Package: TelemetrySpace (via r-universe)

May 23, 2026

Version 1.1.0

Title Spatial point process and random field models for electronic tagging data

Description A collection of tools to fit spatial models to several types of telemetry data. Models are provided to account for the detection process when estimating individual centers of activity from acoustic telemetry data and to incorporate data from stationary test transmitters when available. Bayesian versions of models are fitted using Stan (http://mc-stan.org/). Maximum likelihood versions are fitted using Template Model Builder (https://kaskr.github.io/adcomp/index.html).

License GPL (>=3)

Depends R (>= 3.5), Rcpp (>= 0.12.17)

Imports cli, dplyr, methods, RcppParallel (>= 5.0.1), rlang, rstan (>= 2.18.1), rstantools (>= 2.4.0), stats

Suggests ggplot2, ggpubr, hexbin, knitr, rmarkdown, testthat (>= 3.0.0), tidyr

LinkingTo BH (>= 1.66.0-1), Rcpp (>= 0.12.17), RcppEigen (>= 0.3.3.4.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

Encoding UTF-8

LazyData true

ByteCompile true

SystemRequirements GNU make

NeedsCompilation yes

Biarch true

Config/testthat/edition 3

RoxygenNote 7.3.2

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

Config/pak/sysreqs make

Repository https://mhpob.r-universe.dev

Date/Publication 2025-08-26 12:33:47 UTC

RemoteUrl https://github.com/meganwinton/TelemetrySpace

RemoteRef HEAD

RemoteSha 62494d08afae0521d9ddacf80431a7527b931e4e

Contents

`TelemetrySpace-package`

The 'TelemetrySpace' package.

Description

Spatial point process and random field models for electronic tagging data

Author(s)

Maintainer: Michael O'Brien <mike@obrien.page> (ORCID)

Authors:

- Megan Winton <megan.winton@gmail.com> (ORCID)

Other contributors:

- Benjamin L. Hlina <benjamin.hlina@gmail.com> (ORCID) [contributor]

References

Stan Development Team (2017). RStan: the R interface to Stan. R package version 2.16.2. http://mc-stan.org

`COA_Standard` *Fits a Bayesian Spatial Point Process model to estimate individual centers of activity from acoustic telemetry data using Stan*

Description

Fits a Bayesian Spatial Point Process model to estimate individual centers of activity from acoustic telemetry data using Stan

Usage

```
COA_Standard(nind, nrec, ntime, ntrans, y, recX, recY, xlim, ylim, ...)
```

Arguments

`nind`	Number of tagged individuals
`nrec`	Number of receivers
`ntime`	Number of time steps
`ntrans`	Number of expected transmissions per tag per time interval
`y`	Array of detection data, where row = individual, column = receiver, and matrix = time step
`recX`	Receiver coordinates in the east-west direction (should be projected and scaled for computational efficiency)
`recY`	Receiver coordinates in the north-south direction (should be projected and scaled for computational efficiency)
`xlim`	East-west boundaries of spatial extent (receiver array + buffer)
`ylim`	North-south boundaries of spatial extent (receiver array + buffer)
`...`	Additional arguments passed to `sampling` from `rstan`. This can include setting `chains`, `iter`, `warmup`, and `control`. Please see `rstan::sampling` for more info.

Value

COA_Standard returns an object of class `stanfit` returned by `rstan::sampling`. See the `rstan` package documentation for details.

This function returns a list containing the following components: 1) a summary of the detection function parameters; 2) the time required for model fitting; 3) the estimated COAs for each individual in each time step and 95 percent credible interval; and 4) a dataframe containing values for each parameter and latent parameter from chain iterations. These can be used to plot posterior distributions and the credible interval around each estimated COA.

See Also

`rstan::sampling()`

COA_TagInt *Fits a test-tag integrated Bayesian Spatial Point Process model to estimate individual centers of activity from acoustic telemetry data using Stan*

Description

Fits a test-tag integrated Bayesian Spatial Point Process model to estimate individual centers of activity from acoustic telemetry data using Stan

Usage

```
COA_TagInt(
  nind,
  nrec,
  ntime,
  ntest,
  ntrans,
  y,
  test,
  recX,
  recY,
  xlim,
  ylim,
  testX,
  testY,
  ...
)
```

Arguments

nind	Number of tagged individuals
nrec	Number of receivers
ntime	Number of time steps
ntest	Number of test tags
ntrans	Number of expected transmissions per tag per time interval
y	Array of detection data, where row = individual, column = receiver, and matrix = time step
test	Array of test tag detection data, where row = individual tag, column = receiver, and matrix = time step
recX	Receiver coordinates in the east-west direction (should be projected and scaled for computational efficiency)
recY	Receiver coordinates in the north-south direction (should be projected and scaled for computational efficiency)

xlim	East-west boundaries of spatial extent (receiver array + buffer)
ylim	North-south boundaries of spatial extent (receiver array + buffer)
testX	Test tag coordinates in the east-west direction (should be projected and scaled for computational efficiency)
testY	Test tag coordinates in the north-south direction (should be projected and scaled for computational efficiency)
...	Additional arguments passed to sampling from rstan. This can include setting chains, iter, warmup, and control. Please see rstan::sampling for more info.

Value

COA_TagInt_Bayes returns an object of class stanfit returned by rstan::sampling. See the 'rstan' package documentation for details.

This function returns a list containing the following components: 1) a summary of the detection function parameters; 2) the time required for model fitting; 3) time-varying detection probabilites for each receiver; 4) the estimated COAs for each individual in each time step and 95 percent credible interval; and 5) a dataframe containing values for each parameter and latent parameter from chain iterations. These can be used to plot posterior distributions and the credible interval around each estimated COA.

See Also

rstan::sampling()

COA_TimeVarying *Fits a test-tag integrated Bayesian Spatial Point Process model to estimate individual centers of activity from acoustic telemetry data using Stan*

Description

Fits a test-tag integrated Bayesian Spatial Point Process model to estimate individual centers of activity from acoustic telemetry data using Stan

Usage

```
COA_TimeVarying(nind, nrec, ntime, ntrans, y, recX, recY, xlim, ylim, ...)
```

Arguments

nind	Number of tagged individuals
nrec	Number of receivers
ntime	Number of time steps
ntrans	Number of expected transmissions per tag per time interval

y	Array of detection data, where row = individual, column = receiver, and matrix = time step
recX	Receiver coordinates in the east-west direction (should be projected and scaled for computational efficiency)
recY	Receiver coordinates in the north-south direction (should be projected and scaled for computational efficiency)
xlim	East-west boundaries of spatial extent (receiver array + buffer)
ylim	North-south boundaries of spatial extent (receiver array + buffer)
...	Additional arguments passed to sampling from rstan. This can include setting chains, iter, warmup, and control. Please see rstan::sampling for more info.

Value

COA_TimeVarying returns an object of class stanfit returned by rstan::sampling. See the 'rstan' package documentation for details.

This function returns a list containing the following components: 1) a summary of the detection function parameters; 2) the time required for model fitting; 3) time-varying detection probabilites for each receiver; 4) the estimated COAs for each individual in each time step and 95 percent credible interval; and 5) a dataframe containing values for each parameter and latent parameter from chain iterations. These can be used to plot posterior distributions and the credible interval around each estimated COA.

---

distf *Calculate Euclidean distance between receivers and activity centers*

Description

Calculate Euclidean distance between receivers and activity centers

Usage

```
distf(x, y)
```

Arguments

x	Data frame or matrix containing 2-dimensional coordinates
y	Data frame or matrix containing 2-dimensional coordinates

Value

'distf' returns a matrix containing the Euclidean distance between each location in dataframe x with that in dataframe y

example_extent *Example array extent*

Description

Example array extent used in each model

Usage

```
example_extent
```

Format

A data frame with two variables and two row: `ylim` is the minimum and maximum extent on the y-axis (i.e., latitudinal) for the array and `xlim` is the minimum and maximum extent on the x-axis (i.e., longitudinal) for the array.

fishdat *Black sea bass detection data*

Description

Detection data from a tagged black sea bass aggregated to the hour.

Usage

```
fishdat
```

Format

A data frame with five variables: `Station` Receiver ID, `Transmitter` Transmitter ID, `east` East-West coordinate, `north` North-South coordinate, `hour` Hour of monitoring.

model_param_ex *Example model parameters*

Description

Example model parameters for `COA_standard()`

Usage

```
model_param_ex
```

Format

A data frame with four variables and one row: `nind` which is the number of individuals, `nrec` is the number of receivers in the array, `tsteps` is the number of time steps used in the example, and `ntrans` is the number of expected transmissions within a time step.

rlocs *Receiver locations from a black sea bass array*

Description

Projected and scaled receiver coordinates. Scaling is recommended to reduce computation time and prevent convergence issues.

Usage

```
rlocs
```

Format

A data frame with three variables: `Station` Receiver ID, `east` East-West coordinate, `north` North-South coordinate.

testdat *Stationary test transmitter data*

Description

Detection data from a stationary, known-location test tag aggregated to the hour.

Usage

```
testdat
```

Format

A data frame with five variables: `Station` Receiver ID, `Transmitter` Transmitter ID, `east` East-West coordinate, `north` North-South coordinate, `hour` Hour of monitoring.

testloc *Location of a stationary test transmitter placed in the black sea bass array*

Description

Projected and scaled as for the receiver coordinates.

Usage

```
testloc
```

Format

A data frame with two variables: `east` East-West coordinate, `north` North-South coordinate.

testY *Counts of detection per time steps for test tag*

Description

Array of counts of detection per time step per receivers for test tag

Usage

```
testY
```

Format

An array with dimensions of 1 by 10 (number of time steps) by 30 number of receivers.

Y *Counts of detection per time steps*

Description

Array of counts of detection per time step per receivers

Usage

```
Y
```

Format

An array with dimensions of 1 by 10 (number of time steps) by 30 number of receivers.

Index