

Package ‘ATNr’

February 7, 2022

Type Package

Title Run Allometric Trophic Networks Models

Version 1.0.1

Date 2022-01-25

Description Implements the differential equations associated to different versions of Allometric Trophic Models (ATN) to estimate the temporal dynamics of species biomasses in food webs. It offers several features to generate synthetic food webs and to parametrise models as well as a wrapper to the ODE solver deSolve.

License GPL (>= 2)

Imports Rcpp (>= 1.0.7), methods, stats, utils, deSolve

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.1.2

LazyData true

Depends R (>= 2.10)

Suggests rmarkdown, knitr, testthat, R.rsp

VignetteBuilder knitr, R.rsp

Config/testthat/edition 3

NeedsCompilation yes

Author Benoit Gauzens [cre, aut],
Emilio Berti [aut]

Maintainer Benoit Gauzens <benoit.gauzens@gmail.com>

Repository CRAN

Date/Publication 2022-02-07 11:30:02 UTC

R topics documented:

create_Lmatrix	2
create_matrix_parameter	3
create_model_Scaled	4
create_model_Unscaled	5

create_model_Unscaled_nuts	6
create_niche_model	7
initialise_default_Scaled	8
initialise_default_Unscaled	8
initialise_default_Unscaled_nuts	9
is_connected	10
Jacobian	10
lsoda_wrapper	12
plot_odeweb	13
remove_species	14
run_checks	14
Scaled	15
Scaled_loops	15
schneider	16
sort_input	17
TroLev	18
Unscaled	18
Unscaled_loops	19
Unscaled_nuts	19
Unscaled_nuts_loops	20

Index 21

create_Lmatrix	<i>Make L matrix</i>
----------------	----------------------

Description

Make L matrix

Usage

```
create_Lmatrix(BM, nb_b, Ropt = 100, gamma = 2, th = 0.01)
```

Arguments

BM	float vector, body mass of species.
nb_b	integer, number of basal species.
Ropt	numeric, consumer/resource optimal body mass ratio.
gamma	numeric, the ... of the Ricker function.
th	float, the threshold below which attack rates are considered = 0.

Details

The L matrix contains the probability for an attack event to be successful based on allometric rules and a Ricker function defined by *Ropt* and *gamma*. If at least one species has not resource or consumer (i.e. it is an isolated species), another food web is generated, until a maximum of 100 iterations.

Value

A numeric matrix with the probability for an attack event between two species to be successful.

Examples

```
set.seed(123)
mass <- sort(10 ^ runif(30, 2, 6))
L <- create_Lmatrix(mass, nb_b = 10, Ropt = 100)
image(t(L))
```

```
create_matrix_parameter
```

Make parameter matrix

Description

Make parameter matrix

Usage

```
create_matrix_parameter(BM, b0, bprey, bpred, E, T.K, T0, k)
```

Arguments

BM	float vector, body mass of species.
b0	const
bprey	const
bpred	const
E	const
T.K,	Celsius to Kelvin conversion
T0,	Default temperature in Kelvin
k,	Boltzmann constant

Details

Make a parameter matrix that depends on both predators and prey and that is used to define attack rates and handling times based on the general allometric equation:

$$p_{i,j} = b_0 * BM_i^{bprey} * BM_j^{bpred} * exp(-E * (T_0 - T.K) / (k * T.K * T_0))$$

Value

A matrix filled with estimated values for a model parameter that depends on prey and predator body masses (see details)

create_model_Scaled *Initialize an ATN model, following Delmas et al. 2017, Methods in Ecology and Evolution*

Description

Initialize an ATN model, following Delmas et al. 2017, Methods in Ecology and Evolution

Usage

```
create_model_Scaled(nb_s, nb_b, BM, fw)
```

Arguments

nb_s	integer, number of total species.
nb_b	integer, number of basal species.
BM	float vector, body mass of species.
fw	binary adjacency matrix of the food web.

Details

A model is defined by the total number of species (*nb_s*), the number of basal species (*nb_b*), the number of nutrients (*nb_n*), the body masses (*BM*) of species, and the adjacency matrix (*fw*) representing species interactions.

Value

An object of class *ATN* (*Rcpp_parameters_prefs*).

References

Delmas, E., Brose, U., Gravel, D., Stouffer, D.B. and Poisot, T. (2017), Simulations of biomass dynamics in community food webs. *Methods Ecol Evol*, 8: 881-886. <https://doi.org/10.1111/2041-210X.12713>

Examples

```
library(ATNr)
set.seed(123)
n_species <- 50
n_basal <- 20
masses <- sort(10^runif(n_species, 2, 6)) #body mass of species
L <- create_Lmatrix(masses, n_basal)
fw <- L
fw[fw > 0] <- 1
mod <- create_model_Scaled(n_species, n_basal, masses, fw)
```

create_model_Unscaled *Initialize an ATN model, following Binzer et al. 201, Global Change Biology*

Description

Initialize an ATN model, following Binzer et al. 201, Global Change Biology

Usage

```
create_model_Unscaled(nb_s, nb_b, BM, fw)
```

Arguments

nb_s	integer, number of total species.
nb_b	integer, number of basal species.
BM	float vector, body mass of species.
fw	binary adjacency matrix of the food web.

Details

A model is defined by the total number of species (*nb_s*), the number of basal species (*nb_b*), the number of nutrients (*nb_n*), the body masses (*BM*) of species, and the adjacency matrix (*fw*) representing species interactions.

Value

An object of class *ATN* (*Rcpp_parameters_prefs*).

References

Binzer, A., Guill, C., Rall, B.C. and Brose, U. (2016), Interactive effects of warming, eutrophication and size structure: impacts on biodiversity and food-web structure. *Glob Change Biol*, 22: 220-227. <https://doi.org/10.1111/gcb.13086> Gauzens, B., Rall, B.C., Mendonca, V. et al. Biodiversity of intertidal food webs in response to warming across latitudes. *Nat. Clim. Chang.* 10, 264-269 (2020). <https://doi.org/10.1038/s41558-020-0698-z>

Examples

```
library(ATNr)
set.seed(123)
n_species <- 50
n_basal <- 20
masses <- sort(10^runif(n_species, 1, 6)) #body mass of species
L <- create_Lmatrix(masses, n_basal)
fw <- L
fw[fw > 0] <- 1
mod <- create_model_Unscaled(n_species, n_basal, masses, fw)
```

```
create_model_Unscaled_nuts
```

Initialize an ATN model, following Schneider et al. 2016, Nature Communication

Description

Initialize an ATN model, following Schneider et al. 2016, Nature Communication

Usage

```
create_model_Unscaled_nuts(nb_s, nb_b, nb_n = 2, BM, fw)
```

Arguments

<code>nb_s</code>	integer, number of total species.
<code>nb_b</code>	integer, number of basal species.
<code>nb_n</code>	integer, number of nutrients.
<code>BM</code>	float vector, body mass of species.
<code>fw</code>	binary adjacency matrix of the food web.

Details

A model is defined by the total number of species (*nb_s*), the number of basal species (*nb_b*), the number of nutrients (*nb_n*), the body masses (*BM*) of species, and the adjacency matrix (*fw*) representing species interactions. Nutrients are not counted as species.

Value

An object of class *ATN* (*Rcpp_parameters_prefs*).

Examples

```
library(ATNr)
set.seed(123)
n_species <- 50
n_basal <- 20
n_nutrients <- 2
masses <- sort(10^runif(n_species, 2, 6)) #body mass of species
L <- create_Lmatrix(masses, n_basal)
fw <- L
fw[fw > 0] <- 1
mod <- create_model_Unscaled_nuts(n_species, n_basal, n_nutrients, masses, fw)
```

create_niche_model *Create a food web based on the niche model*

Description

Function to generate a food web based on the niche model (Williams and Martinez, 2000) based on the number of species and connectance. Corrections from Allesina et al. (2008) are used.

Usage

```
create_niche_model(S, C)
```

Arguments

S integer, number of species.
C numeric, connectance i.e. the number of realized links over the all possible links.

Details

If at least one species has not resource or consumer (i.e. it is an isolated species), another food web is generated, until a maximum of 100 iterations.

Value

A (square) matrix with zeros (no interaction) and ones (species j consume species i).

References

Williams, R. J., & Martinez, N. D. (2000). Simple rules yield complex food webs. *Nature*, 404(6774), 180-183.
Allesina, S., Alonso, D., & Pascual, M. (2008). A general model for food web structure. *science*, 320(5876), 658-661.

Examples

```
set.seed(123)  
web_niche <- create_niche_model(30, .1)  
image(t(web_niche))
```

initialise_default_Scaled

Default parameters for the scaled version of ATN as in Delmas et al. 2016

Description

Initialise the default parametrisation for the scaled version of the ATN model as in Delmas et al. (2016).

Usage

```
initialise_default_Scaled(model)
```

Arguments

model an object of class *Rcpp_Scaled*.

Value

An object of class *Rcpp_Scaled* with default parameters as in Delmas et al. (2017).

References

Delmas, E., Brose, U., Gravel, D., Stouffer, D.B. and Poisot, T. (2017), Simulations of biomass dynamics in community food webs. *Methods Ecol Evol*, 8: 881-886. <https://doi.org/10.1111/2041-210X.12713>

initialise_default_Unscaled

Default parameters for the scaled version of ATN as in Binzer et al. 2016, with updates from Gauzens et al. 2020

Description

Initialise the default parametrisation for the scaled version of the ATN model as in Binzer et al. (2016), with updates from Gauzens et al. 2020

Usage

```
initialise_default_Unscaled(model, temperature = 20)
```

Arguments

model an object of class *ATN (Rcpp_Unscaled)*.
 temperature numeric, ambient temperature of the ecosystem in Celsius.

Value

An object of class *ATN* (*Rcpp_Unscaled*) with default parameters as in Delmas et al. (2017).

References

Binzer, A., Guill, C., Rall, B. C. & Brose, U. Interactive effects of warming, eutrophication and size structure: impacts on biodiversity and food-web structure. *Glob. Change Biol.* 22, 220-227 (2016).
 Gauzens, B., Rall, B.C., Mendonca, V. et al. Biodiversity of intertidal food webs in response to warming across latitudes. *Nat. Clim. Chang.* 10, 264-269 (2020). <https://doi.org/10.1038/s41558-020-0698-z>

initialise_default_Unscaled_nuts

Default model parameters as in Schneider et al. 2016

Description

Initialise the default parametrisation for the model for Schneider et al. (2016).

Usage

```
initialise_default_Unscaled_nuts(model, L.mat, temperature = 20)
```

Arguments

model	an object of class <i>ATN</i> (<i>Rcpp_Unscaled_nuts</i>).
L.mat	numeric matrix, probability of a consumer to attack and capture an encountered resource. See create_Lmatrix .
temperature	numeric, ambient temperature of the ecosystem in Celsius.

Value

An object of class *ATN* (*Rcpp_Unscaled_nuts*) with default parameters as in Schneider et al. (2016).

References

Schneider, F. D., Brose, U., Rall, B. C., & Guill, C. (2016). Animal diversity and ecosystem functioning in dynamic food webs. *Nature Communications*, 7(1), 1-8.

is_connected	<i>Detect whether a food web is composed of several disconnected sub-networks</i>
--------------	---

Description

Run a deep search first algorithm (DFS)

Usage

```
is_connected(fw)
```

Arguments

fw binary adjacency matrix of the food web.

Value

Boolean: TRUE if the food web is connected, FALSE if several disconnected sub-networks are detected.

Examples

```
library(ATNr)
set.seed(123)
# number of species, nutrients, and body masses
n_species <- 20
n_basal <- 5
n_nutrients <- 3
masses <- sort(10^runif(n_species, 2, 6)) #body mass of species
# create food web matrix
L <- create_Lmatrix(masses, n_basal)
L[, 1:n_basal] <- 0
fw <- L
fw[fw > 0] <- 1
connected <- is_connected(fw)
```

Joacobian	<i>Estimate the Jacobian matrix of a ODE system</i>
-----------	---

Description

Estimate the Jacobian matrix of a ODE system

Usage

```
Joacobian(bioms, ODE, eps = 1e-06)
```

Arguments

bioms	float vector, biomass of species.
ODE	function that computes the ODEs from one of the model available
eps	float, scale precision of the numerical approximation.

Details

The function provides a numerical estimation of the Jacobian matrix based on the 5 points stencil method. The precision of the method is in

$$O(h^5)$$

, where

$$h = eps * bioms$$

. The choice of eps should ensure that

$$h^5$$

is always lower to the extinction threshold.

The dimension of the Jacobian matrix are not always matching the number of species in the system. This is because we considered that a perturbation can not correspond to the recolonisation of an extinct species. Therefore, extinct species are removed from the system to calculate the Jacobian matrix.

Value

A matrix corresponding to the Jacobian of the system estimated at the parameter biomasses

Examples

```
library(ATNr)
set.seed(123)
# first run a model to reach equilibrium
masses <- runif(20, 10, 100) #body mass of species
L <- create_Lmatrix(masses, 10, Ropt = 10)
L[L > 0] <- 1
mod <- create_model_Unscaled_nuts(20, 10, 3, masses, L)
mod <- initialise_default_Unscaled_nuts(mod, L)
biomasses <- masses ^ -0.75 * 10 ^ 4 #biomasses of species
biomasses <- append(runif(3, 20, 30), biomasses)
times <- seq(0, 100, 1)
sol <- lsoda_wrapper(times, biomasses, mod)
# get the final biomasses
final.bioms = sol[nrow(sol), -1]
# estimate jacobian
Joacobian(final.bioms, mod$ODE)
```

lsoda_wrapper	<i>Wrapper for lsoda</i>
---------------	--------------------------

Description

This is a wrapper to call `lsoda` from *deSolve* and solve the ODE. Package `deSolve` needs to be installed to run this wrapper.

Usage

```
lsoda_wrapper(t, y, model, verbose = FALSE, ...)
```

Arguments

<code>t</code>	vector of times.
<code>y</code>	vector of biomasses.
<code>model</code>	object of class <i>ATN</i> (<i>Rcpp_parameters_prefs</i>).
<code>verbose</code>	Boolean, whether a message should be printed when all checks were successful
<code>...</code>	additional arguments to pass to ‘ <code>lsoda</code> ’

Value

A matrix for the ODE solution with species as columns and times as rows.

Examples

```
library(ATNr)
set.seed(123)
masses <- runif(20, 10, 100) #body mass of species
L <- create_Lmatrix(masses, 10, Ropt = 10)
L[L > 0] <- 1
mod <- create_model_Unscaled_nuts(20, 10, 3, masses, L)
mod <- initialise_default_Unscaled_nuts(mod, L)
biomasses <- masses ^ -0.75 * 10 ^ 4 #biomasses of species
biomasses <- append(runif(3, 20, 30), biomasses)
times <- seq(0, 100, 1)
sol <- lsoda_wrapper(times, biomasses, mod)
```

plot_odeweb

Plot food web dynamics

Description

Plot solution of the ODE for the food web. Currently only species and not nutrients are plotted.

Usage

```
plot_odeweb(x, nb_s)
```

Arguments

`x` matrix with solutions. First row should be the time vector.
`nb_s` numeric, number of species as in the model (e.g., `create_model_Unscaled_nuts`).

Value

No return value, called for side effects.

Examples

```
## Not run:
library(ATNr)
library(deSolve)
set.seed(123)
# number of species, nutrients, and body masses
n_species <- 20
n_basal <- 5
n_nutrients <- 3
masses <- sort(10*runif(n_species, 2, 6)) #body mass of species
# create food web matrix
L <- create_Lmatrix(masses, n_basal)
L[, 1:n_basal] <- 0
fw <- L
fw[fw > 0] <- 1
model <- create_model_Unscaled_nuts(
  n_species,
  n_basal,
  n_nutrients,
  masses,
  fw
)
# initialize model as default in Schneider et al. (2016)
model <- initialise_default_Unscaled_nuts(model, L)
model$initialisations()
# defining integration time
times <- seq(0, 500, 5)
biomasses <- runif(n_species + n_nutrients, 2, 3)
```

```
sol <- lsoda_wrapper(times, biomasses, model, verbose = FALSE)
plot_odeweb(sol, model$nb_s)

## End(Not run)
```

remove_species *Function to remove species from a model class*

Description

Function to remove species from a model class

Usage

```
remove_species(species, model, nuts = NULL)
```

Arguments

species	integer vector, the indices of species to remove.
model	model object
nuts	integer vector, the indices of nutrients to remove. Parameter specific to the Un-scaled_nuts model.

Value

A model object where the data structure has been updated to remove the species in parameters.

run_checks *Run checks on model parameters*

Description

Check if the dimensions of vectors and matrices used in the model are correct. If any dimension is not correct, an error message is returned.

Usage

```
run_checks(model, verbose = TRUE)
```

Arguments

model	a model object.
verbose	Boolean, whether a message should be printed when all checks were successful

Value

No return value, only throw an error if parameters are inconsistent.

Scaled	<i>Store parameters and functions associated to the scaled version of ATN</i>
--------	---

Description

Type the name of the class to see its methods

Fields

nb_s Total number of species
 nb_b Number of basal species
 c double: interference competition
 X Vector of metabolic rates (length = number of species)
 max_feed Vector of maximum feeding rates (length = number of consumers)
 e Vector of assimilation efficiencies (length = number of species)
 r Vector of producers maximum growth rates (length = number of basal species)
 BM Vector of body masses (length = number of species)
 dB Vector of local derivatives (length = number of species)
 B0 Vector of half saturation densities (length = number of consumers)
 fw Adjacency matrix of the food-web (dim = number of species * number of species)
 w Matrix of relative consumption rates (dim = number of species * number of consumers)
 F Matrix of per-capita feeding rates (dim = number of species * number of consumers)
 q hill exponent for the type of functional response
 K Carrying capacity of basal species
 ext Extinction threshold for species
 alpha Plant resource competition
 ODE Calculate the derivatives for the scaled version of the ATN model

- Parameter: bioms - Local species biomasses
- Parameter: t - Integration time point
- Returns a vector of growth rate for each species at time t

Scaled_loops	<i>Store parameters and functions associated to the scaled version of ATN</i>
--------------	---

Description

To not use. For testing purpose only. Please use Rcpp_Scaled instead.

 schneider

Default parameters as in Schneider et al. (2016)

Description

A dataset containing the default parameters as in the Schneider et al. (2016) and used to parametrize the default models. See also `create_model_Unscaled_nuts`, `create_Lmatrix`, `initialise_default_Unscaled_nuts`.

Usage

schneider

Format

A list with the default parameters:

Temperature ambient temperature in Celsius
T.K default temperature, 20 degree Celsius in Kelvin
k Boltzmann's constant
T0 20 degree Celsius in Kelvin, used to estimate scaling law of metabolic rates
q Hill's exponent of the functional response
Ropt consumer/resource optimal body mass ratio
gamma shape of the Ricker function
mu_c average predator interference
sd_c standard deviation of predator interference
E.c Activation energy for interference
h0 scaling constant of the power-law of handling time with consumer and resource body mass
hpred exponent associated to predator body mass for the allometric scaling of handling time
hprey exponent associated to prey body mass for the allometric scaling of handling time
E.h Activation energy for handling time
b0 normalisation constant for capture coefficient
bprey exponent associated to prey body mass for the allometric scaling of capture coefficient
bpred exponent associated to predator body mass for the allometric scaling of capture coefficient
E.b Activation energy for capture coefficient
e_P Assimilation efficiency associated to the consumption of a plant species
e_A Assimilation efficiency associated to the consumption of an animal species
x_P scaling constant of the power-law of metabolic demand per unit of plant biomass
x_A scaling constant of the power-law of metabolic demand per unit of animal biomass
E.x Activation energy for metabolic rates
expX TBD

D turnover rate of nutrients
nut_up_min Minimum uptake efficiency of plants
nut_up_max Maximum uptake efficiency of plants
mu_nut Average maximum nutrient concentration
sd_nut standard deviation of maximum nutrient concentration
v relative content of nutrient 1 in plant biomass

References

Schneider, F. D., Brose, U., Rall, B. C., & Guill, C. (2016). Animal diversity and ecosystem functioning in dynamic food webs. *Nature Communications*, 7(1), 1-8.

sort_input	<i>Sort custom input</i>
------------	--------------------------

Description

Sort custom input

Usage

```
sort_input(BM, fw)
```

Arguments

BM	numeric vector, body mass of species.
fw	adjacency matrix of the food web.

Details

Body masses and food web matrix should be arranged with the first elements/columns being for basal species. This is a requirement for the Cpp class and must be enforced before initializing the Rcpp_Schneider and Rcpp_Delmas objects.

Value

A list with sorted body masses (*body.mass*) and food web matrix (*food.web*).

Examples

```
bm <- runif(10, 10, 50)
fw <- matrix(as.numeric(runif(100) > .9), 10, 10)
sort_input(bm, fw)
```

TroLev	<i>Calculate trophic level of species</i>
--------	---

Description

Calculate trophic level of species

Usage

TroLev(fw)

Arguments

fw numeric matrix, the matrix of the food web.

Value

A numeric vector of species' trophic level.

Unscaled	<i>Store parameters and functions associated to the unscaled version of ATN</i>
----------	---

Description

Type the name of the class to see its methods

Fields

nb_s Total number of species
 nb_b Number of basal species
 c double: interference competition
 X Vector of metabolic rates (length = number of species)
 a Matrix of attack rates (dim = number of species * number of consumers)
 h Matrix of handling times (dim = number of species * number of consumers)
 e Vector of assimilation efficiencies (length = number of species)
 r Vector of producers maximum growth rates (length = number of basal species)
 BM Vector of body masses (length = number of species)
 dB Vector of local derivatives (length = number of species)
 fw Adjacency matrix of the food-web (dim = number of species * number of species)
 F Matrix of per-capita feeding rates (dim = number of species * number of consumers)
 q hill exponent for the type of functional response

- K Carrying capacity of basal species
 alpha Plant resource competition
 ext Extinction threshold for species
 ODE Calculate the derivatives for the scaled version of the ATN model
- Parameter: bioms - Local species biomasses
 - Parameter: t - Integration time point
 - Returns a vector of growth rate for each species at time t

Unscaled_loops	<i>Store parameters and functions associated to the unscaled version of ATN</i>
----------------	---

Description

To not use. For testing purpose only. Please use Rcpp_Unscaled instead.

Unscaled_nuts	<i>Store parameters and functions associated to the unscaled version of ATN including nutrient dynamics</i>
---------------	---

Description

Type the name of the class to see its methods

Fields

- nb_s Total number of species
 nb_b Number of basal species
 nb_n Number of nutrient pool
 c double: interference competition
 b Matrix of attack rates (dim = number of species * number of consumers)
 h Matrix of handling times (dim = number of species * number of consumers)
 X vector of metabolic rates (length = number of species)
 K matrix of plant nutrient efficiencies (dim = number of nutrients * number of plants)
 V matrix of plant relative nutrient content (dim = number of nutrients * number of plants)
 S Vector of maximum nutrient concentration (length = number of plants)
 r Vector of maximum growth rate of plant species (length = number of plant species)
 e Vector of assimilation efficiencies (length = number of species)
 BM Vector of body masses (length = number of species)
 dB Vector of local derivatives (length = number of species)

fw Adjacency matrix of the food-web (dim = number of species * number of species)
w Matrix of relative consumption rates (dim = number of species * number of consumers)
F Matrix of per-capita feeding rates (dim = number of species * number of consumers)
q hill exponent for the type of functional response
ext Extinction threshold for species
ODE Calculate the derivatives for the scaled version of the ATN model

- Parameter: bioms - Local species biomasses
- Parameter: t - Integration time point
- Returns a vector of growth rate for each species at time t

Unscaled_nuts_loops *Store parameters and functions associated to the unscaled version of ATN*

Description

To not use. For testing purpose only. Please use Rcpp_Unscaled_nuts instead.

Index

* datasets

- schneider, [16](#)

- create_Lmatrix, [2](#), [9](#)
- create_matrix_parameter, [3](#)
- create_model_Scaled, [4](#)
- create_model_Unscaled, [5](#)
- create_model_Unscaled_nuts, [6](#)
- create_niche_model, [7](#)

- initialise_default_Scaled, [8](#)
- initialise_default_Unscaled, [8](#)
- initialise_default_Unscaled_nuts, [9](#)
- is_connected, [10](#)

- Joacobian, [10](#)

- lsoda_wrapper, [12](#)

- plot_odeweb, [13](#)

- remove_species, [14](#)
- run_checks, [14](#)

- Scaled, [15](#)
- Scaled_loops, [15](#)
- schneider, [16](#)
- sort_input, [17](#)

- TroLev, [18](#)

- Unscaled, [18](#)
- Unscaled_loops, [19](#)
- Unscaled_nuts, [19](#)
- Unscaled_nuts_loops, [20](#)