# Package ‘TrajDataMining’

## May 10, 2018

**Type**  Package  
**Title**  Trajectories Data Mining  
**Version**  0.1.6  
**Date**  2018-05-08  
**Description**  Contains a set of methods for trajectory data preparation, such as filtering, compressing and clustering, and for trajectory pattern discovery.

**License**  MIT + file LICENSE  
**LazyData**  TRUE  
**Encoding**  UTF-8  
**Depends**  R (>= 3.0.0)  
**ByteCompile**  true  
**Imports**  sp, trajectories, xts, spacetime, RPostgreSQL, geosphere, methods, rgdal  
**Suggests**  testthat, knitr, rmarkdown, covr, ggplot2, rgeos, magrittr  
**VignetteBuilder**  knitr  

**URL**  [https://github.com/INPEtrajectories/trajdatamining](https://github.com/INPEtrajectories/trajdatamining)  
**RoxygenNote**  6.0.1  
**NeedsCompilation**  no  
**Author**  Diego Monteiro [aut, cre],  
Karine Reis [ctb],  
Rafael Santos [ctb],  
Carlos Alberto [ctb],  
Felipe Carvalho [ctb]  

**Maintainer**  Diego Monteiro <dvm1607@gmail.com>  
**Repository**  CRAN  
**Date/Publication**  2018-05-10 21:29:42 UTC
R topics documented:

A1 ................................................................. 2
A2 ................................................................. 3
createSpatialCluster ........................................... 3
DataSourceInfo-class ......................................... 4
directionCluster ............................................... 5
douglasPeucker ................................................ 6
douglasPeuckerRP ............................................ 7
IndexToTrack .................................................. 7
LimitedNeighborhood .......................................... 8
owMeratniaBy .................................................. 9
owMeratniaByCollection ...................................... 10
partner ........................................................ 11
RightSize ....................................................... 12
sendPartnerPairsToDB ......................................... 13
singleDifftrack-class ........................................ 14
SlowestNeighborhood ......................................... 14
speedCluster .................................................. 15
speedFilter ..................................................... 16
tracksCollection ............................................. 16

Index 18

A1 A trajectory of elephant sea

Description

object that contains distance, duration, speed and direction.

Usage

A1

Format

A trajectory with 2218 rows and 7 columns:

id  id each point of trajectory
x  latitude
y  longitude
time  time stamps of the track points
endtime  end time stamps of the track points
timeindex  time index
ones  ones of the track
A trajectory of elephant sea

Description

A object that contains distance, duration, speed and direction.

Usage

A2

Format

A trajectory with 2148 rows and 7 columns:

id  id each point of trajectory
x  latitude
y  longitude
time  time stamps of the track points
timeindex  id of time stamps
ones  ones of the track

createSpatialCluster  Create Spatial Cluster

Description

Method for create a spatial cluster

Usage

createSpatialCluster(A3, clusterlist)

## S4 method for signature 'Track,list'
createSpatialCluster(A3, clusterlist)

Arguments

A3  Track object
clusterlist  list of cluster positions create spatial objects
**DataSourceInfo-class**

**Value**

cluster of polygons

**Author(s)**

Deigo Monteiro

**Examples**

```r
data_source <- createSpatialCluster(A1, speedCluster(A1, mean(A1@connections$speed),
                                       12, min(A1@connections$speed)+4))
```

---

**DataSourceInfo-class  Data Source Info**

**Description**

Class to connect in a database

**Slots**

- `user`  User of database
- `title`  Title of database
- `accessDriver`  The database access driver
- `host`  Host of the database (e.g. localhost)
- `port`  Port the database (e.g. 5432)
- `timeout`  Timeout time of connection
- `password`  Password of database
- `db`  Database name
- `encoding`  Encoding of database (e.g. CP1252)
- `dbtype`  The type of the database (e.g. POSTGIS)
- `path`  Path of the database
directionCluster

**Description**

That given a Track and maximum change parameter, returns regions where direction changed more than the defined parameter.

**Usage**

```
directionCluster(track, minD, minT, tolerance)
```

```
## S4 method for signature 'Track,numeric,numeric,numeric'
directionCluster(track, minD, minT, tolerance)
```

**Arguments**

- **track**: Represents a single trajectory followed by a person, animal or object.
- **minD**: Is the minimum direction change.
- **minT**: Is the minimum period at the speed.
- **tolerance**: Is the maximum change parameter.

**Value**

returns regions where direction changed more than the defined parameter.

**Author(s)**

Diego Monteiro

**Examples**

```
cluster <- createSpatialCluster(A1, directionCluster(A1, 5, 10, 2))
```
**Description**

Douglas-Peucker which reduces trajectories by preserving spatial precisions.

**Usage**

```r
douglasPeucker(A1, dist)
```

## S4 method for signature 'Track,numeric'

```r
douglasPeucker(A1, dist)
```

**Arguments**

- `A1` Represents a single trajectory followed by a person, animal or object
- `dist` Distance time series

**Value**

reduces trajectories by preserving spatial precisions

**Author(s)**

Diego Monteiro

**Examples**

```r
## Not run:
library(ggplot2)

dist <- max(A1@connections$distance)
douglasp <- douglasPeucker(A1, dist)
df <- data.frame(x=douglasp@sp@coords[,1], y=douglasp@sp@coords[,2])
ggplot(df, aes(x=df$x, y=df$y)) + geom_path(aes(group = 1))
```

## End(Not run)
**douglasPeuckerRP**  
*Douglas Peucker RP*

---

**Description**

Method that reduces a trajectory spatially with first point and last point

**Usage**

```r
douglasPeuckerRP(A1, firstp, lastp, dist)
```

```
# S4 method for signature 'Track,numeric,numeric,numeric'
douglasPeuckerRP(A1, firstp, lastp, dist)
```

**Arguments**

- `A1` track object
- `firstp` given first point
- `lastp` given last point
- `dist` distance

**Author(s)**

Diego Monteiro

**Examples**

```r
doug <- douglasPeuckerRP(A1, 20, 200, 2000)
```

---

**IndexToTrack**  
*Conversão de index pra track*

---

**Description**

Conversão de index pra track

**Usage**

```r
IndexToTrack(A1, index)
```
**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>track</td>
<td>Represents a single trajectory followed by a person, animal or object</td>
</tr>
<tr>
<td>ini</td>
<td>Order list of track speed</td>
</tr>
<tr>
<td>minT</td>
<td>Is the minimum period at the speed</td>
</tr>
<tr>
<td>cln</td>
<td>Cluster identification</td>
</tr>
<tr>
<td>cl</td>
<td>Empty list</td>
</tr>
<tr>
<td>avg</td>
<td>Average value of speed</td>
</tr>
<tr>
<td>sl</td>
<td>Is the speed limit</td>
</tr>
</tbody>
</table>

**Author(s)**

Diego Monteiro
owMeratniaBy

Description

Method that reduces trajectories spatiotemporally

Usage

owMeratniaBy(A1, dist, speed)

## S4 method for signature 'Track,numeric,numeric'

owMeratniaBy(A1, dist, speed)

Arguments

- **A1**: Represents a single trajectory followed by a person, animal or object.
- **dist**: Distance time series
- **speed**: Speed of track

Value

Reduces trajectories spatiotemporally

Author(s)

Diego Monteiro

Examples

```r
## Not run:
library(ggplot2)

speed <- max (A1@connections$speed)
distance <- max (A1@connections$distance)

ow <- owMeratniaBy(A1,distance,speed)
df <- data.frame(x=ow@sp@coords[,1],y=ow@sp@coords[,2])
ggplot(df,aes(x=df$x,y=df$y))+geom_path(aes(group = 1), arrow = arrow(),color='blue')

## End(Not run)
```
owMeratniaByCollection

Ow Meratnial By Collection

Description
Method that reduces a set of trajectories spatiotemporally

Usage

owMeratniaByCollection(A1, dist, speed)

## S4 method for signature 'TracksCollection,numeric,numeric'
owMeratniaByCollection(A1, dist, speed)

Arguments

A1 Represents a collection of trajectories followed by different persons, animals or objects
dist Distance time series
speed Speed of object

Value
Trajectory spatiotemporally reduced

Author(s)
Diego Monteiro

Examples

library(magrittr)
library(sp)

library(ggplot2)
ow <- owMeratniaByCollection(TracksCollection,13804.84 ,0.03182201) %>% coordinates()
df <- data.frame(x=ow[,1],y=ow[,2])
ggplot(df,aes(x=x,y=y))+geom_path(aes(group = 1), arrow = arrow(),color='blue')
Description

Method to recognize trajectories that stay together, based on trajectory distance time series analysis

Usage

```r
partner(A1, A2, dist, maxtime, mintime, datasource, tablename)
```

```
## S4 method for signature
## 'Track,Track,numéric,numéric,numéric,DataSourceInfo,character'
partner(A1,
   A2, dist, maxtime, mintime, datasource, tablename)
```

```
## S4 method for signature
## 'Track,Track,numéric,numéric,numéric,PostgreSQLConnection,character'
partner(A1,
   A2, dist, maxtime, mintime, datasource, tablename)
```

```
## S4 method for signature
## 'Track,Track,numéric,numéric,numéric,logical,missing'
partner(A1, A2,
   dist, maxtime, mintime, datasource, tablename)
```

```
## S4 method for signature
## 'TracksCollection,missing,numéric,numéric,numéric,missing,missing'
partner(A1,
   A2, dist, maxtime, mintime, datasource, tablename)
```

```
## S4 method for signature
## 'TracksCollection,
##   TracksCollection,
##   numérique,
##   numérique,
##   numérique,
##   missing,
##   missing'
partner(A1,
   A2, dist, maxtime, mintime, datasource, tablename)
```
## RightSize

**Description**

Right size verifier

**Usage**

```r
RightSize(diffTracks, begintime, endtime, sizeMultiplier)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>diffTracks</td>
<td>Diftrack</td>
</tr>
<tr>
<td>begintime</td>
<td>Begin time</td>
</tr>
<tr>
<td>endtime</td>
<td>End time</td>
</tr>
<tr>
<td>sizeMultiplier</td>
<td>number will multiplier the diff</td>
</tr>
</tbody>
</table>

### Arguments

- **A1** Represents a single trajectory followed by a person, animal or object.
- **A2** Represents a single trajectory followed by a person, animal or object.
- **dist** Distance that two objects can stay apart
- **maxtime** Maximum time period that two objects can stay apart
- **mintime** Minimum time period that two objects must stay together
- **datasource** Is object class DataSourceInfo
- **tablename** The name of the table database

### Value

List with begin time and end time stamps of two objects partner

### Author(s)

Diego Monteiro

### Examples

```r
partner(A1,A2,110792,2277,0,FALSE)
```
sendPartnerPairsToDB

Value

boolean

sendPartnerPairsToDB  send Partner Pairs To DataBase

Description

Method that sends found partners to a PostGIS database
Method that sends found partners to a PostGIS database with object DataSourceInfo
Method that sends found partners to a PostGIS database using object PostgreSQLConnection

Usage

sendPartnerPairsToDB(dataframe, dataSourceInfo, tablename)

## S4 method for signature 'list,DataSourceInfo,character'
sendPartnerPairsToDB(dataframe,
    dataSourceInfo, tablename)

## S4 method for signature 'list,PostgreSQLConnection,character'
sendPartnerPairsToDB(dataframe,
    dataSourceInfo, tablename)

Arguments

dataframe  Dataframe list
dataSourceInfo  A object class dataSourceInfo
tablename  Name of table

Value

send the partners list for a database
**SingleDiffTrack**

**Description**

My compare method get distances between 2 Track objects for each point in time where they overlap and create a corresponding line.

**Arguments**

- `tr1` represents a single trajectory followed by a person, animal or object.
- `tr2` represents a single trajectory followed by a person, animal or object.

**Details**

- `@import xts`

**Value**

- a difftrack object

**Author(s)**

- Diego Monteiro

---

**SlowestNeighborhood**

**Description**

Method for check slowest neighborhood

**Usage**

`SlowestNeighborhood(track, ini, minT, cl)`

**Arguments**

- `track` Represents a single trajectory followed by a person, animal or object
- `ini` Order list of track speed
- `minT` Is the minimum period at the speed
- `cl` Empty list

**Author(s)**

- Diego Monteiro
speedCluster

Speed Cluster

Description
Method for check the regions where speed was lower than the defined parameter

Usage
speedCluster(trackL avgL mintL slI

## S4 method for signature 'Track,numeric,numeric,numeric'
speedCluster(track, avg, minT, sl)

Arguments

- **track**
  - Represents a single trajectory followed by a person, animal or object
- **avg**
  - Is the average speed of track
- **minT**
  - Is the minimum period at the speed of track
- **sl**
  - Is the speed limit of track

Details
Order the speed so it will start with the slowest speed cluster

Value
Returns regions where speed was lower than the defined parameter

Author(s)
Diego Monteiro

Examples

```r
avgSpeed <- mean(A1@connections$speed)
minSpeed <- min(A1@connections$speed)
speed <- speedCluster(A1, avgSpeed, minSpeed, 586)
```
**Description**

A speed filter that filters out trajectory observations whose speeds are above a user-defined maximum velocity.

**Usage**

```r
speedFilter(A1, speed)
```

## S4 method for signature 'Track,numeric'
```r
speedFilter(A1, speed)
```

**Arguments**

- `A1`: Represents a single trajectory followed by a person, animal or object.
- `speed`: Is the maximum speed parameter.

**Author(s)**

Diego Monteiro

**Examples**

```r
library(ggplot2)
speed <- min(A1@connections$speed)
sf <- speedFilter(A1, speed)
df <- data.frame(x=sf@sp@coords[,1], y=sf@sp@coords[,2])
ggplot(df, aes(x=df$x, y=df$y)) + geom_path(aes(group = 1), arrow = arrow(), color='blue')
```

---

**tracksCollection**

*A tracks collection of a plataform argos*

**Description**

A dataset containing tracks objects.

**Usage**

```r
tracksCollection
```
Format

A trajectory with 7 columns:

- **id**  id rows
- **xmin**  latitude minimum
- **xmax**  latitude maximum
- **ymin**  longitude minimum
- **ymax**  longitude maximum
- **tmax**  maximum time
- **tmin**  minimum time
- **timeindex**  time index
- **n**  ones of the track
Index

*Topic datasets
- A1, 2
- A2, 3
  - tracksCollection, 16

A1, 2
A2, 3

createSpatialCluster, 3
createSpatialCluster, Track, list-method (createSpatialCluster), 3

DataSourceInfo (DataSourceInfo-class), 4
DataSourceInfo-class, 4
directionCluster, 5
directionCluster, Track, numeric, numeric, numeric-method (directionCluster), 5
douglasPeucker, 6
douglasPeucker, Track, numeric-method (douglasPeucker), 6
douglasPeuckerRP, 7
douglasPeuckerRP, Track, numeric-method (douglasPeuckerRP), 7

IndexToTrack, 7

LimitedNeighborhood, 8

owMeratniaBy, 9
owMeratniaBy, Track, numeric, numeric-method (owMeratniaBy), 9
owMeratniaByCollection, 10
owMeratniaByCollection, TracksCollection, numeric, numeric-method (owMeratniaByCollection), 10

partner, 11
partner, Track, Track, numeric, numeric, numeric, DataSourceInfo, character-method (partner), 11
partner, Track, Track, numeric, numeric, numeric, logical, missing-method (partner), 11
partner, TracksCollection, numeric, numeric, numeric, missing-method (partner), 11
partner, TracksCollection, Track, numeric, numeric, numeric, missing-method (partner), 11
partner, TracksCollection, TracksCollection, numeric, numeric, missing-method (partner), 11

RightSize, 12

sendPartnerPairsToDB, 13
sendPartnerPairsToDB, list, DataSourceInfo, character-method (sendPartnerPairsToDB), 13
sendPartnerPairsToDB, list, PostgreSQLConnection, character-method (sendPartnerPairsToDB), 13
 singledifftrack-class, 14
SlowestNeighborhood, 14
speedCluster, 15
speedCluster, Track, numeric, numeric, numeric-method (speedCluster), 15
speedFilter, 16
speedFilter, Track, numeric-method (speedFilter), 16

tracksCollection, 16