Package ‘albopictus’

October 12, 2022

Type Package

Title Age-Structured Population Dynamics Model

Version 0.5

Date 2018-11-21

Author Kamil Erguler [aut, cre]

Maintainer Kamil Erguler <`k.erguler@cyi.ac.cy`>

Description Implements discrete time deterministic and stochastic age-structured population dynamics models described in Erguler and others (2016) <doi:10.1371/journal.pone.0149282> and Erguler and others (2017) <doi:10.1371/journal.pone.0174293>.

License GPL (>= 3)

URL `https://github.com/kerguler/albopictusR`

LazyLoad yes

Imports methods

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-11-29 08:20:03 UTC

R topics documented:

`albopictus-package` *Age-Structured Population Dynamics Model*

Description

Implements the "spop" class for age-structured population dynamics modelling. For more information, see class description.

Author(s)

Kamil Erguler, Associate Research Scientist, EEWRC, The Cyprus Institute <`k.erguler@cyi.ac.cy`>

References

- Kamil Erguler, Stephanie E Smith-Unna, Joanna Waldock, Yiannis Proestos, George K Christophides, Jos Lelieveld, Paul E Parham. Large-scale modelling of the environmentally-driven population dynamics of temperate Aedes albopictus (Skuse). PLOS ONE, 2016
- Kamil Erguler, Nastassya L Chandra, Yiannis Proestos, Jos Lelieveld, George K Christophides, Paul E Parham. A large-scale stochastic spatiotemporal model for Aedes albopictus-borne chikungunya epidemiology. PLOS ONE, 2017

See Also

Useful links:

- `https://github.com/kerguler/albopictusR`

`add<-` *Add batch*

Description

Introduce a batch of individuals with a given age

Usage

```
add(x) <- value

## S4 replacement method for signature 'spop,data.frame'
add(x) <- value
```

Arguments

`x`	spop class instant
`value`	`data.frame` with `age`, `devcycle`, `development`, and `number` fields

`dead` *Read dead*

Description

Read the number of dead individuals after each iteration

Usage

```
dead(x)

## S4 method for signature 'spop'
dead(x)
```

Arguments

`x`	spop class instant

`developed` *Read developed*

Description

Read the number of individuals designated to complete their development

Usage

```
developed(x)

## S4 method for signature 'spop'
developed(x)
```

Arguments

`x`	spop class instant

`devtable` *Read devtable*

Description

Read the number, age, and development cycles of individuals completing their development after each iteration

Usage

```
devtable(x)

## S4 method for signature 'spop'
devtable(x)
```

Arguments

`x`	spop class instant

`gamma_dist_prob` *Gamma probability of death or development*

Description

Gamma-distributed probability of death or development happening in the next iteration

Usage

```
gamma_dist_prob(xr, mn, std)
```

Arguments

`xr`	age of individuals
`mn`	mean age of death or development
`std`	standard deviation of the age of death or development

`iterate<-` *Iterate population*

Description

Iterate the population for one day

Usage

```
iterate(x) <- value

## S4 replacement method for signature 'spop,data.frame'
iterate(x) <- value
```

Arguments

`x`	spop class instant
`value`	`data.frame` with the following fields - `death`: age-independent daily probability of death - `death_mean` and `death_sd`: age-dependent daily probability of death (`death_sd=0` indicates fixed life time (defined by `death_mean`)) - `dev`: age-independent daily probability of development - `dev_mean` and `dev_sd`: age-dependent daily probability of development (`dev_sd=0` indicates fixed development time (defined by `dev_mean`))

`nbinom_dist_prob` *Negative binomial probability of death or development*

Description

Negative binomial-distributed probability of death or development happening in the next iteration

Usage

```
nbinom_dist_prob(xr, mn, std)
```

Arguments

`xr`	age of individuals
`mn`	mean age of death or development
`std`	standard deviation of the age of death or development

`perturb<-` *Iterate population without incrementing age or development*

Description

Iterate the population for one day keeping age and development fixed

Usage

```
perturb(x) <- value

## S4 replacement method for signature 'spop,data.frame'
perturb(x) <- value
```

Arguments

`x`	spop class instant
`value`	`data.frame` with the following fields • `death`: age-independent daily probability of death • `death_mean` and `death_sd`: age-dependent daily probability of death (`death_sd=0` indicates fixed life time (defined by `death_mean`)) • `dev`: age-independent daily probability of development • `dev_mean` and `dev_sd`: age-dependent daily probability of development (`dev_sd=0` indicates fixed development time (defined by `dev_mean`))

`size` *Read size*

Description

Read the total number of individuals

Usage

```
size(x)

## S4 method for signature 'spop'
size(x)
```

Arguments

`x`	spop class instant

---

spop *An S4 class to represent an age-structured population*

Description

- `spop` implements the deterministic and stochastic age-structured population dynamics models described in Erguler et al. 2016 and 2017
- `add` introduces a batch of individuals with a given age, completed development cycles, and degree of development (default: 0)
- `iterate` iterates the population for one day and calculates (overwrites) the number of dead individuals and the number of individuals designated to complete their development
- `devtable` reads the number, age, and development cycle of individuals designated to complete their development
- `developed` reads the total number of individuals designated to complete their development
- `dead` reads the number of dead individuals after each iteration
- `size` reads the total number of individuals

Details

This is an R implementation of the age-structured population dynamics models described in Erguler et al. 2016 and 2017. The `spop` class records the number and age of individuals and implements two processes to exit from the population: development and death. The two processes act upon the population sequentially; survival is imposed prior to development. If the population survives for one day, then, it is allowed to grow and complete its development. Survival and development are defined either with an age-independent daily probability, or an age-dependent gamma- or negative binomial-distributed probability.

- `stochastic`: a logical value indicating a deterministic or a stochastic population dynamics
- `prob`: a character string indicating the basis of age-dependent survival or development (gamma: gamma-distributed, nbinom: negative binomial-distributed)

Examples

```
# Generate a population with stochastic dynamics
s <- spop(stochastic=TRUE)
# Add 1000 20-day-old individuals
add(s) <- data.frame(number=1000,age=20)

# Iterate one day without death and assume development in 20 (+-5) days (gamma-distributed)
iterate(s) <- data.frame(dev_mean=20,dev_sd=5,death=0)
print(developed(s))

# Iterate another day assuming no development but age-dependent survival
# Let each individual survive for 20 days (+-5) (gamma-distributed)
iterate(s) <- data.frame(death_mean=20,death_sd=5,dev=0)
print(dead(s))
```

```
# Note that the previous values of developed and dead will be overwritten by this command

# Generate a deterministic population and observe the difference
s <- spop(stochastic=FALSE)
add(s) <- data.frame(number=1000,age=20)

iterate(s) <- data.frame(dev_mean=20,dev_sd=5,death=0)
print(developed(s))

iterate(s) <- data.frame(death_mean=20,death_sd=5,dev=0)
print(dead(s))
```

Index