Package 'anipaths'

February 28, 2020

Type Package

Title Animation of Observed Trajectories Using Spline-Based Interpolation

Version 0.9.8

Date 2020-02-27

Author Henry Scharf

Maintainer Henry Scharf <hscharf@sdsu.edu>

Description Animation of observed trajectories using spline-based interpolation (see for example, Buderman, F. E., Hooten, M. B., Ivan, J. S. and Shenk, T. M. (2016), <doi:10.1111/2041-210X.12465> ``A functional model for characterizing long-distance movement behaviour''. Methods Ecol Evol). Intended to be used exploratory data analysis, and perhaps for preparation of presentations.

License GPL-3

RoxygenNote 7.0.2

Depends R (>= 2.10)

Imports animation, RColorBrewer, scales, sp, raster, mgcv, grDevices, ggmap

Suggests ellipse, igraph, knitr

VignetteBuilder knitr

LazyData true

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2020-02-28 17:20:10 UTC

R topics documented:

animate_paths 2
get.network.colors 6
plot.paths_animation 7
vultures 8

Index **9**

`animate_paths` *animate paths*

Description

Animates telemetry data for the purposed of EDA using smoothing splines to interpolate the observed locations. The animations are particularly useful when examining multiple simultaneous trajectories. The output of the call to `animate_paths()` should bring up a browser window that shows the animation. Additionally, the images generated in `images/` (or else the value set for `imgdir`) may be used with ffmpeg, latex, or other presentation software that can build animations directly from a sequence of images.

Usage

```
animate_paths(
  paths,
  times = NULL,
  delta.t = NULL,
  n.frames = NULL,
  interval = 1/12,
  paths.proj = "+proj=longlat",
  coord = c("x", "y"),
  Time.name = "time",
  ID.name = NULL,
  whole.path = FALSE,
  covariate = NULL,
  covariate.colors = c("black", "white"),
  covariate.thresh = NULL,
  covariate.legend.loc = "bottomright",
  par.opts = list(),
  dev.opts = list(),
  background = NULL,
  bg.axes = TRUE,
  bg.opts = NULL,
  bg.misc = NULL,
  method = "html",
  pt.cex = 1,
  pt.colors = NULL,
  dimmed = NULL,
  res = 1.5,
  plot.date = TRUE,
  date.col = "black",
  legend.loc = "topright",
  network = NULL,
  network.times = NULL,
  network.thresh = 0.5,
```

```
  network.colors = NULL,
  network.ring.wt = 3,
  network.ring.trans = 1,
  network.segment.wt = 3,
  network.segment.trans = 0.5,
  tail.wd = 1,
  tail.length = 5,
  tail.colors = "gray87",
  xlim = NULL,
  ylim = NULL,
  main = NULL,
  bs = "'tp', fx=T",
  max.knots = NULL,
  uncertainty.level = NA,
  override = FALSE,
  return.paths = FALSE,
  ...
)
```

Arguments

paths	Either a data.frame with longitudes/eastings, latitudes/northings, IDs, and times (see coord, ID.name, and Time.name), a SpatialPointsDataFrame with IDs and times, or a list of data.frames containing the longitudes, latitudes, and times for each individual (with names provided). If all paths are already synchornous, another option for passing the data is to define paths as a list of matrices, all with the same number of rows, and to specify the times separately via the next argument. This situation might arise when, for example, locations the user wishes to animated correspond to realizations/sampler from a discrete-time movement model. Covariates may be provided as named columns of the matrices in paths.
times	If all paths are already synchornous, another option for passing the data is to define paths as a list of matrices, all with the same number of rows, and to specify the times separately via this argument.
delta.t	The gap in time between each frame in the animation. Specify one of delta.t or n.frames. If both are specified, delta.t is used.
n.frames	The number of frames used to animate the complete time domain of the data.
interval	Seconds per frame in animation. Default is 1/12 (or 12 frames per second).
paths.proj	PROJ.4 string corresponding to the projection of the data. Default is "+proj=longlat".
coord	A character vector of length 2 giving the names of the longitude/easting and latitude/northing columns in the paths data.frame (in that order). This is required if paths is not a SpatialPointsDataFrame.
Time.name	The name of the columns in paths gving the observation times. This column must be of class POSIXt, or numeric.
ID.name	The name of the column in paths that identifies each individual. If left as NULL (default), a single individual is assumed.

whole.path — logical. If TRUE (default = FALSE), the complete interpolated trajectories will be plotted in the background of the animation. If whole.path = TRUE, consider also setting tail.length = 0.

covariate — The name of the column in paths that identifies the covariate to be mapped to a ring of color around each point.

covariate.colors — vector of colors which will be used in their given order to make a color ramp (see colorRamp())

covariate.thresh — if changed from its default value of NULL, the interpolated value of the covariate will be binarized based on this numeric value.

covariate.legend.loc — either the location of the covariate legend, or NA if no legend is desired

par.opts — Options passed to par() before creating each frame.

dev.opts — Options passed to png() before creating each frame.

background — Three possibilities: (1) A single background image over which animation will be overlayed, or a list/stack of images/rasters corresponding to each frame. (2) A list with values center (long/lat), zoom, and maptype (see ggmap::get_googlemap()) which will be used to generate a background for the animation based on Google maps tiles. Additional arguments may be added which will be passed to ggmap::get_googlemap(). (3) A logical value of TRUE, which will cue the function to get the best Google Map tile combination it can come up with. Note: ggmap must be installed for (2) and (3). Note: if you are calling animate_paths() several times in a short period of time you may get an error from Google for trying to pull tiles too often (e.g., Error in download.file(url,destfile = tmp,quiet = !messaging,mode = "wb") : cannot open URL 'http://maps.googleapis...'). Waiting a minute or so usually solves this.

bg.axes — logical: should animation place axis labels when using a background image (default is TRUE). If RGoogleMaps is used to produce background, labels will be "northing" and "easting". Otherwise, the strings given to coord will be used.

bg.opts — Options passed to plot() function call that makes background in each frame. For example, this could be used to specify blue ocean and gray landcover if background is a SpatialPolygonsDataFrame and bg.opts = list(bg = "dodgerblue4",col = "gray",border = "gray").

bg.misc — Character string which will be executed as R code after generating the background, and before adding trajectories, etc.

method — either "html" (default) or "mp4". The latter requires the user has installed ffmpeg (see ?animation::saveVideo()).

pt.cex — A numeric value giving the character expansion (size) of the points for each individual. Default is 1.

pt.colors — A vector of colors to be used for each individual in the animation. Default values come from Color Brewer palettes. When a network is provided, this is ignored and individuals are all colored black. If NA, no plot colors are chosen to distinguish individuals. This can be useful when making animations involving a covariate. Consider also setting legend.loc to NA in this case.

dimmed	Numeric vector of individuals to "dim" in the animation. Order corresponds to the order of the ID.name variable, or order of paths list.
res	Resolution of images in animation. Increase this for higher quality (and larger) images.
plot.date	Logical variable toggling date text at the time center of the animation.
date.col	default is "black"
legend.loc	passed to first argument of legend() function. Default is "topright". NA removes legend.
network	Array of dimensions (# individuals, # individuals, n.frames) that gives a dyanmic network structure among the individuals.
network.times	Numeric vector. If network time grid doesn't match n.frames, supply the times at which the network has been evaluated so it can be interpolated using smoothing splines.
network.thresh	Network structure is summarized in the animation in a binary way, regardless of whether or not the network is continuously weighted or not. The value of network.thresh determines the level below which no connection is shown, and above which an active connection is shown via colored rings and connecting segments.
network.colors	A symmetric matrix of dimension length(paths) $\times$ length(paths) giving the colors associated with each pairwise relationship.
network.ring.wt	thickness of network rings (default is 3)
network.ring.trans	transparency of network segments (default is 1)
network.segment.wt	thickness of network segments (default is 3)
network.segment.trans	transparency of network segments (default is 0.5)
tail.wd	Thickness of tail trailing behind each individual. Default is 1.
tail.length	Length of the tail trailing each individual.
tail.colors	default is "gray87". Can be single color or vector of colors.
xlim	Boundaries for plotting. If left undefined, the range of the data will be used.
ylim	Boundaries for plotting. If left undefined, the range of the data will be used.
main	Title for each frame. SOON: support for changing titles to allow for, say, dates.
bs	default is "'tp'" (thin plate splines), but this can be any spline basis supported by s() in the mgcv package.
max.knots	maximum number of allowed knots. This actual number of knots used in the fitting will be min(max.knots,#observations_i).
uncertainty.level	value in (0, 1) corresponding to level at which to draw uncertainty ellipses. NA (default) results in no ellipses.
override	Logical variable toggling where or not to override warnings about how long the animation procedure will take.

return.paths	logical. Default is FALSE, but if TRUE then the interpolated paths are returned and no animation is produced.
...	other arguments to be passed to ani.options to animation options such as the time interval between image frames.

Value

video file, possibly a directory containing the individual images, or interpolated paths.

Examples

```
##
vultures$POSIX <- as.POSIXct(vultures$timestamp, tz = "UTC")
vultures_paths <- vultures[vultures$POSIX > as.POSIXct("2009-03-01", origin = "1970-01-01") &
                     vultures$POSIX < as.POSIXct("2009-05-01", origin = "1970-01-01"), ]
animate_paths(paths = vultures_paths,
              delta.t = "week",
              coord = c("location.long", "location.lat"),
              Time.name = "POSIX",
              ID.name = "individual.local.identifier")
system("rm -r js; rm -r css; rm -r images; rm index.html")

background <- list(center = c(-90, 10),
                   zoom = 3,
                   maptype = "satellite")
COVARIATE <- cos(as.numeric(vultures_paths$timestamp) /
                   diff(range(as.numeric(vultures_paths$timestamp))) * 4 * pi)
animate_paths(paths = cbind(vultures_paths, COVARIATE),
              delta.t = "week",
              coord = c("location.long", "location.lat"),
              Time.name = "POSIX", covariate = "COVARIATE",
              covariate.colors = RColorBrewer::brewer.pal(n = 9, "RdYlGn"),
              ID.name = "individual.local.identifier",
              background = background)
```

---

get.network.colors *get.network.colors() Finds all maximal cliques in the network at each time point and tries to assign them a useful coloring*

Description

get.network.colors() Finds all maximal cliques in the network at each time point and tries to assign them a useful coloring

Usage

```
get.network.colors(binary.network, network.color.options = NULL)
```

Arguments

binary.network	a 3D array giving the time-varying adjecency matrix of a dynamic network.
network.color.options	vector of colors

Value

a list of two elements: a list of the maximal cliques at each time, and c list with colors for each clique at each time

---

plot.paths_animation *Plot animation path interpolation*

Description

This is mainly intended as a way to check that the interpolations used in the animation are working as expected.

Usage

```
## S3 method for class 'paths_animation'
plot(x, ..., i = 1, level = 0.05, ylim_x = NULL, ylim_y = NULL)
```

Arguments

x	paths_animation object as created through a call to animate_paths().
...	additional arguments passed to plot.
i	index of individual to plot (corresponds to index in unique(paths[,'ID.name')).
level	confidence level for error bands. NA removes bands.
ylim_x	y-axis limits for marginal plots (x, easting, etc.)
ylim_y	y-axis limits for marginal plots (y, northing, etc.)

Examples

```
vultures$POSIX <- as.POSIXct(vultures$timestamp, tz = "UTC")
vultures_paths <- vultures[vultures$POSIX > as.POSIXct("2009-03-22", origin = "1970-01-01") &
                      vultures$POSIX < as.POSIXct("2009-04-05", origin = "1970-01-01"), ]
interpolated_paths <-
animate_paths(paths = vultures_paths,
              delta.t = 3600*6,
              coord = c("location.long", "location.lat"),
              Time.name = "POSIX",
              ID.name = "individual.local.identifier",
              max.knots = 13,
              return.paths = TRUE)
interpolated_paths_gp <-
```

```
animate_paths(paths = vultures_paths,
              delta.t = 3600*6,
              coord = c("location.long", "location.lat"),
              Time.name = "POSIX",
              ID.name = "individual.local.identifier",
              max.knots = 3*13,
              return.paths = TRUE)
plot(interpolated_paths, i = 2)
plot(interpolated_paths_gp, i = 2, level = 0.01)
```

`vultures` *GPS locations of turkey vultures.*

Description

A dataset containing a subset of the locations of turkey vultures (2003–2006), with time stamps, from:

Usage

```
vultures
```

Format

A data frame with 215719 rows and 11 variables:

timestamp time of observation

location.long logitude

location.lat latitude

individual.local.identifier identifier for each individual ...

Details

Dodge S, Bohrer G, Bildstein K, Davidson SC, Weinzierl R, Mechard MJ, Barber D, Kays R, Brandes D, Han J (2014) Environmental drivers of variability in the movement ecology of turkey vultures (Cathartes aura) in North and South America. Philosophical Transactions of the Royal Society B 20130195. doi:10.1098/rstb.2013.0195

Bildstein K, Barber D, Bechard MJ (2014) Data from: Environmental drivers of variability in the movement ecology of turkey vultures (Cathartes aura) in North and South America. Movebank Data Repository. doi:10.5441/001/1.46ft1k05

Source

https://www.datarepository.movebank.org/handle/10255/move.362/ Bildstein K, Barber D, Bechard MJ (2014) Data from: Environmental drivers of variability in the movement ecology of turkey vultures (Cathartes aura) in North and South America. Movebank Data Repository. doi:10.5441/001/1.46ft1k05

Index

∗Topic **datasets**
vultures, 8

animate_paths, 2

get.network.colors, 6

plot.paths_animation, 7

vultures, 8