

# Package ‘DEPONS2R’

October 26, 2020

**Type** Package

**Title** Read, Plot and Analyse Output from the DEPONS Model

**Version** 1.0.0

**Author** Jacob Nabe-Nielsen

**Maintainer** Jacob Nabe-Nielsen <jnn@bios.au.dk>

**Description** Methods for analyzing population dynamics and movement tracks simulated using the DEPONS model <<https://www.depons.eu>>, and for manipulating input raster files.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 3.5.0)

**Suggests** rgdal

**Imports** raster, methods, sp, utils, grDevices, xml2

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2020-10-26 10:40:12 UTC

## R topics documented:

bathymetry . . . . .	2
coastline . . . . .	3
crs . . . . .	3
DEPONS2R . . . . .	4
DeponsBlockdyn-class . . . . .	4
DeponsDyn-class . . . . .	5
DeponsRaster-class . . . . .	6
DeponsTrack-class . . . . .	7
dyn . . . . .	7
get.latest.sim . . . . .	8
get.simtime . . . . .	8

landscape<- . . . . .	9
make.blocksraster . . . . .	9
plot,DeponsBlockdyn,missing-method . . . . .	11
plot,DeponsDyn,missing-method . . . . .	12
plot,DeponsRaster,ANY-method . . . . .	13
plot,DeponsTrack,missing-method . . . . .	15
porpoisebdyn . . . . .	16
porpoisedyn . . . . .	16
porpoisetrack . . . . .	17
read.DeponsBlockdyn . . . . .	17
read.DeponsDyn . . . . .	18
read.DeponsParam . . . . .	19
read.DeponsRaster . . . . .	20
read.DeponsTrack . . . . .	21
startday . . . . .	22
Summary-methods . . . . .	23
tick.to.time . . . . .	24
title<- . . . . .	24
<b>Index</b>	<b>26</b>

---

bathymetry

*Bathymetry of the Kattegat area*


---

## Description

The standard bathymetry file for Kattegat which is used in DEPONS simulations. It is based on a raster file with 1000 rows and 600 columns where each grid cell corresponds to 400 m x 400 m. Cells on land are assigned a missing data value of -9999.

The Kattegat landscapes use the UTM zone 32 projection, (EPSG:32632) as in the study by Nabe-Nielsen et al (2014). The corresponding proj4string is "+proj=utm +zone=32 +datum=WGS84 +units=m +no\_defs" (see <https://epsg.io/32632>).

## Format

DeponsRaster

## References

Nabe-Nielsen, J., Sibly, R. M., Tougaard, J., Teilmann, J., & Sveegaard, S. (2014). Effects of noise and by-catch on a Danish harbour porpoise population. *Ecological Modelling*, 272, 242–251. <https://doi.org/10.1016/j.ecolmodel.2013.09.025>

## See Also

[DeponsRaster-class](#)

---

coastline	<i>Coastline of Northern Europe</i>
-----------	-------------------------------------

---

### Description

An object of class [SpatialPolygonsDataFrame](#) showing the coastline of the North Sea, Kattegat, and the Western Baltic. The map projection used is ETRS89 – EPSG:3035 projection as for the North Sea raster files used by DEPONS. The corresponding proj4string is "+proj=laea +lat\_0=52 +lon\_0=10 +x\_0=4321000 +y\_0=3210000 +datum=WGS84 +units=m +no\_defs".

### Format

SpatialPolygonsDataFrame

---

crs	<i>Get or set map projection</i>
-----	----------------------------------

---

### Description

Get or set the map projection (also known as coordinate reference system, crs) of DeponsRaster and DeponsTrack objects. For sp objects the text string defining the crs is called the [proj4string](#).

### Usage

```
## S4 method for signature 'DeponsRaster'
crs(x)

## S4 method for signature 'DeponsTrack'
crs(x)

## S4 replacement method for signature 'DeponsTrack'
crs(x) <- value
```

### Arguments

x	Object of class class codeDeponsRaster or DeponsTrack
value	<a href="#">proj4string</a> identifying the map projection

---

 DEPONS2R

*Package for analyzing DEPONS simulation output*


---

### Description

Classes and methods for analyzing and plotting movement tracks and population dynamics simulated using the DEPONS model (<http://www.depons.eu>).

The types of simulated data include:

- `DeponsTrack` movement tracks, read from "RandomPorpoise.XXX.csv" files
- `DeponsDyn` population dynamics data, from "Statistics.XXX.csv" files
- `DeponsBlockdyn` data from "PorpoisePerBlock.XXX.csv" files

Here the `DeponsDyn` data include both changes in population size and energetics through time for the entire landscape, whereas `DeponsBlockdyn` data include variations in population size in different parts (or 'blocks') of the landscape. XXX is the date and time when the simulation was finished.

---

 DeponsBlockdyn-class *DeponsBlockdyn-class*


---

### Description

Stores objects containing population size for different parts of the landscape (i.e. different 'blocks')

### Details

The `dyn` slot contains a data frame with the columns 'tick', which indicates the number of half-hourly time steps since the start of the simulation; a column 'block' indicating the region of the landscape where animals were counted, a 'count' column with the number of animals in that block and tick. The 'real.time' column shows the real-world equivalent to 'tick', i.e. the time that has passed since 'startday'.

### Slots

`title` Character. Name of the object or simulation

`landscape` Character. Identifier for the landscape used in the DEPONS simulations. The landscapes 'DanTysk', 'Gemini', 'Kattegat', 'North Sea', 'Homogeneous', and 'User defined' are distributed with the DEPONS model.

`simtime` `POSIXlt` object with the date and time when the simulation was finished. This is read from the name of the input file.

`startday` `POSIXlt` object with the first day of the simulation, i.e. the first day in the period that the simulations are intended to represent in the real world.

`dyn` Data frame with simulation output.

**Note**

DeponsBlockdyn-objects are usually read in from csv files produced during DEPONS simulations. These files are named 'PorpoisePerBlock.XXX.csv', where XXX indicates the date and time when the simulation was finished.

**See Also**

[plot.DeponsBlockdyn](#) and [read.DeponsBlockdyn](#).

**Examples**

```
a.DeponsBlockdyn <- new("DeponsBlockdyn")
a.DeponsBlockdyn
```

---

DeponsDyn-class

*DeponsDyn-class*


---

**Description**

Stores objects containing population dynamics output and energetic output simulated using the DEPONS model.

**Details**

The following columns are included in the simulation output data frame: 'tick', which indicates the number of half-hourly time steps since the start of the simulation; 'count', which indicates the population size at a given time; 'anim.e', showing the average amount of energy stored by simulated animals; 'lands.e', which shows the total amount of energy in the landscape, and 'real.time' which shows the time relative to 'startday'.

**Slots**

**title** Character. Name of the object or simulation

**landscape** Character. Identifier for the landscape used in the DEPONS simulations. The landscapes 'DanTysk', 'Gemini', 'Kattegat', 'North Sea', 'Homogeneous', and 'User defined' are distributed with the DEPONS model.

**simtime** [POSIXlt](#) object with the date and time when the simulation was finished. This is read from the name of the input file.

**startday** [POSIXlt](#) object with the first day of the simulation, i.e. the first day in the period that the simulations are intended to represent in the real world.

**dyn** Data frame with simulation output.

**Note**

DeponsDyn-objects are usually read in from csv files produced during DEPONS simulations. These files are named 'Statistics.XXX.csv', where XXX indicates the date and time when the simulation was finished.

**See Also**

[plot.DeponsDyn](#) and [read.DeponsDyn](#).

**Examples**

```
a.DeponsDyn <- new("DeponsDyn")
a.DeponsDyn
```

---

DeponsRaster-class      *DeponsRaster-class*

---

**Description**

Stores objects containing raster landscapes used as input in DEPONS simulations.

**Slots**

**type** Character. Identifies the kind of data stored in the raster; should be 'food', 'patches', 'bathymetry', 'dte', 'salinity', 'blocks' or 'NA'.

**landscape** Character Identifier for the landscape used in the DEPONS simulations. The landscapes 'DanTysk', 'Gemini', 'Kattegat', 'North Sea', 'Homogeneous', and 'User defined' are distributed with the DEPONS model.

**crs** Object of class "CRS", i.e. the coordinate reference system. This is provided as a [proj4string](#) text string.

**header** Data frame with data on number of columns and rows in the input raster, the coordinates of the lower left corner, the size of each grid cell and the integer value used to represent missing data.

**ext** Data frame with the extent of the landscape.

**data** The actual data values for each of the grid cells.

**Note**

DeponsRaster-objects are typically read in from ascii raster files that have been used for DEPONS simulations.

**See Also**

[plot.DeponsRaster](#), [read.DeponsRaster](#) and [make.blocksraster](#). [bathymetry](#) is an example of a DeponsRaster-object.

**Examples**

```
a.deponsraster <- new("DeponsRaster")
a.deponsraster
```

---

DeponsTrack-class	<i>DeponsTrack-class</i>
-------------------	--------------------------

---

**Description**

Stores objects containing animal movement tracks simulated using the DEPONS model  
 Classes for manipulating and plotting movement tracks generated with DEPONS.

**Slots**

`title` Name of the object (character)  
`landscape` Name of the object (character)  
`simtime` POSIXlt object with the date and time when the simulation was finished. This is read from the name of the input file.  
`crs` CRS object providing the coordinate reference system used; see [CRS](#) for details  
`tracks` List with one or more tracks, each stored as a [SpatialPointsDataFrame](#) object)

**See Also**

[plot.DeponsTrack](#) and [read.DeponsTrack](#)

---

<code>dyn</code>	<i>Extract population dynamics from objects</i>
------------------	---

---

**Description**

Extract population dynamics from objects

**Usage**

```
## S4 method for signature 'DeponsDyn'
dyn(x)

## S4 method for signature 'DeponsBlockdyn'
dyn(x)
```

**Arguments**

`x` Object of class `DeponsBlockdyn`.

get.latest.sim            *Get name of newest file*

---

### Description

Returns the name of the newest simulation output of a particular type within the specified directory. The date and time are extracted from the file name.

### Usage

```
get.latest.sim(type = "dyn", dir)
```

### Arguments

type	Type of simulation output to check; can be one of: "dyn" (for looking in "Statistics.XX.csv" files), "blockdyn" (for looking in "PorpoisePerBlock.XX.csv" files) "track" (for looking in "RandomPorpoise.XX.csv" files)
dir	Directory to look for simulation output in (character string)

### See Also

[read.DeponsBlockdyn](#) for example.

---

get.simtime            *Get simulation date*

---

### Description

Returns the date and time when a specific simulation was finished (a [POSIXlt](#) object)

### Usage

```
get.simtime(fname = NULL)
```

### Arguments

fname	Character string with name of the file to extract the simulation date from, including the path
-------	--

### See Also

[get.latest.sim](#)



---

landscape<-                    *Get or set the landscape name*

---

### Description

Get or set the landscape name

Get or set the landscape name

### Usage

```
## S4 replacement method for signature 'DeponsTrack'
landscape(x) <- value
```

```
## S4 method for signature 'DeponsTrack'
landscape(x)
```

```
## S4 replacement method for signature 'DeponsDyn'
landscape(x) <- value
```

```
## S4 method for signature 'DeponsDyn'
landscape(x)
```

```
## S4 replacement method for signature 'DeponsBlockdyn'
landscape(x) <- value
```

```
## S4 method for signature 'DeponsBlockdyn'
landscape(x)
```

### Arguments

x	Object of class DeponsBlockdyn.
value	Name of the landscape (character)

---

make.blocksraster            *Makes new file with blocks*

---

### Description

Produces a DeponsRaster object of type='blocks' for use in DEPONS simulations. This allows animals to be counted within specific regions (blocks) of the landscape during the simulation. The new blocks can be specified as either matrices or SpatialPolygons objects. For matrices, the blocks are defined as the smallest rectangle that includes all the specified positions.

**Usage**

```
## S4 method for signature 'DeponsRaster'
make.blocksraster(
  template,
  blocks = NA,
  blockvals = NULL,
  NAvalue = -9999,
  plot = FALSE,
  fname = NULL,
  overwrite = FALSE
)
```

**Arguments**

template	DeponsRaster object used as template for new blocks file
blocks	list of areas to be used for new blocks. Each item in 'blocks' should be a matrix (with two columns, corresponding to x- and y-coordinates) or a SpatialPolygons object
blockvals	Vector of integer values defining the labels of the new blocks. The first value defines the background value, so the length of 'blockvals' should equal the number of blocks plus 1
NAvalue	Value used for missing data in the output object
plot	If TRUE, the raster block is plotted
fname	Name of the output raster file (character string ending with '.asc'). No file is written to disk if fname is not provided.
overwrite	Whether to replace existing file.

**Note**

The blocks file should not be modified when running DEPONS simulations using the 'Kattegat' landscape. In this landscape the simulated animals use the blocks file for navigation. Also note that blocks are added to the new blocks raster in the order they are file in the order in which they are listed in 'blocks', so the order matters if the blocks overlap.

**Examples**

```
#Load file to use as template for new blocks file
data("bathymetry")

# Make list of blocks to create
new.blocks <- list()
x <- runif(8, 700000, 760000)
y <- runif(8, 6200000, 6300000)
new.blocks[[1]] <- cbind(x,y)
x <- c(600000, 635000, 670000, 635000)
y <- c(6150000, 6200000, 6150000, 6100000)
library(sp)
sr1 <- list(Polygon(cbind(x,y)))
```

```

Sr1 <- list(Polygons(sr1, ID=as.vector("p")))
new.blocks[[2]] <- SpatialPolygons(Sr1, proj4string=crs(bathymetry))

make.blocksraster(bathymetry, new.blocks, plot=TRUE)
points(new.blocks[[1]])
plot(new.blocks[[2]], add=TRUE)

the.dir <- tempdir()
make.blocksraster(bathymetry, new.blocks, fname=paste0(the.dir, "/test.asc"))

```

---

plot,DeponsBlockdyn,missing-method

*Plot a DeponsBlockdyn object*

---

## Description

Plot population dynamics simulated with DEPONS

## Usage

```

## S4 method for signature 'DeponsBlockdyn,missing'
plot(x, y, dilute = 5, ...)

```

## Arguments

x	DeponsBlockdyn object
y	Not used
dilute	Integer. Plot only one in every 'dilute' values. Defaults to 5, which yields a plot of the first simulated value and one in every five of the following values.
...	Optional plotting parameters

## Note

The function returns a data frame with numbers of blocks with no agents.

## Examples

```

data("porpoisebdyn")
my.col <- c("red", "darkgreen", "orange")
plot(porpoisebdyn, col=my.col)
legend("bottomright", bty="n", fill=my.col, legend=paste("Block", 0:2))

# Show all data points for small range of x-values
plot(porpoisebdyn, xlim=c(1950, 2050), ylim=c(4850, 5050), type="p", dilute=1, col=my.col)

```

---

```
plot,DeponsDyn,missing-method
      Plot a DeponsDyn object
```

---

**Description**

Plot population dynamics simulated with DEPONS

**Usage**

```
## S4 method for signature 'DeponsDyn,missing'
plot(x, y, dilute = 5, plot.energy = TRUE, plot.legend = TRUE, ...)
```

**Arguments**

x	DeponsDyn object
y	Not used
dilute	Integer. Plot only one in every 'dilute' values. Defaults to 5, which yields a plot of the first simulated value and one in every five of the following values.
plot.energy	If set to TRUE it plots the amount of energy stored in simulated and in the landscape in addition to the population count
plot.legend	If set to TRUE, a legend is plotted
...	Optional plotting parameters

**Examples**

```
data("porpoisedyn")

# Plot for specific range of years
rg <- c(as.POSIXlt("2011-01-01"), as.POSIXlt("2018-12-31"))
plot(porpoisedyn, xlim=as.POSIXct(rg), plot.energy=TRUE)

## Not run:
# Read data from default DEPONS simulation directory:
sim.dir <- "/Applications/DEPONS 2.1/DEPONS"
new.sim.name <- get.latest.sim(dir=sim.dir)
new.sim.out <- read.DeponsDyn(fname=paste(sim.dir, new.sim.name, sep="/"))
plot(new.sim.out)

## End(Not run)
```

---

 plot,DeponsRaster,ANY-method

*Plot a DeponsRaster object*


---

### Description

Plot the values in a DeponsRaster object. Porpoisetracks or other kinds of lines, poits etc. can be drawn on top of the plot by adding

### Usage

```
## S4 method for signature 'DeponsRaster,ANY'
plot(
  x,
  y,
  maxpixels = 5e+05,
  col,
  alpha = NULL,
  colNA = NA,
  add = FALSE,
  ext = NULL,
  useRaster = TRUE,
  interpolate = FALSE,
  addfun = NULL,
  nc,
  nr,
  maxnl = 16,
  main,
  npretty = 0,
  axes = TRUE,
  legend = TRUE,
  trackToPlot = 1,
  ...
)
```

### Arguments

x	DeponsRaster object
y	A DeponsTrack object or missing
maxpixels	integer > 0. Maximum number of cells to use for the plot. If maxpixels < ncell(x), sampleRegular is used before plotting.
col	A color palette, i.e. a vector of n contiguous colors. Reasonable defaults are provided.
alpha	Number between 0 and 1 to set transparency. 0 is entirely transparent, 1 is not transparent (NULL is equivalent to 1)

<code>colNA</code>	The color to use for the background (default is transparent)
<code>add</code>	Logical. Whether to add map to the current plot
<code>ext</code>	An extent object allowing to plot only part of the map
<code>useRaster</code>	If TRUE, the <code>rasterImage</code> function is used for plotting. Otherwise the <code>image</code> function is used
<code>interpolate</code>	Logical. Should the image be interpolated (smoothed)? Only used when <code>useRaster = TRUE</code>
<code>addfun</code>	Function to add additional items such as points or polygons to the plot
<code>nc</code>	Not used for plotting <code>DeponsRaster</code> objects
<code>nr</code>	Not used for plotting <code>DeponsRaster</code> objects
<code>maxnl</code>	Not used for plotting <code>DeponsRaster</code> objects
<code>main</code>	Character. Plot title
<code>npretty</code>	Integer. Number of decimals for pretty labels on the axes
<code>axes</code>	Whether to plot tick marks
<code>legend</code>	Whether to plot the colour legend
<code>trackToPlot</code>	Integer indicating which track to plot if the <code>DeponsTrack</code> object contains more than one track. Ignored if <code>y</code> is missing
<code>...</code>	Other optional plotting parameters

**See Also**

See method for [plot](#) in the `raster` package for details.

**Examples**

```
data("bathymetry")
plot(bathymetry)
data("coastline")
library(rgdal)
# Change projection of coastline to match that of bathymetry data
coastline2 <- spTransform(coastline, crs(bathymetry))
plot(coastline2, add=TRUE, col="lightyellow2")

text(512000, 6240000, 'Denmark')
text(800000, 6300000, 'Sweden')

plot(bathymetry, axes=FALSE, legend=FALSE, main="Simulated porpoise track")
data("porpoisetrack")
plot(porpoisetrack, add=TRUE)
```

---

plot,DeponsTrack,missing-method  
*Plot a DeponsTrack object*

---

## Description

Plot the coordinates in a movement track simulated with DEPONS.

## Usage

```
## S4 method for signature 'DeponsTrack,missing'  
plot(x, y, trackToPlot = 1, add = FALSE, ...)
```

## Arguments

x	DeponsTrack object
y	Not used
trackToPlot	Integer; indicates which track to plot if there is more than one track in the object. Defaults to 1
add	Logical, whether to add the track to an existing plot one animal was tracked during the simulation.
...	Optional plotting parameters

## Examples

```
data(porpoisetrack)  
data("porpoisetrack")  
plot(porpoisetrack)  
  
# Optional: Transform and plot coastline if rgdal is installed  
data("coastline")  
library(rgdal)  
coast2 <- sp::spTransform(coastline, CRS("+proj=utm +zone=32 +datum=WGS84 +units=m +no_defs"))  
# Check that the coastline-map is projected the same way as the track before  
# plotting:  
as.character(crs(porpoisetrack)) == as.character(crs(coast2))  
plot(coast2, col="lightyellow2", add=TRUE)
```

---

porpoisebdyn

*Simulated porpoise population dynamics*

---

**Description**

An object of class DeponsBlockdyn with output from a DEPONS simulation based on the North Sea landscape, using a landscape divided into two blocks. Numbers of animals are counted per block.

**Format**

DeponsBlockdyn

**See Also**

[DeponsBlockdyn-class](#), [porpoisedyn](#)

---

porpoisedyn

*Simulated porpoise population dynamics*

---

**Description**

An object of class DeponsDyn with output from a DEPONS simulation based on the Kattegat landscape, assuming that the simulation represents the period 2010-01-01 onward in the real world. Number of animals and energy availability is recorded for the entire landscape.

**Format**

DeponsDyn

**See Also**

[DeponsDyn-class](#), [porpoisebdyn](#)



---

porpoisetrack	<i>Simulated porpoise track</i>
---------------	---------------------------------

---

**Description**

An object with five elements: `title`, `landscape`, `simtime`, `crs`, and `tracks`. The `crs` stores information about the map projection used ("`+proj=utm +zone=32 +datum=WGS84 +units=m +no_defs`"). The `tracks` element is a list of objects of class `SpatialPointsDataFrame`, each of which corresponds to one simulated animal. `simtime` is the simulation date.

**Format**

DeponsTrack

**See Also**

[DeponsTrack-class](#)

---

<code>read.DeponsBlockdyn</code>	<i>Reading simulated population count for blocks</i>
----------------------------------	--

---

**Description**

Function for reading DEPONS simulation output with number of animals per block for each time step.

**Usage**

```
read.DeponsBlockdyn(
  fname,
  title = "NA",
  landscape = "NA",
  simtime = "NA",
  startday = "NA"
)
```

**Arguments**

<code>fname</code>	Name of the file (character) that contains movement data generated by DEPONS. The name includes the path to the directory if this is not the current working directory.
<code>title</code>	Optional character string giving name of simulation
<code>landscape</code>	The landscape used in the simulation
<code>simtime</code>	Optional POSIXlt object with date of simulation. If not provided this is obtained from name of input file
<code>startday</code>	The start of the period that the simulation represents, i.e. the real-world equivalent of 'tick 1' (POSIXlt)

**See Also**

See [DeponsBlockdyn-class](#) for details on what is stored in the output object and [read.DeponsParam](#) for reading the parameters used in the simulation.

**Examples**

```
## Not run:
# File loaded from default location
the.file <- "/Applications/DEPONS 2.1/DEPONS/PorpoisePerBlock.2020.Sep.02.20_24_17.csv"
file.exists(the.file)
porpoise.blockdyn <- read.DeponsBlockdyn(fname=the.file,
  title="Test simulation with two blocks", landscape="North Sea")
porpoise.blockdyn

# Get the latest simulation
the.file <- get.latest.sim(type="blockdyn", dir="/Applications/DEPONS 2.1/DEPONS")
porpoise.blockdyn <- read.DeponsBlockdyn(fname=the.file)

## End(Not run)
```

---

read.DeponsDyn	<i>Reading DEPONS simulation output</i>
----------------	---

---

**Description**

Function for reading simulation output produced by DEPONS.

**Usage**

```
read.DeponsDyn(
  fname,
  title = "NA",
  landscape = "NA",
  simtime = "NA",
  startday = "NA"
)
```

**Arguments**

fname	Name of the file (character) that contains movement data generated by DEPONS. The name includes the path to the directory if this is not the current working directory.
title	Optional character string giving name of simulation
landscape	The landscape used in the simulation
simtime	Optional POSIXlt object with the date and time when the simulation finished. If not provided this is obtained from name of input file
startday	The start of the period that the simulation represents, i.e. the real-world equivalent of 'tick 1' (character string of the form 'yyyy-mm-dd', or POSIXlt)

## See Also

See [DeponsDyn-class](#) for details on what is stored in the output object.

## Examples

```
## Not run:
dyn.file <- "/Applications/DEPONS 2.1/DEPONS/Statistics.2020.Sep.02.20_24_17.csv"
file.exists(dyn.file)
porpoisedyn <- read.DeponsDyn(fname, startday=as.POSIXlt("2010-01-01"))
porpoisedyn

## End(Not run)
```

---

read.DeponsParam	<i>Read simulation parameters</i>
------------------	-----------------------------------

---

## Description

Read the parameters that were used for running a specific DEPONS simulation

## Usage

```
read.DeponsParam(fname)
```

## Arguments

fname	Name of the XML file (character) that contains the parameter list used for running a DEPONS simulation. The name includes the path to the directory if this is not the current working directory.
-------	---

## Details

The parameter file can be generated from within DEPONS by pressing the 'Save' icon after modifying the user settings on the 'Parameters' tab within the main DEPONS model window. See TRACE document for details regarding the parameters in the model: <https://github.com/jacobnabe/DEPONS>. It is strongly recommended that the parameter list is stored with the simulation output.

## Examples

```
## Not run:
# Parameters read from file created by DEPONS run in interactive mode
the.file <- "/Applications/DEPONS 2.1/DEPONS/DEPONS.rs/parameters.xml"
pfile <- read.DeponsParam(the.file)

## End(Not run)
```

---

read.DeponsRaster      *Reading DEPONS raster files*

---

## Description

Function for reading raster files that have been used in DEPONS simulations. DEPONS rasters define amount of food available for simulated animals, spatial distribution of food patches, bathymetry, and distance to coast (dte). The 'blocks' raster enables the user to count animals in specific parts of the landscape during simulations. See Nabe-Nielsen et al. (2018) for details regarding these files. In DEPONS 2.0 the salinity raster file was introduced; see TRACE document for details: <https://github.com/jacobjacob/DEPONS>

## Usage

```
read.DeponsRaster(fname, type = "NA", landscape = "NA", crs = "NA")
```

## Arguments

fname	Filename (character), including the path to the DEPONS raster file.
type	The kind of data stored in the raster; c('food', 'patches', 'bathymetry', 'dte', 'salinity', 'blocks').
landscape	Identifier for the landscape used in the DEPONS simulations; typically set to 'North Sea'.
crs	CRS-object providing the map projection (see <a href="#">CRS</a> ).

## Value

Returns a DeponsRaster object. The object inherits slots from the "RasterLayer" class, including "title", which is used for storing the file name.

## References

Nabe-Nielsen, J., van Beest, F. M., Grimm, V., Sibly, R. M., Teilmann, J., & Thompson, P. M. (2018). Predicting the impacts of anthropogenic disturbances on marine populations. *Conservation Letters*, 11(5), e12563. <https://doi.org/10.1111/conl.12563>

## See Also

[DeponsRaster-class](#)

---

read.DeponsTrack      *Reading DEPONS track files*

---

## Description

Function for reading movement tracks produced by DEPONS. These describe movements of simulated animals within the simulation landscape, where the positions after each 30-min time step are provided using the coordinate reference system that were used for generating these landscapes. See van Beest et al. (2018) and Nabe-Nielsen et al. (2013) for details regarding how these files were generated as a balance between correlated random walk behaviour and spatial memory behaviour, which allows animals to return to previously visited food patches.

## Usage

```
read.DeponsTrack(  
  fname,  
  title = "NA",  
  landscape = "NA",  
  simtime = "NA",  
  crs = CRS(as.character(NA)),  
  tz = "UTC"  
)
```

## Arguments

<code>fname</code>	Name of the file (character) that contains movement data generated by DEPONS. The name includes the path to the directory if this is not the current working directory.
<code>title</code>	Optional character string giving name of simulation
<code>landscape</code>	Optional character string with the landscape used in the simulation
<code>simtime</code>	Optional POSIXlt object with date of simulation. If not provided this is obtained from name of input file
<code>crs</code>	Character, coordinate reference system (map projection)
<code>tz</code>	Time zone used in simulations. Defaults to UTC/GMT. #'

## Value

Returns an object with the elements `title`, `simtime`, `crs`, and `tracks`. The date is extracted from input data if not provided explicitly and stored as a [POSIXlt](#) object. The element `tracks` is a list of objects of class [SpatialPointsDataFrame](#), each of which corresponds to one simulated animal (several animals can be tracked in one simulation).

**Examples**

```

data(porpoisetrack) # Load data for use in example

# Use standard DEPONS coordinate reference system / map projection:
the.crs <- "+proj=laea +lat_0=52 +lon_0=10 +x_0=4321000 +y_0=3210000
+datum=WGS84 +units=m +no_defs"

## Not run:
one.fname <- "~/Applications/DEPONS/
RandomPorpoise.2020.Jul.31.09_43_10.csv"

porpoisetrack <- read.DeponsTrack(one.fname, title="Track simulated using DEPONS 2.0",
  crs=the.crs)

## End(Not run)

# Plot the first of the simulated tracks
plot(porpoisetrack)

```

---

startday

*Get or set start date for simulation*


---

**Description**

Get or set start date for simulation

Get or set start date for simulation

**Usage**

```

## S4 method for signature 'DeponsBlockdyn'
startday(x)

## S4 method for signature 'DeponsDyn'
startday(x)

## S4 replacement method for signature 'DeponsBlockdyn'
startday(x) <- value

## S4 replacement method for signature 'DeponsDyn'
startday(x) <- value

```

**Arguments**

x	Object of class DeponsDyn
value	POSIXlt or character string of the form 'yyyy-mm-dd'

**Details**

The start date indicates the start of the period that the simulation is supposed to represent.

The start date indicates the start of the period that the simulation is supposed to represent.

**Note**

The assignment of a new start time is currently quite time consuming.

---

Summary-methods	<i>Summary</i>
-----------------	----------------

---

**Description**

Summarizes different kinds of objects created based on output from the DEPONS model

**Usage**

```
## S4 method for signature 'DeponsBlockdyn'  
summary(object)
```

```
## S4 method for signature 'DeponsDyn'  
summary(object)
```

```
## S4 method for signature 'DeponsRaster'  
summary(object)
```

```
## S4 method for signature 'DeponsTrack'  
summary(object)
```

**Arguments**

object            Depons\* object

**Details**

The summary method is available for [DeponsTrack-class](#), [DeponsDyn-class](#), [DeponsRaster-class](#), and [DeponsBlockdyn-class](#)-objects.

---

tick.to.time	<i>Convert tick number to date</i>
--------------	------------------------------------

---

### Description

Converts the number of ticks since the start of the simulation to a specific date while taking into account that DEPONS assumes that there is 30 days per month. Returns an object of class [as.POSIXlt](#)

### Usage

```
tick.to.time(tick, ...)
```

### Arguments

tick	Numeric; tick number
...	Time zone (tz) and other parameters

### Note

The conversion from ticks to date is problematic in February, with <30 days. Here time is forced to 'stands still'.

---

title<-	<i>Get or set the title of DeponsTrack objects</i>
---------	--

---

### Description

Get or set the title of DeponsTrack objects

Get or set the title of DeponsDyn objects

### Usage

```
## S4 replacement method for signature 'DeponsTrack'
title(x) <- value
```

```
## S4 method for signature 'DeponsTrack'
title(x, value)
```

```
## S4 replacement method for signature 'DeponsDyn'
title(x) <- value
```

```
## S4 method for signature 'DeponsDyn'
title(x, value)
```



*title*<-

25

**Arguments**

x	Object of class DeponsDyn.
value	Character string

# Index

- \* **datasets**
  - bathymetry, [2](#)
  - coastline, [3](#)
  - porpoisebdyn, [16](#)
  - porpoisedyn, [16](#)
  - porpoisetrack, [17](#)
- as.POSIXlt, [24](#)
- bathymetry, [2](#), [6](#)
- coastline, [3](#)
- CRS, [7](#), [20](#)
- crs, [3](#)
- crs, DeponsRaster-method (crs), [3](#)
- crs, DeponsTrack-method (crs), [3](#)
- crs<- (crs), [3](#)
- crs<- , DeponsTrack-method (crs), [3](#)
- DEPONS2R, [4](#)
- DeponsBlockdyn, [4](#)
- DeponsBlockdyn-class, [4](#)
- DeponsDyn, [4](#)
- DeponsDyn-class, [5](#)
- DeponsRaster-class, [6](#)
- DeponsTrack, [4](#)
- DeponsTrack-class, [7](#)
- dyn, [7](#)
- dyn, DeponsBlockdyn-method (dyn), [7](#)
- dyn, DeponsDyn-method (dyn), [7](#)
- get.latest.sim, [8](#), [8](#)
- get.simtime, [8](#)
- landscape (landscape<-), [9](#)
- landscape, DeponsBlockdyn-method (landscape<-), [9](#)
- landscape, DeponsDyn-method (landscape<-), [9](#)
- landscape, DeponsTrack-method (landscape<-), [9](#)
- landscape<- , [9](#)
- landscape<- , DeponsBlockdyn-method (landscape<-), [9](#)
- landscape<- , DeponsDyn-method (landscape<-), [9](#)
- landscape<- , DeponsTrack-method (landscape<-), [9](#)
- make.blocksraster, [6](#), [9](#)
- make.blocksraster, DeponsRaster-method (make.blocksraster), [9](#)
- plot, [14](#)
- plot, DeponsBlockdyn, missing-method, [11](#)
- plot, DeponsDyn, missing-method, [12](#)
- plot, DeponsRaster, ANY-method, [13](#)
- plot, DeponsRaster, DeponsTrack-method (plot, DeponsRaster, ANY-method), [13](#)
- plot, DeponsTrack, missing-method, [15](#)
- plot.DeponsBlockdyn, [5](#)
- plot.DeponsBlockdyn (plot, DeponsBlockdyn, missing-method), [11](#)
- plot.DeponsDyn, [6](#)
- plot.DeponsDyn (plot, DeponsDyn, missing-method), [12](#)
- plot.DeponsRaster, [6](#)
- plot.DeponsRaster (plot, DeponsRaster, ANY-method), [13](#)
- plot.DeponsTrack, [7](#)
- plot.DeponsTrack (plot, DeponsTrack, missing-method), [15](#)
- porpoisebdyn, [16](#), [16](#)
- porpoisedyn, [16](#), [16](#)
- porpoisetrack, [17](#)
- POSIXlt, [4](#), [5](#), [8](#), [21](#)

proj4string, [3](#), [6](#)

read.DeponsBlockdyn, [5](#), [8](#), [17](#)  
read.DeponsDyn, [6](#), [18](#)  
read.DeponsParam, [18](#), [19](#)  
read.DeponsRaster, [6](#), [20](#)  
read.DeponsTrack, [7](#), [21](#)

SpatialPointsDataFrame, [7](#), [17](#), [21](#)  
SpatialPolygonsDataFrame, [3](#)  
startday, [22](#)  
startday, DeponsBlockdyn-method  
    (startday), [22](#)  
startday, DeponsDyn-method (startday), [22](#)  
startday<- (startday), [22](#)  
startday<- , DeponsBlockdyn-method  
    (startday), [22](#)  
startday<- , DeponsDyn-method (startday),  
    [22](#)  
summary (Summary-methods), [23](#)  
summary, DeponsBlockdyn-method  
    (Summary-methods), [23](#)  
summary, DeponsDyn-method  
    (Summary-methods), [23](#)  
summary, DeponsRaster-method  
    (Summary-methods), [23](#)  
summary, DeponsTrack-method  
    (Summary-methods), [23](#)  
Summary-methods, [23](#)

tick.to.time, [24](#)  
title (title<-), [24](#)  
title, DeponsDyn-method (title<-), [24](#)  
title, DeponsTrack-method (title<-), [24](#)  
title<- , [24](#)  
title<- , DeponsDyn-method (title<-), [24](#)  
title<- , DeponsTrack-method (title<-), [24](#)