Package ‘osmose’

April 28, 2020

Type Package

Title Object Oriented Simulator of Marine Ecosystems

Version 3.3.4

Date 2020-04-03

Description The multispecies and individual-based model (IBM) 'OSMOSE' (Shin and Curry (2001) <doi:10.1016/S0090-7440(01)01106-8> and Shin and Curry (2004) <doi:10.1139/f03-154>) focuses on fish species. This model assumes opportunistic predation based on spatial co-occurrence and size adequacy between a predator and its prey (size-based opportunistic predation). It represents fish individuals grouped into schools, which are characterized by their size, weight, age, taxonomy and geographical location (2D model), and which undergo major processes of fish life cycle (growth, explicit predation, natural and starvation mortalities, reproduction and migration) and fishing exploitation. The model needs basic biological parameters that are often available for a wide range of species, and which can be found in 'FishBase' for instance (see <http://www.fishbase.org/search.php>), and fish spatial distribution data. This package provides tools to build and run simulations using the 'OSMOSE' model.

License CeCILL

Encoding UTF-8

Depends R (>= 3.5.0)

Imports graphics, grDevices, rlist, stats, knitr, rmarkdown, stringr, utils, ncdf4, mgcv, fields

URL http://www.osmose-model.org/

LazyData FALSE

BugReports https://github.com/osmose-model/osmose/issues

SystemRequirements Java (>= 8)

VignetteBuilder knitr

RoxygenNote 7.1.0

NeedsCompilation no
Author Yunne-Jai Shin [aut] (<https://orcid.org/0000-0002-7259-9265>),
Travers Morgane [aut] (<https://orcid.org/0000-0003-1493-662X>),
Verley Philippe [aut] (<https://orcid.org/0000-0002-5686-2764>),
Ricardo Oliveros-Ramos [aut] (<https://orcid.org/0000-0002-8069-2101>),
Laure Velez [aut],
Nicolas Barrier [cre] (<https://orcid.org/0000-0002-1693-4719>),
Criscey Lujan [ctb],
Michael Hurtado [ctb],
Wencheng Lau-Medrano [ctb]

Maintainer Nicolas Barrier <nicolas.barrier@ird.fr>

Repository CRAN

Date/Publication 2020-04-28 13:00:03 UTC

R topics documented:

- osmose-package .................................................. 3
- .osmose.barplot .................................................. 4
- .osmose.format_data_stacked ................................... 4
- .plot_osmose_dietMatrix ....................................... 5
- .read_2D ............................................................... 6
- .read_osmose_ncdf ................................................ 6
- cacheManager .................................................... 7
- configureCalibration .......................................... 7
- getVar .............................................................. 8
- get_var.osmose .................................................... 8
- get_var.osmose.config ......................................... 9
- osmose2R .......................................................... 10
- osmose_demo ...................................................... 11
- plot.osmose ....................................................... 12
- plot.osmose.config ............................................. 22
- print.osmose ...................................................... 24
- print.summary.osmose ......................................... 25
- readOsmoseConfiguration ...................................... 25
- readOsmoseFiles ................................................ 26
- report .............................................................. 26
- report.osmose ..................................................... 27
- runOsmose ........................................................ 27
- summary.osmose ................................................ 29
- summary.osmose.dietMatrix .................................. 29
- summary.osmose.mortalityRate ................................ 30
- updateCache ...................................................... 30
- write.osmose ..................................................... 31

Index 32
Description

OSMOSE is a multispecies and Individual-based model (IBM) which focuses on fish species. This model assumes opportunistic predation based on spatial co-occurrence and size adequacy between a predator and its prey (size-based opportunistic predation). It represents fish individuals grouped into schools, which are characterized by their size, weight, age, taxonomy and geographical location (2D model), and which undergo major processes of fish life cycle (growth, explicit predation, natural and starvation mortalities, reproduction and migration) and a fishing mortality distinct for each species (Shin and Cury 2001, 2004). The model needs basic parameters that are often available for a wide range of species, and which can be found in FishBase for instance. This package provides tools to build a model and run simulations using the OSMOSE model. See the Official website for more details.

Details

osmose package is well documented by help descriptions, demos and vignettes.

**Demo scripts:**

# Check all the available topics
demo(package = "osmose")

# Select and run one of the topics (e.g. osmose.config_class)
demo(package = "osmose", topic = "osmose.config_class")

**Vignettes:**

# Check all the available topics
vignette(package = "osmose")

# Select and run one of the topics (e.g. create_run_read)
vignette(package = "osmose", topic = "create_run_read")

Author(s)

Yunne-Jai Shin
Ricardo Oliveros-Ramos
Laure Velez
Criscely Luján
Philippe Verley
Maintainer: Nicolas Barrier <nicolas.barrier@ird.fr>

References

[Official website](#) [Documentation website](#)
### .osmose.barplot

*Plots a barplot, with xlabels rotated with a 45degree angle*

**Description**

Plots a barplot, with xlabels rotated with a 45degree angle.

**Usage**

```
.osmose.barplot(x, add_text = TRUE, color = NULL, ...)
```

**Arguments**

- **x**:
  Data array.
- **add_text**:
  True if text should be added.
- **color**:
  Color of the barplot.
- **...**:
  Additional arguments to the barplot function (color, etc.).

### .osmose.format_data_stacked

*Reformat the data into the ggplot2 stacked plot format. It returns a dataframe with specie, time and predation rate in the columns.*

**Description**

Reformat the data into the ggplot2 stacked plot format. It returns a dataframe with specie, time and predation rate in the columns.

**Usage**

```
.osmose.format_data_stacked(data, time = NULL)
```

**Arguments**

- **data**:
  Data to plot.
- **time**:
  Time vector.
Description

Plots diet matrix

Usage

.plot_osmose_dietMatrix(
  x,
  time.mean,
  species,
  thres,
  color,
  add_text,
  plot_name,
  legsize = 1,
  ...
)

Arguments

x        Diet matrix
time.mean If TRUE, the time mean diet matrix is computed is displayed.
species  Species name
thres    Thresholds (in percentage). Time-average predation rates below this threshold
          are binned together ("other" column).
color    Color of the barplot.
add_text TRUE if text should be added.
plot_name Label for ylab in diet matrix plot.
legsize  Size of the legend (default 1), if time.mean is FALSE.
...      Additional plot arguments passed to barplot.

Value

None
.read_2D

Generic function to read 2D output files (dietMatrix, sizeSpectrum, etc). If no file is found, it returns NULL.

Description

Generic function to read 2D output files (dietMatrix, sizeSpectrum, etc). If no file is found, it returns NULL.

Usage

.read_2D(files, path, ...)

Arguments

files List of input files
path Data directory
... Extra arguments passed to the method.

Value

A 3D array (time, species, replicates) or NULL if no file is found.

---

.read_osmose_ncdf

Function to read osmose netcdf files

Description

Function to read osmose netcdf files

Usage

.read_osmose_ncdf(files, path, ...)

Arguments

files String of name of the file that will be read.
path String of path of the file that will be read
... Extra arguments
cacheManager

Manage the cache

Description
The function creates a folder (only once) and then controls files that need to be downloaded and placed in this folder.

Usage
cacheManager(nameFile)

Arguments
nameFile: the name of the file or folder

Value
the path to the file we need

Localisation
The localisation of the folder is defined in the '.Renviron' file, by setting the 'OSMOSE_DIR' environment variable:
- On Linux/Mac Os X: 'OSMOSE_DIR=/Users/Nicolas/Desktop/OSMOSE_TEST/R'
- On Windows: 'OSMOSE_DIR=C:\Users\Nicolas\Desktop\OSMOSE_TEST\R'
If this variable is not set, the files will be downloaded into a temporary directory.

Author(s)
Arthur PERE
Nicolas BARRIER

configureCalibration
Reads calibration parameters from an osmose.config list.

Description
The configuration argument must contain a "calibration" entry to work.

Usage
configureCalibration(L1)

Arguments
L1: osmose.config object (see readOsmoseConfiguration)
Value

A list of parameters to calibrate ("guess", "max", "min", "phase")

getVar

Get variable from an osmose-like object.

Description

Function to get a variable from an object of osmose class. This function uses the get_var method (see the get_var.osmose).

Usage

getVar(object, what, how, ...)

get_var(object, what, how, ...)

Arguments

object Object of osmose class (see the read_osmose function).
what Variable to extract
how Output format
... Additional arguments of the function.

Value

An array or a list containing the extracted data.

get_var.osmose

get_var method for osmose outputs objects

Description

Get a variable from an osmose object.

Usage

## S3 method for class 'osmose'
get_var(object, what, how = c("matrix", "list"), expected = FALSE, ...)
**Arguments**

- **object**: Object of osmose class (see the `read_osmose` function).
- **what**: Name of variable to extract. See Details.
- **how**: How to return the object. Current options are "matrix" and "list".
- **expected**: A logical parameter. If TRUE, the average over the last dimensions will be performed (only if the output is an array).
- **...**: Additional arguments of the function.

**Details**

what can be any available variable contained on object (e.g. biomass, abundance, yield, yieldN, etc).

**Value**

An matrix or a list containing the data.

---

**get_var.osmose.config**  
*get_var method for osmose configuration objects*

**Description**

Get the configuration files from

**Usage**

```r
## S3 method for class 'osmose.config'
get_var(object, what, ...)
```

**Arguments**

- **object**: Object of osmose.config class. See the `read_osmose` and `readOsmoseConfiguration` functions for more information about this object.
- **what**: Name of the variable to extract from the configuration file.
- **...**: Extra arguments for plotting method.

**Value**

An object of list class containing all the relevant information about the variable extracted.
osmose2R  

Read OSMOSE outputs into an R object

Description
This function creates an object of class `osmose` with the outputs from OSMOSE in the path folder.

Usage

```r
osmose2R(path = NULL, version = "v3r2", species.names = NULL, ...)

read_osmose(
  path = NULL,
  input = NULL,
  version = "3.3.3",
  species.names = NULL,
  absolute = TRUE,
  ...
)
```

Arguments

- `path`  
  Path to the directory containing OSMOSE outputs.

- `version`  
  OSMOSE version used to run the model.

- `species.names`  
  Display names for species, overwrite the species names provided to the OSMOSE model. Used for plots and summaries.

- `...`  
  Additional arguments.

- `input`  
  Path to a main OSMOSE configuration file.

- `absolute`  
  Whether the path is absolute (TRUE) or relative (FALSE). Only used if input is not NULL.

Details

`read_osmose` will return a list of fields with the information of whether an OSMOSE running or the configuration that is going to be used in a running. Output class will depend on the read info: If `path` is specified, output class will be `osmose`; otherwise, if ONLY `input` is given, the class will be `osmose.config`. If both are specified, the `osmose` class output will contain inside a field (`config`) of class `osmose.config`.

Individual elements can be extracted using the `get_var` function.

Author(s)
Ricardo Oliveros-Ramos, Laure Velez
osmose_demo

Examples

# Read outputs generated by demo (check osmose_demo function)
outdir = cacheManager("outputs")
read_osmose(path = outdir)

---

osmose_demo

Generates required OSMOSE configuration files to run a demo.

Description

This function

Usage

osmose_demo(path = NULL, config = "gog")

Arguments

- **path**: Path where to put the Osmose configuration file.
- **config**: Reference configuration to run ("gog").

Value

A list containing the configuration file to use (config_file) for running the code and the output directory to use when reading data.

Note

So far, only one configuration is propose ("gog")

Examples

```r
## Not run:
rm(list=ls())
library("osmose")

copy configuration files into the proper directory
demo = osmose_demo(path="../", config="gog")
un the osmose model
run_osmose(demo$config_file, parameters=NULL, output=NULL, version="3.3.3",
               options=NULL, verbose=TRUE, clean=TRUE)

# reads output data
data = read_osmose(demo$output_dir)

# summarize output data
```
plot.osmose

Plot method for osmose objects

Description

This method takes an osmose object and produces useful plots.

This function implements a plot method for different osmose classes.

Usage

```r
## S3 method for class 'osmose'
plot(x, what = "biomass", ...)

## S3 method for class 'osmose.biomass'
plot(
x,
ts = TRUE,
type = 1,
species = NULL,
speciesNames = NULL,
start = NULL,
end = NULL,
initialYear = NULL,
replicates = TRUE,
freq = 12,
horizontal = FALSE,
conf = 0.95,
factor = 0.001,
xlim = NULL,
ylim = NULL,
col = NULL,
alpha = NULL,
border = NULL,
lt = 1,
lwd = 1,
axes = TRUE,
legend = TRUE,
units = "tonnes",
...
)
```
## S3 method for class 'osmose.abundance'
plot(
  x,
  species = NULL,
  speciesNames = NULL,
  start = NULL,
  end = NULL,
  initialYear = NULL,
  ts = TRUE,
  type = 1,
  replicates = TRUE,
  freq = 12,
  horizontal = FALSE,
  conf = 0.95,
  factor = 0.001,
  xlim = NULL,
  ylim = NULL,
  col = NULL,
  alpha = NULL,
  border = NULL,
  lty = 1,
  lwd = 1,
  axes = TRUE,
  legend = TRUE,
  units = "individuals",
  ...
)

## S3 method for class 'osmose.yield'
plot(
  x,
  species = NULL,
  speciesNames = NULL,
  start = NULL,
  end = NULL,
  initialYear = NULL,
  ts = TRUE,
  type = 1,
  replicates = TRUE,
  freq = 12,
  horizontal = FALSE,
  conf = 0.95,
  factor = 0.001,
  xlim = NULL,
  ylim = NULL,
  col = NULL,
  alpha = NULL,
plot.osmose

border = NULL,
lty = 1,
lwd = 1,
axes = TRUE,
legend = TRUE,
units = "tonnes",
...
)

## S3 method for class 'osmose.yieldN'
plot(
  x,
  species = NULL,
speciesNames = NULL,
start = NULL,
end = NULL,
initialYear = NULL,
ts = TRUE,
type = 1,
replicates = TRUE,
freq = 12,
horizontal = FALSE,
conf = 0.95,
factor = 0.001,
xlim = NULL,
ylim = NULL,
col = NULL,
alpha = NULL,
border = NULL,
lty = 1,
lwd = 1,
axes = TRUE,
legend = TRUE,
units = "individuals",
...
)

## S3 method for class 'osmose.biomassByTL'
plot(
  x,
  type = 1,
  species = NULL,
speciesNames = NULL,
start = NULL,
end = NULL,
freq = 12,
horizontal = FALSE,
factor = 0.001,
xlim = NULL,
ylim = NULL,
col = NULL,
border = NULL,
legend = TRUE,
lty = 1,
lwd = 1,
axes = TRUE,
units = list(x = "", y = "tonnes"),
...
)

## S3 method for class 'osmose.biomassBySize'
plot(
  x,
  type = 1,
  species = NULL,
  speciesNames = NULL,
  start = NULL,
  end = NULL,
  freq = 12,
  horizontal = FALSE,
  factor = 0.001,
  xlim = NULL,
  ylim = NULL,
  col = NULL,
  border = NULL,
  legend = TRUE,
  lty = 1,
  lwd = 1,
  axes = TRUE,
  units = list(x = "cm", y = "tonnes"),
  ...
)

## S3 method for class 'osmose.biomassByAge'
plot(
  x,
  type = 1,
  species = NULL,
  speciesNames = NULL,
  start = NULL,
  end = NULL,
  freq = 12,
  horizontal = FALSE,
  factor = 0.001,
  xlim = NULL,
  ylim = NULL,
col = NULL,
border = NULL,
legend = TRUE,
lty = 1,
lwd = 1,
axes = TRUE,
units = list(x = "years", y = "tonnes"),
...
)

## S3 method for class 'osmose.abundanceByTL'
plot(
x,
type = 1,
species = NULL,
speciesNames = NULL,
start = NULL,
end = NULL,
freq = 12,
horizontal = FALSE,
factor = 0.001,
xlim = NULL,
ylim = NULL,
col = NULL,
border = NULL,
legend = TRUE,
lty = 1,
lwd = 1,
axes = TRUE,
units = list(x = ",", y = "individuals"),
...
)

## S3 method for class 'osmose.abundanceBySize'
plot(
x,
type = 1,
species = NULL,
speciesNames = NULL,
start = NULL,
end = NULL,
freq = 12,
horizontal = FALSE,
factor = 0.001,
xlim = NULL,
ylim = NULL,
col = NULL,
border = NULL,
plot.osmose

legend = TRUE,
lty = 1,
lwd = 1,
axes = TRUE,
units = list(x = "cm", y = "individuals"),
...
)

## S3 method for class 'osmose.abundanceByAge'
plot(
x,
type = 1,
species = NULL,
speciesNames = NULL,
start = NULL,
end = NULL,
freq = 12,
horizontal = FALSE,
factor = 0.001,
xlim = NULL,
ylim = NULL,
col = NULL,
border = NULL,
legend = TRUE,
lty = 1,
lwd = 1,
axes = TRUE,
units = list(x = "years", y = "individuals"),
...
)

## S3 method for class 'osmose.yieldNBySize'
plot(
x,
type = 1,
species = NULL,
speciesNames = NULL,
start = NULL,
end = NULL,
freq = 12,
horizontal = FALSE,
factor = 0.001,
xlim = NULL,
ylim = NULL,
col = NULL,
border = NULL,
legend = TRUE,
lty = 1,
lwd = 1,
axes = TRUE,
units = list(x = "cm", y = "individuals"),
...
)

### S3 method for class 'osmose.yieldNByAge'
plot(
x,
type = 1,
species = NULL,
speciesNames = NULL,
start = NULL,
end = NULL,
freq = 12,
horizontal = FALSE,
factor = 0.001,
xlim = NULL,
ylim = NULL,
col = NULL,
border = NULL,
legend = TRUE,
lty = 1,
lwd = 1,
axes = TRUE,
units = list(x = "years", y = "individuals"),
...
)

### S3 method for class 'osmose.yieldBySize'
plot(
x,
type = 1,
species = NULL,
speciesNames = NULL,
start = NULL,
end = NULL,
freq = 12,
horizontal = FALSE,
factor = 0.001,
xlim = NULL,
ylim = NULL,
col = NULL,
border = NULL,
legend = TRUE,
lty = 1,
lwd = 1,
axes = TRUE,
plot.osmose

```r
units = list(x = "cm", y = "tonnes"),
...
)

## S3 method for class 'osmose.yieldByAge'
plot(
x,
type = 1,
species = NULL,
speciesNames = NULL,
start = NULL,
end = NULL,
freq = 12,
horizontal = FALSE,
factor = 0.001,
xlim = NULL,
ylim = NULL,
col = NULL,
border = NULL,
legend = TRUE,
lty = 1,
lwd = 1,
axes = TRUE,
units = list(x = "years", y = "tonnes"),
...
)

Arguments

x osmose object.
what Variable name to plot. By default is what = "biomass". See Details
...
Extra arguments of the function. See Details.
ts logical parameter. By default ts = TRUE and the plots are going to be time series where the x-axis is the time. For plots where the x-axis is not the time ts = FALSE.
type A numeric value, indicating the type of plot to be used (type = 1, by default). See Details.
species A numeric vector specifying the species that will be showed. If NULL (default), all the available species time series will be plotted. See Details.
speciesNames A vector with the names for each species. If NULL (default) the names of x will be used.
start A numeric value indicating the first element of the time indexation of the object x is specified with this parameter. By default start = NULL and the time indexation start with the first element of x.
end A numeric value. The last element of the time indexation of the object x is specified with this parameter. By default end = NULL and the time indexation finish with the last element of x.
```
initialYear A numeric value. It specifies the first element that is going to be used on the x axis for the plots.
replicates logical. It controls to show whether the values in each simulation (TRUE) or the median of the values along the time (FALSE).
freq A numeric value to indicate the steps by year used in the time series. Default values, see Details.
horizontal a logical value. If FALSE, plot types that do not represent time series (boxplots and barplots) will be drawn horizontally.
conf numeric single value indicating the confidence interval showed in the plot as shadow. By default conf = 0.95.
factor A number to indicate the scale of the variable on the y-axis. By default factor = 1e-3.
xlim, ylim numeric vectors of length 2, giving the x and y coordinates ranges.
col A vector with the color names for the plots. By default col = NULL and the colors will be chosen by the function.
alpha factor modifying the opacity alpha given to confidence interval (CI) polygons (check adjustcolor). By default (NULL), its value will depend on the plot type. See Details.
border the color to draw the border of CI polygons, bar plots and box plots. The default value (NULL) will depend on the type, see Details.
lty The line type (check par).
lwd The line width (check par).
axes a logical value indicating whether both axes should be drawn on the plot.
legend logical do you want to show a legend of species? (valid only for those plot types TS-2)
units String with units of the variable that defines the label set on top of the plot. If NULL (default), the value will depend on the class, see Details.

Details

what argument can take next values:

- "biomass" to plot the species biomass (see plot.osmose.biomass)
- "abundance" to plot the species abundance (see plot.osmose.abundance)
- "yield" to plot the species yield (see plot.osmose.yield)
- "yieldN" to plot the species yield (in numbers) (see plot.osmose.yieldN)

User can select within different plot types depending on the osmose class. So, for classes biomass, abundance, yield and yieldN (Group 1) the user must use ts and type arguments to specify the type of plot to draw.

So, for the Group 1 and ts = TRUE, user will select within:

- type = 1: Generates a matrix plot of the selected variable plotted independently.
• type = 2: Generates a single plot with overlaped time series. You can include confidence interval shadow around lines.
• type = 3: Generates a single plot with cumulated time series. So, the method will sort and overlap the time series values for the selected variable
• type = 4: ONLY valid for a single species. It generates a bar plot of the time series.

For the Group 1 and ts = FALSE, user will select within:
• type = 1: Generates a bar plot of the selected variable by species, including interval confidence bars.
• type = 2: Generates a boxplot of the selected variable by species.

The Group 2 refers classes of Group 1 with By specification (e.g. yieldNBySize, abundanceByAge, biomassByTL). There are 2 available plot types:
• type = 1: Generates a bar plot of the selected variable by species. The ... will be passed to the internal barplot function.
• type = 2: Generates a plot using image.plot function, placing the species in the left axis, the By-variable (e.g. Size) on the bottom and the main variable (e.g. biomass) in a color scale. The ... will be passed to the internal image.plot function.

Both alpha and border controls the opacity and the color of border, respectively, for polygons, bar plots and box plots. So depending on the ts-type, they will have different default values:
• alpha = 0.3 & border = NA for TS = TRUE with type = 1 or 2.
• alpha = 1 & border = TRUE for Everything else.

species argument follows the indexation way of java: starting in zero, as osmose-java returns species outputs.

Default value for freq will be calculated from x: \( freq = \frac{1}{x \cdot model \cdot start} \).

units could be whether a single character vector (currently applied on classes without By, e.g. osmose.biomass, osmose.yieldN) or a list of length 2 specifying the units for x and y (currently used with classes with By, e.g. plot.osmose.biomassBySize, plot.osmose.yieldNByAge).

Extra arguments can be passed from plot.default using ...: cex, cex.axis, border (useful for polygon, boxplots and barplots), etc. It is important to notice that, depending on where the ellipsys is passed, it allows and refuse the use of certain arguments. For instance, if the selected plot is a bar plot, the ... will be passed to the internal barplot call, so if cex argument is used, it will create a conflict and an error (a typical error of using cex in barplot function).

Value
A graph of an osmose object.

Author(s)
Criscely Lujan Paredes
### plot.osmose.config

#### Plot method for osmose.config objects

#### Description

This method takes a `osmose.config` object and produce useful plots.

This function implements a plot method for different `osmose` classes.

#### Usage

```R
## S3 method for class 'osmose.config'
plot(x, what = "predation", ...)

## S3 method for class 'osmose.config.reproduction'
plot(
  x,
  type = 1,
  species = 0,
  speciesNames = NULL,
  start = NULL,
  end = NULL,
  initialYear = NULL,
  freq = 12,
  xlim = NULL,
  ylim = NULL,
  col = "black",
  axes = TRUE,
  border = NA,
  legend = TRUE,
  ...
)

## S3 method for class 'osmose.config.species'
plot(
  x,
  n = 100,
  type = 1,
  species = 0,
  speciesNames = NULL,
  addElements = c("segments", "points", "polygon", "text"),
  axes = TRUE,
  border = NA,
  xlim = NULL,
  ylim = NULL,
  legend = TRUE,
  col = "black",
  ```
## S3 method for class 'osmose.config.predation'
plot(  
  x,  
  type = 1,  
  species = 0,  
  speciesNames = NULL,  
  addElements = c("segments", "points", "text"),  
  axes = TRUE,  
  border = NA,  
  xlim = NULL,  
  ylim = NULL,  
  col = "gray70",  
  legend = TRUE,  
  ...  
)

Arguments

- **x** osmose like object.
- **what** Variable name to plot. By default is what = "predation". See Details.
- **...** Extra arguments of the function.
- **type** A numeric value, indicating the type of plot to be used (type = 1, by default). See Details.
- **species** A numeric vector specifying the species that will be showed. If NULL (default), all the available species time series will be plotted. See Details.
- **speciesNames** A vector with the names for each species. If NULL (default) the names of x will be used.
- **start** A numeric value indicating the first element of the time indexation of the object x is specified with this parameter. By default start = NULL and the time indexation start with the first element of x.
- **end** A numeric value. The last element of the time indexation of the object x is specified with this parameter. By default end = NULL and the time indexation finish with the last element of x.
- **initialYear** A numeric value. It specifies the first element that is going to be used on the x axis for the plots.
- **freq** A numeric value to indicate the steps by year used in the time series. Default values, see Details.
- **xlim, ylim** numeric vectors of length 2, giving the x and y coordinates ranges.
- **col** A vector with the color names for the plots. By default col = NULL and the colors will be chosen by the function.
- **axes** a logical value indicating whether both axes should be drawn on the plot.
print.osmose

border the color to draw the border of CI polygons, bar plots and box plots. By default, border = NA, which means that no border will be drawn.

legend logical do you want to show a legend of species? (valid only for those plot types TS-2)

n numeric value indicating the number of steps that are going to be used to plot the growth curve (n = 100 as default). The larger is n, the more resolution the curve will have.

addElements A character vector indicating extra graphical elements that can be included.

Details

Plot types will depend on the class of x, which is defined by what. Thereby,

• what = predation: Generates a plot of size range as shadows for a selected species.
• what = reproduction: Generates a single plots of seasonality of reproduction whether as lines (type = 1) or bars (type = 2)
• what = species: Generates a plot of growth curve following the VB parameters defined on configuration files (type = 1).

species argument follows the indexation way of java: starting in zero, as osmose-java returns species outputs.

Default value for freq will be calculated from x: freq = 1/xmodelstart.

Extra arguments can be passed using ... and depending on the type, you can modify arguments like: cex, cex.axis, border (useful for polygon and barplots), etc.

Note

Ellipsis (...) must be used carefully, since it will pass the arguments to different generic plot functions. For instance, type = 2 of osmose.config.reproduction method will use ... to pass arguments to barplot, so some arguments like cex may match with many formal arguments (e.g. cex.axis and cex.names), so it may cause errors.

Author(s)

Criscely Lujan Paredes

---

**print.osmose**

Print information for an osmose object

Description

Print information for an osmose object

Usage

```r
# S3 method for class 'osmose'
print(x, ...)
```
print.summary.osmose

Arguments

- x: Osmose class object (see the `read_osmose` function).
- ...: Additional arguments for `print`.

Description

Print the summary informations about Osmose outputs.

Usage

```r
## S3 method for class 'summary.osmose'
print(x, ...)
```

Arguments

- x: Osmose outputs (see the `read_osmose` function).
- ...: Additional arguments for `print`.

readOsmoseConfiguration

Reads Osmose configuration files.

Description

Reads Osmose configuration files.

Usage

```r
readOsmoseConfiguration(file, config = NULL, absolute = TRUE)
```

Arguments

- file: Main configuration file
- config: Configuration object to which file parameters are appended
- absolute: Whether the path is absolute (TRUE) or relative (FALSE)

Value

A list tree.
readOsmoseFiles  Read Osmose output file

Description
Read Osmose output file

Usage
readOsmoseFiles(path, type, bySpecies = FALSE, ext = "csv", ...)

Arguments
- **path**: Osmose output path
- **type**: Data type ("biomass", etc)
- **bySpecies**: TRUE if should read one file per species.
- **ext**: The extension of the files which will be read.
- **...**: Additional arguments

Value
Output data frame

report  Report method

Description
This function built a report for each class including on osmose package.

Usage
report(x, format, output, ...)

Arguments
- **x**: Object of class osmose.
- **format**: The format to export the report.
- **output**: Folder where the report will be saved.
- **...**: Extra arguments passed to report function.

Value
A report on specific format.
Description

Build and export a report of osmose objects using R markdown.

Usage

```r
## S3 method for class 'osmose'
report(
  x,
  format = "pdf_document",
  output = NULL,
  tangle = FALSE,
  open = TRUE,
  ...
)
```

Arguments

- `x` Object of osmose class.
- `format` The R Markdown output format to convert to (check `render`).
- `output` The output directory for the rendered output file.
- `tangle` Boolean; whether to tangle the R code from the input file (check `render`).
- `open` Do you want to open the output file at the end? (only useful for MS Windows environment).
- `...` Extra arguments passed to `render`.

Description

This function create a valid configuration by several input files from user input parameters.

Usage

```r
runOsmose(
  input,
  parameters = NULL,
  output = "output",
  log = "osmose.log",
  version = "4.1.0",
```
runOsmose

```r
osmose = NULL,
java = "java",
options = NULL,
verbose = TRUE,
clean = TRUE
)
run_osmose(
  input,
  parameters = NULL,
  output = NULL,
  log = "osmose.log",
  version = "3.3.3",
  osmose = NULL,
  java = "java",
  options = NULL,
  verbose = TRUE,
  clean = TRUE
)
```

**Arguments**

- **input**  
  Filename of the main configuration file

- **parameters**  
  Parameters to be passed to osmose (version 4 or higher).

- **output**  
  Output directory. If NULL, the value set in the configuration file is used.

- **log**  
  File to save OSMOSE execution messages.

- **version**  
  OSMOSE version. Integer (2, 3, etc.) or releases ('v3r2') are accepted.

- **osmose**  
  Path to a OSMOSE .jar executable. By default (NULL), uses the stable jar for the current version.

- **java**  
  Path to the java executable. The default assumes 'java' is on the search path.

- **options**  
  Java options (e.g. -Xmx2048m to increase memory limit).

- **verbose**  
  Show messages? (output in the log file if FALSE).

- **clean**  
  TRUE if the output directory should be cleaned before running OSMOSE.

**Details**

Basic configurations may not need the use of buildConfiguration, but it is required for configuration using interannual inputs or fishing selectivity.

**Author(s)**

Ricardo Oliveros-Ramos
Examples

```r
{  
    ## Not run:  
    path = cacheManager("gog")  
    filename = file.path(path, "osm_all-parameters.csv")  
    run_osmose(filename)  
    
    ## End(Not run)
}
```

summary.osmose  

osmose object summaries

Description

osmose object summaries

Usage

```r
## S3 method for class 'osmose'
summary(object, ..., digits = 1L)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>osmose class object (see the read_osmose function).</td>
</tr>
<tr>
<td>...</td>
<td>Additional arguments for summary.</td>
</tr>
<tr>
<td>digits</td>
<td>integer, used for number formatting (by default, 1L). Check summary.default.</td>
</tr>
</tbody>
</table>

summary.osmose.dietMatrix

Title

Description

Title

Usage

```r
## S3 method for class 'osmose.dietMatrix'
summary(object, species = NULL, thres = 1, ...)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>an object of class osmose.mortalityRate for which a summary is desired.</td>
</tr>
<tr>
<td>species</td>
<td>Name of the species to get a summary.</td>
</tr>
<tr>
<td>thres</td>
<td>Threshold which is used to keep values of species matrix.</td>
</tr>
<tr>
<td>...</td>
<td>Extra arguments passed to the method.</td>
</tr>
</tbody>
</table>
**updateCache**

---

**summary.osmose.mortalityRate**

*Title*

**Description**

Title

**Usage**

```r
## S3 method for class 'osmose.mortalityRate'
summary(object, species = NULL, ...)
```

**Arguments**

- `object`: an object of class `osmose.mortalityRate` for which a summary is desired.
- `species`: Name of the species to get a summary.
- `...`: Extra arguments passed to the method.

---

**updateCache**

*Update the cache for the different function*

**Description**

This function updates the cache for the environmental variables: - wc2-5 - CWD - E

**Usage**

`updateCache(nameFile)`

**Arguments**

- `nameFile`: The name of the file you want to update. If it’s ‘NULL’ the function will update all the files.

**Author(s)**

Arthur PERE

**Examples**

```r
## Not run:
updateCache()
```

```r
## End(Not run)
```
Write an array or dataframe in the Osmose format. The separator is ";", there are no quotes and a blank column is added for the row names column.

```r
write.osmose(x, file)
write_osmose(
  x,
  file,
  sep = ",",
  col.names = NA,
  quote = FALSE,
  row.names = TRUE,
  ...
)
```

**Arguments**

- `x`: Object to be written (table or data frame)
- `file`: Output file
- `sep`: The field separator string. Values within each row of `x` are separated by this string.
- `col.names`: either a logical value indicating whether the column names of `x` are to be written along with `x`, or a character vector of column names to be written. See the section on ‘CSV files’ for the meaning of `col.names = NA`.
- `quote`: A logical value (`TRUE` or `FALSE`) or a numeric vector.
- `row.names`: either a logical value indicating whether the row names of `x` are to be written along with `x`, or a character vector of row names to be written.
- `...`: Extra arguments passed to `write.table` function.
Index

*Topic **Internal**
  cacheManager, 7
*Topic **ecosystems**
  osmose-package, 3
*Topic **marine**
  osmose-package, 3
*Topic **modelling**
  osmose-package, 3
  .osmose.barplot, 4
  .osmose.format_data_stacked, 4
  .plot_osmose_dietMatrix, 5
  .read_2D, 6
  .read_osmose_ncdf, 6
adjustcolor, 20
barplot, 21, 24
boxplot, 21

cacheManager, 7
configureCalibration, 7

get_var, 10
get_var(getVar), 8
get_var.osmose, 8, 8
get_var.osmose.config, 9
getVar, 8

image.plot, 21

osmose (osmose-package), 3
osmose-package, 3
osmose2R, 10
osmose_demo, 11

par, 20
plot.default, 21
plot.osmose, 12
plot.osmose.abundance, 20
plot.osmose.biomass, 20
plot.osmose.config, 22
plot.osmose.yield, 20
plot.osmose.yieldN, 20
polygon, 21
print.osmose, 24
print.summary.osmose, 25
read_osmose, 8, 9, 25, 29
read_osmose(osmose2R), 10
readOsmoseConfiguration, 7, 9, 25
readOsmoseFiles, 26
render, 27
report, 26, 26
report.osmose, 27
run_osmose(runOsmose), 27
runOsmose, 27
summary.default, 29
summary.osmose, 29
summary.osmose.dietMatrix, 29
summary.osmose.mortalityRate, 30
updateCache, 30
write.osmose, 31
write_osmose (write.osmose), 31