

Package ‘ibmpm’

July 18, 2012

Type Package

Title Individual-based metapopulation model

Version 0.1

Date 2012-06-24

Encoding UTF-8

Author Kamil Bartoń

Maintainer Kamil Bartoń <kamil.barton@uni-wuerzburg.de>

Description Individual-based metapopulation simulation including genetics

License GPL-2

R topics documented:

ibmpm-package	1
metapop	2
pop.model	3
read.structure	5
Index	7

ibmpm-package	<i>Individual-based metapopulation simulation including genetics</i>
---------------	--

Description

This package provides an individual-based metapopulation model, including genetic dynamics.

Details

The package 'ibmpm' includes a main function 'metapop', which implements a simulation model of the metapopulation dynamics. The model is individual-based, and each individuals carry a suite of neutral genes. Genes are diploid and are represented as integer numbers. Individuals reproduce sexually, offspring inherits each allele randomly from one of the parents.

Each time step is a sequence of following events: male dispersal, female dispersal, mating, reproduction. The order and number of these events can be specified. Each mated female produces a number of offspring being a function of maximum number of offspring, carrying capacity. This function (the population growth model) is defined by user. Several population models are provided, including logistic (and its stochastic variations) and truncated exponential growth.

Mutations alter the alleles according to the provided mutation function.

For more details see help page for the function [metapop](#).

Author(s)

Kamil Bartoń

metapop	<i>Metapopulation simulation</i>
---------	----------------------------------

Description

Simulate a metapopulation with genetic dynamics.

Usage

```
metapop(disperm, data, sim.len, n.offspring, carr.cap,
  event.order = c("m.disperse", "f.disperse", "mate", "reproduce"),
  pop.mod = tr.exp.growth, output.interval = NA, m.mult.mating = FALSE,
  mutation = NULL, p.mutation = 0, ...)
```

Arguments

disp.mat	a square matrix of size n.patches x n.patches containing the dispersal probabilities. The source patches are given in rows, target patches in columns, so disp.mat[i, j] is the probability of transfer from patch i to patch j. If the rows sum to less than one, the sum gives the probability of survival during the dispersal event. On the diagonal (disp.mat[i, i]) are the probabilities of staying in the patch i.
data	a data.frame with columns 'patch', 'sex', and at least one additional column for allele lengths. Each row describes one individual, or, if column n is present it describes n individuals of the same genotype, sex, and location.
sim.len	numeric, the simulation length (number of time steps).
n.offspring	maximum number of offspring per female. This can be either a numeric scalar or vector of length n.patches, in the latter case values are per patch
carr.cap	numeric scalar or vector of length n.patches, carrying capacity of the patches.
event.order	a character vector giving the sequence of events in a single time step. Accepted values are: "m.disperse", "f.disperse", "mate" and "reproduce". The names may be abbreviated. Note that there may be multiple events of the same type.

pop.mod	function implementing the population model. See pop.model .
output.interval	interval at which the population state is returned (not yet implemented).
m.mult.mating	logical, is multiple mating of males allowed?
mutation	function to carry out mutations in the 'genome' of one individual. It has to accept a numeric vector as the only argument, and return a numeric vector of the same length.
p.mutation	numeric scalar, the mutation probability
...	further arguments passed to pop.mod

Details

In the data, each two homologous alleles must occur in columns next to each other, e.g. 'allele1A', 'allele1B', 'allele2A', 'allele2B', etc.

Value

A data.frame with the same column types as the input data provided as data argument, where rows represent individuals (or genotypes/sex/location and their counts if column n was present) at time = sim.len.

- patch numeric giving the patch number, referring to the row and column of disp.mat.
- sex factor with two levels, "F" and "M";
- n optional, number of individuals of that kind;
- ... remaining columns contain allele values.

Author(s)

Kamil Bartoń

See Also

[pop.model](#)

Examples

```
# TODO
```

pop.model

Population models

Description

Population models to use with 'metapop': truncated exponential growth and variants of logistic growth: plain, with extinction at even time intervals, uniform stochastic and gamma stochastic.

Usage

```
tr.exp.growth(n, r, k, ...)
logistic.growth(n, r, k, ...)
ext.logistic.growth(n, r, k, t, ext.interval = 10L, r.prop = .5, ...)
st.logistic.growth(n, r, k, r.prop = 0.9, ...)
gst.logistic.growth(n, r, k, shape = .01, rate = 1, ...)
```

Arguments

n	population size (number of reproducing females).
r	maximum number of offspring per female.
k	carrying capacity.
t	time step.
shape, rate	parameters of the gamma distribution.
ext.interval	interval of the extinction (or population crash) events.
r.prop	proportion of r to which the growth rate decreases during the collapse (or minimum of it)
...	other arguments (not used). The user-defined function should always contain the ... argument.

Details

These function are examples of a population model function that can be used with 'metapop' (supplied as 'pop.model' argument).

Value

A vector of per-capita offspring sizes.

Author(s)

Kamil Bartoń

References

REF?

See Also

[metapop](#)

Examples

```
popmods <- c("tr.exp.growth", "logistic.growth", "st.logistic.growth",
             "gst.logistic.growth", "ext.logistic.growth")

clr <- structure(seq_along(popmods), names = popmods)

RNGkind("Mersenne")
set.seed(17)
```

```

l <- 500
k <- 100
r <- 1.05
plot(c(0, l), c(0, k), xlab = "time", ylab = "population size",
     type = "n", ylim = c(0, k * 1.05), yaxs = "i", axes = FALSE)
axis(1)
axis(2, c(0, k / c(2,1)), labels = c(0, "K/2", "K"), las = 1)
box()
for(pm in popmods) {
  x <- numeric(l)
  x[1] <- 1
  for(i in 2:l) x[i] <- x[i - 1] * get(pm)(x[i - 1], r, k, t = i,
    r.prop = .95, ext.interval = 20, shape = .001, rate = 1)
  lines(1:l, x, col = clr[pm])
}
abline(h = c(0, k), lty= 3)
legend("bottomright", popmods, text.col = clr, cex = .75,
      title = "Function:", title.adj = 0.1, lty = 1, col = clr,
      bg = "white")

```

read.structure	<i>Read 'Structure' data file</i>
----------------	-----------------------------------

Description

Read a data file formatted as input for 'Structure'

Usage

```
read.structure(file, suffix = c("A", "B"))
```

Arguments

file	file name.
suffix	suffix to be attached to column names containing homologous alleles.

Value

A data.frame, where rows represent individuals. The first column patch gives the patch number, and the remaining columns contain integer allele values.

Author(s)

Kamil Bartoń

Examples

```
#read example file:
dat <- read.structure(system.file(package = "ibmpm", "examples", "structure.txt"))

# infer sex from rownames
dat$sex <- factor(gsub("^.*([MF])[0-9]+$", "\\1", rownames(dat), perl = TRUE),
  levels = c("F", "M"))

head(dat)

n.patch <- max(dat$patch)
# random dispersal matrix
disp.mat <- array(runif(n.patch^2), dim = c(n.patch, n.patch))

dat <- metapop(disp.mat, dat, sim.len = 20, n.offspring = 5, carr.cap = 100)
```

Index

- *Topic **data**
 - read.structure, [5](#)
- *Topic **package**
 - ibmpm-package, [1](#)
- *Topic **simulation**
 - metapop, [2](#)
 - pop.model, [3](#)
- ext.logistic.growth (pop.model), [3](#)
- gst.logistic.growth (pop.model), [3](#)
- ibmpm-package, [1](#)
- logistic.growth (pop.model), [3](#)
- metapop, [2](#), [2](#), [4](#)
- pop.model, [3](#), [3](#)
- read.structure, [5](#)
- st.logistic.growth (pop.model), [3](#)
- tr.exp.growth (pop.model), [3](#)