Package ‘MareyMap’

October 12, 2022

Title Estimation of Meiotic Recombination Rates Using Marey Maps

Version 1.3.6

Depends R (>= 2.10)

Imports graphics, grDevices, methods, stats, tcltk, tools, utils

Suggests tkrplot

Author Aurélie Siberchicot, Clément Rezvoy, Delphine Charif, Laurent Gueguen and Gabriel Marais

Maintainer Aurélie Siberchicot <aurelie.siberchicot@univ-lyon1.fr>

License GPL (>= 2)

Description Local recombination rates are graphically estimated across a genome using Marey maps.

BugReports https://github.com/aursiber/MareyMap/issues

URL https://lbbe.univ-lyon1.fr/-MareyMap-.html ;
http://lbbe-shiny.univ-lyon1.fr/MareyMapOnline/

LazyData no

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2020-12-04 10:40:05 UTC

R topics documented:

`Arabidopsis_thaliana` *The marey map for the species Arabidopsis thaliana*

Description

A marey map is an interpolation between genetical and physical maps of a mapomosome. This data file contains marey map for the species Arabidopsis thaliana. This map is held into an object of class MapSet.

Usage

```
data(Arabidopsis_thaliana)
```

Format

Arabidopsis_thaliana contains an object of the class `MapSet-class`. The maps inside this object (objects of the class `MareyMap-class`) can be accessed using the operator "[[" (see example) This set contains 5 maps called "Chromosome 1", ..., "Chromosome 5".

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

Source

The physical positions were also recorded for those genetic markers that have both been mapped to the Recombinant Inbred (RI) recombination map (see `http://arabidopsis.info/`), and have been precisely physically mapped on the basis of flanking sequence, using the marker position information from TAIR. (release 1). Wright SI, Agrawal N, Bureau TE. Effects of recombination rate and gene density on transposable element distributions in Arabidopsis thaliana.Genome Res. 2003 Aug;13(8):1897-903.

Examples

```
#data(Arabidopsis_thaliana)
#map <- Arabidopsis_thaliana[["Chromosome 1"]]
#plot(map)
#names(Arabidopsis_thaliana@maps)
```

`argList-methods` *Methods for function argList in package 'MareyMap'*

Description

Methods for function `argList` in package **MareyMap** returns the argument list for object

Methods

object = "Interpolation" returns the basic interpolation parameters (for use in createOrder)

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`Interpolation-class`

`Caenorhabditis_elegans` *The marey map for the species Caenorhabditis_elegans*

Description

A marey map is an interpolation between genetical and physical maps of a mapomosome. This data file contains marey map for the species Caenorhabditis_elegans. This map is held into an object of class MapSet.

Usage

```
data(Caenorhabditis_elegans)
```

Format

Caenorhabditis_elegans contains an object of the class `MapSet-class`. The maps inside this object (objects of the class `MareyMap-class`) can be accessed using the operator "[[" (see example) This set contains 6 maps called "I", "II", "III", "IV", "V" and "X".

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

Source

Wormbase Release WS160, `https://wormbase.org//`

Examples

```
data(Caenorhabditis_elegans)
map <- Caenorhabditis_elegans[["I"]]
plot(map)
names(Caenorhabditis_elegans@maps)
```

`color-methods` *Accessors for slot color in package 'MareyMap'*

Description

Methods for function `color` in package **MareyMap**

Methods

object = "Interpolation" Returns or sets the value of the slot color in the `interpolation` object

Author(s)

Aurélie Siberchicot `<aurelie.siberchicot@univ-lyon1.fr>` and Clément Rezvoy

See Also

`Interpolation-class`

`createOrder-methods` *createOrder*

Description

Returns a character containing the code to recreate the object as it is at the moment of the call (e.g. "new(objCLass, arg1=...)")

Methods

object = "MMSlidingWindow" createOrder for method MMSlidingWindow

object = "MMLoess" createOrder for method MMLoess

object = "MMSpline3" createOrder for method MMSpline3

Author(s)

Aurélie Siberchicot `<aurelie.siberchicot@univ-lyon1.fr>` and Clément Rezvoy

See Also

`Interpolation-class`

`Drosophila_melanogaster` *The marey map for the species Drosophila melanogaster*

Description

A marey map is an interpolation between genetical and physical maps of a mapomosome. This data file contains marey map for the species Drosophila melanogaster. This map is held into an object of class MapSet.

Usage

```
data(Drosophila_melanogaster)
```

Format

Drosophila_melanogaster contains an object of the class `MapSet-class`. The maps inside this object (objects of the class `MareyMap-class`) can be accessed using the operator "[[" (see example) This set contains 6 maps called "2L", "2R", "3L", "3R", "4", "X".

Author(s)

Aurélie Siberchicot `<aurelie.siberchicot@univ-lyon1.fr>` and Clément Rezvoy

Source

2nd release of D. mel genome, v 01/1998 Flybase. Marais G, Piganeau G.Hill-Robertson interference is a minor determinant of variations in codon bias across Drosophila melanogaster and Caenorhabditis elegans genomes. Mol Biol Evol. 2002 Sep;19(9):1399-406.

Examples

```
data(Drosophila_melanogaster)
map <- Drosophila_melanogaster[["2L"]]
plot(map)
names(Drosophila_melanogaster@maps)
```

`geneticDistances-methods` *Accessors for slot geneticDistances*

Description

Methods for function `geneticDistances` in package **MareyMap**

Methods

object = "MareyMap" returns or set the value of the slot geneticDistances in MareyMap

Author(s)

Aurélie Siberchicot `<aurelie.siberchicot@univ-lyon1.fr>` and Clément Rezvoy

See Also

`MareyMap-class`

`getInterList-methods` *returns a list of interpolation methods.*

Description

returns the list of the interpolation methods present in the interpolation register.

Methods

dummy = "missing" default method.

Author(s)

Aurélie Siberchicot `<aurelie.siberchicot@univ-lyon1.fr>` and Clément Rezvoy

`Homo_sapiens` *The marey map for the species Homo sapiens (female, male and mean of the two)*

Description

A marey map is an interpolation between genetical and physical maps of a mapomosome. This data file contains marey map for the species Homo sapiens. This map is held into an object of class MapSet.

Usage

```
data(Homo_sapiens_mean)
```

Format

Homo sapiens (female, male and mean) contains an object of the class `MapSet-class`. The maps inside this object (objects of the class `MareyMap-class`) can be accessed using the operator "[[" (see example) Each of the 3 sets (female, male and mean) contains 22 maps called "Chromosome 01", ... , "Chromosome 22".

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

Source

Rutgers Combined Linkage-Physical Maps, version 2.0 (Build 35). Xiangyang Komg and Tara Matise 12/08/2004

Examples

```
data(Homo_sapiens_female)
map <- Homo_sapiens_female[["Chromosome 01"]]
plot(map)
names(Homo_sapiens_female@maps)
```

`interpolate-methods` *interpolate*

Description

performs an interpolation on `map` using the interpolation method provided by `object`

Methods

object = "Interpolation", map = "MareyMap" DO NOT USE in an 'real' programming language, this would be a virtual function

object = "MMSlidingWindow", map = "MareyMap" interpolate using sliding windows

object = "MMLoess", map = "MareyMap" interpolate using loess

object = "MMSpline3", map = "MareyMap" interpolate using cubic splines.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`Interpolation-class`

`Interpolation-class` *Class "Interpolation"*

Description

This class serves as a base for interpolation methods. Although it is not defined as virtual for technical reason, it should not be used as is and is only useful after having been extended.

Objects from the Class

Objects from the class should not be created

Slots

`name`**:** Object of class `"character"` holding the name of the interpolation

`color`**:** Object of class `"character"` holding the name or the code (\#RRGGBB) of the color of the interpolation on the plot

`physicalPositions`**:** Object of class `"vector"` holding the valid physical position of the map on which the interpolation is calculated. This redundancy is necessary to cary out the plotting of the map.

`rates`**:** Object of class `"vector"` holding the local recombination rate estimations at the location of each markers.

`visible`**:** Object of class `"logical"`, whether or not the interpolation should be plotted

`persistent`**:** Object of class `"logical"`, whether or not the interpolation should be saved when the map is saved to text file.

Methods

argList `signature(object = "Interpolation")`: This function is used in the function createList (c.f. code examples).

color<- `signature(object = "Interpolation")`: changes the color of the interpolation.

color `signature(object = "Interpolation")`: returns the color of the interpolation.

interpolate `signature(object = "Interpolation", map = "MareyMap")`: recalculate the interpolation.

name<- `signature(object = "Interpolation")`: changes the name of the interpolation.

name `signature(object = "Interpolation")`: returns the name of the interpolation.

persistent<- `signature(object = "Interpolation")`: changes the persistence state of the interpolation.

persistent `signature(object = "Interpolation")`: returns the persistence state of the interpolation.

plotModel `signature(object = "Interpolation")`: this function plots only the model of the marey map calculated by the method.

- **plotRate** `signature(object = "Interpolation")`: this function plots the recombination rates calculated by the method.
- **query** `signature(object = "Interpolation", pos = "integer")`: returns the recombination rate calculated by this method on a given location on the sequence.
- **rates<-** `signature(object = "Interpolation")`: replaces the rates with a new vector.
- **rates** `signature(object = "Interpolation")`: returns the rates of the interpolation.
- **userParam** `signature(object = "Interpolation")`: function returning information about the parameters modifiable by the user (c.f. code example.)
- **visible<-** `signature(object = "Interpolation")`: changes the visibility of the interpolation.
- **visible** `signature(object = "Interpolation")`: returns the visibility of the interpolation

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`InterpolationParam-class` `MareyMap-class`

`interpolation-methods` *set or returns an interpolation.*

Description

Methods for function `interpolation` in package **MareyMap**

Methods

- **object = "MareyMap", inter_name = "character"** set or returns the interpolation `inter\_name` in object `MareyMap-class`

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`Interpolation-class`

`InterpolationParam-class`

Class "InterpolationParam"

Description

Class holding informations about a parameter of an interpolation method. This class is used in the function `userParam,Interpolation-method`. These informations are used by the GUI to dynamically create the dialog to modify this parameter.

Objects from the Class

Objects can be created by calls of the form `new("InterpolationParam", ...)` or using the method`InterpolationParam()`

Slots

`paramName`**:** Object of class `"character"` holding the name of the parameter.

`paramType`**:** Object of class `"character"` holding the type of the parameter, currently supported parameter types are: `logical`, `numeric`, `integer` and `color`.

`paramDesc`**:** Object of class `"character"` giving a small description of the parameter's signification.

`paramDefault`**:** Object of class `"ANY"`, default value for the parameter.

`paramValues`**:** Object of class `"vector"` containing the set of valid values for the parameter (can be NULL).

`paramMin`**:** Object of class `"ANY"`, a minimum value for the parameter, or minimal string length if the parameter is of type `character`

`paramMax`**:** Object of class `"ANY"` , a minimum value for the parameter, or maximum string length if the parameter is of type `character`

`paramFun`**:** Object of class `"character"`, the nae of function that is to be called to update the parameter's value. ex if paramFun is `foo` the parameter's value will be set using `foo(interpolation)<-new_value`

Methods

paramDefault<- `signature(object = "InterpolationParam")`: ...

paramDefault `signature(object = "InterpolationParam")`: ...

paramDesc<- `signature(object = "InterpolationParam", value = "character")`: ...

paramDesc `signature(object = "InterpolationParam")`: ...

paramFun<- `signature(object = "InterpolationParam", value = "character")`: ...

paramFun `signature(object = "InterpolationParam")`: ...

paramMax<- `signature(object = "InterpolationParam")`: ...

paramMax `signature(object = "InterpolationParam")`: ...

paramMin<- `signature(object = "InterpolationParam")`: ...

paramMin `signature(object = "InterpolationParam")`: ...

paramName<- `signature(object = "InterpolationParam", value = "character")`: ...

paramName `signature(object = "InterpolationParam")`: ...

paramType<- `signature(object = "InterpolationParam", value = "character")`: ...

paramType `signature(object = "InterpolationParam")`: ...

paramValues<- `signature(object = "InterpolationParam", value = "vector")`: ...

paramValues `signature(object = "InterpolationParam")`: ...

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`Interpolation-class`

Examples

```
  nam <- InterpolationParam()
paramName(nam) <- "name"
paramDesc(nam) <- "The name of the interpolation.\n"
paramType(nam) <- "character"
paramDefault(nam) <- "default name"
paramMin(nam) <- 1
paramMax(nam) <- NULL
paramFun(nam) <- "name"

vis <- InterpolationParam()
paramName(vis) <- "visible"
paramDesc(vis) <- "Whether the line is visible on the plot or not"
paramType(vis) <- "logical"
paramDefault(vis) <- TRUE
paramFun(vis) <- "visible"

sav <- InterpolationParam()
paramName(sav) <- "persistent"
paramDesc(sav) <- "Indicate if the interpolation is to be be kept \n\
  when the map is saved to text file"
paramType(sav) <- "logical"
paramDefault(sav) <- TRUE
paramFun(sav) <- "persistent"

col <- InterpolationParam()
paramName(col) <- "line color"
paramDesc(col) <- "color of the line"
paramType(col) <- "color"
paramDefault(col) <- "#000000"
paramFun(col) <- "color"
```

`InterpolationParam-methods` *Methods for Function InterpolationParam in Package 'MareyMap'*

Description

constructor for class `InterpolationParam-class` in package **MareyMap**

Methods

dummy = "missing" default constructor.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`link{InterpolationParam-class}`

`interpolations-methods` *accessors for slot interpolations*

Description

set or get the value of the slot `interpolation` in `object`

Methods

object = "MareyMap" set or get slot `interpolations` in `MareyMap-class`

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`MareyMap`

`MapCollection-class` *Class "MapCollection"*

Description

This class holds several `MareyMap-class` objects organised in sets (using `MapSet-class` object).

Objects from the Class

Objects can be created by calls of the form `new("MapCollection", ...)` or using the function MapCollection. You may as well use the default collection provided by the package (as `data()`).

Slots

`sets`: Object of class `"list"` containing a `MapSet-class` for each set$* of the collection.

Methods

[`signature(x = "MapCollection")`: ...

[[<- `signature(x = "MapCollection", i = "ANY", j = "ANY", value = "MapSet")`: ...

[[<- `signature(x = "MapCollection", i = "ANY", j = "ANY", value = "MareyMap")`: ...

[[`signature(x = "MapCollection")`: obtain the MapSet corresponding to a set name

$ `signature(x = "MapCollection")`: subset by set name

+ `signature(e1 = "MapCollection", e2 = "MareyMap")`: adds a map to the collection

+ `signature(e1 = "MapCollection", e2 = "MapSet")`: adds a set t the collection

- `signature(e1 = "MapCollection", e2 = "character")`: removes the map named e2 from the object e1.

coerce `signature(from = "MapCollection", to = "data.frame")`: --> `as(object,"data.frame")`

setNames `signature(object = "MapCollection")`: returns the list of the names of all the sets present in the collection

length `signature(object = "MapCollection")`: returns the length of the collection

textFile `signature(object = "MapCollection", file = "character")`: Writes the content of the collection to text file `text`.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`MapSet-class` `MareyMap-class`

Examples

```
data(Homo_sapiens_mean)
map1 <- Homo_sapiens_mean[["Chromosome 01"]]
# do some useful changes to the map...
mapName(map1)<- "Chrom 1 (modified)"
```

`MapCollection-methods` *Constructor for class MapCollection*

Description

Creates an object of class `MapCollection`

Methods

x = "missing" Default constructor.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`MapCollection-class`

`mapName-methods` *accessors for the slot mapName*

Description

Set or get the value of the slot `mapName` in `object`.

Methods

object = "MareyMap" Set or get the value of the slot `mapName` in `object`.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`MareyMap-class`

`mapNames-methods` *mapNames*

Description

Returns the names of the maps contained in `object`.

Methods

object = "MapSet" Returns the names of the maps contained in `MapSet`

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`MapSet`

`MapSet-class` *Class "MapSet"*

Description

This class is a collection of MareyMap for a given set. Not that useful on its own, it is used for internal organization inside a MapCollection

Objects from the Class

Objects can be created by calls of the form `new("MapSet", ...)`. Or using the function `MapSet()`.

Slots

`maps`**:** Object of class `"list"` containing all the `MareyMap` of the set.

`setName`**:** Object of class `"character"` containing the name of the set

Methods

[`signature(x = "MapSet")`: returns a map from the list

[[<- `signature(x = "MapSet", i = "ANY", j = "ANY", value = "MareyMap")`: replaces a map in the list with a new one

[[`signature(x = "MapSet")`: returns a map from the list.

$ `signature(x = "MapSet")`: access a map by name in the list.

+ `signature(e1 = "MapSet", e2 = "MareyMap")`: adds a map to the collection.

- **-** `signature(e1 = "MapSet", e2 = "character")`: removes the map named e2 from the object e1.

coerce `signature(from = "MapSet", to = "data.frame")`: converts the specieMaps object into a data.frame.

mapNames `signature(object = "MapSet")`: returns a vector containing the names of all the maps.

setName<- `signature(object = "MapSet", value = "character")`: changes the name of the set. Also changes the named of all MareyMap objects inside the list.

setName `signature(object = "MapSet")`: returns the name of the set.

length `signature(object = "MapCollection")`: returns the length of the set

textFile `signature(object = "MapSet", file = "character")`: Writes the content of the MapSet object to file.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`MapCollection-class` `MareyMap-class`

Examples

```
data(Homo_sapiens_mean)
set <- Homo_sapiens_mean
setName(set) <- "Human"
set[["Chromosome 01"]]
```

`MapSet-methods` *Constructor for the class MapSet.*

Description

Creates an object of the class `MapSet`.

Methods

set_name = "character" creates a new object of the class `MapSet` with the name set_name.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

`MareyMap-class` *Class "MareyMap"*

Description

The class holds the physical and genetic positions of a marey map

Objects from the Class

Objects can be created by calls of the form `new("MareyMap", ...)` or using the function `MareyMap()`. However, most of the time you can more conveniently use already constructed objects contained in (c.f. code examples)

Slots

`setName`**:** Object of class `"character"`

`mapName`**:** Object of class `"character"` containing the name of the map, usually the name of the mapomosome it is mapping.

`markerNames`**:** Object of class `"vector"` containing the names of the markers of the map.

`physicalPositions`**:** Object of class `"vector"` containing the physical positions of the markers.

`geneticDistances`**:** Object of class `"vector"` containing the positions of the markers on the genetic map.

`markerValidity`**:** Object of class `"vector"` indicating for each marker if it is to be taken into account for interpolations.

`interpolations`**:** Object of class `"list"` containing the `interpolations` calculated on the map.

Methods

[`signature(x = "MareyMap")`: returns a vector combining marker name, physical position, genetic distance and marker validity for the i th marker

[[<- `signature(x = "MareyMap")`: replaces marker name, physical position, genetic distance and marker validity for the i th marker. value must be a vector containing 4 column named "name", "phys", "gen" and "valid".

[[<- `signature(x = "MareyMap", i = "ANY", j = "ANY", value = "ANY")`: replaces marker name, physical position, genetic distance and marker validity for the i th marker. value must be a vector containing 4 column named "name", "phys", "gen" and "valid".

[[`signature(x = "MareyMap")`: returns a vector combining marker name, physical position, genetic distance and marker validity for the i th marker

+ `signature(e1 = "MareyMap", e2 = "Interpolation")`: adds an interpolation to the map

coerce `signature(from = "MareyMap", to = "data.frame")`: convert the Marey map into a data.frame.

coerce `signature(from = "MareyMap", to = "NULL")`: returns NULL

coerce `signature(from = "data.frame", to = "MareyMap")`: create a MareyMap from a data.frame.

geneticDistances<- `signature(object = "MareyMap")`: replace the genetic distance with a new vector.

geneticDistances `signature(object = "MareyMap")`: returns the genetic distances of the markers

interpolation<- `signature(object = "MareyMap", inter_name = "character", value = "Interpolation")`: replaces the interpolation which name matches `inter_name` with the content of the parameter `value`.

interpolation `signature(object = "MareyMap", inter_name = "character")`: returns the interpolation which name matches `inter_name`.

interpolations<- `signature(object = "MareyMap")`: replace the list of interpolation with a new list.

interpolations `signature(object = "MareyMap")`: returns the list of interpolations.

mapName<- `signature(object = "MareyMap")`: replaces the name of the map

mapName `signature(object = "MareyMap")`: returns the name of the map

markerNames<- `signature(object = "MareyMap")`: replaces the marker names with a new vector of names

markerNames `signature(object = "MareyMap")`: returns a vector containing the names of the markers

markerValidity<- `signature(object = "MareyMap")`: replaces the marker validity with a new vector

markerValidity `signature(object = "MareyMap")`: returns a vector containing information about the validity of the markers

physicalPositions<- `signature(object = "MareyMap")`: replaces the vector of physical positions with a new vector

physicalPositions `signature(object = "MareyMap")`: returns a vector containing the physical positions of the markers

plot `signature(x = "MareyMap", y = "missing")`: plots the map as well as the interpolations (if any)

plotMarkers `signature(object = "MareyMap")`: plots only the markers of the map, without plotting the interpolations

plotModels `signature(object = "MareyMap")`: plot only the models of the interpolations

plotRates `signature(object = "MareyMap")`: plots only the recombination rates

query `signature(object = "MareyMap", pos = "numeric")`: returns the value of the local recombination rate at physical position pos as estimated by the interpolations defined on the map. Returns a list of integers (or NAs). each individual result may be accessed via result[[interpolation_name]]

removeMarker `signature(object = "MareyMap", value = "integer")`: remove the marker number `value`.

setName<- `signature(object = "MareyMap", value = "ANY")`: replaces the name of the set with a new `character`.

setName `signature(object = "MareyMap")`: returns the name of the set.

validPositions `signature(object = "MareyMap")`:

textFile `signature(object = "MareyMap", file = "character")`: Writes the map to text file.

Warning

vectors `markerNames`, `physicalPositions`, `geneticDistances` and `markerValidity` must always retain the same length.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`MapCollection-class` `MapSet-class`

Examples

```
data(Homo_sapiens_male)
chr7 <- Homo_sapiens_male[["Chromosome 07"]]
chr7 <- chr7 + MMLoess()
par(mfrow = c(2, 1))
plotMarkers(chr7)
```

`MareyMap-methods` *Constructor for object of the class MareyMap.*

Description

Creates an object of the class `MareyMap-class`.

Methods

data_table = "missing", column_names = "missing", set_name = "missing", map_name = "missing" Default constructor

data_table = "data.frame" fill the class using the provided data.frame

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`MareyMap-class`

`markerNames-methods` *Accessors for the class markerNames.*

Description

Get or Set slot `markerNames` in `object`.

Methods

object = "MareyMap" Get or Set slot `markerNames` in `MareyMap`.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`MareyMap-class`

`markerValidity-methods` *accessors for slot markerValidity*

Description

Get or set the value of the slot `MarkerValidity` in `object`.

Methods

object = "MareyMap" Get or set the value of the slot `MarkerValidity` in `MareyMap`.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`MareyMap-class`

`MMLoess-class` *Class "MMLoess". Interpolation method plug-in for MareyMap*

Description

This packages provides an interpolation method that estimates the recombination rates using `loess`. For more details on how this method works, see `loess`.

Objects from the Class

Objects can be created by calls of the form `new("MMLoess", ...)` or simply by calling `MMLoess()`.

Slots

`span`**:** Object of class `"numeric"`, parameters that controls the degree of smoothing.

`degree`**:** Object of class `"integer"`, the degree of the polynomials to be used, up to 2.

`model`**:** Object of class `"ANY"`

`name`**:** Object of class `"character"`, name of the interpolation.

`color`**:** Object of class `"character"`, color of the interpolation on the map.

`physicalPositions`**:** Object of class `"vector"`, physical positions of the markers of the map on which the interpolation is calculated.

`rates`**:** Object of class `"vector"`, the local estimation of the recombination rate at the position of each markers.

`visible`**:** Object of class `"logical"`, whether the interpolation should be plotted or not.

`persistent`**:** Object of class `"logical"`, whether the interpolation should be saved along with the map in the text file

Extends

Class `"Interpolation"`, directly.

Methods

createOrder `signature(object = "MMLoess")`: see `Interpolation-class`.

degree<- `signature(object = "MMLoess")`: changes the degree of the polynomials.

degree `signature(object = "MMLoess")`: returns the degree of the polynomials.

interpolate `signature(object = "MMLoess", map = "MareyMap")`: see `Interpolation-class`.

plotModel `signature(object = "MMLoess")`: see `Interpolation-class`.

plotRate `signature(object = "MMLoess")`: see `Interpolation-class`.

query `signature(object = "MMLoess", pos = "numeric")`: see `Interpolation-class`.

span<- `signature(object = "MMLoess")`: changes the value of the span.

span `signature(object = "MMLoess")`: returns the value of the span.

userParam `signature(object = "MMLoess")`: see `Interpolation-class`.

Author(s)

Aurélie Siberchicot `<aurelie.siberchicot@univ-lyon1.fr>` and Clément Rezvoy

See Also

`loess` `Interpolation-class`

Examples

```
data(Homo_sapiens_male)
chr7 <- Homo_sapiens_male[["Chromosome 07"]]
itr1 <- MMLoess()
color(itr1) <- "red"
span(itr1) <- 0.05
chr7 <- chr7 + itr1
```

`MMSlidingWindow-class` *Class "MMSlidingWindow" Interpolation method plug-in for MareyMap*

Description

This package provides an interpolation method that estimates the recombination rates using a sliding window approach.

Objects from the Class

Objects can be created by calls of the form `new("MMSlidingWindow", ...)` or using the method `SlidingWindow()`.

Slots

`size`**:** Object of class `"integer"`, the size of the windows in base pairs

`shift`**:** Object of class `"integer"`, the distance in number of base pairs between the centers of two consecutive windows.

`threshold`**:** Object of class `"integer"` minimum number of markers in a window to perform the regression

`results`**:** Object of class `"vector"` position of the model curve estimated at the center f each window.

`name`**:** Object of class `"character"` see `Interpolation-class`

`color`**:** Object of class `"character"` see `Interpolation-class`

`physicalPositions`**:** Object of class `"vector"` see `Interpolation-class`

`rates`**:** Object of class `"vector"` see `Interpolation-class`

`visible`**:** Object of class `"logical"` see `Interpolation-class`

`persistent`**:** Object of class `"logical"` see `Interpolation-class`

`geneticalDistances`**:**

Extends

Class `"Interpolation"`, directly.

Methods

createOrder `signature(object = "MMSlidingWindow")`: see `Interpolation-class`

interpolate `signature(object = "MMSlidingWindow", map = "MareyMap")`: see `Interpolation-class`

plotModel `signature(object = "MMSlidingWindow")`: see `Interpolation-class`

plotRate `signature(object = "MMSlidingWindow")`: see `Interpolation-class`

query `signature(object = "MMSlidingWindow", pos = "numeric")`: see `Interpolation-class`

shift<- `signature(object = "MMSlidingWindow")`: updates the value of the parameter `shift`

shift `signature(object = "MMSlidingWindow")`: returns the value of the parameter `shift`

size<- `signature(object = "MMSlidingWindow")`: updates the value of the parameter `size`

size `signature(object = "MMSlidingWindow")`: returns the value of the parameter `size`

threshold<- `signature(object = "MMSlidingWindow")`: updates the value of the parameter `threshold`

threshold `signature(object = "MMSlidingWindow")`: returns the value of the parameter `threshold`

userParam `signature(object = "MMSlidingWindow")`: see `Interpolation-class`

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`Interpolation-class`

Examples

```
data(Homo_sapiens_mean)
human7 <- Homo_sapiens_mean[["Chromosome 07"]]
itr1 <- MMSlidingWindow()
color(itr1) <- "red"
size(itr1) <- 5000000
shift(itr1) <- 1000000
human7 <- human7 + itr1
plot(human7)
```

`MMSpline3-class` *Class "MMSpline3" Interpolation using cubic splines*

Description

MMSPline3 is a S4 class providing an interpolation method for MareyMap. Interpolation is done using cubic splines. The class uses smooth.spline to carry out the interpolation.

Objects from the Class

Objects can be created by calls of the form `new("MMSpline3", ...)`. However you may instead use the function `MMSpline3`() and adjust the parameters afterwards using the accessor functions.

Slots

- `type`**:** Object of class `"character"` indicate which type of smoothing is applied, applicable values are "cross-validation", "spar" and "degree of freedom"
- `gcv`**:** Object of class `"logical"` indicating whether generalized cross validation is to be used when smoothing via cross validation.
- `df`**:** Object of class `"numeric"` holding the degree of freedom to use when the smoothing is done using degree of freedom
- `spar`**:** Object of class `"numeric"` value of the "spar" to be taken when smoothing the "spar" method.
- `model`**:** Object of class `"ANY"` slot used to keep the smooth.spline object after the interpolation.
- `name`**:** Object of class `"character"` holding the name of the interpolation.
- `color`**:** Object of class `"character"` holding the color of the interpolation
- `physicalPositions`**:** Object of class `"vector"` holding the valid physical positions of the map on which the interpolation is applied.
- `rates`**:** Object of class `"vector"` holding the local recombination rates at the position of each marker.
- `visible`**:** Object of class `"logical"` indicating whether the interpolation is to be drawn or not.
- `persistent`**:** Object of class `"logical"` indicating whether or not the interpolation should be taken into account when saving to text file.

Extends

Class `"Interpolation"`, directly.

Methods

- **df<-** `signature(object = "MMSpline3", value = "numeric")`: Changes the value of the degree of freedom.
- **df** `signature(object = "MMSpline3")`: returns the value of the degree of freedom.

gcv<- `signature(object = "MMSpline3", value = "logical")`: Turns on or off the generalized cross validation.

gcv `signature(object = "MMSpline3")`: indicates whether the generalized cross validation is on or not.

createOrder `signature(object = "MMSpline3")`: see `Interpolation-class`

interpolate `signature(object = "MMSpline3", map = "MareyMap")`: carries out the interpolation on the map passed as parameter. Called automatically when an interpolation is added to a map

plotModel `signature(object = "MMSpline3")`: Plot the interpolation's model of the Marey curve.

plotRate `signature(object = "MMSpline3")`: Plots the variations of the recombination rates across a mapomosome.

query `signature(object = "MMSpline3", pos = "numeric")`: query the estimation of the recombination rate in a specific position (or a vector of specific positions)

spar<- `signature(object = "MMSpline3", value = "numeric")`: Changes the value of the spar.

spar `signature(object = "MMSpline3")`: returns the value of the spar.

type<- `signature(object = "MMSpline3", value = "character")`: Changes the type of smoothing to be used.

type `signature(object = "MMSpline3")`: returns the smoothing method used.

userParam `signature(object = "MMSpline3")`: returns informations about the user parameter, used by tcl/tk interface but not that useful for a command line user.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`smooth.spline` `Interpolation-class` `MareyMap-class`

Examples

```
data(Homo_sapiens_mean)
human7 <- Homo_sapiens_mean[["Chromosome 07"]]
itr1 <- MMSpline3()
color(itr1) <- "red"
spar(itr1) <- 0.05
human7 <- human7 + itr1
plot(human7)
```

`name-methods` *Accessor for slot name.*

Description

Get or set the value of the slot `name` in `object`

Methods

object = "Interpolation" Get or set the value of the slot `name` in `Interpolation`.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`Interpolation`

`paramDefault-methods` *Accessors for slot paramDefault*

Description

Get or set the value of the slot `paramDefault` in `object`.

Methods

object = "InterpolationParam" Get or set the value of the slot `paramDefault` in `MareyMap-class`.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`InterpolationParam`

`paramDesc-methods` *Accessors for the slot paramDesc*

Description

Accessors for the slot `paramDesc` in `object`

Methods

object = "InterpolationParam" Accessors for the slot `paramDesc` in `InterpolationParam`

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`InterpolationParam`

`paramFun-methods` *accessors for the slot paramFun*

Description

Get or set the value of the slot `paramFun` in `object`

Methods

object = "InterpolationParam" Get or set the value of the slot `paramFun` in `InterpolationParam-class`.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`InterpolationParam`

`paramMax-methods` *Accessors for the slot paramMax.*

Description

Get or set the value of the slot `paramMax` in `object`.

Methods

object = "InterpolationParam" Get or set the value of the slot `paramMax` in `InterpolationParam`.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`InterpolationParam`

`paramMin-methods` *Accessors for slot paramMin.*

Description

Get or set the value of the parameter `paramMin` in `object`.

Methods

object = "InterpolationParam" Get or set the value of the parameter `paramMin` in `InterpolationParam`.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`InterpolationParam-class`

---

paramName-methods *Accessors for the slot paramName.*

Description

Get or set the value of the slot `paramName` in `object`.

Methods

object = "InterpolationParam" Get or set the value of the slot `paramName` in `InterpolationParam`.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`InterpolationParam-class`

---

paramType-methods *Accessors for the slot paramType.*

Description

Get or set the value of the slot `paramType` in `object`.

Methods

object = "InterpolationParam" Get or set the value of the slot `paramType` in `InterpolationParam`.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`InterpolationParam-class`

`paramValues-methods` *Accessor for the slot paramValues.*

Description

Get or set the value of the slot `paramValues` in `object`.

Methods

object = "InterpolationParam" Get or set the value of the slot `paramValues` in `InterpolationParam`.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`InterpolationParam-class`

`persistent-methods` *Accessors for the slot persistent.*

Description

Get or set the value of the slot `persistent` in `object`.

Methods

object = "Interpolation" Get or set the value of the slot `persistent` in `Interpolation`.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`Interpolation-class`

`physicalPositions-methods` *Accessors for the slot physicalPositions*

Description

Get or set the value of the slot `physicalPositions` `object`

Methods

object = "MareyMap" Get or set the value of the slot `physicalPositions` `MareyMap-class`

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`MareyMap-class`

`plotMarkers-methods` *plotMarkers*

Description

Plots the markers contained in `object`.

Methods

object = "MareyMap" Plots the markers of the map `MareyMap`

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`MareyMap-class`

`plotModel-methods` *plotModel*

Description

Plot the model of `object`

Methods

object = "Interpolation" Default function, plots nothing

object = "MMSlidingWindow" Plots the model of an interpolation of type MMSlidingWindow

object = "MMLoess" Plots the model of an interpolation of type MMLoess-class

object = "MMSpline3" Plots the model of an interpolation of type MMSpline3-class

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`Interpolation-class`

`plotModels-methods` *plotModels*

Description

plot models from `object`

Methods

object = "MareyMap" Calls plotModel on all the interpolation defined on the `MareyMap`.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`MareyMap-class`

`plotRate-methods` *plotRate*

Description

Plots the variation of the recombination rate as calculated by `object`

Methods

object = "Interpolation" Plots the variation of the recombination rate as calculated by the `Interpolation`.

object = "MMSlidingWindow" Plots the variation of the recombination rate as calculated by MMSlidingWindow.

object = "MMLoess" Plots the variation of the recombination rate as calculated by MMLoess.

object = "MMSpline3" Plots the variation of the recombination rate as calculated by MMSpline3.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`Interpolation-class`

`plotRates-methods` *plotRates*

Description

Plots the variations of the recombination rate as calculated by all the interpolations defined on the object.

Methods

object = "MareyMap" Plots the variations of the recombination rate as calculated by all the interpolations defined on the `MareyMap`.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`MareyMap-class`

`query-methods` *Methods for Function query*

Description

query an `object` for a given (or several) `position`(s).

Methods

- **object = "Interpolation", pos = "integer"** Default method.
- **object = "MareyMap", pos = "numeric"** Query the local recombination rate as calculated by all the Interpolation defined on the map.
- **object = "MMSlidingWindow", pos = "numeric"** Query the local recombination rate as calculated by MMSlidingWindow.
- **object = "MMLoess", pos = "numeric"** Query the local recombination rate as calculated by MMLoess.
- **object = "MMSpline3", pos = "numeric"** Query the local recombination rate as calculated by MMSpline3.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`Interpolation-class`

`rates-methods` *Accessors for the slot rates*

Description

Get or set the value of the slot `rates` in `object`

Methods

- **object = "Interpolation"** Get or set the value of the slot `rates` in `Interpolation`.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

`registerInterpolationMethod-methods` *registerInterpolationMethods*

Description

Adds an `Interpolation` method to the interpolation register.

Methods

name = "character", classname = "character" Register an interpolation method with a given `name` and a contructor function `classname`.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`Interpolation-class`

`removeMarker-methods` *removeMarker*

Description

Removes a marker from an `object`

Methods

object = "MareyMap", value = "integer" Removes the marker number `value` from a `MareyMap-class`.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`MareyMap-class`

`setName-methods` *Accessors for the slot setName.*

Description

Get or set the value of the slot `setName` in `object`.

Methods

object = "MareyMap" Get or set the value of the slot `setName` in `MareyMap`.

object = "MapSet" Get or set the value of the slot `setName` in `MapSet`.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`MareyMap` `MapSet`

`setNames-methods` *setNames*

Description

returns the names of the set contained in object.

Methods

object = "MapCollection" Returns the name of the set contained in `MapCollection`.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`MapCollection`.

`startMareyMapGUI-methods` *MareyMapGUI launcher*

Description

launches the graphical user interface of MareyMapGUI

Methods

dummy = "missing" Default method.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

`textFile-methods` *textFile*

Description

Writes object to text file.

Methods

object = "MapCollection", file = "character" Writes a `MapCollection` to text file.

object = "MareyMap", file = "character" Writes a `MareyMap` to text file.

object = "MapSet", file = "character" Writes a `MapSet` to text file.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`MareyMap-class` `MapCollection-class` `MapSet-class`

`userParam-methods` *userParam*

Description

Returns information about the parameters modifiable by the user in object

Methods

- **object = "Interpolation"** Returns information about the parameters modifiable by the user in `Interpolation`
- **object = "MMSlidingWindow"** Returns information about the parameters modifiable by the user in MMSlidingWindow
- **object = "MMLoess"** Returns information about the parameters modifiable by the user in MMLoess
- **object = "MMSpline3"** Returns information about the parameters modifiable by the user in MMSpline3

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`Interpolation-class` `InterpolationParam-class`

`validPositions-methods` *validPositions*

Description

Returns the positions of object which are valid.

Methods

- **object = "MareyMap"** Returns the physical positions of MareyMap which are valid.

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

See Also

`MareyMap-class`

`visible-methods` *Accessors for the slot visible*

Description

Get of set the value of the slot `visible` in `object`

Methods

object = "Interpolation" Get of set the value of the slot `visible` in `Interpolation-class`

Author(s)

Aurélie Siberchicot <`aurelie.siberchicot@univ-lyon1.fr`> and Clément Rezvoy

Index