Package: TelemetrySpace (via r-universe)

October 13, 2026

Version 1.3.3

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 (>= 4.1), Rcpp (>= 0.12.17)

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

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

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

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

URL https://telemetryspace.trackyverse.org,
https://github.com/trackyverse/TelemetrySpace

BugReports https://github.com/trackyverse/TelemetrySpace/issues

Config/roxygen2/version 8.0.0

RoxygenNote 8.0.0

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev make libicu-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

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

Date/Publication 2026-10-13 13:46:01 UTC

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

RemoteRef HEAD

RemoteSha fbc983a36f811df95f0909d8b863f43a146616ce

Contents

TelemetrySpace-package . . . 3
build_functions . . . 3
COA_Standard . . . 6
COA_TagInt . . . 7
COA_TimeVarying . . . 9
distf . . . 10
example_extent . . . 11
fishdat . . . 11
model_param_ex . . . 12
ps . . . 12
ps_det_example . . . 13
ps_det_test_tag . . . 13
ps_rec_loc . . . 14
ps_test_tag_loc . . . 14
rlocs . . . 15
testdat . . . 15
testloc . . . 16
testY . . . 16
Y . . . 17

Index **18**

`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:

- Michael O'Brien <mike@obrien.page> (ORCID)
- Megan Winton <megan.winton@gmail.com> (ORCID)
- Benjamin L. Hlina <benjamin.hlina@gmail.com> (ORCID)

References

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

See Also

Useful links:

- https://telemetryspace.trackyverse.org
- https://github.com/trackyverse/TelemetrySpace
- Report bugs at https://github.com/trackyverse/TelemetrySpace/issues

`build_functions` *Build Functions*

Description

These functions structure and build data components that are needed by the model. Each function will either return an object that is properly built and structured for 'Stan" or will produce an object to be latter used in the pre-process phase of analysis.

Usage

```
build_aeqd(array_sf)

build_bbox(coord_df, buffer = NULL)

build_counts(df, nrec, rec_id, rec_names = NULL)

build_init(coord_df, nind, tstep)

build_ntrans(df, type = c("mean", "min", "max", "custom"), custom_delay = NULL)

build_pixel_grid(bnd_sf, res, crs)

build_rec_coords(obj_sf)

build_time_bin(df, unit = "1 hour")

build_tstep(x)
```

Arguments

array_sf	the receiver array as an sf object.
coord_df	a data.frame that contains two columns named recX and recY created by build_rec_coords().
buffer	a numerical value to set the buffer. Defaults to 1. Considering the default Azimuthal Equidistant projection is km, 1 represents a 1 km buffer.
df	a data.frame that contains the following column names tag_serial_no
nrec	a numerical value that is number of receivers in the telemetry array.
rec_id	a vector that matches the length of nrec and needs to be index of the receivers in the telemetry array.
rec_names	an optional vector that contains the station names of the receivers. If not supplied it will default to using rec_id
nind	a numerical value to the number of individuals.
tstep	a numerical value to the number of time steps.
type	can either be mean, min, max, or custom. This will change how the number of transmissions that are expected within a time bin are calculated. When set to mean (the deafult), the function will calculate the mean time delay (s) between the minimumm and maximum delay (s), while min and max, will use the minimumm or maximum delay value, respectively. If specifying custom (see custom_delay), the user can enter the delay of their choosing. Adjusting the delays used can useful becaue the number of detections within a time bin can exceed the number of transmissions expected wtihin the time bin which will cause the model to fail.
custom_delay	Only needed when type is set to "custom". When supplied, this argument, which is a numeric will allow a custom value to be used for the tag delays.

bnd_sf	a sf object that is boundary that is desired to impose
res	the resolution desired.
crs	the Azimuthal Equidistant projection desired. If the supplied projection string does not contain "+proj=aeqd". The function will error.
obj_sf	a sf object that the receiver locations as POINT geometry. The sf object has to be in Azimuthal Equidistant projection.
unit	a character that is the unit desired to bin. Default is "1 hour". See lubridate::floor_date()'s unit argument for more details.
x	a 3-dimensional count array.

Details

build_aeqd()- Azimuthal Equidistant projection is needed by the model which relies on creating a centroid and creating equal distances from the centroid. This function quickly creates the project string needed to transform an exisiting crs to Azimuthal Equidistant projection.

build_bbox()- builds the boundary box of receiver array needed by the model.

build_counts() - builds a 3-dimensional count array. The models need the counts at each recevier for each time bin for each individual structure in 3-dimensional array. This functions takes in a detection data.frame, creates a count data.frame and then transforms it into a 3-dimensional array that will be pased to the Stan model.

build_init()- builds the initial values to supply to the model. Supplying these values can helps the model know where to intially start.

build_ntrans() - builds the nubmer of transmissions to be expected within a given time bin.

build_pixel_grid() - builds barrier grid for the model. We need to convert the boundary into pixels that can be used to recongize where to estimate detection probablity. The boundary that is supplied needs to be in UTMs, with the returned values being in Azimuthal Equidistant projection. To build this projection see build_aeqd(). When supplying the desired resolution remember that this is in m so a value of 1 would be quite small while of 1000 is 1 km which makes a less dense grid.

build_rec_coords() - builds receiver coordinates as the models need the easting and northing (i.e., x and y) coordinates of the receivers.

build_time_bin()- builds and adds time bins to the detection data.frame.

build_tstep() - builds the number of total time steps that exist whithin the supplied 3-dimensional count array.

Value

build_aeqd()- retruns a vector containing the site specific projection string for the array to be able to transform the crs.

build_bbox()- returns a data.frame containing two columns named xlim and ylim which are the minimum and maximum values +/- a buffer for x and y.

build_counts() - returns a 3-dimensional array containing the number of detections for the following dimensions, the number of invividuals, by the number of time bins, by the number of receivers.

build_init()- returns a list containing two matrices named sx and sy. Each matrix is the mean of either recX or recY produced by build_rec_coords() by the number of individuals by the number of timesteps.

build_ntrans() - returns a single value vector.

build_pixel_grid() - returns a list contain the the number of pixels n_pixel, the pixel x coordinates (pix_x) and the pixel y coordinates (pix_y).

build_rec_coords() - a data.frame containing two columns named recX and recY which are the receiver locations transformed into Azimuthal Equidistant projection.

build_time_bin()- returns a data.frame that has had the columns time_bin and timeadded. time is an index value of time_bin and is needed by the model. This will further be used by build_counts().

build_tstep() - a numerical value that is the number of timesteps.

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,
  decay = "gaussian",
  ndraws = NULL,
  ...
)
```

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 = time step, and matrix = receiver
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).
decay	desired decay function. Currently one of "gaussian" or "logistic". Default is "gaussian".
ndraws	to be passed to generated_quantities. Changes the number of draws. Default is 10.
...	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,
  decay = "gaussian",
  ndraws = NULL,
  ...
)
```

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 = time step, and matrix = receiver
test	Array of test tag detection data, where row = individual tag, column = time step, and matrix = receiver
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)
decay	desired decay function. Currently one of "gaussian" or "logistic". Default is "gaussian".

ndraws	to be passed to generated_quantities. Changes the number of draws. Default is 10.
...	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 time-varying Bayesian Spatial Point Process model to estimate individual centers of activity from acoustic telemetry data using Stan*

Description

Fits a time-varying 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,
  decay = "gaussian",
  ndraws = NULL,
  ...
)
```

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 = time step, and matrix = receiver
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)
decay	desired decay function. Currently one of "gaussian" or "logistic". Default is "gaussian".
ndraws	to be passed to `generated_quantities`. Changes the number of draws. Default is 10.
...	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.

ps *Parry Sound* `sf` *object*

Description

A `MULTIPOLYGON` `sf` object that is Parry Sound which is a large embayment on Georgian Bay, Lake Huron

Usage

```
ps
```

Format

A `sf` object with a single geomety feature. Current coordinate reference system is WGS 84.

ps_det_example *Detection* data.frame *for a tagged Lake Trout*

Description

A data.frame that contains detections data for a tagged Lake Trout in Parry Sound which is a large embayment on Georgian Bay, Lake Huron

Usage

```
ps_det_example
```

Format

A data.frame with 5 columns and 577 rows.

detection_timestamp_utc The detection timestamp as POSIXct with a tz of UTC

station_no The receiver station number

tag_serial_no The tag serial number

min_delay The minimum delay between transmissions in seconds

max_delay The maximum delay between transmissions in seconds

ps_det_test_tag *Detection* data.frame *for a test tag*

Description

A data.frame that contains detections data for aan internal transmitter of a VR2AR deployed in Parry Sound which is a large embayment on Georgian Bay, Lake Huron

Usage

```
ps_det_test_tag
```

Format

A data.frame with 7 columns and 200 rows.

detection_timestamp_utc The detection timestamp as POSIXct with a tz of UTC

station_no The receiver station number the tag was detected on

tag_serial_no The serial number of the VR2AR receiver

tag_id The tag id value from of the VR2AR receiver

tag_station_no The station number of the tag

min_delay The minimum delay between transmissions in seconds

max_delay The maximum delay between transmissions in seconds

ps_rec_loc *Parry Sound - Receiver Location*

Description

A `data.frame` that contains the locations of 80 acoustic telemetry receivers deployed in Parry Sound which is a large embayment on Georgian Bay, Lake Huron

Usage

```
ps_rec_loc
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 80 rows and 3 columns.

Details

A `data.frame` with 3 columns and 80 rows.

station_no The station number of the receiver

deploy_lat The latitude of the deployed receiver

deploy_long The longitude of the deployed receiver

ps_test_tag_loc *Parry Sound - Test Tag Locations object*

Description

A `data.frame` that contains the `tag_station_no` and the deployment latitude and longitude of the test tag deployed in Parry Sound which is a large embayment on Georgian Bay, Lake Huron

Usage

```
ps_test_tag_loc
```

Format

An object of class `data.table` (inherits from `data.frame`) with 1 rows and 3 columns.

Details

A `data.frame` with 3 columns and 1 rows.

station_no The station number of the tag

deploy_lat The latitude of the deployed tag

deploy_long The longitude of the deployed tag

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 receiver for the test tag. Originally with dimensions of 1 (individual) x 30 (receivers) x 10 (time steps), changed to 1 x 10 x 30 on 2026-05-13 for computational efficiency.

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 receiver. Originally with dimensions of 1 (individual) x 30 (receivers) x 10 (time steps), changed to 1 x 10 x 30 on 2026-05-13 for computational efficiency.

Usage

```
Y
```

Format

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

Index

∗ **datasets**
  example_extent, 11
  fishdat, 11
  model_param_ex, 12
  ps, 12
  ps_det_example, 13
  ps_det_test_tag, 13
  ps_rec_loc, 14
  ps_test_tag_loc, 14
  rlocs, 15
  testdat, 15
  testloc, 16
  testY, 16
  Y, 17

build_aeqd *(*build_functions*)*, 3
build_bbox *(*build_functions*)*, 3
build_counts *(*build_functions*)*, 3
build_functions, 3
build_init *(*build_functions*)*, 3
build_ntrans *(*build_functions*)*, 3
build_pixel_grid *(*build_functions*)*, 3
build_rec_coords *(*build_functions*)*, 3
build_time_bin *(*build_functions*)*, 3
build_tstep *(*build_functions*)*, 3

COA_Standard, 6
COA_TagInt, 7
COA_TimeVarying, 9

distf, 10

example_extent, 11

fishdat, 11

model_param_ex, 12

ps, 12
ps_det_example, 13
ps_det_test_tag, 13
ps_rec_loc, 14
ps_test_tag_loc, 14

rlocs, 15
rstan::sampling(), *7*, *9*

TelemetrySpace
        *(*TelemetrySpace-package*)*, 3
TelemetrySpace-package, 3
testdat, 15
testloc, 16
testY, 16

Y, 17