Package 'movedesign'

June 24, 2025

Title Study Design Toolbox for Movement Ecology Studies

Version 0.3.1

Maintainer Ines Silva <i.simoes-silva@hzdr.de>

Description Toolbox and 'shiny' application to help researchers design movement ecology studies, focusing on two key objectives: estimating home range areas, and estimating fine-scale movement behavior, specifically speed and distance traveled. It provides interactive simulations and methodological guidance to support study planning and decision-making. The application is described in Silva et al. (2023) <doi:10.1111/2041-210X.14153>.

License GPL (>= 3)

URL https://ecoisilva.github.io/movedesign/, https://ecoisilva.r-universe.dev/movedesign

BugReports https://github.com/ecoisilva/movedesign/issues

Depends R (>= 3.5.0)

Imports bayestestR, bsplus, combinat, config (>= 0.3.1), crayon, ctmm (>= 0.6.1), data.table, dplyr, fontawesome, gdtools, ggiraph, ggplot2, ggpubr, ggtext, golem (>= 0.3.2), grDevices, gsl, lubridate, parallel, parsedate, quarto, reactable, rintrojs, rlang, scales, shiny (>= 1.7.1), shinyalert, shinybusy, shinydashboard, shinydashboardPlus, shinyFeedback, shinyjs, shinyWidgets, stats, stringr, terra, tidyr, utils, viridis

Suggests knitr, rmarkdown, shinytest2

VignetteBuilder knitr, quarto

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.3.2

NeedsCompilation no

Author Ines Silva [cre, aut, cph] (ORCID:
<https://orcid.org/0000-0003-4850-6193>)

Repository CRAN

Date/Publication 2025-06-24 08:50:01 UTC

Contents

`estimate_hr` *Estimate home range*

Description

Estimates home range areas with the Autocorrelated Kernel Density Estimator (AKDE) for each simulated movement dataset using the fitted movement models.

Usage

```
estimate_hr(rv)
```

Arguments

`rv`	A reactive values list containing: • `simList` - A list of simulated movement datasets. • `simfitList` - A list of fitted movement models corresponding to `simList`.

Details

The function applies `ctmm::akde()` to estimate home range areas while handling potential warnings and errors gracefully. Any failed computations return `NULL`.

Value

A named list of `ctmm` objects, one per simulation.

estimate_speed *Estimate movement speed*

Description

Estimates movement speed using continuous-time speed and distance (CTSD) for each simulated movement dataset using the corresponding fitted movement model.

Usage

```
estimate_speed(rv)
```

Arguments

rv	A reactive values list containing: • simList - A list of simulated movement datasets. • simfitList - A list of fitted movement models corresponding to simList.

Details

The function applies ctmm::speed() to estimate movement speed while handling potential warnings and errors gracefully.

Value

A named list of ctmm objects, one per simulation.

fitting_model *Fit continuous-time movement models*

Description

This function fits continuous-time movement models to simulated location data using the ctmm package. It estimates movement parameters for each simulated trajectory, optionally running in parallel for efficiency.

Usage

```
fitting_model(
  obj,
  set_target = c("hr", "ctsd"),
  .dur = NULL,
  .dti = NULL,
  .tau_p = NULL,
  .tau_v = NULL,
```

```
  .error_m = NULL,
  .check_sampling = FALSE,
  .rerun = FALSE,
  .parallel = TRUE,
  .trace = FALSE
)
```

Arguments

obj	A list of simulated movement datasets.
set_target	A character vector indicating the research target(s). Options: • "hr" - Home range estimation. • "ctsd" - Speed and distance estimation.
.dur	Numeric, sampling duration of the simulated data (required if .check_sampling = TRUE).
.dti	Numeric, sampling interval of simulated data (required if .check_sampling = TRUE).
.tau_p	List, position autocorrelation timescale (optional).
.tau_v	List, velocity autocorrelation timescale (optional).
.error_m	Numeric, if simulating a dataset with location error (in meters).
.check_sampling	Logical; if TRUE, checks if the sampling schedule is optimal for ctmm.fit().
.rerun	Logical; if TRUE, re-runs model selection if effective sample sizes fall below threshold.
.parallel	Logical; if TRUE, enables parallel computation for efficiency. Default is TRUE.
.trace	Logical; if TRUE, prints additional information.

Details

The function first generates initial parameter estimates using ctmm::ctmm.guess(). It then selects the best movement model for each simulation using par.ctmm.select(). The function ensures that each fitted model is centered at the origin (x = 0, y = 0) before returning.

Value

A list of fitted movement models, one per simulation.

fixrates *Fix rates of animal tracking devices.*

Description

Dataset with a list of fix rates typically available for animal tracking devices.

Usage

```
fixrates
```

Format

An object of class `"data.frame"`, with 40 rows and seven variables: `dti_notes`, `dti`, `highlight`, `frq`, and `frq_hrs`.

dti_notes Sampling interval as text (e.g., "1 fix every month")

dti Sampling interval in seconds

frq Sampling frequency in seconds

frq_hrs Sampling frequency in hours

highlighted Highlight a group of commonly-used GPS fix rates ...

movmods *Table of movement processes.*

Description

Table listing all current continuous-time movement processes used within 'ctmm'.

Usage

```
movmods
```

Format

An object of class `"data.frame"`, with five rows and six variables: `name`, `name_short`, `tau_p`, `tau_v`, `hrange`, and `pars`.

References

Calabrese et al. (2016) doi:10.1111/2041-210X.12559

name Movement process names (e.g., "Ind. Ident. Distr. (IID)")

name_short Abbreviated movement process names

tau_p A Boolean denoting whether or not the movement process includes position autocorrelation

tau_v A Boolean denoting whether or not the movement process includes velocity autocorrelation

hrange A Boolean denoting whether or not the movement process includes range residency

pars Character variable, to display autocorrelation parameters in HTML ...

`run_app` *Run movedesign' R Shiny Application*

Description

Run movedesign' R Shiny Application

Usage

```
run_app(
  onStart = NULL,
  options = list(),
  enableBookmarking = NULL,
  uiPattern = "/",
  ...
)
```

Arguments

`onStart`	A function that will be called before the app is actually run. This is only needed for `shinyAppObj`, since in the `shinyAppDir` case, a `global.R` file can be used for this purpose.
`options`	Named options that should be passed to the `runApp` call (these can be any of the following: "port", "launch.browser", "host", "quiet", "display.mode" and "test.mode"). You can also specify `width` and `height` parameters which provide a hint to the embedding environment about the ideal height/width for the app.
`enableBookmarking`	Can be one of `"url"`, `"server"`, or `"disable"`. The default value, `NULL`, will respect the setting from any previous calls to `enableBookmarking()`. See `enableBookmarking()` for more information on bookmarking your app.
`uiPattern`	A regular expression that will be applied to each `GET` request to determine whether the `ui` should be used to handle the request. Note that the entire request path must match the regular expression in order for the match to be considered successful.
`...`	arguments to pass to golem_opts. See `?golem::get_golem_options` for more details.

Value

No return value, called for side effects.

`run_meta` *Running hierarchical models*

Description

This function wraps around the `run_meta_resampled` function to run hierarchical models (with no resamples) for a quick evaluation.

Usage

```
run_meta(
  rv,
  set_target = c("hr", "ctsd"),
  subpop = FALSE,
  trace = FALSE,
  iter_step = 2,
  .automate_seq = FALSE,
  .only_max_m = FALSE,
  .lists = NULL
)
```

Arguments

`rv`	A list containing outputs, settings and data objects. Must not be NULL.
`set_target`	Character. Research target: `"hr"` for home range or `"ctsd"` for speed & distance.
`subpop`	Logical. If TRUE, will run meta-analyses with groups. Default is FALSE.
`trace`	Logical. If TRUE, prints progress messages. Default is FALSE.
`iter_step`	Numeric. The size of each iteration step. Default is 2.
`.automate_seq`	Logical. If TRUE, overwrites sequence automatically to improve plot readability. Default is FALSE.
`.only_max_m`	Logical. If TRUE, will only run the maximum number of individuals. Default is FALSE.
`.lists`	A list containing already created meta inputs. Default is NULL.

Value

The outputs of the `run_meta_resampled` function for a single combination.

`run_meta_loocv` *Running LOOCV on hierarchical model outputs*

Description

This function runs hierarchical models on movement tracking data, for mean home range area (AKDE) or continuous-time speed and distance (CTSD) estimates for a sampled population. It leverages the `ctmm` R package, specifically the `meta()` function, to obtain population-level mean parameters. This function helps to assess outputs via leave-one-out cross-validation (LOOCV).

Usage

```
run_meta_loocv(
  rv,
  set_target = c("hr", "ctsd"),
  subpop = FALSE,
  trace = FALSE,
  .progress = FALSE,
  .only_max_m = TRUE,
  .lists = NULL
)
```

Arguments

`rv`	A list containing outputs, settings and data objects. Must not be NULL.
`set_target`	Character. Research target: `"hr"` for home range or `"ctsd"` for speed & distance.
`subpop`	Logical. If TRUE, will run meta-analyses with groups. Default is FALSE.
`trace`	Logical. If TRUE, prints progress messages. Default is FALSE.
`.progress`	Logical. If TRUE, will display a progress bar. Default is FALSE.
`.only_max_m`	Logical. If TRUE, will only run the maximum number of individuals. Default is FALSE.
`.lists`	A list containing already created meta inputs. Default is NULL.

Value

A data frame containing meta-analysis outputs, including estimates, errors, confidence intervals, and group information.

Author(s)

Inês Silva <i.simoes-silva@hzdr.de>

Examples

```
if(interactive()) {
run_meta_loocv(rv, set_target = "hr")
}
```

`run_meta_resampled` *Running hierarchical model meta-analyses (with resamples)*

Description

This function performs a meta-analysis on movement tracking data, for mean home range area (AKDE) or continuous-time speed and distance (CTSD) estimates for a sampled population. It leverages the `ctmm` R package, specifically the `meta()` function, to obtain population-level mean parameters. This function helps to evaluate the significance of results under combination testing.

Usage

```
run_meta_resampled(
  rv,
  set_target = c("hr", "ctsd"),
  subpop = FALSE,
  random = FALSE,
  max_samples = 100,
  iter_step = 2,
  trace = FALSE,
  .automate_seq = FALSE,
  .only_max_m = FALSE,
  .lists = NULL
)
```

Arguments

`rv`	A list containing outputs, settings and data objects. Must not be NULL.
`set_target`	Character. Research target: `"hr"` for home range or `"ctsd"` for speed & distance.
`subpop`	Logical. If TRUE, will run meta-analyses with groups. Default is FALSE.
`random`	Logical. If TRUE, samples random subsets of individuals. Default is FALSE.
`max_samples`	Integer. Maximum number of resamples when `random = TRUE`. Must be positive. Default is 100.
`iter_step`	Numeric. The size of each iteration step. Default is 2.
`trace`	Logical. If TRUE, prints progress messages. Default is FALSE.
`.automate_seq`	Logical. If TRUE, overwrites sequence automatically to improve plot readability. Default is FALSE.
`.only_max_m`	Logical. If TRUE, will only run the maximum number of individuals. Default is FALSE.
`.lists`	A list containing already created meta inputs. Default is NULL.

Value

A data frame containing meta-analysis outputs, including estimates, errors, confidence intervals, and group information.

Author(s)

Inês Silva <i.simoes-silva@hzdr.de>

Examples

```
if(interactive()) {
run_meta_resampled(rv, set_target = "hr")
}
```

---

simulating_data *Simulate Movement Data from Fitted Models*

Description

This function generates simulated location data using movement models. The function supports both single and grouped simulations based on whether the data is simulated or derived from an emulated or fitted model.

Usage

```
simulating_data(rv)
```

Arguments

rv	A reactive values list containing:

- dur - A list specifying the sampling duration (value and unit).
- dti - A list specifying the time interval between locations (value and unit).
- data_type - A character string indicating data type.
- is_emulate - Logical; if TRUE, the function generates an emulated model.
- modList - A list of fitted movement models for simulation.
- meanfitList - A list of a mean model for emulation.
- grouped - Logical; if TRUE, the simulation considers grouped movement models.
- which_meta - A character vector indicating whether to compare models.
- tau_p, tau_v, sigma, mu - Lists of movement model parameters.
- seed0 - An integer used for random seed initialization.

Details

The function first constructs a time sequence based on the provided duration and interval. If the data is fully simulated from scratch (not conditioned on existing data), it retrieves movement model(s) from `rv$modList`. Otherwise, it either emulates a model using `rv$meanfitList` and a random seed or constructs a model from movement parameters.

If estimate comparisons are enabled (via `which_meta`), two models are prepared. The function then runs `ctmm::simulate()` to generate simulated movement data. The resulting trajectories are pseudonymized before returning.

Value

A list containing one or two simulated movement datasets (depending on grouping):

- If `grouped = FALSE`, returns a list with a single simulated dataset.
- If `grouped = TRUE`, returns a list with two simulated datasets (for groups `A` and `B`).

Index