Package: Rpath (via r-universe)

October 3, 2024

Type Package

Title R implementation of Ecopath with Ecosim

Version 0.9.1

Date 2024-05-31

Description This package implements the core mass balance equations made popular by Ecopath with Ecosim. There are two parts to the package, the static snapshot of energy flow through the system using rpath and dynamic simulations run using the rsim family of functions.

License file LICENSE

Encoding UTF-8

Imports data.table, methods, graphics, grDevices, MASS, Rcpp (>= 0.11.3), stats, utils

LinkingTo Rcpp

Suggests here, knitr, rmarkdown, distillery, ggplot2, ggpubr, rlist, stringr, viridis, qpdf, usethis, dplyr, generics, testthat (>= 3.0.0)

VignetteBuilder knitr

RoxygenNote 7.3.1

URL https://github.com/NOAA-EDAB/Rpath

BugReports https://github.com/NOAA-EDAB/Rpath/issues

Depends R (>= 2.10)

Config/testthat/edition 3

LazyData true

Repository https://noaa-fisheries-integrated-toolbox.r-universe.dev

RemoteUrl https://github.com/NOAA-EDAB/Rpath

RemoteRef HEAD

RemoteSha 3c75a6d86d8705f42ef41e5755f49c21dba3dec0

Contents

Rpath-package	*Rpath: A package implementing mass balance algorithms designed to work with fisheries data sources.*

Description

The Rpath package provides two categories of important functions: rpath and rsim.

rpath functions

The rpath functions generate the balanced snap shot of energy flow through the system. This is done by solving a series of linear equations for unknown biomass or ecotrophic efficiency.

Author(s)

Maintainer: Sean Lucey <Sean.Lucey@NOAA.gov>

Authors:

- Kerim Aydin <Kerim.Aydin@NOAA.gov>
- Sarah Gaichas <Sarah.Gaichas@NOAA.gov>

See Also

Useful links:

- <https://github.com/NOAA-EDAB/Rpath>
- Report bugs at <https://github.com/NOAA-EDAB/Rpath/issues>

AB.params	*AB.params*

Description

A tutorial ecosystem for teaching Rpath

Usage

```
AB.params
```

Format

An Rpath balanced model object that contains:

model A data.table containing Ecopath unbalanced ecosystem parameters (base parameters and fisheries).

diet A data.table containing the Ecopath model's diet matrix.

stanzas Parameters for multistanza groups.

pedigree A data.table containing the data quality (pedigree) for the Ecopath model. ...

---

adjust.fishing   *Adjust Fishing Mortality*

Description

Modifies the fishing mortality value for a species by a particular gear. Parameters that can be adjusted using this function are: *ForcedEffort*, *ForcedFRate*, or *ForcedCatch*.

Usage

```
adjust.fishing(
  Rsim.scenario,
  parameter,
  group = NA,
  sim.year = 1,
  sim.month = 0,
  value
)
```

Arguments

Rsim.scenario	Scenario object that contains all of the rsim rates and forcing functions generated by rsim.scenario.
parameter	Parameters to be modified (See Description)
group	Name of the group whose parameter is being changed.
sim.year	Year of the simulation that should be modified. Can be a range of years.
sim.month	Month of the year that should be modified. If set to 0, all months of the year are modified.
value	New value for the parameter.

Value

Returns an Rsim.scenario object with the new fishing parameter values.

See Also

Other Adjust functions: [adjust.scenario](#)()

adjust.forcing	*Adjust Forcing Parameters*

Description

Modifies the various forcing parameters of the rsim scenario object.

Usage

```
adjust.forcing(
  Rsim.scenario,
  parameter,
  group,
  sim.year = 1,
  sim.month = 0,
  bymonth = F,
  value
)
```

Arguments

`Rsim.scenario`	Object generated by rsim.scenario.
`parameter`	The Rsim.scenario forcing parameter to be modified.
`group`	The model group that the parameter change will affect.
`sim.year`	Year of the simulation that should be modified. Can be a range of years.
`sim.month`	Month of the year that should be modified. If set to 0, all months of the year are modified.
`bymonth`	Boolean value that denotes whether to use sim.year/sim.month combo or just sim.month as a sequential vector starting at 1.
`value`	The new value for the parameter.

Value

Returns an Rsim.scenario object with the new parameter.

See Also

Other Rpath functions: `MTI()`, `check.rpath.params()`, `create.rpath.params()`, `extract.node()`, `frate.table()`, `read.rpath.params()`, `rpath()`, `rpath.groups()`, `rpath.stanzas()`, `stanzaplot()`, `webplot()`, `write.Rpath()`, `write.Rsim()`, `write.rpath.params()`

---

adjust.scenario   *Adjust Rsim.scenario parameters*

Description

Modifies the various parameters of the `rsim.scenario` object. Parameters that can be adjusted using this function are:

Usage

```
adjust.scenario(Rsim.scenario, parameter, group, groupto = NA, value)
```

Arguments

Rsim.scenario	Scenario object that contains all of the rsim rates and forcing functions generated by `rsim.scenario`.
parameter	Parameters to be modified (See Description)
group	The model group that the parameter change will affect. Note that a value of *'all'* will affect all groups associate with the groupto variable.
groupto	The corresponding group who's parameter is affecting the group variable.
value	New value for the parameter.

Value

Returns an `Rsim.scenario` object with the new parameter.

See Also

Other Adjust functions: [adjust.fishing](#)()

---

check.rpath.params   *Check Rpath parameter files*

Description

Logical check that the parameter files are filled out correctly, i.e. data is entered where it is expected.

Usage

```
check.rpath.params(Rpath.params)
```

Arguments

Rpath.params	R object containing the Rpath parameters. This is generated either by the create.rpath.params or read.rpath.params functions.

Value

Checks Rpath parameter files for consistency. An error message will be produced if one of the logical checks fails. Checks include: (NOTE: This does not ensure data is correct just that it is in the right places).

See Also

Other Rpath functions: MTI(), adjust.forcing(), create.rpath.params(), extract.node(), frate.table(), read.rpath.params(), rpath(), rpath.groups(), rpath.stanzas(), stanzaplot(), webplot(), write.Rpath(), write.Rsim(), write.rpath.params()

---

create.rpath.params   *Creates a shell for the Rpath.params object*

Description

Creates a shell of the Rpath.params list object which contains the model, diet, multistanza, and pedigree parameters.

Usage

```
create.rpath.params(group, type, stgroup = NA)
```

Arguments

group	Vector of group names.
type	Numeric vector of group type. Living = 0, Producer = 1, Detritus = 2, Fleet = 3.
stgroup	Vector of multistanza group names. Include NA for non-stanza groups.

Value

Outputs a list object of Rpath parameters which are populated with values of NA or logical default values. Values can then be filled in using R. Use check.rpath.params() to ensure parameter files are filled out correctly (NOTE: This does not ensure data is correct just that it is in the right places).

See Also

Other Rpath functions: MTI(), adjust.forcing(), check.rpath.params(), extract.node(), frate.table(), read.rpath.params(), rpath(), rpath.groups(), rpath.stanzas(), stanzaplot(), webplot(), write.Rpath(), write.Rsim(), write.rpath.params()

Ecosense.EBS   *Eastern Bering Sea 1990s Ecopath model (53 biological groups and 1*
                                fleet).

Description

Rpath inputs (unbalanaced model).

Usage

Ecosense.EBS

Format

An Rpath unbalanced model object that contains:

model A data.table containing Ecopath unbalanced ecosystem parameters (base parameters and
fisheries).

diet A data.table containing the Ecopath model's diet matrix.

stanzas Parameters for multistanza groups.

pedigree A data.table containing the data quality (pedigree) for the Ecopath model. ...

Source

Whitehouse and Aydin 2020. Assessing the sensitivity of three Alaska marine food webs to pertur-
bations: an example of Ecosim simulations using Rpath. https://doi.org/10.1016/j.ecolmodel.2020.109074

Ecosense.ECS   *Eastern Chukchi Sea Ecopath model (52 biological groups and 1 fleet).*

Description

Rpath inputs (unbalanaced model).

Usage

Ecosense.ECS

Format

An Rpath unbalanced model object that contains:

model A data.table containing Ecopath unbalanced ecosystem parameters (base parameters and
fisheries).

diet A data.table containing the Ecopath model's diet matrix.

stanzas Parameters for multistanza groups.

pedigree A data.table containing the data quality (pedigree) for the Ecopath model. ...

Source

Whitehouse and Aydin 2020. Assessing the sensitivity of three Alaska marine food webs to perturbations: an example of Ecosim simulations using Rpath. https://doi.org/10.1016/j.ecolmodel.2020.109074

Ecosense.GOA	*Gulf of Alaska (west/central) 1990s Ecopath model (49 biological groups and 1 fleet).*

Description

Rpath inputs (unbalanaced model).

Usage

```
Ecosense.GOA
```

Format

An Rpath unbalanced model object that contains:

model A data.table containing Ecopath unbalanced ecosystem parameters (base parameters and fisheries).

diet A data.table containing the Ecopath model's diet matrix.

stanzas Parameters for multistanza groups.

pedigree A data.table containing the data quality (pedigree) for the Ecopath model. ...

Source

Whitehouse and Aydin 2020. Assessing the sensitivity of three Alaska marine food webs to perturbations: an example of Ecosim simulations using Rpath. https://doi.org/10.1016/j.ecolmodel.2020.109074

---

extract.node	*Extract node data*

Description

Creates a list object with node specific data using Rsim.output object.

Usage

```
extract.node(Rsim.output, group)
```

Arguments

Rsim.output	Object generated by rsim.run.
group	Name of the node for which you want to extract data.

Value

Returns a list object with node specific data.

See Also

Other Rpath functions: MTI(), adjust.forcing(), check.rpath.params(), create.rpath.params(), frate.table(), read.rpath.params(), rpath(), rpath.groups(), rpath.stanzas(), stanzaplot(), webplot(), write.Rpath(), write.Rsim(), write.rpath.params()

---

    frate.table   *Fishing Mortality Table*

Description

Creates a table of fishing mortalities by species group and gear for an Rsim.scenario object.

Usage

```
frate.table(Rsim.scenario)
```

Arguments

Rsim.scenario   Scenario object that contains all of the rsim rates and forcing functions generated by rsim.scenario.

Value

Returns a data table of F values for each species/gear combination.

See Also

Other Rpath functions: MTI(), adjust.forcing(), check.rpath.params(), create.rpath.params(), extract.node(), read.rpath.params(), rpath(), rpath.groups(), rpath.stanzas(), stanzaplot(), webplot(), write.Rpath(), write.Rsim(), write.rpath.params()

get.rsim.fishing *Retrieve Rsim fishing forcing parameters*

Description

Helper function that will retrieve the fishing forcing parameters that were used in an Rsim scenario

Usage

```
get.rsim.fishing(Rsim.scenario)
```

Arguments

Rsim.scenario   Scenario object that contains all of the rsim rates and forcing functions generated by rsim.scenario.

Value

Returns an Rsim.fishing object.

See Also

Other Get functions: `get.rsim.forcing()`, `get.rsim.params()`, `get.rsim.stanzas()`, `get.rsim.start_state()`

get.rsim.forcing *Retrieve Rsim forcing parameters*

Description

Helper function that will retrieve the forcing parameters that were used in an Rsim scenario

Usage

```
get.rsim.forcing(Rsim.scenario)
```

Arguments

Rsim.scenario   Scenario object that contains all of the rsim rates and forcing functions generated by rsim.scenario.

Value

Returns an Rsim.forcing object.

See Also

Other Get functions: `get.rsim.fishing()`, `get.rsim.params()`, `get.rsim.stanzas()`, `get.rsim.start_state()`

---

get.rsim.params   *Retrieve Rsim scenario parameters*

Description

Helper function that will retrieve parameters that were used in an Rsim scenario

Usage

```
get.rsim.params(Rsim.scenario)
```

Arguments

Rsim.scenario   Scenario object that contains all of the rsim rates and forcing functions generated
                 by rsim.scenario.

Value

Returns an Rsim.params object.

See Also

Other Get functions: get.rsim.fishing(), get.rsim.forcing(), get.rsim.stanzas(), get.rsim.start_state()

---

get.rsim.stanzas   *Retrieve Rsim stanza parameters*

Description

Helper function that will retrieve the stanza parameters that were used in an Rsim scenario

Usage

```
get.rsim.stanzas(Rsim.scenario)
```

Arguments

Rsim.scenario   Scenario object that contains all of the rsim rates and forcing functions generated
                 by rsim.scenario.

Value

Returns an Rsim.stanzas object.

See Also

Other Get functions: get.rsim.fishing(), get.rsim.forcing(), get.rsim.params(), get.rsim.start_state()

---

get.rsim.start_state *Retrieve Rsim starting state values*

Description

Helper function that will retrieve the starting state values that were used in an Rsim scenario

Usage

```
get.rsim.start_state(Rsim.scenario)
```

Arguments

Rsim.scenario Scenario object that contains all of the rsim rates and forcing functions generated by rsim.scenario.

Value

Returns an Rsim.state object.

See Also

Other Get functions: get.rsim.fishing(), get.rsim.forcing(), get.rsim.params(), get.rsim.stanzas()

MTI *Mixed Trophic Impacts*

Description

Need to edit.

Usage

```
MTI(Rpath, Rpath.params, increase = T)
```

Arguments

Rpath   Balanced Rpath model generated by rpath.

Rpath.params R object containing the Rpath parameters. This is generated either by the create.rpath.params or read.rpath.params functions.

increase   Logical value indicating whether a marginal increase is applied.

Value

Returns a matrix of mixed trophic impacts.

See Also

Other Rpath functions: adjust.forcing(), check.rpath.params(), create.rpath.params(), extract.node(), frate.table(), read.rpath.params(), rpath(), rpath.groups(), rpath.stanzas(), stanzaplot(), webplot(), write.Rpath(), write.Rsim(), write.rpath.params()

---

read.rpath.params   *Read Rpath parameters from .csv files*

Description

Creates an Rpath.param object from a series of .csv files.

Usage

```
read.rpath.params(
  modfile,
  dietfile,
  pedfile = NA,
  stanzagroupfile = NA,
  stanzafile = NA
)
```

Arguments

modfile	file location of the flat file containing the model parameters.
dietfile	file location of the flat file containing the diet parameters.
pedfile	file location of the flat file containg the pedgigree parameters.
stanzagroupfile	
	file location of the flat file containing the group parameters for multistanza groups. If not specified a blank stanza list will be created.
stanzafile	file location of the flat file containing the individual stanza parameters for multistanza groups. If not specified a blank stanza list will be created.

Value

Outputs an Rpath.param object that can be used for Rpath and subsequently Rsim. (NOTE: This does function does not ensure data is correct or in the correct locations...run check.rpath.param to ensure the appropriate columns are present).

See Also

Other Rpath functions: MTI(), adjust.forcing(), check.rpath.params(), create.rpath.params(), extract.node(), frate.table(), rpath(), rpath.groups(), rpath.stanzas(), stanzaplot(), webplot(), write.Rpath(), write.Rsim(), write.rpath.params()

REco.params   *REco.params*

Description

A sample ecosystem for Rpath - Anchovy Bay.

Usage

```
REco.params
```

Format

An Rpath balanced model object that contains:

model A data.table containing Ecopath unbalanced ecosystem parameters (base parameters and fisheries).

diet A data.table containing the Ecopath model's diet matrix.

stanzas Parameters for multistanza groups.

pedigree A data.table containing the data quality (pedigree) for the Ecopath model. ...

---

rpath   *Ecopath module of Rpath*

Description

Performs initial mass balance using a model parameter file and diet matrix file.

Usage

```
rpath(Rpath.params, eco.name = NA, eco.area = 1)
```

Arguments

Rpath.params	R object containing the Rpath parameters. This is generated either by the create.rpath.params or read.rpath.params functions.
eco.name	Optional name of the ecosystem which becomes an attribute of rpath object.
eco.area	Optional area of the ecosystem which becomes an attribute of the rpath object.

Value

Returns an Rpath object that can be supplied to the rsim.scenario function.

See Also

Other Rpath functions: MTI(), adjust.forcing(), check.rpath.params(), create.rpath.params(), extract.node(), frate.table(), read.rpath.params(), rpath.groups(), rpath.stanzas(), stanzaplot(), webplot(), write.Rpath(), write.Rsim(), write.rpath.params()

---

rpath.groups   *Rpath functional group names*

Description

Get a character vector of functional group names from an Rpath object (balanced model) or an Rpath.params object (unbalanced model parameters) based on the 'type' input parameter as follows: (0: consumers, 1: producers, 2: detrital, 3: gears, 0<type<1: mixotrophs). Living groups are consumers + producers. Note that mixotrophs are not returned as either consumers or producers, only separately.

Usage

```
rpath.groups(Rpath)

rpath.living(Rpath)

rpath.detrital(Rpath)

rpath.gears(Rpath)

rpath.producers(Rpath)

rpath.consumers(Rpath)

rpath.mixotrophs(Rpath)
```

Arguments

Rpath   Balanced Rpath model generated by rpath.

Value

Returns a character vector containing the names of Rpath functional groups by category (group type).

See Also

Other Rpath functions: MTI(), adjust.forcing(), check.rpath.params(), create.rpath.params(), extract.node(), frate.table(), read.rpath.params(), rpath(), rpath.stanzas(), stanzaplot(), webplot(), write.Rpath(), write.Rsim(), write.rpath.params()

---

rpath.stanzas *Calculate biomass and consumption for multistanza groups*

Description

Uses the leading stanza to calculate the biomass and consumption of other stanzas necessary to support the leading stanza.

Usage

```
rpath.stanzas(Rpath.params)
```

Arguments

Rpath.params   R object containing the Rpath parameters. This is generated either by the create.rpath.params or read.rpath.params functions.

Value

Calculates and adds biomass and consumption for trailing stanza groups. Also adds weight at age and number at age for multi-staza groups.

See Also

Other Rpath functions: [MTI()](), [adjust.forcing()](), [check.rpath.params()](), [create.rpath.params()](),
[extract.node()](), [frate.table()](), [read.rpath.params()](), [rpath()](), [rpath.groups()](), [stanzaplot()](),
[webplot()](), [write.Rpath()](), [write.Rsim()](), [write.rpath.params()]()

---

rsim.deriv *Calculate the derivatives for a time step*

Description

Calculates the derivative for a single time step and saves the output

Usage

```
rsim.deriv(Rsim.scenario, sim.year = 0, sim.month = 0, tstep = 0)
```

Arguments

Rsim.scenario   Scenario object that contains all of the rsim rates and forcing functions generated by rsim.scenario.

sim.year   Will inherit from apply functions

sim.month   Will inherit from apply functions

tstep   Sub-monthly time step usually set to 0.

---

rsim.diet   *Output consumption by a group*

Description

Creates a matrix of consumption of prey by a particular predator.

Usage

```
rsim.diet(Rsim.output, group)
```

Arguments

Rsim.output	R object containing the output from rsim.run.
group	Group from the Rpath model that is of interest

---

rsim.fishing   *Generate Rsim fishing matrix*

Description

Creates a matrix for forcing functions related to fishing (*ForcedEffort*, *ForcedFRate*, *ForcedCatch*)

Usage

```
rsim.fishing(params, years)
```

Arguments

params	Rsim parameter object generated by rsim.params.
years	A vector of each year of the simulation.

---

rsim.forcing *Generate Rsim forcing matrix*

Description

Creates a matrix for forcing functions not related to fishing (*ForcedPrey ForcedMort*, *ForcedRecs*, *ForcedSearch*, *ForcedMigrate*, *ForcedBio*).

Usage

```
rsim.forcing(params, years)
```

Arguments

params   Rsim parameter object generated by `rsim.params`.

years   A vector of each year of the simulation.

---

rsim.mort *Output mortality on a group*

Description

Creates a matrix of mortality by predators on a particular prey.

Usage

```
rsim.mort(Rsim.output, group)
```

Arguments

Rsim.output   R object containing the output from `rsim.run`.

group   Group from the `Rpath` model that is of interest

---

rsim.params	*Initial set up for Rsim module of Rpath*

Description

Converts the outputs from Rpath into rates for use in Rsim.

Usage

```
rsim.params(
  Rpath,
  mscramble = 2,
  mhandle = 1000,
  preyswitch = 1,
  scrambleselfwt = 0,
  handleselfwt = 0,
  steps_yr = 12,
  steps_m = 1
)
```

Arguments

Rpath	R object containing a balanced Rpath model.
mscramble	WILL REMOVE
mhandle	WILL REMOVE
preyswitch	WILL REMOVE - Adjust with adjust.scenario
scrambleselfwt	Value of 1 indicates no overlap while 0 indicates complete overlap.
handleselfwt	Value of 1 indicates no overlap while 0 indicates complete overlap.
steps_yr	Number of time steps per year.
steps_m	Number of time steps per month.

Value

Returns an Rsim.params object that is passed to the rsim.run function via the rsim.scenario function.

See Also

Other Rsim functions: rsim.plot(), rsim.run(), rsim.scenario(), rsim.step()

---

rsim.plot *Plot routine for Rsim runs*

Description

Plots the relative biomass of each group from an `Rsim.output` object.

Usage

```
rsim.plot(Rsim.output, spname = "all", indplot = F)
```

Arguments

`Rsim.output`	R object containing the output from `rsim.run`.
`spname`	Vector of species names to be displayed. The default "all" will display all group names.
`indplot`	Logical value of whether to plot a single group or multiple groups

Value

Creates a figure of relative biomass.

See Also

Other Rsim functions: `rsim.params()`, `rsim.run()`, `rsim.scenario()`, `rsim.step()`

---

rsim.run *Run Rsim*

Description

Carries out the numerical integration of the Rsim alogrithms.

Usage

```
rsim.run(Rsim.scenario, method = "RK4", years = 1:100)
```

Arguments

`Rsim.scenario`	Scenario object that contains all of the rsim rates and forcing functions generated by `rsim.scenario`.
`method`	Numerical integration method. Either *'AB'* for Adams-Bashforth or *'RK4'* for 4th order Runge-Kutta.
`years`	A vector of each year of the simulation.

See Also

Other Rsim functions: rsim.params(), rsim.plot(), rsim.scenario(), rsim.step()

---

rsim.scenario	*Rsim module of* Rpath

Description

Uses a balanced Rpath model and creates a scenario consisting of 5 list objects: params, start_state, forcing, fishing, and stanzas.

Usage

```
rsim.scenario(Rpath, Rpath.params, years = 1:100)
```

Arguments

Rpath	R object containing a balanced Rpath model.
Rpath.params	R object containing the Rpath parameters. This is generated either by the create.rpath.params or read.rpath.params functions.
years	A vector of each year of the simulation.

Value

Returns an Rsim.scenario object that can be supplied to the rsim.run function.

See Also

Other Rsim functions: rsim.params(), rsim.plot(), rsim.run(), rsim.step()

---

rsim.sense	*Ecosense function for rpath (rsim.sense)*

Description

Kerim Aydin 31 December 2019

Usage

```
rsim.sense(Rsim.scenario, Rpath.params, Vvary = c(0, 0), Dvary = c(0, 0))
```

Arguments

Rsim.scenario	Rsim scenario object used as center of distributions (base model) during random parameter generation.
Rpath.params	Rpath parameter object (unbalanced model) used for data pedigree input.
Vvary	length-2 vector with (lower,upper) bounds of vulnerability generation in log-space - 1, Vvary = log(X-1) so Walters et al. 1997 range of (1..inf) centered on 2 becomes (-inf,+inf) centered on 0.
Dvary	length-2 vector with (lower,upper) bounds of handling time generation in log-space - 1, scaled as Vvary, above.

Value

Returns an Rsim.params object that can be substituted for the params in an Rsim.scenario object.

---

rsim.sense.path   *Ecosense function for rpath (rsim.sense.path)*

Description

5 November 2019

Usage

```
rsim.sense.path(Rsim.scenario, Rpath, Rpath.params, steps_yr = 12, steps_m = 1)
```

Value

Returns an Rsim.scenario object that can be supplied to the rsim.run function.

---

rsim.stanzas   *Generate Rsim stanza matrix*

Description

Creates a matrix of stanza variables to be used by `rsim.run`.

Usage

```
rsim.stanzas(Rpath.params, state, params)
```

Arguments

Rpath.params	R object containing the Rpath parameters. This is generated either by the create.rpath.params or read.rpath.params functions.
state	List object of state variables generated by `rsim.state`.
params	Rsim parameter object generated by `rsim.params`.

---

rsim.state   *Generate Rsim state matrix*

Description

Creates a matrix of state variables used by Rsim.

Usage

```
rsim.state(params)
```

Arguments

params   Rsim parameter object generated by `rsim.params`.

---

rsim.step   *Function to add a time step to an Rsim run*

Description

Runs `rsim.run` and appends the output to a previous `Rsim.output`. Currently only works with the Adams-Bashforth numerical integration method (AB).

Usage

```
rsim.step(Rsim.scenario, Rsim.output, method = "AB", year.end)
```

Arguments

Rsim.scenario	Scenario object that contains all of the rsim rates and forcing functions generated by `rsim.scenario`.
Rsim.output	R object containing the output from `rsim.run`.
method	Numerical integration method. Either *'AB'* for Adams-Bashforth or *'RK4'* for 4th order Runge-Kutta.
year.end	The final year for the step interval

Value

Returns an `Rsim.output` object.

See Also

Other Rsim functions: `rsim.params()`, `rsim.plot()`, `rsim.run()`, `rsim.scenario()`

---

set.rsim.scene   *Set Rsim.scenario parameters*

Description

Modifies the various parameters of the `rsim.scenario` object. Parameters that can be adjusted using this function are:

Usage

```
set.rsim.scene(
  Rsim.scenario,
  params = NULL,
  start_state = NULL,
  forcing = NULL,
  fishing = NULL,
  stanzas = NULL
)
```

Arguments

Rsim.scenario	Scenario object that contains all of the rsim rates and forcing functions generated by `rsim.scenario`.
params	Rsim parameter object generated by `rsim.params`.
start_state	Rsim starting values object generated by `rsim.state`
forcing	Rsim forcing matrix object generated by `rsim.forcing`
fishing	Rsim fishing matrix object generated by `rsim.fishing`
stanzas	Rsim stanza parameters object generated by `rsim.stanzas`

Value

Returns an `Rsim.scenario` object with the new parameter.

---

stanzaplot   *Plot routine for Ecopath multistanzas*

Description

Plots the biomass composition of multistanza groups from an Rpath.stanzas object.

Usage

```
stanzaplot(
  Rpath.params,
  StanzaGroup,
  line.cols = c("black", "green", "blue", "red")
)
```

Arguments

Rpath.params	R object containing the Rpath parameters. This is generated either by the create.rpath.params or read.rpath.params functions.
StanzaGroup	The Stanza group's name to be plotted.
line.cols	A vector of four colors used to represent the population biomass, relative number, indvidual weights, and stanza separation lines.

Value

Creates a figure showing the break down of biomass and number per stanza.

See Also

Other Rpath functions: `MTI()`, `adjust.forcing()`, `check.rpath.params()`, `create.rpath.params()`, `extract.node()`, `frate.table()`, `read.rpath.params()`, `rpath()`, `rpath.groups()`, `rpath.stanzas()`, `webplot()`, `write.Rpath()`, `write.Rsim()`, `write.rpath.params()`

---

webplot	*Plot routine for Ecopath food web*

Description

Plots the food web associated with an Rpath object.

Usage

```
webplot(
  Rpath.obj,
  eco.name = attr(Rpath.obj, "eco.name"),
  line.col = "grey",
  highlight = NULL,
  highlight.col = c("black", "red", "orange"),
  labels = FALSE,
  label.pos = NULL,
  label.num = FALSE,
  label.cex = 1,
  fleets = FALSE,
  type.col = "black",
  box.order = NULL
)
```

Arguments

`Rpath.obj`	Rpath model created by the ecopath() function.
`eco.name`	Optional name of the ecosystem. Default is the eco.name attribute from the rpath object.
`line.col`	The color of the lines between nodes of the food web.
`highlight`	Set to the group number or name to highlight the connections of that group.
`highlight.col`	Color of the connections to the highlighted group.
`labels`	Logical whether or not to display group names. If True and label.pos is Null, no points will be ploted, just label names.
`label.pos`	A position specifier for the labels. Values of 1, 2, 3, 4, respectively indicate positions below, to the left of, above, and to the right of the points. A null value will cause the labels to be ploted without the points (Assuming that labels = TRUE).
`label.num`	Logical value indication whether group numbers should be used for labels instead of names.
`label.cex`	The relative size of the labels within the plot.
`fleets`	Logical value indicating whether or not to include fishing fleets in the food web.
`type.col`	The color of the points cooresponding to the types of the group. Can either be of length 1 or 4. Color order will be living, primary producers, detrital, and fleet groups.
`box.order`	Vector of box numbers to change the default plot order. Must include all box numbers

Value

Creates a figure of the food web.

See Also

Other Rpath functions: `MTI()`, `adjust.forcing()`, `check.rpath.params()`, `create.rpath.params()`, `extract.node()`, `frate.table()`, `read.rpath.params()`, `rpath()`, `rpath.groups()`, `rpath.stanzas()`, `stanzaplot()`, `write.Rpath()`, `write.Rsim()`, `write.rpath.params()`

`write.Rpath`	*Write function for Ecopath object*

Description

Outputs basic parameters or mortalities to a .csv file.

Usage

```
write.Rpath(x, file = NA, morts = F)
```

Arguments

x	Rpath model created by the ecopath() function.
file	file name for resultant file. Need to specify ".csv" or ".RData".
morts	Logical value whether to output basic parameters or mortalities.

Value

Writes a .csv file with the basic parameters or mortalities from an Rpath object.

See Also

Other Rpath functions: `MTI()`, `adjust.forcing()`, `check.rpath.params()`, `create.rpath.params()`, `extract.node()`, `frate.table()`, `read.rpath.params()`, `rpath()`, `rpath.groups()`, `rpath.stanzas()`, `stanzaplot()`, `webplot()`, `write.Rsim()`, `write.rpath.params()`

---

write.rpath.params   *Write Rpath parameters to .csv files*

Description

Creates a series of .csv files from an Rpath.params object.

Usage

```
write.rpath.params(Rpath.params, eco.name, path = "")
```

Arguments

Rpath.params	R object containing the Rpath parameters. Most likely this was created using create.rpath.params or read.rpath.params.
eco.name	ecosystem name that will be included in all the file names.
path	location for the output files.

Value

Outputs a series of .csv files named by the provided eco.name and the parameters they represent. For example the model parameters will be named "eco.name_model.csv".

See Also

Other Rpath functions: `MTI()`, `adjust.forcing()`, `check.rpath.params()`, `create.rpath.params()`, `extract.node()`, `frate.table()`, `read.rpath.params()`, `rpath()`, `rpath.groups()`, `rpath.stanzas()`, `stanzaplot()`, `webplot()`, `write.Rpath()`, `write.Rsim()`

write.Rsim

Write function for Ecosim object

Description

Outputs start/end biomass and catch to a .csv file.

Usage

```
write.Rsim(Rsim.output, file = NA)
```

Arguments

Rsim.output	object created by the rsim.run() function.
file	file name for resultant .csv file. Be sure to include ".csv".

Value

Writes a .csv file with the start and end biomass and catch per group from an Rpath.sim object.

See Also

Other Rpath functions: `MTI()`, `adjust.forcing()`, `check.rpath.params()`, `create.rpath.params()`, `extract.node()`, `frate.table()`, `read.rpath.params()`, `rpath()`, `rpath.groups()`, `rpath.stanzas()`, `stanzaplot()`, `webplot()`, `write.Rpath()`, `write.rpath.params()`

Index