Package ‘HostSwitch’

October 12, 2022

Type Package

Title Simulate the Extent of Host Switching by Consumers

Version 0.1.2

Description Using a simulation-based approach, the 'HostSwitch' package provides functions to investigate host switches by consumers.
The individual-based model is based on the concept of ecological fitting.
The mockup model is published by Araujo et al. 2015. Understanding Host-Switching by Ecological Fitting (<doi:10.1371/journal.pone.0139225>).
The package provides an R-friendly and modified version of this model which can be applied to different consumer-resource scenarios.

License GPL (>= 3)

URL `https://github.com/berndpanassiti/HostSwitch`

BugReports `https://github.com/berndpanassiti/HostSwitch/issues`

Depends R (>= 3.5.0),

Imports shiny, ggplot2, stats, utils, checkmate (>= 2.1.0), methods, purrr, tippy

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

VignetteBuilder knitr

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.2.0

Config/testthat/edition 3

NeedsCompilation no

Author Valeria Trivellone [aut] (<`https://orcid.org/0000-0003-1415-4097`>),
Sabrina B.L. Araujo [aut] (<`https://orcid.org/0000-0002-8759-8310`>),
Bernd Panassiti [cre, aut] (<`https://orcid.org/0000-0002-5899-4584`>)

Maintainer Bernd Panassiti <`bernd.panassiti@gmail.com`>

Repository CRAN

Date/Publication 2022-08-09 17:30:02 UTC

R topics documented:

`parli` *Consumer life cycle parameters to simulate host switches*

Description

Data derived from real world experiments (see sources). Data includes life cycle parameters for a wildlife ecology, agricultural and biomedical arena to simulate host switches.
Semantics of parli = **par**ameter of **li**fe cycle

Usage

```
data(parli)
```

Format

An object of class "list". The list contains three matrices: $Cephaloleia, $Cacopsylla, and $SarsMers.

1. *parli$Cephaloleia* is a data frame with 11 rows (parameters) and 8 columns (variables).

   The parameters are classified in three groups:

   - Arbitrary varying: mig (probability of migration) and jump_back (to the original Resource).
   - Arbitrary fixed: K (carrying capacity), N-generations, seed, n_sim, pRes_min, pRes_max, sd (standard deviation of mutation), and sigma (standard deviation of survival).
   - From real data: b (net reproduction rate) (Garcia-Robledo and Horvitz, 2011).

   The column name acronyms are assigned after:

   - two *Cephaloleia* species: **Cb** for *C. belti* and **Cp** for *C. placida*.
   - two levels of variation for mig: **mL** for low and **mH** for high migration.
   - two levels for jump_back: **jY** for jumping back and **jN** for no jumping back individuals.

2. *parli$Cacopsylla* is a data frame with 11 rows and 8 columns. The parameters are classified in three groups:

   - Arbitrary varying: sd and jump_back.
   - Arbitrary fixed: K, N-generations, seed, n_sim, pRes_min, pRes_max
   - From real data: b, sigma and mig (Malagnini et al. 2010, 2011; Mayer et al. 2013).

   The column name acronyms are assigned after:

  - two *Cacopsylla melanoneura* populations: **CmA** for *C. melanoneura* adapted to Apple and **CmH** for *C. melanoneura* adapted to hawthorn.
  - two levels of variation for sd: **sdL** for low and **sdH** for high standard deviation of mutation.
  - two levels for jump_back: **jY** for jumping back and **jN** for no jumping back individuals.

3. *parli$SarsMers*is a data frame with 11 rows and 8 columns. The parameters are classified in three groups:

  - Arbitrary varying: mig
  - Arbitrary fixed: K, N-generations, seed, n_sim, pRes_min, pRes_max
  - From real data: b, sigma, and sd (Kim et al. 2021; van Dorp et al. 2020)

   The column name acronyms are assigned after:

  - two coronaviruses: **Sars** for *Sarbecovirus* sp. and **Mers** for *Merbacovirus* sp.
  - three levels of variation for mig: **migL** for low, **migM** for medium and **migH** for high probability of migration.

Source

Garcia-Robledo C and Horvitz CC (2011) Experimental demography and the vital rates of generalist and specialist insect herbivores on native and novel host plants. Journal of Animal Ecology, 80(5):976-989.

Kim KS, Ejima K, Iwanami S, Fujita Y, Ohashi H, Koizumi Y, Asai Y, Nakaoka S, Watashi K, Aihara K, et al. (2021) A quantitative model used to compare within-host sars-cov-2, mers-cov, and sars-cov dynamics provides insights into the pathogenesis and treatment of sars-cov-2. PLoS biology, 19(3):e3001128.

Malagnini V, Pedrazzoli F, Gualandri V, Forno F, Zasso R, Pozzebon A, and Ioriatti C. (2010) A study of the effects of 'candidatus phytoplasma mali' on the psyllid Cacopsylla melanoneura (Hemiptera: Psyllidae). Journal of invertebrate pathology, 103(1):65-67.

Malagnini V, Pedrazzoli F, Papetti C, Cainelli C, Zasso R, Gualandri V, Pozzebon A, and Ioriatti C. (2013) Ecological and genetic differences between Cacopsylla melanoneura (Hemiptera, Psyllidae) populations reveal species host plant preference. PloS one, 8(7):e69663.

Mayer CJ, Vilcinskas A, and Gross J. (2011) Chemically mediated multitrophic interactions in a plant-insect vector-phytoplasma system compared with a partially nonvector species. Agricultural and Forest Entomology, 13(1):25-35.

van Dorp L, Acman M, Richard D, Shaw LP, Ford CE, Ormond L, Owen CJ, Pang J, Tan CC, Boshier FA, et al (2020) Emergence of genomic diversity and recurrent mutations in sars-cov-2. Infection, Genetics and Evolution, 83:104351.

References

Trivellone V, Araujo SBL and Panassiti B (2021) HostSwitch: An R Package to Simulate the Extent of Host-Switching by a Consumer (submitted).

Examples

```
data(parli)
knitr::kable(parli$Cephaloleia) # tibble::as.tibble(parli$Cephaloleia)
```

plotHostSwitch *HostSwitch standard plot*

Description

This function creates a standard plot for the simulated host switches

Usage

```
plotHostSwitch(HostSwitch_simulated_quantities, sim_n = 1)
```

Arguments

`HostSwitch_simulated_quantities`
The object HostSwitch created by `simHostSwitch()`

`sim_n` for HostSwitch object including more than 1 simulation saved, "sim_n" specifies which simulation have to be plotted. The plot from the first simulation is shown as default (sim_n = 1). If the called simulation number is not present in the HostSwitch object, an error message will be returned.

Details

The function plots dispersal and colonization (host-switching events) of Consumers on a novel host offered at each generation given the values of parameters related to carrying capacity, fitness space, migration, reproduction, selection, and biological model. The X-axis shows the total number of possible generations defined in the object HostSwitch, the Y-axis the phenotype values of the Consumer.

Black dots are the phenotype values of the Consumer after each event of reproduction. The green squares represent the value of phenotype favored by the novel Resource offered at each generation. The red squares are the phenotype values of the Consumer favored by the current Resource.The blue dots represents the phenotypes of dispersing Consumers, and the yellow dots the successfully colonizing Consumers.

Value

An S3 object with class gg/ggplot

Examples

```
m1 = simHostSwitch(n_sim=100) # create an HostSwitch object with 100 simulations.

## The results of the first simulation (sim_n=1) are plotted as default
plotHostSwitch(m1)

## The 50th simulation of the model is plotted
plotHostSwitch(m1,sim_n=50)
```

`shinyHostSwitch` *Shiny-plot of consumer host-switching*

Description

Shiny-plot of consumer host-switching

Usage

```
shinyHostSwitch()
```

Details

The function creates an interactive web-based front-end using Shiny App. The Shiny-Plot simulates dispersal and colonization events of a consumer on novel hosts.

The parameter values included in the function `simHostSwitch()` can be modified by the slider bars. The refresh button on the top left (Refresh simulation) plots the new simulation.

Black dots are the phenotype values of the Consumer after each event of reproduction. The green squares represent the value of phenotype values of the Consumer favored by the novel host proposed at each generation. The red squares are the phenotype values favored by the current host.

The total number of dispersal and successful host switch events (or individuals) by Consumers for each are reported below the plot.

Value

a dynamic interface plotting the dispersion events and host switch simulations and reacting to user input.

Examples

```
## Only run examples in interactive R sessions
if (interactive()) {
 shinyHostSwitch()
}
```

`simHostSwitch` *Simulates the number of dispersion and successful host switch events by individuals of the Consumer until all individuals die.*

Description

Simulates the number of dispersion and successful host switch events by individuals of the Consumer until all individuals die.

Usage

```
simHostSwitch(
  data = NULL,
  column = NULL,
  K = 100,
  b = 10,
  mig = 0.01,
  sd = 0.2,
  sigma = 1,
  pRes_min = 1,
  pRes_max = 10,
  n_generations = 200,
  jump_back = "no",
  seed = NULL,
  n_sim = 1,
  nInitConsumer = 20
)
```

Arguments

data
A matrix or dataset, the columns may indicate different types of Consumes characterized by a specific set of parameters (rows), see details. Default value: NULL.

column
Used together with data argument; indicate the column name, string. Default value: NULL.

K
Carrying capacity, positive integer (min=1, max=1000), default value: 100.

b
net reproduction rate; average number of offspring that a population of the Consumer produces at each generation, numeric value (min=0), default value: 10.

mig
define the proportion of successful migrating individuals at each generation, numeric value (min=0, max=1), default value: 0.01.

sd
Standard deviation for mutation, numeric value (min=0, max=10), default value: 0.2.

sigma
Standard deviation of the survival function, numeric value (min=0, max=10), default value: 1.

pRes_min
smallest optimum phenotype value imposed by the Resource, numeric value (min=1, max=pRes_max), default value: 1.

pRes_max
highest optimum phenotype value imposed by the Resource, numeric value (min=pRes_min, max=100), default value: 10.

n_generations
Number of generations, positive integer (min=1, max=50000), default value: 200.

jump_back
Option for consumers that do not survive on the novel resource. If "yes" the consumer(s) jump back to the current resource and will be considered in the selective pressure and reproduction stage for the n+1 generation, if "no" (default) it dies on the new host.

`seed` a single value useful for creating simulations or random objects that can be reproduced, positive integer (>0), default value: NULL.

`n_sim` Number of simulations, positive integer (min=1, max = 50000), default value: 1.

`nInitConsumer` propagule size (or number of initial individuals) at the generation n = 0, default value: 20.

Details

This function simulates the number of host switches by the population of a consumer. There are 2 ways to provide parameters to the `simHostSwitch()` function:

data **"data","column"**: Provide names of matrix/dataframe and column, e.g. data= "parli$Cephaloleia", column = "Cb.mLxjN"

parameter **individual parameter**: e.g. b=5, n_generations=500, etc...

If no data/column or individual parameters are provided, default parameter values are used. The rownames of the data must match the parameter argument names. You may use one of the `parli()` datasets as a template.

Results are stored to an object of class ‘HostSwitch’. to make use of summary and plotting functions in the **HostSwitch** package. Please note that when arguments "data" and "column" are provided, the results are stored to the global environment using the colname provided to the argument "column" (in our example above Cb.mLxjN).

The object of class ‘’HostSwitch’ includes the following simulated quantities:

pRes_sim **$pRes_sim**: a vector of the optimum phenotypes (one for each generation) that Consumers should have to be favored by the current Resource.

pRes_new_sim **$pRes_new_sim**: a vector of the optimum phenotypes (one for each generation) that Consumers should have to be favored by the novel Resource.

pInd_sim **$pInd_sim**: list of vectors that includes the individual phenotype values of the Consumers in the population of each generation.

pInd_jump_sim **$pInd_jump_sim**: vector of number of migrating individuals at each generation. The vector length is always equal to the ’n_generation’ parameter, if the simulation ends before the ’n_generation’ value then the vector will include a ’NA’ by default.

pInd_whichjump_sim **$pInd_whichjump_sim**: list of vectors that extracts the individual phenotype values of the Consumers who disperse in a novel Resource in each population and generation.

pInd_whichsurv_sim **$pInd_whichsurv_sim**: list of vectors that extracts the individual phenotype values of the Consumers who successful colonize a novel Resource in each population and generation.

These simulated quantities of interest are available for each generation step and can be used for summary statistics and plots using functions `summaryHostSwitch()` and `plotHostSwitch()`, respectively.

Note: One important aspect of *simHostswitch* is that it is based on the `survivalProbability()` function.

Value

An object of class ‘HostSwitch’.

See Also

survivalProbability(), summaryHostSwitch(), plotHostSwitch()

Examples

```
m1 = simHostSwitch() # using default values for arguments

data(parli)
Cephaloleia=parli$Cephaloleia
m2 = simHostSwitch(data=Cephaloleia, column="Cb.mLxjN")

## Not run:
simHostSwitch(sigma=100)
## End(Not run)
```

summaryHostSwitch *Summary statistics of HostSwitch simulation*

Description

Summary statistics of HostSwitch simulation

Usage

```
summaryHostSwitch(HostSwitch_simulated_quantities, warmup = 1)
```

Arguments

`HostSwitch_simulated_quantities`	An object created by simHostSwitch()
`warmup`	warmup is the number of initial generations to be excluded from summary statistics, see details. Possible value are NULL or positive integer (min=1, max=50), default value = 1

Details

This function generates summary statistics for HostSwitch simulations. Quantities of interest for each simulation are averaged. If *n_sim = 1*, these averages for this single simulation are shown. If *n_sim > 1*, summary statistics are applied on the simulation averages.

Warmup represents the initial condition for the simulation, the users may defined it as an adaptation stage of the simulation model. If warmup = 1 the generation at time 0 is excluded from summary, if warmup = 2 the generations at times 0 and 1 are excluded and so on. If warmup = NULL all generations are considered for summary statistics.

Value

Summary of HostSwitch simulations

Examples

```
## Create an object HostSwitch with 100 simulations and default values for all the other parameters
m1 = simHostSwitch(n_sim=100)

summaryHostSwitch(m1)
```

survivalProbability *Survival probability of the Consumer in a novel Resource*

Description

Survival probability of the Consumer in a novel Resource

Usage

```
survivalProbability(pInd, pOpt, sigma)
```

Arguments

pInd	Phenotype of ith Consumer attempting to disperse on the novel Resource
pOpt	The optimum phenotype the Consumer should have to maximize the colonization success
sigma	Standard deviation of the survival function (see Details for more explanations)

Details

This function calculates the survival probability of individual consumers that attempt dispersal to a new host. It is the core function of `simHostSwitch()`. The probability of survival of each individual of the consumer to a novel Resource follows a normal distribution. The formula is formalized as follows

$$P(pInd, pOpt) = e^{-\frac{(pInd-pOpt)^2}{2\sigma^2}}$$

The normalizing constant

$$NC = \frac{1}{\sigma(\sqrt(2\pi))}$$

is ignored here.\ "Sigma" the higher the sigma, the lower the selection and the higher the probability of surviving. Ecologically this value may be related to the niche breadth for the Consumer (species).

Value

The survival probability of the consumer

Examples

```
## Example 1a - The ith consumer has the phenotype that maximize its
## colonization success on the new host, then pInd is equal to pOpt
## (pInd = pOpt), and the survival probability is 1.
survivalProbability(pInd=5,pOpt=5,sigma=1)

## Example 1b - Increasing |pInd-pOpt| the survival probability decreases
survivalProbability(pInd=5,pOpt=30,sigma=1)

## Example 1c - Given a |pInd-pOpt|> 1, increasing sigma results in
## increased survival probability
survivalProbability(pInd=5,pOpt=30,sigma=1)
```

`testHostSwitch` *Test for the significance of the difference between two difference of two HostSwitch objects*

Description

Test for the significance of the difference between two difference of two HostSwitch objects

Usage

```
testHostSwitch(
  simulated_quantities1,
  simulated_quantities2,
  parameter,
  test,
  warmup = NULL,
  plot = FALSE
)
```

Arguments

`simulated_quantities1`
: An object created by `simHostSwitch()`

`simulated_quantities2`
: An object created by `simHostSwitch()`

`parameter`
: Quantity of interest, possible values are:

  - "j" for total number of dispersing events (or **j**umps)
  - "s" for total number of **s**uccessful host switch events
  - "d" for **d**istance between the pRes_sim and pRes_new_sim in case of host switch

`test`
: Statistical test, available tests are:

- ”t” for **t**-test (parametric)
- ”w” for **W**ilcoxon-test (non-parametric)

`warmup` Number of warmup steps to be excluded when comparing models, see details. Possible value are NULL (default) or positive integer (min=1,max=50).

`plot` If *TRUE*, a boxplot is drawn. Default = FALSE.

Details

This function tests the significance of the difference between two objects generated by simHostSwitch function. Warmup represents the initial condition that we want to exclude from the test. The initial condition corresponds to the number of generations (n_generations): warmup = 1 means that the generation at time 0 is excluded from comparison; warmup = 2 means generations at times 0 and 1 are excluded and so on. If warmup = NULL all generations are considered for comparison, i.e. initial condition is not considered.

Value

An object of class testHostSwitch

Examples

```
m1 = simHostSwitch(n_generations=100,n_sim=100)
m2 = simHostSwitch(n_generations=50,n_sim=50)
testHostSwitch(simulated_quantities1=m1,simulated_quantities2=m2,
parameter=”j”,test=”t”,plot=TRUE)
```

Index