Package: aniMotum (via r-universe)

August 31, 2024

Title Fit Continuous-Time State-Space and Latent Variable Models for Quality Control of Argos Satellite (and Other) Telemetry Data and for Estimating Changes in Animal Movement

Version 1.2-06

Date 2024-07-02

Author Ian Jonsen [aut, cre, cph], W. James Grecian [aut, ctb], Toby Patterson [aut, ctb]

Maintainer Ian Jonsen <ian.jonsen@mq.edu.au>

Description Fits continuous-time random walk, correlated random walk and move persistence state-space models for location estimation and behavioural inference from animal tracking data ('Argos', processed light-level 'geolocation', 'GPS'). Template Model Builder ('TMB') is used for fast random-effects estimation. The 'Argos' data can be: (older) least squares-based locations; (newer) Kalman filter-based locations with error ellipse information; or a mixture of both. The models estimate two sets of location states corresponding to: 1) each observation, which are (usually) irregularly timed; and 2) user-specified time intervals (regular or irregular). A track re-routing function is provided to adjust location estimates for known movement barriers. Track simulation functions are provided. Latent variable models are also provided to estimate move persistence from track data not requiring state-space model filtering.

URL https://github.com/ianjonsen/aniMotum/, https://ianjonsen.github.io/aniMotum/

BugReports https://github.com/ianjonsen/aniMotum/issues/

License MIT + file LICENSE

LazyData true

NeedsCompilation yes

Encoding UTF-8

RoxygenNote 7.3.1

Imports tibble (>= 2.1.3), ggplot2 (>= 3.4.0), TMB (>= 1.9.6), sf (>=
1.0-14), tidyr, dplyr (>= 1.0.0), patchwork, CircStats,
mvtnorm, tmvtnorm, rnaturalearth, traipse, terra, trip (>=
1.10.0)

LinkingTo TMB, RcppEigen

Suggests testthat, covr, knitr, rmarkdown, future, furrr, parallelly,
ggspatial, rosm, rnaturalearthhires, rnaturalearthdata,
pathroutr (>= 0.2.1)

Depends R (>= 4.1.0)

SystemRequirements GDAL (>= 2.4.2), GEOS (>= 3.7.0), PROJ (>= 5.2.0),
pandoc (>=2.7.3)

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

Remotes jmlondon/pathroutr, ropensci/rnaturalearthhires

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

RemoteUrl https://github.com/ianjonsen/aniMotum

RemoteRef HEAD

RemoteSha e599da1808198a0fb82f4f6b5cfd817fce3b3482

Contents

---

aniMotum-package **aniMotum**

Description

fit Continuous-Time Random Walk and Correlated Random Walk state-space models to filter Argos Least Squares or Kalman Filter location data

Author(s)

Maintainer: Ian Jonsen ian.jonsen@mq.edu.au ORCID

Contributor: Toby Patterson ORCID

Contributor: W. James Grecian ORCID

References

Jonsen ID, Grecian WJ, Phillips L, et al. (2023) aniMotum, an R package for animal movement data: rapid quality control, behavioural estimation and simulation. Methods in Ecology and Evolution 14:806-816.

Jonsen ID, Patterson TA, Costa DP, et al. (2020) A continuous-time state-space model for rapid quality-control of Argos locations from animal-borne tags. Movement Ecology 8:31.

Jonsen ID, McMahon CR, Patterson TA, et al. (2019) Movement responses to environment: fast inference of variation among southern elephant seals with a mixed effects model. Ecology. 100(1):e02566.

See Also

Useful Links:

- https://ianjonsen.github.io/aniMotum/

- https://github.com/ianjonsen/aniMotum/

- Report bugs/issues at https://github.com/ianjonsen/aniMotum/issues/

---

aes_lst   *aes_lst*

Description

set aesthetics as a named list for: 1) mapping i) estimated locations; ii) estimated confidence el-
lipses; iii) estimated track lines; iv) observed locations; v) land masses; vi) water bodies; and 2)
colour palettes for i) behavioural indices or ii) individual animal tracks

Usage

```
aes_lst(
  est = TRUE,
  conf = TRUE,
  line = FALSE,
  mp = TRUE,
  obs = FALSE,
  shape = c(19, NA, NA, 17, NA, NA),
  size = c(1.25, NA, 0.2, 0.8, NA, NA),
  col = c("dodgerblue", NA, "grey50", "orange", NA, NA),
  fill = c("dodgerblue", "dodgerblue", NA, "orange", "grey60", "grey85"),
  alpha = c(1, 0.4, 1, 1, 1, NA),
  mp_pal = hcl.colors(100, palette = "Plasma", rev = FALSE),
  id_pal = "Harmonic",
  date_pal = hcl.colors(100, palette = "Viridis", rev = FALSE)
)
```

Arguments

est	logical; turns on estimated locations (default = TRUE)
conf	logical; turns on estimated confidence ellipses. Default varies depending on whether behavioural index is being mapped &/or if single versus multiple tracks are being mapped.
line	logical; turns on estimated track line(s) (default varies)
mp	logical; turns on move persistence index (default = TRUE, if present in model fit)
obs	logical; turns on observed locations (default = FALSE)
shape	gpplot2 shape value (integer: 0, 25) for estimated & observed locations
size	ggplot2 size value for estimated locations & track lines, and observed locations
col	colour for estimated locations and track lines, and observed locations
fill	fill colour for estimated locations & confidence ellipses, observed locations, land polygons, and water
alpha	transparency for specified fills/colours
mp_pal	continuous colour palette for move persistence values
id_pal	discrete colour palette for track id's
date_pal	continuous colour palette for displaying date along track

Value

a named list, with 9 elements, of map components and aesthetics

1. elements 1-5 - map components: est, conf, line, mp, obs
2. element 6: a data.frame named df containing ggplot2 aesthetics: shape, size, col, fill, and alpha. df has 6 rows, corresponding to the map features: estimated locations, confidence ellipses, track lines, observed locations, land polygons, and water
3. element 7: mp_pal
4. element 8: id_pal
5. element 9: date_pal

Examples

```
# generate custom aes list
aes <- aes_lst(conf = FALSE, mp_pal = hcl.colors(n=100, palette = "RdBu"))

# modify aesthetics
aes$df$size[1] <- 1.5
aes$df$fill[6] <- grey(0.9)
```

---

dummy   *Roxygen commands*

Description

Roxygen commands

Usage

```
dummy()
```

---

ellie   *Southern elephant seal Argos satellite data (1 individual, sub-sampled for testing speed)*

Description

Example elephant seal Argos tracking data. Data were sourced from the Integrated Marine Observing System (IMOS) Sourced from the Australian Integrated Marine Observing System (IMOS) deployments at Davis Research Station, Antarctica and are publicly available (http:// imos.aodn.org.au). IMOS is supported by the Australian Government through the National Collaborative Research Infrastructure Strategy and the Super Science Initiative.

Format

.RData

---

emf   *emf*

Description

emf

Usage

```
emf(
  gps = 0.1,
  emf.x = c(1, 1.54, 3.72, 13.51, 23.9, 44.22),
  emf.y = c(1, 1.29, 2.55, 14.99, 22, 32.53)
)
```

Arguments

gps   error multiplication factor(s) for GPS locations, can be a scalar (x = y) or vector
              of length 2 (x != y)

emf.x   error multiplication factors for Argos longitude classes 3, 2, 1, 0, A, B (Z as-
              sumed equal to B)

emf.y   error multiplication factors for Argos latitude classes 3, 2, 1, 0, A, B (Z assumed
              equal to B)

Details

Error Multiplication Factors for Argos (and GPS) locations. Default assumption is that GPS loca-
tions are 10x more accurate than Argos lc 3 in both x and y directions.

User-specified Error Multiplication Factors (emf). emf's must be provided as a data.frame with the
following columns:

emf.x emf values for the x direction

emf.y emf values for y direction

lc location class designations

The location class designations can be the standard Argos lc values: 3, 2, 1, 0, A, B, Z or other
values. The number of classes specified is flexible though may not be amenable to a large number
of classes. Whatever class designations are chosen must also appear in the input data lc column.
A GPS location class ("G") is provided by default and assumes that GPS locations are 10 x more
precise than Argos lc 3 locations.

---

fit_mpm   *fit a* **Move Persistence Model** *(mpm)*

Description

fit a random walk with time-varying move persistence to temporally regular or irregular location data

Usage

```
fit_mpm(
  x,
  what = "predicted",
  model = c("jmpm", "mpm"),
  coords = 3:4,
  control = mpm_control(),
  inner.control = NULL,
  optim = NULL,
  optMeth = NULL,
  verbose = NULL
)
```

Arguments

x	a ssm_df fit object or a data frame of observations (see details)
what	if a ssm_df fit object is supplied then what determines whether fitted, predicted (default), or rerouted values are mapped; ignored if x is a data frame
model	mpm model to fit; either mpm with unpooled random walk variance parameters (sigma_(g,i)) or jmpm with a single, pooled random variance parameter (sigma_g)
coords	column numbers of the location coordinates (default = 3:4)
control	list of control settings for the outer optimizer (see mpm_control for details)
inner.control	list of control parameters for the inner optimization
optim	is deprecated, use ssm_control(optim = "optim") instead, see ssm_control for details
optMeth	is deprecated, use ssm_control(method = "L-BFGS-B") instead, see ssm_control for details
verbose	is deprecated, use ssm_control(verbose = 1) instead, see ssm_control for details

Value

a list with components

- fitted a dataframe of fitted locations
- par model parameter summary

- data input data.frame
- tmb the TMB object
- opt the object returned by the optimizer

References

Jonsen ID, McMahon CR, Patterson TA, et al. (2019) Movement responses to environment: fast inference of variation among southern elephant seals with a mixed effects model. Ecology. 100(1):e02566

Examples

```
## fit jmpm to two southern elephant seal tracks
xs <- fit_ssm(sese2, spdf=FALSE, model = "rw", time.step=72, control = ssm_control(verbose = 0))

fmpm <- fit_mpm(xs, model = "jmpm")
```

---

fit_ssm   *Fit Continuous-Time State-Space Models to filter Argos satellite ge-*
                                   olocation data

Description

fits: i) a simple random walk (rw) ii) a correlated random walk (crw - a random walk on velocity), or iii) a time-varying move persistence model (mp), all in continuous-time, to filter Argos LS, and/or KF/KS location data, GPS data, and/or generic locations with associated standard errors (e.g., processed light-level geolocation data, or high-resolution acoustic telemetry data). Location data of different types can combined in a single data frame (see details). Predicts locations at user-specified time intervals (regular or irregular).

Usage

```
fit_ssm(
  x,
  vmax = 5,
  ang = c(15, 25),
  distlim = c(2500, 5000),
  spdf = TRUE,
  min.dt = 0,
  pf = FALSE,
  model = "crw",
  time.step = NA,
  emf = NULL,
  map = NULL,
  parameters = NULL,
  fit.to.subset = TRUE,
  control = ssm_control(),
```

```
   inner.control = NULL,
   ...
 )
```

Arguments

x	a `data.frame`, `tibble` or `sf-tibble` of observations, depending on the tracking data type. See more in the Details section, below, and the Overview vignette `vignette("Overview", package = "aniMotum")`.
vmax	max travel rate (m/s) to identify implausible locations
ang	angles (deg) of implausible location "spikes"
distlim	lengths (m) of implausible location "spikes"
spdf	(logical) turn pre-filtering on (default; TRUE) or off
min.dt	minimum allowable time difference between observations; `dt <= min.dt` will be ignored by the SSM. Default is 0: all time differences > 0 are allowed.
pf	just pre-filter the data, do not fit the SSM (default is FALSE)
model	fit a simple random walk (`rw`), correlated random walk (`crw`), or a time-varying move persistence model (`mp`), all as continuous-time process models
time.step	options: 1) the regular time interval, in hours, to predict to; 2) a vector of prediction times, possibly not regular, must be specified as a data.frame with id and POSIXt dates; 3) NA - turns off prediction and locations are only estimated at observation times.
emf	optionally supplied data.frame of error multiplication factors for Argos location quality classes. Default behaviour is to use the factors supplied by [emf]
map	a named list of parameters as factors that are to be fixed during estimation, e.g., `list(psi = factor(NA))`
parameters	a list of initial values for all model parameters and unobserved states, default is to let sfilter specify these. Only play with this if you know what you are doing
fit.to.subset	fit the SSM to the data subset determined by [prefilter] (default is TRUE)
control	list of control settings for the outer optimizer (see [ssm_control] for details)
inner.control	list of control settings for the inner optimizer (see [TMB::MakeADFun] for additional details)
...	variable name arguments passed to format_data, see [format_data] for details

Details

x is a `data.frame`, `tibble`, or `sf-tibble` with 5, 7 or 8 columns (the default format), depending on the tracking data type. Argos Least-Squares and GPS data should have 5 columns in the following order: `id`, `date`, `lc`, `lon`, `lat`. Where date can be a POSIX object or text string in YYYY-MM-DD HH:MM:SS format. If a text string is supplied then the time zone is assumed to be `UTC`. lc (location class) can include the following values: 3, 2, 1, 0, A, B, Z, G, or GL. The latter two are for GPS locations and 'Generic Locations', respectively. Class Z values are assumed to have the same error variances as class B. By default, class `G` (GPS) locations are assumed to have error variances 10x smaller than Argos class 3 variances, but unlike Argos error variances the GPS variances are the same for longitude and latitude.

The [format_data](#) function can be used as a data pre-processing step or called automatically within `fit_ssm` to restructure data that is not in one of the above default formats. The minimum essential variables: id**,** date**,** lc**,** lon**,** lat must exist in the input data but they can have different names and exist in a different column order. See [format_data](#) for details.

See [emf](#) for details on how to modify these assumptions.

Argos Kalman Filter (or Kalman Smoother) data should have 8 columns, including the above 5 plus smaj**,** smin**,** eor that contain Argos error ellipse variables (in m for smaj, smin and deg for eor).

Generic locations can be modelled provided each longitude and latitude (or X and Y) coordinate has a corresponding standard error. These data should have 7 columns, including the above 5 plus two extra columns, typically named x.sd, y.sd that provide the standard errors for the longitude, latitude (or X, Y) coordinates. Longitude and latitude standard errors should be in degrees, whereas X and Y standard errors should be in m. In either case, all lc values should be set to GL (Generic Location), the helper function [format_data](#) will add the lc variable to the input data automatically.

Multiple location data types can be combined in a single data frame (see the Overview vignette for examples).

When data are provided as an `sf-tibble`, the user-specified projection is respected, although projected units are always transformed to km to improve SSM convergence efficiency. Otherwise, longlat data are re-projected internally to a global Mercator grid and provided as the default output. A simple `tibble`, without a geom, of lon,lat and x,y location estimates can be obtained by using [grab](#) with the argument as_sf = FALSE.

Value

a list with components

- call the matched call
- predicted an sf tbl of predicted location states
- fitted an sf tbl of fitted locations
- par model parameter summary
- data an augmented sf tbl of the formatted input data
- inits a list of initial values
- pm the process model fit, either "rw" or "crw"
- ts time time.step in h used
- opt the object returned by the optimizer
- tmb the TMB object
- rep TMB sdreport
- aic the calculated Akaike Information Criterion
- time the processing time for sfilter

References

Jonsen ID, Patterson TA, Costa DP, et al. (2020) A continuous-time state-space model for rapid quality-control of Argos locations from animal-borne tags. Movement Ecology 8:31

Jonsen ID, McMahon CR, Patterson TA, et al. (2019) Movement responses to environment: fast inference of variation among southern elephant seals with a mixed effects model. Ecology. 100(1):e02566

Examples

```
## fit crw model to Argos LS data
fit <- fit_ssm(ellie, vmax = 4, model = "crw", time.step = 24,
control = ssm_control(verbose = 0))

## time series plots of fitted values and observations
plot(fit, what = "fitted", type = 1, ask = FALSE)

## 2-D tracks plots of predicted values and observations
plot(fit, what = "predicted", type = 2, ask = FALSE)
```

fmap	*fmap*

Description

map aniMotum fitted or predicted locations, with or without Argos observations, optionally apply a different projection

Usage

```
fmap(
  x,
  y = NULL,
  what = c("fitted", "predicted", "rerouted"),
  conf = TRUE,
  obs = FALSE,
  obs.shp = 17,
  by.id = TRUE,
  by.date = TRUE,
  crs = NULL,
  ext.rng = c(0.05, 0.05),
  size = 0.25,
  col.ssm = "dodgerblue",
  col.obs = "black",
  lines = FALSE,
  landfill = grey(0.6),
  map_type = "default",
  pal = "Cividis",
  rev = FALSE,
  last_loc = NULL,
  ...
)
```

Arguments

x	a aniMotum ssm fit object with class ssm_df
y	optionally, a aniMotum mpm fit object with class mpm_df; default is NULL
what	specify which location estimates to map: fitted, predicted or rerouted
conf	include confidence regions around estimated location (logical; default = TRUE, unless y is an mpm fit object then conf is FALSE)
obs	include Argos observations on map (logical; default = FALSE)
obs.shp	point shape for observations (default = 17)
by.id	when mapping multiple tracks, should locations be coloured by id (logical; default = TRUE if nrow(x) > 1 else FALSE)
by.date	when mapping single tracks, should locations be coloured by date (logical; default = TRUE if nrow(x) == 1 else FALSE)
crs	proj4string for re-projecting locations, if NULL the default projection ("+proj=merc") for the fitting the SSM will be used
ext.rng	factors to extend the plot range in x and y dimensions (can exceed 1)
size	size of estimated location points (size = NA will draw no points). Optionally, a vector of length 2 with size of observed locations given by 2nd value (ignored if obs = FALSE)
col.ssm	colour of ssm-fitted or ssm-predicted locations (ignored if by.id = TRUE)
col.obs	colour of observed locations (ignored if obs = FALSE)
lines	logical indicating if lines are added to connect estimated locations (default = FALSE)
landfill	colour to use for land (default = grey(0.6))
map_type	background map type (default = NULL, which uses rnaturalearth to add landmasses); if packages ggspatial & rosm are installed then any map type returned by rosm::osm.types can be used for a more detailed map background.
pal	hcl.colors palette to use (default: "Cividis"; type hcl.pals() for options)
rev	reverse colour palette (logical)
last_loc	colour to render last location of each track (default = NULL)
...	additional arguments passed to ggspatial::annotation_map_tile

---

format_data   *Coerce input data into expected* aniMotum *format*

Description

format data by mapping supplied variable names to those expected by fit_ssm(), and ensuring variables are put into the expected order. Can be run manually by user as a data pre-processing step prior to calling fit_ssm() or can be called automatically by fit_ssm(). In the latter case, any custom variable names must be declared as arguments to fit_ssm(); see examples, below.

Usage

```
format_data(
  x,
  id = "id",
  date = "date",
  lc = "lc",
  coord = c("lon", "lat"),
  epar = c("smaj", "smin", "eor"),
  sderr = c("x.sd", "y.sd"),
  tz = "UTC"
)
```

Arguments

x	input data
id	the name (as a quoted character string) of id variable: a unique identifier for individual (animal) track data sets.
date	the name (as a quoted character string)of the date/time variable: date and time (as YYYY-MM-DD HH:MM:SS) of each observation.
lc	the name (as a quoted character string) of the location quality class variable: Argos location quality class (values in the set: 3,2,1,0,"A","B","Z"). Can also include "G" for GPS data and/or "GL" for light-level geolocation (GLS) and other data types.
coord	the names (as quoted character strings) of the location coordinate variables: defaults are c("lon","lat"), but could also be c("x","y") for planar coordinates; or if input data is an sf object then "geometry". If input data is an sf object then coord is set to "geometry" by default.
epar	the names (as quoted character strings) of the Argos error ellipse parameters: defaults are "smaj" (ellipse semi-major axis), "smin" (ellipse semi-minor axis), and "eor" (ellipse orientation). Ignored if these variables are missing from the input data.
sderr	the names (as quoted character strings) of provided standard errors for lon,lat or x,y: default names are x.sd, y.sd. Typically, these are only provided for generic location data such as processed light-level geolocations, or high-resolution acoustic detections. The argument is ignored if these variables are missing from the input data.
tz	the timezone the applies to the data/time variable if they are not in tz = 'UTC'. A list of valid timezone names can be viewed via OlsonNames()

Value

a data.frame or sf-tibble of input data in expected aniMotum format. Additional columns required by fit_ssm(), if missing, will be added to the formatted tibble: smaj, smin, eor, x.sd, and y.sd.

Examples

```
## as a data pre-processing step
data(sese2_n)
head(sese2_n, 5)
d <- format_data(sese2_n, date = "time", coord = c("longitude","latitude"),
tz = "America/Halifax")
fit <- fit_ssm(d, model = "crw", time.step = 24)

## called automatically within fit_ssm()
fit <- fit_ssm(sese2_n, date = "time", coord = c("longitude", "latitude"),
tz = "America/Halifax", model = "crw", time.step = 24)
```

---

grab   *grab* `tibble`*'s by name from a* `aniMotum` *model object*

Description

`grab()` lets you obtain `fitted`, `predicted`, `rerouted` or `data` tibble's from a compound `tibble` created when fitting to multiple individual data sets. The specified `tibble`'s are appended to a single output `tibble`.

Usage

```
grab(x, what = "fitted", as_sf = FALSE, normalise = FALSE, group = FALSE)
```

Arguments

x	a aniMotum `ssm_df` or `mpm_df` model object
what	the tibble to be grabbed; either `fitted`, `predicted`, `rerouted` (`ssm_df` only), or `data` (single letters can be used).
as_sf	logical; if FALSE (default) then return a `tibble` with un-projected longlat coordinates, otherwise return an `sf tibble`. Ignored if x is an `mpm` model object.
normalise	logical; if output includes a move persistence estimate, should g (the move persistence index) be normalised to have minimum = 0 and maximum = 1 (default = FALSE). Note, this normalisation is not applied to the standard errors of the logit-scale move persistence estimates (`logit_g`, `logit_g.se`).
group	logical; should g be normalised among individuals as a group, a 'relative g', or to individuals separately to highlight regions of lowest and highest move persistence along single tracks (default = FALSE).

Details

if multiple `ssm_df` model objects are present in x, `as_sf = TRUE`, and at least 1 estimated track has a coordinate reference system (`crs`) with longitude centered on 180 (e.g. a track straddling -180,180) then all tracks will be re-projected to that `crs`.

Value

a `tibble` with all individual `tibble`'s appended

Examples

```
## generate an ssm fit object
xs <- fit_ssm(ellie, spdf=FALSE, model = "rw", time.step=24, control = ssm_control(verbose = 0))

## grab predicted values as an un-projected tibble
preds <- grab(xs, what = "predicted")
```

---

join	*join an mpm-estimated behavioural index to ssm-predicted locations*

Description

`join()` joins ssm-predicted locations and mpm-estimated behavioural index into a single tibble. If the ssm-predicted tibble is a projected sf object then the output of join will also be an sf object (default). This can be avoided by using `as_sf = FALSE`.

Usage

```
join(
  ssm,
  mpm,
  what.ssm = "predicted",
  as_sf = FALSE,
  normalise = FALSE,
  group = FALSE
)
```

Arguments

`ssm`	an `aniMotum` ssm fitted model object
`mpm`	an `aniMotum` mpm fitted model object
`what.ssm`	specifies whether ssm `predicted` or `fitted` values are to be extracted
`as_sf`	logical; if FALSE then return a tibble with un-projected lonlat coordinates, otherwise return an sf tibble
`normalise`	logical; if output includes a move persistence estimate, should g (the move persistence index) be normalised to have minimum = 0 and maximum = 1 (default = FALSE).
`group`	logical; should g be normalised among individuals as a group, a 'relative g', or separately to highlight regions of lowest and highest move persistence along a track (default = FALSE).

Value

a single tbl with all individuals

Examples

```
## load example aniMotum fit objects (to save time)
## generate a ssm fit object
xs <- fit_ssm(ellie, spdf=FALSE, model = "rw", time.step=24, control = ssm_control(verbose = 0))
xm <- fit_mpm(xs, what = "p", model = "mpm")

## join predicted values as an un-projected tibble
xsm <- join(xs, xm)
xsm
```

---

map	*map*

Description

map aniMotum-estimated locations and behavioural indices with coastline and projection options

Usage

```
map(
  x,
  y = NULL,
  what = c("fitted", "predicted", "rerouted"),
  aes = aes_lst(),
  by.id = TRUE,
  by.date = FALSE,
  crs = NULL,
  ext.rng = c(0.05, 0.05),
  buffer = 10000,
  map_type = "default",
  normalise = TRUE,
  group = FALSE,
  silent = FALSE,
  ...
)
```

Arguments

x	a aniMotum ssm fit object with class ssm_df or (old) fG_ssm
y	optionally, a aniMotum mpm fit object with class mpm_df or (old) fG_mpm
what	specify which location estimates to map: fitted, predicted or rerouted

map 17

aes	a list of map controls and aesthetics (shape, size, col, fill, alpha) for each map feature (estimated locations, confidence ellipses, track lines, observed locations, land masses, water bodies). Constructed by aes_lst() and can be modified for custom maps (see examples)
by.id	when mapping multiple tracks, should locations be coloured by id (logical; default = TRUE if nrow(x) > 1 else FALSE; ignored if behavioural index provided)
by.date	when mapping single tracks, should locations be coloured by date (logical; default = FALSE; ignored if behavioural index provided)
crs	proj4string for re-projecting locations, if NULL the default projection (Mercator) for the fitting the SSM will be used
ext.rng	proportion (can exceed 1) to extend the plot range in x and y dimensions
buffer	distance (in km) to buffer locations for subsetting land polygons (default = 10000). If map extents are expanded by many factors then the buffer distance may need to be increased, otherwise this should not be used. Ignored if map_type != "default".
map_type	background map type ("default" uses [rnaturalearth::ne_countries](#) to add land polygons). If the rnaturalearthdata package is installed then high-resolution land polygons will be used. If the ggspatial and rosm packages are installed then any tile map type returned by [rosm::osm.types](#) can be used for a potentially more detailed coastline at fine spatial scales, given appropriate zoom settings (see [ggspatial::annotation_map_tile](#) for details).
normalise	logical; if output includes a move persistence estimate, should g (the move persistence index) be normalised to have minimum = 0 and maximum = 1 (default = TRUE).
group	logical; should g be normalised among individuals as a group, a 'relative g', or separately to highlight regions of lowest and highest move persistence along a track (default = FALSE).
silent	logical; generate maps silently (default = FALSE).
...	additional arguments passed to [ggspatial::annotation_map_tile](#)

Value

a map as a ggplot2 object

Examples

```
# create an ssm fit object

fit <- fit_ssm(ellie, model = "rw", time.step = 24, control = ssm_control(verbose = 0))

# render default map

map(fit, what = "p")
```

---

mpm_control   *Control Values for* fit_mpm.

Description

mpm_control selects the numerical minimizer, method, associated control parameters, and parameter bounds used by fit_mpm.

Usage

```
mpm_control(
  optim = c("nlminb", "optim"),
  method = c("L-BFGS-B", "BFGS", "Nelder-Mead", "CG", "SANN", "Brent"),
  lower = NULL,
  upper = NULL,
  verbose = 1,
  ...
)
```

Arguments

optim	the numerical optimizer used in the fit
method	if optim = "optim" then the optimization method to be used can be one of "BFGS", "L-BFGS-B", "Nelder-Mead", "CG", "SANN", or "Brent" see optim for details
lower	a list named parameter lower bounds, if NULL then built in defaults are used when method = "L-BFGS-B". Possible parameter names are: l_sigma a vector of length 2, log scale; l_rho_p a scalar, logit scale; l_D a scalar, log scale; l_psi a scalar, log scale; l_tau a vector of length 2, log scale; l_rho_o a scalar, logit scale
upper	a list of named parameter upper bounds, if NULL then built in defaults are used when method = "L-BFGS-B". Possible parameter names are same as lower
verbose	integer; report progress during minimization: 0 = silent; 1 = optimizer trace; 2 = parameter trace (default))
...	control parameters for the chosen optimizer

Details

The optimizer used to minimize the objective function is selected by the optim argument. Additional control parameters specific to the chosen optimizer are specified via the dots argument. See nlminb and optim for available options. Adapted from S. Wotherspoon https://github.com/SWotherspoon/RWalc/blob/master/R/RWalc.R

Value

Returns a list with components

optim	the name of the numerical optimizer as a string, "nlminb" or "optim"
method	optimization method to be used
lower	named list of lower parameter bounds
upper	named list of upper parameter bounds
verbose	level of tracing information to be reported
control	list of control parameters for the optimizer

See Also

[nlminb](), [optim]().

---

osar	*calculate one-step-ahead (prediction) residuals from a* aniMotum ssm *fit*

Description

calculate one-step-ahead (prediction) residuals from a aniMotum ssm fit

Usage

```
osar(x, method = "fullGaussian", ...)
```

Arguments

x	a aniMotum ssm fit object with class ssm_df
method	method to calculate prediction residuals (default is oneStepGaussianOffMode; see [TMB::oneStepPredict]() for details)
...	other arguments to [TMB::oneStepPredict]()

Details

One-step-ahead residuals are useful for assessing goodness-of-fit in latent variable models. This is a wrapper function for [TMB::oneStepPredict]() (beta version). osar tries the fullGaussian (fastest) method first and falls back to the oneStepGaussianOffMode (slower) method for any failures. Subsequent failures are dropped from the output and a warning message is given. Note, OSA residuals can take a considerable time to calculate if there are many individual fits and/or deployments are long. The method is automatically parallelised across 2 x the number of individual fits, up to the number of processor cores available.

References

Thygesen, U. H., C. M. Albertsen, C. W. Berg, K. Kristensen, and A. Neilsen. 2017. Validation
of ecological state space models using the Laplace approximation. Environmental and Ecological
Statistics 24:317–339.

Examples

```
# generate a ssm fit object (call is for speed only)
xs <- fit_ssm(ellie, spdf=FALSE, model = "rw", time.step=24, control = ssm_control(verbose = 0))

res <- osar(xs)
```

---

plot.mpm_df	*plot*

Description

visualize fits from an mpm object

Usage

```
## S3 method for class 'mpm_df'
plot(
  x,
  y = NULL,
  se = FALSE,
  pages = 0,
  ncol = 1,
  ask = TRUE,
  pal = "Plasma",
  rev = FALSE,
  ...
)
```

Arguments

x	a `aniMotum` mpm fit object with class `mpm_df`
y	optional `ssm` fit object with class `ssm_df` corresponding to x. If absent, 1-d plots of `gamma_t` time series are rendered otherwise, 2-d track plots with locations coloured by 'gamma_t" are rendered.
se	logical (default = FALSE); should points be scaled by `gamma_t` uncertainty (ignored if y is not supplied)
pages	plots of all individuals on a single page (pages = 1; default) or each individual on a separate page (pages = 0)

ncol	number of columns to use for faceting. Default is ncol = 1 but this may be increased for multi-individual objects. Ignored if pages = 0
ask	logical; if TRUE (default) user is asked for input before each plot is rendered. set to FALSE to return ggplot objects
pal	[grDevices::hcl.colors](#) palette to use (default: "Plasma"; see [grDevices::hcl.pals](#) for options)
rev	reverse colour palette (logical)
...	additional arguments to be ignored

Value

a ggplot object with either: 1-d time series of gamma_t estimates (if y not provided), with estimation uncertainty ribbons (95 % CI's); or 2-d track plots (if y provided) coloured by gamma_t, with smaller points having greater uncertainty (size is proportional to SE^-2, if se = TRUE). Plots can be rendered all on a single page (pages = 1) or on separate pages.

Examples

```
# generate a ssm fit object (call is for speed only)
xs <- fit_ssm(sese2, spdf=FALSE, model = "rw", time.step=72, control = ssm_control(verbose = 0))

# fit mpm to ssm fits
xm <- fit_mpm(xs, model = "jmpm")

# plot 1-D mp timeseries on 1 page
plot(xm, pages = 1)
```

---

plot.osar	*plot*

Description

plot **One-Step-Ahead (prediction) residuals** from a aniMotum osar object

Usage

```
## S3 method for class 'osar'
plot(
  x,
  type = c("ts", "qqnorm", "acf"),
  pages = 1,
  ncol = 1,
  ask = TRUE,
  pal = "Zissou1",
  ...
)
```

Arguments

x	a aniMotum osar object with class osar
type	type of residual plot to generate: time-series (ts; default), qqnorm (qq), or acf
pages	plots of all individuals on a single page (pages = 1; default) or each individual on a separate page (pages = 0)
ncol	number of columns to use for faceting. Default is ncol = 2 but this may be increased for multi-individual fit objects
ask	logical; if TRUE (default) user is asked for input before each plot is rendered. set to FALSE to return ggplot objects
pal	grDevices::hcl.colors colour palette to use (default = "Zissou1"; see `grDevices::hcl.pals()` for options)
...	additional arguments to be ignored

Examples

```
## generate a fG_ssm fit object (call is for speed only)
xs <- fit_ssm(ellie, spdf=FALSE, model = "rw", time.step=24,
control = ssm_control(se = FALSE, verbose = 0))

res <- osar(xs)

plot(res, type = "qq")
```

---

plot.sim	*plot*

Description

visualize simulated tracks from a sim data.frame

Usage

```
## S3 method for class 'sim'
plot(x, type = 2, error = FALSE, pal = "Plasma", rev = FALSE, ...)
```

Arguments

x	a aniMotum simulation data.frame with class sim
type	either 1, a 1-D time-series of speed (if model is rw or crw specified behavioural states) or move persistence (g; if model is mp); or 2 (default), a 2-D track with location coloured by move persistence (g; if model = "mp")
error	logical, plot locations with error (TRUE) or without. Ignored in 1-D time-series plots

pal	grDevices::hcl.colors palette to use (default: "Plasma"); see `grDevices::hcl.pals()` for options
rev	reverse direction of colour palette; logical (default = FALSE)
...	additional arguments to be ignored

Value

Plots of simulated tracks. Can be rendered all on a single page (pages = 1) or on separate pages (pages = 0).

Examples

```
tr <- sim(N=200, model = "mp")
plot(tr)
```

---

plot.sim_fit   *plot*

Description

visualize tracks simulated from a aniMotum model fit

Usage

```
## S3 method for class 'sim_fit'
plot(x, type = NULL, zoom = TRUE, ncol = 1, hires = FALSE, ortho = TRUE, ...)
```

Arguments

x	a aniMotum simulation data.frame with class sim_fit
type	deprecated. All tracks are rendered as points.
zoom	logical; should map extent be defined by track extent (TRUE; default) or should global map be drawn (FALSE).
ncol	number of columns to arrange multiple plots
hires	logical; use high-resolution coastline data. Attempts to use high-res coastline data via rnaturalearth::ne_countries with scale = 10, if the rnaturalearthhires data package is installed. This extends the plot rendering time so is set to FALSE by default, in which case rnaturalearth::ne_countries with scale = 50 data are used.
ortho	logical; use an orthographic projection centered on the track starting location(s) (TRUE; default). An orthographic projection may be optimal for high latitude tracks and/or tracks that traverse long distances. If FALSE then a global Mercator projection is used.
...	additional arguments to be ignored

Value

Plots of simulated tracks.

Examples

```
fit <- fit_ssm(ellie, model = "crw", time.step = 24)
trs <- sim_fit(fit, what = "p", reps = 2)
plot(trs)
```

---

plot.sim_post   *plot*

Description

visualize tracks simulated from a aniMotum model fit

Usage

```
## S3 method for class 'sim_post'
plot(
  x,
  type = c("lines", "points", "both"),
  zoom = TRUE,
  ncol = 1,
  hires = TRUE,
  ortho = TRUE,
  alpha = 0.5,
  ...
)
```

Arguments

x	a aniMotum simulation data.frame with class sim_fit
type	plots tracks as "line", "points" or "both" (default).
zoom	logical; should map extent be defined by track extent (TRUE; default) or should global map be drawn (FALSE).
ncol	number of columns to arrange multiple plots
hires	logical; use high-resolution coastline data. Attempts to use high-res coastline data via rnaturalearth::ne_countries with scale = 10, if the rnaturalearthhires data package is installed. If not, then rnaturalearth::ne_countries with scale = 50 data are used.
ortho	logical; use an orthographic projection centered on the track starting location(s) (TRUE; default). An orthographic projection may be optimal for high latitude tracks and/or tracks that traverse long distances. If FALSE then a global Mercator projection is used.

alpha	opacity of simulated track points/lines. Lower opacity can ease visualization when multiple simulated overlap one another.
...	additional arguments to be ignored

Value

Plots of posterior simulated tracks.

Examples

```
fit <- fit_ssm(ellie, model = "crw", time.step = 24)
psim <- sim_post(fit, what = "p", reps = 10)
plot(psim, type = "lines")
```

---

plot.ssm_df   *plot*

Description

visualize fits from an ssm object

Usage

```
## S3 method for class 'ssm_df'
plot(
  x,
  what = c("fitted", "predicted", "rerouted"),
  type = 1,
  outlier = TRUE,
  alpha = 0.3,
  pages = 0,
  ncol = 1,
  ask = TRUE,
  pal = "default",
  normalise = TRUE,
  group = FALSE,
  ...
)
```

Arguments

x	a aniMotum ssm fit object with class ssm_df
what	specify which location estimates to display on time-series plots: fitted, predicted, or rerouted

type	of plot to generate: 1-d time series for lon and lat separately (type = 1, default); 2-d track plot (type = 2); 1-d time series of move persistence estimates (type = 3; if fitted model was mp); 2-d track plot with locations coloured by move persistence (type = 4; if fitted model was mp)
outlier	include outlier locations dropped by prefilter (outlier = TRUE, default)
alpha	opacity of standard errors. Lower opacity can ease visualization when multiple ellipses overlap one another
pages	each individual is plotted on a separate page by default (pages = 0), multiple individuals can be combined on a single page; pages = 1
ncol	number of columns to arrange plots when combining individuals on a single page (ignored if pages = 0)
ask	logical; if TRUE (default) user is asked for input before each plot is rendered. set to FALSE to return ggplot objects
pal	grDevices::hcl.colors palette to use (see grDevices::hcl.pals() for options)
normalise	logical; if plotting move persistence estimates from an mp model fit, should estimates be normalised to 0,1 (default = TRUE).
group	logical; should g be normalised among individuals as a group, a 'relative g', or to individuals separately to highlight regions of lowest and highest move persistence along single tracks (default = FALSE).
...	additional arguments to be ignored

Value

a ggplot object with either: (type = 1) 1-d time series of fits to data, separated into x and y components (units = km) with prediction uncertainty ribbons (2 x SE); or (type = 2) 2-d fits to data (units = km)

Examples

```
## generate a ssm fit object (call is for speed only)
xs <- fit_ssm(sese2, spdf=FALSE, model = "rw", time.step=72, control = ssm_control(verbose = 0))

# plot fitted locations as 1-D timeseries on 1 page
plot(xs, what = "f", pages = 1)
```

---

print.ssm   *print* aniMotum *fit object summary information*

Description

print aniMotum fit object summary information

Usage

```
## S3 method for class 'ssm'
print(x, ...)
```

Arguments

x	a aniMotum ssm fit object
...	unused. For compatibility with the generic method.

Examples

```
## see summary fit output
## generate a ssm fit object (call is for speed only)
xs <- fit_ssm(ellie, spdf=FALSE, model = "rw", time.step=24,
control = ssm_control(se = FALSE, verbose = 0))

xs$ssm[[1]]
```

---

route_path	*Re-route a movement path around land using the* pathroutr *package*

Description

A wrapper function that uses the pathroutr package <https://jmlondon.github.io/pathroutr/> to re-route movement paths that cross a land barrier. The current implementation will take either the output from a fit_ssm model or the simulations generated by sim_fit.

Usage

```
route_path(
  x,
  what = c("fitted", "predicted"),
  map_scale = 50,
  dist = 50000,
  append = TRUE,
  ...
)
```

Arguments

x	either a ssm fit object or a sim_fit object containing simulated paths
what	if using a ssm object should the fitted (typically irregular in time) or predicted (typically regular in time) locations be re-routed.
map_scale	scale of rnaturalearth map to use for land mass: one of 110, 50 (default), or 10. Note that map_scale = 10 is only available if you have the rnaturalearthhires package installed see: <https://github.com/ropensci/rnaturalearthhires>

dist	buffer distance (m) to add around track locations. The convex hull of these buffered locations defines the size of land polygon used to aid re-routing of points on land. Larger buffers can result in longer computation times. See London (2020) for further details. The default buffer distance is a constant 50000 m.
append	should re-routed locations be appended to the ssm (ssm fit) object (default = TRUE), or returned as a tibble.
...	additional arguments passed to pathroutr::prt_visgraph

Details

route_path uses rnaturalearth::ne_countries at the medium (50) scale, by default, to generate a land barrier. For efficient computation, route_path clips the polygons to the buffered bounds (set by dist (in m) of the movement track(s).

When the input is a ssm object route_path can append the re-routed path locations to the ssm (ssm fit) object. This is useful when move persistence is to be estimated from the re-routed locations via fit_mpm, or tracks are to be visualised with map. route_path can also return a standalone tibble of the re-routed path with the same number of locations as either the original fitted or predicted locations.

When the re-routed path is appended to the ssm object, the path can be extracted using the grab function, e.g. grab(fit, what = "rerouted").

When the input is a sim_fit object then route_path returns the same object but with the locations within each simulation re-routed.

We recommend that users working on complex rerouting problems and/or requiring higher resolution land barrier data work with the pathroutr package directly by first exctracting aniMotum-estimated locations with grab. Higher resolution land barrier data (polygon shapefiles) must be obtained independently.

References

Josh M. London. (2020) pathroutr: An R Package for (Re-)Routing Paths Around Barriers (Version v0.2.1) https://zenodo.org/record/5522909#.YnPxEy_b1qs

Examples

```
# if 'pathroutr' is installed then ok to use route_path()
if(requireNamespace("pathroutr", quietly = TRUE)) {
  fit <- fit_ssm(ellie, vmax = 4, model = "crw", time.step = 24)
  fit <- route_path(fit, what = "predicted")
  grab(fit, what = "rerouted")
}
```

| sese | *Southern elephant seal Argos satellite data (3 individuals)* |

Description

Example elephant seal Argos tracking data. Data were sourced from the Integrated Marine Observing System (IMOS) Sourced from the Australian Integrated Marine Observing System (IMOS) in collaboration with the French IPEV and SNO-MEMO project deployments at Iles Kerguelen and are publicly available (http:// imos.aodn.org.au). IMOS is supported by the Australian Government through the National Collaborative Research Infrastructure Strategy and the Super Science Initiative.

Format

.RData

| sese2 | *Southern elephant seal Argos satellite data (2 highly sub-sampled individuals)* |

Description

Example elephant seal Argos tracking data, highly sub-sampled. These example data are included purely to speed up examples where a fit object is required. Generating a fit object is preferred as storing an example fit risks GDAL errors on platforms with older GDAL libraries. Sourced from the Australian Integrated Marine Observing System (IMOS) in collaboration with the French IPEV and SNO-MEMO project deployments at Iles Kerguelen and are publicly available (http:// imos.aodn.org.au). IMOS is supported by the Australian Government through the National Collaborative Research Infrastructure Strategy and the Super Science Initiative.

Format

.RData

---

sese2_n	*Southern elephant seal Argos satellite data (2 highly sub-sampled individuals)*

Description

Example elephant seal Argos tracking data, highly sub-sampled with default variable order scrambled and renamed. These example data are included purely to speed up examples where a fit object is required. Generating a fit object is preferred as storing an example fit risks GDAL errors on platforms with older GDAL libraries. Sourced from the Australian Integrated Marine Observing System (IMOS) in collaboration with the French IPEV and SNO-MEMO project deployments at Iles Kerguelen and are publicly available (http:// imos.aodn.org.au). IMOS is supported by the Australian Government through the National Collaborative Research Infrastructure Strategy and the Super Science Initiative.

Format

.RData

---

sim	*simulate animal tracks*

Description

simulate from the rw, crw, or mp process models to generate a set of x,y (or lon,lat) coordinates with or without error from supplied input parameters.

Usage

```
sim(
  N = 100,
  start = list(c(0, 0), as.POSIXct(format(Sys.time(), tz = "UTC", usetz = TRUE))),
  model = c("rw", "crw", "mp"),
  vmax = 4,
  sigma = c(4, 4),
  rho_p = 0,
  D = 0.05,
  sigma_g = 1.25,
  error = c("ls", "kf"),
  tau = c(1.5, 0.75),
  rho_o = 0,
  tdist = c("reg", "gamma"),
  ts = 6,
  tpar = 1.2,
  alpha = c(0.9, 0.8)
)
```

Arguments

N	number of time steps to simulate
start	coordinates and datetime of start location for simulated track
model	simulate from the rw, crw or mp process models
vmax	maximum travel rate (m/s) of simulated animal
sigma	a vector of process error sd's for the rw model (ignored if model != "rw")
rho_p	correlation parameter for rw model process covariance matrix (ignored if model != "rw")
D	diffusion coefficient for crw model process covariance matrix (ignored if model != "crw")
sigma_g	random walk sd for time-varying move persistence parameter (ignored if model != "mp")
error	indicates whether measurement error should mimic Argos Least-Squares (ls) or Argos Kalman Filter (kf)
tau	vector of LS measurement error sd's (ignored if error = "kf")
rho_o	correlation parameter for LS covariance matrix (ignored if error = "kf")
tdist	distribution for simulating location times (reg generates locations at regular ts intervals, in h; gamma uses a gamma distribution to generate random time intervals)
ts	time interval in h
tpar	rate parameter for the gamma distributed times, shape is take to be ts * tpar for a mean interval of approximately ts h (ignored if tdist = "reg")
alpha	transition probabilities switching model versions of rw or crw models. Probabilities are the transition matrix diagonals (ignored if sigma has length 2 or D has length 1)

Value

a tibble is returned with columns that can include some or all of the following, depending on the arguments used

- date time as POSIXct, tz = UTC (default)
- lc Argos location class
- lon longitude with error
- lat latitude with error
- x x in km from arbitrary origin without error
- y y in km from arbitrary origin without error
- x.err a random deviate drawn from Argos LS or KF error distribution
- y.err a random deviate drawn from Argos LS or KF error distribution
- smaj Argos error ellipse semi-major axis in m (if error = "kf")
- smin Argos error ellipse semi-minor axis in m (if error = "kf")

- eor Argos error ellipse orientation in degrees (if error = "kf")

- u velocity in x direction (if model = "crw"), unit = km/h

- v velocity in y direction (if model = "crw"), unit = km/h

- b behavioural state (if model = "rw" or model = "crw" and multiple process variances given, see examples)

- g movement persistence - the autocorrelation between successive movements on the interval 0,1 (if model = "mp")

Examples

```
tr <- sim(N = 200, model = "crw", D = 0.1, error = "kf", tdist = "reg", ts=12)
plot(tr, error = TRUE)

tr <- sim(N = 200, model = "mp", sigma_g = 1.2, error = "ls", tau = c(2, 1.5), ts=12,
tdist = "gamma", tpar = 1.5)
plot(tr, error = TRUE, pal = "Cividis")
```

---

sim_filter   *Filter* sim_fit *simulations*

Description

This function calculates the similarity between the simulations generated by sim_fit and the SSM-estimated path from the ssm fit, and returns a sim_fit object containing the most similar tracks based on a user specified quantile. In this context, similarity is calculated as the sum of normalised differences in net displacement (km) and overall bearing (deg) between the SSM-estimated path and the simulated paths.

Usage

```
sim_filter(trs, keep = 0.25, flag = 2, var = NULL, FUN = "mean", ...)
```

Arguments

trs	a sim_fit object
keep	the quantile of flag values to retain
flag	the similarity flag method (see details). Ignored if var != NULL.
var	the name(s) of the appended variable(s) to use for similarity calculations. Default is NULL, in which case similarity is calculated based on distance and bearing - e.g., Hazen et al (2017).
FUN	one of the following functions in quotes: mean, median, var, sd, sum, min, or max. Ignored if var = NULL.
...	additional arguments to the specified FUN (e.g., na.rm = TRUE). Ignored if var = NULL.

Details

- `flag = 1` will use an index based on Hazen (2017)
- `flag = 2` (the default) will use a custom index

Value

a `sim_fit` object containing the filtered paths

References

Hazen et al. (2017) WhaleWatch: a dynamic management tool for predicting blue whale density in the California Current J. Appl. Ecol. 54: 1415-1428

Examples

```
## fit crw model to Argos LS data
fit <- fit_ssm(ellie, model = "crw", time.step = 72)

set.seed(pi)
## generate 5 simulated paths from ssm fit
trs <- sim_fit(fit, what = "predicted", reps = 5)

## filter simulations and keep paths in top 40% of flag values
trs_f <- sim_filter(trs, keep = 0.4, flag = 2)

## compare unfiltered and filtered simulated paths

plot(trs) | plot(trs_f)
```

---

sim_fit   *simulate animal tracks from a* ssm *fit*

Description

simulate from the `rw` or `crw` process models to generate either a set of x,y or lon,lat coordinates from a `ssm` fit with length equal to the number of observations used in the SSM fit.

Usage

```
sim_fit(
  x,
  what = c("fitted", "predicted"),
  reps = 1,
  start = NULL,
  end = NULL,
  grad = NULL,
```

```
  beta = c(-300, -300),
  cpf = FALSE,
  sim_only = FALSE
)
```

Arguments

x	a ssm fit object with class ssm_df
what	simulate fitted (typically irregular in time) or predicted (typically regular in time) locations
reps	number of replicate tracks to simulate from an ssm model fit object
start	a 2-element vector for the simulated track start location (lon,lat or x,y)
end	a 2-element vector for the simulated track end location (lon,lat or x,y)
grad	a SpatRaster of x- and y-gradients as separate layers (see details)
beta	a 2-element vector of parameters defining the potential function magnitude in x- and y-directions (ignored if is.null(grad), ie. no potential function; see details).
cpf	logical; should simulated tracks return to their start point (ie. a central-place forager)
sim_only	logical, do not include ssm estimated location in output (default is FALSE)

Details

A potential function can be applied to the simulated paths to help avoid locations on land (or in water), using the grad and beta arguments. A coarse-resolution rasterStack of global x- and y-gradients of distance to land are provided. Stronger beta parameters result in stronger land (water) avoidance but may also introduce undesirable/unrealistic artefacts (zig-zags) in the simulated paths. See Brillinger et al. (2012) and vignette("momentuHMM", package = "momentuHMM") for more details on the use of potential functions for simulating constrained animal movements. WARNING: This application of potential functions to constrain simulated paths is experimental, likely to change in future releases, and NOT guaranteed to work enitrely as intended, especially if cpf = TRUE!

Value

a fG_sim_fit object containing the paths simulated from a ssm fit object

References

Brillinger DR, Preisler HK, Ager AA, Kie J (2012) The use of potential functions in modelling animal movement. In: Guttorp P., Brillinger D. (eds) Selected Works of David Brillinger. Selected Works in Probability and Statistics. Springer, New York. pp. 385-409.

Examples

```
fit <- fit_ssm(ellie, model = "crw", time.step = 24)
trs <- sim_fit(fit, what = "predicted", reps = 3)
plot(trs)
```

---

sim_post *simulate from the posterior of a* ssm *fit.*

Description

simulates track locations from the joint precision matrix of a ssm model fit. Currently, the joint precision of the SSM movement parameters is not included (ie. a full posterior simulation).

Usage

```
sim_post(x, what = "predicted", reps = 1, sim_only = FALSE)
```

Arguments

x	a ssm fit object with class ssm_df
what	simulate fitted or predicted locations
reps	number of replicate tracks to simulate from the ssm model fit object
sim_only	logical, do not include ssm estimated locations in output (default is FALSE)

Value

a fG_sim_post object containing the paths simulated from a ssm fit object

Examples

```
fit <- fit_ssm(ellie, model = "crw", time.step = 24)
psim <- sim_post(fit, "p", reps = 10)
plot(psim, type = "lines")
```

---

ssm_control *Control Values for* fit_ssm.

Description

ssm_control selects the numerical minimizer, method, associated control parameters, and parameter bounds used by fit_ssm.

Usage

```
ssm_control(
  optim = c("nlminb", "optim"),
  method = c("L-BFGS-B", "BFGS", "Nelder-Mead", "CG", "SANN", "Brent"),
  lower = NULL,
  upper = NULL,
  verbose = 1,
  se = FALSE,
  ...
)
```

Arguments

optim	the numerical optimizer used in the fit
method	if optim = "optim" then the optimization method to be used can be one of "BFGS", "L-BFGS-B", "Nelder-Mead", "CG", "SANN", or "Brent" see [optim](#) for details
lower	a list named parameter lower bounds, if NULL then built in defaults are used when method = "L-BFGS-B". Possible parameter names are: l_sigma a vector of length 2, log scale; l_rho_p a scalar, logit scale; l_D a scalar, log scale; l_psi a scalar, log scale; l_tau a vector of length 2, log scale; l_rho_o a scalar, logit scale
upper	a list of named parameter upper bounds, if NULL then built in defaults are used when method = "L-BFGS-B". Possible parameter names are same as lower
verbose	integer; report progress during minimization: 0 = silent; 1 = parameter trace (default); 2 = optimizer trace
se	logical; should standard errors for speed estimates be calculated (default = FALSE). Turning this on will slow down computation time but provide SE's for speed-along-track calculations
...	control parameters for the chosen optimizer

Details

The optimizer used to minimize the objective function is selected by the optim argument. Additional control parameters specific to the chosen optimizer are specified via the dots argument. See [nlminb](#) and [optim](#) for available options. Adapted from S. Wotherspoon https://github.com/SWotherspoon/RWalc/blob/master/R/RWalc.R

Value

Returns a list with components

optim	the name of the numerical optimizer as a string, "nlminb" or "optim"
method	optimization method to be used
lower	named list of lower parameter bounds
upper	named list of upper parameter bounds

| verbose | level of tracing information to be reported |
| control | list of control parameters for the optimizer |

See Also

[nlminb](), [optim]().

Examples

```
fit <- fit_ssm(ellie,
vmax = 4,
model = "crw",
time.step = 72,
control = ssm_control(
    optim = "nlminb",
    eval.max = 2000)
    )
```

---

summary.ssm_df   *object summaries*

Description

return a summary of an ssm_df fit object

Usage

```
## S3 method for class 'ssm_df'
summary(object, ...)
```

Arguments

| object | an ssm_df fit object |
| ... | additional arguments to be ignored |

---

wese_sb   *Weddell seal Argos satellite data (1 individual)*

Description

Example Weddell seal Argos tracking data, deployed at Scott Base, Ross Island, Antarctica. This example data set is included for demonstration purposes. Sourced from the Australian Integrated Marine Observing System (IMOS) & the New Zealand National Institute of Water and Atmospheric Research (NIWA) deployments at Scott Base, Antarctica and are publicly available (http://imos.aodn.org.au). IMOS is supported by the Australian Government through the National Collaborative Research Infrastructure Strategy and the Super Science Initiative.

Format

.RData

Index