

# Package ‘foieGras’

April 4, 2019

**Title** Fit Continuous-Time State-Space Models for Filtering Argos  
Satellite (and Other) Telemetry Data

**Version** 0.2.1

**Author** Ian Jonsen [aut, cre],  
Toby Patterson [aut, ctb]

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

**Description** Fits continuous-time random walk and correlated random walk state-space models to filter Argos satellite location data. Template Model Builder (‘TMB’) is used for fast 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. Separate measurement models are used for these two data types. 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). Jonsen I, McMahon CR, Patterson TA, Auger-Methe M, Harcourt R, Hindell MA, Bestley S (2019) Movement responses to environment: fast inference of variation among southern elephant seals with a mixed effects model. *Ecology* 100:e02566 <doi:10.1002/ecy.2566>.

**License** AGPL-3

**LazyData** true

**Encoding** UTF-8

**RoxygenNote** 6.1.1

**LinkingTo** TMB, RcppEigen

**Imports** dplyr, tibble, argosfilter, ggplot2, gridExtra, lubridate,  
TMB, sf, stringr, magrittr

**Suggests** testthat, covr, knitr, rmarkdown, rnaturalearth, rgeos,  
ggspatial, units

**VignetteBuilder** knitr

**Depends** R (>= 3.3.0)

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2019-04-04 13:10:02 UTC

**R topics documented:**

foieGras-package . . . . .	2
dummy . . . . .	3
ellie . . . . .	3
fit . . . . .	3
fit_ssm . . . . .	4
grab . . . . .	5
plot.foieGras . . . . .	6
pluck . . . . .	7
prefilter . . . . .	7
print.foieGras . . . . .	8
rope . . . . .	9
sfilter . . . . .	9
wrap_lon . . . . .	10

<b>Index</b>	<b>11</b>
--------------	-----------

---

foieGras-package	<b>foieGras</b>
------------------	-----------------

---

**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)**

Ian Jonsen, Toby Patterson

**References**

Jonsen I, McMahon CR, Patterson TA, Auger-Methe M, Harcourt R, Hindell MA, Bestley S (2019) Movement responses to environment: fast inference of variation among southern elephant seals with a mixed effects model. *Ecology* 100:e02566

**See Also**

fit\_ssm

---

dummy	<i>Roxygen commands</i>
-------	-------------------------

---

**Description**

Roxygen commands

**Usage**

dummy()

---

ellie	<i>Elephant seal Argos satellite data (1 individual)</i>
-------	--

---

**Description**

Example elephant seal Argos tracking data. Data were sourced from the Integrated Marine Observing System (IMOS) - IMOS is supported by the Australian Government through the National Collaborative Research Infrastructure Strategy and the Super Science Initiative.

**Format**

.RData

---

fit	<i>foieGras example fit object</i>
-----	------------------------------------

---

**Description**

Example foieGras fit object, using ellie example data and the following call: `fit <- fit_ssm(ellie, model="rw", time.step=24)`. This example fit is included purely to speed up examples where a fit object is required but fitting to data is not the focus of the example.

**Format**

.RData

---

fit_ssm	<i>Fit a continuous-time state-space model to filter Argos satellite geolocation data</i>
---------	---

---

### Description

fits either a simple random walk or a correlated random walk (a random walk on velocity) in continuous time to filter Argos KF and/or LS data and predict locations at user-specified time intervals (regular or irregular)

### Usage

```
fit_ssm(d, vmax = 50, ang = -1, distlim = c(2500, 5000),
        spdf = TRUE, min.dt = 60, pf = FALSE, model = "rw",
        time.step = 6, parameters = NULL, fit.to.subset = TRUE,
        optim = "nlsminb", verbose = 1, inner.control = NULL)
```

### Arguments

d	a data frame of observations including Argos KF error ellipse info
vmax	max travel rate (m/s) passed to argosfilter::sdafilter to define outlier locations
ang	angles of outlier location "spikes" - see ?argosfilter::sdafilter for details
distlim	lengths of outlier location "spikes" - see ?argosfilter::sdafilter for details
spdf	(logical) turn argosfilter::sdafilter on (default; TRUE) or off
min.dt	minimum allowable time difference between observations; dt <= min.dt will be ignored by the SSM
pf	just pre-filter the data, do not fit the ctrw (default is FALSE)
model	fit either a simple random walk ("rw") or correlated random walk ("crw") as a continuous-time process model
time.step	the regular time interval, in hours, to predict to. Alternatively, a vector of prediction times, possibly not regular, must be specified as a data.frame with id and POSIXt dates.
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)
optim	numerical optimizer to be used ("nlsminb" or "optim")
verbose	report progress during minimization; 0 for complete silence; 1 for progress bar only; 2 for minimizer trace but not progress bar
inner.control	list of control settings for the inner optimization (see ?TMB::MakeADFUN for additional details)

**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 summmary
data	an augmented sf tbl of the 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

**Examples**

```
## fit crw model to multiple individuals with Argos LS data
data(ellie)
fit <- fit_ssm(ellie, model = "rw", time.step = 24)
plot(fit$ssm[[1]])

data(rope)
fls <- fit_ssm(rope, model = "crw", time.step = 12)

## simple diagnostic plot for individual 3,
## showing predicted value time-series
plot(fls$ssm[[3]], what = "predicted")
```

---

grab

*grab tbl's by name from a foieGras fitted model object*

---

**Description**

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

**Usage**

```
grab(x, what = "fitted", as_sf = TRUE)
```

**Arguments**

x	a foieGras fitted model object
what	the tibble to be grabbed; either 'fitted', 'predicted', or 'data' (single letters can be used)
as_sf	logical; if FALSE then return a tibble with unprojected lonlat coordinates, otherwise return an sf tibble

**Value**

a tbl with all individual tbl's appended

**Examples**

```
## load example foieGras fit object (to save time)
data(fit)
## grab predicted values as an unprojected tibble
preds <- grab(fit, what = "p", as_sf = FALSE)
```

---

plot.foieGras      *plot*

---

**Description**

visualise foieGras SSM fits to track data

**Usage**

```
## S3 method for class 'foieGras'
plot(x, what = c("fitted", "predicted"),
     outlier = FALSE, ...)
```

**Arguments**

x	a foieGras fitted object
what	specify which location estimates to display on time-series plots: fitted or predicted
outlier	include all extreme outliers flagged by prefilter in plots (logical)
...	additional arguments to be ignored

**Examples**

```
## load example foieGras fit object (to save time)
data(fit)
plot(fit$ssm[[1]])
```

---

pluck	<i>Deprecated functions.</i>
-------	------------------------------

---

### Description

extract and pluck have been deprecated to avoid conflict with raster::extract, tidyr::extract, purrr::pluck. Instead use grab.

### Usage

```
pluck(...)
```

```
extract(...)
```

### Arguments

```
...          ignored
```

---

prefilter	<i>Prepare Argos data for fitting state-space model</i>
-----------	---

---

### Description

prefilter (1) determines Argos data type (LS or KF); (2) converts dates to POSIXt & identifies observations with duplicate dates; (3) orders observations in time; (4) removes duplicate observations; (5) removes observations occurring within 60 s of one another (keeps first); (6) shifts longitudes that straddle -180,180 to 0,360 and vice-versa; (7) projects lonlat coords to mercator x,y coords (in km); (8) adds location error multiplication factors based on Argos location class (for type LS); (9) uses a argosfilter::sdafilter to identify potential outlier locations (by distance only) to be ignored when fitting the ctw model

### Usage

```
prefilter(data, vmax = 50, ang = -1, distlim = c(2500, 5000),
          spdf = TRUE, min.dt = 60)
```

### Arguments

data	input data - must have 5 (LS), or 8 (KF) columns (see details)
vmax	max travel rate (m/s) - see ?argosfilter::sdafilter for details
ang	angles of outlier location "spikes" - see ?argosfilter::sdafilter for details
distlim	lengths of outlier location "spikes" - see ?argosfilter::sdafilter for details
spdf	turn speed filter on/off (logical; default is TRUE)
min.dt	minimum allowable time difference between observations; dt < min.dt will be ignored by the SSM

**Details**

called by fit\_ssm.

**Value**

an sf object with all observations passed from data and the following appended columns

keep	logical indicating whether observation should be ignored by sfilter (FALSE)
obs.type	flag indicating whether KF or LS measurement model applies
amf_x	Argos error multiplication factor for x direction
amf_y	Argos error multiplication factor for y direction
geometry	sf POINT object giving x,y coordinates in km

**Examples**

```
data(ellie)
pf <- prefilter(ellie, vmax=10, ang=c(15,25), min.dt=120)
pf
```

---

print.foieGras      *print foieGras object summary information*

---

**Description**

print foieGras object summary information

**Usage**

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

**Arguments**

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

**Examples**

```
## see summary fit output
## load example foieGras fit object (to save time)
data(fit)
fit
```



---

rope	<i>Royal penguin Argos satellite data (13 individuals)</i>
------	--

---

**Description**

Example penguin Argos tracking data.

**Format**

.RData

---

sfilter	<i>fit the state-space model to prefilter-ed data</i>
---------	---

---

**Description**

generates initial values for model parameters and unobserved states; structures data and initial values for C++ TMB template; fits state-space model; minimises the joint log-likelihood via the selected optimizer (nlminb or optim); structures and passes output object to fit\_ssm

**Usage**

```
sfilter(x, model = c("rw", "crw"), time.step = 6, parameters = NULL,
       fit.to.subset = TRUE, optim = c("nlminb", "optim"),
       verbose = FALSE, inner.control = NULL)
```

**Arguments**

x	Argos data passed through prefilter()
model	specify which SSM is to be fit: "rw" or "crw"
time.step	the regular time interval, in hours, to predict to. Alternatively, a vector of prediction times, possibly not regular, must be specified as a data.frame with id and POSIXt dates.
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)
optim	numerical optimizer to be used ("nlminb" or "optim")
verbose	report progress during minimization
inner.control	list of control settings for the inner optimization (see ?TMB::MakeADFUN for additional details)

**Details**

called by fit\_ssm. sfilter can only fit to an individual track, use fit\_ssm to fit to multiple tracks (see ?fit\_ssm).

**Examples**

```
data(ellie)
pf <- prefilter(ellie, vmax=10, ang=c(15,25), min.dt=120)
out <- sfilter(pf, model="rw", time.step=24)
```

---

wrap\_lon

*Utility functions*

---

**Description**

Utility functions

**Usage**

```
wrap_lon(lon, lon_min = -180)
```

**Arguments**

lon	a vector of longitudes
lon_min	the minimum longitude value to wrap appropriately, eg. 0 to wrap -180, 180 on to 0, 360 and -180 to wrap 0,360 on to -180,180

**Details**

called by prefilter

**Examples**

```
lon <- seq(-180,180)
lon1 <- wrap_lon(lon, 0)
range(lon)
range(lon1)
```

# Index

\*Topic **data**

ellie, 3

fit, 3

rope, 9

\*Topic **foieGras**

foieGras-package, 2

dummy, 3

ellie, 3

extract (pluck), 7

fit, 3

fit\_ssm, 4

foieGras (foieGras-package), 2

foieGras-package, 2

grab, 5

plot.foieGras, 6

pluck, 7

prefilter, 7

print.foieGras, 8

rope, 9

sfilter, 9

wrap\_lon, 10