

Package ‘osmose’

January 9, 2018

Type Package

Title Object Oriented Simulator of Marine Ecosystems

Version 0.1.1

Date 2017-12-06

Description The multispecies and individual-based model (IBM) 'OSMOSE' (Shin and Curry (2001) <doi:10.1016/S0990-7440(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 (>= 2.15)

Imports graphics, grDevices, rlist, stats, stringr, utils

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

LazyData FALSE

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

SystemRequirements Java (>= 8)

RoxygenNote 6.0.1

NeedsCompilation no

Author Yunne-Jai Shin [aut],
Travers Morgane [aut],
Verley Philippe [aut],

Ricardo Oliveros-Ramos [aut],
 Laure Velez [aut],
 Nicolas Barrier [cre],
 Criscely Lujan [ctb],
 Michael Hurtado [ctb]

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

Repository CRAN

Date/Publication 2018-01-09 17:29:12 UTC

R topics documented:

osmose-package	3
configureCalibration	3
getAverageMortality	4
getFishingBaseRate	4
getFishingBaseRate.byDt	5
getFishingBaseRate.byYear	6
getFishingBaseRate.constant	6
getFishingBaseRate.linear	7
getFishingMortality	8
getMortality	9
getMortalityDeviation	9
getOsmoseParameter	10
getSizeSpectrum	11
getVar	11
getVar.osmose	12
plot.osmose	12
print.osmose	14
print.summary.osmose	14
readOsmoseConfiguration	15
readOsmoseFiles	15
read_osmose	16
runOsmose	17
summary.osmose	18
write.osmose	18
writeFishingFiles	19

Index	20
--------------	-----------

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 <http://www.osmose-model.org/> for more details.

Author(s)

Yunne-Jai Shin

Ricardo Oliveros-Ramos

Laure Velez

Criscely Luján

Philippe Verley

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

References

osmose: Modelling Marine Exploited Ecosystems

`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")

getAverageMortality *Computes the average mortality.*

Description

It computes the mean mortality, which is multiplied by the frequency.

Usage

```
getAverageMortality(x, stage = "adults", freq = 12)
```

Arguments

x	Mortality dataframe
stage	Stage ("adults", etc.)
freq	Time frequency (months?)

Value

An array

Examples

```
{
  dirin = system.file("extdata", package="osmose")
  outdir = paste(dirin, "/outputs", sep="")
  data = read_osmose(outdir)
  mortality_df = data$global$mortality
  mort = getAverageMortality(mortality_df, stage="juveniles", freq=12)
}
```

getFishingBaseRate *Get fishing base rate.*

Description

Get fishing base rate.

Usage

```
getFishingBaseRate(sp, fishing, T, ndt)
```

Arguments

sp	Current specie (sp0, sp1, etc.)
fishing	Fishing parameters
T	Number of years
ndt	Time step

Details

It assumes that in the Osmose configuration, there is a "fishing" entry. It first check for the "method" parameter for the current specie (*mortality.fishing.rate.method.spX*), whose value can be

- constant: use *rate.spX*
- byRegime: use *rate.byRegime.file.spX*
- linear : use *rate.slope.spX*
- byYear : use *rate.byYear.spX*
- byDt : use *rate.byDt.spX*

Examples

```
{
filename = system.file("extdata", "data_fishing.csv", package="osmose")
conf = readOsmoseConfiguration(filename)
ndt = getOsmoseParameter(conf, "simulation", "time", "ndtperyear")
nyear = getOsmoseParameter(conf, "simulation", "time", "nyear")

fishing = conf$mortality$fishing

fish0 = getFishingBaseRate("sp0", fishing, nyear, ndt)
}
```

```
getFishingBaseRate.byDt
```

Get fishing base rate using time-step rates.

Description

It reads the time-step values either from an external file or from the configuration file. It must contain a number of elements which is a multiple of the fishing period.

Warning: The number of rate elements must be a multiple of the fishing period since **recycling** is performed!)

Usage

```
getFishingBaseRate.byDt(sp, fishing, T, ndt)
```

Arguments

sp	Current specie (sp0, sp1, etc.)
fishing	Fishing parameters
T	Number of years
ndt	Time step

```
getFishingBaseRate.byYear
```

Get fishing base rate using annual rates.

Description

It reads the annual values either from an external file or from the configuration file. It must contain a number of elements which is a multiple of the fishing period.

Warning: The number of rate elements must be a multiple of the fishing period since **recycling** is performed!)

Usage

```
getFishingBaseRate.byYear(sp, fishing, T, ndt)
```

Arguments

sp	Current specie (sp0, sp1, etc.)
fishing	Fishing parameters
T	Number of years
ndt	Time step

```
getFishingBaseRate.constant
```

Get fishing base rate using constant rate

Description

Get fishing base rate using constant rate

Usage

```
getFishingBaseRate.constant(sp, fishing, T, ndt)
```

Arguments

sp	Current specie (sp0, sp1, etc.)
fishing	Fishing parameters
T	Number of years
ndt	Time step

Details

It assumes that in the Osmose configuration, there is a "fishing" entry. It reads the *rate.spX* parameter, which is repeated $T \times ndt$ times.

`getFishingBaseRate.linear`

Get fishing base rate using a linear rate

Description

Get fishing base rate using a linear rate

Usage

`getFishingBaseRate.linear(sp, fishing, T, ndt)`

Arguments

sp	Current specie (sp0, sp1, etc.)
fishing	Fishing parameters
T	Number of years
ndt	Time step

Details

It assumes that in the Osmose configuration, there is a "fishing" entry. It reads the *rate.spX* and *rate.slope.spX* parameters. The slope is computed any time the fishing frequency changes (fishing-period parameter) For instance, if $ndt=20$, $period=5$, there are 4 time steps between two fishing time step. Therefore, the value will be (i0, i0, i0, i0, i1, i1, i1, i1, ...)

getFishingMortality *Get fishing mortality rate.*

Description

Get fishing mortality rate.

Usage

```
getFishingMortality(sp, fishing, T, ndt)
```

Arguments

sp	Species index (sp0, sp1, sp2, etc.)
fishing	List of fishing parameters
T	Number of years
ndt	Time step

Details

$$F = \frac{B \times \exp^{A+S}}{ndt}$$

with B =mortality rate, A =deviates by year, S =deviates by seasons

Value

Fishing mortality rate.

Examples

```
{
filename = system.file("extdata", "data_fishing.csv", package="osmose")
conf = readOsmoseConfiguration(filename)
ndt = getOsmoseParameter(conf, "simulation", "time", "ndtperyear")
nyear = getOsmoseParameter(conf, "simulation", "time", "nyear")

fishing = conf$mortality$fishing

fish0 = getFishingMortality("sp0", fishing, nyear, ndt)
}
```

getMortality	<i>Get the total mortality rate.</i>
--------------	--------------------------------------

Description

Get the total mortality rate.

Usage

```
getMortality(x, stage = "adults", type = "total")
```

Arguments

x	Mortality dataframe
stage	Stage ("adults", etc.)
type	Mortality type ("pred", "starv", "other", "out", "total"). The latter is computed as the sum of all mortality types

Value

A mortality array

Examples

```
{
  dirin = system.file("extdata", package="osmose")
  outdir = paste(dirin, "/outputs", sep="")
  data = read_osmose(outdir)
  mortality_df = data$global$mortality
  mort = getMortality(mortality_df, stage="juveniles", type="total")
}
```

getMortalityDeviation	<i>Computes the mortality deviation. The "proxy", which is removed, can be provided by the user in the "pars" argument.</i>
-----------------------	-----------------------------------------------------------------------------------------------------------------------------

Description

Computes the mortality deviation. The "proxy", which is removed, can be provided by the user in the "pars" argument.

Usage

```
getMortalityDeviation(x, stage, type, pars = NULL)
```

Arguments

x	Mortality dataframe
stage	Stage ("adults", etc.)
type	Mortality type
pars	A list or data frame containing <i>dt.save</i> , <i>M.proxy</i> , <i>dt</i> entries. If NULL, then proxy = colMeans(x)

Value

An array

Examples

```
{
  dirin = system.file("extdata", package="osmose")
  outdir = paste(dirin, "/outputs", sep="")
  data = read_osmose(outdir)
  mortality_df = data$global$mortality
  mortdev = getMortalityDeviation(mortality_df, stage="juveniles", type="total")
}
```

getOsmoseParameter *Get a parameter from a name chain (error if not found).*

Description

Get a parameter from a name chain (error if not found).

Usage

```
getOsmoseParameter(par, ..., keep.att = FALSE)
```

Arguments

par	Output of the link{readOsmoseConfiguration} function
...	String arguments
keep.att	Whether parameter attributes should be kept

Examples

```
{
  filename = system.file("extdata", "inputs/osm_all-parameters.csv", package="osmose")
  par = readOsmoseConfiguration(filename)
  getOsmoseParameter(par, "population", "seeding", "year", "max", keep.att=FALSE)
}
```

getSizeSpectrum	<i>Get size spectrum</i>
-----------------	--------------------------

Description

Get size spectrum

Usage

```
getSizeSpectrum(file, sep = ",", ...)
```

Arguments

file	File to read
sep	File separator
...	Additional arguments of the read.csv function

Value

A 3D array (time, length, species)

Examples

```
{  
  dirin = system.file("extdata", package="osmose")  
  file = paste(dirin, "/outputs/SizeIndicators/gogosm_yieldDistribBySize_Simu0.csv", sep="")  
  size = getSizeSpectrum(file)  
}
```

getVar	<i>Get variable</i>
--------	---------------------

Description

Function to get a variable from an object of osmose class. This function uses the `getVar` method (see the [getVar.osmose](#)).

Usage

```
getVar(object, var, ...)
```

Arguments

object	Object of osmose class (see the read_osmose function).
var	Name of variable to extract. It could be: "biomass", "abundance", "yield", "yieldN".
...	Additional arguments of the function.

Value

An array or a list containing the extracted data.

getVar.osmose	<i>GetVar method for osmose objects</i>
---------------	-----------------------------------------

Description

Get a variable from an osmose object.

Usage

```
## S3 method for class 'osmose'
getVar(object, var, type = "global", expected = FALSE, ...)
```

Arguments

object	Object of osmose class (see the read_osmose function).
var	Name of variable to extract. It could be: "biomass", "abundance", "yield", "yieldN".
type	Type of the variable to extract. By default is type = "global".
expected	A logical parameter. TRUE if the average over the last dimensions should be performed if the output is an array. By default is expected = FALSE.
...	Additional arguments of the function.

Value

An array or a list containing the data.

plot.osmose	<i>Plot method for osmose objects</i>
-------------	---------------------------------------

Description

This method takes a osmose object to make useful plot for each variables.

Usage

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

Arguments

x	Object of osmose class (see the read_osmose function).
type	Name of the variable to plot. By default is type = "biomass" but it could be: <ul style="list-style-type: none"> • "biomass" to plot the species biomass. • "abundance" to plot the species abundance. • "yield" to plot the species yield. • "yieldN" to plot the species yield (in numbers).
...	Extra arguments of the function. These could be: <ul style="list-style-type: none"> • aggregate A logical parameter. By default aggregate = FALSE show the evolution of the variable over the time. aggregate = TRUE show the aggregation of the variable for each species. • start A number to indicate the first x-axis element to plot when aggregate = FALSE. By default start = NULL and start with the first element of the osmose object over the time. • conf A number to indicate the confidence interval to plot. By default conf = 0.95. • factor A number to indicate the variable scale on the y axis. By default factor = 1e-6. • replicates A logical parameter. replicates = FALSE show the mean value of the variable over the time but replicates = TRUE show the values obtained in each replicates of the osmose object. • nrep A number to indicate the number of replicates to show. This parameter is used only when replicates = TRUE. By default nrep = 3. • col The color of the lines (aggregate = FALSE), barplot (for variables biomass or abundance using aggregate = TRUE) or boxplot (for variables yield or yieldN using aggregate = TRUE). By default col = "black". • alpha A number between 0 and 1. Indicate the transparency with which the confidence interval is colored. By default alpha = 0.5.

Details

The parameters: start, conf, factor, replicates, nrep, alpha are used only when aggregate = FALSE.

When aggregate = TRUE a barplot is generated for the variables "biomass" and "abundance". Each bar of the barplot represents the average value of the variable over the time for each species. For the variables "yield" and "yieldN" a boxplot is generated. Each box of the boxplot represents one species.

The plots with the evolution of the variable over the time (aggregate = FALSE) show the variable of each species included in a osmose object, where the x axis represents the time and the y axis the variable. The line (by default a black line) is the mean value over the time and the light grey part represent the confidence interval (by default conf = 0.95).

Value

A graph of a osmose object.

Author(s)

Ricardo Oliveros-Ramos

print.osmose *Print informations about Osmose outputs*

Description

Print informations about Osmose outputs

Usage

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

Arguments

x Osmose outputs (see the [read_osmose](#) function)
... Additional arguments for print.

print.summary.osmose *Print the summary informations about Osmose outputs*

Description

Print the summary informations about Osmose outputs

Usage

```
## 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

```
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.

Examples

```
{
  filename = system.file("extdata", "inputs/osm_all-parameters.csv", package="osmose")
  par = readOsmoseConfiguration(filename)
}
```

readOsmoseFiles *Read Osmose output file*

Description

Read Osmose output file

Usage

```
readOsmoseFiles(path, type, bySpecies = FALSE, ...)
```

Arguments

path	Osmose output path
type	Data type ("biomass", etc)
bySpecies	TRUE if should read one file per species.
...	Additional arguments

Value

Output data frame

read_osmose	<i>Read OSMOSE outputs into an R object</i>
-------------	---------------------------------------------

Description

This function create object of class osmose with the outputs from OSMOSE in the path folder.

Usage

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

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

Details

A list of class osmose is created, individual elements can be extracted using the function getVar.

Author(s)

Ricardo Oliveros-Ramos

Laure Velez

Examples

```
{
  outdir = system.file("extdata", "outputs/", package="osmose")
  read_osmose(outdir)
}
```

runOsmose	<i>Run an OSMOSE configuration</i>
-----------	------------------------------------

Description

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

Usage

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

```
{
  filename = system.file("extdata", "inputs/osm_all-parameters.csv", package="osmose")
  runOsmose(filename)
}
```

summary.osmose	<i>Summarizes informations about Osmose outputs</i>
----------------	-----------------------------------------------------

Description

Summarizes informations about Osmose outputs

Usage

```
## S3 method for class 'osmose'
summary(object, ...)
```

Arguments

object	Osmose outputs (see the read_osmose function)
...	Additional arguments for summary.

write.osmose	<i>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.</i>
--------------	----------------------------------------------------------------------------------------------------------------------------------------------------------

Description

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.

Usage

```
write.osmose(x, file)
```

Arguments

x	Object to be written (table or data frame)
file	Output file

`writeFishingFiles` *Write fishing files from osmose configuration list.*

Description

Write fishing files from osmose configuration list.

Usage

`writeFishingFiles(L1, outputPath)`

Arguments

<code>L1</code>	Osmose configuration (see readOsmoseConfiguration)
<code>outputPath</code>	Output path

Index

- *Topic **ecosystems**
 - osmose-package, 3
- *Topic **marine**
 - osmose-package, 3
- *Topic **modelling**
 - osmose-package, 3

- configureCalibration, 3

- getAverageMortality, 4
- getFishingBaseRate, 4
- getFishingBaseRate.byDt, 5
- getFishingBaseRate.byYear, 6
- getFishingBaseRate.constant, 6
- getFishingBaseRate.linear, 7
- getFishingMortality, 8
- getMortality, 9
- getMortalityDeviation, 9
- getOsmoseParameter, 10
- getSizeSpectrum, 11
- getVar, 11
- getVar.osmose, 11, 12

- osmose (osmose-package), 3
- osmose-package, 3
- osmose2R (read_osmose), 16

- plot.osmose, 12
- print.osmose, 14
- print.summary.osmose, 14

- read_osmose, 11–14, 16, 18
- readOsmoseConfiguration, 3, 15, 19
- readOsmoseFiles, 15
- runOsmose, 17

- summary.osmose, 18

- write.osmose, 18
- writeFishingFiles, 19