Package: RSP (via r-universe)

August 21, 2024

Title Refined Shortest Paths

Version 1.0.5

Description The RSP toolkit is a method for analysing the fine scale movements of aquatic animals tracked with passive acoustic telemetry in estuarine environments, that accounts for the surrounding land masses. The animal movements between detections are recreated to have occurred exclusively in water and the utilisation distribution areas are limited by the land contours, providing realistic estimations of space use.

License GPL-3 + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Depends R (>= 4.0)

Imports actel, cmocean, crayon, data.table, dplyr, gdistance, geosphere, gganimate, ggplot2, graphics, lubridate, magrittr, methods, move, plyr, raster, stats, stringr, sf, terra, tidyr, utils, reshape2, callr

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

URL https://github.com/YuriNiella/RSP

BugReports https://github.com/YuriNiella/RSP/issues

Repository https://ocean-tracking-network.r-universe.dev

RemoteUrl https://github.com/yuriniella/RSP

RemoteRef HEAD

RemoteSha 5d3ea47eaa24d24a79fac0981aad91055511e461

Contents

addCentroids . . . 2
addRecaptures . . . 4
addStations . . . 4
animateTracks . . . 5
calcRSP . . . 7
dynBBMM . . . 8
getAreas . . . 9
getAreaStep . . . 10
getCentroids . . . 12
getDistances . . . 13
getDistPoint . . . 14
getOverlaps . . . 15
includeRSP . . . 16
nameTracks . . . 17
plotAreas . . . 17
plotContours . . . 18
plotDensities . . . 20
plotDistances . . . 21
plotOverlaps . . . 22
plotRaster . . . 23
plotTracks . . . 25
prepareDetections . . . 26
runRSP . . . 27
suggestSize . . . 28

Index **30**

`addCentroids` *Add group centroid location to an existing plot*

Description

Add group centroid location to an existing plot

Usage

```
addCentroids(
  input,
  type,
  tag = NULL,
  track = NULL,
  timeslot = NULL,
  shape = 21,
  size = 1.5,
  colour = "white",
  fill = "cyan"
)
```

Arguments

input	The output of getCentroids
type	One of "group" or "track".
tag	Animal of interest, when type = "track".
track	Track of interest, when type = "track".
timeslot	The timeslot of interest to plot the centroid location
shape	The shape of the points
size	The size of the points
colour	The colour of the points
fill	The fill of the points

Value

A ggplot with centroid locations

Examples

```
# Import river shapefile
water <- actel::shapeToRaster(shape = paste0(system.file(package = "RSP"), "/River_latlon.shp"),
size = 0.0001, buffer = 0.05)

# Create a transition layer with 8 directions
tl <- actel::transitionLayer(x = water, directions = 8)

# Import example output from actel::explore()
data(input.example)

# Run RSP analysis
rsp.data <- runRSP(input = input.example, t.layer = tl, coord.x = "Longitude", coord.y = "Latitude")

# Run dynamic Brownian Bridge Movement Model (dBBMM) with timeslots:
dbbmm.data <- dynBBMM(input = rsp.data, base.raster = water, UTM = 56, timeframe = 2)

# Get dBBMM areas at group level
areas.group <- getAreas(dbbmm.data, type = "group", breaks = c(0.5, 0.95))

# Obtaing centroid coordinate locations of dBBMM:
df.centroid <- getCentroids(input = dbbmm.data, type = "group", areas = areas.group,
   level = 0.95, group = "G1", UTM = 56)

# Plot group centroid location:
plotAreas(areas.group, base.raster = water, group = "G1", timeslot = 7) +
   addCentroids(input = df.centroid, type = "group", timeslot = 7)
```

addRecaptures *Add recapture locations to an existing plot*

Description

Add recapture locations to an existing plot

Usage

```
addRecaptures(
  Signal,
  shape = 21,
  size = 1.5,
  colour = "white",
  fill = "dodgerblue"
)
```

Arguments

Signal	The signal of the transmitter of interest
shape	The shape of the points
size	The size of the points
colour	The colour of the points
fill	The fill of the points

Value

A ggplot with the recapture locations

addStations *Add receiver stations to an existing plot*

Description

Add receiver stations to an existing plot

Usage

```
addStations(input, shape = 21, size = 1.5, colour = "white", fill = "black")
```

Arguments

input	The output of runRSP or dynBBMM
shape	The shape of the points
size	The size of the points
colour	The colour of the points
fill	The fill of the points

Value

A ggplot with stations

Examples

```
# Import river shapefile
water <- actel::shapeToRaster(shape = paste0(system.file(package = "RSP"), "/River_latlon.shp"),
size = 0.0001, buffer = 0.05)

# Create a transition layer with 8 directions
tl <- actel::transitionLayer(x = water, directions = 8)

# Import example output from actel::explore()
data(input.example)

# Run RSP analysis
rsp.data <- runRSP(input = input.example, t.layer = tl, coord.x = "Longitude", coord.y = "Latitude")

# Run dynamic Brownian Bridge Movement Model (dBBMM)
dbbmm.data <- dynBBMM(input = rsp.data, base.raster = water, UTM = 56)

# Plot example dBBMM with acoustic stations
plotContours(dbbmm.data, tag = "A69-9001-1111", track = 1) + addStations(rsp.data)
```

---

animateTracks *Animate the RSP tracks*

Description

This function can be used to generate an animated plot of the RSP tracks.

Usage

```
animateTracks(
  input,
  base.raster,
  tags = NULL,
```

```
  drop.groups = NULL,
  by.group = FALSE,
  start.time,
  stop.time,
  land.col = "#BABCBF80",
  add.legend = TRUE,
  add.stations = FALSE,
  save.gif = FALSE,
  gif.name = "Animation.gif",
  height = 720,
  width = 720,
  xlim = NULL,
  ylim = NULL,
  nframes = 100,
  fps = 10
)
```

Arguments

input	The output of runRSP.
base.raster	The raster used to generate the transition layer used in runRSP.
tags	Character vector specifying which tags to include in the animation.
drop.groups	Character vector specifying any group(s) to the be removed from the animation.
by.group	Logical, if TRUE one facet will be plotted for each tracked group. Defauly is FALSE.
start.time	Character vector of the start point (format = "Y-m-d H:M:S") for the animation.
stop.time	Character vector of the stop point (format = "Y-m-d H:M:S") for the animation.
land.col	Colour of the land masses. Defaults to semi-transparent grey.
add.legend	Logical, if TRUE (default) a colour legend representing the monitored tags will be included.
add.stations	Logical, if TRUE the stations will be added to the animaltion. Default is FALSE. Only works if by.group = FALSE.
save.gif	Logical defining if the animation should be saved.
gif.name	If save.gif = TRUE, character vector for the GIF name.
height	If save.gif = TRUE, number of pixels for the GIF height.
width	If save.gif = TRUE, number of pixels for the GIF width.
xlim	Numeric vector defining the horizontal limits of the map.
ylim	Numeric vector defining the vertical limits of the map.
nframes	The number of frames to render (default 100).
fps	The framerate of the animation in frames/sec (default 10).

Value

An animation of the RSP tracks.

Examples

```
# Import river shapefile
water <- actel::shapeToRaster(shape = paste0(system.file(package = "RSP"), "/River_latlon.shp"),
size = 0.0001, buffer = 0.05)

# Create a transition layer with 8 directions
tl <- actel::transitionLayer(x = water, directions = 8)

# Import example output from actel::explore()
data(input.example)

# Run RSP analysis
rsp.data <- runRSP(input = input.example, t.layer = tl, coord.x = "Longitude", coord.y = "Latitude")

# Animate and RSP track:
animateTracks(input = rsp.data, base.raster = water, tags = "A69-9001-1111", add.stations = TRUE)
```

`calcRSP` *Recreating RSP for a particular tracked animal*

Description

Estimates the RSP individually for all tracks of a particular animal.

Usage

```
calcRSP(
  df.track,
  tz,
  distance,
  min.time,
  time.step,
  transition,
  er.ad,
  path.list,
  verbose
)
```

Arguments

`df.track`	Detection data for that individual as imported using RSPete.
`tz`	Time zone of the study area.
`distance`	Maximum distance between RSP locations.
`min.time`	Minimum time required between receiver detections (in minutes) for RSP to be calculated. Default to 10 minutes.

time.step	Time lapse in minutes to be considered for consecutive detections at the same station.
transition	TransitionLayer object as returned by LTDpath.
er.ad	Incremental error per additional RSP point.
path.list	A list of previously calculated paths.
verbose	Logical: If TRUE, detailed messages and progression are displayed. Otherwise, a single progress bar is shown.

Value

A dataframe with the RSP estimations for all identified tracks for that animal.

---

dynBBMM *Total dynamic Brownian Bridge Movement Model*

Description

Calculates dynamic Brownian Bridge Movement Model (dBBMM) for each track and transmitter. Tracks shorter than 30 minutes are automatically identified and not included in the analysis.

Usage

```
dynBBMM(
  input,
  base.raster,
  tags = NULL,
  start.time,
  stop.time,
  timeframe = NULL,
  UTM,
  debug = FALSE,
  verbose = TRUE,
  window.size = 7,
  margin = 3
)
```

Arguments

input	The output of runRSP.
base.raster	The water raster of the study area. For example the output of shapeToRaster.
tags	Vector of transmitters to be analysed. By default all transmitters from runRSP will be analysed.
start.time	Sets the start point for analysis (format = "Y-m-d H:M:S").
stop.time	Sets the stop point for analysis (format = "Y-m-d H:M:S").

timeframe	Temporal window size for fine-scale dBBMM in hours. If left NULL, a single dBBMM is calculated for the whole period.
UTM	The UTM zone of the study area. Only relevant if a latlon-to-metric conversion is required.
debug	Logical: If TRUE, the function progress is saved to an RData file.
verbose	Logical: If TRUE, detailed check messages are displayed. Otherwise, only a summary is displayed.
window.size	The size of the moving window along the track. Larger windows provide more stable/accurate estimates of the brownian motion variance but are less well able to capture more frequent changes in behavior. This number has to be odd.
margin	The margin used for the behavioral change point analysis. This number has to be odd.

Value

List of calculated dBBMMs and metadata on each track used for the modelling.

Examples

```
# Import river shapefile
water <- actel::shapeToRaster(shape = paste0(system.file(package = "RSP"), "/River_latlon.shp"),
size = 0.0001, buffer = 0.05)

# Create a transition layer with 8 directions
tl <- actel::transitionLayer(x = water, directions = 8)

# Import example output from actel::explore()
data(input.example)

# Run RSP analysis
rsp.data <- runRSP(input = input.example, t.layer = tl, coord.x = "Longitude", coord.y = "Latitude")

# Run dynamic Brownian Bridge Movement Model (dBBMM)
dbbmm.data <- dynBBMM(input = rsp.data, base.raster = water, UTM = 56)
```

---

getAreas *Calculate water areas per group or track*

Description

Calculate water areas per group or track

Usage

```
getAreas(input, type = c("group", "track"), breaks = c(0.5, 0.95))
```

Arguments

input	The output of dynBBMM
type	one of "group" or "track". If set to "track", UD rasters for each track are also supplied.
breaks	The contours for calculating usage areas in squared metres. By default the 95% and 50% contours are used.

Value

A list of areas per track, per group

Examples

```
# Import river shapefile
water <- actel::shapeToRaster(shape = paste0(system.file(package = "RSP"), "/River_latlon.shp"),
size = 0.0001, buffer = 0.05)

# Create a transition layer with 8 directions
tl <- actel::transitionLayer(x = water, directions = 8)

# Import example output from actel::explore()
data(input.example)

# Run RSP analysis
rsp.data <- runRSP(input = input.example, t.layer = tl, coord.x = "Longitude", coord.y = "Latitude")

# Run dynamic Brownian Bridge Movement Model (dBBMM)
dbbmm.data <- dynBBMM(input = rsp.data, base.raster = water, UTM = 56)

# Get dBBMM areas at group level
areas.group <- getAreas(input = dbbmm.data, type = "group", breaks = c(0.5, 0.95))
```

---

getAreaStep *Calculate dBBMM and overlapping areas in steps*

Description

When a long study period is analysed the dBBMM can crash R since they are computationally heavy. You can use this function to calculate the dBBMMs and overlapping areas (between pairs of groups) according to a fixed step (i.e. timeframe in number of days), and export the output to disk as it goes. If your computer runs out of memory and kills R, you can then resume the calculations by setting a new output name (name.new) and specifying the previous one already stored in disk (name.file), and defining the new start date to resume the calculations (start.time). It currently only works with the 50

Usage

```
getAreaStep(
  input,
  base.raster,
  UTM,
  timeframe = 1,
  start.time = NULL,
  save = TRUE,
  name.new = NULL,
  name.file = NULL,
  groups = NULL
)
```

Arguments

input	The output of runRSP.
base.raster	The water raster of the study area. For example the output of shapeToRaster.
UTM	The UTM zone of the study area. Only relevant if a latlon-to-metric conversion is required.
timeframe	The intended temporal interval of interest (in number of days) to perform the calculations. Default is 1 day.
start.time	Character vector identifying the initial date (format = "Y-m-d") to start the calculations.
save	Logical (default is TRUE). Do you want to save the calculated areas to disk?
name.new	File name (character) to save calculations output to disk.
name.file	File name (character) of previous calculations output to be imported (if any).
groups	Vector of group names (character) of interest to perform calculations.

Value

A dataframe of dBBMM areas per group and the corresponding overlaps

Examples

```
# Import river shapefile
water <- actel::shapeToRaster(shape = paste0(system.file(package = "RSP"), "/River_latlon.shp"),
size = 0.0001, buffer = 0.05)

# Create a transition layer with 8 directions
tl <- actel::transitionLayer(x = water, directions = 8)

# Import example output from actel::explore()
data(input.example)

# Run RSP analysis
rsp.data <- runRSP(input = input.example, t.layer = tl, coord.x = "Longitude", coord.y = "Latitude")
```

```
# Calculate dBBMM and overlaps in steps:
df.areas <- getAreaStep(input = rsp.data, base.raster = water, UTM = 56, timeframe = 1,
 name.new = "save.csv", groups = c("G1", "G2"))
```

---

getCentroids *Get centroid locations of dBBMM*

Description

When a timeslot dBBMM analysis is conducted, this function can be used to obtain centroid latitude and longitude locations between all utilization distribution contours at group or track level.

Usage

```
getCentroids(input, areas, type, level, group, UTM)
```

Arguments

input	The output of dynBBMM.
areas	The output of getAreas.
type	Character vector specifying the type of getAreas analysis performed: "group" or "track".
level	Numeric vector defining the contour level of dBBMM of interest to extract the centroid positions.
group	Character vector defining the group of interest for the analysis, when getAreas is of type "group".
UTM	Numeric vector representing the UTM zone of the study area.

Value

A dataframe containing the centroid positions per each timeslot

Examples

```
# Import river shapefile
water <- actel::shapeToRaster(shape = paste0(system.file(package = "RSP"), "/River_latlon.shp"),
size = 0.0001, buffer = 0.05)

# Create a transition layer with 8 directions
tl <- actel::transitionLayer(x = water, directions = 8)

# Import example output from actel::explore()
data(input.example)

# Run RSP analysis
```

```
rsp.data <- runRSP(input = input.example, t.layer = tl, coord.x = "Longitude", coord.y = "Latitude")

# Run dynamic Brownian Bridge Movement Model (dBBMM) with timeslots:
dbbmm.data <- dynBBMM(input = rsp.data, base.raster = water, UTM = 56, timeframe = 2)

# Get dBBMM areas at group level
areas.group <- getAreas(dbbmm.data, type = "group", breaks = c(0.5, 0.95))

# Obtaing centroid coordinate locations of dBBMM:
df.centroid <- getCentroids(input = dbbmm.data, areas = areas.group, type = "group",
   level = 0.95, group = "G1", UTM = 56)
```

`getDistances` *Get total distances travelled*

Description

Obtain the total distances travelled (in metres) for the tracked animals, using only the receiver locations and also adding the RSP positions.

Usage

```
getDistances(input, t.layer)
```

Arguments

`input`	RSP dataset as returned by RSP.
`t.layer`	A transition layer. Can be calculated using the function `transitionLayer`.

Value

A dataframe containing the total distances travelled during each RSP track.

Examples

```
# Import river shapefile
water <- actel::shapeToRaster(shape = paste0(system.file(package = "RSP"), "/River_latlon.shp"),
size = 0.0001, buffer = 0.05)

# Create a transition layer with 8 directions
tl <- actel::transitionLayer(x = water, directions = 8)

# Import example output from actel::explore()
data(input.example)

# Run RSP analysis
rsp.data <- runRSP(input = input.example, t.layer = tl, coord.x = "Longitude", coord.y = "Latitude")
```

```
# Calculate distances travelled
distance.data <- getDistances(rsp.data, t.layer = tl)
```

`getDistPoint` *Calculate in-water distances from RSP locations to a point of reference*

Description

Calculate in-water distances from RSP locations to a point of reference

Usage

```
getDistPoint(input, point, t.layer, transmitter = NULL)
```

Arguments

`input`	The output of `runRSP`
`point`	Point of reference (lon, lat) to where distances will be calculated.
`t.layer`	A transition layer. Can be calculated using the function `transitionLayer`.
`transmitter`	The animal(s) of interest for calculating distances. If not specified, by default, distances will be calculated for all animals in the dataset.

Value

The RSP detections object with a distance (in metres) column appended.

Examples

```
# Import river shapefile
water <- actel::shapeToRaster(shape = paste0(system.file(package = "RSP"), "/River_latlon.shp"),
size = 0.0001, buffer = 0.05)

# Create a transition layer with 8 directions
tl <- actel::transitionLayer(x = water, directions = 8)

# Import example output from actel::explore()
data(input.example)

# Run RSP analysis
rsp.data <- runRSP(input = input.example, t.layer = tl, coord.x = "Longitude", coord.y = "Latitude")

# Calculate distances to a point of reference
df.dist <- getDistPoint(input = rsp.data, point = c(151.0291, -33.81771), t.layer = tl)
```

getOverlaps *Calculate overlaps between different groups*

Description

Calculate overlaps between different groups

Usage

```
getOverlaps(input)
```

Arguments

input	The output of getAreas

Value

A list of Overlaps (per timeslot if relevant), as well as the respective overlap rasters.

Examples

```
# Import river shapefile
water <- actel::shapeToRaster(shape = paste0(system.file(package = "RSP"), "/River_latlon.shp"),
size = 0.0001, buffer = 0.05)

# Create a transition layer with 8 directions
tl <- actel::transitionLayer(x = water, directions = 8)

# Import example output from actel::explore()
data(input.example)

# Run RSP analysis
rsp.data <- runRSP(input = input.example, t.layer = tl, coord.x = "Longitude", coord.y = "Latitude")

# Run dynamic Brownian Bridge Movement Model (dBBMM)
dbbmm.data <- dynBBMM(input = rsp.data, base.raster = water, UTM = 56)

# Get dBBMM areas at group level
areas.group <- getAreas(dbbmm.data, type = "group", breaks = c(0.5, 0.95))

# Get overlaps between groups
overlap.data <- getOverlaps(areas.group)
```

includeRSP *Recreating RSP for all tracked animals*

Description

Automatically estimates the RSP for all tracked individuals within a particular study area.

Usage

```
includeRSP(
  detections,
  transition,
  tz,
  distance,
  time.step,
  er.ad,
  min.time,
  max.time,
  verbose,
  debug = FALSE,
  recaptures
)
```

Arguments

detections	Detection data for that individual as imported using RSPete.
transition	TransitionLayer object as returned by LTDpath.
tz	Timezone of the study area.
distance	Distance (in metres) by which RSP point should be spaced (between detections at different stations). Defaults to 250 metres.
time.step	Time lapse (in minutes) between RSP points added between detections at the same station. Defaults to 10 minutes. Must not be larger than min.time.
er.ad	Increment rate of the position errors for the estimated locations (in metres). If left unset, defaults to 5% of the distance argument.
min.time	Minimum time required between receiver detections (in minutes) for RSP to be calculated. Default to 10 minutes.
max.time	Maximum time allowed between receiver detections (in hours) for RSP to be calculated. Defaults to 24 hours.
verbose	Logical: If TRUE, detailed messages and progression are displayed. Otherwise, a single progress bar is shown.
debug	Logical: If TRUE, the function progress is saved to an RData file.
recaptures	If the recapture locations will be included in the analysis.

Value

A list with the RSP estimations of individual tracks per transmitter.

nameTracks *Identify potential fine-scale data for analysis*

Description

Identifies fine-scale data among total detection dataset to be used for RSP estimation. Tracks are then named based on the interval between consecutive detection dates.

Usage

```
nameTracks(detections, max.time = 24, recaptures, tz)
```

Arguments

detections	Detections data frame
max.time	Temporal lag in hours to be considered for the fine-scale tracking. Default is to consider 1-day intervals.
recaptures	If the recapture locations will be included in the analysis.
tz	Time zone of the study area.

Value

A dataframe with identified and named individual tracks for RSP estimation.

plotAreas *Plot areas*

Description

Plot areas for a specific group and, if relevant, track and timeslot. If the base raster is in a geographic coordinate system, plotAreas will attempt to convert the dbbmm results to that same geographic system, so everything falls in place.

Usage

```
plotAreas(
  areas,
  base.raster,
  group,
  timeslot,
  title = NULL,
  col,
  land.col = "#BABCBF80"
)
```

Arguments

areas	The areas object used to calculate the space use areas at group level.
base.raster	The raster used in the dbbmm calculations.
group	Character vector indicating the group to be displayed.
timeslot	The timeslot to be displayed. Only relevant for timeslot dbbmms.
title	Plot title.
col	Character vector of colours to be used in the plot (same length as the number of contour levels).
land.col	Colour of the land masses. Defaults to semi-transparent grey.

Value

A plot of the overlapping areas between two groups.

Examples

```
# Import river shapefile
water <- actel::shapeToRaster(shape = paste0(system.file(package = "RSP"), "/River_latlon.shp"),
size = 0.0001, buffer = 0.05)

# Create a transition layer with 8 directions
tl <- actel::transitionLayer(x = water, directions = 8)

# Import example output from actel::explore()
data(input.example)

# Run RSP analysis
rsp.data <- runRSP(input = input.example, t.layer = tl, coord.x = "Longitude", coord.y = "Latitude")

# Run dynamic Brownian Bridge Movement Model (dBBMM)
dbbmm.data <- dynBBMM(input = rsp.data, base.raster = water, UTM = 56)

# Get dBBMM areas at group level
areas.group <- getAreas(dbbmm.data, type = "group", breaks = c(0.5, 0.95))

# Plot areas at group level
plotAreas(areas.group, group = "G1", base.raster = water)
```

---

plotContours *Plot dynamic Brownian Bridge Movement Model (dBBMM) contours*

Description

Plot dynamic Brownian Bridge Movement Model (dBBMM) contours

Usage

```
plotContours(
  input,
  tag,
  track = NULL,
  timeslot,
  scale.type = "categorical",
  breaks = c(0.95, 0.75, 0.5, 0.25),
  col,
  title,
  land.col = "#BABCBF80"
)
```

Arguments

input	The dbbmm object as returned by dynBBMM.
tag	Choose a single tag to plot
track	If a single tag was chosen, you can use 'track' to define a specific track to be plotted.
timeslot	The timeslot to be plotted. Only relevant for timeslot dbbmms.
scale.type	Character vector selecting the type of scale to plot space use areas. By default a "categorical" scale is set, but alternatively can be set to "continuous" to return the space use areas with a continuous scale.
breaks	When scale.type = "categorical", this is a numeric vector selecting the use areas to plot. By default, the 99%, 95%, 75%, 50% and 25% areas will be returned.
col	The colours to be used when scale.type = "categorical". Must match the number of breaks.
title	The title of the plot.
land.col	Colour of the land mass.

Value

dynamic Brownian Bridge Movement Model plot.

Examples

```
# Import river shapefile
water <- actel::shapeToRaster(shape = paste0(system.file(package = "RSP"), "/River_latlon.shp"),
size = 0.0001, buffer = 0.05)

# Create a transition layer with 8 directions
tl <- actel::transitionLayer(x = water, directions = 8)

# Import example output from actel::explore()
data(input.example)

# Run RSP analysis
```

```
rsp.data <- runRSP(input = input.example, t.layer = tl, coord.x = "Longitude", coord.y = "Latitude")

# Run dynamic Brownian Bridge Movement Model (dBBMM)
dbbmm.data <- dynBBMM(input = rsp.data, base.raster = water, UTM = 56)

# Plot example dBBMM
plotContours(dbbmm.data, tag = "A69-9001-1111", track = 1)
```

`plotDensities` *Density plot of elapsed times between consecutive acoustic detections*

Description

Generates a density plot for inspecting the distribution of elapsed times (in hours) between all consecutive acustic detections. By default the plot is created including all monitored groups and transmitters. Alternatively, can be set to be performed at group level using the type argument.

Usage

```
plotDensities(input, group)
```

Arguments

`input`	RSP dataset as returned by RSP.
`group`	Character vector defining the group to which calculate density distributions. By default, density is calculated for all animals and groups tracked.

Value

Density plots of hours elapsed between consecutive acoustic detections.

Examples

```
# Import river shapefile
water <- actel::shapeToRaster(shape = paste0(system.file(package = "RSP"), "/River_latlon.shp"),
size = 0.0001, buffer = 0.05)

# Create a transition layer with 8 directions
tl <- actel::transitionLayer(x = water, directions = 8)

# Import example output from actel::explore()
data(input.example)

# Run RSP analysis
rsp.data <- runRSP(input = input.example, t.layer = tl, coord.x = "Longitude", coord.y = "Latitude")

# Plot distribution of acoustic detections
```

```
plotDensities(rsp.data, group = "G1")
```

---

plotDistances *Plot total distances travelled*

Description

Compare the outputs of total distances travelled (in kilometres) for the tracked animals, using only the receiver locations and adding the RSP positions. Data on the total distances travelled are stored in the 'distances' objtect.

Usage

```
plotDistances(input, group, compare = TRUE)
```

Arguments

input	output of getDistances.
group	Define a specific group to be plotted, rather than the overall results.
compare	By default, a comparative plot is returned showing distances travelled with Receiver and RSP location types. If FALSE, only the RSP total distances travelled will be returned.

Value

A barplot of total distances travelled as a function of location type (Loc.type) and the distances travelled during each RSP track.

Examples

```
# Import river shapefile
water <- actel::shapeToRaster(shape = paste0(system.file(package = "RSP"), "/River_latlon.shp"),
size = 0.0001, buffer = 0.05)

# Create a transition layer with 8 directions
tl <- actel::transitionLayer(x = water, directions = 8)

# Import example output from actel::explore()
data(input.example)

# Run RSP analysis
rsp.data <- runRSP(input = input.example, t.layer = tl, coord.x = "Longitude", coord.y = "Latitude")

# Calculate distances travelled
distance.data <- getDistances(rsp.data, t.layer = tl)

# Plot distances travelled
```

```
plotDistances(distance.data, group = "G1")
```

`plotOverlaps` *Plot overlapping contours*

Description

Plot specific dBBMM overlapping areas for a specific combination of groups and, if relevant, a specific timeslot. If the base raster is in a geographic coordinate system, plotOverlaps will attempt to convert the dbbmm results to that same geographic system, so everything falls in place.

Usage

```
plotOverlaps(
  overlaps,
  areas,
  base.raster,
  groups,
  timeslot,
  level,
  title = NULL,
  col,
  land.col = "#BABCBF80"
)
```

Arguments

`overlaps`	An overlap object as returned by `getOverlaps`.
`areas`	The areas object used to calculate the overlaps.
`base.raster`	The raster used in the dbbmm calculations.
`groups`	Character vector indicating the two groups to be displayed.
`timeslot`	The timeslot to be displayed. Only relevant for timeslot dbbmms.
`level`	Value of the use area to plot. Must match one the levels calculated in the overlaps.
`title`	Plot title. By default, the names of the groups being compared are displayed.
`col`	Character vector of three colours to be used in the plot (one for each group and one for the overlap).
`land.col`	Colour of the land masses. Defaults to semi-transparent grey.

Details

If one of your groups has more than one usage area, or an overlaps contour has more than one area (both potentially caused by having multiple tags/tracks in a single group), ggplot2 will issue the following warning when plotting the map: Warning message: Raster pixels are placed at uneven horizontal intervals and will be shifted. Consider using geom_tile() instead. This is simply because empty cells are cleared out to improve plotting efficiency, which means there will be an empty space between the multiple areas to be drawn. Please be aware that this has no effect on the plot itself.

Value

A plot of the overlapping areas between two groups.

Examples

```
# Import river shapefile
water <- actel::shapeToRaster(shape = paste0(system.file(package = "RSP"), "/River_latlon.shp"),
size = 0.0001, buffer = 0.05)

# Create a transition layer with 8 directions
tl <- actel::transitionLayer(x = water, directions = 8)

# Import example output from actel::explore()
data(input.example)

# Run RSP analysis
rsp.data <- runRSP(input = input.example, t.layer = tl, coord.x = "Longitude", coord.y = "Latitude")

# Run dynamic Brownian Bridge Movement Model (dBBMM)
dbbmm.data <- dynBBMM(input = rsp.data, base.raster = water, UTM = 56)

# Get dBBMM areas at group level
areas.group <- getAreas(dbbmm.data, type = "group", breaks = c(0.5, 0.95))

# Get overlaps between groups
overlap.data <- getOverlaps(areas.group)

# Plot overlaps
plotOverlaps(overlaps = overlap.data, areas = areas.group, base.raster = water,
 groups = c("G1", "G2"), level = 0.95)
```

---

plotRaster *Check input data quality for the RSP analysis*

Description

If you are reading this it's because RSP failed to detect all of your receivers within the base raster provided, or any of your receiver location was found to be in land. This function allows you to visually identify the station(s) with problem. Please either extend your raster to include all stations or fix receiver locations to be in-water.

Usage

```
plotRaster(
  input,
  base.raster,
  coord.x,
  coord.y,
  size = 1,
  land.col = "#BABCBF80"
)
```

Arguments

input	Either a data frame containing the coordinates of the stations or the output of one of actel's main functions (explore, migration or residency).
base.raster	Raster object. Imported for example using shapeToRaster.
coord.x, coord.y	The names of the columns containing the x and y positions of the stations in the spatial object.
size	The size of the station dots
land.col	Colour of the land masses. Defaults to semi-transparent grey.

Value

A plot of your base raster extent and the receiver locations.

Examples

```
# Import river shapefile
water <- actel::shapeToRaster(shape = paste0(system.file(package = "RSP"), "/River_latlon.shp"),
size = 0.0001, buffer = 0.05)

# Create a transition layer with 8 directions
tl <- actel::transitionLayer(x = water, directions = 8)

# Import example output from actel::explore()
data(input.example)

# Plot raster and acoustic stations
plotRaster(input.example, base.raster = water, coord.x = "Longitude",
 coord.y = "Latitude", size = 1)
```

plotTracks *Plot the RSP tracks*

Description

This function can be used to plot a map of a particular RSP track of interest.

Usage

```
plotTracks(
  input,
  base.raster,
  type = c("both", "points", "lines"),
  group,
  tag,
  track,
  size = c(0.33, 0.3),
  alpha = c(0.5, 0.5),
  land.col = "#BABCBF80"
)
```

Arguments

input	The output of runRSP.
base.raster	The raster used to generate the transition layer used in runRSP
type	One of "points", "line" or "both". Defaults to "both", i.e. both lines and points are plotted for the generated tracks.
group	Choose a single group of fish to plot
tag	Choose a single tag to plot
track	If a single tag was chosen, you can use 'track' to define a specific track to be plotted.
size	The size/width of the points and lines to be plotted. if type = "both", the line size will be the one specified and the point size will be 10% larger than the specified.
alpha	One or two transparency values (for points and lines, respectively). For no transparency, alpha = 1.
land.col	Colour of the land masses. Defaults to semi-transparent grey.

Value

A plot showing the RSP track locations.

Examples

```
# Import river shapefile
water <- actel::shapeToRaster(shape = paste0(system.file(package = "RSP"), "/River_latlon.shp"),
size = 0.0001, buffer = 0.05)

# Create a transition layer with 8 directions
tl <- actel::transitionLayer(x = water, directions = 8)

# Import example output from actel::explore()
data(input.example)

# Run RSP analysis
rsp.data <- runRSP(input = input.example, t.layer = tl, coord.x = "Longitude", coord.y = "Latitude")

# Plot a specific RSP track
plotTracks(rsp.data, base.raster = water, tag = "A69-9001-1111", track = "Track_1")
```

---

prepareDetections *Prepare detection data for RSP calculations*

Description

Open and sort the detections dataset for applying RSP estimation, using the tagging data to assign species names and indexes for each tracked animal.

Usage

```
prepareDetections(detections, spatial, coord.x, coord.y)
```

Arguments

`detections`	A list of detections provided by an actel function.
`spatial`	A list of spatial objects in the study area
`coord.x, coord.y`	The names of the columns containing the x and y positions of the stations in the spatial object.

Value

A standardised data frame to be used for RSP calculation.

runRSP *Calculate refined shortest paths between detections*

Description

Estimates the RSP for a series of animals tracked with acoustic transmitters. Intermediate locations between consecutive acoustic detections (either on the same or different receivers) are estimated according to the defined `distance` and `time.step` arguments. The error of estimated locations increase proportionally as the animal moves away from the first detection, and decreases as it approaches the second detection (argument `er.ad`). If the animal is not detected for a long time (default is a daily absence), the detections are broken into a new track (argument `max.time`).

Usage

```
runRSP(
  input,
  t.layer,
  coord.x,
  coord.y,
  distance = 250,
  tags = NULL,
  recaptures = FALSE,
  time.step = 10,
  min.time = 10,
  max.time = 24,
  er.ad,
  verbose = FALSE,
  debug = FALSE
)
```

Arguments

`input`	The output of one of `actel`'s main functions (`explore`, `migration` or `residency`)
`t.layer`	A transition layer. Can be calculated using the function `transitionLayer`.
`coord.x, coord.y`	The names of the columns containing the x and y positions of the stations in the spatial object.
`distance`	Distance (in metres) by which RSP point should be spaced (between detections at different stations). Defaults to 250 metres.
`tags`	Vector of transmitters for which to calculate RSP. By default all transmitters will be analysed.
`recaptures`	Logical: if TRUE, a recapture.csv dataset containing the recapture locations of tracked animals will be included in the analysis.
`time.step`	Time lapse (in minutes) between RSP points added between detections at the same station. Defaults to 10 minutes. Must not be larger than `min.time`.

min.time	Minimum time required between receiver detections (in minutes) for RSP to be calculated. Default to 10 minutes.
max.time	Maximum time allowed between receiver detections (in hours) for RSP to be calculated. Defaults to 24 hours.
er.ad	Increment rate of the position errors for the estimated locations (in metres). If left unset, defaults to 5% of the distance argument.
verbose	Logical: If TRUE, detailed messages and progression are displayed. Otherwise, a single progress bar is shown.
debug	Logical: If TRUE, the function progress is saved to an RData file.

Value

Returns a list of RSP tracks for each transmitter detected, as well as auxiliary information.

Examples

```
# Import river shapefile
water <- actel::shapeToRaster(shape = paste0(system.file(package = "RSP"), "/River_latlon.shp"),
size = 0.0001, buffer = 0.05)

# Create a transition layer with 8 directions
tl <- actel::transitionLayer(x = water, directions = 8)

# Import example output from actel::explore()
data(input.example)

# Run RSP analysis
rsp.data <- runRSP(input = input.example, t.layer = tl, coord.x = "Longitude", coord.y = "Latitude")
```

---

suggestSize *Suggest plot dimensions for a given raster*

Description

Suggest plot dimensions for a given raster

Usage

```
suggestSize(input, max)
```

Arguments

input	The raster being plotted
max	the desired size for the longest edge

Value

A width/height vector (rounded)

Examples

```
# Import river shapefile
water <- actel::shapeToRaster(shape = paste0(system.file(package = "RSP"), "/River_latlon.shp"),
size = 0.0001, buffer = 0.05)

# Find suggested size to save projected map
suggestSize(water, max = 10)
```

Index

actel, *24*, *27*
addCentroids, 2
addRecaptures, 4
addStations, 4
animateTracks, 5

calcRSP, 7

dynBBMM, *5*, 8, *10*, *12*, *19*

explore, *24*, *27*

getAreas, 9, *12*, *15*
getAreaStep, 10
getCentroids, *3*, 12
getDistances, 13, *21*
getDistPoint, 14
getOverlaps, 15, *22*

includeRSP, 16

migration, *24*, *27*

nameTracks, 17

plotAreas, 17
plotContours, 18
plotDensities, 20
plotDistances, 21
plotOverlaps, 22
plotRaster, 23
plotTracks, 25
prepareDetections, 26

residency, *24*, *27*
runRSP, *5*, *11*, *14*, 27

shapeToRaster, *8*, *11*, *24*
suggestSize, 28

transitionLayer, *13, 14*, *27*