

R documentation

of function `sorcering()`

from package 'sorcering'

June 24, 2021

Version	0.9.2
Title	SORCERING: Soil ORganic Carbon & CN Ratio drIven Nitrogen modelling Groundwork
Authors	Marc Scherstjanoi* & René Dechow [◇] * Thünen Institute of Forest Ecosystems, Eberswalde, Germany [◇] Thünen Institute of Climate-Smart Agriculture, Braunschweig, Germany
Maintainer	Marc Scherstjanoi <marc.scherstjanoi@thuenen.de>
LazyData	true
Depends	R (>= 3.5.0)
License	GLP (>= 2)
Imports	Rcpp (>= 1.0.6)
LinkingTo	Rcpp, RcppArmadillo

Description

With SORCERING the fate of soil organic carbon, soil organic nitrogen and nitrogen mineralisation rates can be assessed. It provides a groundwork for solving numerically differential equations of first-order kinetics pool-based soil organic carbon models and adding nitrogen as a new component to them. Basic inputs are parameters of a given soil organic carbon model and carbon and nitrogen input values. The numerical method applied is a fourth-order Runge-Kutta algorithm.

Usage

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

Arguments

A	transfer matrix. Defines number of pools, decomposition and transfer rates. Must be quadratic. $n \times n$ elements with n = number of pools. Diagonal values are decomposition rates [yr^{-1}]. Off-diagonals represent the transfer among pools [yr^{-1}].
t_sim	integer. Number of simulation time steps. If NULL, set to 2. Must correspond to the number of rows of Cin, Nin and xi.
tsteps	character. 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.
Cin	matrix with a number of columns equal to 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 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 $\text{Cin} > 0$, if calcN = TRUE.
xi	matrix with a number of columns equal to number of pools and a number of rows corresponding to t_sim. Contains environmental factors. If NULL, filled with ones.
calcN	logical. Indicates whether the development of soil organic nitrogen should be simulated.
calcNbalance	logical. Indicates whether the balance of nitrogen cycling should be calculated.

Details

SORCERING is a general model framework to describe soil carbon dynamics and soil nitrogen dynamics based on models of first order kinetics. N mineralisation and immobilisation are described under the assumption of unlimited mineral N availability. The approach has already been successfully tested to describe soil carbon dynamics of Yasso (Tuomi et al. 2009), RothC (Coleman and Jenkinson 1996) and C-Tool (Taghizadeh-Toosi et al. 2014). Those approaches were enhanced by an additional N mineralisation / N immobilisation module. SORCERING is a lightweight alternative to the widely developed and multifunctional R package SoilR (Sierra et al. 2012, Sierra and Mueller 2014) but additionally offers the possibility of modelling nitrogen immobilisation and mineralisation.

SORCERING produces a list of results. The list elements are:

- C
- N (if calcN = TRUE)
- Nmin, Nmin2, Nmin.sink.1, ..., Nmin.sink.n (if calcN = TRUE)
- Nbalance (if calcN = TRUE & calcNbalance = TRUE)

All list elements have a number of rows corresponding to t_{sim} . All list elements but Nbalance have a number of columns equal to number of pools. Nbalance has three columns. All entries are in $t \text{ ha}^{-1}$.

C

Soil organic carbon models applied here are defined by a number of pools, each characterised by specific decomposition kinetics. Decomposition kinetics and soil carbon fluxes among pools can be described by a set of partial differential equations represented by the transfer matrix A . Boundary conditions are specified by Cin . Initial conditions must be defined for every soil organic carbon pool by the vector $C0$.

The underlying equation of first-order kinetics defines the change of carbon concentration in time as:

$$\frac{dC(t)}{dt} = Cin(t) + A_e(t) \cdot C(t) \quad (1)$$

This equation is valid for scalar values of A_e , C and Cin as well as for a square matrix A_e with side length n and related one dimensional vectors C and Cin with length n , where n would be the number of soil carbon pools. Analytical solutions of this equation are exponential functions and can be very complex with A_e containing more off-diagonals, i.e. more types of carbon transfer among pools. Therefore, numerical solutions are an efficient way to solve the resulting complex equation system. To account for time dependency of Cin and A_e , Cin must be defined beforehand and A_e is influenced by the (model-specifically generated) environmental factor xi in a way that

$$\begin{aligned} A_e(t) &= (A^T \cdot xi(t))^T \\ &= A \cdot diag(xi(t)) \end{aligned} \quad (2)$$

with

$$xi(t) = (xi_1, xi_2, \dots, xi_n)(t) \quad (3)$$

In SORCERING, this equation system is solved by applying the fourth-order Runge-Kutta method:

$$C(t) = C(t-1) + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) \quad (4)$$

with

$$\begin{aligned}
 K_1 &= Cin(t-1) + A \cdot diag(xi(t-1)) \cdot C(t-1) \\
 K_2 &= Cin(t-1) + A \cdot diag\left(\frac{xi(t-1) + xi(t)}{2}\right) \cdot \left(C(t-1) + \frac{K_1}{2}\right) \\
 K_3 &= Cin(t-1) + A \cdot diag\left(\frac{xi(t-1) + xi(t)}{2}\right) \cdot \left(C(t-1) + \frac{K_2}{2}\right) \\
 K_4 &= Cin(t-1) + A \cdot diag(xi(t)) \cdot (C(t-1) + K_3)
 \end{aligned} \tag{5}$$

For more information on general functioning and other possibilities of solving first-order kinetics soil carbon models see Sierra et al. (2012).

N

Additionally, SORCERING allows the modelling of soil organic nitrogen. It is based on the following simplifying assumptions: (1) Nitrogen transfer and turnover rates are equal to carbon rates. (2) There is no nitrogen 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. Transfer of organic matter between pools does not affect CN ratios. As for carbon, the development of nitrogen depends on initial and side conditions: N_0 and N_{in} .

Given the amount of carbon decomposed

$$C_{decomp} = C_{1..n}(t-1) + Cin_{1..n} - C_{1..n}(t) \tag{6}$$

and the amount of nitrogen decomposed

$$N_{decomp} = N_{1..n}(t-1) + Nin_{1..n} - N_{1..n}(t) \tag{7}$$

in each pool 1..n between time points t and $t-1$, and assuming proportional carbon and nitrogen changes

$$\frac{C_{decomp}}{C_{1..n}(t)} = \frac{N_{decomp}}{N_{1..n}(t)} \tag{8}$$

the amount of nitrogen at each simulation time step is calculated as

$$N_{1..n}(t) = \frac{N_{1..n}(t-1) + Nin_{1..n}}{\left(\frac{C_{decomp}}{C_{1..n}(t)} + 1\right)} \tag{9}$$

$Nmin$, $Nmin2$, $Nmin.sink.1$, ..., $Nmin.sink.n$

This output contains mineralisation rates and information on the nitrogen cycling. It is generated automatically when nitrogen has been modelled. The following explanations refer to a single output time step, i.e. one row of the output.

- $Nmin$ contains the amount of nitrogen mineralisation per pool. If values are negative, nitrogen immobilisation exceeds mineralisation.
- $Nmin2$ represents the nitrogen balance of the single pools.
- $Nmin.sink_{1..n}$ contain the components p of each pool (=element) $j = 1..n$ of $Nmin$, with n being the total number of pools. If j is not equal to p , it describes the amount of nitrogen transferred from pool j to pool p . If j equals p , it describes the amount of nitrogen degraded in that pool. The sum of the elements of $Nmin.sink_j$ equals the j^{th} element of $Nmin$

Using the fourth-order Runge-Kutta Method, $Nmin$ and $Nmin.sink_{1..n}$ are calculated as follows:

$$Nmin(t) = (Nmin(t)(1), \dots, Nmin(t)(n)) \quad (10)$$

$$Nmin(t)(j) = \left(\sum_{i=1}^n Nmin_{mat}(t)(i, j) \right)_{j=1}^n \quad (11)$$

$$Nmin.sink_j(t) = (Nmin.sink_j(t)(1), \dots, Nmin.sink_j(t)(n)) \quad (12)$$

$$Nmin.sink_j(t)(p) = (Nmin_{mat}(t)(j, p))_{p=1}^n \quad (13)$$

where

$$Nmin_{mat}(t) = -\frac{1}{6}(Kn_1 + 2Kn_2 + 2Kn_3 + Kn_4) \quad (14)$$

and

$$\begin{aligned} Kn_1 &= (A \cdot \text{diag}(xi(t-1) \cdot C(t-1)))^T \cdot \text{diag}\left(\frac{1}{CN(t)}\right) \\ Kn_2 &= \left(A \cdot \text{diag}\left(\frac{xi(t-1) + xi(t)}{2} \cdot \left(C(t-1) + \frac{K_1}{2}\right)\right) \right)^T \cdot \text{diag}\left(\frac{1}{CN(t)}\right) \\ Kn_3 &= \left(A \cdot \text{diag}\left(\frac{xi(t-1) + xi(t)}{2} \cdot \left(C(t-1) + \frac{K_2}{2}\right)\right) \right)^T \cdot \text{diag}\left(\frac{1}{CN(t)}\right) \\ Kn_4 &= (A \cdot \text{diag}(xi(t) \cdot (C(t-1) + K_3)))^T \cdot \text{diag}\left(\frac{1}{CN(t)}\right) \end{aligned} \quad (15)$$

with

$$CN(t) = \begin{cases} \frac{C(t)}{N(t)}, & \forall n \in N(t) > 0 \\ n.c., & \text{otherwise} \end{cases} \quad (16)$$

and $K_1 - K_3$ taken from eq. system (5). Note that $K_{n_{1..4}}$ are matrices and $K_{1..3}$ are vectors.

N_{min2} is calculated as:

$$N_{min2}(t) = N(t-1) + N_{in}(t-1) - N(t) \quad (17)$$

As changes in nitrogen must match the sums of all mineralisation paths, the sums over soil pools of N_{min} and N_{min2} , respectively, must be equal for all simulation time points:

$$\sum_{p=1}^n N_{min_p}(t) = \sum_{p=1}^n N_{min2_p}(t) \quad \forall t \in tseq \quad (18)$$

A verification of this relation is given by $N_{balance}$ (see below).

Nbalance

$N_{balance}$ is a verification output that is only calculated when `calcNbalance = TRUE`. It consists of the columns ΔN , N_{bal1} and N_{bal2} . The latter two are based on N_{min} and N_{min2} , respectively, and should both be close to zero.

$$\Delta N(t) = \sum_{p=1}^n N_p(t) - \sum_{p=1}^n N_p(t-1) \quad (19)$$

$$N_{bal1}(t) = \sum_{p=1}^n N_{in_p}(t-1) - \Delta N(t) - \sum_{p=1}^n N_{min_p}(t) \approx 0 \quad (20)$$

$$N_{bal2}(t) = \sum_{p=1}^n N_{in_p}(t-1) - \Delta N(t) - \sum_{p=1}^n N_{min2_p}(t) \approx 0 \quad (21)$$

Package Building Information

We wrote the SORCERING code in C++ using the R packages `Rcpp` (Eddelbuettel et al. 2021a) and `RcppArmadillo` (Eddelbuettel et al. 2021b).

Example

```

#FICTIONAL 5 POOL SOIL CARBON MODEL

#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
par(mfrow=c(1,1),mar=c(4,4,1,4))
plot(rowSums(out$N),axes=FALSE, col=1, cex.lab=2,xlab="",ylab="",ylim=c(0,9),pch=20)
par(new=TRUE)
plot(rowSums(Cin_ex)/rowSums(Nin_ex),
      axes=FALSE,col=2, cex.lab=2,xlab="",ylab="",ylim=c(0,60),pch=20)
axis(side=2, pos = 0,
      labels = (0:6) 1.5, at=(0:6) 10, hadj=1, padj = 0.5, cex.axis=2,las=1,col.axis=1)
axis(side=4, pos = 60,
      labels = (0:6) 10, at=(0:6) 10, hadj=0, padj = 0.5, cex.axis=2, las=1,col.axis=2)
axis(side=1, pos = 0,
      labels = (0:6) 10 , at=(0:6) 10, hadj=0.5, padj = 0, cex.axis=2)
title(ylab="total N", line=2, cex.lab=2)
title(ylab="C input / N input", line=-30, cex.lab=2,col.lab=2)
title(xlab="time", line= 2, cex.lab=2)

```

References

- Coleman, K., Jenkinson, D.S., 1996. Rothc-26.3 - a model for the turnover of carbon in soil, in: Powlson, D.S., Smith, P., Smith, J.U. (Eds.), *Evaluation of Soil Organic Matter Models*, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 237–246.
- Eddelbuettel, D., Francois, R., Allaire, J., Ushey, K., Kou, Q., Russell, N., Bates, D., Chambers, J., 2021a. Rcpp: Seamless R and C++ Integration. R package version 1.0.6, <https://cran.r-project.org/web/packages/Rcpp/index.html>.
- Eddelbuettel, D., Francois, R., Bates, D., Ni, B., 2021b. RcppArmadillo: 'Rcpp' Integration for the 'Armadillo' Templated Linear Algebra Library. R package version 0.10.4.0.0, <https://cran.r-project.org/web/packages/RcppArmadillo/index.html>.
- Sierra, C.A., Mueller, M., 2014. SoilR: Models of Soil Organic Matter Decomposition. R package version 1.1-23, <https://cran.r-project.org/web/packages/mathjaxr/index.html>.
- Sierra, C.A., Müller, M., Trumbore, S.E., 2012. Models of soil organic matter decomposition: the SoilR package, version 1.0. *Geoscientific Model Development* 5, 1045–1060.
- Taghizadeh-Toosi, A., Christensen, B.T., Hutchings, N.J., Vejlin, J., Kätterer, T., Glendining, M., Olesen, J.E., 2014. C-TOOL: A simple model for simulating whole-profile carbon storage in temperate agricultural soils. *Ecological Modelling* 292, 11 – 25.
- Tuomi, M., Thum, T., Järvinen, H., Fronzek, S., Berg, B., Harmon, M., Trofymow, J., Sevanto, S., Liski, J., 2009. Leaf litter decomposition—Estimates of global variability based on Yasso07 model. *Ecological Modelling* 220, 3362 – 3371.