Package ‘trackR’

May 19, 2020

Type Package

Title R tools and functions to compute jump analysis, describe and infer from single-molecule tracking data

Version 0.1.3

Author Julien Godet

Maintainer JuG <julien.godet@unistra.fr>

Description The jump-distance (JD), defined as the distance travelled by a single molecule during a fixed time lag Tau has a distribution (JDD) over the population of molecules that reflects fine features of the underlying transport. The JDD analysis takes advantage of the fact that modern, powerful imaging systems and tracking algorithm produce generally a large number of individual trajectories allowing to plot well-resolved JD distributions

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

R topics documented:

calcDist *Compute euclidian distance between 2 points*

Description

Compute euclidian distance between 2 points

Usage

```
calcDist(data, spaceRes = 1)
```

Arguments

`data`	a dataframe with 2 lines and 2 or 3 columns (x,y,(z))

Author(s)

JuG

Examples

```
data2D <- data.frame(x =c(0,1), y=c(0,1))
calcDist(data2D)
data3D <- data.frame(x =c(0,1), y=c(0,1), z=c(0,1))
calcDist(data3D)
```

calcMSD *Calculate Mean Square Distance*

Description

Calculate Mean Square Distance

Usage

```
calcMSD(data, N)
```

Arguments

`data`	data frame (including trace, x, y and t)
`N`	for MSD, dt should be up to 1/N of number of data points (4 recommended)

Author(s)

JuG

Examples

```
xmlPath <- "//Users/jgodet/Seafile/MaBibliotheque/Hanna/1_3_MMStack_Pos0.ome.xml"
xmlPath <- "/Users/jgodet/Seafile/MaBibliotheque/Code/TrackMate/nmeth.2808-sv1.xml"
data <- readTrackMateXML(XMLpath = xmlPath)
data$jump<-jump(data, spaceRes=1)
msdMat <- calcMSD(data, N=3)
plotMSD(msdMat, col='blue')
```

cleanNearest *Do something*

Description

Do something

Usage

```
cleanNearest(data, k = 10)
```

Author(s)

JuG

clusterTraces *cluster traces in space (and/or time) using DBscan*

Description

cluster traces in space (and/or time) using DBscan

Usage

```
clusterTraces(dtf, coord = c("x", "y"), eps, minPts, method, ...)
```

Arguments

`dtf`	data frame with coordinates and traces ID
`coord`	coordinates dataframe (x,y) or (x,y,t)
`eps`	Reachability distance, see Ester et al. (1996).
`minPts`	Reachability minimum no. of points, see Ester et al. (1996).
`method`	"dist" treats data as distance matrix (relatively fast but memory expensive), "raw" treats data as raw data and avoids calculating a distance matrix (saves memory but may be slow), "hybrid" expects also raw data, but calculates partial distance matrices (very fast with moderate memory requirements).

Author(s)

JuG

Examples

```
xmlPath <- "//Users/jgodet/Seafile/MaBibliotheque/Hanna/1_3_MMStack_Pos0.ome.xml"
data <- readTrackMateXML(XMLpath = xmlPath)
data$jump<-jump(data, spaceRes=1)
bacteria <- clusterTraces(dtf = data, eps = .2,minPts = 100)
table(bacteria)
bact1 <- data[which(bacteria==7),]
summary(bact1)
bact1 %>% select(x,y) %>% plot(., asp=1)
drawRod(data = getContour(bact1),col='green')
drawRod(data = getContour(cleanNearest(data = bact1,k = 3)),col='blue')
```

createSpheroCylinderCoord

Do something

Description

None

Usage

```
createSpheroCylinderCoord(
  N = 1000,
  l = 2,
  r = 0.3,
  initialPos = c(1, 1, 0),
  rot = 0
)
```

Arguments

N	umber of points
l	length of the rod
r	radius
initialPos	Initial position
rot	Rotation

Details

None

Author(s)

JuG

Examples

```
coord <- createSpheroCylinderCoord(rot=10)
plot(coord,asp=1)
```

`createTestEllipse` *Create a test ellipse*

Description

Create a test ellipse

Usage

```
createTestEllipse(
  Rx = 300,
  Ry = 200,
  Cx = 250,
  Cy = 150,
  Rotation = 0.4,
  N = 200,
  NoiseLevel = 10
)
```

Arguments

`Rx`	X-radius
`Ry`	Y-radius
`Cx`	X-center
`Cy`	Y-center
`Rotation`	Radians
`N`	Nb of points
`NoiseLevel`	Gaussian Noise level

Author(s)

JuG

`drawRod` *Draw rod contour*

Description

Draw rod contour

Usage

```
drawRod(data, lwd, ...)
```

Arguments

`data`	coordinates of the contour

Author(s)

JuG

Examples

```
coord <- createSpheroCylinderCoord( l =1.8,r=.3,rot=120)
plot(coord[,1:2],asp=1,pch='.')
gcont <- getContour(coord[,1:2],drawContour = TRUE)
drawRod(gcont,lty=3)
```

fitEllipse *Adjust an ellipse on a contour plot*

Description

Least squares fitting of an ellipse to point data

Usage

```
fitEllipse(x, y = NULL)
```

Arguments

x	x.coordinates
y	y.coordinates

Details

# Least squares fitting of an ellipse to point data using the algorithm described in Radim Halir & Jan Flusser. 1998 Adapted from the original Matlab code by Michael Bedward (2010) michael.bedward@gmail.com Subsequently improved by John Minter (2012) http://lastresortsoftware.blogspot.fr/2012/09/fitting-ellipse-to-point-data.html

Author(s)

JuG

References

Radim Halir & Jan Flusser. 1998. Numerically stable direct least squares fitting of ellipses. Proceedings of the 6th International Conference in Central Europe on Computer Graphics and Visualization. WSCG ’98, p. 125-132

Examples

```
ellipse <- createTestEllipse(NoiseLevel = 50)
plot(ellipse)
efit <- fitEllipse(ellipse)
e <- getEllipse(efit)
lines(e,col='red')
```

`fitJumpDistECDF` *Do something*

Description

Do something

Usage

```
fitJumpDistECDF(
  data,
  nbPop = 1,
  initECDF = list(D1 = 0.05),
  timeRes = 1,
  xlab,
  ylab,
  main,
  cex,
  colLine = "red",
  breaks = 150,
  ...
)
```

Arguments

`data`	dataframe with jump
`nbPop`	number of populations of diffusing species (1 or 2)
`timeRes`	time resolution per unit of jump
`initHist`	initialisation parameters for the nls fit for exmaple list(D2 = 200, D1=0.1) or list(D2 = .01, D1=.1, D3=10, D4=100)

Author(s)

JuG

Examples

```
xmlPath <- "/Users/jgodet/Seafile/MaBibliotheque/Code/TrackMate/nmeth.2808-sv1.xml"
xmlPath <- "/Users/jgodet/Seafile/MaBibliotheque/Code/TrackMate/nmeth.2808-sv3.xml"
data <- readTrackMateXML(XMLpath = xmlPath)
data$jump<-jump(data, spaceRes=1)
fitJumpDistECDF(data=data, nbPop = 1)
fitJumpDistECDF(data=data, nbPop = 1, cex=.85, colLine = 'green')
fitJumpDistECDF(data=data, nbPop = 2,  initECDF = list(D1=0.001, D2=.4, D3=.1), cex=.85, colLine = 'green')
```

fitJumpDistHist *Fit jump distribution histogram*

Description

Fit jump distribution histogram

Usage

```
fitJumpDistHist(
  data,
  nbPop = 1,
  initHist = list(D2 = 200, D1 = 0.1),
  timeRes = 1,
  breaks = 100,
  xlim,
  ylim,
  xlab,
  main,
  ...
)
```

Arguments

data	dataframe with jump
nbPop	number of populations of diffusing species (1 or 2)
initHist	initialisation parameters for the nls fit for exmaple list(D2 = 200, D1=0.1) or list(D2 = .01, D1=.1, D3=10, D4=100)
timeRes	time resolution per unit of jump
breaks	binning of the histogram (number of breaks)

Author(s)

JuG

Examples

```
xmlPath <- "/Users/jgodet/Seafile/MaBibliotheque/Code/TrackMate/nmeth.2808-sv1.xml"
data <- readTrackMateXML(XMLpath = xmlPath)
data$jump<-jump(data, spaceRes=1)
fitJumpDistHist(data=data, nbPop = 1, initHist = list(D2 = 200,  D1=0.1))
fitJumpDistHist(data=data, xlim=c(0,10), col=rgb(.2,.2,.2,.2), main="test")
# Results are affected by the histogram binning
fitJumpDistHist(data=data, breaks = 30)
fitJumpDistHist(data=data, breaks = 150)
fitJumpDistHist(data=data, nbPop=3)
fitJumpDistHist(data=data, nbPop=2, initHist = list(D2 = .01,  D1=.1, D3=10, D4=100))
```

`getContour` *Return the convexhull contour of coordinates*

Description

Return the convexhull contour of corrdinates

Usage

```
getContour(data, drawContour = FALSE, ...)
```

Arguments

`data`	coordinate data
`drawContour`	if TRUE draw a convexhull contour line
`nbPix`	box size for the owin in ppp object

Author(s)

JuG

`getEllipse` *Create coordinates of an ellipse from fit ellipse parameters*

Description

Create coordinates of an ellipse from fit ellipse parameters

Usage

```
getEllipse(fit, n = 360)
```

Arguments

`fit`	a fit object from fitEllipse
`n`	number of corrdinates points to build

Author(s)

JuG

histJd *Plot JD histogram*

Description

Plot JD histogram (inherit from hist)

Usage

```
histJd(jd, breaks, xlim, freq, pixSize, xlab, panel.first, ...)
```

Arguments

`jd`	jump distribution vector
`breaks`	histogram breaks
`xlim`	xlim
`pixSize`	pixel size in µm (if picels are not calibrated in trackMate)
`xlab`	xlab
`panel.first`	panel first options (use for exmaple grid())

Value

graph

Author(s)

JuG

Examples

```
histJd(jd, main="", panel.first=grid())
box()
```

jump *Compute jump distance*

Description

Compute jump distance within a frame between two consecutive frames. Note that if a gap exist between consecutive frames (gap-closing frame gap) the jump is calculated as the average displacement (distance / delta frame)

Usage

```
jump(data, spaceRes)
```

Arguments

`data`	Dataframe with track (trace identification) and Frame variable
`spaceRes`	Pixel size

Author(s)

Julien Godet

Examples

```
a <- 2
b <- 2 * a
```

plotDiffusionMap *Plot Diffusion Map*

Description

Plot Diffusion Map

Usage

```
plotDiffusionMap(data, traceNb, nbPix = 20, jumpMax, nbCol = 10)
```

Arguments

`data` data frame (muste contain trace, x, y, and jump)

`traceNb` (optional) number id of the traces to plot

`nbPix` number of pixel (or calibrated distance) to add on each x-and y-range tail

Details

Plot Diffusion Map or more exactly jump distance map as in Fig 2c of Gasser et al. 2020 doi:10.1017/S003358351900015

Value

plot

Author(s)

JuG

Examples

```
xmlPath <- "/Users/jgodet/Seafile/MaBibliotheque/Code/TrackMate/nmeth.2808-sv1.xml"
data <- readTrackMateXML(XMLpath = xmlPath)
data$jump<-jump(data, spaceRes=1)
plotDiffusionMap(data = data,traceNb = c(12),nbPix = 5 )
plotDiffusionMap(data = data,nbPix = 5 )
```

plotMSD *Plot MSD*

Description

Plot MSD

Usage

```
plotMSD(
  msdData,
  deltaT,
  fitMSD = TRUE,
  printMSDfit = TRUE,
  npoint4fit = 4,
  ...
)
```

Arguments

`msdData`	MSD data (list) (output of calcMSD function)
`deltaT`	time elapsed between two consecutive frame
`fitMSD`	boolean
`printMSDfit`	boolean
`npoint4fit`	number of points to use for MSD fitting

Author(s)

JuG

readTrackMateXML *read TrackMate XML output files*

Description

read TrackMate XML output files

Usage

```
readTrackMateXML(XMLpath, multiCore = TRUE)
```

Arguments

`XMLpath`	path to the xml file
`multiCore`	use multi-cores (boolean)

Value

data frame

Author(s)

JuG

Examples

```
xmlPath <- "/Users/jgodet/Seafile/MaBibliotheque/Code/TrackMate/nmeth.2808-sv1.xml"
system.time(
data <- readTrackMateXML(XMLpath = xmlPath)
)
data <- readTrackMateXML(XMLpath = xmlPath, multiCore = FALSE)
data$jump<-jump(data, spaceRes=1)
hist(data$jump, breaks=150)
```

---

readTrackMateXmlTrace *read TrackMate XML traces output files*

Description

Function to read TrackMate XML traces output files: much faster than reading the complete TRack Mate XML file

Usage

```
readTrackMateXmlTrace(XMLpath, timeRes = 1)
```

Arguments

XMLpath	XML file path
timeRes	time exposure per frame

Author(s)

JuG

Examples

```
xmlPath <- "/Users/jgodet/Seafile/MaBibliotheque/Code/TrackMate/nmeth.2808-sv1Trace.xml"
data <- readTrackMateXmlTrace(XMLpath = xmlPath)

data$jump<-jump(data, spaceRes=1)
hist(data$jump, breaks=150)
fitJumpDistHist(data=data, breaks = 150)
msdMat <- calcMSD(data, N=2)
plotMSD(msdMat, col='blue')
```

Index