Package 'sorcering'

July 30, 2021

Type Package

Title Soil Organic Carbon and CN Ratio Driven Nitrogen Modelling Framework

Version 0.9.2

Date 2021-07-14

Author Marc Scherstjanoi [aut, cre], Rene Dechow [aut]

Maintainer Marc Scherstjanoi <marc.scherstjanoi@thuenen.de>

Description Can be used to model the fate of soil organic carbon and soil organic nitrogen and to calculate N mineralisation rates. Provides a framework that numerically solves differential equations of soil organic carbon models based on first-order kinetics and extends these models to include the nitrogen component. The name 'sorcering' is an acronym for 'Soil ORganic Carbon & CN Ratio drIven Nitrogen modellinG framework'.

LazyData true

Depends R (>= 3.5.0)

License GPL (>= 2)

Imports Rcpp (>= 1.0.6), mathjaxr, Rdpack

LinkingTo Rcpp, RcppArmadillo

RdMacros mathjaxr, Rdpack

R topics documented:

C0_ex *Initial Carbon Data*

Description

Fictional initial carbon for a model with five soil pools

Usage

```
C0_ex
```

Format

A vector containing five numeric entries

See Also

sorcering.

Cin_ex *Carbon Input Data*

Description

Fictional carbon input for a model with five soil pools. Columns stand for pools and rows for simulation time steps.

Usage

```
Cin_ex
```

Format

A matrix of 5 columns and 60 rows

See Also

sorcering.

fget_A_RothC *RothC Transfer Matrix Building Function*

Description

Builds a RothC transfer matrix. Parameters taken from Coleman and Jenkinson (1996).

Usage

```
fget_A_RothC(    clay = 23.4
)
```

Arguments

clay	double. Soil clay content in %.

Value

`fget_A_RothC()` returns a 5×5 matrix that contains RothC specific carbon transfer parameters based on clay content.

Author(s)

Marc Scherstjanoi <marc.scherstjanoi@thuenen.de>, Rene Dechow

References

Coleman K, Jenkinson DS (1996). "RothC-26.3 - A Model for the turnover of carbon in soil." In Powlson DS, Smith P, Smith JU (eds.), *Evaluation of Soil Organic Matter Models*, 237–246. ISBN 978-3-642-61094-3.

See Also

sorcering.

Examples

```
fget_A_RothC(clay=30)
```

N0_ex *Initial Nitrogen Data*

Description

Fictional initial nitrogen for a model with five soil pools

Usage

```
N0_ex
```

Format

A vector containing five numeric entries

See Also

sorcering.

Nin_ex *Nitrogen Input Data*

Description

Fictional nitrogen input for a model with five soil pools. Columns stand for pools and rows for simulation time steps.

Usage

```
Nin_ex
```

Format

A matrix of 5 columns and 60 rows

See Also

sorcering.

sorcering *Soil ORganic Carbon & CN Ratio drIven Nitrogen modellinG framework*

Description

SORCERING can be used to model the fate of soil organic carbon (SOC) and soil organic nitrogen (SON) and to calculate N mineralisation rates. It provides a framework that numerically solves differential equations of SOC models based on first-order kinetics. Thus, SOC models can be simply defined and run to predict the temporal development of SOC. Beyond this, SORCERING determines the fluxes of SON and N mineralisation / immobilisation. Basic inputs are (1) the model parameters of a given SOC model expressed as the C transfer matrix (including information on decomposition and transfer rates between model pools), (2) the initial distributions of C and N among model pools and (3) time series of C and N inputs and rate modulating environmental factors. The fourth-order Runge-Kutta algorithm is used to numerically solve the system of differential equations.

Usage

```
sorcering(    A = NULL,
              t_sim = 2,
              tsteps = "monthly",
              C0 = NULL,
              N0 = NULL,
              Cin = NULL,
              Nin = NULL,
              xi = NULL,
              calcN = FALSE,
              calcNbalance = FALSE)
```

Arguments

A	transfer matrix. Defines number of pools, decomposition and transfer rates. Must be a square matrix. n×n elements with n = number of pools. Diagonal values are decomposition rates [yr^{-1}]. Off-diagonals represent the transfer between pools [yr^{-1}].
t_sim	number of simulation time steps. Must correspond to the number of rows of Cin, Nin and xi.
tsteps	character indicating the type of simulation time steps. valid options are annually, monthly (recommended) or weekly.
C0	vector with a length equal to the number of pools. Contains initial soil organic carbon per pool [$tC\ ha^{-1}$]. If NULL, filled with zeros.
N0	vector with a length equal to the number of pools. Contains initial soil organic nitrogen per pool [$tN\ ha^{-1}$]. If NULL, filled with zeros. Only used when calcN = TRUE.

Cin	matrix with a number of columns equal to the number of pools and a number of rows corresponding to t_sim. Contains carbon input per pool and time step [$tC\ ha^{-1}$]. If NULL, filled with zeros.
Nin	matrix with a number of columns equal to the number of pools and a number of rows corresponding to t_sim. Contains nitrogen input per pool and time step [$tN\ ha^{-1}$]. If NULL, filled with zeros. Must be >0 where Cin>0. Only used when calcN = TRUE.
xi	matrix with a number of columns equal to the number of pools and a number of rows corresponding to t_sim. Contains environmental factors. If NULL, filled with ones.
calcN	logical indicating whether the development of soil organic nitrogen should be simulated.
calcNbalance	logical indicating whether the balance of nitrogen cycling should be calculated.

Details

SORCERING is a general model framework to describe soil organic carbon (SOC) dynamics and soil organic nitrogen (SON) dynamics based on models of first-order kinetics. It can be applied to any given SOC first-order kinetics model. The approach has already been successfully tested to describe SOC dynamics of Yasso (Tuomi et al. 2009), RothC (Coleman and Jenkinson 1996) and C-Tool (Taghizadeh-Toosi et al. 2014). Therefore, SORCERING is a lightweight alternative to the widely developed and multifunctional R package SoilR (Sierra et al. 2012; Sierra and Mueller 2014). Moreover, it additionally offers the possibility of modelling N immobilisation and mineralisation by enhancing given SOC models by an additional N module.

The following is a description of each element calculated, which also corresponds to the output values (see section 'Value'). For a detailed mathematical description of the SORCERING function see the extended pdf documentation at browseVignettes("sorcering").

C

SORCERING calculates SOC applying a given SOC model for every simulation time step defined by tsteps and t_sim. SOC models applied here are defined by a number of pools, each characterised by specific decomposition and turnover rates. The model-specific decomposition kinetics and SOC fluxes among pools are described by a set of partial differential equations represented by the transfer matrix A. Each row and column of A represent SOC pools. Off-diagonal elements of A describe SOC fluxes and diagonal elements describe SOC decomposition. The differential equations furthermore contain the boundary conditions Cin and xi. The change of SOC concentration in time thus is defined as:

$$\frac{dC(t)}{dt} = Cin(t) + A_e(t) \cdot C(t)$$

with

$$A_e(t) = A \cdot diag(xi(t))$$

Initial conditions must be defined for every SOC pool by C0. A description of the numerical solution can be found in the extended pdf documentation at browseVignettes("sorcering"). For

more information on the functioning and possibilities of solving first-order kinetics SOC models see Sierra et al. (2012).

N

As an extension to SOC modelling, `SORCERING` allows the modelling of SON coupled to the modelling of SOC. Its implementation is based on the following simplifying assumptions: (1) Nitrogen transfer and turnover rates are equal to carbon rates. (2) There is no N limitation in the soil, i.e. mineral N is always available for N immobilisation processes. (3) CN ratios of single pools are only affected by external inputs of N and C. The transfer of organic matter among pools does not affect CN ratios. As for SOC, the development of SON depends on initial and boundary conditions: `N0` and `Nin`. As N decomposition is proportional to C decomposition, SON is calculated based on the results of the SOC calculations and input conditions (for details see the extended pdf documentation at `browseVignettes("sorcering")`).

Nloss, Nmin, Nmin.sink<1>, ..., Nmin.sink<n>

Along with modelling SON, further quantities are determined. Nitrogen losses are calculated as:

$$Nloss(t) = N(t-1) + Nin(t-1) - N(t)$$

In contrast, mineralisation rates contain information about sources and sinks of SON. They are calculated based on the CN ratios in the pools and the turnover rates (for details see the extended pdf documentation at `browseVignettes("sorcering")`). Pool-specific N mineralisation $Nmin.sink\,\langle 1\rangle, ..., Nmin.sink\,\langle n\rangle$ and N mineralisation $Nmin$ are related the following:

$$Nmin_j(t) = \sum_{p=1}^{n} Nmin.sink\,\langle j\rangle_p\,(t)$$

for each simulation time point t, each pool $j = 1, ..., n$ and each pool $p = 1, ..., n$ and n total pools. Or in other words, the row sum of $Nmin.sink\,\langle j\rangle$ at one simulation time point equals the j^{th} column of $Nmin$ at that time point.

As changes in SON must match the sums of all mineralisation paths, the sums over soil pools of `Nloss` and `Nmin`, respectively, must be approximately equal for all simulation time points:

$$\sum_{p=1}^{n} Nloss_p(t) \approx \sum_{p=1}^{n} Nmin_p(t), \forall t \in tseq$$

A verification of this relation is given by `"Nbalance"` (see below).

Nbalance

The overall N change between two time steps is calculated as:

$$\Delta N(t) = \sum_{p=1}^{n} N_p(t-1) - \sum_{p=1}^{n} N_p(t)$$

The total system N balance serves as a verification output. Both of the following equations should give results close to zero:

$$N_{bal1}(t) = \sum_{p=1}^{n} Nin_p(t-1) + \Delta N(t) - \sum_{p=1}^{n} Nloss_p(t) \approx 0$$

$$N_{bal2}(t) = \sum_{p=1}^{n} Nin_p(t-1) + \Delta N(t) - \sum_{p=1}^{n} Nmin_p(t) \approx 0$$

Value

`sorcering()` returns a list object with components:

`C` array with a number of rows corresponding to `t_sim` and a number of columns equal to the number of pools. Contains soil organic carbon [$tC\ ha^{-1}$].

`N` array with a number of rows corresponding to `t_sim` and a number of columns equal to the number of pools. Contains soil organic nitrogen [$tN\ ha^{-1}$]. Only generated if `calcN = TRUE`.

`Nloss` array with a number of rows corresponding to `t_sim` and a number of columns equal to the number of pools. Contains nitrogen losses [$tN\ ha^{-1}$]. Only generated if `calcN = TRUE`.

`Nmin` array with a number of rows corresponding to `t_sim` and a number of columns equal to the number of pools. Contains nitrogen mineralisation [$tN\ ha^{-1}$]. If values are negative, nitrogen immobilisation exceeds mineralisation. Only generated if `calcN = TRUE`.

`Nmin.sink.1, ..., Nmin.sink.n`
arrays with a number of rows corresponding to `t_sim` and a number of columns equal to the number of pools `n`. Contain pool-specific nitrogen mineralisation sinks [$tN\ ha^{-1}$] (from the pool according to variable index [1, ..., n] to the pool according to column number). If the sink is the pool itself (index equals column number) the amount of decomposition is recorded. Only generated if `calcN = TRUE`.

`Nbalance` array with a number of rows corresponding to `t_sim` and three columns. Contains information on overall N changes in the soil between two time steps (first column) and information on total system N balance calculated based on total `Nloss` (second column) and based on total `Nmin` (third column) [$tN\ ha^{-1}$]. Only generated if `calcN = TRUE` and `calcNbalance = TRUE`.

Package Building Information

The `SORCERING` code was written in C++ using the R packages `Rcpp` (Eddelbuettel et al. 2021) and `RcppArmadillo` (Eddelbuettel et al. 2021). This documentation was built with the help of the R packages `mathjaxr` (Viechtbauer 2021) and `Rdpack` (Boshnakov 2021).

Author(s)

Marc Scherstjanoi <marc.scherstjanoi@thuenen.de>, Rene Dechow

References

Boshnakov GN (2021). *Rdpack: Update and Manipulate Rd Documentation Objects*. R package version 2.1.1, https://CRAN.R-project.org/package=Rdpack.

Coleman K, Jenkinson DS (1996). "RothC-26.3 - A Model for the turnover of carbon in soil." In Powlson DS, Smith P, Smith JU (eds.), *Evaluation of Soil Organic Matter Models*, 237–246. ISBN 978-3-642-61094-3.

Eddelbuettel D, Francois R, Allaire JJ, Ushey K, Kou Q, Russell N, Bates D, Chambers J (2021). *Rcpp: Seamless R and C++ Integration*. R package version 1.0.6, https://CRAN.R-project.org/package=Rcpp.

Eddelbuettel D, Francois R, Bates D, Ni B (2021). *RcppArmadillo: 'Rcpp' Integration for the 'Armadillo' Templated Linear Algebra Library*. R package version 0.10.4.0.0, https://CRAN.R-project.org/package=RcppArmadillo.

Sierra CA, Mueller M (2014). *SoilR: Models of Soil Organic Matter Decomposition*. R package version 1.1-23, https://CRAN.R-project.org/package=SoilR.

Sierra CA, Mueller M, Trumbore SE (2012). "Models of soil organic matter decomposition: the SoilR package, version 1.0." *Geoscientific Model Development*, **5**(4), 1045–1060. doi: 10.5194/gmd510452012, https://gmd.copernicus.org/articles/5/1045/2012/.

Taghizadeh-Toosi A, Christensen BT, Hutchings NJ, Vejlin J, Kaetterer T, Glendining M, Olesen JE (2014). "C-TOOL: A simple model for simulating whole-profile carbon storage in temperate agricultural soils." *Ecological Modelling*, **292**, 11–25. ISSN 0304-3800, doi: 10.1016/j.ecolmodel.2014.08.016, https://doi.org/10.1016/j.ecolmodel.2014.08.016.

Tuomi M, Thum T, Jaervinen H, Fronzek S, Berg B, Harmon M, Trofymow JA, Sevanto S, Liski J (2009). "Leaf litter decomposition-Estimates of global variability based on Yasso07 model." *Ecological Modelling*, **220**(23), 3362–3371. ISSN 0304-3800, doi: 10.1016/j.ecolmodel.2009.05.016, https://doi.org/10.1016/j.ecolmodel.2009.05.016.

Viechtbauer W (2021). *mathjaxr: Using 'Mathjax' in Rd Files*. R package version 1.4-0, https://CRAN.R-project.org/package=mathjaxr.

Examples

```
#EXAMPLE OF RothC application with fictional input

#1. Input

data(Cin_ex, Nin_ex, N0_ex, C0_ex, xi_ex) #fictional data
A_RothC <- fget_A_RothC(clay=30) #create transfer matrix for RothC
```

```
#2. simulation

out <- sorcering(A=A_RothC, t_sim=60, Cin=Cin_ex, Nin=Nin_ex,
N0=N0_ex, C0=C0_ex, xi=xi_ex, calcN=TRUE, tsteps="monthly")

#3. results

#output structure summary
summary(out)

#sample plot
oldpar <- par(no.readonly = TRUE) #save old par
par(mfrow=c(1,1),mar=c(4,5,2,5))
plot(rowSums(out$N),axes=FALSE, col=1, cex.lab=1.5,xlab="",ylab="",ylim=c(0,9),pch=20)
par(new=TRUE)
plot(rowSums(Cin_ex)/rowSums(Nin_ex),
  axes=FALSE,col=2, cex.lab=1.5,xlab="",ylab="",ylim=c(0,60),pch=20)
axis(side=2, pos = 0, labels = c((0:4)*1.5,"N",9),
  at=(0:6)*10, hadj=1, padj = 0.5, cex.axis=1.5,las=1,col.axis=1)
axis(side=4, pos = 60, labels = c((0:4)*10,"Cin/Nin",60),
  at=(0:6)*10, hadj=0, padj = 0.5, cex.axis=1.5, las=1,col.axis=2)
axis(side=1, pos = 0, labels =  c((0:4)*10,"t",60),
  at=(0:6)*10, hadj=0.5, padj = 0, cex.axis=1.5)
par(oldpar) #back to old par
```

---

xi_ex *Environmental Factors Data*

Description

Fictional environmental factors for a model with five soil pools. Columns stand for pools and rows for simulation time steps.

Usage

```
xi_ex
```

Format

A matrix of 5 columns and 60 rows

See Also

sorcering.

Index