 3.5.0), VanillaICE (>= 1.47.1)

Imports methods, BiocGenerics, MatrixGenerics, Biobase, S4Vectors (>= 0.23.18), IRanges, GenomeInfoDb, GenomicRanges (>= 1.17.16), SummarizedExperiment (>= 1.15.4), oligoClasses, DNAcopy, ff, foreach, matrixStats, lattice, data.table, grid, stats, utils

Suggests human610quadv1bCrlmm (>= 1.0.3), BSgenome.Hsapiens.UCSC.hg18, BSgenome.Hsapiens.UCSC.hg19, RUnit

Collate 'help.R' 'AllClasses.R' 'AllGenerics.R' 'coercion-methods.R'

'datasets.R' 'defunct.R' 'functions.R' 'generator-funs.R'
' lattice-methods.R' 'mad-methods.R'
'methods-AnnnotatedDataFrame.R' 'methods-AssayData.R'
'methods-FileViews.R' 'methods-FilterParamMD.R'
'methods-ILimit.R' 'methods-MDRanges.R'
'methods-MinDistExperiment.R' 'methods-MinDistGRanges.R'
'methods-MinDistParam.R' 'methods-MinDistPosterior.R'
'methods-ParentOffspring.R' 'methods-ParentOffspringList.R'
'methods-Pedigree.R' 'methods-PennParam.R'
'methods-ShallowSimpleList.R' 'methods-SummarizedExperiment.R'
'methods-TrioSet.R' 'methods-TrioSetList.R'
'methods-ff_array.R' 'methods-grid.R' 'methods-list.R'
'methods-matrix.R' 'segment2-methods.R' 'utils.R' 'zzz.R'

Enhances snow, doSNOW

LazyLoad yes

biocViews Microarray, SNP, CopyNumberVariation

RoxygenNote 7.1.1
Encoding UTF-8

**git_url** https://git.bioconductor.org/packages/MinimumDistance

**git_branch** RELEASE_3_18

**git_last_commit** b1328d8

**git_last_commit_date** 2023-10-24

**Repository** Bioconductor 3.18

**Date/Publication** 2024-03-22

**Maintainer** Robert Scharpf <rscharpf@jhu.edu>

---

**R topics documented:**

- acf2 .................................................. 3
- calculateMindist .................................... 4
- coerce ............................................. 5
- Defunct ............................................ 5
- denovoHemizygous .................................. 6
- DNAdcopyParam .................................... 7
- exampleTrioSetList .............................. 7
- filterExperiment .................................. 8
- FilterParamMD .................................... 8
- FilterParamMD-class ............................. 9
- mad2 .............................................. 10
- MAP ................................................ 10
- MAP2 ............................................. 11
- mdLegend .......................................... 12
- MDRanges-class ................................... 13
- md_exp .......................................... 13
- md_gr ............................................ 14
- mindist .......................................... 15
- MinDistExperiment ............................... 15
- MinDistExperiment-class ....................... 16
- MinDistGRanges ................................ 17
- MinDistGRanges-class ........................... 18
- MinDistParam .................................... 19
- MinDistParam-class ............................. 20
- MinDistPosterior-class ......................... 21
- MinimumDistance ................................ 22
- nMAD ............................................. 22
- offspringNames .................................. 22
- ParentOffspring .................................. 23
- ParentOffspring-class .......................... 24
- Pedigree .......................................... 25
- Pedigree-class .................................. 26
- pedigreeGrid .................................... 27
- pedigreeName .................................... 28
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
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 |O-F| denote the absolute difference in the log R ratios comparing offspring to father and |O-M| the absolute difference in the log R ratios comparing offspring to mother. The minimum distance at a marker is the signed minimum of |O-M| and |O-F|.

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

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

Examples

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

Description

Coercion methods in MinimumDistance package

Coerces a TrioSetList to a TrioSet

Usage

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

---

Defunct  

Defunct functions/classes/methods in the MinimumDistance package

Description

The function, class, or data object you asked is defunct.
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"))
denovo(object, filters = FilterParamMD(state = c("220", "221", "224")))
denovoDuplication(object, filters = FilterParamMD(state = "224"))
```

## S4 method for signature 'MinDistPosterior'

denovoHemizygous(object, filters = FilterParamMD(state = "221"))

## S4 method for signature 'MinDistPosterior'

denovoHomozygous(object, filters = FilterParamMD(state = "220"))

## S4 method for signature 'MinDistPosterior'

denovoDuplication(object, filters = FilterParamMD(state = "224"))

## S4 method for signature 'MinDistPosterior'

denovo(object, filters = FilterParamMD(state = c("220", "221", "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)

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

Arguments

state  trio copy number states to select
seqnames  chromosome names to select
...  additional arguments passed to FilterParam-class

Examples

library(VanillaICE)
data(md_gr)
data(md_exp)
mdparam <- MinDistParam()
fit <- MAP2(md_exp, md_gr, mdparam)
## return all segments
segs(fit)

## Default filters
param <- FilterParamMD()
param
cnvFilter(fit, param)

param2 <- FilterParamMD(seqnames="chr22", probability=0.9, numberFeatures=10)
cnvFilter(fit, param2)
denovoHemizygous(fit)

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'
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

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

- **byrow**
  - logical if TRUE, compute the median absolute deviation of the rows of a matrix

#### 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.

MAP2 Computes maximum a posteriori estimate for the trio copy number state

Description

Computes maximum a posteriori estimate for the trio copy number state

Usage

MAP2(object, mdgr, param = MinDistParam(), ...)

## 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(), ...)

mdLegend

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)

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**

```r
MDRanges(..., posteriors)
```

**Arguments**

```r
... additional arguments to GRanges constructor
posteriors a DataFrame
```

**Examples**

```r
MDRanges()
```

---

**md_exp**

An example **MinDistExperiment**

**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)
```
```r
names(fgr) <- features[["Name"]]
s1 <- seqlevels(BSgenome.Hsapiens.UCSC.hg18)
seqlevels(fgr) <- s1[s1 %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[])
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)))))
sample_info <- read.csv(file.path(extdir, "sample_data.csv"), stringsAsFactors=FALSE)
ind_id <- setNames(gsub(" ", ",", sample_info$IndividualID, sample_info$File)
colnames(views) <- ind_id[gsub(".", ",", colnames(views))]
md_exp <- MinDistExperiment(views, pedigree=ped_list[[2]])
seqlevels(md_exp, pruning.mode="coarse") <- "chr22"
params <- MinDistParam()
md_gr <- segment2(md_exp, params)
save(md_exp, file="~/Software/bridge/MinimumDistance/data/md_exp.rda")
save(md_gr, file="~/Software/bridge/MinimumDistance/data/md_gr.rda")
```

---

**md_gr**

An example MinDistGRanges object
Prior to inferring de novo trio copy number states, the log R ratios are segmented independently for each individual in a ParentOffspring 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

**Description**

Getter and setter for the minimum distance statistic

**Usage**

```r
mindist(object)

mindist(object) <- value
```

**Arguments**

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

### MinDistExperiment

**Description**

Constructor for MinDistExperiment class

**Usage**

```r
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`

---

**MinDistExperiment-class**

*Class and methods for MinDistExperiment*

---

**Description**

Class and methods for MinDistExperiment

**Usage**

```r
## 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'
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

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

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

## 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)
MinDistParam

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

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

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

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

### Arguments

- `x` a `MinDistGRanges` object
- `object` a `MinDistGRanges` object
- `value` a `GRangesList` object

### Slots

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

### Examples

```r
data(md_gr)
offspring(md_gr)
father(md_gr)
mother(md_gr)
mindist(md_gr)
```

---

**MinDistParam**  
*Constructor for MinDistParam class*

### Description

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

MinDistParam(
  nMAD = 0.75,
  dnacopy = DNAcopyParam(),
  penncnv = PennParam(),
  emission = EmissionParam(),
  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 esti-
mates 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 cor-
respond 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)
MinDistPosterior-class

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

## 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

nMAD | Setter 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
ParentOffspring-class

Object containing the sample identifiers for members in a pedigree

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

Usage

offspring(object)
mother(object)
father(object)

ParentOffspring(
  id = character(),
  father = character(),
  mother = character(),
  offspring = character(),
  parsedPath = character()
)

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

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

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

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

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

## S4 method for signature 'ParentOffspring'
names(x)

Arguments

object a ParentOffspring object
ParentOffspringList-class

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 low-level data

`x` a ParentOffspring object

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`.

Usage

ParentOffspringList(pedigrees = list(), id)

```r
## 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,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()
```

---

**Pedigree**

*Deprecated function for constructing an instance of class Pedigree*

#### 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

```r
## 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

**Usage**

```r
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**

```r
library(VanillaICE)
require(grid)
## marker-level summaries
data(md_exp)
seqlevels(md_exp, pruning.mode="coarse") <- "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)
p <- pedigreeGrid(g=g[1], vps=vps, figs=p)
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"))
```
## pedigreeViewports

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

### pedigreeName

**Accessor for pedigree name**

**Description**

Accessor for pedigree name

**Usage**

```r
pedigreeName(object)
```

**Arguments**

- `object` a `ParentOffspring` or `ParentOffspringList` object

**See Also**

- `ParentOffspring`
- `ParentOffspringList`

### pedigreeViewports

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

**Description**

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

**Usage**

```r
pedigreeViewports()
```
PennParam

**See Also**

plotDenovo pedigreeGrid

**Examples**

```r
vps <- pedigreeViewports()
```

---

**PennParam**

*Constructor for class PennParam*

**Description**

Parameters for the PennCNV Hidden Markov model

**Usage**

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

```r
## S4 method for signature 'PennParam'
show(object)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>states</td>
<td>character vector of possible trio states</td>
</tr>
<tr>
<td>referenceState</td>
<td>the reference (normal) trio copy number state (typically '222')</td>
</tr>
</tbody>
</table>
| 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
plotDenovo

*Plot marker-level summaries for a genomic interval of interest*

**Description**

Plot marker-level summaries for a genomic interval of interest

**Usage**

```r
plotDenovo(object, g, param)
```

```r
## 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

segment2(object, ...)

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

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

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

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

Arguments

object

see showMethods{segment2}

... Additional arguments passed to DNAcopy's segment.

md a matrix of the minimum distance

segmentParents logical. Whether to segment the log R ratios of the parents using circular binary segmentation.

verbose logical. Whether to display messages that indicate progress.

pos integer vector of physical position of markers in the genome

chrom character or integer vector of chromosome names

id character vector of trio identifiers for subsetting object

featureNames character vector specifying marker names for subsetting object
TrioSet

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

Value
TrioSet
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]
```

Arguments

- `object` - a TrioSet object
- `value` - a matrix
TrioSetList

Description

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

Usage

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

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
- featureData: a GenomeAnnotatedDataFrame
- cdfname: a character string indicating the annotation package
- fname: prefix for ff-filenames
- genome: character string indicating genome build

TrioSetList-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 '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'
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)
- **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
TrioSetListLD

ranges a GRanges object
id a character vector of trio identifiers
TAUP a 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

TrioSetListLD(
  path,
  fnames,
  ext = "",
  samplesheet,
  row.names,
  pedigreeData,
  featureData,
  annotationPkg,
  outdir = ldPath(),
  fprefix = "",
  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

* internal

  Defunct, 5

  [, MinDistExperiment, ANY, ANY, ANY-method
    (MinDistExperiment-class), 16

  [, MinDistPosterior, ANY, ANY, ANY-method
    (MinDistPosterior-class), 21

  [, ParentOffspringList, ANY, ANY, ANY-method
    (ParentOffspringList-class), 24

  [, ParentOffspringList, ANY-method
    (ParentOffspringList-class), 24

  [, Pedigree, ANY, ANY, ANY-method
    (Pedigree-class), 26

  [, Pedigree, ANY-method
    (Pedigree-class), 26

  [, TrioSet, ANY, ANY, ANY-method
    (TrioSet-class), 33

  [, TrioSet, ANY-method
    (TrioSet-class), 33

  [, TrioSetList, ANY, ANY, ANY-method
    (TrioSetList-class), 35

  [, TrioSetList, ANY-method
    (TrioSetList-class), 35

  [, ff_array, ANY, ANY, ANY-method
    (TrioSet-class), 33

  [, ff_array, ANY-method
    (TrioSet-class), 33

  [, ParentOffspringList, ANY, ANY-method
    (ParentOffspringList-class), 24

  [, TrioSetList, ANY, ANY-method
    (TrioSetList-class), 35

  $, TrioSetList-method
    (TrioSetList-class), 35

  acf, 3

  acf2, 3

  calculateMindist, 4

  calculateMindist, arrayORff_array-method
    (calculateMindist), 4

  calculateMindist, list-method
    (calculateMindist), 4

  calculateMindist, matrix-method
    (calculateMindist), 4

  calculateMindist, TrioSet-method
    (calculateMindist), 4

  calculateMindist, TrioSetList-method
    (calculateMindist), 4

  coerce, 5

  coerce, Pedigree, ParentOffspring-method
    (coerce), 5

  coerce, RangedDataCNV, GRanges-class
    (Defunct), 5

  coerce, TrioSet, data.frame-method
    (coerce), 5

  coerce, TrioSet, MinDistExperiment-method
    (coerce), 5

  coerce, TrioSet, TrioSetList-method
    (coerce), 5

  coerce, TrioSetList, MinDistExperiment-method
    (coerce), 5

  coerce, TrioSetList, RangedSummarizedExperiment-method
    (TrioSetList-class), 35

  Defunct, 5

  denovo, 21

  denovo (denovoHemizygous), 6

  denovo, MinDistPosterior-method
    (denovoHemizygous), 6

  denovoDuplication (denovoHemizygous), 6

  denovoDuplication, MinDistPosterior-method
    (denovoHemizygous), 6

  denovoHemizygous, 6

  denovoHemizygous, MinDistPosterior-method
    (denovoHemizygous), 6

  denovoHomozygous (denovoHemizygous), 6

  denovoHomozygous, MinDistPosterior-method
    (denovoHemizygous), 6

  dim, Pedigree-method (Pedigree-class), 26

  dim, TrioSet-method (TrioSet-class), 33

  DNAcopyParam, 7
exampleTrioSetList, 7
father (ParentOffspring-class), 23
dist, 15
mindist, MinDistExperiment-method
(MinDistExperiment-class), 16
mindist, MinDistGRanges-method
(MinDistGRanges-class), 18
mindist, TrioSet-method (TrioSet-class), 33
mindist<-(mindist), 15
mindist<-, MinDistExperiment, ANY-method
(MinDistExperiment-class), 16
mindist<-, MinDistGRanges, GRangesList-method
(MinDistGRanges-class), 18
mindist<-, TrioSet, matrix-method
(TrioSet-class), 33
MAP, 10
MinDistExperiment, ArrayViews, ParentOffspring-method
(MinDistExperiment-class), 15
MinDistExperiment-class, 16
MinDistGRanges, 17
MinDistGRanges-class, 18
MinDistParam, 19
MinDistParam-class, 20
MinDistPosterior-class, 21
MinimumDistance, 22
mother (ParentOffspring-class), 23
MAD, 10
MDRange (MDRange-class), 13
MDRange-class, 13
nMAD, 22
nMAD, MinDistParam-method
(MinDistParam-class), 20
nMAD<-(nMAD), 22
nMAD<-, MinDistParam, numeric-method
(MinDistParam-class), 20
offspring (ParentOffspring-class), 23
offspring, MinDistExperiment-method (MinDistExperiment-class), 16
offspring, MinDistGRanges-method (MinDistGRanges-class), 18
offspring, ParentOffspring-method (ParentOffspring-class), 23
offspring, RangedSummarizedExperiment-method (MinDistExperiment-class), 16
offspringNames, 22
offspringNames, Pedigree-method (Pedigree-class), 26
offspringNames, TrioSetList-method (TrioSetList-class), 35
offspringNames<- (offspringNames), 22
ParentOffspring, 24, 28
ParentOffspring (ParentOffspring-class), 23
ParentOffspring-class, 23
ParentOffspringList, 22, 28
ParentOffspringList (ParentOffspringList-class), 24
ParentOffspringList-class, 24
Pedigree, 25
pedigree (offspringNames), 22
pedigree, MinDistExperiment-method (MinDistExperiment-class), 16
pedigree, MinDistGRanges-method (MinDistGRanges-class), 18
pedigree, TrioSet-method (TrioSet-class), 33
pedigree, TrioSetList-method (TrioSetList-class), 35
Pedigree-class, 26
pedigree<- (offspringNames), 22
pedigree<-, MinDistExperiment-method (MinDistExperiment-class), 16
pedigreeGrid, 27, 29
pedigreeName, 28
pedigreeName, ParentOffspring-method (ParentOffspring-class), 23
pedigreeName, ParentOffspringList-method (ParentOffspringList-class), 24
pedigreeViewports, 27, 28
PennParam, 29
plotDenovo, 27, 29, 30
plotDenovo, MinDistExperiment, MDRanges-method (plotDenovo), 30
range, ILimit-method, 30
segment, 7, 20, 32
segment2, 31
segment2, arrayORff_array-method (segment2), 31
segment2, ff_matrix-method (segment2), 31
segment2, list-method (TrioSetList-class), 35
segment2, matrix-method (segment2), 31
segment2, MinDistExperiment-method (MinDistExperiment-class), 16
segment2, TrioSet-method (segment2), 31
segment2, TrioSetList-method (TrioSetList-class), 35
show, DNACopyParam-method (MinDistParam-class), 20
show, FilterParamMD-method (FilterParamMD-class), 9
show, MinDistExperiment-method (MinDistExperiment-class), 16
show, MinDistGRanges-method (MinDistGRanges-class), 18
show, MinDistParam-method (MinDistParam-class), 20
show, MinDistPosterior-method (MinDistPosterior-class), 21
show, ParentOffspring-method (ParentOffspring-class), 23
show, ParentOffspringList-method (ParentOffspringList-class), 24
show, Pedigree-method (Pedigree-class), 26
show, PennParam-method (PennParam), 29
show, TrioSet-method (TrioSet-class), 33
show, TrioSetList-method (TrioSetList-class), 35
stack, TrioSetLiset-method (coerce), 5
stack, TrioSetList-method (coerce), 5
trios (offspringNames), 22
trios, Pedigree-method (Pedigree-class), 26
trios, TrioSet-method (TrioSet-class), 33
trios, TrioSetList-method (TrioSetList-class), 35
TrioSet, 32
TrioSet-class, 33
TrioSetList, 34, 38
trioSetList (exampleTrioSetList), 7
TrioSetList-class, 35
TrioSetListLD, 37