PROTOCOL

RadEro package
Version: 1.0.5

18/11/2024

Created with R v4.4.0 in Windows 11 Pro

RadEro package

Software name: RadEro
Developers: Arturo Catalá, Borja Latorre*, Leticia Gaspar* and Ana Navas**
**Soil and Water Department, Estación Experimental de Aula Dei (EEAD-CSIC), Spain*
Contact email: radero@eead.csic.es
GitHub link: https://github.com/eead-csic-eesa/RadEro
First year available: 2024
Program language: R
License: GPL

RadEro, is a straightforward model to estimate soil migration rates across various soil contexts. Building on the compartmental, vertically-resolved, physically-based mass balance model of Soto and Navas (2004, 2008), RadEro is accessible as a user-friendly R package. Input data, including ^{137}Cs inventories and parameters directly derived from soil samples (e.g., fine fraction density, effective volume), accurately capture ^{137}Cs distribution within the soil profile. The model simulates annual ^{137}Cs fallout, radioactive decay, and vertical diffusion, using a diffusion coefficient derived from ^{137}Cs reference inventory profiles. RadEro also accommodates user-defined parameters as calibration coefficients. The package, code, and test data are openly accessible for widespread use.

PROTOCOL

updated in 18/11/2024

Step 1: Installing the RadEro Package

a) **Open R**: It is recommended to use RStudio with R version 4.4.0 or later.
b) **Install RadEro:**
RadEro can be installed in two ways:
- **Install from CRAN**
- **Install from GitHub**

You can download the RadEro_1.0.5.zip file from the RadEro GitHub repository (https://github.com/eead-csic-eesa/RadEro). In this repository, you will also find a script called RadEro_QuickStart.R (Fig 1), which outlines the general steps for installation and function usage. However, it is recommended to first read this protocol to understand the details of the data, configuration input files, and results.

RadEro Functions Overview (Content from: *RadEro_QuickStart.R***)**

RadEro has two main functions:

RadEro_example (*target_dir = NULL, overwrite = TRUE****)***

The RadEro_example function copies example files (e.g., .js and .csv) from the "data" directory of the package to a specified working directory.

The function has the following arguments:

- **target_dir:** Specifies the directory where the files will be copied.
- **overwrite:** A logical argument that determines whether to overwrite existing files in the target directory. The default is TRUE.

The RadEro_example function creates two example files in the specified directory:

- "input-data_example.csv"
- "input-config_example.js"

These files serve as simple examples to demonstrate how to structure the input data and configuration files for the model. Once you understand the parameters used in these files, you can use them as templates for your own projects.

The parameter values for "k-samples" and "e-samples" in the "input-config_example.js" file are set to low defaults for quick checks of model functionality. However, the resulting simulation may not be optimal. For more accurate simulations and redistribution rate estimations, it is recommended to increase the "k-samples" and "e-samples" values to higher numbers, such as "k-samples = 1000" and "e-samples = 1000".

RadEro_run (*data, config, AxisMaxValue = NULL, output_dir = NULL****)***

The RadEro_run function runs the model based on the Cs137 inventory data from soil profiles, using the configuration defined in the working directory.

The function requires the following:

- **data:** A data frame containing the Cs137 inventory data from soil profiles. This file should be located in the working directory.
- **config:** A list or configuration file that defines the parameters for the model run. This file must also be located in the working directory.
- **AxisMaxValue**: This is an optional numeric value that specifies the maximum value for the axes in the plot. The default is NULL.
- **output_dir:** Specifies the directory where the output files will be stored.

To run the RadEro_run function at a given location, both the input data and configuration files must be in the same directory.

The function generates a "results" folder containing the following (**MORE INFO AT THE END OF THIS PROTOCOL**):

- **ID_plot.png:** Experimental (blue) and simulated (red) inventory depth profile plots.
- **ID_tempfile:** The experimental inventory (Bq/kg) per cell unit in the defined profile depth.
- **results.txt:** A summary file containing the following information.

Additionally, a "temp" folder is created for advanced users, which contains configuration and inventory files related to individual analyses for each profile.

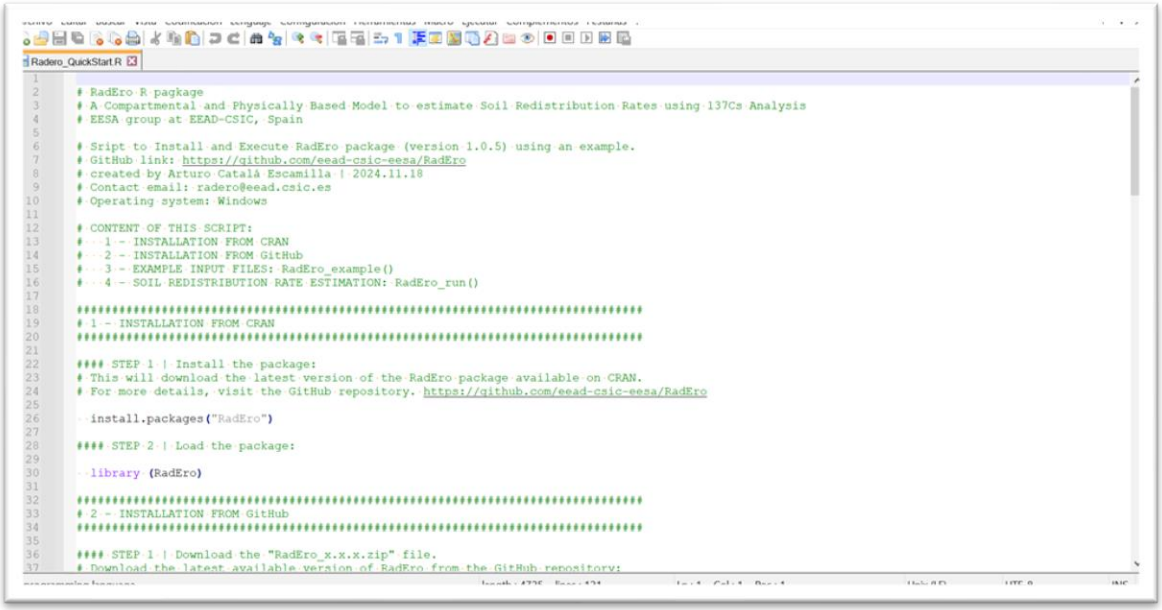

Fig 1. Screenshot of the *Radero_QuickStart.R* content.

Step 2: Creation of input data file

To work more practically and intuitively with cells, we recommend opening the *.csv* files in Excel, transforming them as needed, and then exporting them back to *.csv* format.

The template *input-data.csv* shows the structure for the input data, with **predefined columns that MUST NOT be renamed or reordered**:

- **Id**: Sample identification
- **depth_i**: Initial depth of the interval (meters)
- **depth_f**: Final depth of the interval (meters)
- **Cs137_invt**: ^{137}Cs inventory for the interval (Bq/m²)
- **Ref_Cs137_invt**: Reference ^{137}Cs inventory in the study area (Bq/m²)
- **effVol**: Effective volume of the fine fraction at the point (cm³)

Calculate Effective Volume (Veff) and Fine Particle Density

Veff (cm³/cm³) = Volume of Fines (cm³) / Total Volume (cm³)

Volume of Fines (cm³) = Total Volume (cm³) - Volume of Stones (cm³)

Total Volume (cm³) = Sample Area (cm²) × Interval Depth (cm)

Volume of Stones* (cm³) = Weight of Stones (g) / Density of Stones (2.65 g/cm³)

**Term "Stones" refers to the sample fraction > 2mm.*

For **Sectioned Profiles**: Calculate Veff for each interval and use the average value.
For **Bulk Profiles:** Use the Veff of the entire profile.

- **density**: Density of the fine fraction at the point (g/cm³)

Calculate Fine Particle Density

No Cultivated:

Density of Fines = Weight of Fines (g) / Volume of Fines (cm³)

Sectioned Profiles: Use the density of the first interval. The first interval is affected by soil redistribution processes.
Bulk Profiles: Use the density of the entire profile.

Cultivated:

Density of Fines = Weight of Fines (g) / Volume of Fines (cm³)

Density = Weight of Fines (g) / Total Volume (cm³)

Explanation: Density is calculated as the weight of fines divided by the total volume of the "device." If stoniness is low, this density is practically equivalent to the one considering the volume of fines.

Sectioned Profiles: Use the average density of the intervals included within the ploughing depth defined. Ploughing depth is considered because density is uniform within this mixed depth. Below the plow, the soil profile remains undisturbed, and the radioisotope decreases exponentially with depth.
Bulk Profiles: Use the density of the entire profile.

- **k**: The diffusion coefficient, representing the vertical extent of soil affected by diffusion processes (cm²/month), calculated from sectioned profiles. There is a significant difference between natural, undisturbed profiles and cultivated profiles, with the latter having much lower values. Leave this blank if it is a unploughed sectioned reference profile.

The ^{137}Cs method relies on comparing the measured soil ^{137}Cs inventories (total activity per unit area) at a sampling site with a measured inventory from an undisturbed reference site. The reference or control site, which has not experienced erosion or deposition, represents the initial ^{137}Cs deposited in the study region after accounting for its natural decay.

- **e**: Leave this blank; the model calculates it. It represents the migration rate (cm/month), quantifying the net vertical soil movement per month (positive for gain, negative for loss).

- **mix_depth_i**: Upper plough depth (0 meters). Leave this blank if it is an unploughed profile.
- **mix_depth_f**: Maximum plough depth (meters). Leave this blank if it is an unploughed profile.
- **mix_year_i**: Start year of ploughing within the period of ^{137}Cs fallout (minimum year 1954). Leave this blank if it is an unploughed profile.
- **mix_year_f**: End year of ploughing and abandonment of the fields. Is there is any abandonment, this final year is the sampling year. Leave this blank if it is an unploughed profile.
- **mix_frequency**: Annual ploughing frequency (normally set to 1, indicating one ploughing per year). Leave this blank if it is an unploughed profile.
- **lower_boundary**: Total depth used for model calculations (meters). This should be at least 10 cm deeper than the sampling depth. Unless there are impermeable layers, this depth should exceed the sampling depth to avoid boundary condition influence on the calculation. For example, if the sampling depth is 50-60 cm, set the lower boundary to 1.0 meters. Accuracy is not lost by having the boundary condition at a greater depth than the sampling depth.

The *input-data.csv* template contains four example data types: two with sectioned profiles and two with bulk profiles.

- **Ex1**: Sectioned profile, unploughed
- **Ex2**: Sectioned profile, ploughed
- **Ex3**: Bulk profile, unploughed
- **Ex4**: Bulk profile, ploughed

For **sectioned profiles**, the Ref_Cs137_invt, effVol, density, k, e, mix_depth_i, mix_depth_f, mix_year_i, mix_year_f, mix_frequency and lower_boundary of the sampling point should be recorded in the final interval of the profile.

For **bulk profiles**, all data should be recorded in a single row.

Step 3: Creation of input configuration file

The template *input-config.js* shows the structure for the input configuration of the model. Parameter fields name **MUST NOT be renamed or reordered**. The **values that can be edited** by the user are marked in **blue** below. For each input data file, there is a corresponding configuration file. If any of the following parameters are not defined in the input data file — Ref_Cs137_invt, effVol, density, k, e, mix_depth_i, mix_depth_f, mix_year_i, mix_year_f, mix_frequency, and lower_boundary— the model will use the values from this general

configuration file. However, we recommend defining these parameters in the input data file for each individual point.

```
 "simulate-until-year"   : 2003.0,        (Year until which the simulation runs)

 "tracer-lifetime"       : 28.8811,       (Radioisotope decay constant, e.g., 137Cs)

 "soil-thickness"        : 0.60,          (Input data parameter = lower_boundary)
 "soil-effective-volume": 1.0,            (Input data parameter = effVol)
 "soil-density"          : 1.0,           (Input data parameter = density)

 "cell-thickness"        : 0.01,          (Model resolution: recommended: 0.01 m)
 "numerical-cfl"         : 0.5,           (Diffusion velocity. Dimensionless time step between 0
and 1)

 "mix-from-depth"        : 0.0,           (Input data parameter = mix_depth_i)
 "mix-to-depth"          : 0.0,           (Input data parameter = mix_depth_f)
 "mix-interval"          : 1.0,           (Input data parameter = mix_frequency)
 "mix-inicial-year"      : 0.0,           (Input data parameter = mix_year_i)
 "mix-final-year"        : 0.0,           (Input data parameter = mix_year_f)

 "fallout" :             (Specific for each latitude, e.g., 137Cs in the Northern Hemisphere)
  {
   "initial-year"        : 1954.0,        (Initial 137Cs fallout year)
   "final-year"          : 1983.0,        (Final 137Cs fallout year. If Chernobyl (1986) is
influential, add years to the curve and additional deposition data)

   "mix-depth"           : 0.01,          (Fallout depth of 137Cs. Although in mm, use the
maximum cell size)

   "curve" :             (Annual fallout data available, e.g., Northern Hemisphere)
    [
      50.0,  150.0,  170.0,  190.0,  320.0,
     350.0,  100.0,  140.0,  530.0, 1220.0,
     670.0,  290.0,  180.0,   70.0,   70.0,
      40.0,   70.0,   70.0,   40.0,   20.0,
      40.0,   20.0,   20.0,   30.0,   30.0,
      10.0,   10.0,   20.0,   10.0,   10.0
    ],
   "reference-inventory":1570.0       (Input data parameter = Ref_Cs137_invt)
  },

 "optimization" :        (Sampling domain parameters. READ "Optimization parameters: k and
e")
  {
   "k-initial"           : 0.00,          (Lower limit of the k value range. See Fig.2)
   "k-final"             : 0.40,          (Upper limit of the k value range. See Fig.2)
   "e-initial"           : -0.02,         (Lowe limit of the e value range. See Fig.2)
   "e-final"             : 0.02,          (Upper limit of the e value range. See Fig.2)
Samples with high erosion or deposit rates, e range limits defined could be small. In that case,
the e value in the output results will be equal to the lower e range limits defined. You has to
```

increase the limits of the e value range(e-initial and e-final) to be able to determine higher redistribution rates.

WARNING: *Note that if a sample has 0 inventory the ^{137}Cs mass balance model can't be used. If you introduce a point with no inventory, the model will produce an inaccurate result and an e value in the output that is equal to the upper limit of the e range defined. Don't confuse this with the previous case.*

"k-samples" : **100**, *(Model resolution that defines the model iterations*
"e-samples" : **100** *To determine the k value in reference profiles, e.g., a resolution of either 100 x 100 (1% error) or 1000 x 1000. To determine erosion and deposition rates with an already defined k (k-samples will be 1), the e-samples parameter is the only one that can be adjusted, e.g., e-samples = 1000 iterations typically yield reliable results.)*

WARNING: *High k and e samples implies high resolutions, which can result in long computation times for the model.* ***Before performing high-resolution analysis, try conducting an exploratory test with a low resolution, such as 100x100.***

}
{

Optimization parameters: k and e

Soil redistribution rates are determined by comparing a simulated inventory, generated by the model, to a stable reference inventory assuming no erosion or deposition. Optimal model performance is achieved when the simulated inventory closely matches the measured inventory. Each simulated inventory is defined by two model parameters: the migration rate e and the diffusion coefficient k.

The model identifies the optimal simulated inventory by determining specific k and e values that best fit the input data. To achieve this, a Monte Carlo method is applied to a user-defined **sampling domain of k and e values**. Through numerous iterations, the Monte Carlo simulation generates a finite set of potential inventory values within this defined space (Fig. 1). The model then selects the value closest to the measured inventory, enabling the calculation of the soil migration rate (Mg ha^{-1} yr^{-1}).

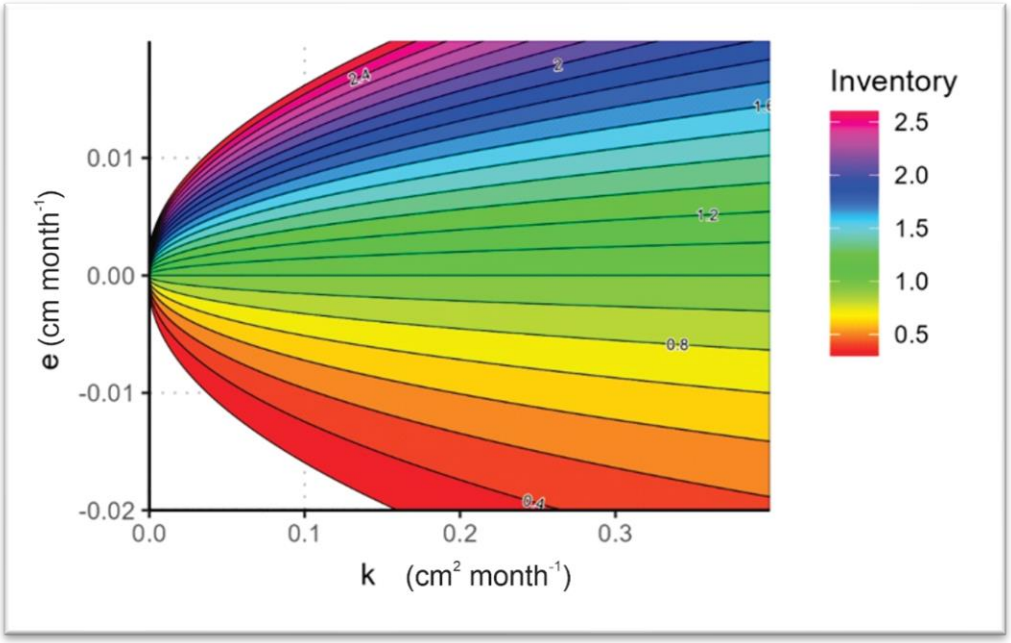

Fig.2 Graphical representation of a sampling domain defined by the range of values for diffusion coefficient k (cm^2 $month^{-1}$), and migration rate e (cm $month^{-1}$). The model simulates a finite number of potential inventory values within this domain based on the specified number of iterations. Isolines and color-coded areas represent dimensionless inventory values across the workspace.

Before calculating rates for samples, whether bulk or sectioned, it is essential to determine the representative diffusion coefficient for the study area. This coefficient is calculated in the model with input reference profiles that are sectioned and unploughed. In the input file of a reference profile, the columns for `k` and `e` will be left blank. The sampling domain (ranges of the possible `k` and `e` values where the model can iterate) must be defined by the user in the "optimization" (In Fig.2 those ranges are represented in the axis).

Step 4: Results files

Once the calculations are finished, the model will generate the results in the defined working directory (the ***project folder*** defined on the **Step 1**). The example_solved folder demonstrates how the results files are created.

"results" folder: Each point analysed will generate a profile plot.png and a temp folder with is corresponding ID. In addition, the whole set of results of all the points will be reported in a results.txt file.

- ID_plot.png : Experimental (blue) and simulated (red) inventory depth profile plots. To represent both inventories on the same scale, they are plotted in Bq m^{-3} per cell unit (m). The right column shows the experimental inventories in blue, displaying the values for each interval in the sectioned profiles or the single value for the bulk profiles. Total experimental and simulated inventory is display below the plot.

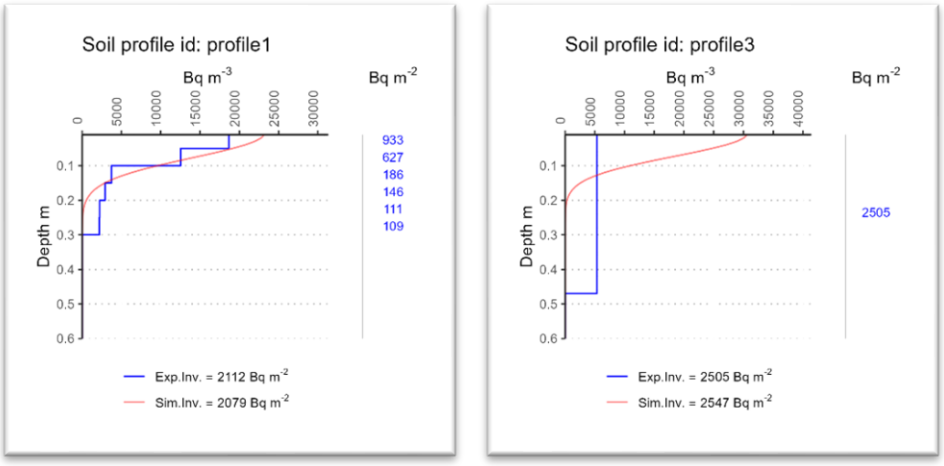

Fig.3 Example plots from the example_solved folder.

- ID_tempfile : Experimental inventory (Bq kg^{-1}) per cell unit in the profile depth defined.
- results.txt :
    - id : Point/sample/profile ID
    - e (cm $month^{-1}$) : Migration rate
    - m (Mg ha^{-1} yr^{-1}) : Erosion rate
    - k (cm^2 $month^{-1}$) : Diffusion coefficient (estimated with reference sectioned profiles)
    - kv (cm^2 $month^{-1}$) : Diffusion coefficient considering effective volume (k*Veff)
    - Experimental-inventory (Bq m^{-2}) : Total experimental inventory
    - Simulated-inventory (Bq m^{-2}) : Total simulated inventory

"temp" folder: Folder for **advanced users** in the ^{137}Cs technique and programming. This temporal folder is created for each profile and recovers temporal files of each individual analysis. All the temporal folders are saved in the results folder with its corresponding ID.

Inner files are:

- _config.js : Specific configuration file created for each sample in the input data.
- _exp.txt : Experimental inventory (Bq kg^{-1}) per cell unit in the profile depth defined.
- _num.txt : Simulated inventory (Bq kg^{-1}) per cell unit in the profile depth defined.
- _num2.txt : Simulated inventory (Bq m^{-3}) per cell unit in the profile depth defined beginning in 0.00 m.
- _num3.txt : Simulated inventory (Bq m^{-3}) per cell unit in the profile depth defined beginning in 0.01 m.
- resultstemp.txt : results file for the profile (see the **"results" folder** above)
- Other codify files created by C++ code.