Package: upriver (via r-universe)

October 31, 2024

Type Package

Title An R package for in-river fish migration and timing calculation

Version 0.1

Date 2015-02-26

Author person(``Bryce'', ``Mecum'', ``brycemecum@gmail.com'', c(``aur'', ``cre''))

Maintainer Bryce Mecum <brycemecum@gmail.com>

Description An R package for in-river fish migration and timing calculation.

License file LICENSE

LazyData TRUE

Suggests testthat

RoxygenNote 6.0.1

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

RemoteUrl https://github.com/amoeba/upriver

RemoteRef master

RemoteSha ffdd3a787d6364554fc30ea4b52ac61cb424055e

Contents

`expand_parameters` *Expand movement parameters*

Description

Expands the movement parameters to include a 0th and n+1th reach.

Usage

```
expand_parameters(parameters)
```

Arguments

`parameters`	(list) Movement parameters (names: `rates`, `distances`)

Value

(list) Expanded parameters

`median_timing` *Calculate median timing*

Description

Calculate median timing given a set of parameters

Usage

```
median_timing(location, arrival, parameters, arrival_position = 0)
```

Arguments

`location`	(numeric) The location at which you want to calculate median timing.
`arrival`	(data.frame) A data.frame of daily arrivals by proportion. See details.
`parameters`	(list) See `positions` for details.
`arrival_position`	(numeric) Defaults 0. Where to start movement.

Details

#' `arrival` should be a `data.frame` describing a set of daily arrivals of fish at `arrival_position` as a series of daily proportions. It should have a column `day` and a corresponding column `proportion`, which should sum to one.

Examples

```
median_timing(1000,
           data.frame(day=0:40, proportion=dnorm(-20:20, 0, 5)/sum(dnorm(-20:20, 0, 5))),
              list(rates = 50, distances = 1000))
```

---

percentile_timing *Calculate an arbitrary timing percentile*

Description

Calculate an arbitrary timing percentile given a set of parameters

Usage

```
percentile_timing(percentile, location, arrival, parameters,
  arrival_position = 0)
```

Arguments

`percentile`	(numeric) An arbitrary timing percentile (between 0 and 1)
`location`	(numeric) The location at which you want to calculate median timing.
`arrival`	(data.frame) A data.frame of daily arrivals by proportion. See details.
`parameters`	(list) See `positions` for details.
`arrival_position`	(numeric) Defaults 0. Where to start movement.

Details

#' `arrival` should be a `data.frame` describing a set of daily arrivals of fish at `arrival_position` as a series of daily proportions. It should have a column day and a corresponding column `proportion`, which should sum to one.

Examples

```
percentile_timing(0.75,
                  1000,
            data.frame(day=0:40, proportion=dnorm(-20:20, 0, 5)/sum(dnorm(-20:20, 0, 5))),
                  list(rates = 50, distances = 1000))
```

`positions` *Calculate positions after some time.*

Description

Calculate positions after `ndays` has elapsed, given the movement parameters provided in `parameters`.

Usage

```
positions(ndays, parameters)
```

Arguments

`ndays`	(numeric) The number of days to calculate positions for.
`parameters`	(list) List with names reaches, rates, distances. See details.

Details

`parameters` should be a list with three elements, `rates`, and `distances`, each of equal size. `rates` should be a numeric vector of reach-specific daily movement rates, in whatever unit your analysis needs. `distances` should be a numeric vector of reach-specific reach lengths (end-to-end), in the (ideally) the same or compatible units to the units used in `rates`

Examples

```
# Simple upriver movement
positions(10, list(rates = 50, distances = 1000))
# Movement with reach-to-reach variation
positions(10,
          list(rates = c(1, 2, 3, 4, 5),
          distances = c(20, 20, 20, 20, 20)))
```

`timings` *Calculate run timings*

Description

Calculates run timing given a set of parameters.

Usage

```
timings(location, arrival, parameters, arrival_position = 0)
```

Arguments

`location`	(numeric) The location at which to calculate to run timing.
`arrival`	(data.frame) A data.frame of daily arrivals by proportion. See details.
`parameters`	(list) See `positions` for details.
`arrival_position`	(numeric) Defaults 0. Where to start movement.

Details

`arrival` should be a `data.frame` describing a set of daily arrivals of fish at `arrival_position` as a series of daily proportions. It should have a column `day` and a corresponding column `proportion`, which should sum to one.

Examples

```
timings(1000,
        data.frame(day=0:40, proportion = dnorm(-20:20, 0, 5)/sum(dnorm(-20:20, 0, 5))),
        list(rates = 50, distances = 1000))
```

Index