

Package ‘IxPopDyMod’

October 23, 2023

Title Framework for Tick Population and Infection Modeling

Version 0.3.0

Maintainer Myles Stokowski <mylesstokowski@gmail.com>

Description Code to specify, run, and then visualize and analyze the results of Ixodidae (hard-bodied ticks) population and infection dynamics models. Such models exist in the literature, but the source code to run them is not always available. 'IxPopDyMod' provides an easy way for these models to be written and shared.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

URL <https://github.com/dallenmidd/IxPopDyMod>

Depends R (>= 3.2.4)

Suggests covr, knitr, rmarkdown, testthat (>= 3.0.0)

Imports magrittr, stats, checkmate

Language en-US

Config/testthat/edition 3

NeedsCompilation no

Author Myles Stokowski [aut, cre],
David Allen [aut] (<<https://orcid.org/0000-0002-0712-9603>>)

Repository CRAN

Date/Publication 2023-10-23 20:20:02 UTC

R topics documented:

annual_growth_rate	2
config	3
config_ex_1	4
config_ex_2	5

constant_fun	5
density_fun	6
expo_fun	6
feed_fun	7
find_n_feed	8
format.predictor_spec	9
get_pred_from_table	9
growth_rate	10
host_example_config	10
infect_example_config	11
infect_fun	11
life_cycle	12
new_transition_function	12
ogden2005	13
ogden_feed_fun	14
parameters	15
predictors	15
predictor_spec	16
print.predictor_spec	17
print.transition	17
run	18
snow_cover_fun	18
temp_example_config	19
transition	19
winter_tick	20

Index	22
--------------	-----------

annual_growth_rate	<i>Calculate annual growth rate</i>
--------------------	-------------------------------------

Description

Calculate annual growth rate

Usage

```
annual_growth_rate(out)
```

Arguments

out	Model output data frame
-----	-------------------------

Value

Numeric vector of length one representing the annual factor by which the total tick population changes. To use this function, it is best to run the model for at least three years.

Examples

```
## Not run:
out <- run(ogden2005)
annual_growth_rate(out)

## End(Not run)
```

config	<i>Create a config object</i>
--------	-------------------------------

Description

Make a config object from the input parameters, and ensure that the inputs meet the requirements for the model. The returned object is a complete description of a model run scenario.

Usage

```
config(
  cycle,
  initial_population,
  preds = NULL,
  steps = 365L,
  max_duration = 365L,
  verbose = TRUE
)
```

Arguments

cycle	A tick's life_cycle test
initial_population	Named numeric vector indicating starting population for each life stage. Life stages not specified are assumed to be 0.
preds	Optional input predictors data
steps	Numeric vector of length one indicating the duration to run the model over in days.
max_duration	Numeric vector of length one. Determines the maximum number of days that a duration-based transition can last, after which ticks are removed from the model/die. Default of 365 is likely sensible for most cases.
verbose	Boolean; whether to warn about coercion to inputs

Value

A config object

Examples

```
# We build a simple example config
my_config <- config(
  cycle = life_cycle(
    transition("a", "b", function() 0.1, "probability"),
    transition("b", "a", function() 10, "probability")
  ),
  initial_population = c(a = 1)
)

# If we make a change to an existing `config`, it is a good idea to check
# whether it is still valid by calling `config()` on it again. For example,
# here we set the initial_population of a life stage that is not included in
# the life cycle.
my_config$initial_population <- c(a = 1, c = 1)

## Not run:
# Now, we re-run the validations, which will throw an error
do.call(config, my_config)

## End(Not run)
```

config_ex_1

Simple model configuration example

Description

This model configuration uses only non-delay transitions, and no transitions depend on predictors (e.g. weather or host community). Parameter values are selected so that the population is stable over time.

Usage

```
config_ex_1
```

Format

A [config](#)

`config_ex_2`*Simple model configuration example using delays*

Description

This model configuration uses delay transitions for all transitions except the adult to eggs transition. As in `config_ex_1`, no transitions depend on predictors, and the population is stable over time.

Usage`config_ex_2`**Format**

A [config](#)

`constant_fun`*Constant function*

Description

Constant function

Usage`constant_fun(a)`**Arguments**

`a` Parameter a in parameters table.

Value

Numeric vector of length 1 equal to input parameter a

Examples`constant_fun(1)`

density_fun *Density dependent mortality*

Description

Density dependent mortality

Usage

```
density_fun(x, y, a, b, c, pref)
```

Arguments

x	Predictor 1 in transitions table. Numeric vector indicating host density for each of the host species. Length should be equal to the number of host species.
y	Predictor 2 in transitions table. Number of feeding ticks in life stages specified by predictor 2.
a	Parameter a in parameters table.
b	Parameter b in parameters table.
c	Parameter c in parameters table.
pref	Parameters named pref in parameters table. Numeric vector of length equal to the number of host species. Values are the preference for ticks in a given transition for each host species.

Value

Numeric vector of length 1, indicating mortality rate

Examples

```
density_fun(c(10, 20), 100, .1, .3, .2, c(.5, .8))
```

expo_fun *Exponential function*

Description

Exponential function

Usage

```
expo_fun(x, a, b)
```

Arguments

x	Predictor 1 in transitions table.
a	Parameter a in parameters table.
b	Parameter b in parameters table.

Value

Numeric vector of length 1

Examples

```
expo_fun(.5, .1, .3)
```

feed_fun	<i>Probability of actively questing and then finding a host</i>
----------	---

Description

Probability of actively questing and then finding a host

Usage

```
feed_fun(x, y, a, pref, q, tmin, tmax)
```

Arguments

x	Predictor 1 in transitions table. Numeric vector indicating host density for each of the host species. Length should be equal to the number of host species.
y	Predictor 2 in transitions table. Numeric vector of length 1 indicating temperature.
a	Parameter a in parameters table.
pref	Parameters named pref in parameters table. Numeric vector of length equal to the number of host species. Values are the preference for ticks in a given transition for each host species.
q	Parameter q in parameters table. Used in Briere function.
tmin	Parameter tmin in parameters table. Indicates minimum temperature at which ticks actively quest.
tmax	Parameter tmax in parameters table. Indicates maximum temperature at which ticks actively quest.

Details

Product of binomial and Briere functions (prob of finding a host) * (prob of active questing)

Value

Numeric vector of length 1

Examples

```
feed_fun(10, 30, .001, .1, .5, 20, 40)
```

find_n_feed

Probability of finding a host and successfully feeding on it

Description

Probability of finding a host and successfully feeding on it

Usage

```
find_n_feed(x, a, pref, feed_success)
```

Arguments

x	Predictor 1 in transitions table. Numeric vector indicating host density for each of the host species. Length should be equal to the number of host species.
a	Parameter a is the probability that a tick finds any one host.
pref	Parameters named pref in parameters table. Numeric vector of length equal to the number of host species. Values are the preference for ticks in a given transition for each host species.
feed_success	Parameters named feed_success in parameters table. Numeric vector of length equal to the number of host species. Values are the feeding success rate for ticks in a given transition while feeding on each host species.

Value

Numeric vector of length 1 indicating probability that ticks find any host and then successfully feed on that host.

Examples

```
find_n_feed(10, .1, 1, .5)
find_n_feed(runif(2) * 10, .1, runif(2), runif(2))
```

format.predictor_spec *Format a predictor_spec*

Description

Format a predictor_spec

Usage

```
## S3 method for class 'predictor_spec'  
format(x, ...)
```

Arguments

x	a predictor_spec
...	not used

Value

string representation of input

get_pred_from_table *Get a predictor from input data*

Description

Get a predictor from input data

Usage

```
get_pred_from_table(time, pred, table)
```

Arguments

time	Numeric vector of days to get data. Ignored if input is constant over time (as indicated by NA value in 'j_day' column)
pred	string specifying the name of the predictor, e.g. "host_den"
table	input predictors table

Value

a numeric vector of predictor values

growth_rate *Calculate multiplicative growth rate of population*

Description

Calculate multiplicative growth rate of population

Usage

```
growth_rate(out)
```

Arguments

out Model output data frame

Value

Numeric vector of length one representing daily growth rate.

Examples

```
out <- run(config_ex_1)
growth_rate(out)
```

host_example_config *Configuration for showing how we can modify host community data*

Description

Configuration for showing how we can modify host community data

Usage

```
host_example_config
```

Format

A [config](#)

infect_example_config *Configuration for showing infection dynamics*

Description

Configuration for showing infection dynamics

Usage

infect_example_config

Format

A [config](#)

infect_fun	<i>Probability that a feeding tick becomes engorged infected or uninfected</i>
------------	--

Description

Probability that a feeding tick becomes engorged infected or uninfected

Usage

infect_fun(x, from_infected, to_infected, host_rc, pref)

Arguments

x	Predictor 1 in transitions table. Numeric vector indicating host density for each of the host species. Length should be equal to the number of host species.
from_infected	Parameter from_infected in parameters table. Value should be 1 if transition is from an infected tick stage, 0 otherwise.
to_infected	Parameter to_infected in parameters table. Value should be 1 if transition is to an infected tick stage, 0 otherwise.
host_rc	Parameters named host_rc in parameters table. Numeric vector of length equal to the number of host species. Values are the host reservoir competence for each host species.
pref	Parameters named pref in parameters table. Numeric vector of length equal to the number of host species. Values are the preference for ticks in a given transition for each host species.

Details

Since density dependent mortality is subtracted later, in this function we assume that all feeding ticks feed successfully and become engorged.

Value

Numeric vector of length 1

Examples

```
infect_fun(10, 0, 0, .3, 1)
infect_fun(10, 0, 1, .3, 1)
infect_fun(10, 1, 1, .3, 1)
```

life_cycle	<i>Create a life_cycle from a collection of transitions</i>
------------	---

Description

Create a life_cycle from a collection of transitions

Usage

```
life_cycle(...)
```

Arguments

... A set of [transitions](#)

Value

a life_cycle

new_transition_function	<i>Constructor for transition functions</i>
-------------------------	---

Description

Constructor for transition functions

Usage

```
new_transition_function(fun)
```

Arguments

fun A function
 Transition functions must return a numeric vector. See [constant_fun](#), [expo_fun](#) and [infect_fun](#) for examples for how to write custom functions.

ogden2005	<i>Configuration for Ixodes scapularis population dynamics model from Ogden et al. 2005</i>
-----------	---

Description

This model configuration recreates the *Ixodes scapularis* (blacklegged tick) population dynamics model from Ogden et al. 2005. This is a relatively complete model of tick population dynamics, including the effects of both temperature and the host community on tick life-stage transitions. We include this configuration to show that our package can be used to recreate existing models.

Usage

```
ogden2005
```

Format

A [config](#)

Details

In this config, the population starts with 10000 questing adults. The predictor data includes average temperature for each day, and density of hosts over the model run. Here the host community is stable with 20 deer and 200 mice.

See Also

Ogden et al. (2005) [doi:10.1016/j.ijpara.2004.12.013](https://doi.org/10.1016/j.ijpara.2004.12.013)

Examples

```
data(ogden2005)
## Not run:
output <- run(ogden2005)
graph_population_each_group(output)

## End(Not run)
```

ogden_feed_fun *Probability of actively questing times constant host finding probability*

Description

Probability of actively questing times constant host finding probability

Usage

```
ogden_feed_fun(x, a, q, tmin, tmax)
```

Arguments

x	Predictor 1 in transitions table. Numeric vector of length 1 indicating temperature.
a	Parameter a in parameters table.
q	Parameter q in parameters table. Used in Briere function.
tmin	Parameter tmin in parameters table. Indicates minimum temperature at which ticks actively quest.
tmax	Parameter tmax in parameters table. Indicates maximum temperature at which ticks actively quest.

Details

(const prob of finding a host) * (prob of active questing)

Value

Numeric vector of length 1

See Also

Based on Ogden et al. (2005) [doi:10.1016/j.ijpara.2004.12.013](https://doi.org/10.1016/j.ijpara.2004.12.013)

Examples

```
ogden_feed_fun(30, .03, .01, 10, 35)
```

parameters	<i>Create a set of parameters</i>
------------	-----------------------------------

Description

Create a set of parameters

Usage

```
parameters(...)
```

Arguments

... A set of named numeric vectors, each corresponding to a parameter. If a parameter is of length > 1, each element must be named.

Value

a parameters object

Examples

```
# create a set of scalar parameters
parameters(a = 1, b = 2)

# parameters of length > 1 may be useful for host-related parameters that
# differ between host species, for example tick feeding success
parameters(a = 1, feeding_success = c(deer = 0.49, squirrel = 0.17))
```

predictors	<i>Create a table of predictors</i>
------------	-------------------------------------

Description

A data frame of input data to be used in as predictor values in transition functions.

Usage

```
predictors(df, verbose = FALSE)
```

Arguments

df	input data frame, with columns: pred String specifying the name of the predictor, e.g. "temp" or "host_den" pred_subcategory This column allows specifying predictors for which there are multiple values for a given j_day. Predictor values are sorted by this column in the config set up. This ensures that when accessing a predictor with multiple values for the same j_day, we get a vector of predictor values ordered by this column. A typical use for this column is to specify the host density of each host species. j_day Integer specifying the Julian day, or NA for predictors with constant value over time value Numeric value of predictor
verbose	Boolean; whether to warn about reordering the df

Value

a predictors object

predictor_spec	<i>Specify how a single predictor should be used</i>
----------------	--

Description

Specify how a single predictor should be used

Usage

```
predictor_spec(pred, first_day_only = TRUE)
```

Arguments

pred	String indicating where to get predictor data. Can be one of: <ul style="list-style-type: none"> • A string in the "pred" column in the predictors table. In this case, the predictor value passed to the containing transition's fun is the corresponding value of that predictor in the table. • A string that matches at least one life stage name via regex. In this case, the value passed to the containing transition's fun is the sum of the population sizes of all matched life stages.
first_day_only	Boolean indicating whether to repeat the predictor data value from the first day of a transition when evaluating it (TRUE case), or to use the range of predictor data over the duration of a transition (FALSE case). FALSE is only valid for transitions with "duration" as the transition_type, because "probability" type transitions only last one day. A value of FALSE also requires the name parameter to be a value in the predictors table "pred" column, not a tick life stage.

Value

a predictor_spec list-based object

`print.predictor_spec` *Print a predictor_spec*

Description

Print a predictor_spec

Usage

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

Arguments

x	a predictor_spec
...	not used

`print.transition` *Print a transition*

Description

Print a transition

Usage

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

Arguments

x	A transition
...	not used

run	<i>Run the model</i>
-----	----------------------

Description

Run the model

Usage

```
run(cfg, progress = TRUE)
```

Arguments

cfg	An IxPopDyMod::config object
progress	Boolean indicating whether to log progress every 100 steps

Value

Data frame of population of ticks of each life stage each day

Examples

```
run(config_ex_1)
```

snow_cover_fun	<i>Mortality as a function of whether there is a snow on the ground</i>
----------------	---

Description

Mortality as a function of whether there is a snow on the ground

Usage

```
snow_cover_fun(x, no_snow_mort, snow_mort)
```

Arguments

x	amount of snow on ground
no_snow_mort	mortality with no snow on the ground
snow_mort	mortality with snow on the ground

temp_example_config	<i>Configuration for showing how we can modify climate data</i>
---------------------	---

Description

Configuration for showing how we can modify climate data

Usage

```
temp_example_config
```

Format

A [config](#)

transition	<i>Create a transition object</i>
------------	-----------------------------------

Description

A transition object represents a single transition between two tick life stages, or the mortality rate from a life stage.

Usage

```
transition(
  from,
  to,
  fun,
  transition_type,
  mortality_type = NULL,
  predictors = NULL,
  parameters = list()
)
```

Arguments

from	The name of the life stage a tick is transitioning from.
to	The name of the life stage a tick is transitioning to, or NULL if the transition is representing mortality.
fun	The transition function to evaluate. The inputs of the function are predictors and parameters. The output is the daily probability of completing the transition, for "probability" transitions, or the daily rate the transition takes place, for "duration" transitions.

transition_type	One of: "probability": the evaluated transition is interpreted as the daily fraction of ticks that complete the transition. Ticks remain in the original life stage if they do not complete a transition or undergo mortality. "duration": the transition is complete on the first day that the cumulative sum of the evaluated transition is greater than or equal to 1. No ticks remain in the original life stage at the end of a transition - they either complete the transition or die.
mortality_type	One of: NULL: the default, indicating that the transition is not mortality. "per_day": indicates that the evaluated transition is the fraction of ticks that dies each day. "throughout_transition": only valid for "duration" type transitions, where it indicates that the evaluated transition is the fraction of ticks that die throughout the entire transition.
predictors	Optional, a named list of predictor_spec objects that specify how any predictor data should be used in evaluating fun. The names are matched with the formal args to fun to determine which input in fun each predictor will be passed to.
parameters	Optional, a parameters object, or a named list of numeric vectors.

Value

a transition object

winter_tick

Configuration for winter tick population dynamics model

Description

This is a model configuration based on a literature search on the factors affect the winter tick life cycle. Many of the transitions and parameters in this configuration are drawn from Drew and Samuel (1986). We include this configuration to show that our package is flexible for modeling multiple tick species with different life histories.

Usage

winter_tick

Format

An object of class config of length 5.

See Also

Drew and Samuel (1986) [doi:10.1139/z86105](https://doi.org/10.1139/z86105)
 Drew and Samuel (1985) [doi:10.7589/0090355821.3.274](https://doi.org/10.7589/0090355821.3.274)
 Addison and McLaughlin (1988) [doi:10.2307/3282188](https://doi.org/10.2307/3282188)
 Ogden et al. (2005) [doi:10.1016/j.ijpara.2004.12.013](https://doi.org/10.1016/j.ijpara.2004.12.013)

Examples

```
data(winter_tick)
## Not run:
output <- run(winter_tick)
graph_population_each_group(winter_tick)

## End(Not run)
```

Index

* datasets

- config_ex_1, 4
- config_ex_2, 5
- host_example_config, 10
- infect_example_config, 11
- ogden2005, 13
- temp_example_config, 19
- winter_tick, 20

annual_growth_rate, 2

config, 3, 4, 5, 10, 11, 13, 19

config_ex_1, 4

config_ex_2, 5

constant_fun, 5, 12

density_fun, 6

expo_fun, 6, 12

feed_fun, 7

find_n_feed, 8

format.predictor_spec, 9

get_pred_from_table, 9

growth_rate, 10

host_example_config, 10

infect_example_config, 11

infect_fun, 11, 12

life_cycle, 3, 12

new_transition_function, 12

ogden2005, 13

ogden_feed_fun, 14

parameters, 15, 20

predictor_spec, 16, 20

predictors, 3, 15, 16

print.predictor_spec, 17

print.transition, 17

run, 18

snow_cover_fun, 18

temp_example_config, 19

transition, 12, 16, 19

winter_tick, 20