Package 'ecolMod'

November 15, 2022

Version 1.2.6.4

Date 2022-11-13

Title ``A Practical Guide to Ecological Modelling - Using R as a Simulation Platform''

Author
Karline Soetaert <karline.soetaert@nioz.nl>, Peter MJ Herman <peter.herman@nioz.nl>

Maintainer Karline Soetaert <karline.soetaert@nioz.nl>

Depends R (>= 2.01), rootSolve, shape, diagram, deSolve

Imports stats, graphics, grDevices

Suggests maps, seacarb, scatterplot3d, deldir

Description
Figures, data sets and examples from the book ``A practical guide to ecological modelling - using R as a simulation platform'' by Karline Soetaert and Peter MJ Herman (2009). Springer. All figures from chapter x can be generated by ``demo(chapx)'', where x = 1 to 11. The R-scripts of the model examples discussed in the book are in subdirectory ``examples'', ordered per chapter. Solutions to model projects are in the same subdirectories.

License GPL (>= 2)

LazyData yes

NeedsCompilation yes

Repository CRAN

Date/Publication 2022-11-15 19:50:05 UTC

R topics documented:

---

ecolMod-package	*A practical guide to ecological modelling - using R as a simulation platform*

Description

Figures, data sets and examples from the book "A practical guide to ecological modelling - using R as a simulation platform" by Karline Soetaert and Peter MJ Herman (2009). Springer, 372 pp.

http://www.springer.com/life+sciences/ecology/book/978-1-4020-8623-6

All figures from chapter x can be generated by demo(chapx), where x = 1 to 11.

The R-scripts of the model examples discussed in the book are in subdirectory "examples", ordered per chapter.

Solutions to model projects are in the same subdirectories.

Details

Package:	ecolMod
Type:	Package
Version:	1.2.3
Date:	2011-03-15
License:	GNU Public License 2 or above

Author(s)

Karline Soetaert (Maintainer) and Peter M.J. Herman

See Also

- OMEXDIAsteady steady-state application of the OMEXDIA diagenetic model - a fortran DLL
- SCOC a Sediment Community Oxygen Consumption (SCOC) dataset
- Zoogrowth a zooplankton growth dataset
- deepCmin results of the calibration exercise from chapter 4.4.4
- dilution Draws dilution culture setup
- pricefit Pseudo-random search algorithm of Price (1997)

Examples

```
## Not run:
## show examples (see respective help pages for details)
example(pricefit)
```

```
example(SCOC)

## run demos
demo("chap1") # chapter 1. Introduction
demo("chap2") # chapter 2. Model formulation
demo("chap3") # chapter 3. Spatial components and transport
demo("chap4") # chapter 4. Parameterisation
demo("chap5") # chapter 5. Model solution - analytical methods
demo("chap6") # chapter 6. Model solution - numerical methods
demo("chap7") # chapter 7. Stability and steady-state
demo("chap8") # chapter 8. Multiple time scales and equilibrium processes
demo("chap9") # chapter 9. Discrete time models
demo("chap10") # chapter 10. Dynamic programming
demo("chap11") # chapter 11. Testing and validating the model

## open the directory with source code of demos
browseURL(paste(system.file(package="ecolMod"), "/demo", sep=""))

## open the directory with book examples
browseURL(paste(system.file(package="ecolMod"), "/doc/examples", sep=""))

## ecolMod vignette:
vignette("ecolMod")

## End(Not run)
```

---

deepCmin   *results of the calibration exercise from chapter 4.4.4*

Description

This datafile contains output from the calibration exercise from chapter 4.4.4. They are:
vectors kseries, multser
vector outcost
matrices optpar, optpar20, optpar25

Usage

```
deepCmin
```

Author(s)

Karline Soetaert <karline.soetaert@nioz.nl>

Examples

```
pgr <- gray.colors(n = 25, start = 0.95, end = 0.0)
with (deepCmin,
filled.contour(x = multser, y = kseries, z = outcost,
    ylab = "k (/day)", xlab = "multiplication factor (-)",
```

```
    main = "Model cost landscape", col = pgr, nlevels = 25,
    plot.axes = {
      axis(1); axis(2);
      points(optpar20$poppar[,2], optpar20$poppar[,1], pch = "o", cex = .5);
      points(optpar25$poppar[,2], optpar25$poppar[,1], pch = "+", cex = 1);
      points(optpar$par[2], optpar$par[1], pch = 16, cex = 2)
    }
  )
  )
```

---

dilution   *Draws dilution culture setup*

Description

Draws the framework of a dilution culture. Used as a template to plot the flow diagrams for dilution-type models, where there is a continuous inflow of medium from a vessel into a well-stirred tank (the culture vessel). The volume in the culture tank stays constant.

Usage

```
dilution(main = c("",""), int = "")
```

Arguments

main	main text, consiting of two strings, positioned above medium vessel (1st string) and culture vessel (2nd string) of dilution diagram plot
int	text above the dripping outlets

Value

a list containing:

p1	two-valued vector with the x-y positions of the middle of the (large) culture vessel
p2	two-valued vector with the x-y positions of the middle of the (small) medium vessel

Author(s)

Karline Soetaert <karline.soetaert@nioz.nl>

Examples

```
par(mar = c(0, 0, 0, 0))
dd <- dilution(main = c("Stock", "Stirred tank"), int = "Flow,Q")
text(dd$p2[1], dd$p2[2], "Ain", font = 2)
text(dd$p1[1], dd$p1[2]+0.03, "Volume V", font = 2)
text(dd$p1[1], dd$p1[2]-0.03, "[A]", font = 2)
```

| OMEXDIAsteady | *steady-state application of the OMEXDIA diagenetic model - fortran DLL* |

Description

A 1-D model of Carbon, nitrogen and oxygen diagenesis in a marine sediment.
The model describes six state variables, in **100** layers:

- 2 fractions of organic carbon (FDET,SDET): fast and slow decaying, solid substance
- Oxygen (O2), dissolved substance
- Nitrate (NO3), dissolved substance
- Ammonium (NH3), dissolved substance
- Oxygen demand unit (ODU), dissolved substance, as lump sum of all reduced substances other than ammonium

See Soetaert et al., 1996 for further details of the model.

This is a simplified version of the OMEXDIA model, added just to create a figure in the book.
A more complete version will be published in a separate R-package that will deal with reactive transport modelling in R.
The name of this package is not yet decided upon.

Usage

```
OMEXDIAsteady()
```

Details

The model application just estimates the steady-state condition of the OMEXDIA model, the parameter values are such that there is almost no overlap between the oxic and anoxic zone of the sediment.

For efficiency reasons, the OMEXDIA diagenetic model was written in Fortran, and this code linked to the package.

Value

a list containing:

steady	The steady-state condition of the state variables, a vector containing steady-state concentrations of FDET(0-100), SDET(101-200), O2 (201-300), NO3 (301-400), NH3 (401-500) and ODU (501-600)
precis	the precision of the steady-state solution
Solved	a logical, TRUE when steady-state has been reached

Author(s)

Karline Soetaert <karline.soetaert@nioz.nl>

References

Soetaert K, PMJ Herman and JJ Middelburg, 1996a. A model of early diagenetic processes from the shelf to abyssal depths. Geochimica Cosmochimica Acta, 60(6):1019-1040.

Soetaert K, PMJ Herman and JJ Middelburg, 1996b. Dynamic response of deep-sea sediments to seasonal variation: a model. Limnol. Oceanogr. 41(8): 1651-1668.

Examples

```
N     <- 100
Depth <- seq(0.05, by = 0.1, len = 100)
out   <- OMEXDIAsteady()

# Steady-state concentrations in sediment
CONC  <- out$steady

FDET  <- CONC[1:N]
SDET  <- CONC[(N+1)  :(2*N)]
O2    <- CONC[(2*N+1):(3*N)]
NO3   <- CONC[(3*N+1):(4*N)]
NH3   <- CONC[(4*N+1):(5*N)]
ODU   <- CONC[(5*N+1):(6*N)]

TOC  <- (FDET+SDET)*1200/10^9/2.5    # % organic carbon (excess)

par(mfrow=c(2, 2))
plot(TOC, Depth, ylim = c(10, 0), xlab = "procent", main = "TOC",
       type = "l", lwd=2)
plot(O2, Depth, ylim = c(10, 0), xlab = "mmol/m3", main = "O2",
       type = "l", lwd = 2)
plot(NO3, Depth, ylim = c(10, 0), xlab = "mmol/m3", main = "NO3",
       type = "l", lwd = 2)
plot(NH3, Depth, ylim = c(10, 0), xlab = "mmol/m3", main = "NH3",
       type = "l", lwd = 2)


mtext(outer=TRUE,side=3,line=-2,cex=1.5,"OMEXDIAmodel")
```

---

pricefit	*Pseudo-random search algorithm of Price (1997)*

Description

Pseudo-random search algorithm of Price (1997). Used in the book as an example of a random-based fitting technique, and as an example of how to create a function in R.

Usage

```
pricefit(par, minpar = rep(-1e8, length(par)),
         maxpar = rep(1e8, length(par)), func,
         npop = max(5*length(par),50),
         numiter =10000, centroid = 3, varleft = 1e-8,...)
```

Arguments

par	initial values of the parameters to be optimised
minpar	minimal values of the parameters to be optimised
maxpar	maximal values of the parameters to be optimised
func	function to be minimised, its first argument should bw the vector of parameters over which minimization is to take place. It should return a scalar result, the model cost, e.g the sum of squared residuals.
npop	number of elements in population
numiter	number of iterations to be performed. Defaults to 10000. There is no other stopping criterion.
centroid	number of elements from which to estimate new parameter vector
varleft	relative variation remaining; if below this value algorithm stops
...	arguments passed to funtion func

Details

see the book of Soetaert and Herman for a description of the algorithm AND for a line to line explanation of the function code.

Value

a list containing:

par	the optimised parameter values
cost	the model cost, or function evaluation associated to the optimised parameter values, i.e. the minimal cost
poppar	all parameter vectors remaining in the population, matrix of dimension (npop,length(par))
popcost	model costs associated with all population parameter vectors, vector of length npop

Author(s)

Karline Soetaert <karline.soetaert@nioz.nl>

References

Soetaert, K. and P.M.J. Herman, 2009. A Practical Guide to Ecological Modelling. Using R as a Simulation Platform. Springer, 372 pp.

Price, W.L., 1977. A controlled random search procedure for global optimisation. The Computer Journal, 20: 367-370.

See Also

[optim](#) the R default.

Examples

```
pricefit  # will display the code

amp    <- 6
period <- 5
phase  <- 0.5

x <- runif(20)*13
y <- amp*sin(2*pi*x/period+phase) + rnorm(20, mean = 0, sd = 0.05)
plot(x, y, pch = 16)


cost <- function(par) sum((par[1]*sin(2*pi*x/par[2]+par[3])-y)^2)

p1 <- optim(par = c(amplitude = 1, phase = 1, period = 1), cost)
p2 <- optim(par = c(amplitude = 1, phase = 1, period = 1), cost,
            method = "SANN")
p3 <- pricefit(par = c(amplitude = 1, phase = 1, period = 1),
            minpar = c(0, 1e-8,0), maxpar = c(100, 2*pi,100),
            func = cost, numiter = 3000)

curve(p1$par[1]*sin(2*pi*x/p1$par[2] + p1$par[3]), lty = 2, add = TRUE)
curve(p2$par[1]*sin(2*pi*x/p2$par[2] + p2$par[3]), lty = 3, add = TRUE)
curve(p3$par[1]*sin(2*pi*x/p3$par[2] + p3$par[3]), lty = 1, add = TRUE)

legend ("bottomright", lty = c(1, 2, 3),
        c("Price", "Mathematical", "Simulated annealing"))
```

SCOC   *a Sediment Community Oxygen Consumption (SCOC) dataset*

Description

This literature dataset, compiled by Andersson et al. (2004) contains 584 measurements of sediment community oxygen consumption rates, as a function of water depth, and performed in deep-water sediments, either by in situ incubations or via modelling of oxygen microprofiles.

It is used in the book to demonstrate how one can obtain order-of-magnitude estimates of model parameters (i.c. sediment oxygen consumption rate, a measure of deposition flux) by performing log-log regression with water depth.

Usage

```
SCOC
```

Format

a dataframe with 584 rows, and with following columns:
Depth.m, the water depth at which the measurement was performed.
SCOC.mmol/m2/d, the oxygen consumption rate of the sediment, [mmolO2/m2/d]

Author(s)

Karline Soetaert <karline.soetaert@nioz.nl>

References

Andersson, H., Wijsman, J., Herman, P., Middelburg, J., Soetaert, K., Heip, C., 2004. Respiration patterns in the deep ocean. Geophysical Research Letters 31, LO3304.

See Also

[Zoogrowth](), a dataset containing zooplankton maximal growth rates
see the paper of Andersson et al. for a description of the original literature sources of this dataset

Examples

```
plot(SCOC[,1], SCOC[,2], log = "xy", xlab = "water depth, m",
     ylab = "", main = "SCOC, mmol O2/m2/d", pch = 16,
     xaxt = "n", yaxt = "n", cex.main = 1)

axis(1, at = c(0.5, 5, 50, 500, 5000),
     labels = c("0.5", "5", "50", "500", "5000"))
axis(2, at = c(0.1, 1, 10, 100),
     labels = c("0.1", "1", "10", "100"))

ll <- lm(log(SCOC[,2])~ log(SCOC[,1]))
rr <- summary(ll)$r.squared
A  <- exp(coef(ll)[1])
B  <- (coef(ll)[2])
curve(A*x^B, add = TRUE, lwd = 2)
AA <- round(A*100)/100
BB <- round(B*100)/100
expr <- substitute(y==A*x^B, list(A=AA,B=BB))
text(1, .1, expr, adj = 0)
expr2 <- substitute(r^2==rr, list(rr=round(rr*100)/100))
text(1, 0.04, expr2, adj = 0)
```

Zoogrowth   *a zooplankton growth dataset*

Description

This literature dataset, compiled by Hansen et al. (1997) contains 84 measurements of maximal growth rates as a function of organism volume and temperature for various species of zooplankton. The maximal growth rates were obtained from laboratory experiments.

It is used in the book to demonstrate how one can obtain order-of-magnitude estimates of model parameters (i.c. maximal growth) via allometric relations, i.e. by performing log-log regression of organism rates versus size.

Usage

```
Zoogrowth
```

Format

a dataframe with 84 rows, and with following columns:
Volume.um3, the volume in [um 3].
Mumax.hr, the maximal growht rate, [/hour]
Species, the name of reared zooplankton species
Temp.dgC, the rearing temperature, [dg C]
Group, the systematic group to which the organism belongs, one of Nanoflagellate, Dinoflagellate, Ciliate, Rotifer, Meroplankton, Copepod

Author(s)

Karline Soetaert <karline.soetaert@nioz.nl>

References

Hansen PJ, Bjornsen PK, Hansen BW, 1997. Zooplankton grazing and growth: Scaling within the 2-2,000-mu m body size range. Limnology and Oceanograpy 42: 687-704.

See Also

[SCOC](#), a dataset containing sediment community oxygen consumption rates
see the paper of Hansen et al. 1997 for a description of the original literature sources of this dataset

Examples

```
ii <- which(Zoogrowth[ ,2]>0)
plot(Zoogrowth[ii, 1], Zoogrowth[ii, 2], log = "xy",
     xlab = "zooplankton volume, micrometer ^ 3", ylab = "" ,
     main = "maximal growth rate, /hr", pch = 16, cex.main = 1)

ll <- lm(log(Zoogrowth[ii,2])~ log(Zoogrowth[ii,1]))
rr <- summary(ll)$r.squared
A  <- exp(coef(ll)[1])
B  <- (coef(ll)[2])
curve(A*x^B, add = TRUE, lwd = 2)
AA <- round(A*100)/100
BB <- round(B*100)/100
```

```
expr <- substitute(y==A*x^B, list(A=AA,B=BB))
text(100, .0035, expr, adj = 0)
expr2 <- substitute(r^2==rr, list(rr=round(rr*100)/100))
text(100, 0.002, expr2, adj = 0)
```

Index